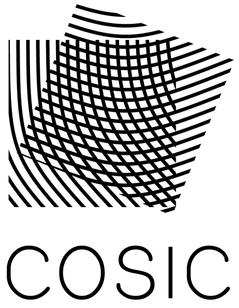
Solving Data Availability Limitations in CSV with UTxO Binding

Yunwen Liu, Bo Wang, Ren Zhang

COSIC-KU Leuven, UTxO Stack, Cryptape & Nervos



CAAW'26, 06 - Mar - 2026

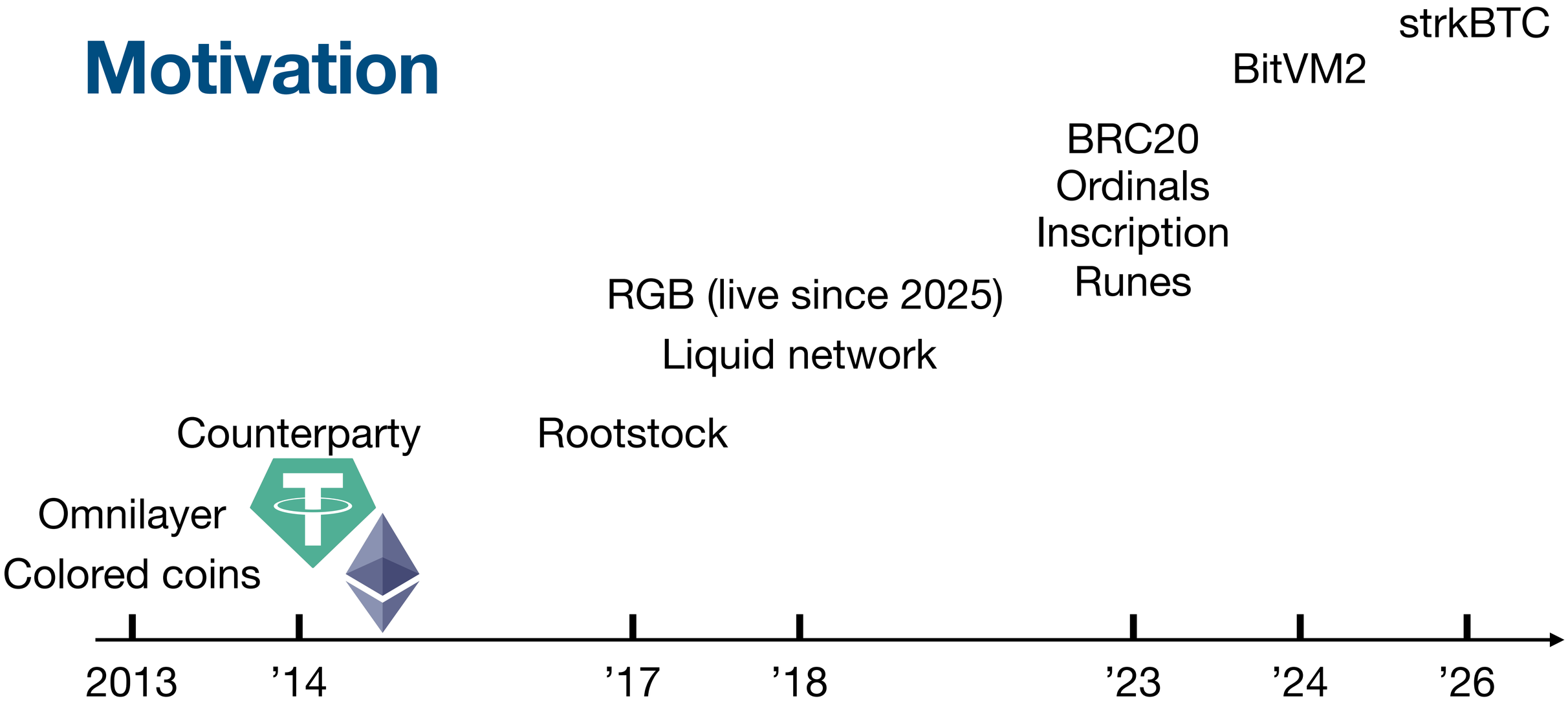
Motivation

- Bitcoin’s programmability was designed to be limited
  - No advanced smart contracts
  - No native user-defined tokens
  - Weak interoperability

