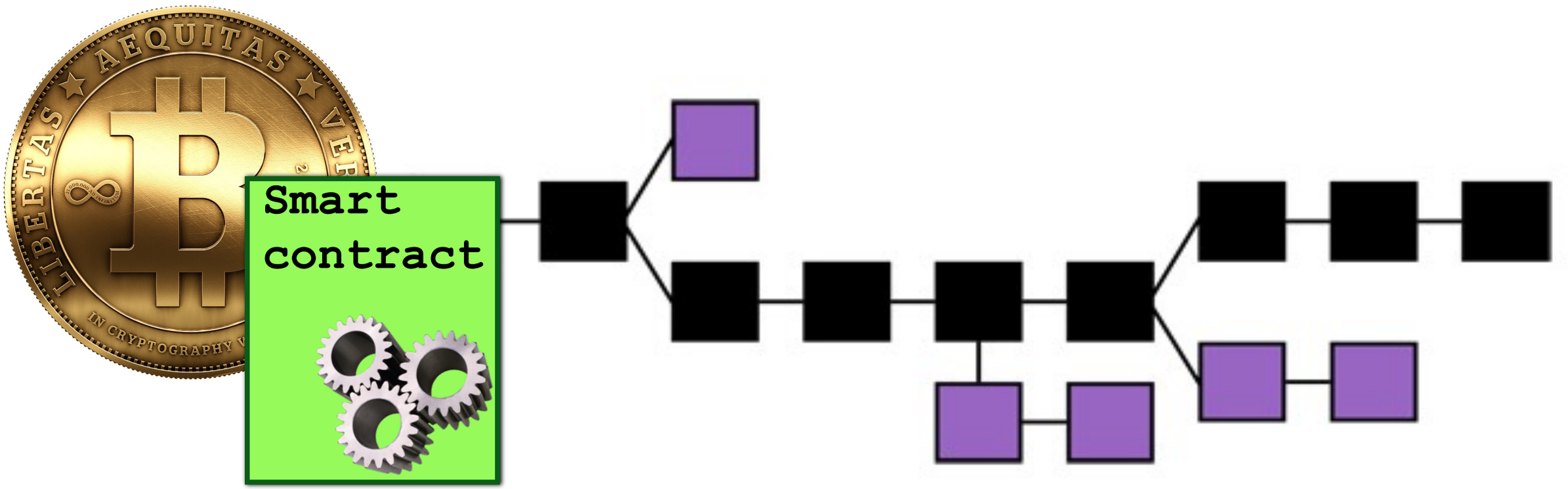
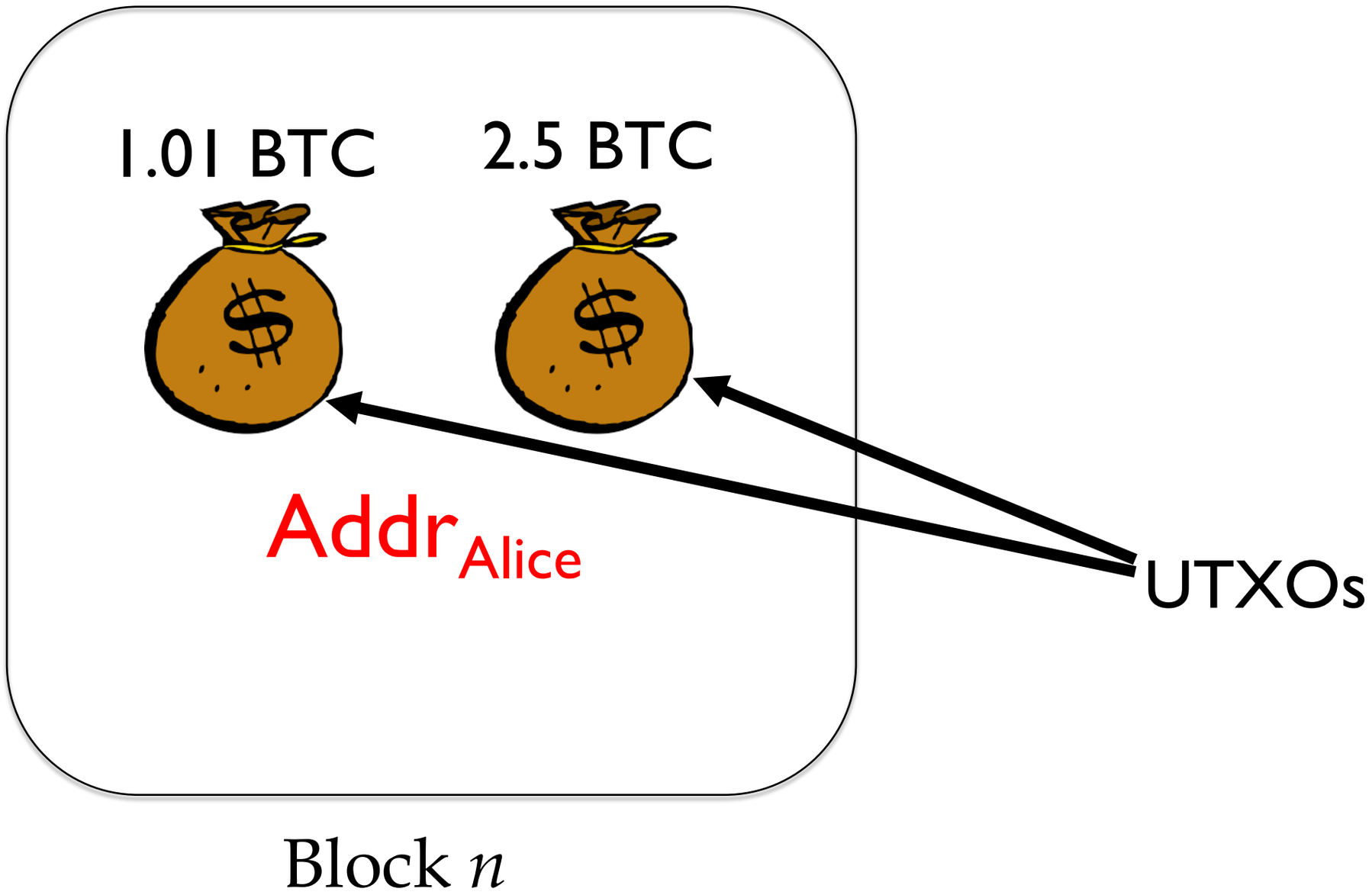
NBAY 5710:
Cryptocurrencies and Blockchains



Lecture 6: Bitcoin UTXOs and Scripts

Instructor: Ari Juels
Spring 2024

The UTXO model
(Unspent Transaction Outputs)



