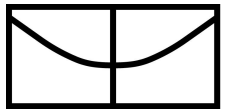


MIT Digital Currency Initiative and the University of Brasilia presents

Cryptocurrency Design and Engineering

Lecture 6: State Models
Taught by: Neha Narula
September 30, 2025
MAS.S62

State Models

- A cryptocurrency from a log
- Bitcoin - UTXOs
- Ethereum – Contracts and accounts

A cryptocurrency from a log

- Everything that came before was just about agreeing on a log…
- How do we get a cryptocurrency from a log?
  - Operations, initial state, state transition function, validation

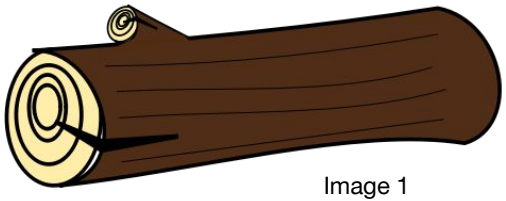


Image 1

What do we need in a transaction?

Traditional digital payments

Alice: $10
Bob: $0

"I, Alice, would like to send Bob $5"

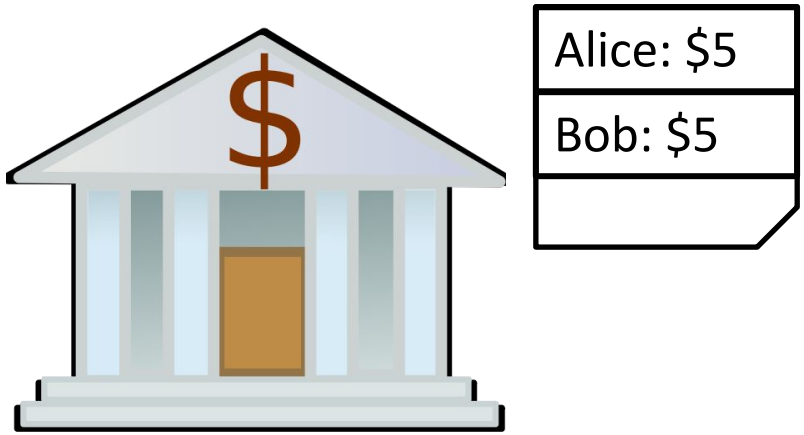
Image 2

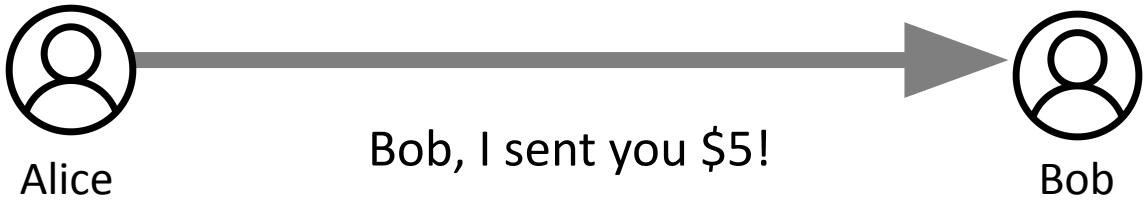
Alice

Bob

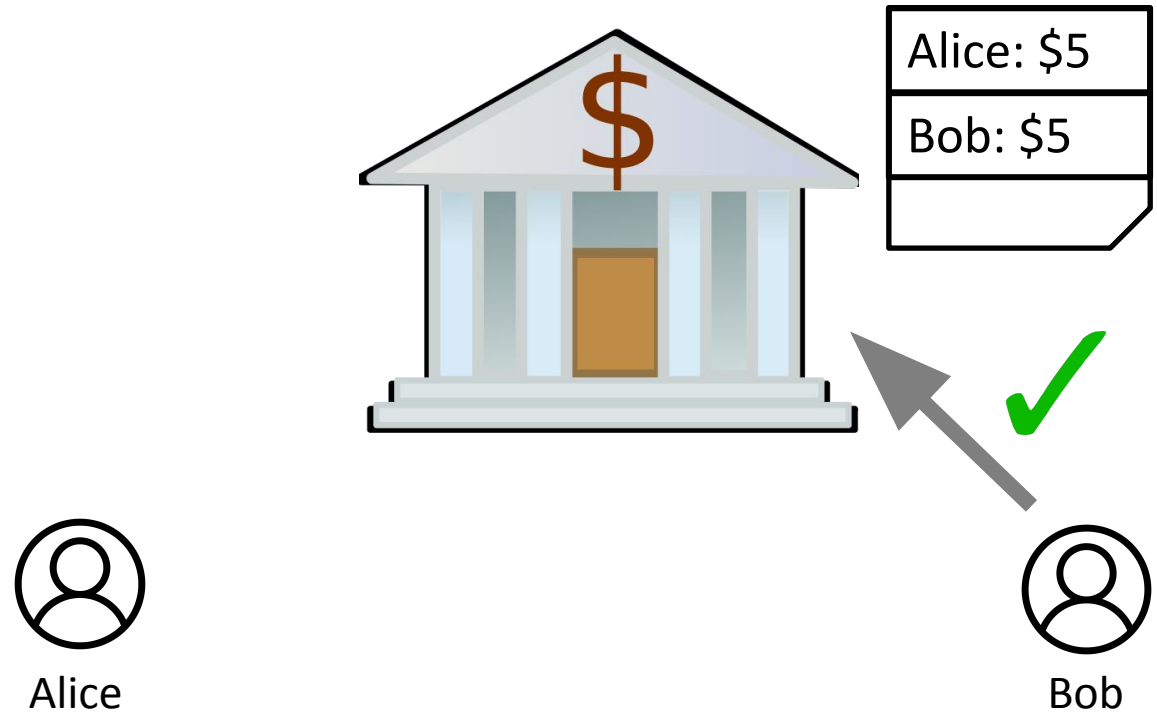
Image 3

Traditional digital payments





Traditional digital payments



Traditional digital payments

Alice: $5
Bob: $5

What do we need in a transaction?

