Laboratorio di web scraping

AA. 2024-2025

docente: Laura Ricci

laura.ricci@unipi.it

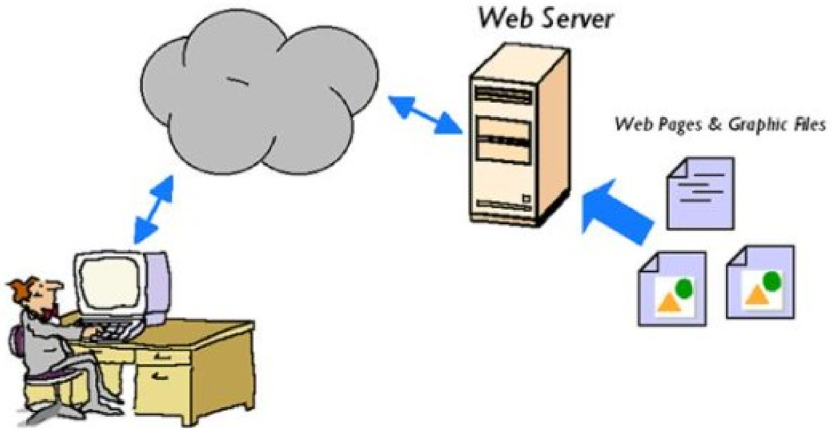
Lezione 18

Analysis of Bitcoin transactions

9/4/2025

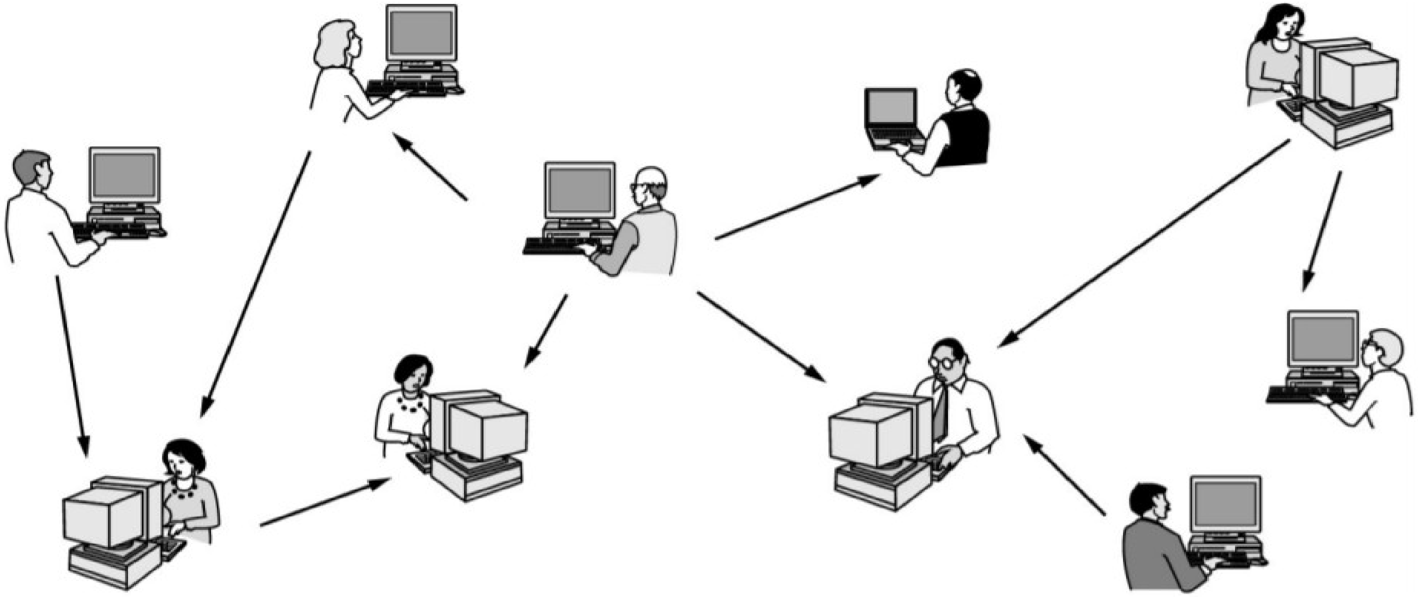
THE CLIENT SERVER MODEL FOR THE WEB

- the web client is your browser
  - makes an HTTP request to a specific web server
- the web server receives the request
  - sends back the requested document to the client
  - an HTML page, possibly with CCS and JavaScript
- the web client interprets the information returned by the server and displays it appropriately

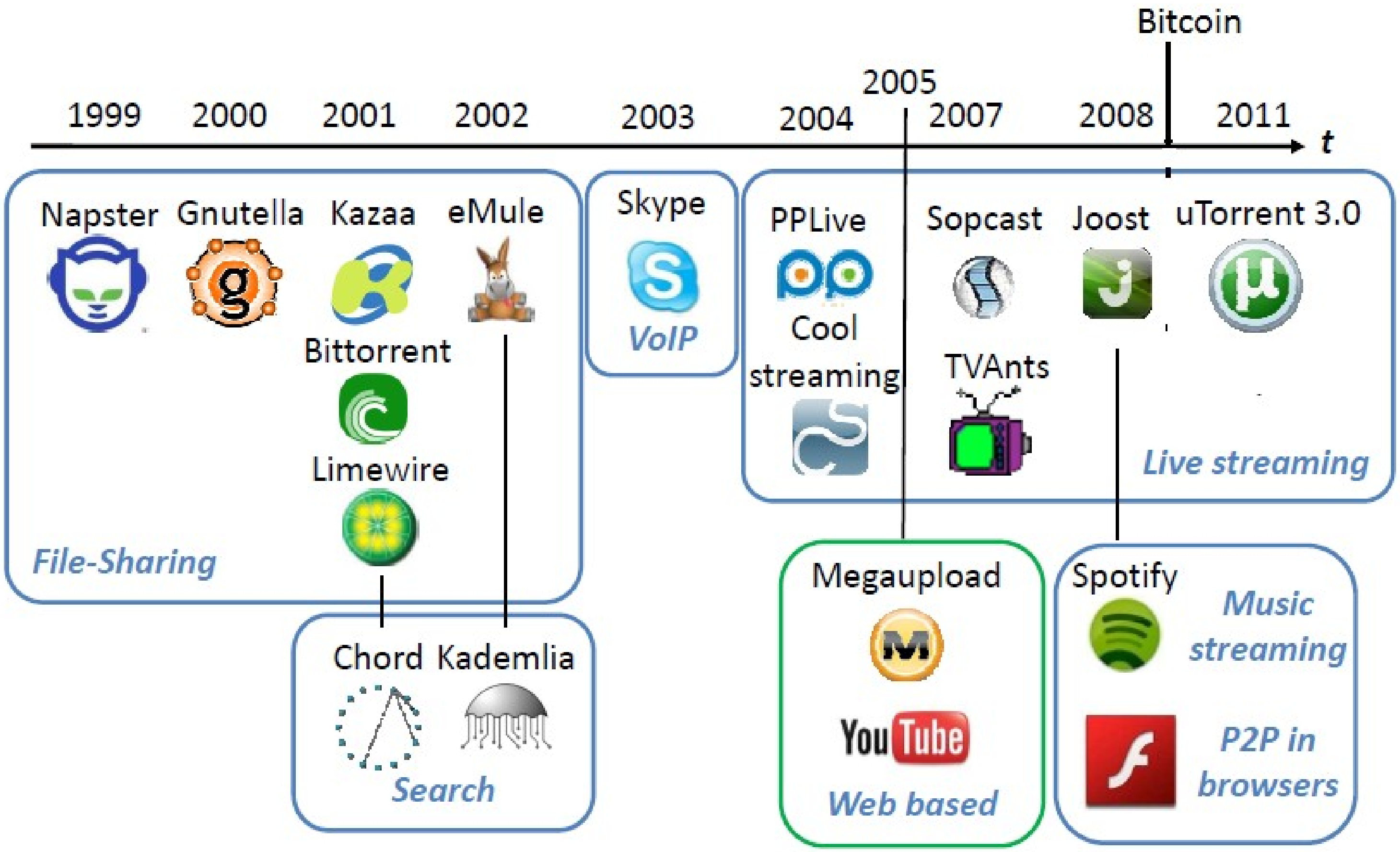


A CHANGE OF PARADIGM: PEER TO PEER

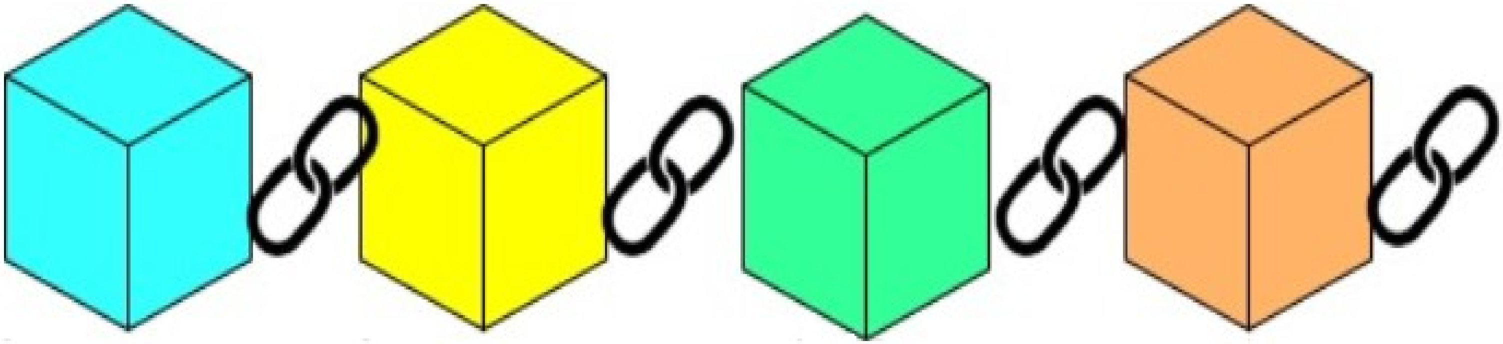
- by the turn of the twenty-first century, new models for delivering online services emerged.
- instead of relying on a centralized server, parties began experimenting with peer-to- peer (P2P) networks
    - a networks consisting of nodes (peers) working together based on equals rights/functionalities



PEER TO PEER: FROM FILE SHARING TO BITCOIN

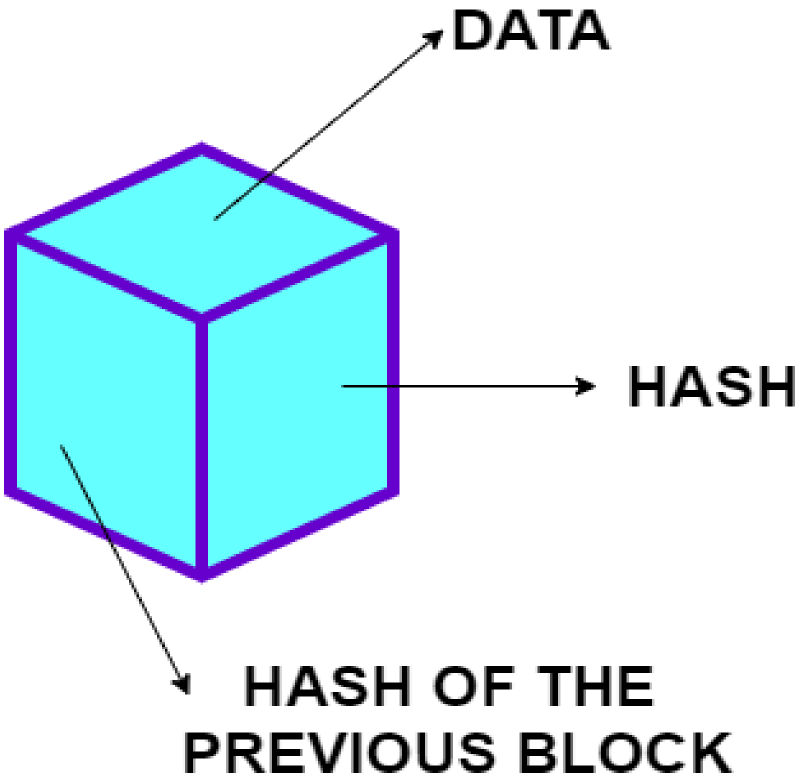


BLOCKCHAIN "AT A GLANCE"



- the 'killer P2P application'
- a distributed ledger which is replicated between the nodes of a peer-to-peer network
  - like a "notary"
  - characterized by the tamper freeness property

LOOKING INSIDE A BLOCK

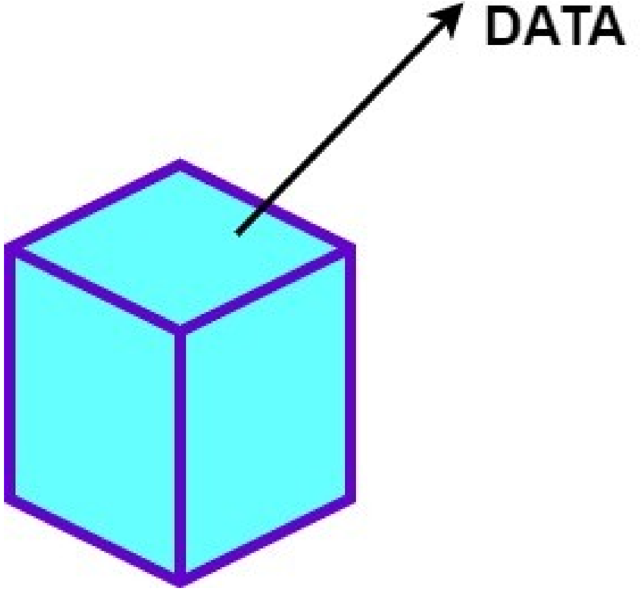


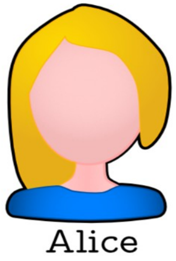
DATA

HASH

HASH OF THE PREVIOUS BLOCK

WHICH KIND OF DATA CAN WE FIND IN A BLOCK?



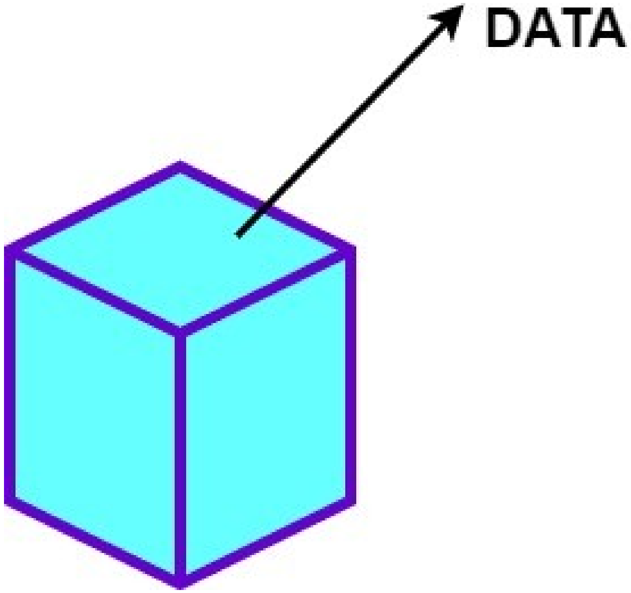


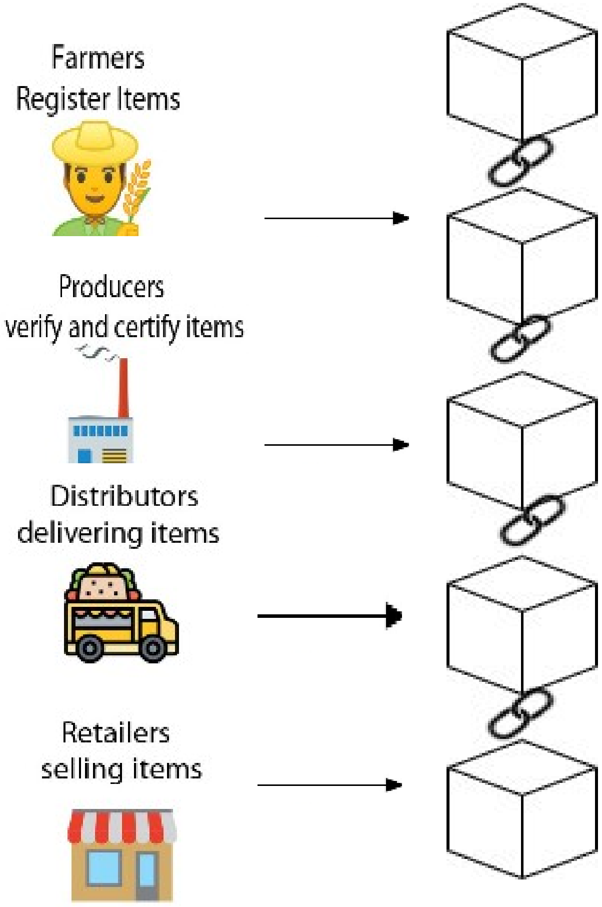
FROM

TO

AMOUNT

WHICH KIND OF DATA CAN WE FIND IN A BLOCK?