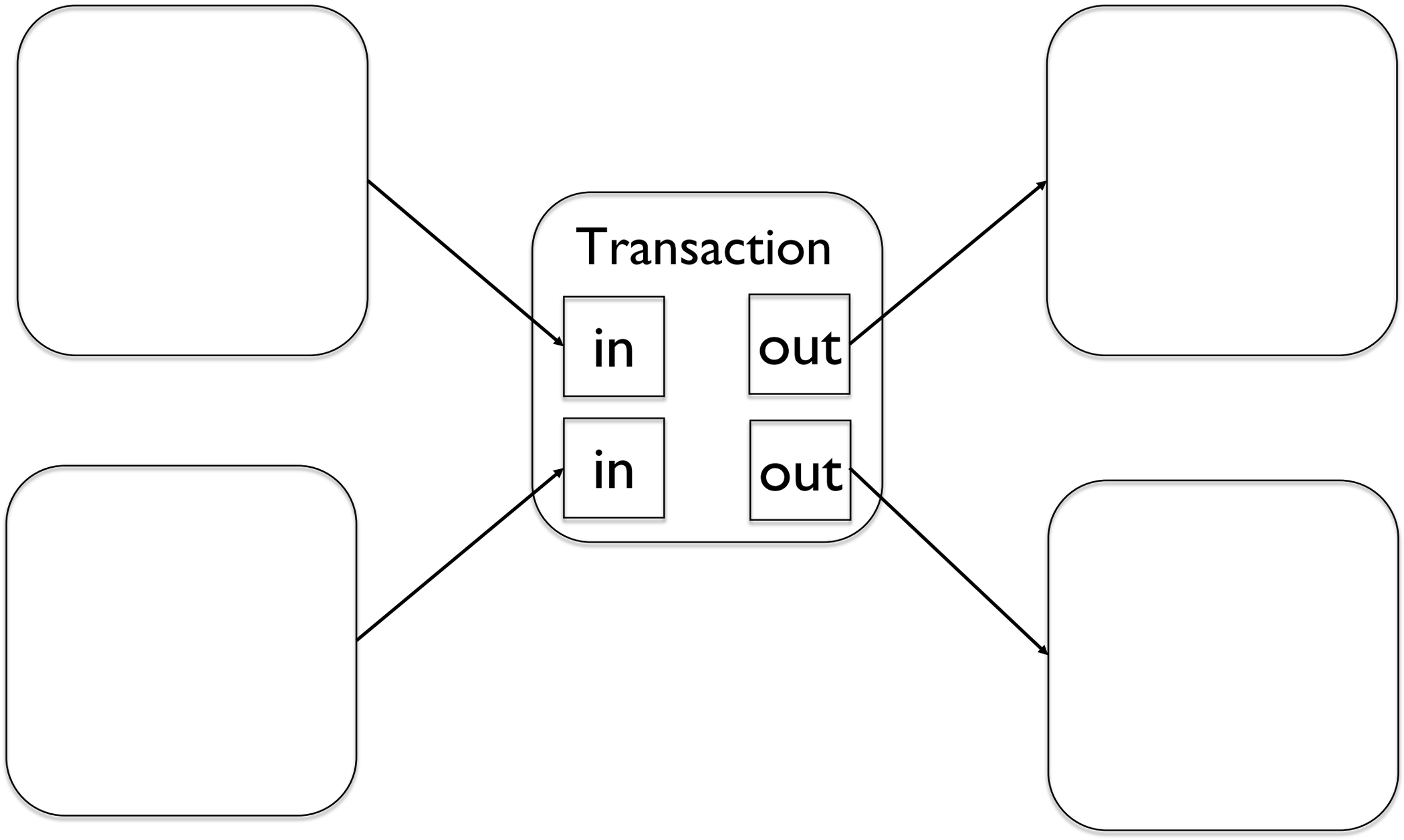
Block *n*

In Bitcoin

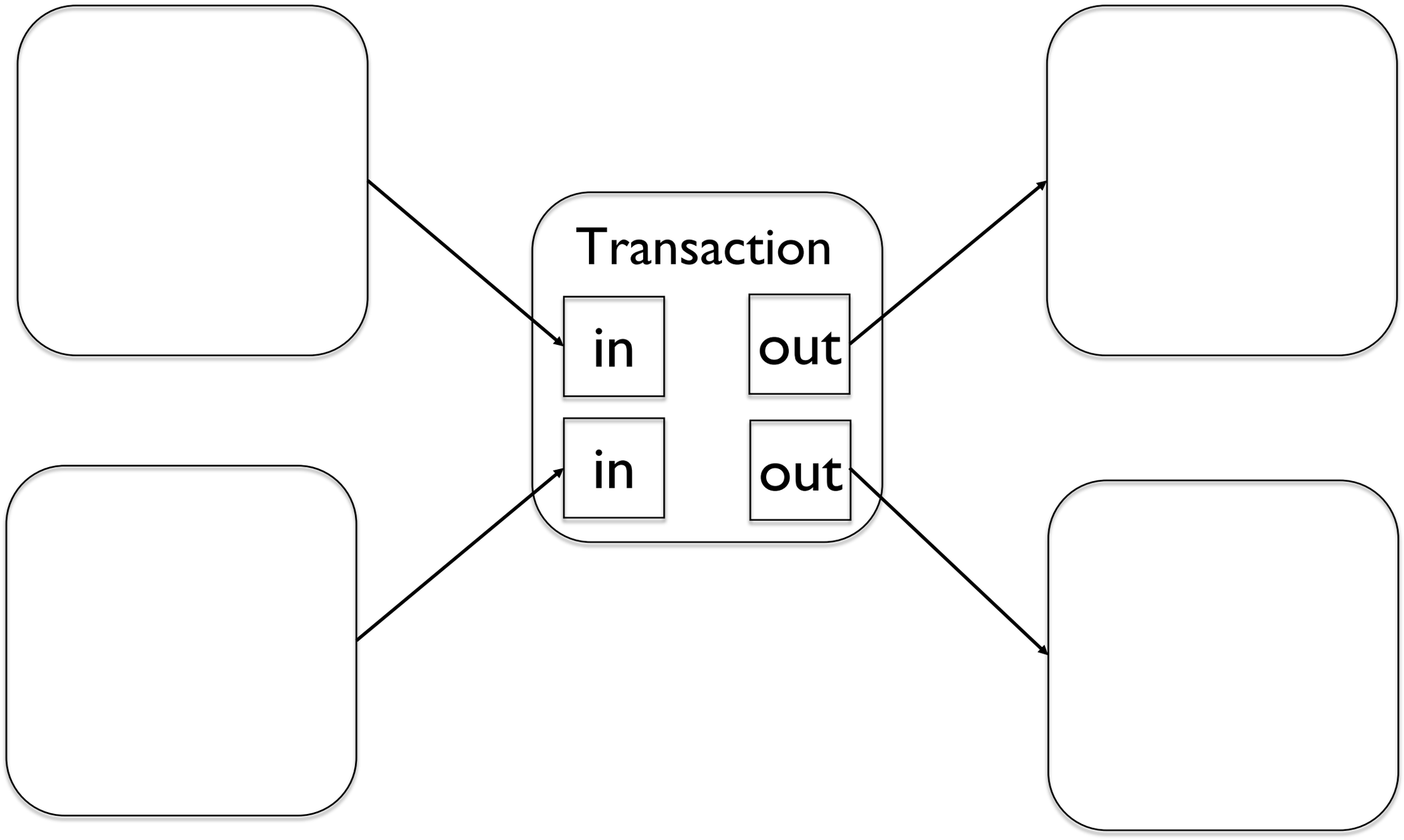
- **No explicit balances**
- Only a set of *transactions*
- Circulating money consists of…

Unspent Transaction Outputs (UTXO)