Low On-chain Activities

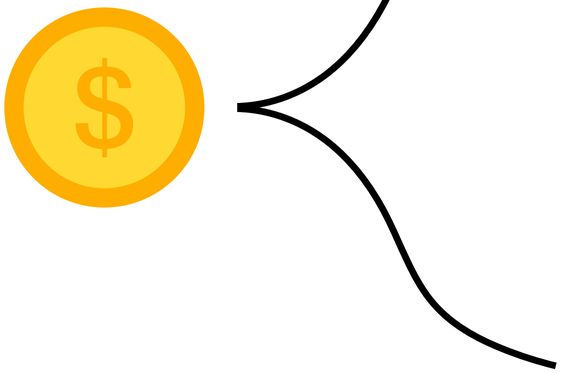
How to issue tokens and assets for DeFi applications on Bitcoin?

Motivation



A Brief Timeline of BTC-Fi

How to issue tokens on Bitcoin?



Token Carrier

Ownership

- Bitcoin UTxO
- Transfer on Bitcoin blockchain

Token Contract

Functionality

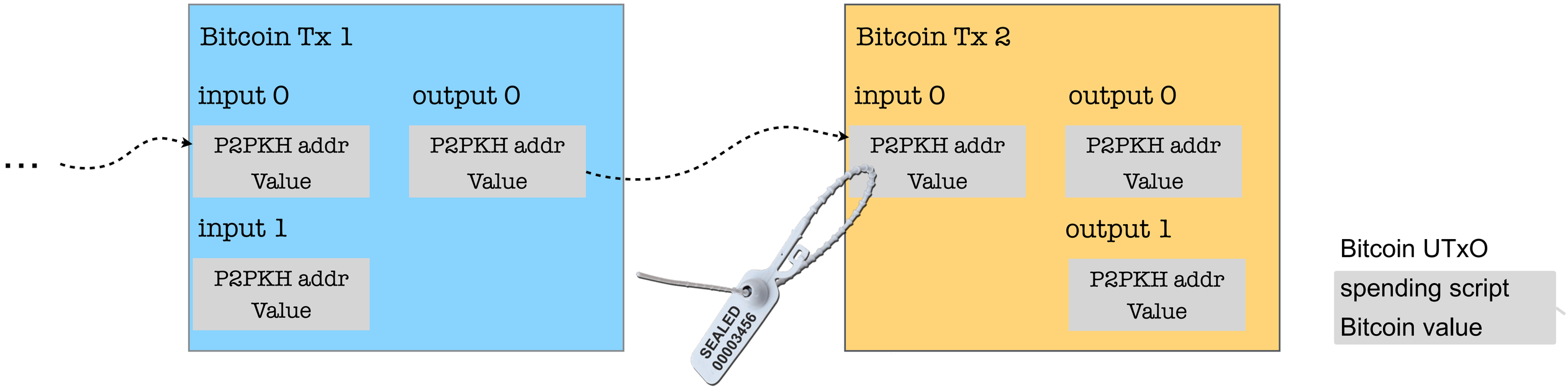
- Issuance & Trading rules
- Auxiliary data

Client-side validation

- **Idea**: Execute complex functionalities off-chain
- Tx Finality: Commit token transfer information to Bitcoin blockchain
- Users maintain Tx data **locally** —— Client-side **Double-spending??**
- Observers on-chain will only see ordinary Bitcoin transactions
  - details of the transferred user-defined tokens are oblivious to them

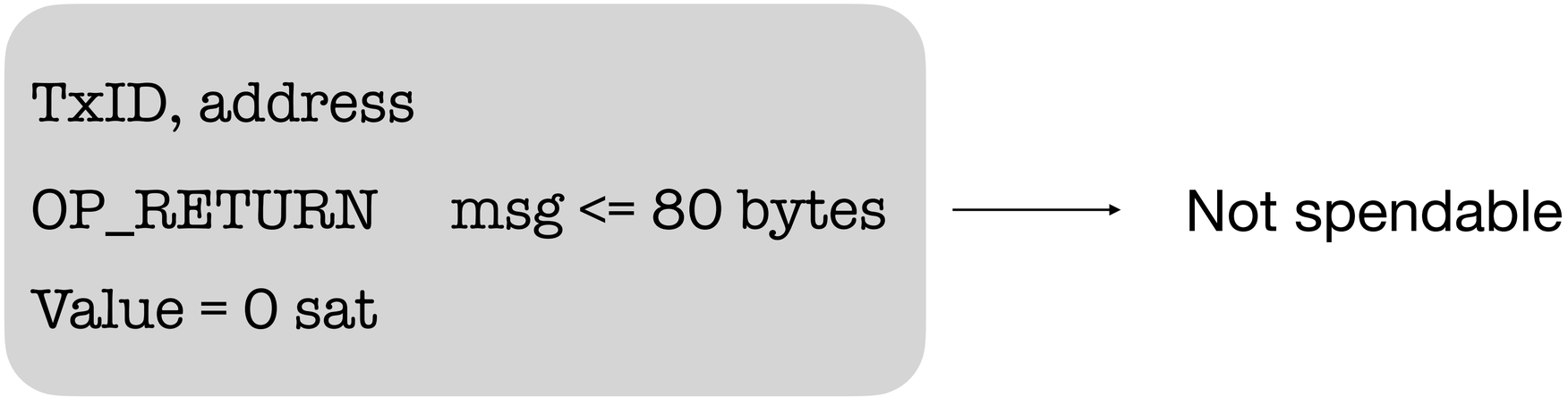
Client-side validation - linearity guarantee