- Amount
- User, authorization
- Who you're paying

```
Who: Alice
Amount: $5
Payee: Bob
Auth: Sig_Alice(??)
```

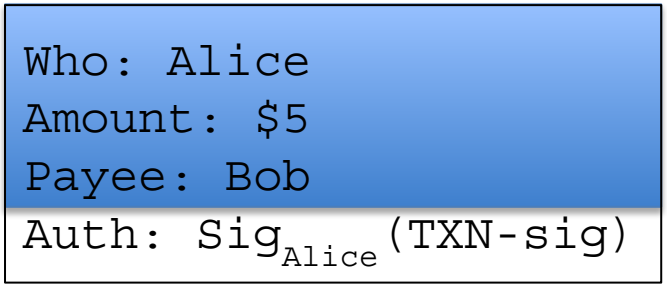
What do we need in a transaction?

- Amount
- User, authorization
- Who you're paying

```
Who: Alice
Amount: $5
Payee: Bob
Auth: Sig_Alice(TXN)
```

What do we need in a transaction?

- Amount
- User, authorization
- Who you're paying



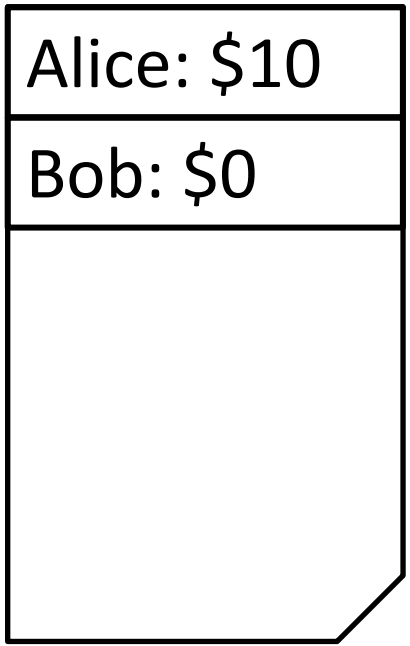
What do we need in a transaction?

- Amount
- User, authorization
- Who you're paying

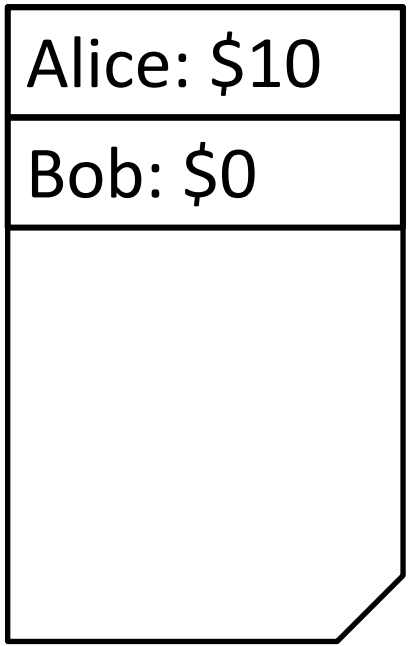
```
Who: Alice
Amount: $5
Payee: Bob
```

Auth: $Sig_{Alice}(H(TXN\text{-}sig))$

Account based model

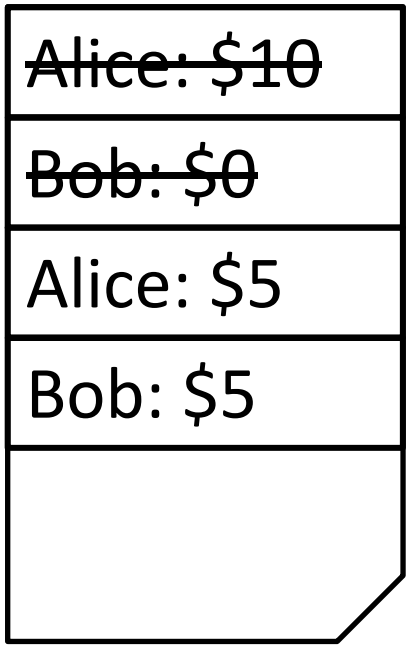


Account based model




```
Who: Alice
Amount: $5
Payee: Bob
Auth: Sig_Alice(H(TXN-sig))
```