Bitcoin transaction structure



Bitcoin transaction structure

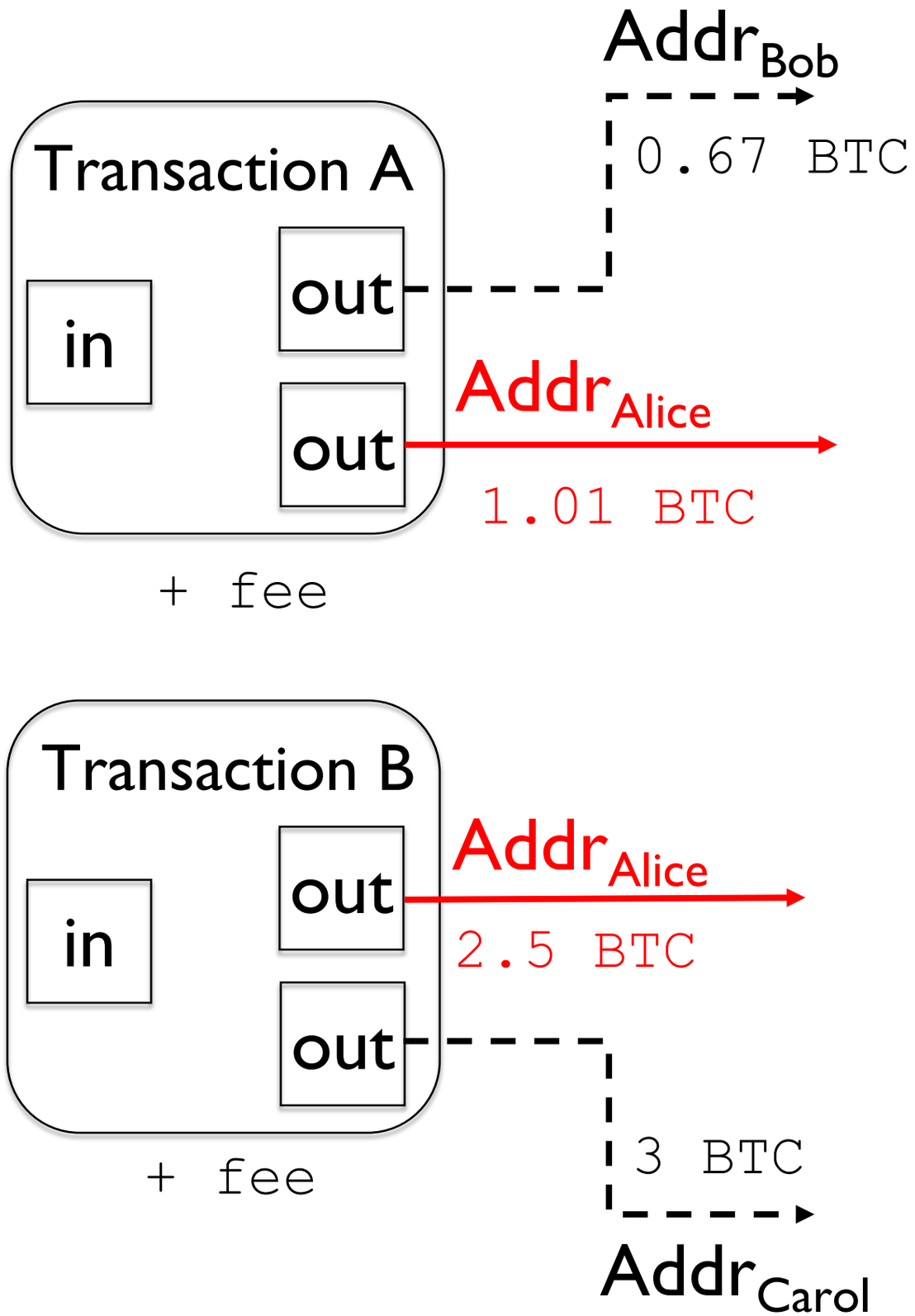


Can have arbitrarily large input and output sets…

Some record transactions…

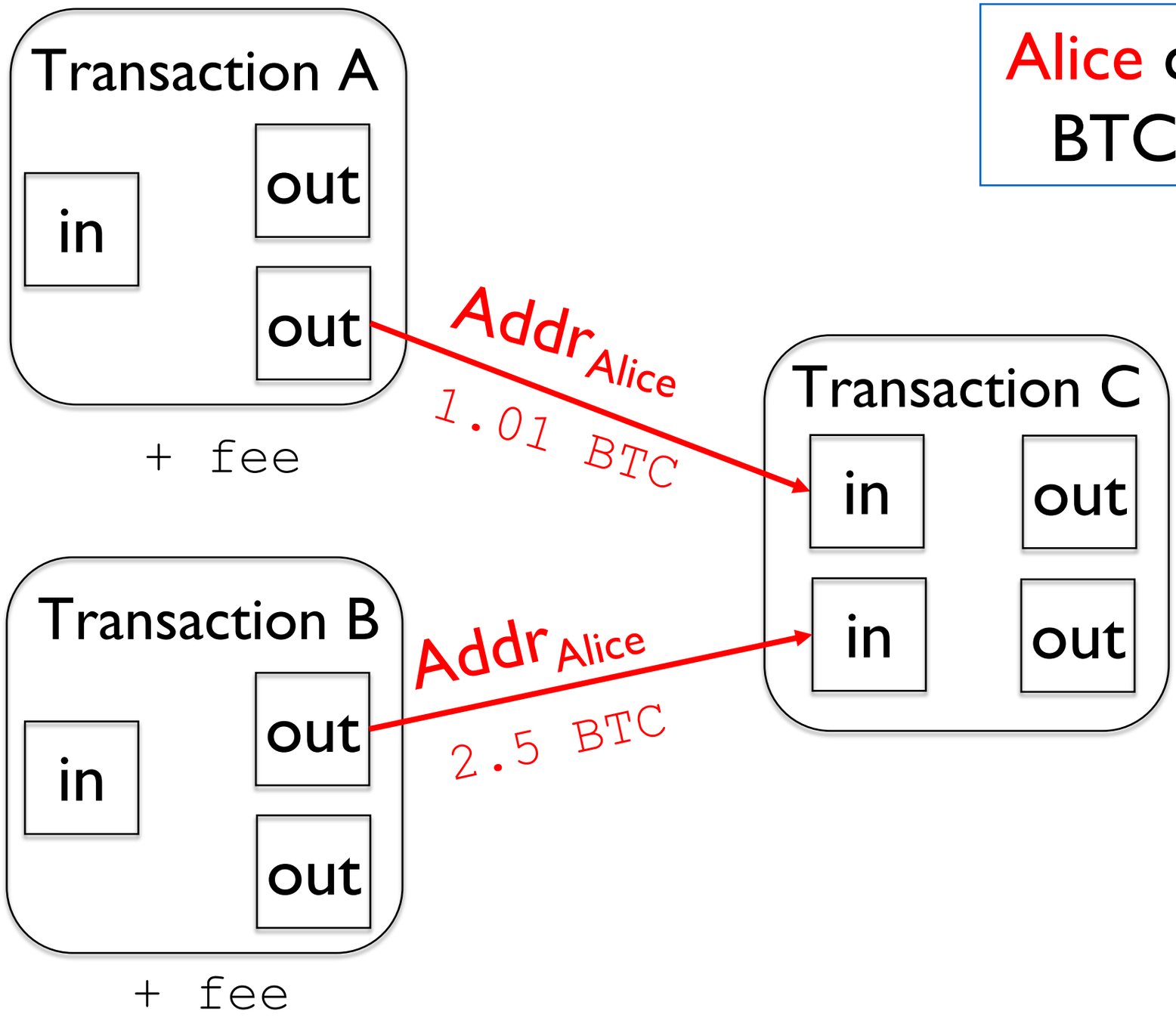
- Largest number of inputs: 2585
- Tx ID: 659135664894e50040830edb516a76f704fd2be409ecd8d1ea9916c002ab28a2
- Largest number of outputs: 3075
- Tx IDs: 623463a2a8a949e0590ffe6b2fd3e4e1028b2b99c747e82e899da4485eb0b6be and 5143cf232576ae53e8991ca389334563f14ea7a7c507a3e081fbef2538c84f6e