Farmers
Register Items

Producers
verify and certify items

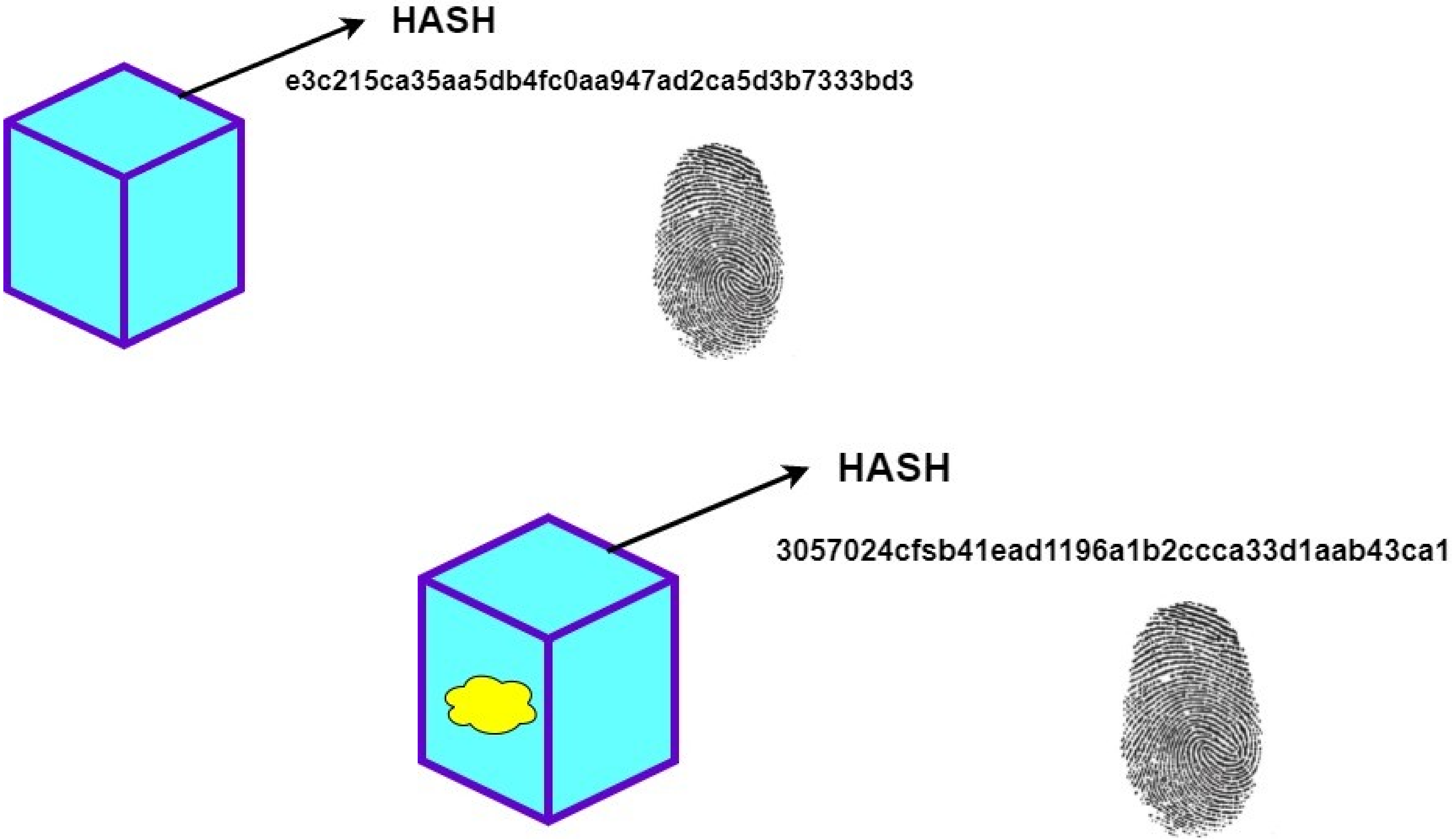
Distributors
delivering items

Retailers
selling items

WHICH KIND OF DATA CAN WE FIND IN A BLOCK?

- in general, transactions
  - cryptocurrencies transactions (Bitcoin, Ethereum,...) or finantial assets
- but not only!
  - sensor measurements IoT, ECG.,...
  - supply chain (e.g.: diamonds,....)
  - asset certification (NFT)
  - intellectual property (audio/video content)
  - health-care contents
- all data are public
  - ideal scenario for this course: lots of free data to analyse!

WHICH KIND OF DATA CAN WE FIND IN A BLOCK?



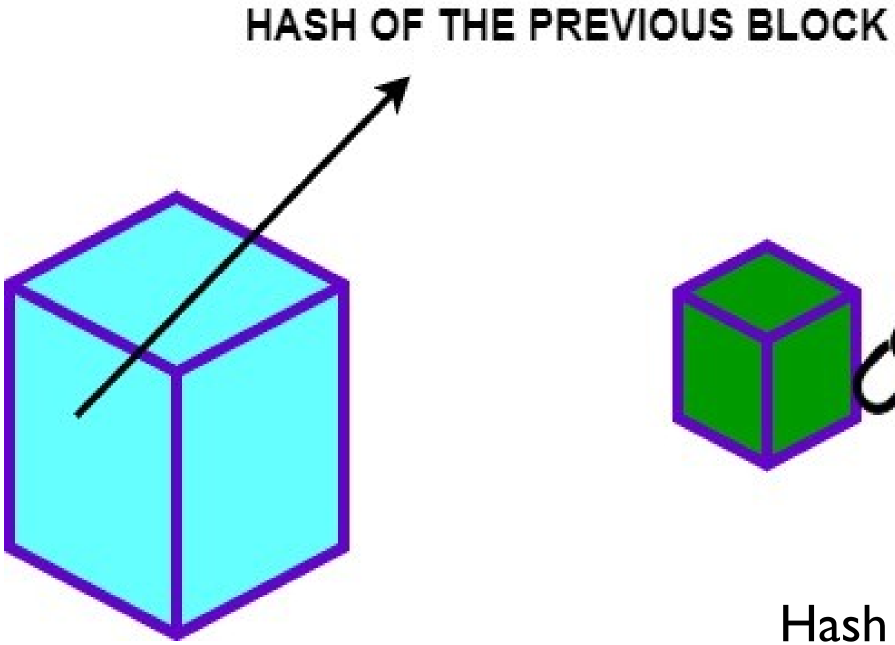
HASH

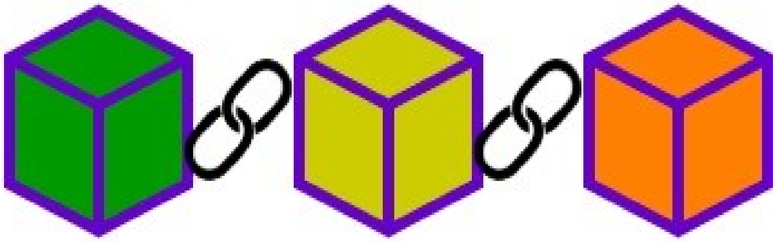
e3c215ca35aa5db4fc0aa947ad2ca5d3b7333bd3

HASH

3057024cfsb41ead1196a1b2ccca33d1aab43ca1

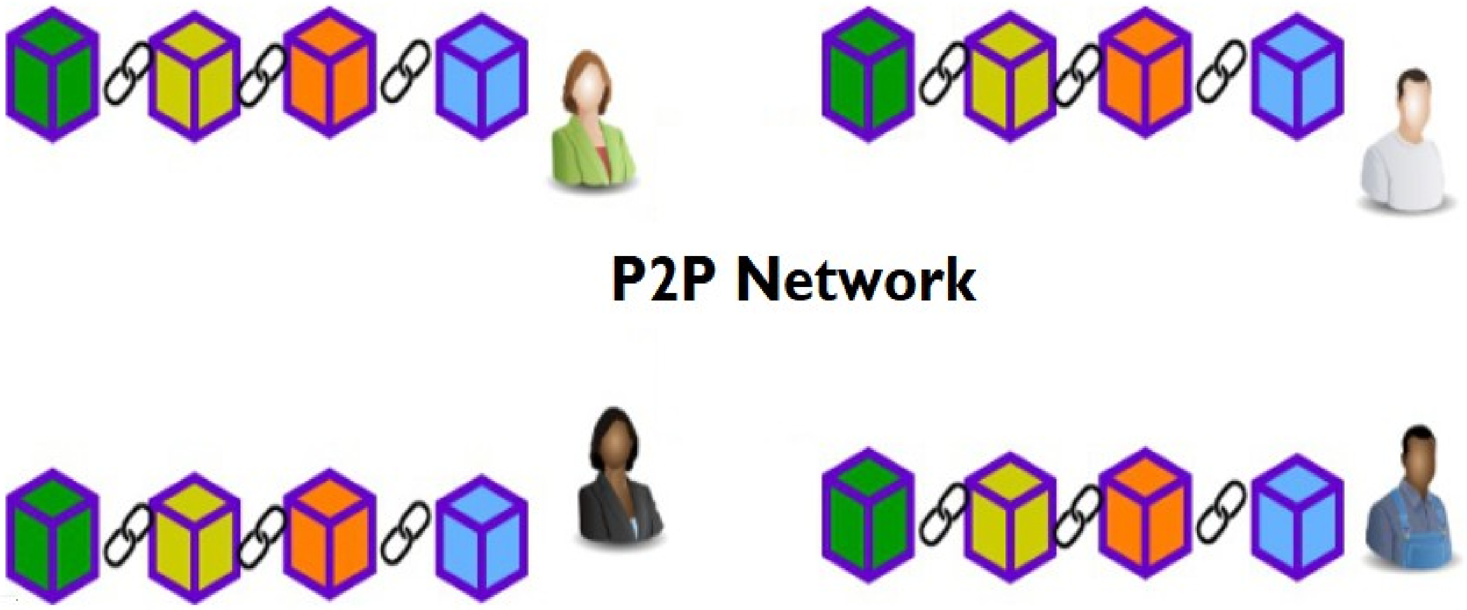
WHICH KIND OF DATA CAN WE FIND IN A BLOCK?





Hash pointers create the chain

THE DISTIBUTE LEDGER

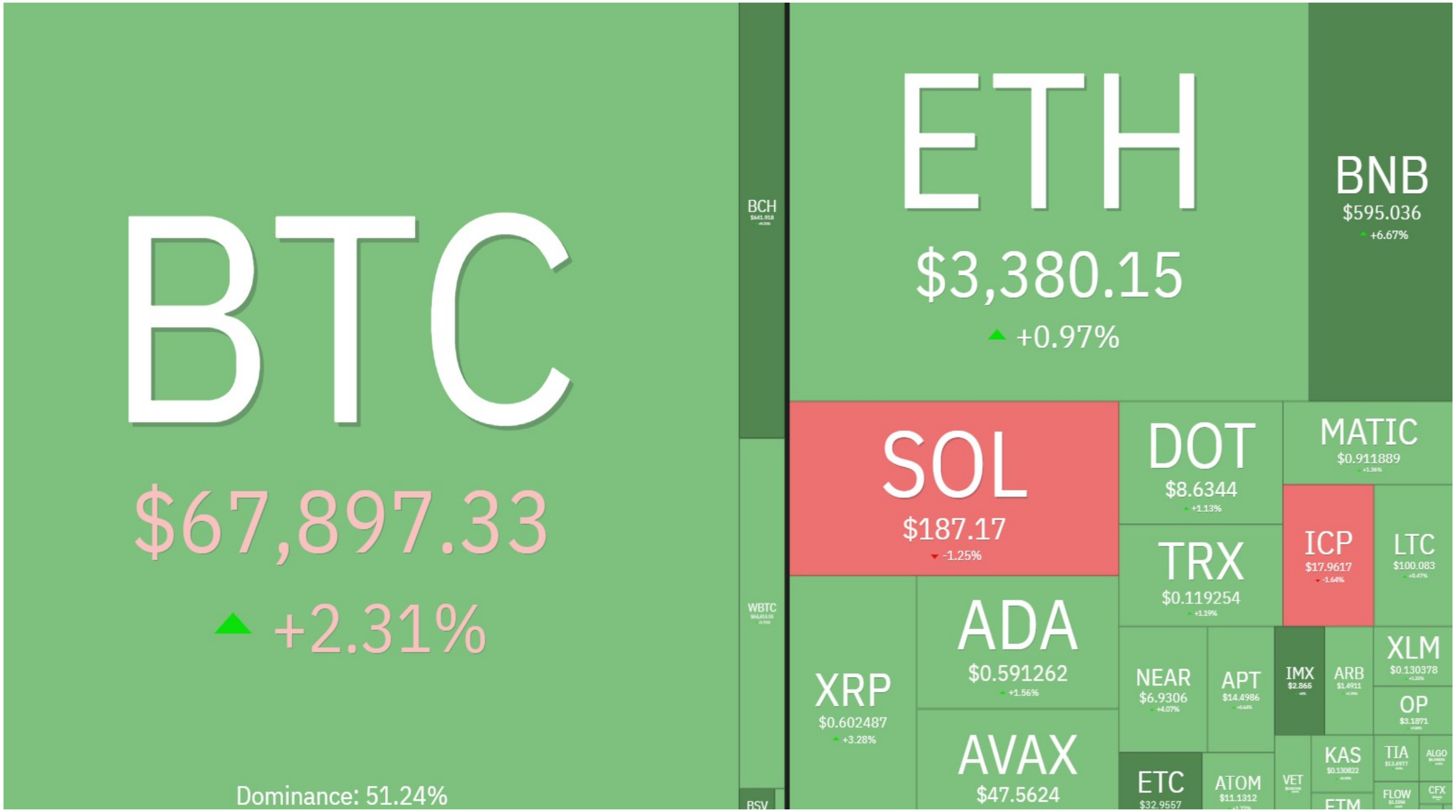


- distributed consensus to decide which block has to be added to the blockchain
- not a topic of this course

BITCOIN: WHAT IS IT?

- the "killer application" for blockchain
- a cryptocurrency invented by `Satoshi Nakamoto`, started in `2009`
- a decentralized digital currency
  - no central authority (no central bank or state issue or guarantee the currency)
  - cryptographic methods guarantee that no-one is cheating
    - issuing their own coins
    - stealing coins
    - double spending
    - etc….
- the protocol is still evolving, the official `Bitcoin` core is a `GitHub` repository, on which anyone can propose contributions (https://github.com/bitcoin/bitcoin)
  - goal: more efficient, faster, more secure, more anonymous,…

TONS OF BLOCKCHAINS: WHY FOCUS ON BITCOIN?