Account based model

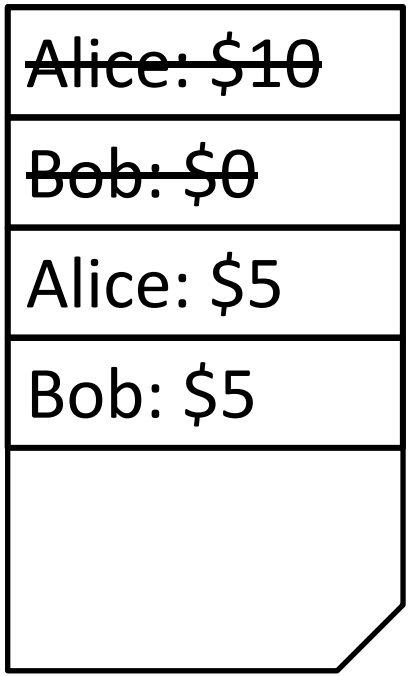


Who: Alice
Amount: $5
Payee: Bob
Auth: $Sig_{Alice}(H(TXN\text{-}sig))$

Account based model

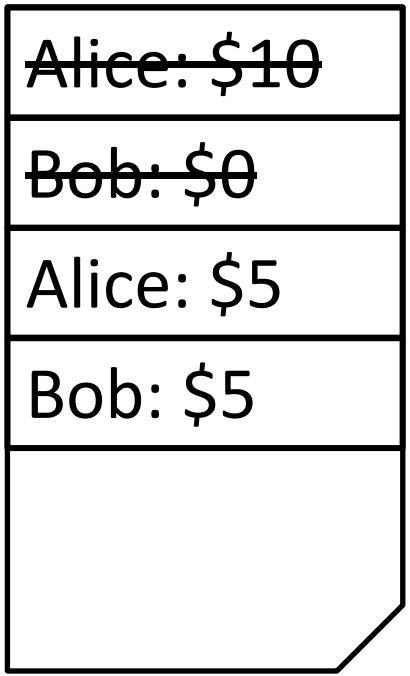
- Store list of accounts and balances (S)
- A transaction is valid if there is enough balance in the account
- Sender debited, receiver credited