Transaction structure / ownership



Example: 3.51 BTC owned by Alice

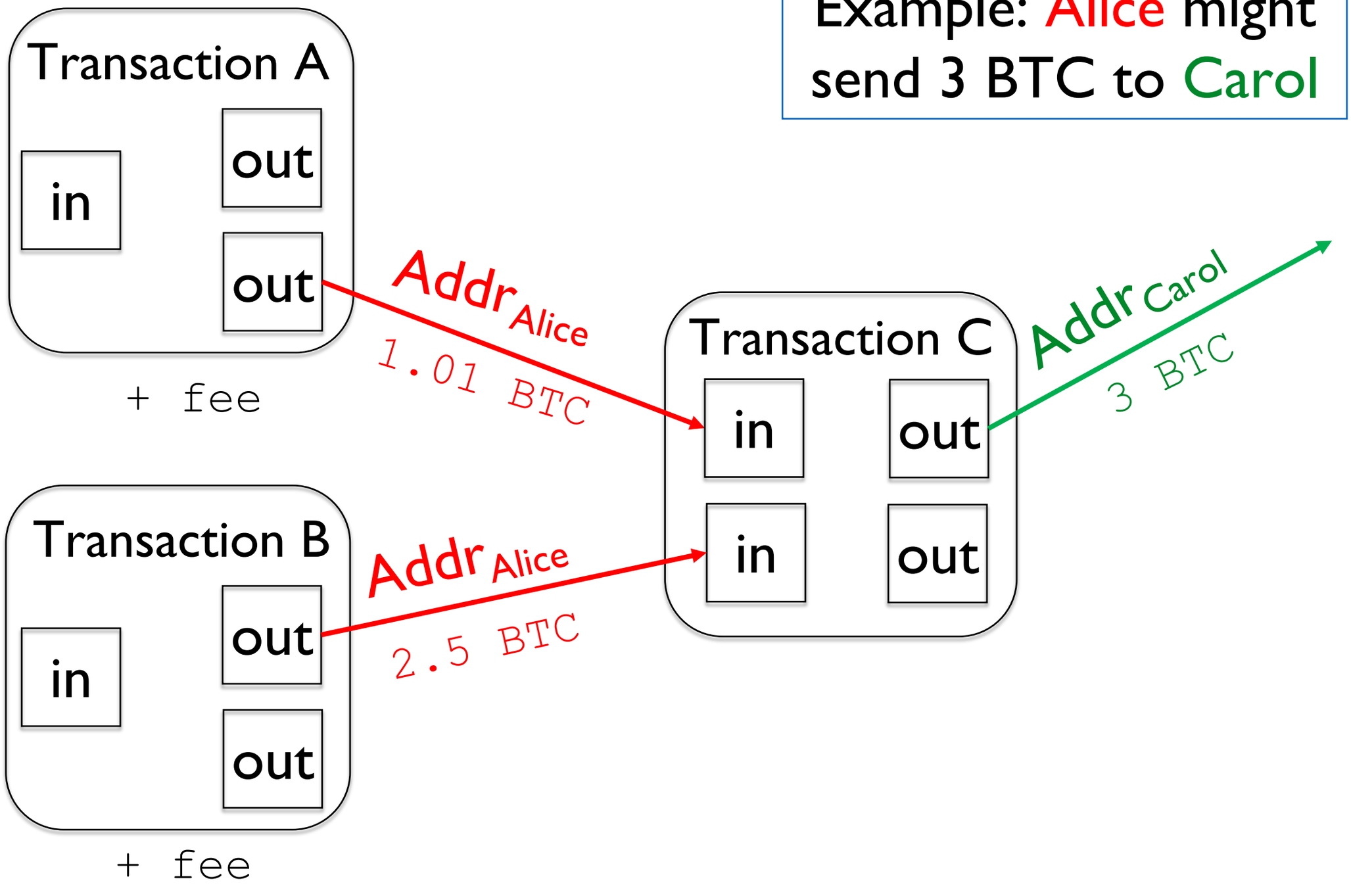
Bitcoin transaction structure



Alice can spend her BTC in a new tx

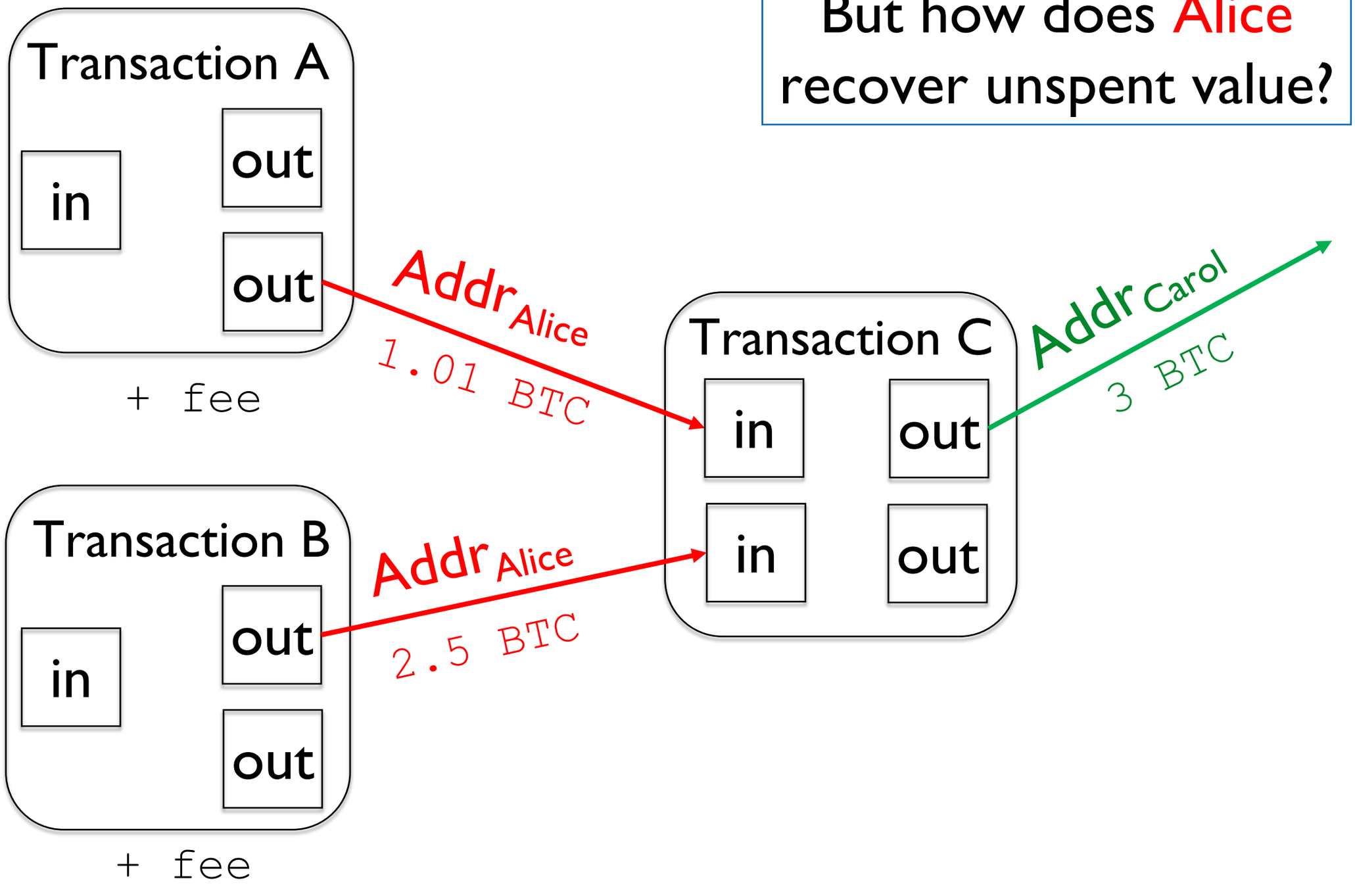
Bitcoin transaction structure

Example: Alice might send 3 BTC to Carol



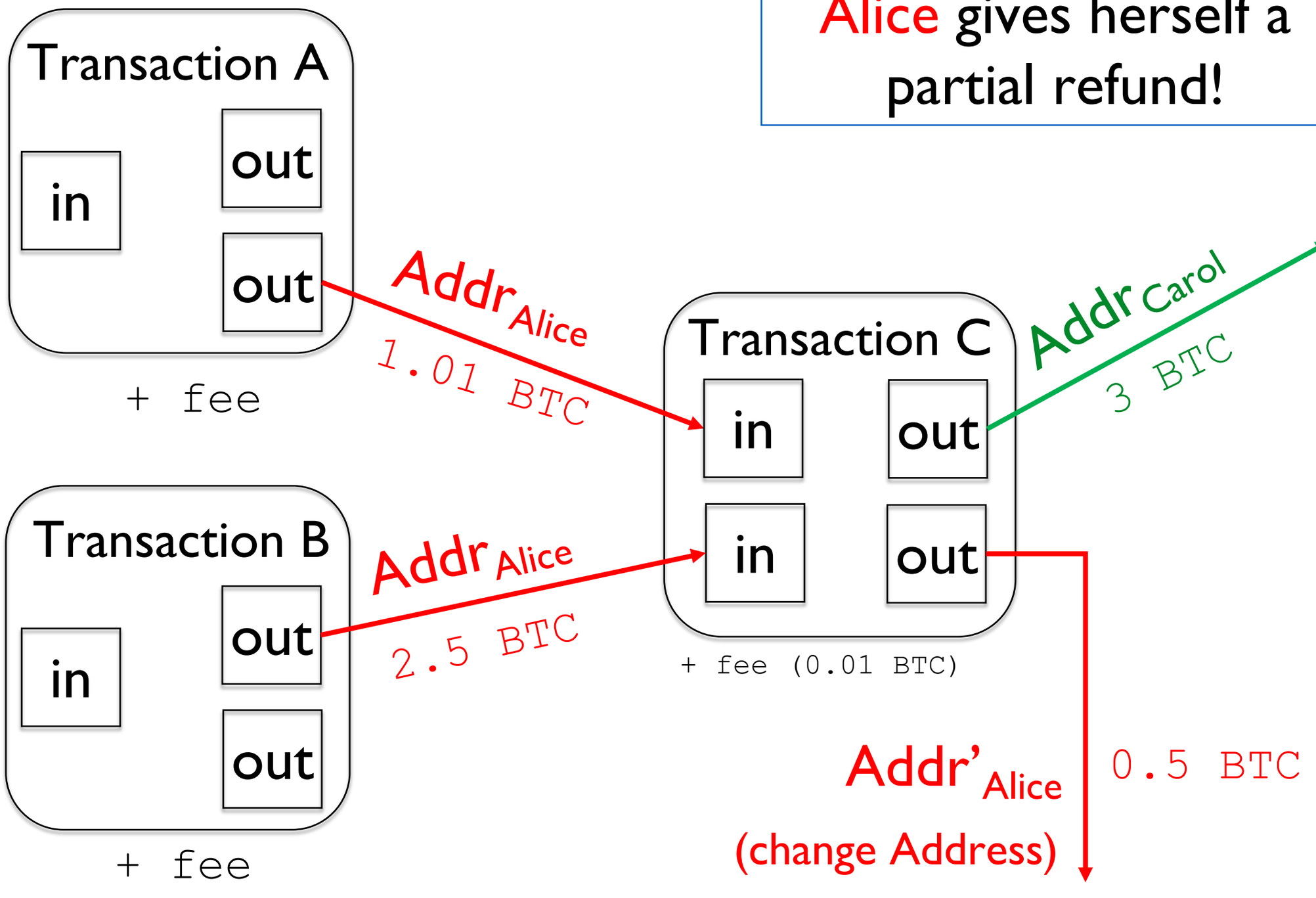
Bitcoin transaction structure

But how does Alice recover unspent value?



Bitcoin transaction structure

Alice gives herself a partial refund!



Scripts

2-input, 1-output transaction

metadata

input(s)

output(s)


```
{
  "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