BITCOIN: AN UTXO BASED BLOCKCHAIN

- why data analysis mainly from the Bitcoin blockchain?
  - has by far the highest capitalization
  - many other cryptocurrencies are based on similar models
  - a class of blockchains: UTXO based blockchain
- the structure of Bitcoin transactions is based on a model called UXTO
  - UTXO: Unspent Transaction Output
  - why focusing on this model?
    - because it is adopted by many other cryptocurrencies (Litecoin, Solana, Cardano, ...)
- Ethereum uses a different model: account-based model
  - different transaction format
  - account based as in banks

THE STRUCTURE OF A BITCOIN

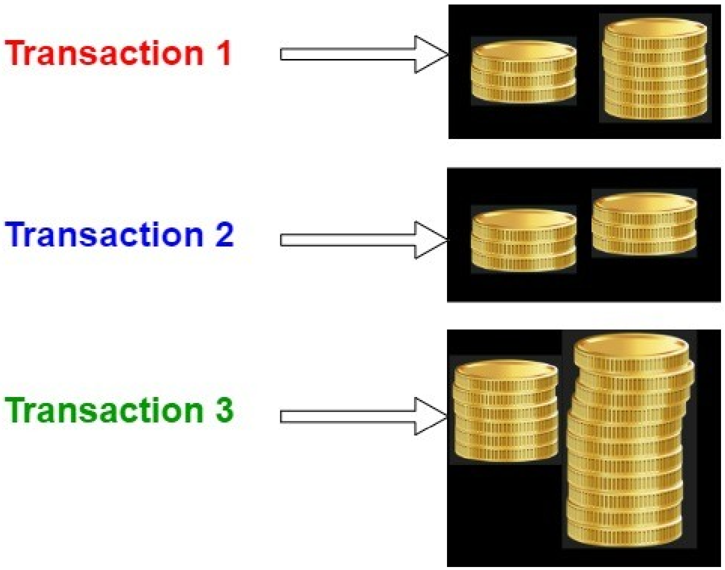
- a bank transaction

FROM	TO	AMOUNT
account1	account2	10

- a transfer of money from an account to another account
- Bitcoin address is, in some way, is similar to an account number that holds bitcoin, but there is an important difference
  - it keeps separate the bitcoin received by different transactions
  - like having a different money box for each pile of bitcoin received
  - an address is a container for several bitcoin piles
  - users may have, in their wallet, several addresses

KEEP EACH PILE OF BITCOIN SEPARATE

ADDRESS: bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh



Transaction 1

Transaction 2

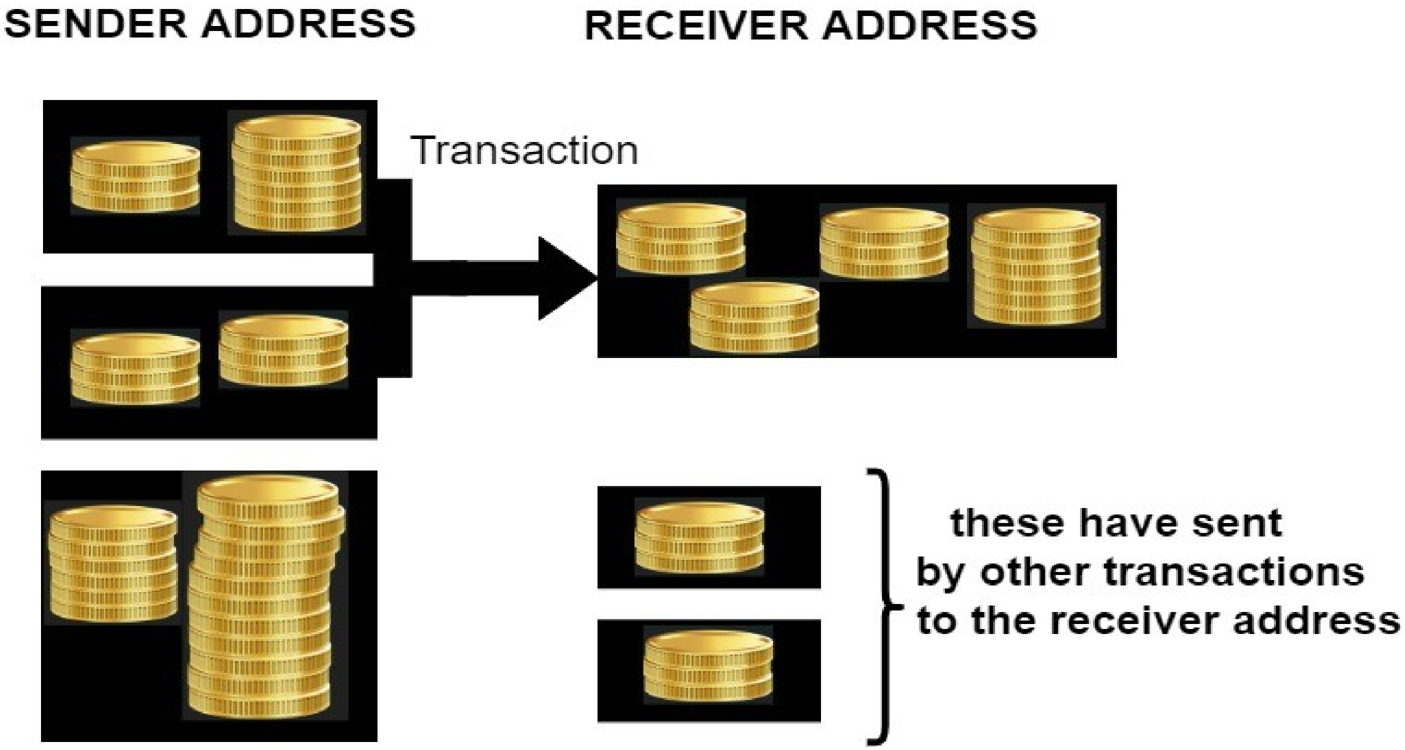
Transaction 3

- different coin piles for bitcoin received from different transactions
- in a shop, each pile corresponds to the bitcoin received from a different sale
- many piles for the same address or also different addresses with different piles

THE UXTO MODEL

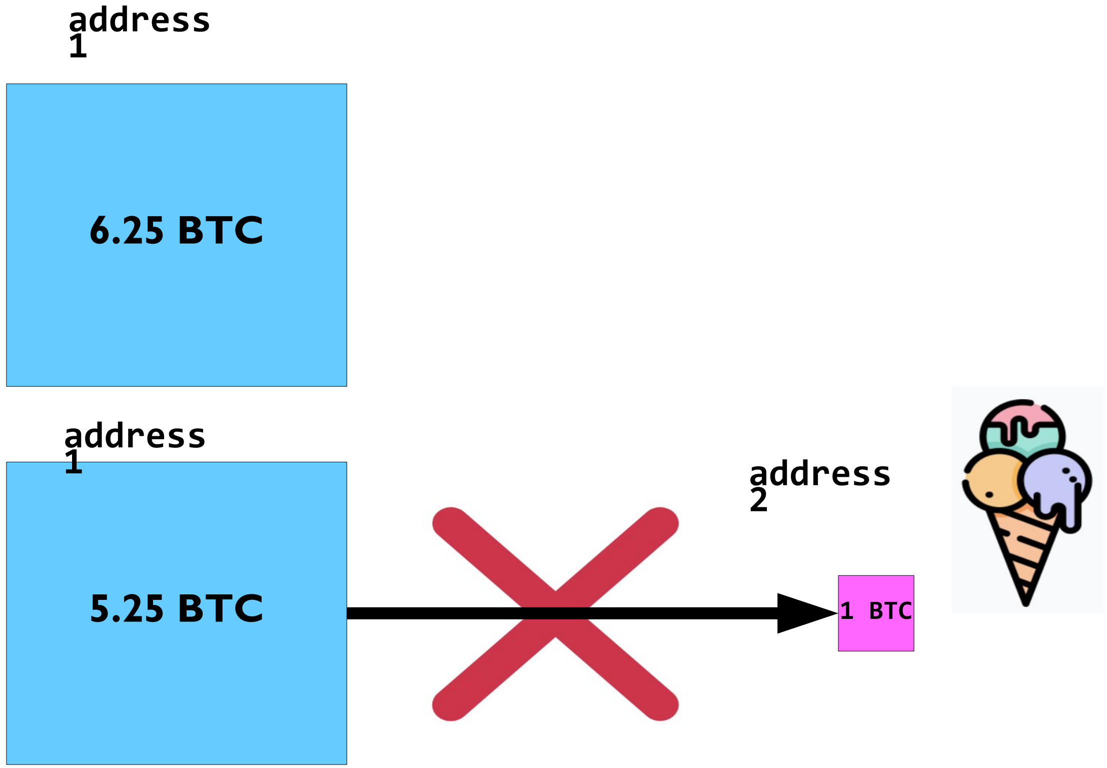
- when the user want to pay something, creates a transaction
  - uses a set of Bitcoin piles to generate a new pile to send to the receiver.
  - takes the whole amount from one or more of the stacks and sends it to the address of the receiver
  - no coin can remain in a pile
  - the receiver maintains the received bitcoin pile separated from the other one in his/her addresses
- this model is called UTXO model (Unspent Transaction Output)
  - unspent bitcoin piles are spread in Bitcoin transaction which are spread on the blockchain
  - it seems a bit complex, but it has several technological advantages (anonymity, parallelism,.....)
  - adopted by several blockchains: Bitcoin, Litecoin, Cardano,....

WHAT IS A BITCOIN TRANSACTION?



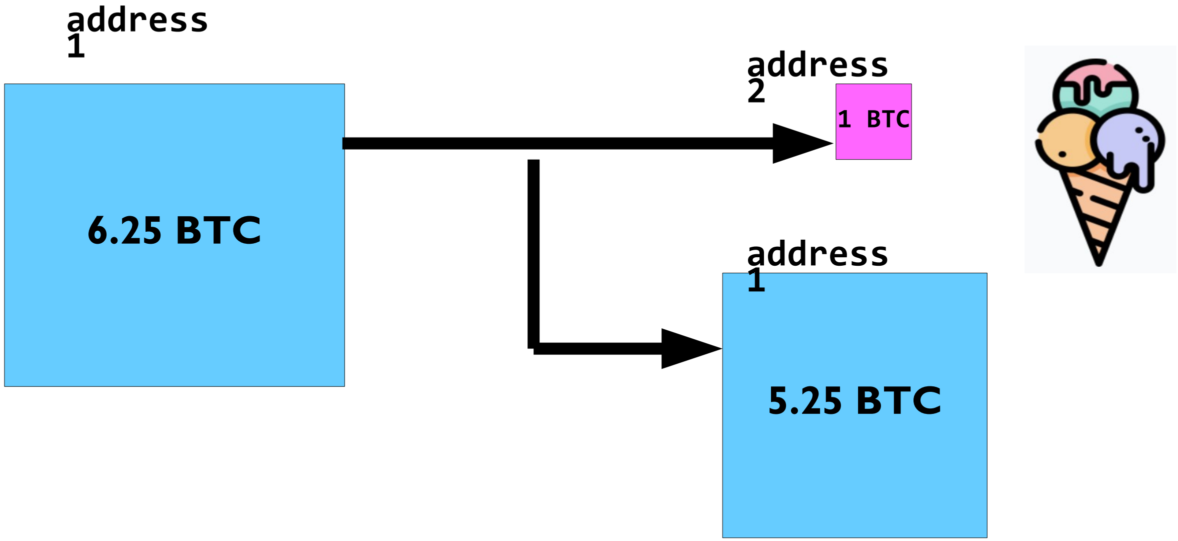
- when the user wants to pay something, creates a transaction
  - takes 2 batches of Bitcoin from the same address or from different addresses and send to the receiver
- the received Bitcoin are a separate pile in the receiver address

TRANSACTION EXAMPLES



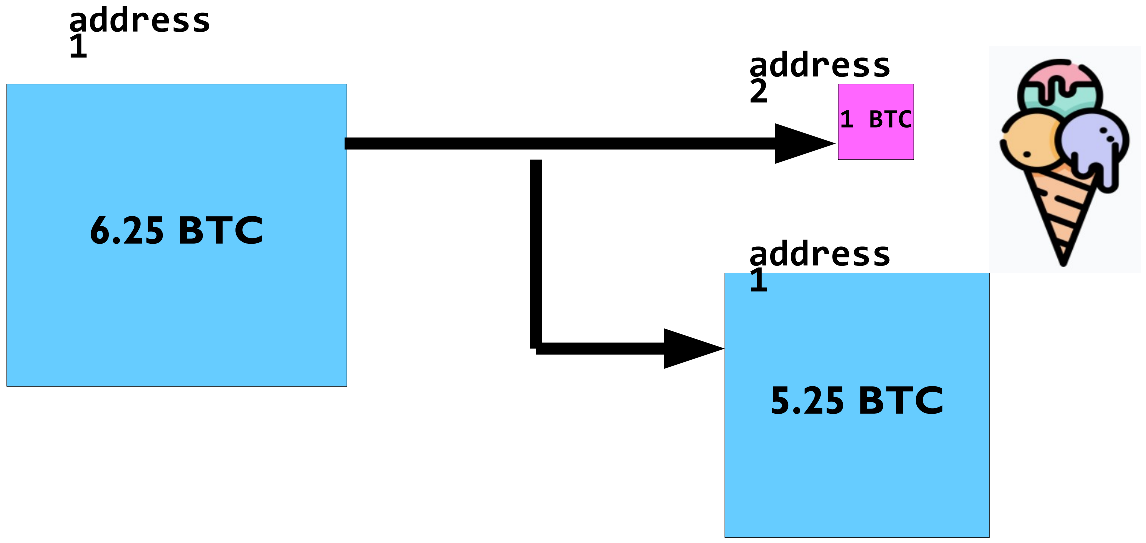
- what can I do with these `bitcoin`? Buy an ice cream cone!
- use one of these bitcoin to pay the ice cream (it's a joke, it would be a very expensive ice cream!)
- but this does not work with the UTXO model, because all the bitcoin in the pile must be spent

TRANSACTION EXAMPLES



- in the UTXO model, you have to consume the entire pile of `6.25 BTC`:
- can leave a change in the same address but this is a new pile in a new transaction output (but in general you use a different address)
- spit the pile of `bitcoin` up and send it to two destinations
  - the ice cream shop (the payment, in pink)
  - back to our own address (the change, a new pile, in blue)
- the original batch of `6.25` bitcoin has now been "used up", does not exist any more

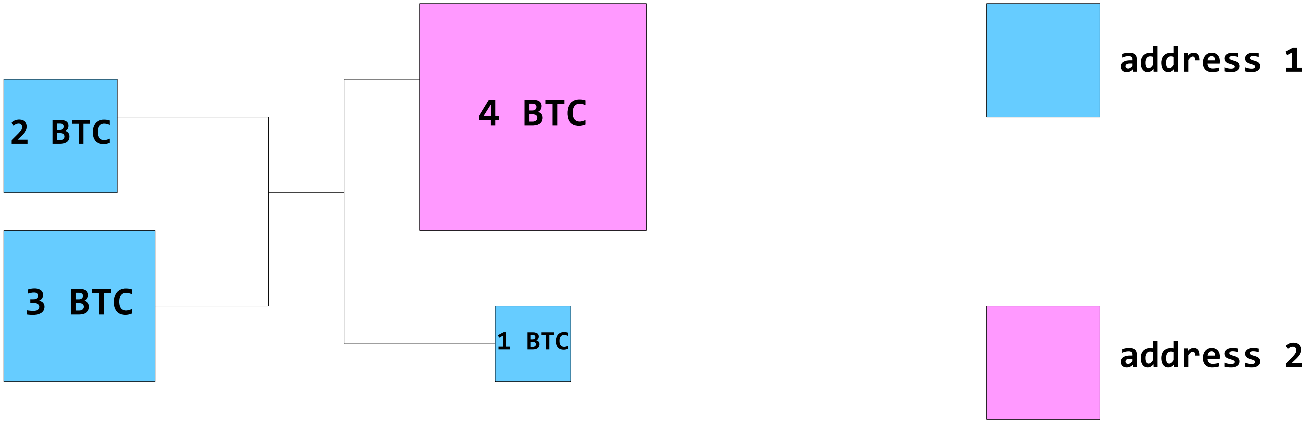
TRANSACTION EXAMPLES



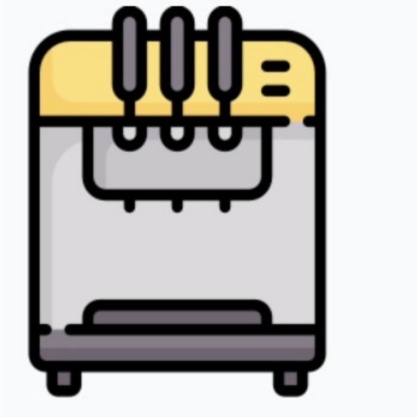
- this is what the Bitcoin transaction system is designed to do
  - take an existing transaction output (a pile of bitcoins)
  - this is a transaction through which you have received bitcoin
  - create newly-sized outputs (batches) from it
  - send those outputs to different addresses

GENERIC STRUCTURE OF A TRANSACTION

- generally the total of the bitcoin's pile may add up to more than the user want to spend
  - in this case just add another output to the transaction and send the difference back to this output
    - like a change
  - generate a new money box for the change
  - the procecedure is performed automatically by your wallet software

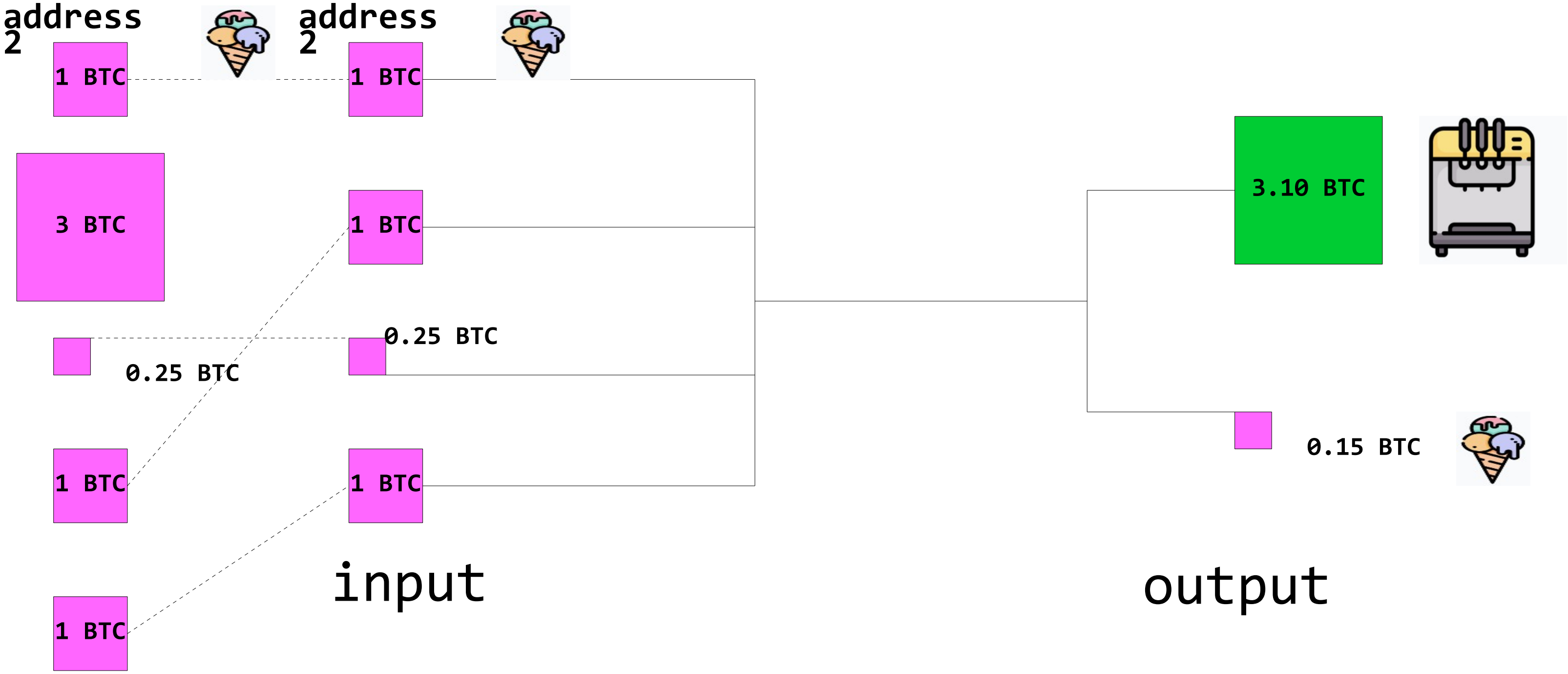


TRANSACTION EXAMPLES



- after a while, the ice cream shop has received a lot of payments
  - the ice cream business is booming!
- the shop decides to buy a new ice cream machine for `3.10 BTC`