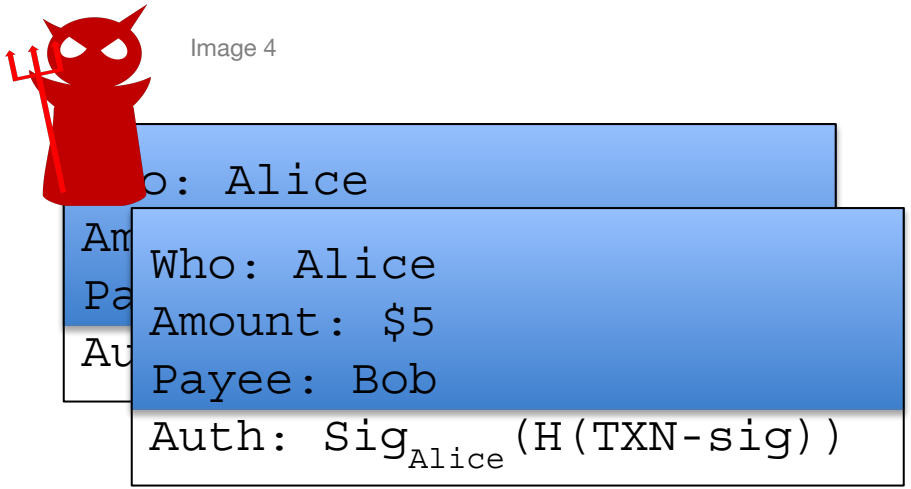
Replay attacks



Who: Alice
Amount: $5
Payee: Bob
Auth: $Sig_{Alice}(H(TXN\text{-}sig))$

Replay attacks





Replay attacks

~~Alice: $10~~

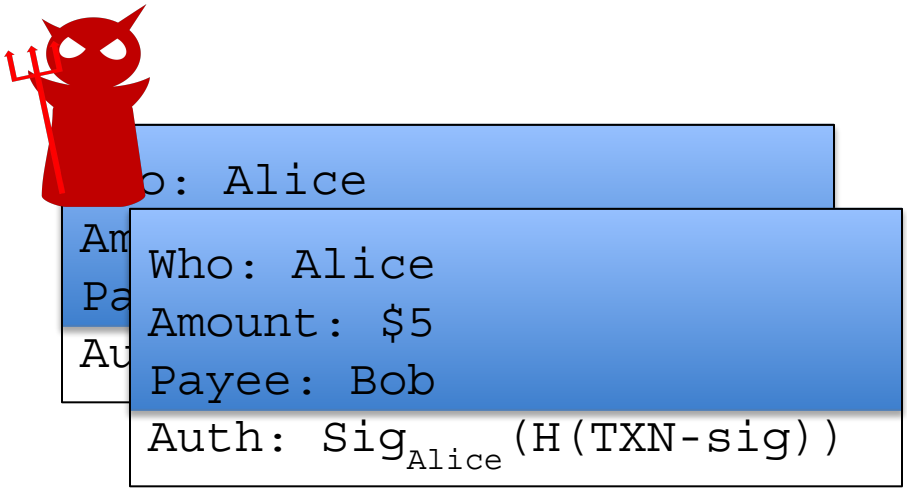
~~Bob: $0~~

~~Alice: $5~~

~~Bob: $5~~

Alice: $0

Bob: $10



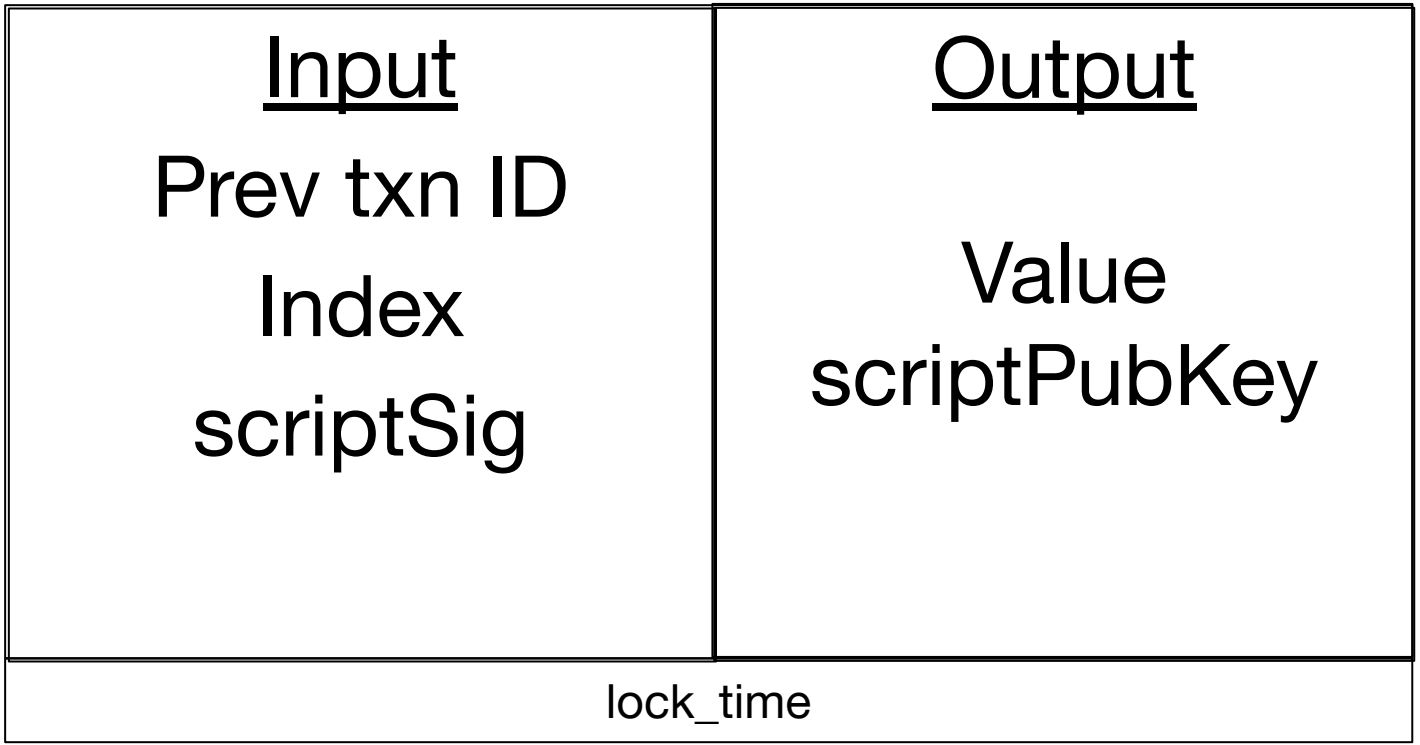
State Models

- A cryptocurrency from a log
- Bitcoin - UTXOs
- Ethereum – Contracts and accounts