  "ver":1,
  "vin_sz":2,
  "vout_sz":1,
  "lock_time":0,
  "size":404,
  "in":[
   {
    "prev_out":{
     "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
     "n":0
    },
     "scriptSig":"30440..."
   },
   {
    "prev_out":{
     "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
     "n":0
    },
    "scriptSig":"3f3a4ce81...."
   }
  ],
  "out":[
   {
    "value":"10.12287097",
    "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
   }
  ]
}
```


Fig. 3.3 in NBFMG

2-input, 1-output transaction

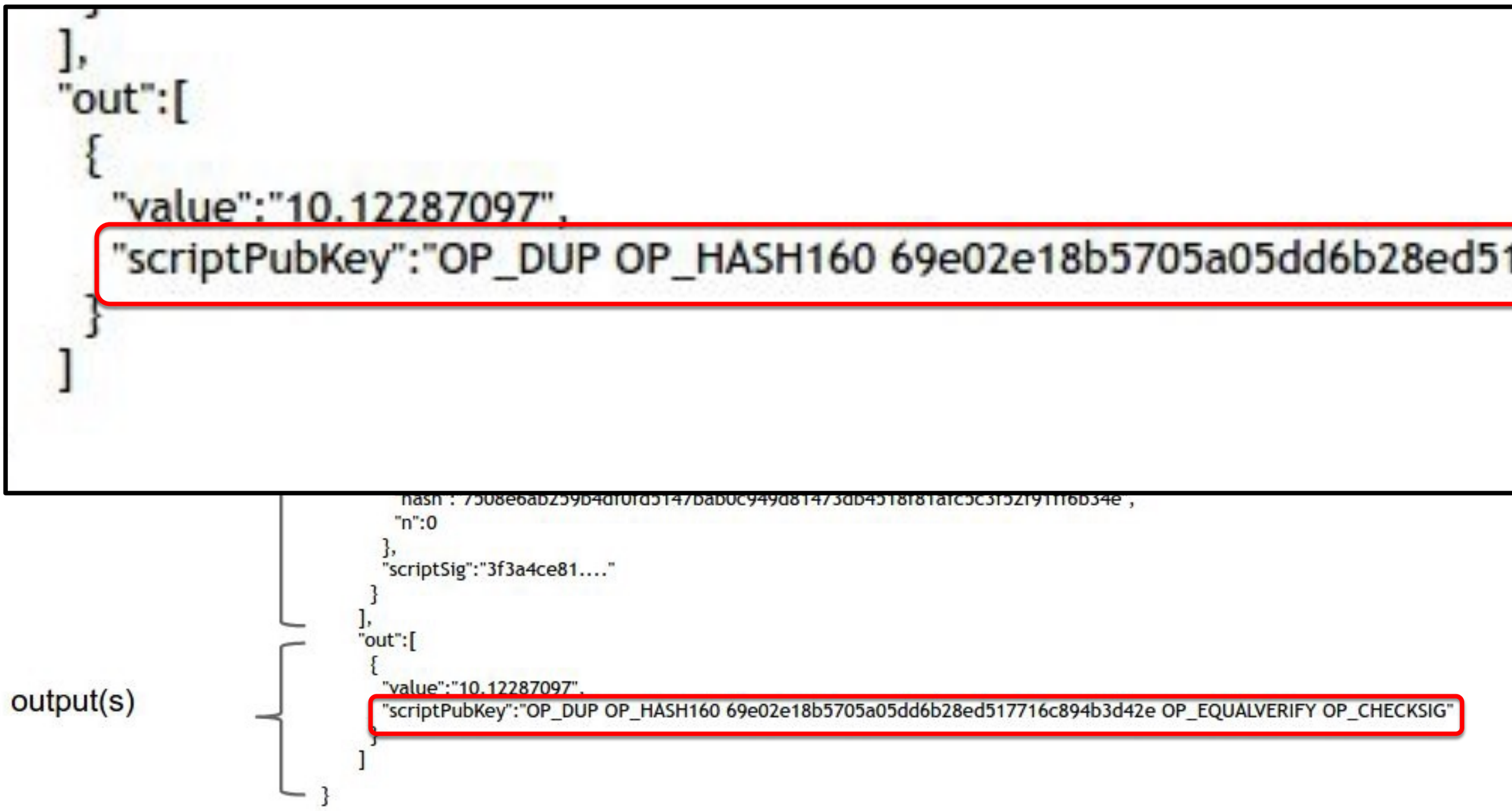
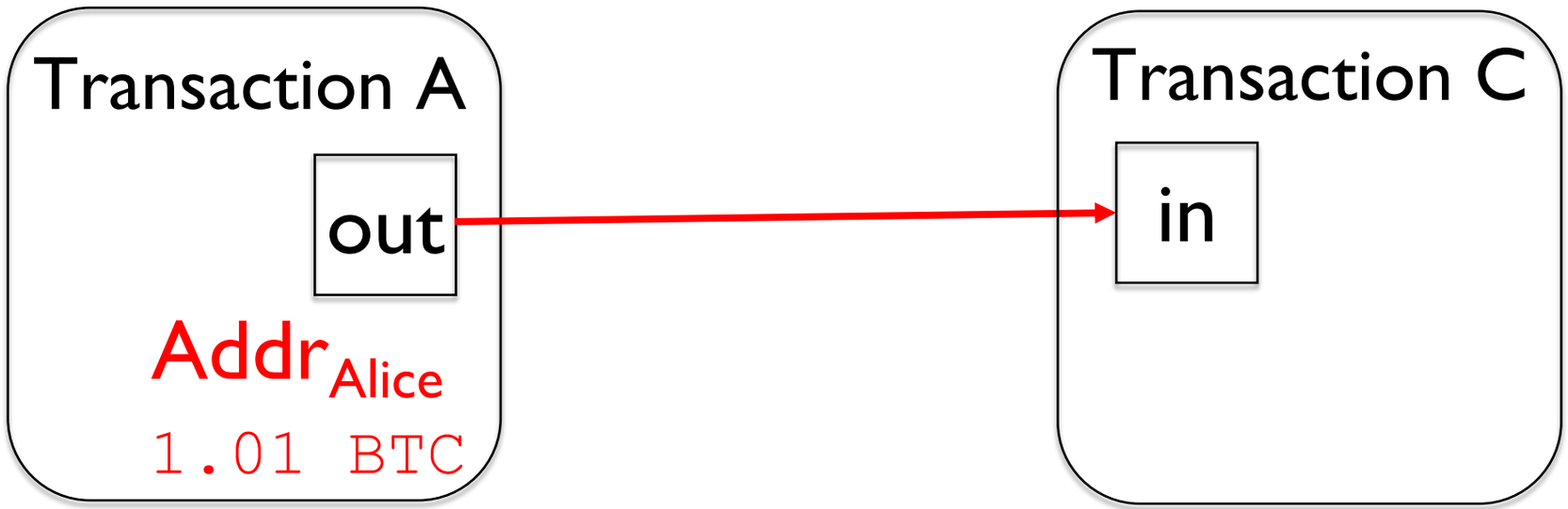