- Todd's single-use seal, 2016
  - Each UTxO can only be spent on-chain once
  - Attach a single-use seal to an UTxO, *token-binding UTxO*, so that spending the UTxO means opening the seal



Client-side validation - OP_RETURN

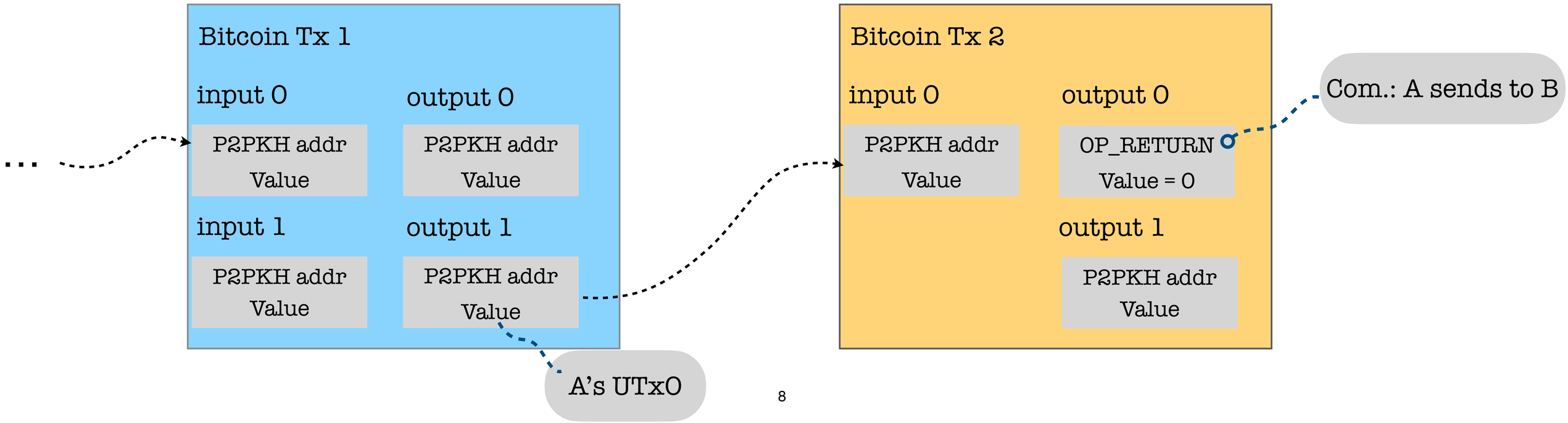
- Verification on the client side (off-chain)
  - How to write information on Bitcoin's UTxOs?
  - How to maintain issuance and transaction data off-chain?