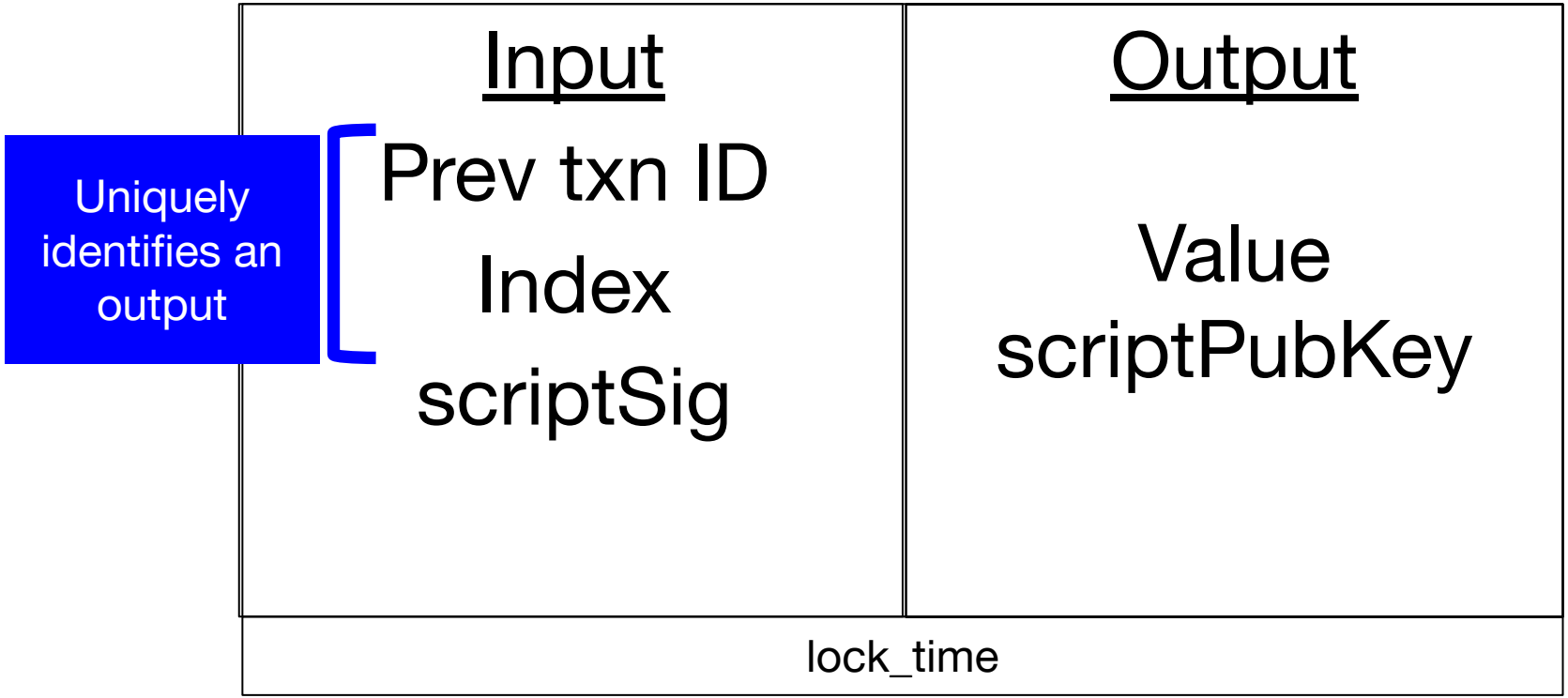
Unspent Transaction Outputs

- All coins are not the same
- Refer to specific coins when spending
- Coins are consumed; create new ones
- A coin can only be spent once