Fig. 3.3 in NBFMG

Intuition



- What needs to be shown in [in] to prove legitimate use of [out]?
- [in] must:
  - Include “unlocking code”…
  - with valid signature *sig* for Transaction C under SK_{Alice}!

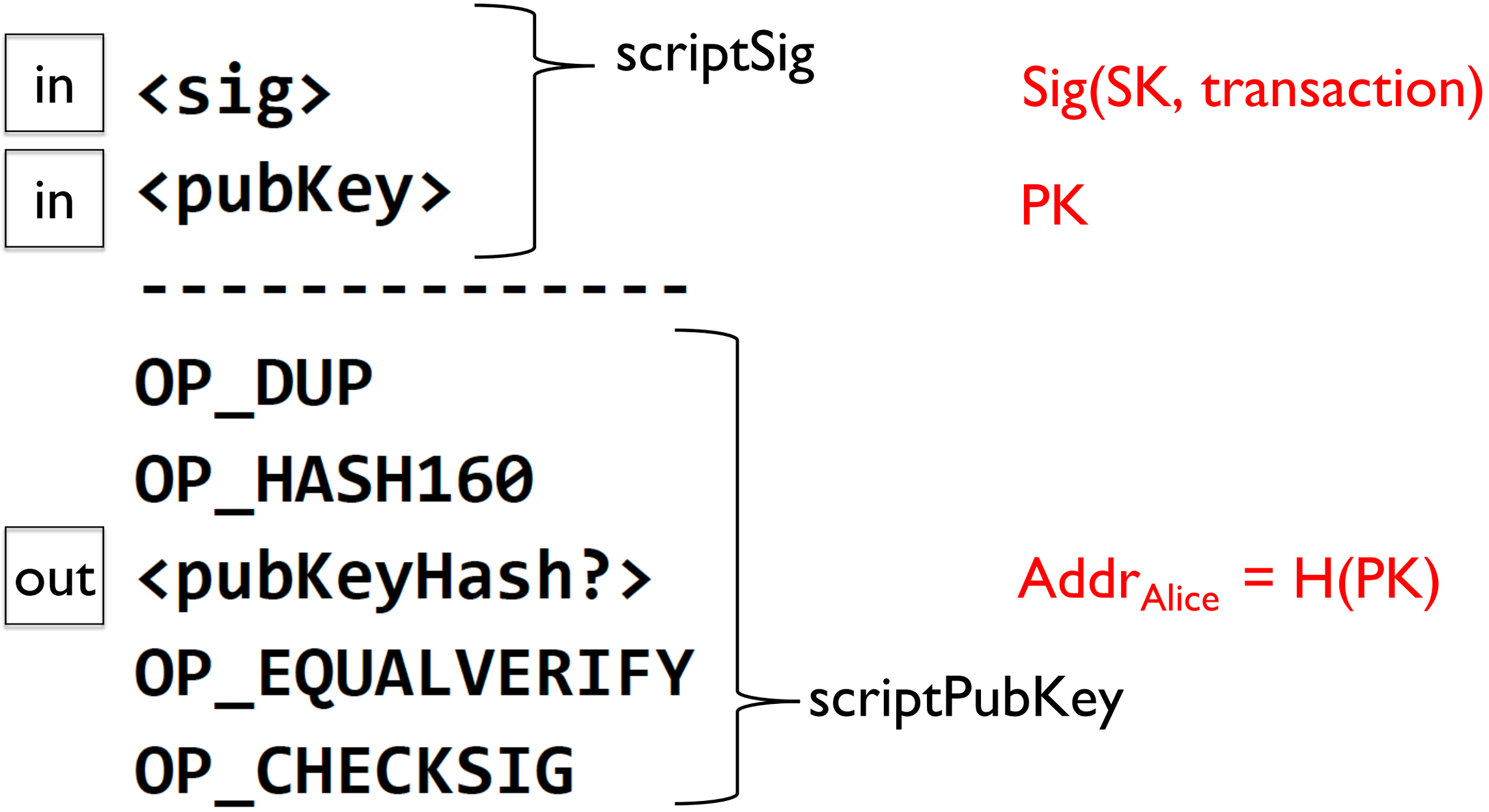
Bitcoin script

Pay to PubKey Hash (P2PKH)

in	`<sig>`	Sig(SK, transaction)
in	`<pubKey>`	PK
	`----------`	
	`OP_DUP`	
	`OP_HASH160`	
out	`<pubKeyHash?>`	$Addr_{Alice} = H(PK)$
	`OP_EQUALVERIFY`	
	`OP_CHECKSIG`	