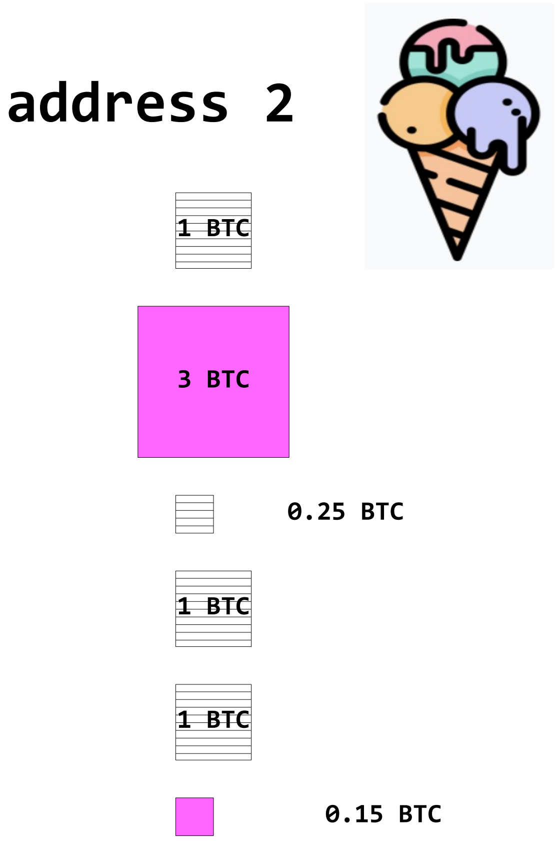
TRANSACTION EXAMPLES



the shop decides to buy a new ice cream machine for `3.10 BTC`

- the ice cream shop does not have a single pile at its address to cover the cost of the machine
- it gathers a handful of outputs to have a sum `> 3.10`
- the output gathered from previous batches (transaction outputs) are the inputs for this transaction

STATE OF THE BATCHES AFTER THE TRANSACTION



address 2

1 BTC

3 BTC

0.25 BTC

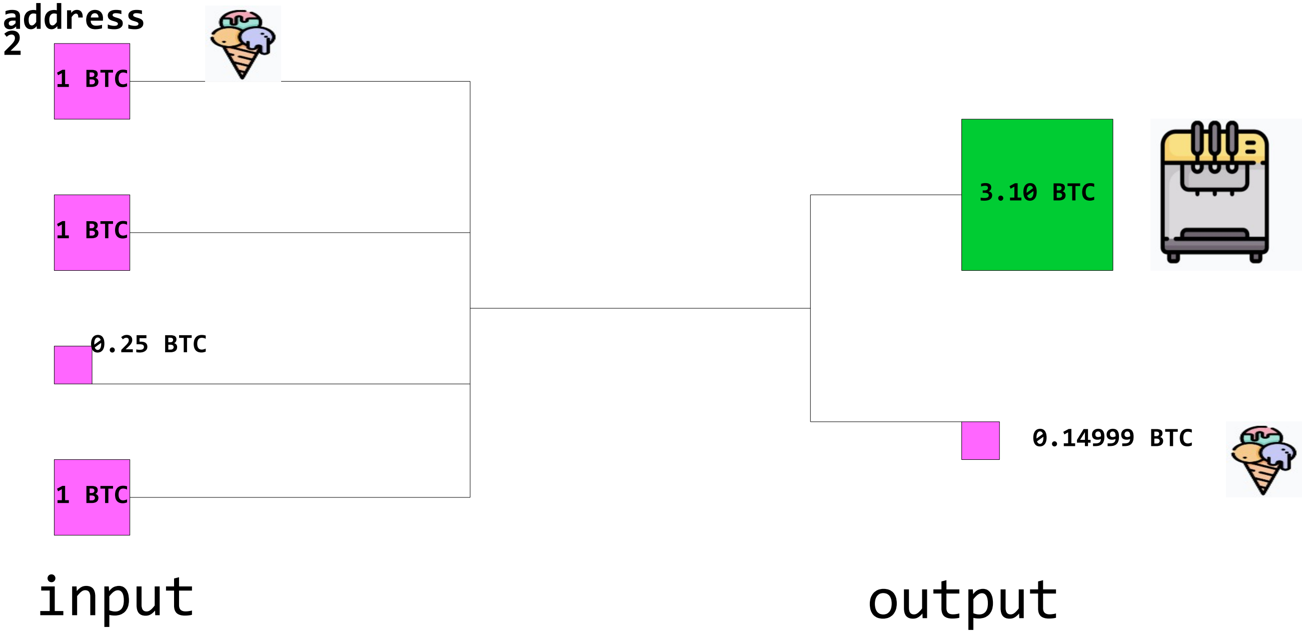
1 BTC

1 BTC

0.15 BTC

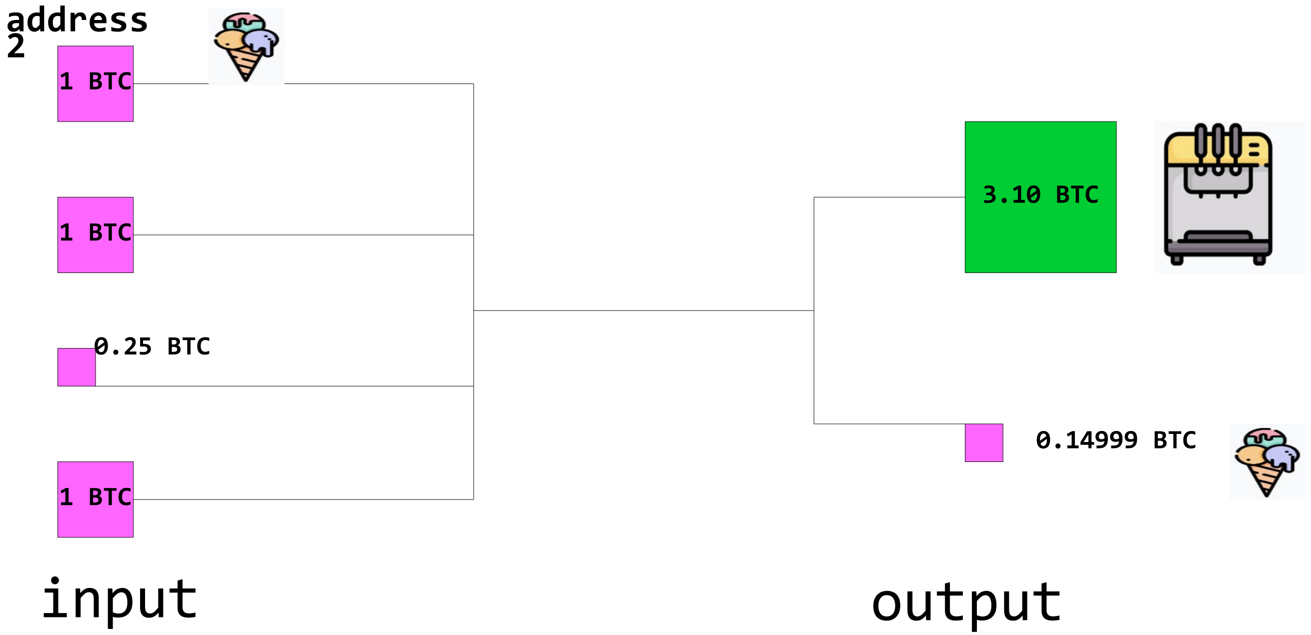
- the "unspent output" (purple squares) are still good for spending
  - these are called `Unspent Transaction Outputs (UTXO)`
- the total number of bitcoins of an address is the sum of the address's `UTXO`

TRANSACTION FEES



- observe that in this transaction the total of the outputs is less than the total of the inputs
  - 3.25 bitcoin as input 3.249999 the output
- there are some remaining bitcoins that aren't being used up.
  - this "left over" amount is the transaction fee.

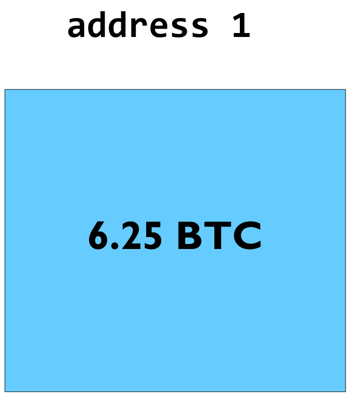
TRANSACTION FEES



- the transaction fees are picked up by miners when they mine a block
  - adding a transaction fee basically is an incentive for miners to include your transaction in a block: gives your transaction priority.
- without a transaction fee, a transactions will probably take a while to get included in to a block.

TRANSACTION EXAMPLES: COINBASE

- Coinbase transactions
  - transactions minting bitcoins
  - send “new bitcoin” to miners solving the Proof of Work, as a reward



address 1

6.25 BTC

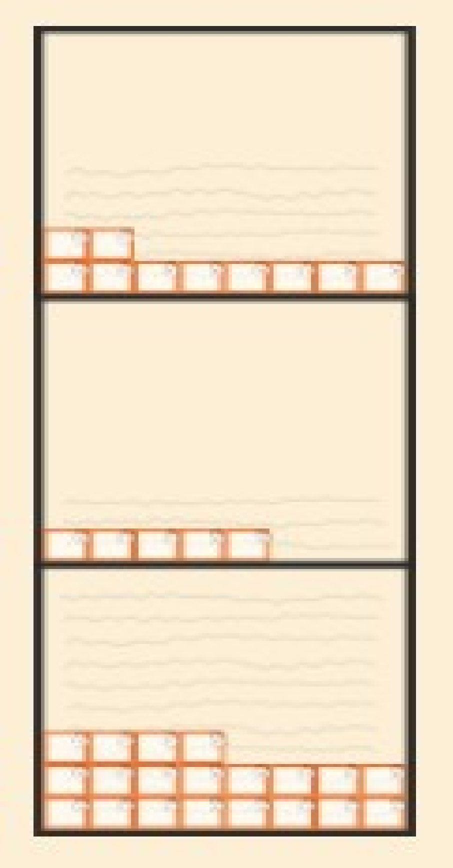
- no input to this transaction
- Bitcoin are not taken from a previous pile, they are generated by the system!

THE UXTO MODEL

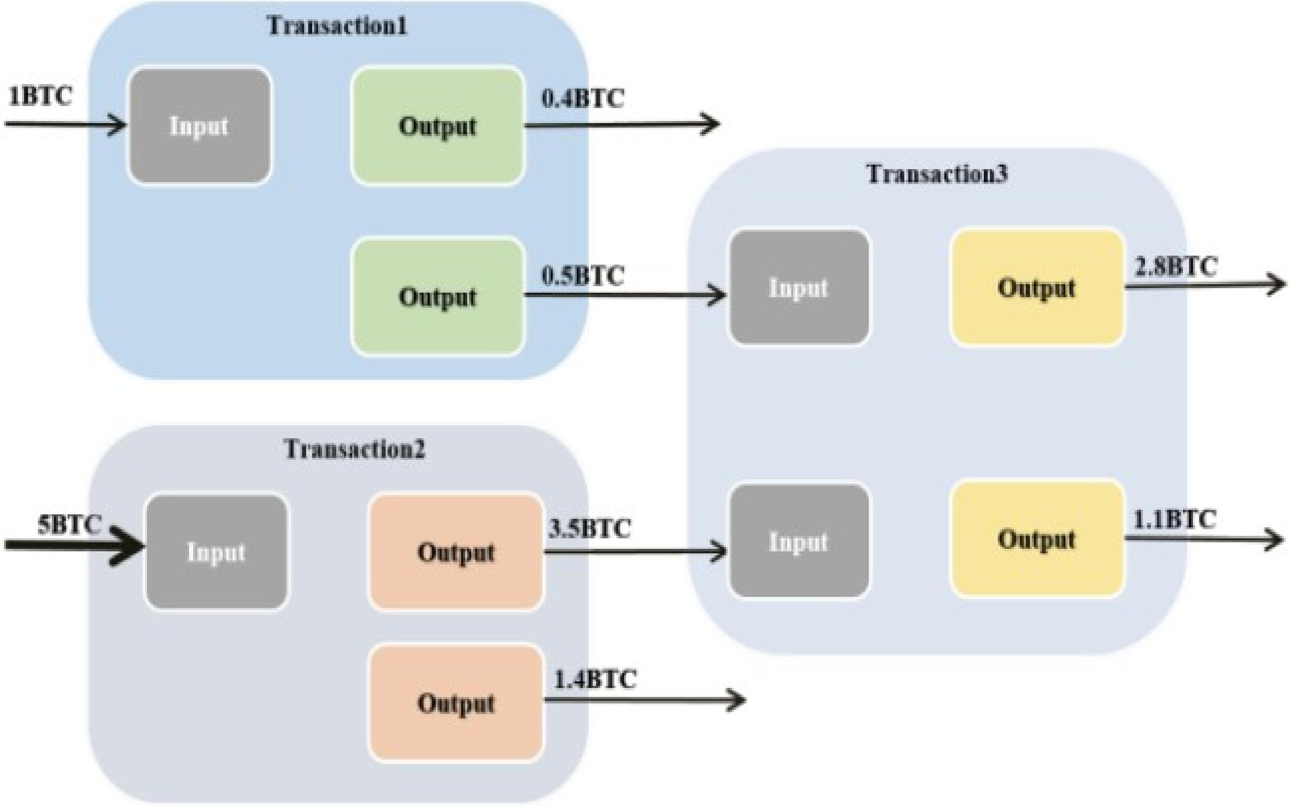
- transactions are stored on the blockchain

- output of transaction are like slot on the blockchain storing "pile of bitcoin" to be spent