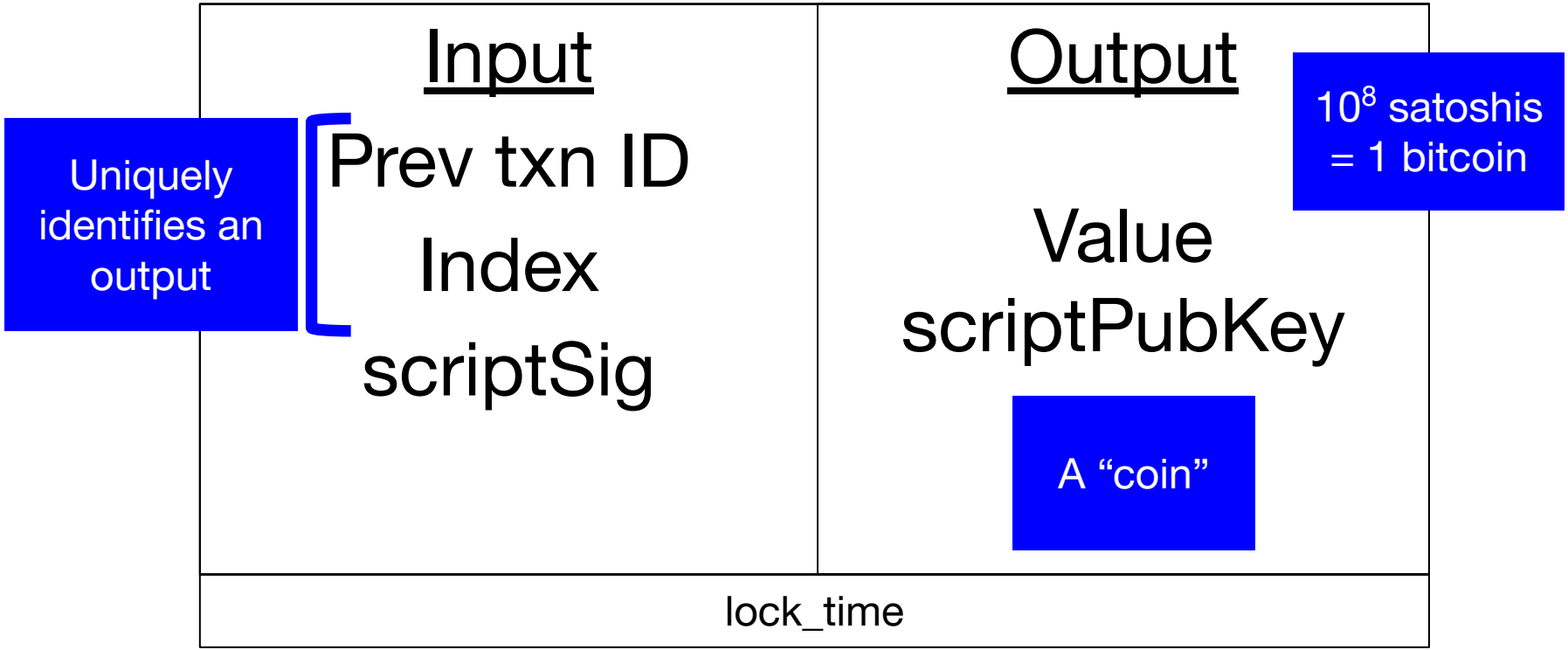
Transaction format



Transaction format



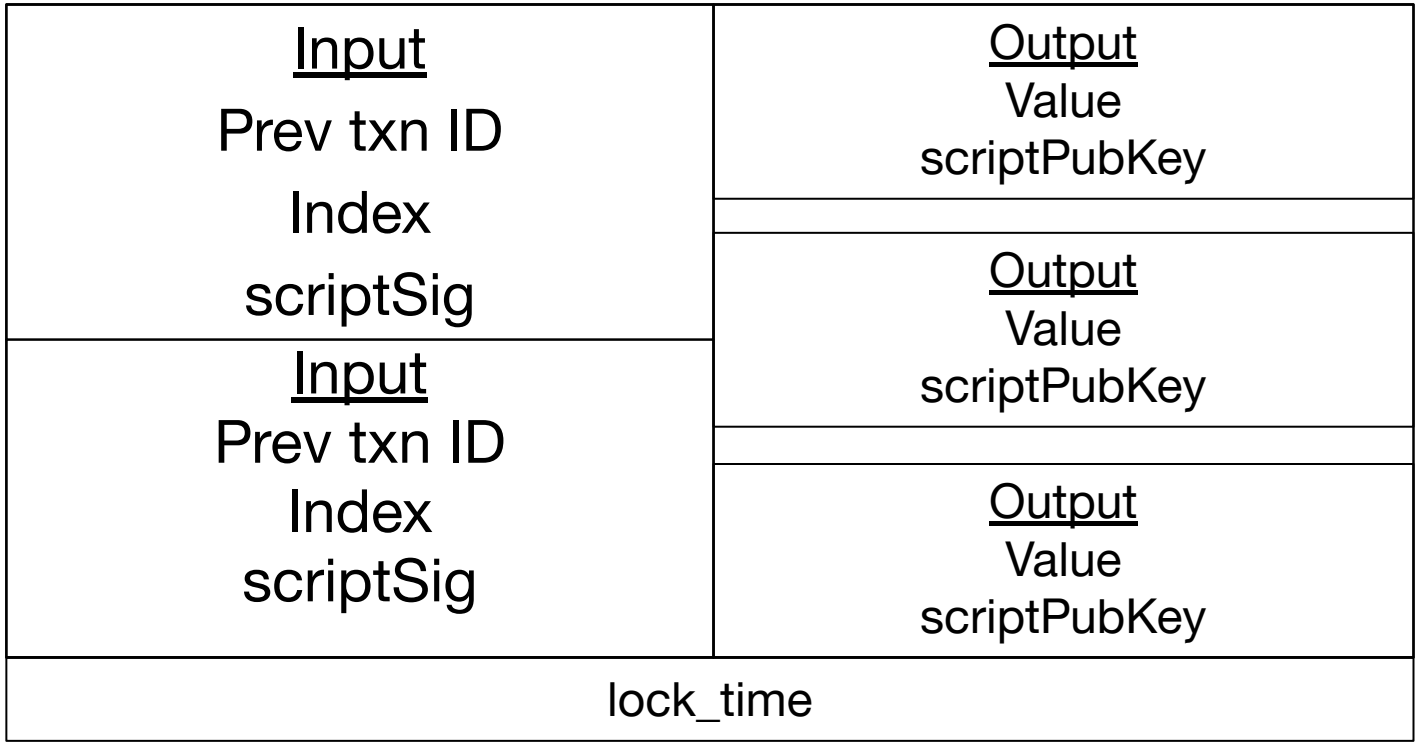
Transaction format



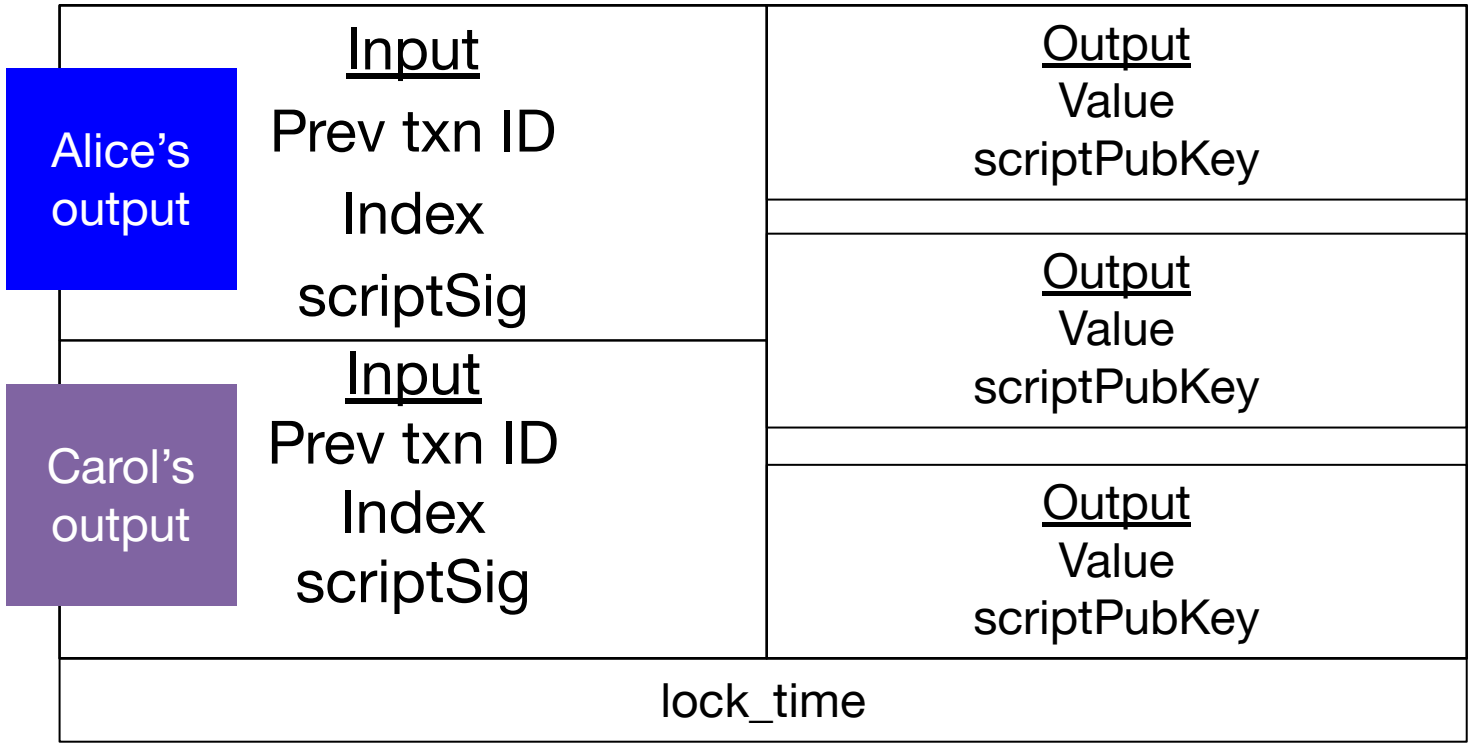
ScriptSigs and scriptPubkeys

- ScriptPubkeys are predicates
- ScriptSigs help satisfy the predicates
- When can you spend a coin? You know how to produce a satisfying scriptSig