Bitcoin script

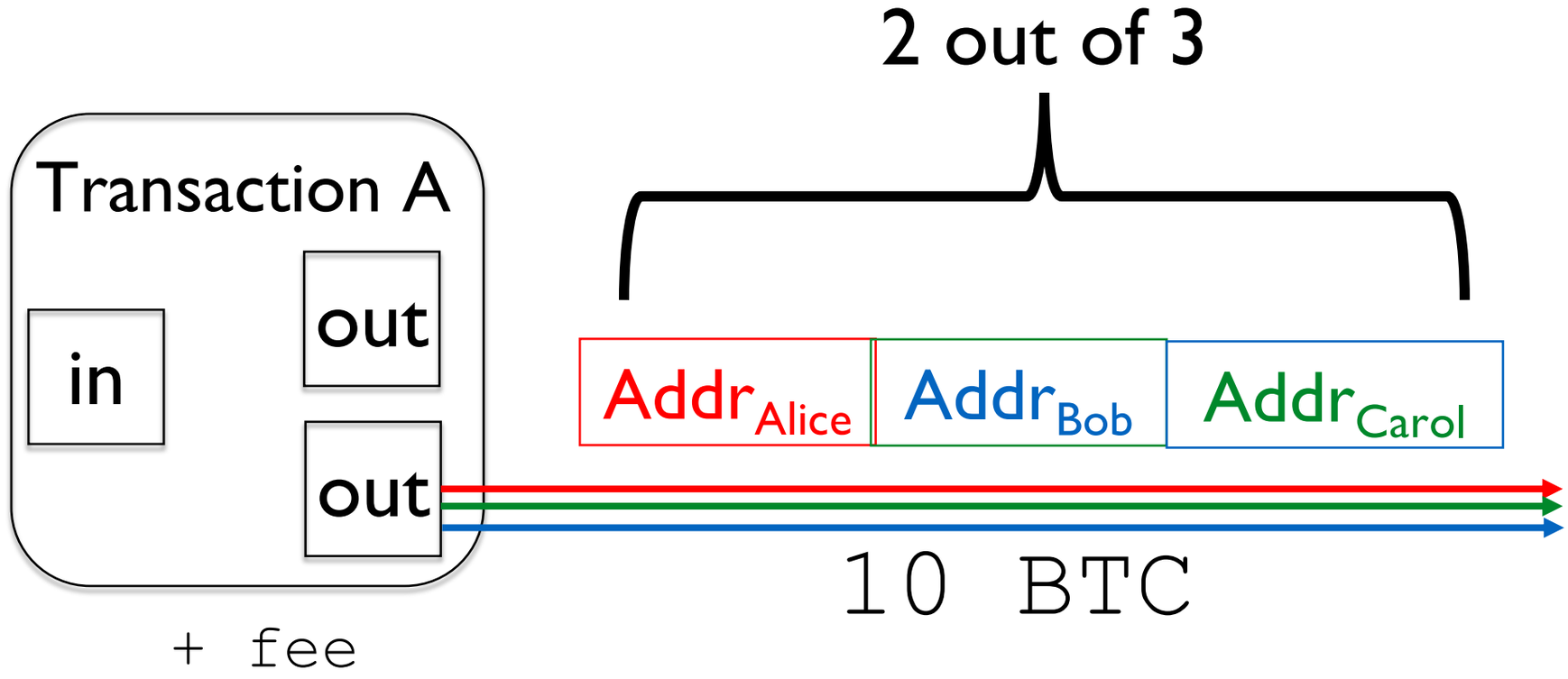
Pay to PubKey Hash (P2PKH)



More Scripts

Multsig transactions

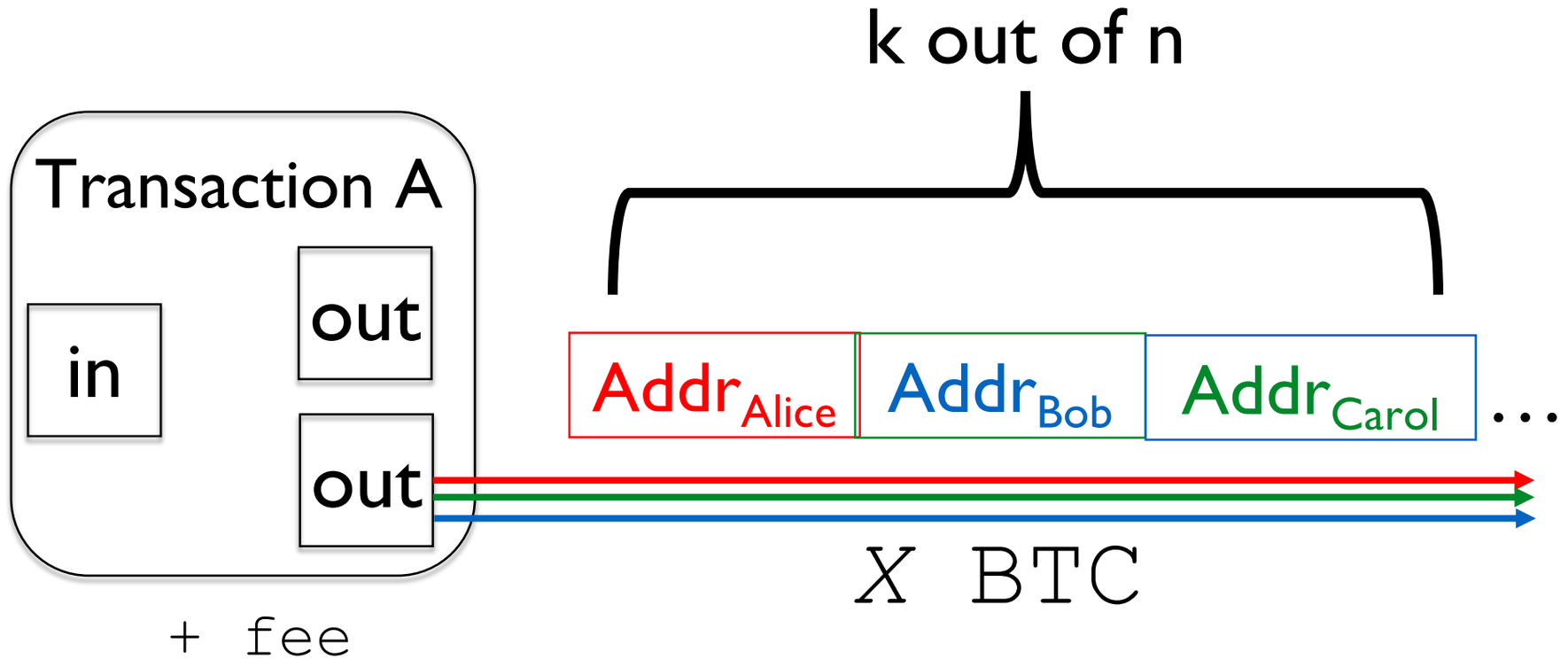
Example:



Lots of flexibility!

More generally

Example:



Multisig application: Joint control

- Alice, Bob, and Carol run a charitable organization: Bitcoin Songwriters of America (BSA).
  - BSA funds writers of Bitcoin songs.
- BSA holds 100 BTC.
- Why might they want to use a 2-out-of-3 multisig? Why not one sig or 3-out-of-3?
  - Ensure no one steals money
  - Ensure collective agreement on which music videos to fund
  - And… protect against loss of one key

https://www.youtube.com/watch?v=RIsZyg8OXII

Multisig application: Escrow

- Alice is selling Bob a Lamborghini for 10 BTC
- What if Bob sends the money but… Alice doesn’t deliver the Lambo?
- What if Alice delivers the Lambo but… Bob doesn’t pay?

Exercise

Suppose Carol is a trusted third party (and can verify delivery of Lambo).

- How do use a multisig so that:
  - If Alice and Bob agree, Carol isn't bothered.
  - If there's a dispute, Carol can make sure money goes to right person?