- from now on, we will represent a transaction output with □

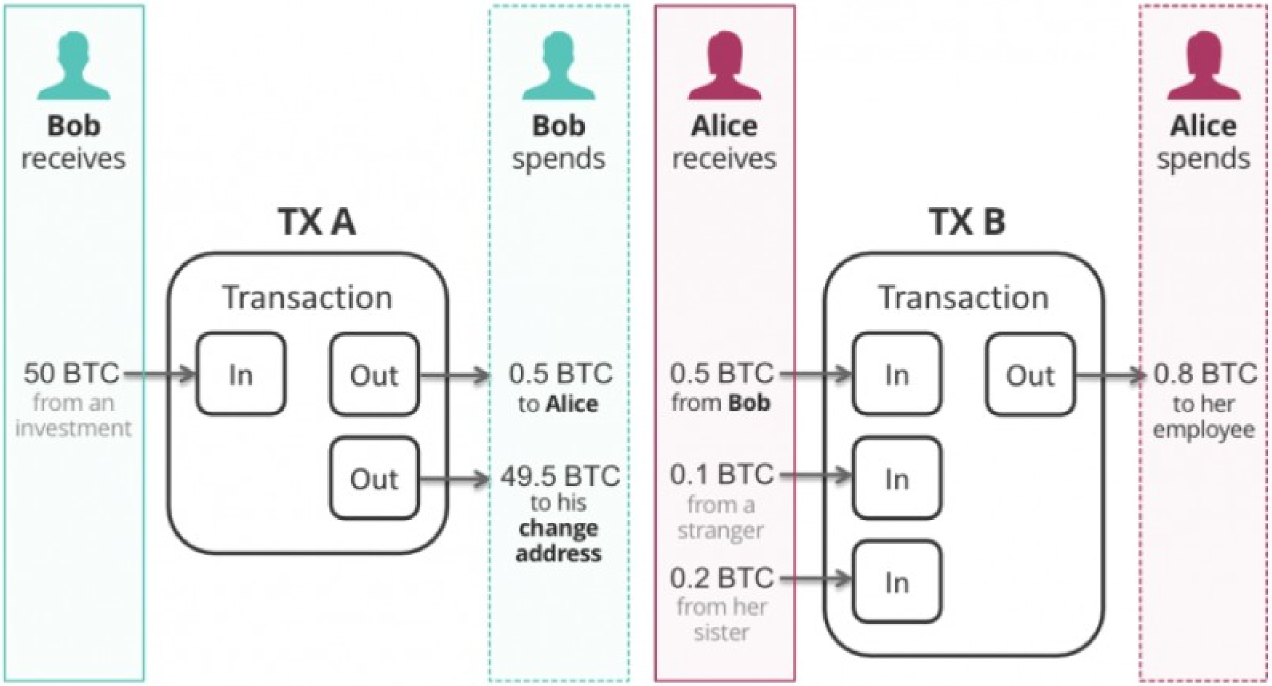


THE FORMAT OF THE TRANSACTIONS



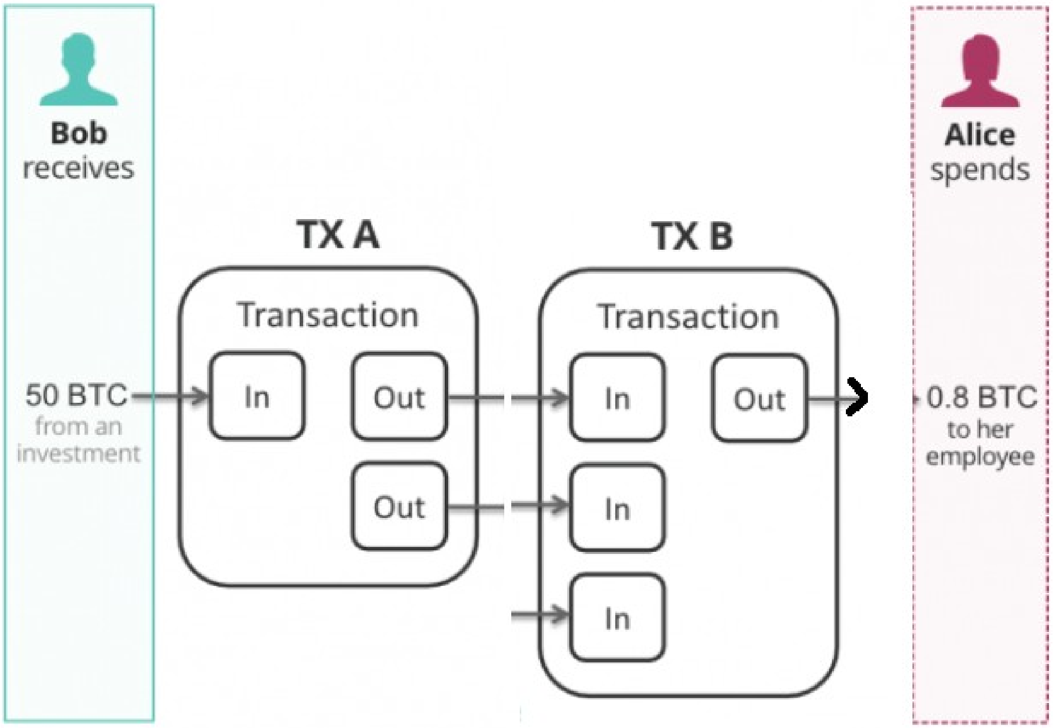
- but where are really these stack of bitcoins?
- actually, they are stored in the output of the transactions, which are registered on the blocks of the blockchain

BITCOIN TRANSACTIONS CHAIN



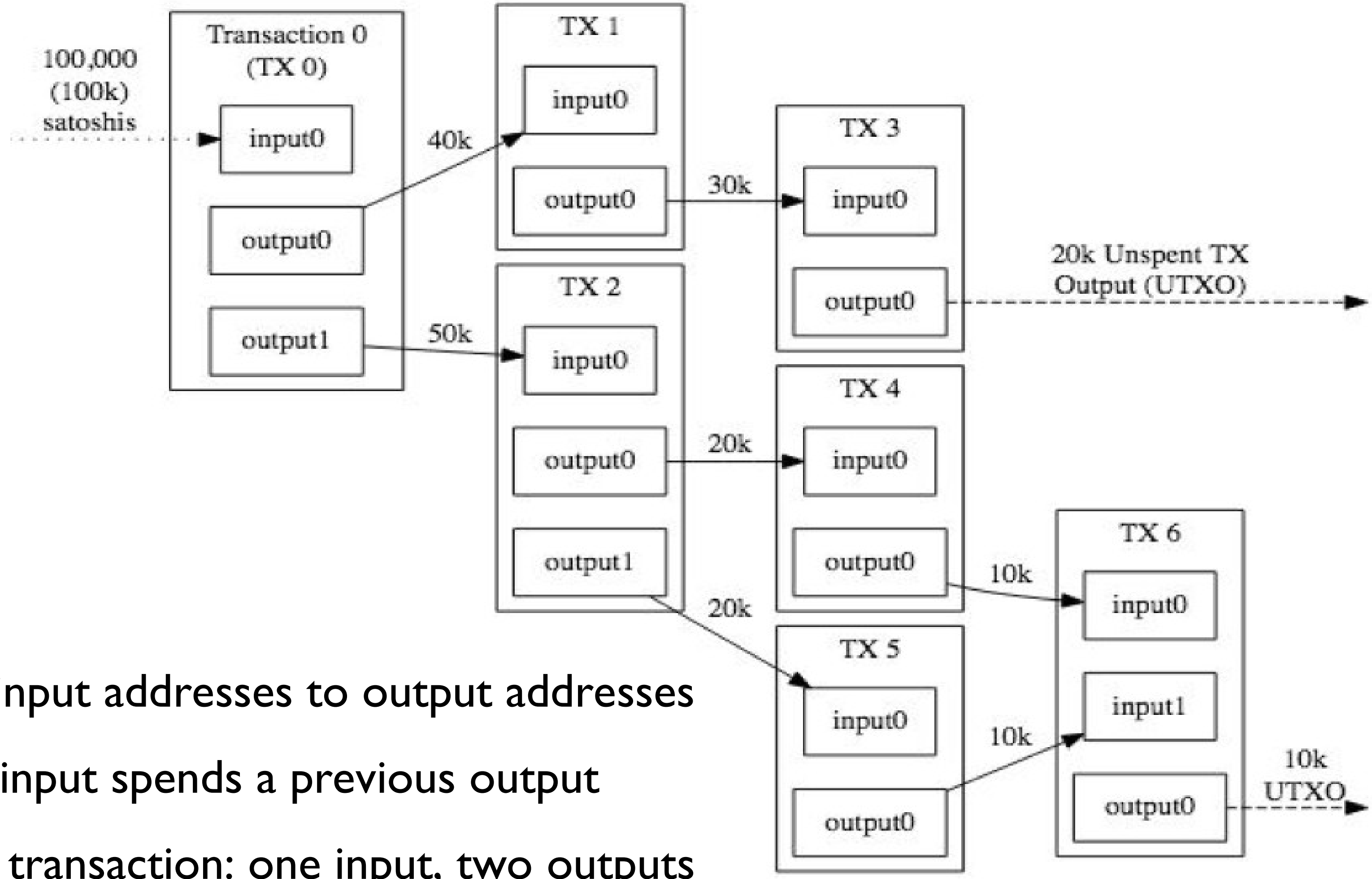
- a chain of ownership is registered on the blockchain
- value is moved from address to address

BITCOIN TRANSACTIONS CHAIN



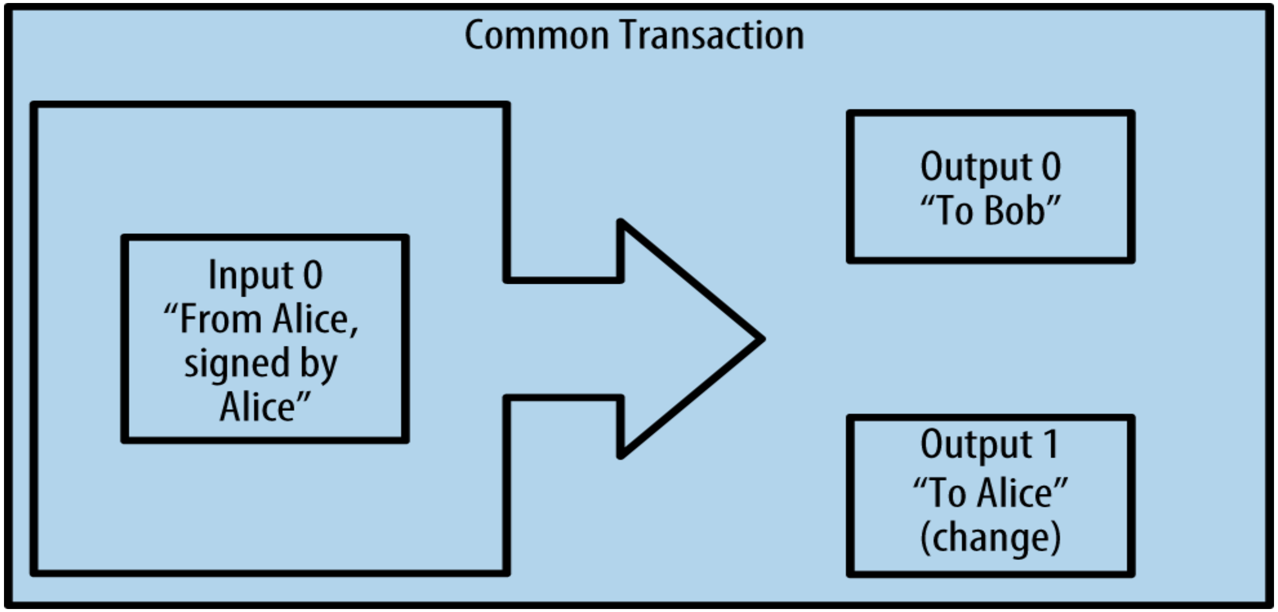
- a chain of ownership is registered on the blockchain
- value is moved from address to address...

THE UXTO MODEL



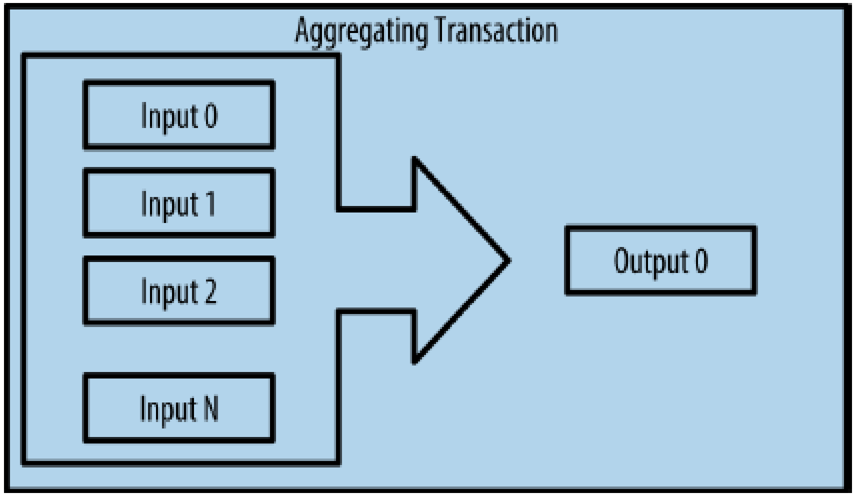
- map input addresses to output addresses
- each input spends a previous output
- usual transaction: one input, two outputs
- only unspent output are significative: mainteined in the UXTO

COMMON TYPE OF TRANSACTIONS



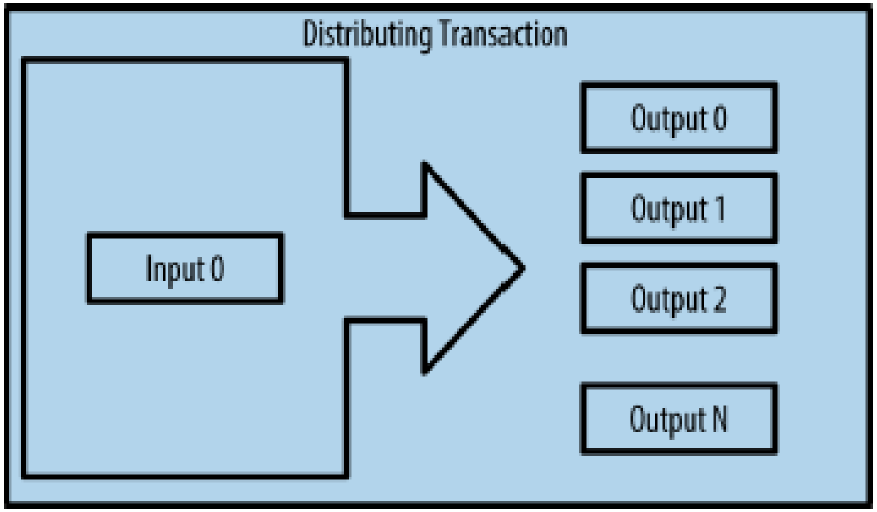
- the most common form of transaction: a simple payment from one address to another
- often includes some "change" returned to the original address.
- this type of transaction has one input and two outputs

TYPE OF TRANSACRIONS: AGGREGATING FUNDS



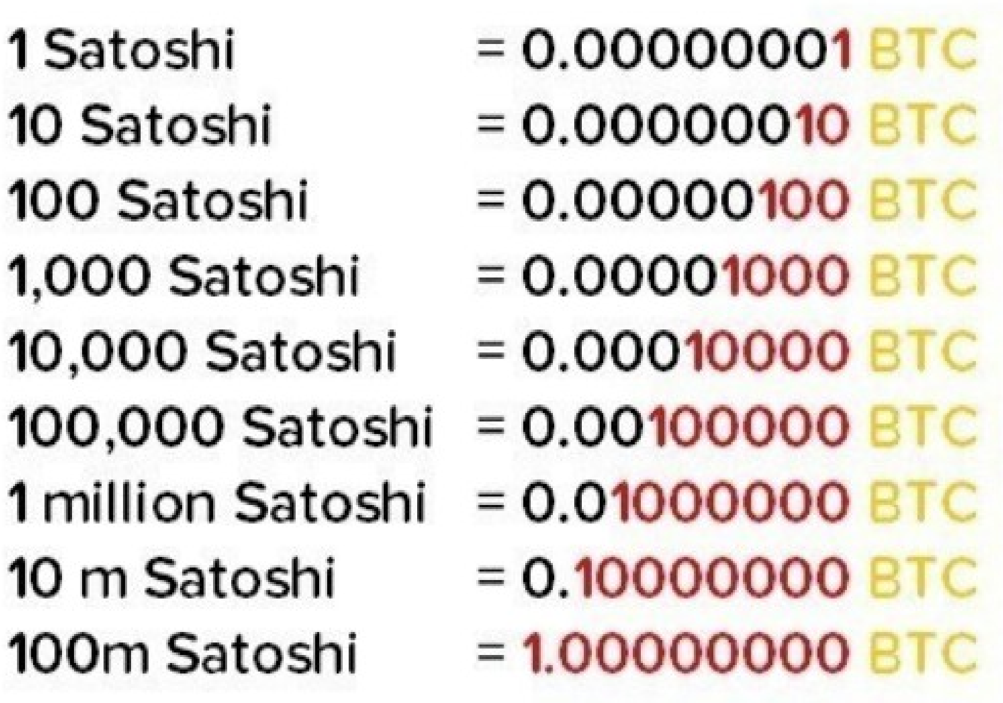
- a transaction aggegating several inputs into a single output
  - the equivalent of echanging a pile of coins for a single larger note
- may be generated to clean up lots of smaller amounts that were received as change for payments (generated by wallet applications)
- merging funds belonging to the same user in the output of the transaction, but exploited also for joint payments (multisignature transactions)

TYPE OF TRANSACTIONS: DISTRIBUTING FUNDS



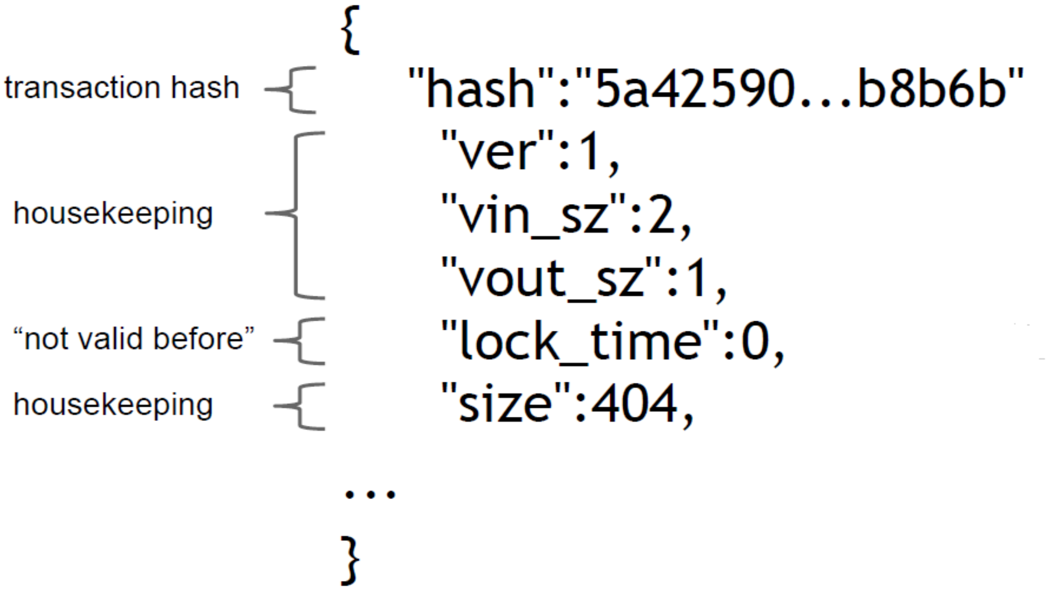
- transactions distributing one input to multiple outputs representing multiple recipients
- used to distribute funds, for instance processing payroll payments to multiple employees