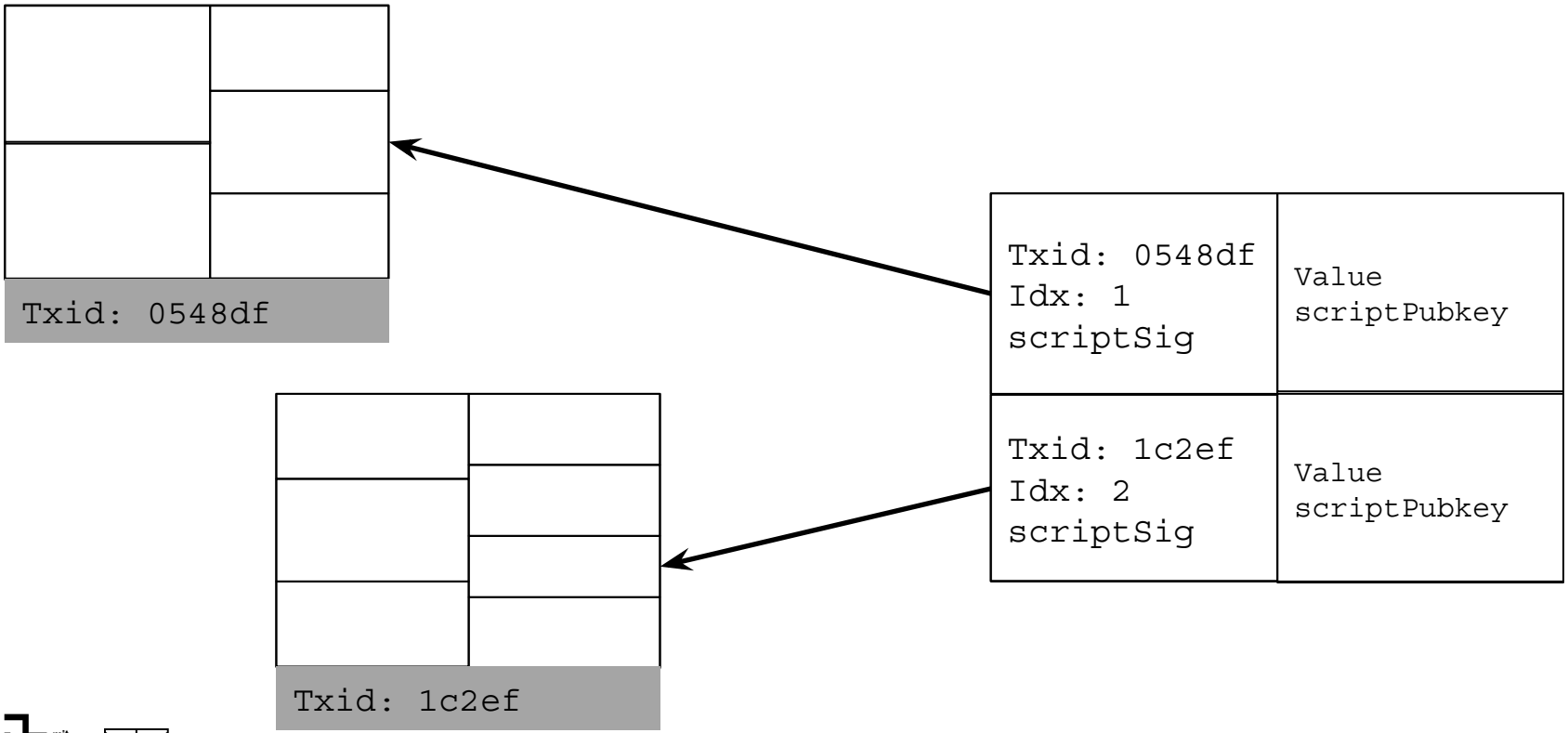
Multiple inputs and outputs



Inputs and outputs are independent



Transactions



```
"txid" : "c80b343d2ce2b5d829c2de9854c7c8d423c0e33bda264c40138d834aab4c0638",
"hash" : "c80b343d2ce2b5d829c2de9854c7c8d423c0e33bda264c40138d834aab4c0638",
"size" : 85,
"vsize" : 85,
"version" : 1,
"locktime" : 0,
"vin" : [
    {
        "txid" : "3f4fa19803dec4d6a84fae3821da7ac7577080ef75451294e71f9b20e0ab1e7b",
        "vout" : 0,
        "scriptSig" : {
            "asm" : "",
            "hex" : ""
        },
        "sequence" : 4294967295
    }
],
"vout" : [
    {
        "value" : 49.99990000,
        "n" : 0,
        "scriptPubKey" : {
            "asm" : "OP_DUP OP_HASH160 cbc20a7664f2f69e5355aa427045bc15e7c6c772 OP_EQUALVERIFY OP_CHECKSIG",
            "hex" : "76a914cbc20a7664f2f69e5355aa427045bc15e7c6c77288ac",
            "reqSigs" : 1,
            "type" : "pubkeyhash",
            "addresses" : [ "mz6KvC4aoUeo6wSxtiVQTo7FDwPnkp6URG" ]
        }
    }
]
```

Transaction validity rules

- Sum(inputs) >= Sum(outputs)
  - One exception: coinbase transactions
  - Why not equal? Fees!
- For every input, Eval(scriptSig+scriptPubKey) == true
- Output has not already been spent
- lock_time

Reject ANY BLOCKCHAIN that contains transactions that don't follow these rules!