Exercise

- Bob pays 10 BTC into 2-out-of-3 multisig with Alice, Bob, and Carol
- If Lambo delivered, and Alice and Bob honest: Money paid to Alice
- If Lambo not delivered or Bob refuses to sign: Carol and honest player direct money

Bitcoin scripting feature: Timelock

- nLockTime
  - Part of original Bitcoin–in every transaction
  - Specifies earliest time / height transaction is valid
  - Applies to *whole transaction*

Bitcoin scripting feature: Timelock

- ClockLockTimeVerify (CLTV) opcode
  - Added opcode like nLockTime but *output-specific*
- CheckSequenceVerify (CSV) opcode
  - Specifies *relative time* Δ at which output is valid
  - I.e., output valid at time / height *now* + Δ

2. Payment channels

- Problem: On-chain Bitcoin transactions are *expensive* and *slow*
  - *Question: How slow?*
- Solution: Make Bitcoin payments (mostly) without using blockchain (???)
- Mechanism called *payment channel*
- Main implementation: *Lightning network*

TOSHI TIMES

Bonus question: Why pizza?

Bitcoin

Buy Domino's Pizza With BTC Using Bitcoin's Lightning Network

By Rasmus Pihl *4 days ago* *0* Feb. 2019

On 22 May 2010,[147] Laszlo Hanyecz made the first real-world transaction by buying two pizzas in Jacksonville, Florida, for 10,000 BTC; an amount that would be nearly $40 million if held today (as of February 2019).[148][149]

Follow

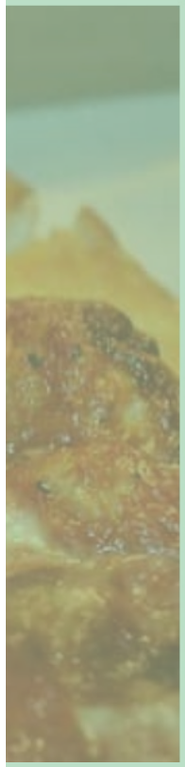
Laszlo Hanyecz
@HanyeczLaszlo

I am the person who bought the 10000btc pizza 8 years ago. I am poor now. Feel free to donate any amount of btc:

BTC: 1NosDYmVU4VHv5Yd9CuNsStjptttM1Y6HW

Joined May 2018

4 Following **1,745** Followers



Problem

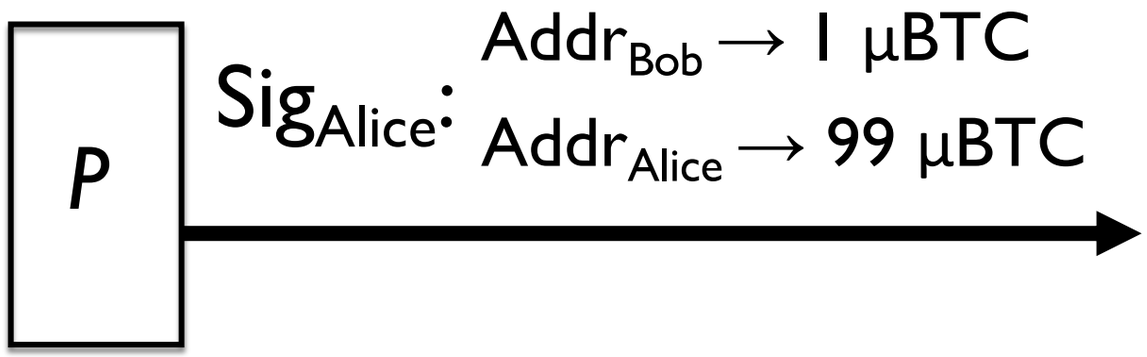
- Alice wants to make multiple small payments to Bob
- E.g., She's buying articles from Bob's news site
  - Each article costs 1 µBTC
- Alice prefunds channel
  - E.g., pays in 100 µBTC

Unidirectional payment channel

Alice Bob

Setup (payment into channel):

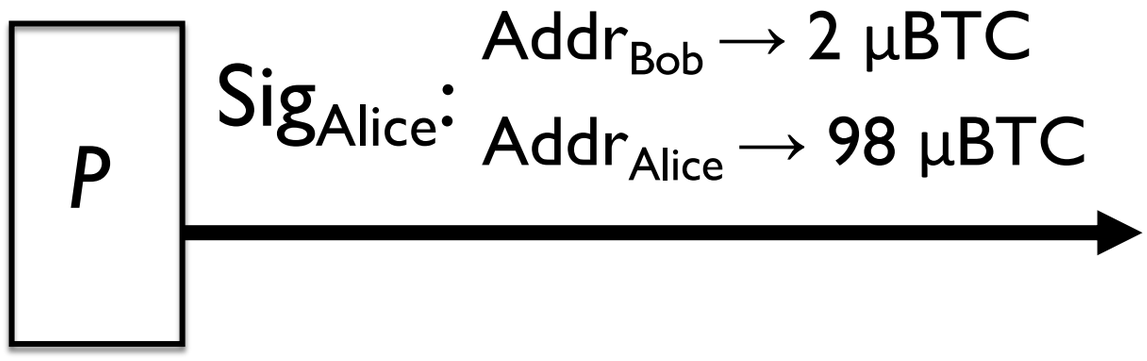
Payment 1 (unposted transaction):



Unidirectional payment channel

Alice Bob

Payment 2 (unposted transaction):



Payment i (unposted transaction): :

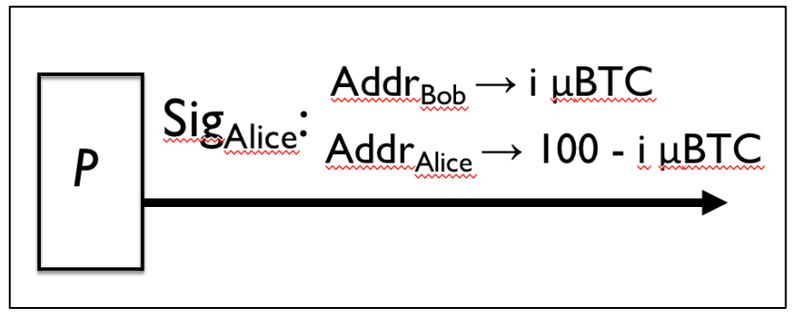
Unidirectional payment channel

Alice

Bob

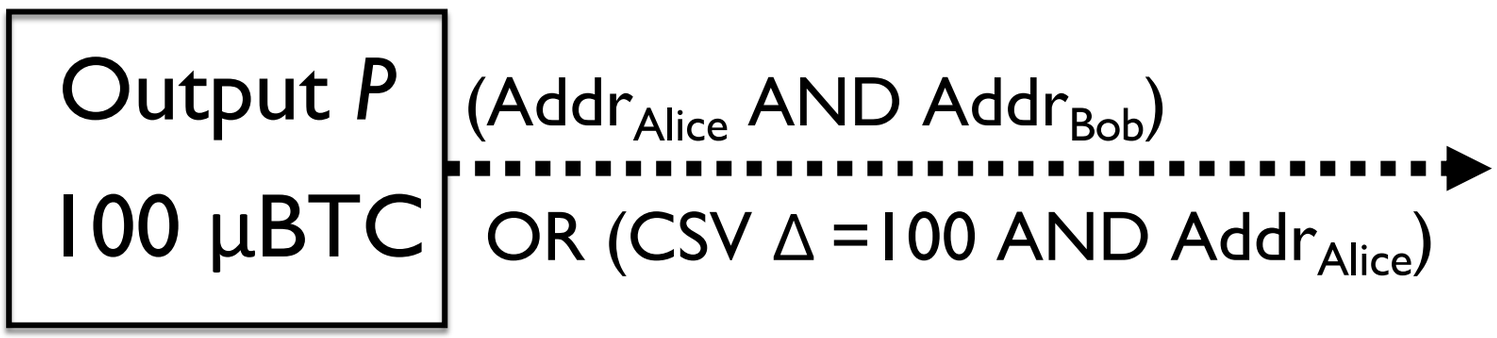
Channel closeout:

Posts Payment i



Unidirectional payment channel

- What happens if Bob never responds /posts?



- What downsides does payment channel have?
  - Hint: What if Alice pays 10 BTC into the channel with Δ = 1 year?