Output UTxO with OP_RETURN script

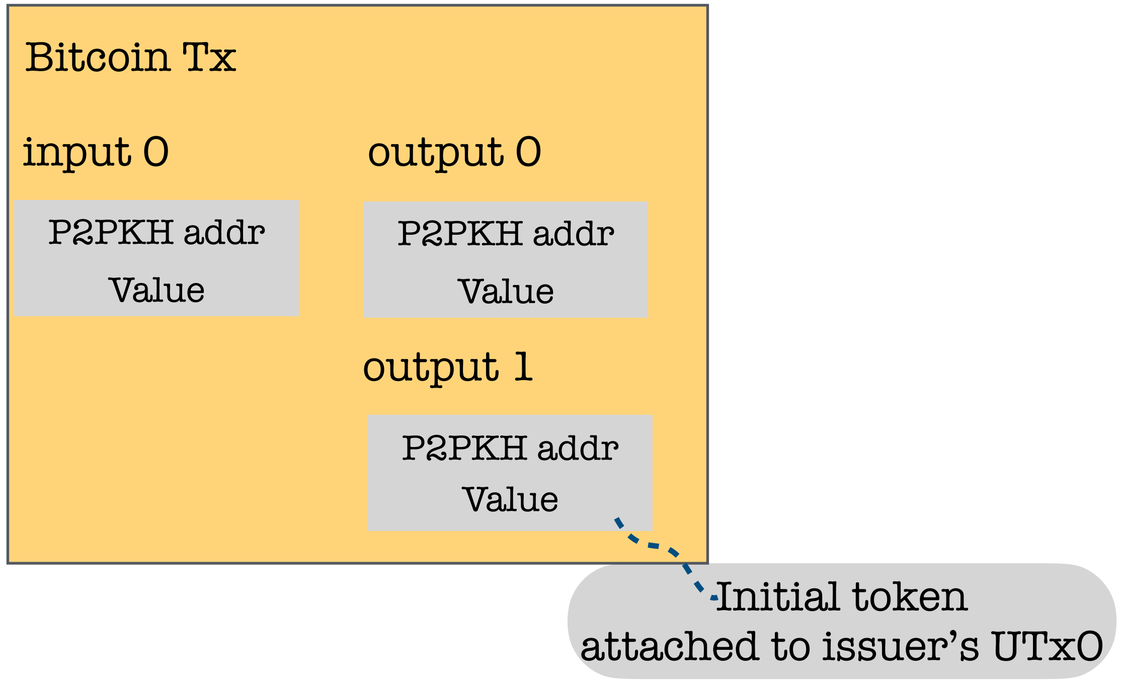
Client-side validation - RGB protocol

- Bob generates Seal_B = H(txid_B, index_B, salt) with an existing UTxO
- Alice computes Commitment = H(Tx(A -> seal_B)), spends her UTxO and embeds the commitment to OP_RETURN
- Alice sends transaction info Tx(A -> seal_B) to B



Client-side validation - RGB protocol

- Users store data locally: token ownership, token type, commitment, Tx history
  - *Schema*: smart contracts run on AluVM off-chain
  - *Issuance policy*: total supply, precision, spend rules
- Schema and token issuance policy stored and published off-chain