Pay to Pubkey Hash (P2PKH)

- Idea: Send money to a pubkey
- Pubkeys are big, a hash of a pubkey is only 32 bytes (+1 byte for prefix)
- scriptPubkey: instructions on how to verify a signature of a pubkey that is hashed
- scriptSig: signature, pubkey

Will see details in programmability lecture

The state: the UTXO set

- Every transaction destroys its inputs and creates its outputs
- Every Bitcoin node applies all transactions and computes the set of Unspent Transaction Outputs (UTXO set) from the blockchain
- Represents valid set of unspent coins
- Currently:
  - ~170M UTXOs
  - ~11GB

Benefits of UTXOs

- Storage
  - State is O(unspent coins), not O(accounts)
- Less linkage between transactions
  - Can generate new pubkeys
- Scalability
  - Except for existence, transactions can be validated independently

Downsides of UTXOs

- Complex!
- No global state; can’t do more complex programming

State Models

- A cryptocurrency from a log
- Bitcoin - UTXOs
- Ethereum – Contracts and accounts

Ethereum’s state model

- Much richer than Bitcoin: transactions can execute a whole program
- Global shared state: set of accounts
  - Externally owned accounts (EOA)
  - Contract accounts

Ethereum accounts

- Every account has the following fields:
  - Address
  - Code
  - Storage root
  - Balance
  - Nonce

10^{18} Wei = 1 ETH

- Account storage: storage array S[] in a Merkle Patricia Tree

Ethereum transaction format

To: 32 byte address

If To == 0: Create new contract with [code]

Else: [data]

From: address,

Value

Tx fees

Nonce

Function call and arguments

Prevents replay attacks

Benefits of Ethereum's model

- "Turing complete" programmability
  - Composability
  - Much easier to write complex applications
- Fees: you pay for what you use
- Intuitive account balance data model

Downsides of Ethereum's model

- Valid transactions always execute: You pay fees even if it doesn't do what you want
- Complexity breeds the ability to shoot yourself in the foot (later: DAO hack and re-entry bugs)
- Synchronizing on global state affects performance
- Wasteful? Every node executes every line of every program

YCHARTS DATA TOOLS SUPPORT PLANS Enter text or press "/"

Ethereum Chain Full Sync Data Size (I:ECFSDS)

1405.84 GB for Sep 21 2025

Overview | Interactive Chart

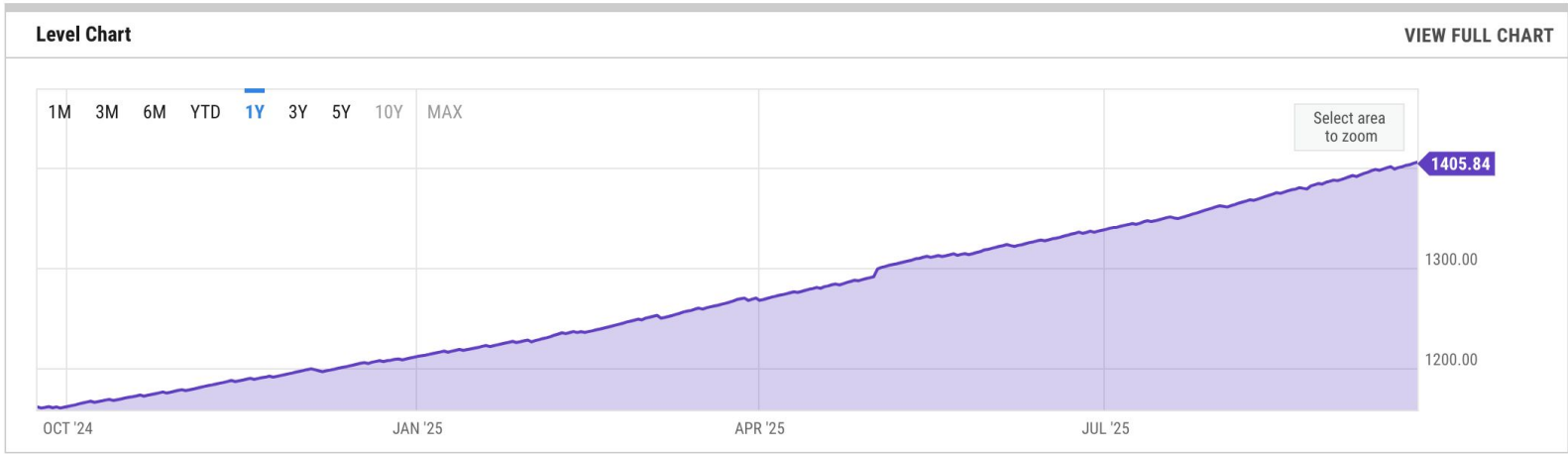


Image 5

Image Copyrights

Image 1: https://freesvg.org/log
Image 2: https://freesvg.org/bank-icon-vector -- public domain
Image 3: https://freesvg.org/user-icon – public domain
Image 4: https://clipart-library.com/clipart/demons-cliparts-5.htm demonss #3415674 (License. Personal Use)
Image 5: https://ycharts.com/indicators/ethereum_chain_full_sync_data_size