BITCOINs AND SATOSHIS



1 Satoshi	= 0.00000001 BTC
10 Satoshi	= 0.00000010 BTC
100 Satoshi	= 0.00000100 BTC
1,000 Satoshi	= 0.00001000 BTC
10,000 Satoshi	= 0.00010000 BTC
100,000 Satoshi	= 0.00100000 BTC
1 million Satoshi	= 0.01000000 BTC
10 m Satoshi	= 0.10000000 BTC
100m Satoshi	= 1.00000000 BTC

A REAL BITCOIN TRANSACTION (JSON)

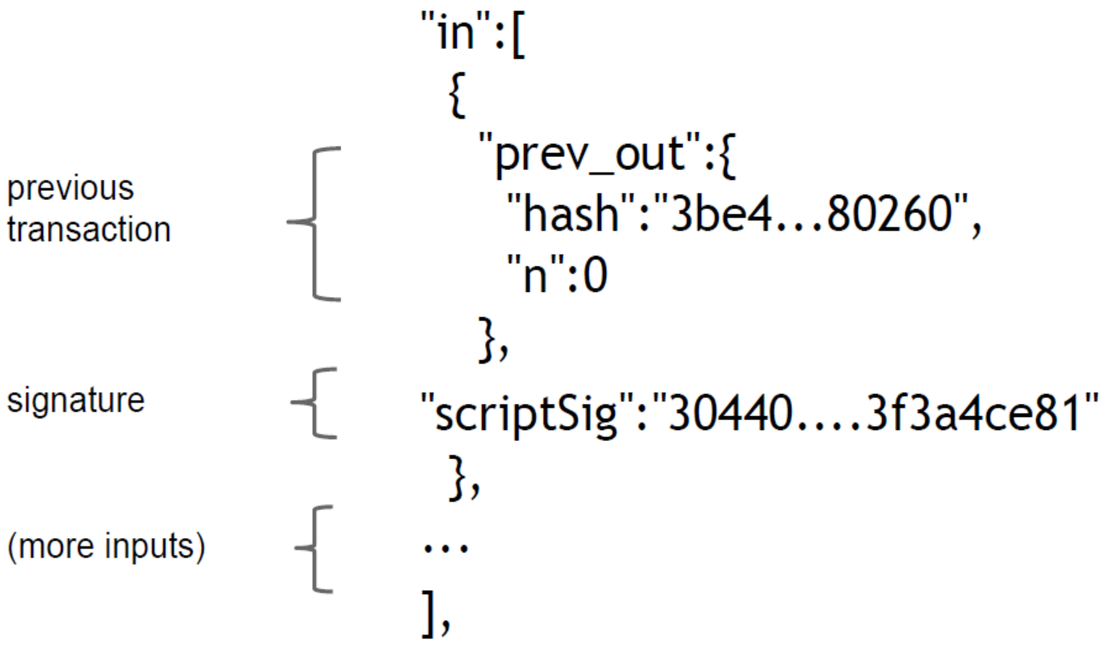


```
{
   "hash":"5a42590...b8b6b"
    "ver":1,
    "vin_sz":2,
    "vout_sz":1,
    "lock_time":0,
    "size":404,

...
}
```

- `hash` of the entire transaction, an unique identifier
- `version` allow different interpretation of some fields
- `locktime` defines the earliest time that a transaction can be added to the blockchain
    - set to zero in most transactions to indicate immediate execution
    - used in escrow and for the lightning network (a payment channel)

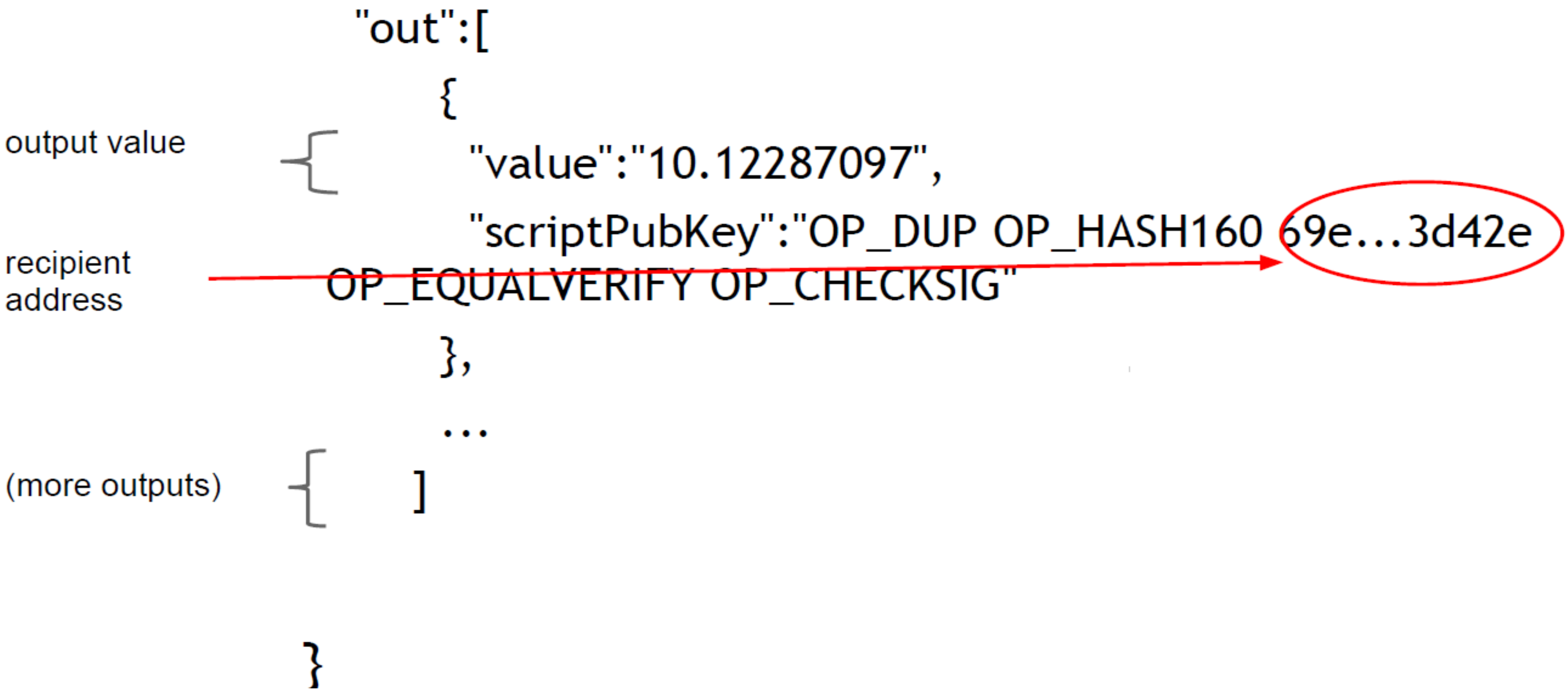
A REAL BITCOIN TRANSACTION (JSON)



```
"in":[
  {
    "prev_out":{
      "hash":"3be4...80260",
      "n":0
    },
"scriptSig":"30440....3f3a4ce81"
  },
...
],
```

- a JSON array: each element contains a key value pair
  - hash pointer to a previous transaction and index of the previous transaction's output to be spent
  - an unlocking script

A REAL BITCOIN TRANSACTION (JSON)

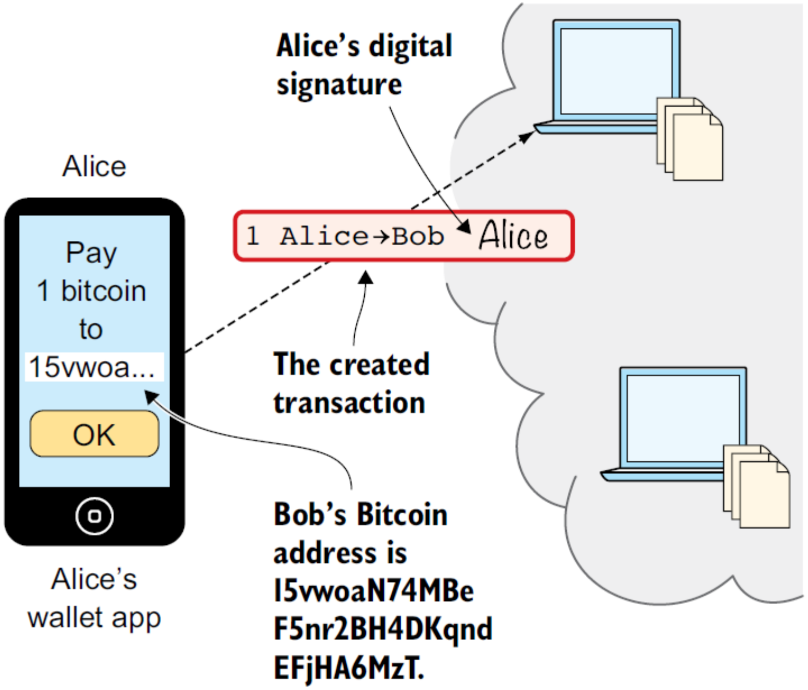


```
"out":[
      {
        "value":"10.12287097",
        "scriptPubKey":"OP_DUP OP_HASH160 69e...3d42e
OP_EQUALVERIFY OP_CHECKSIG"
      },
      ...
   ]

}
```

- a JSON array where each element contains a pair
  - value to be transferred in that output
  - a locking script containing the address where that value has to be transferred

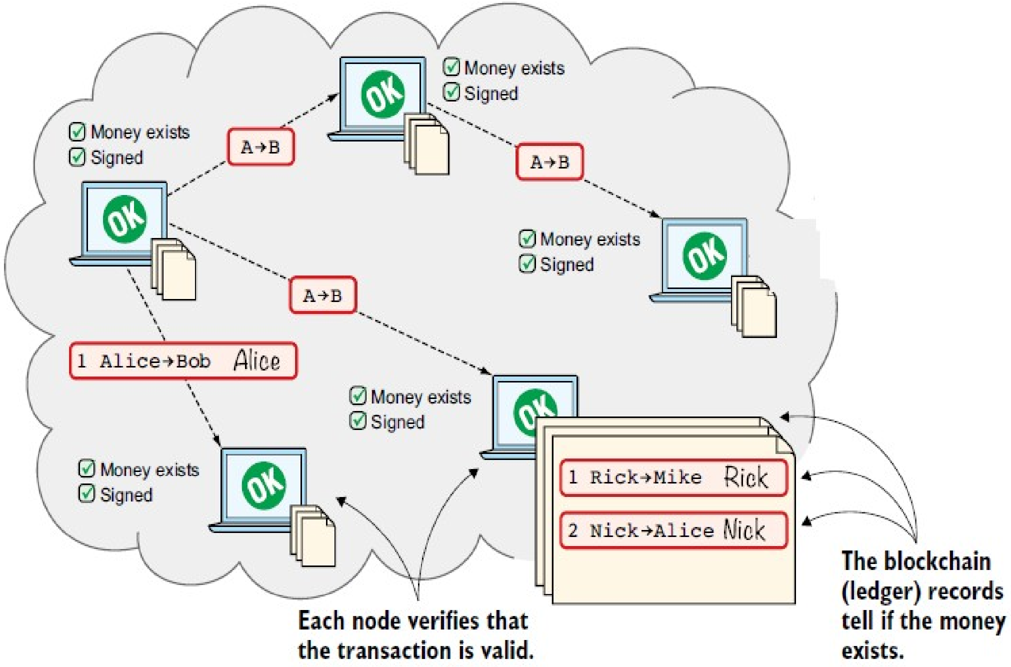
BITCOIN IN A NUTSHELL: RECAP



- what do you find inside a transaction?
  - amount to move (ex: `1bitcoin`)
  - the address of the receiver, something like
    I5vwoaN74MBeF5nr2BH4DKqndEFjHA6MzT
  - a digital signature
    - created through Alice's private key
- in this case, the transaction starts from Alice's mobile wallet app

BITCOIN IN A NUTSHELL: RECAP

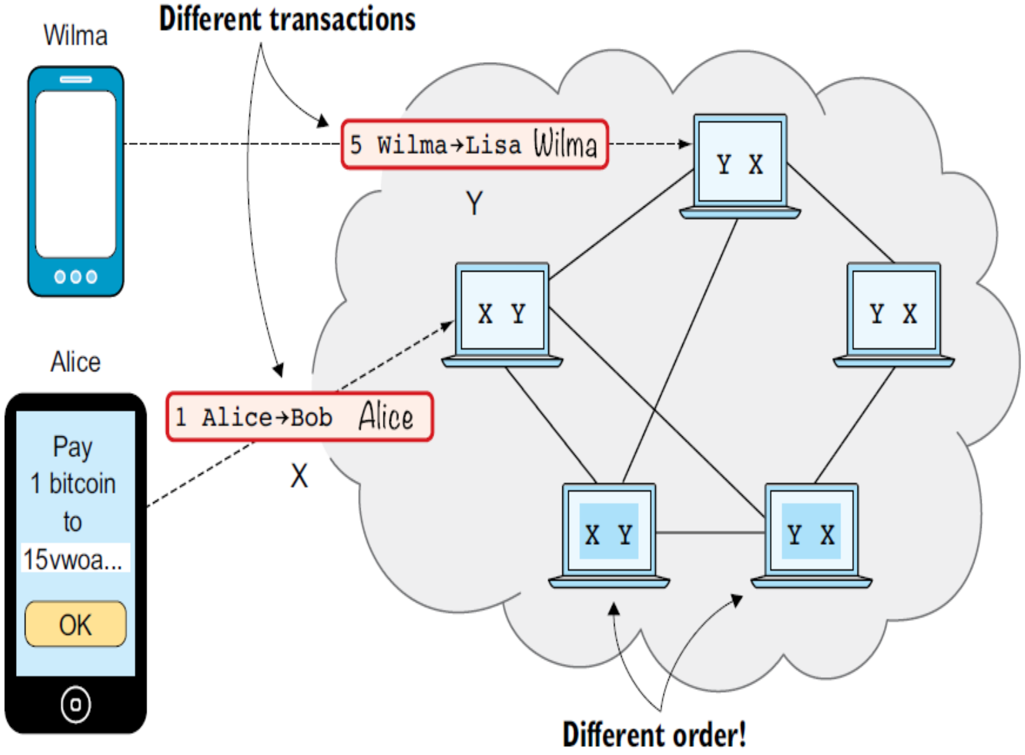
- the transaction is propagated on the P2P network
- each node checks that the transaction is valid
  - the bitcoin Alice is spending exists in a transaction output
  - Alice's digital signature is valid
- to decide if Alice can spend the bitcoin, check the blockchain



Each node verifies that the transaction is valid.

The blockchain (ledger) records tell if the money exists.