Client-side validation - Limitations

Data Availability

Users maintain their full TX history data loss = token loss

Client Coherence

Participant must apply exactly the same validation rules

Peer Discovery

Transmitting transaction history data to the recipient

State Integrity

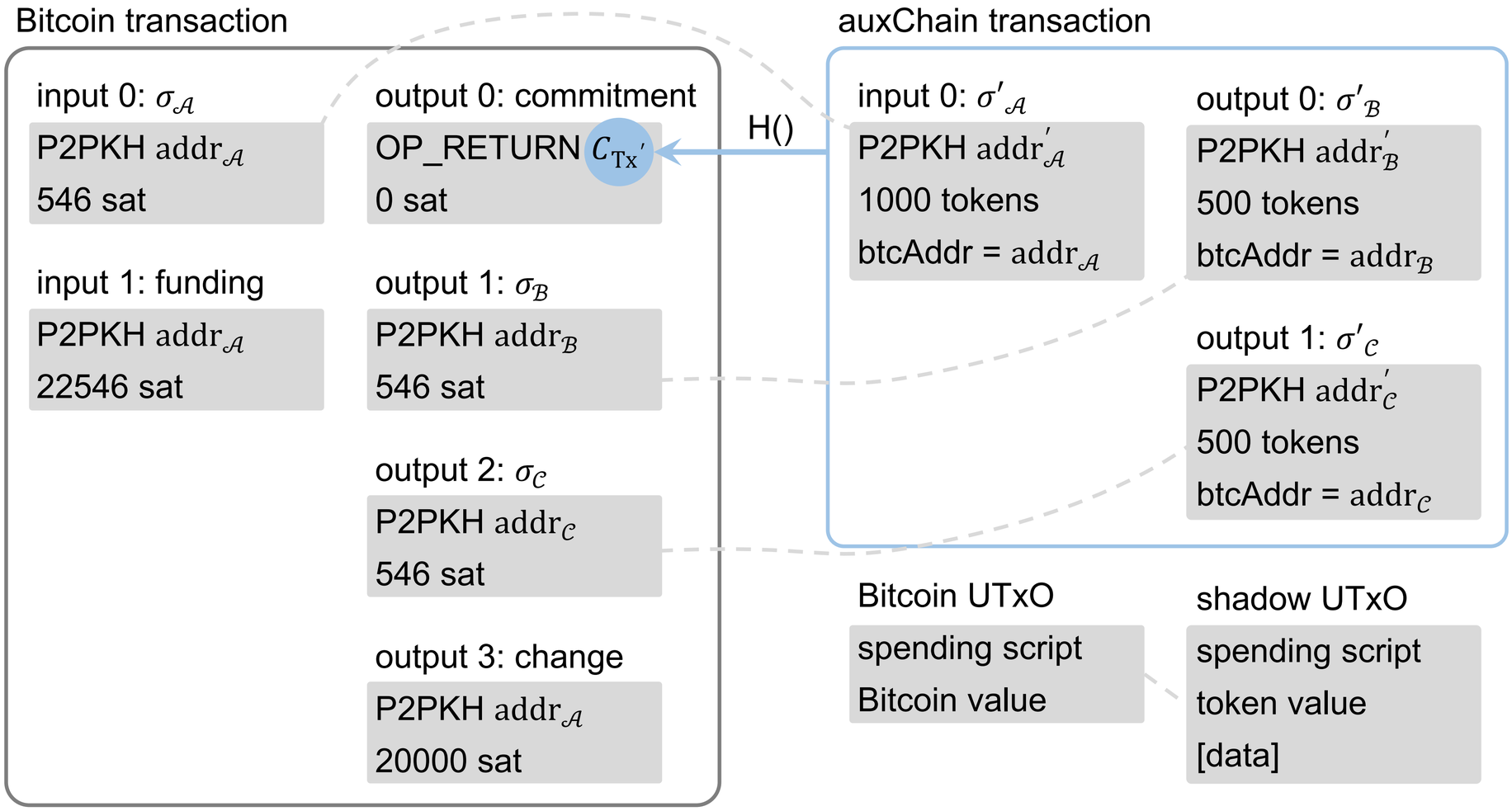
The full token ledger is *not* stored on-chain

Our solution: UTxO binding

UTxO binding - Generic design

- Shadow UTxO
  - UTxO on a shadow blockchain ***AuxChain*** that supports complex contracts
  - Not a side chain or bridge of Bitcoin
  - Properties
    - **Exclusive Binding**: a shadow UTxO is uniquely bound to a Bitcoin UTxO
    - **Unforgeability**: an attacker cannot invalidate token binding and transfer
    - **Public Verifiability**: anyone who has access to Bitcoin and AuxChain can detect double-spending or over-issuing

UTxO binding - Generic design



UTxO binding - Autonomous design

- AuxChain implements a Bitcoin light client
  - Token transaction verification can be performed autonomously on the AuxChain
  - Shadow UTxO’s locking script is conditioned on the spending status of its bitcoin UTxO
  - A new shadow UTxO’s token property must replicate its ancestor

Implementation

- We implemented autonomous UTxO binding on CKB, a UTxO-model Turing complete chain
  - The main computational and economic cost to use UTxO binding is on Bitcoin's side
    - Each Bitcoin token-binding UTxO must carry 546 satoshis ≈ 0.4 USD
    - Transaction fee on Bitcoin >> transaction fees on most other blockchains that can function as AuxChain
    - Transaction latency: ~ Bitcoin's block time

Discussions

- Comparing UTxO binding with existing Bitcoin layer 2 protocols

	Data storage	Programmability	Privacy	Deployment
Colored Coins, Omni Layer, Counterparty	Bitcoin	✗	○ public	✓
Ordinals	off-chain	✗	○ public	✓
RGB, Taproot Assets	off-chain	✗	◐ selective	✓
Intmax2, Shielded CSV	off-chain	✗	● high	✗
Lightning Network	off-chain	✗	◐ off-chain	✓
UTxO Binding	Nervos CKB	✓	○ public	✓

Conclusion

- Solving data availability problem in CSV with UTxO binding technique
  - UTxO binding is **not** a side chain or bridge
  - Generic and autonomous UTxO binding
  - Protocol specification and implementation in full version ☞
- Open questions
  - Privacy-enhancing binding to hide transaction history?
  - Extend to cross-chain swaps?

Thank you!

COSIC

CRYPTAPE

NERVOS