BITCOIN IN A NUTSHELL: RECAP



- the ledger stores the history of all transactions

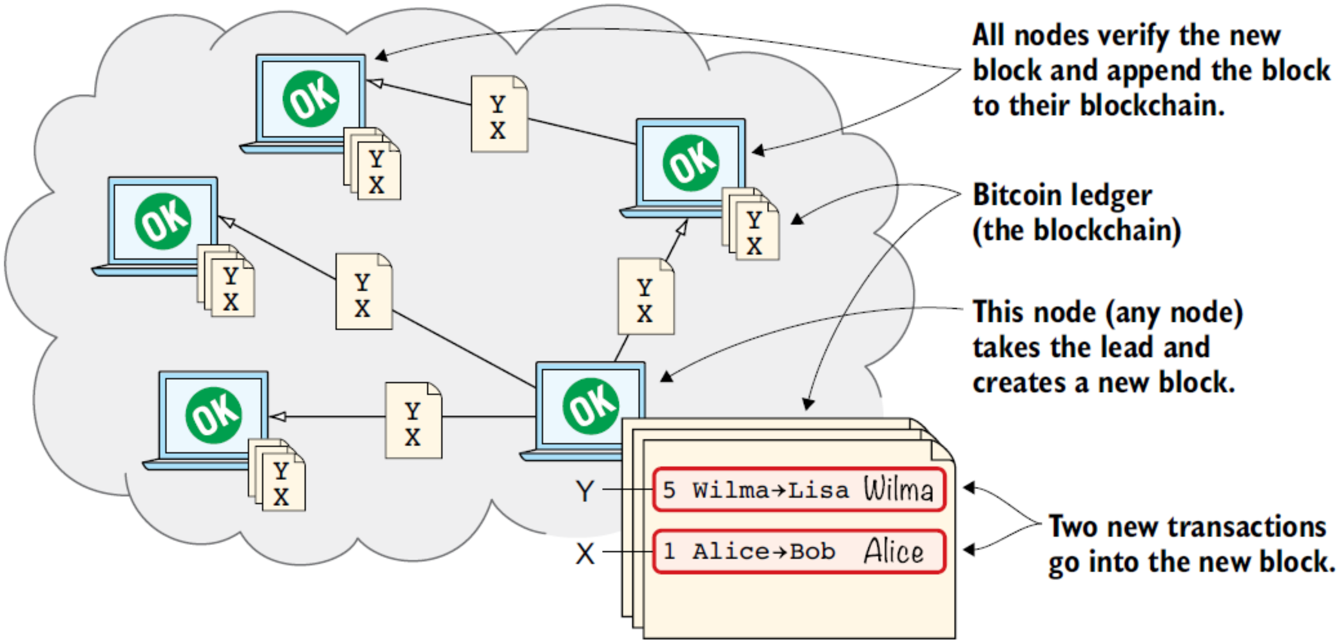
transactions

- may arrive in different orders on different nodes
- due to different network delays

what is needed?

- one node taking the lead and deciding the next transaction to add
- all the other nodes agree

BITCOIN IN A NUTSHELL: RECAP



All nodes verify the new block and append the block to their blockchain.

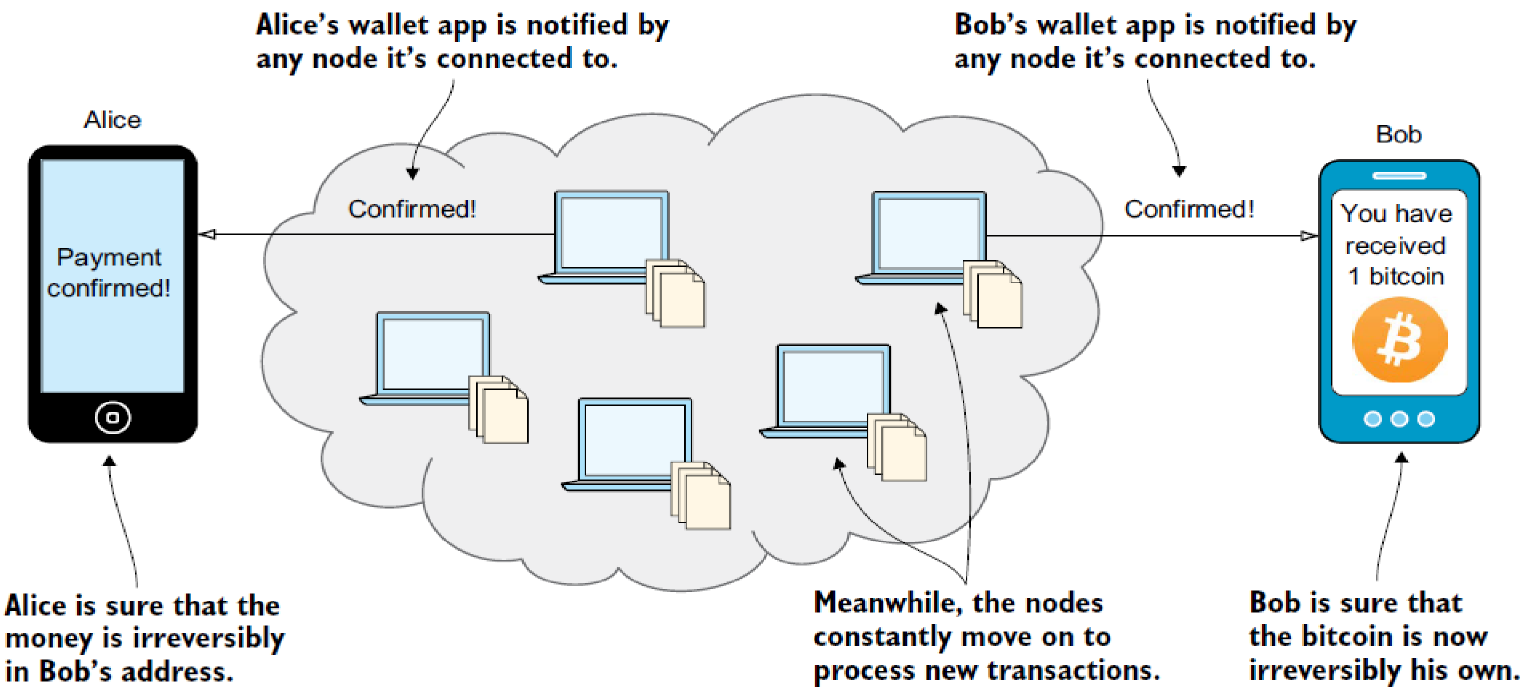
Bitcoin ledger (the blockchain)

This node (any node) takes the lead and creates a new block.

Two new transactions go into the new block.

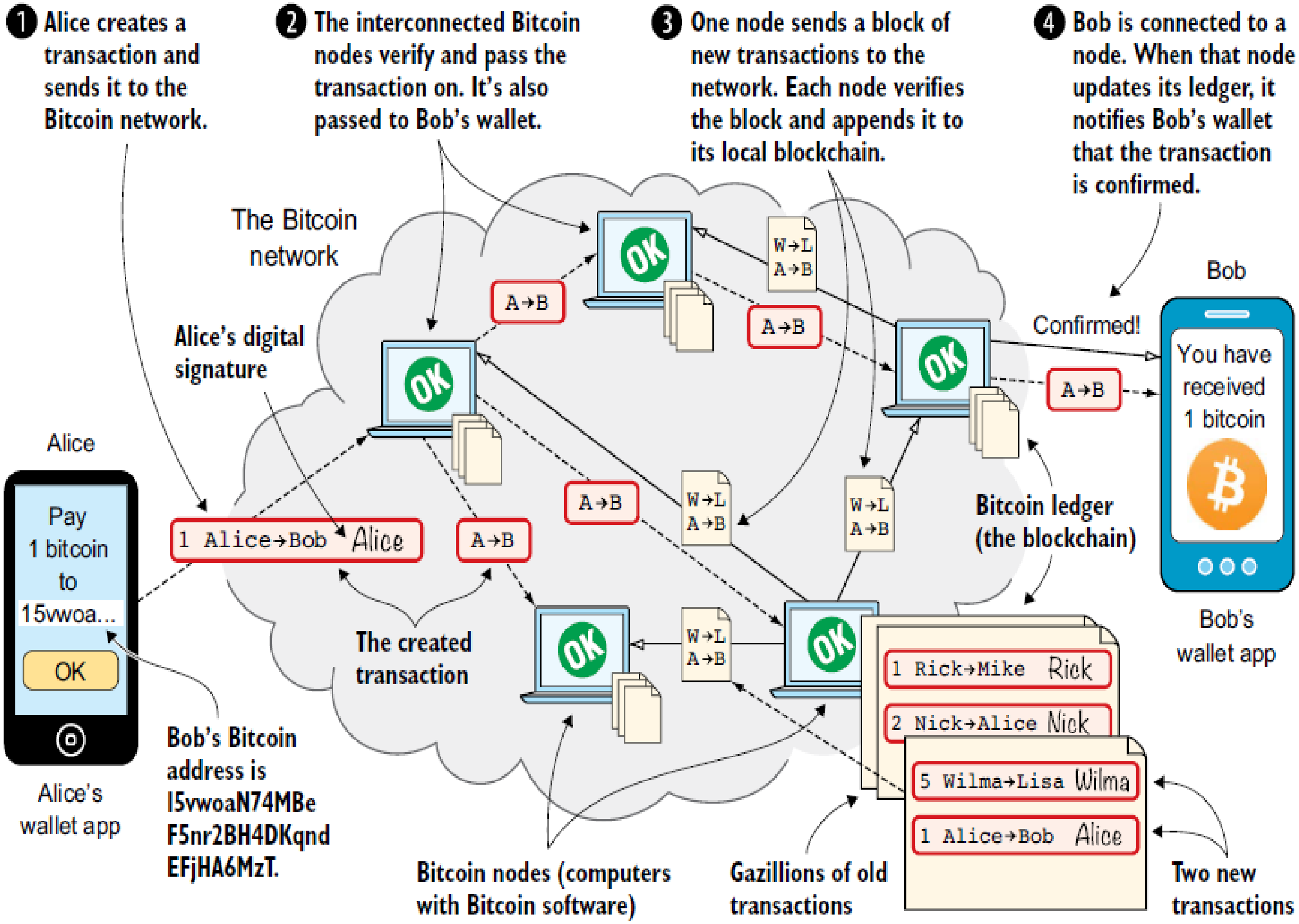
- one node takes the lead and tells the others in what order to add transactions.
  - yhis is implemented through consensus
  - sends the block on the network, as the transactions in the previous step
- the other nodes verify the block and update their blockchain copies accordingly.

BITCOIN IN A NUTSHELL: RECAP


Alice's wallet app is notified by any node it's connected to.
Bob's wallet app is notified by any node it's connected to.
Alice
Payment confirmed!
Confirmed!
Confirmed!
Bob
You have received 1 bitcoin
Alice is sure that the money is irreversibly in Bob's address.
Meanwhile, the nodes constantly move on to process new transactions.
Bob is sure that the bitcoin is now irreversibly his own.

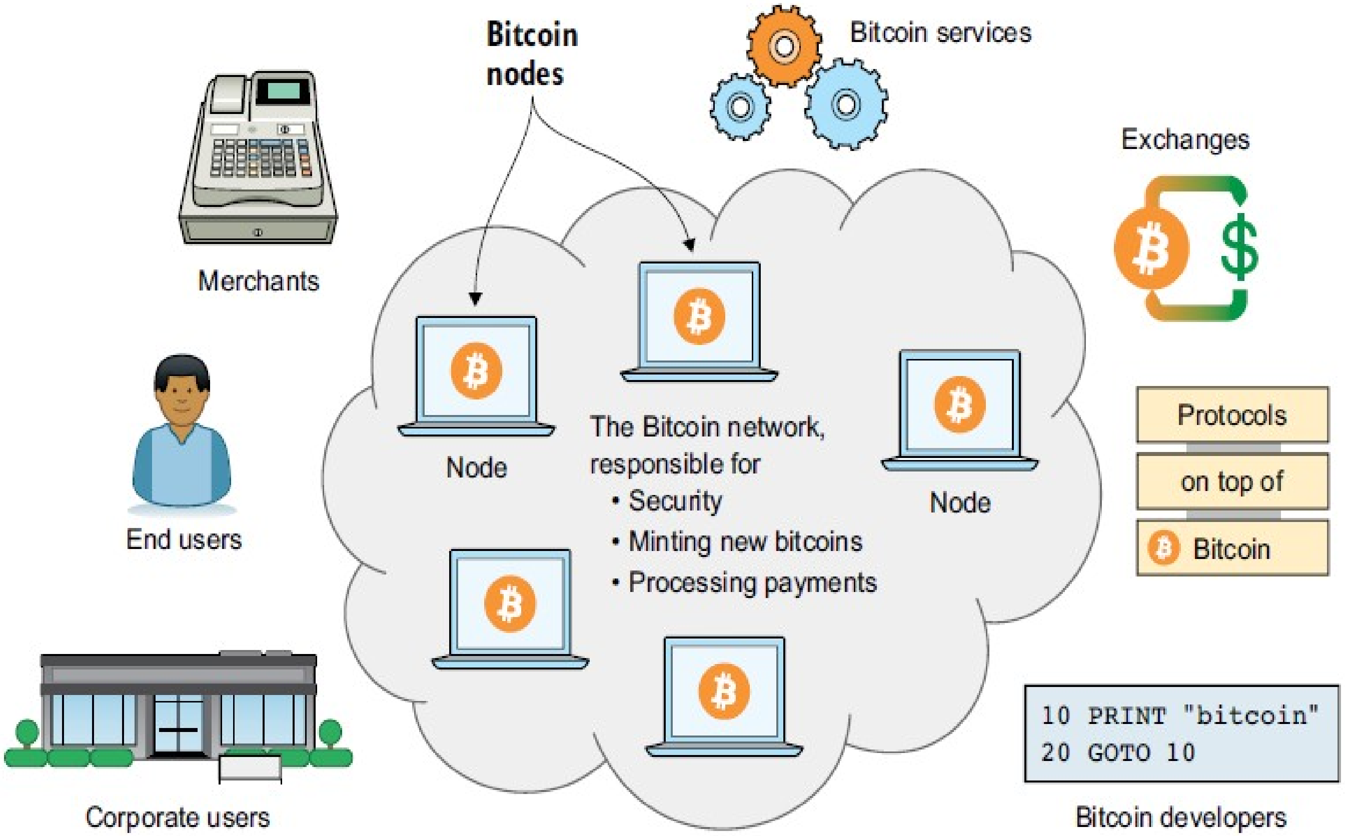
- when all the nodes have updated their local copy of the blockchain, the transaction of `Alice` has been accepted
- the node that the `Bob`'s wallet is connected to will notify `Bob`'s wallet
- the wallet display a message to `Bob` notifying that `Bob` has received 1 bitcoin

THE BITCOIN SYSTEM: ALL TOGETHER



❶ Alice creates a transaction and sends it to the Bitcoin network.
❷ The interconnected Bitcoin nodes verify and pass the transaction on. It's also passed to Bob's wallet.
❸ One node sends a block of new transactions to the network. Each node verifies the block and appends it to its local blockchain.
❹ Bob is connected to a node. When that node updates its ledger, it notifies Bob's wallet that the transaction is confirmed.
The Bitcoin network
Alice's digital signature
Alice
Pay 1 bitcoin to 15vwoa...
OK
Alice's wallet app
1 Alice→Bob Alice
A→B
W→L
A→B
OK
The created transaction
Bob's Bitcoin address is 15vwoaN74MBeF5nr2BH4DKqndEFjHA6MzT.
Bitcoin nodes (computers with Bitcoin software)
Gazillions of old transactions
1 Rick→Mike Rick
2 Nick→Alice Nick
5 Wilma→Lisa Wilma
1 Alice→Bob Alice
Two new transactions
Bitcoin ledger (the blockchain)
Confirmed!
Bob
You have received 1 bitcoin
Bob's wallet app

THE BITCOIN ECOSYSTEM



The Bitcoin network, responsible for

- Security
- Minting new bitcoins
- Processing payments

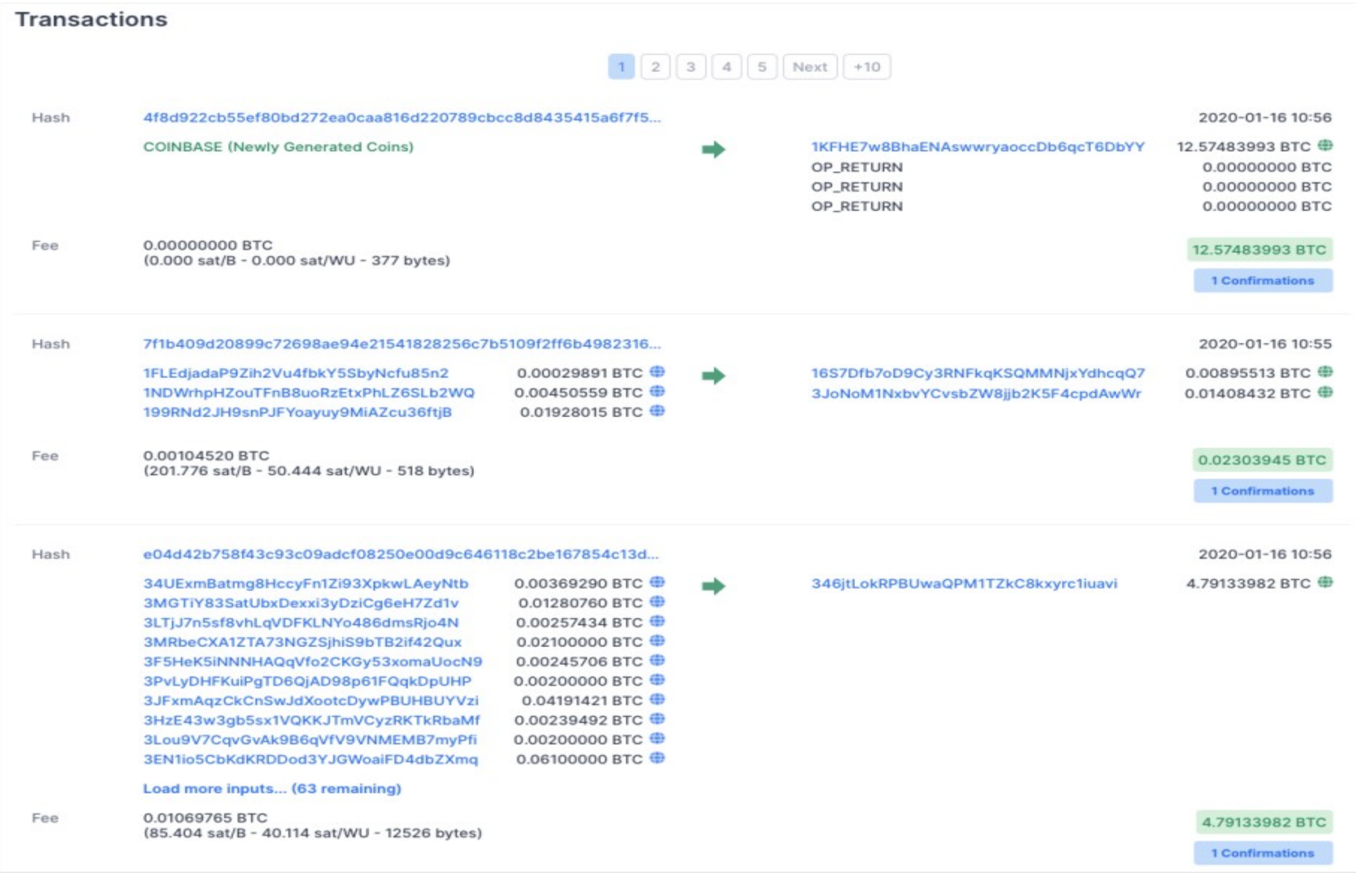
```
10 PRINT "bitcoin"
20 GOTO 10
```

TRANSACTIONS ANALYSIS: TAINT ANALYSIS

- Taint analysis is utilised as a tracking method in cryptocurrency
- tracks targeted cryptocurrency coins using transaction information in the blockchain.
    - ramsomware
    - classify the targeted cryptocurrency coins (e.g., stolen `Bitcoins` resulting from a known theft transaction) as tainted (or "dirty")
- determine the association between two addresses in a transaction
    - any address that uses or transfers them will be considered a totally or partially tainted address.
    - coins that are unrelated to tainted coins are considered clean coins.
    - different taint analysis strategy with specific rule-set to estimate how the targeted cryptocurrency coins are distributed in the subsequent transactions.

SCRAPING TO RETRIEVE BITCOIN TRANSACTIONS

- tools like `Bitcoin explorer` (`https://www.blockchain.com/explorer`)



Transactions

Hash 4f8d922cb55ef80bd272ea0caa816d220789cbcc8d8435415a6f7f5... 2020-01-16 10:56

COINBASE (Newly Generated Coins) → 1KFHE7w8BhaENAswwryaoccDb6qcT6DbYY 12.57483993 BTC
OP_RETURN 0.00000000 BTC
OP_RETURN 0.00000000 BTC
OP_RETURN 0.00000000 BTC

Fee 0.00000000 BTC
(0.000 sat/B - 0.000 sat/WU - 377 bytes)

12.57483993 BTC
1 Confirmations

Hash 7f1b409d20899c72698ae94e21541828256c7b5109f2ff6b4982316... 2020-01-16 10:55

1FLEdjadaP9Zih2Vu4fbkY5SbyNcfu85n2 0.00029891 BTC
1NDWrhpHZouTFnB8uoRzEtxPhLZ6SLb2WQ 0.00450559 BTC
199RNd2JH9snPJFYoayuy9MiAZcu36ftjB 0.01928015 BTC
→
16S7Dfb7oD9Cy3RNFkqKSQMMNjxYdhcqQ7 0.00895513 BTC
3JoNoM1NxbvYCvsbZW8jjb2K5F4cpdAwWr 0.01408432 BTC

Fee 0.00104520 BTC
(201.776 sat/B - 50.444 sat/WU - 518 bytes)

0.02303945 BTC
1 Confirmations

Hash e04d42b758f43c93c09adcf08250e00d9c646118c2be167854c13d... 2020-01-16 10:56

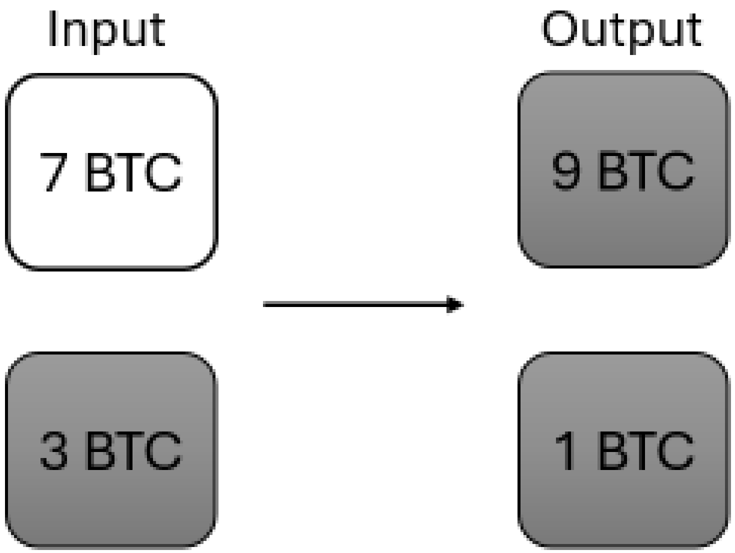
34UExmBatmg8HccyFn1Zi93XpkwLAeyNtb 0.00369290 BTC
3MGTiY83SatUbxDexxi3yDziCg6eH7Zd1v 0.01280760 BTC
3LTjJ7n5sf8vhLqVDFKLNYo486dmsRjo4N 0.00257434 BTC
3MRbeCXA1ZTA73NGZSjhiS9bTB2if42Qux 0.02100000 BTC
3F5HeK5iNNNHAQqVfo2CKGy53xomaUocN9 0.00245706 BTC
3PvLyDHFKuiPgTD6QjAD98p61FQqkDpUHP 0.00200000 BTC
3JFxmAqzCkCnSwJdXootcDywPBUHBUYVzi 0.04191421 BTC
3HzE43w3gb5sx1VQKKJTmVCyzRKTkRbaMf 0.00239492 BTC
3Lou9V7CqvGvAk9B6qVfV9VNMEMB7myPfi 0.00200000 BTC
3EN1io5CbKdKRDDod3YJGWoaiFD4dbZXmq 0.06100000 BTC
→
346jtLokRPBUwaQPM1TZkC8kxyrc1iuavi 4.79133982 BTC

Load more inputs... (63 remaining)

Fee 0.01069765 BTC
(85.404 sat/B - 40.114 sat/WU - 12526 bytes)

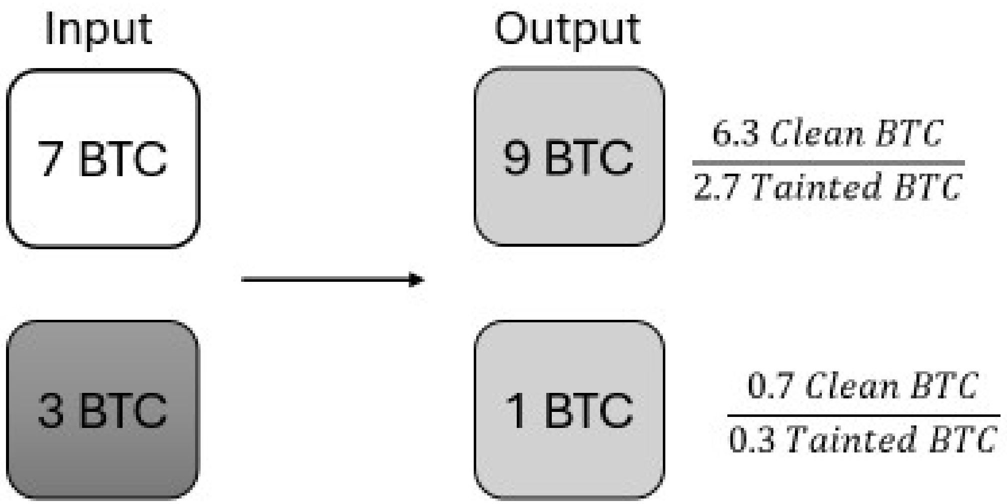
4.79133982 BTC
1 Confirmations

TAINT ANALYSIS: POISON



ogni output di una transazione è tainted se almeno un input di quella transazione lo è

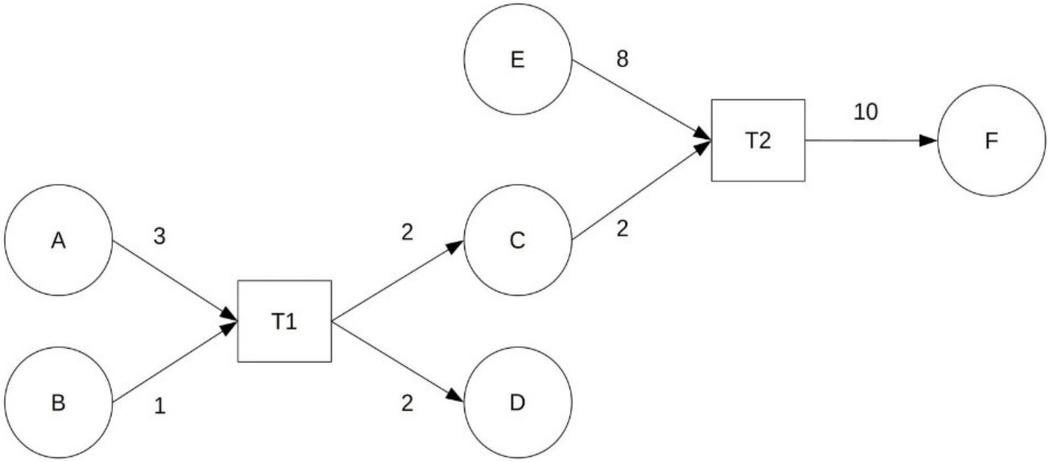
TAINT ANALYSIS: HAIRCUT


Input
Output
7 BTC
3 BTC
9 BTC
1 BTC
$\frac{6.3\ Clean\ BTC}{2.7\ Tainted\ BTC}$
$\frac{0.7\ Clean\ BTC}{0.3\ Tainted\ BTC}$

ogni output di una transazione riceve una percentuale di bitcoin proporzianata la valore di quell’output

CRYPTO ANALYSIS: HAIRCUT

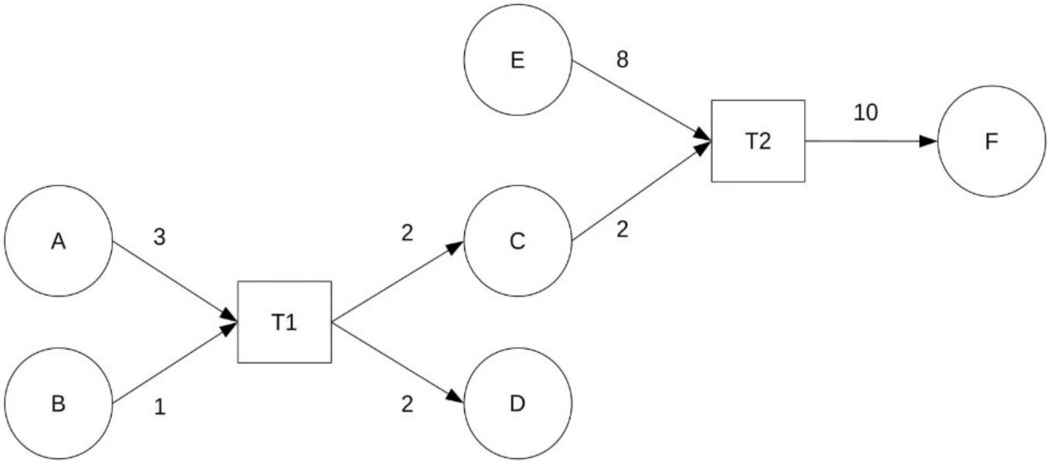
- shows how large is the percentage of a coin in one address from another address



- taint = contaminazione
- address A and B contribute to a transaction T1 with 3 coins and 1 coin, respectively.
- since transactions do not explicitly say which input goes to which output, we assume that they are split evenly.
- so, address C contains ¾ of the coin from address A and ¼ of the coin from address B.

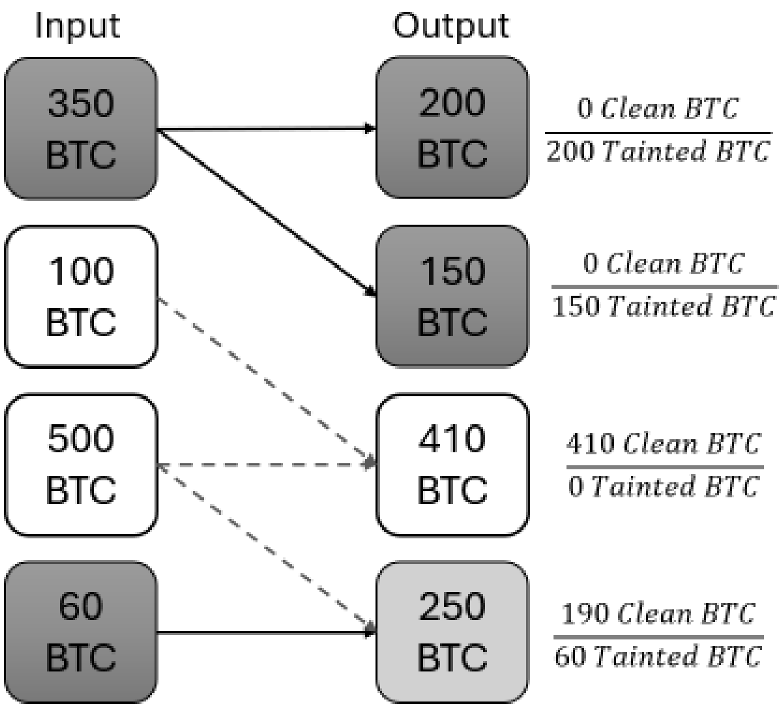
CRYPTO ANALYSIS: HAIRCUT

- shows how large is the percentage of a coin in one address from another address

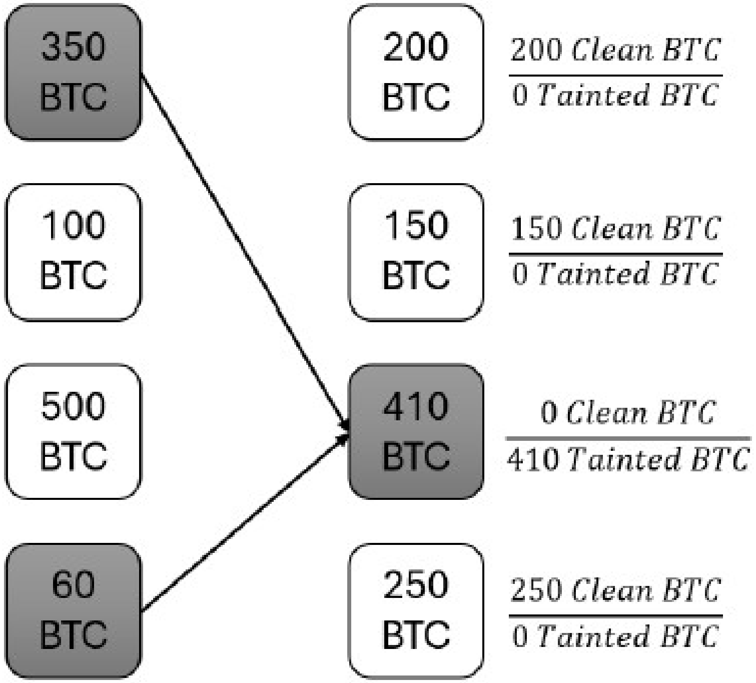


- now consider transaction T2, where address E combines its UTXO with address C.
- there are eight out of ten parts that come from address E, and two out of ten from address C. So
- Taint of A in F = 0.2*0.75=0.15
- Taint of B in F = 0.2*0.25=0.05

TAINT ANALYSIS: FIFO


Input
Output
350 BTC
100 BTC
500 BTC
60 BTC
200 BTC
150 BTC
410 BTC
250 BTC
$\frac{0\ Clean\ BTC}{200\ Tainted\ BTC}$
$\frac{0\ Clean\ BTC}{150\ Tainted\ BTC}$
$\frac{410\ Clean\ BTC}{0\ Tainted\ BTC}$
$\frac{190\ Clean\ BTC}{60\ Tainted\ BTC}$

TAINT ANALYSIS: TAINT IN HIGHEST OUTPUT (TIHO)


350 BTC
100 BTC
500 BTC
60 BTC
200 BTC
150 BTC
410 BTC
250 BTC
$\frac{200\ Clean\ BTC}{0\ Tainted\ BTC}$
$\frac{150\ Clean\ BTC}{0\ Tainted\ BTC}$
$\frac{0\ Clean\ BTC}{410\ Tainted\ BTC}$
$\frac{250\ Clean\ BTC}{0\ Tainted\ BTC}$