Bitcoin

The world's first decentralized digital currency

Meni Rosenfeld
Bitcoil

Bitcoin adoption (Jan 2013)

- Bitcoin "Market capitalization": $200M
- Users: 100K
- Bitcoin-accepting businesses: 2000, including
  - Wordpress.com
  - Freelancers, server hosting, software, books, clothing, video games, electronics, groceries, car accessories, ad networks, restaurants…
- Accepting donations: FSF, Wikileaks, Internet Archive, xkcd…
- Academic research: WIS (Adi Shamir), Microsoft, Cornell, ETH Zurich…
- Reports: FBI, ECB…

Bitcoin is a *currency*

- "Money can be exchanged for goods and services"
- Currency facilitates the trade of one good for another
- A good currency must be:
  - Scarce, portable, durable, fungible, divisible, current
  - Does *not* need to have "intrinsic" value
- The value of each unit of currency is determined by equilibrium between supply and demand
  - Total value of a currency is proportional to total trade using it
  - Value per unit = Total value / Number of units

Bitcoin is *digital*

- Ownership of bitcoins is digital information
- Typically used with a computer and the internet
- Based on cryptography

Bitcoin is *decentralized*

- There is no company "Bitcoin Ltd."
- There is no central issuer or controller
- Based on a public protocol
- Run by a p2p network of computers running FLOSS
- Multiple parties are each "doing their own thing"
  - Just like Linux!

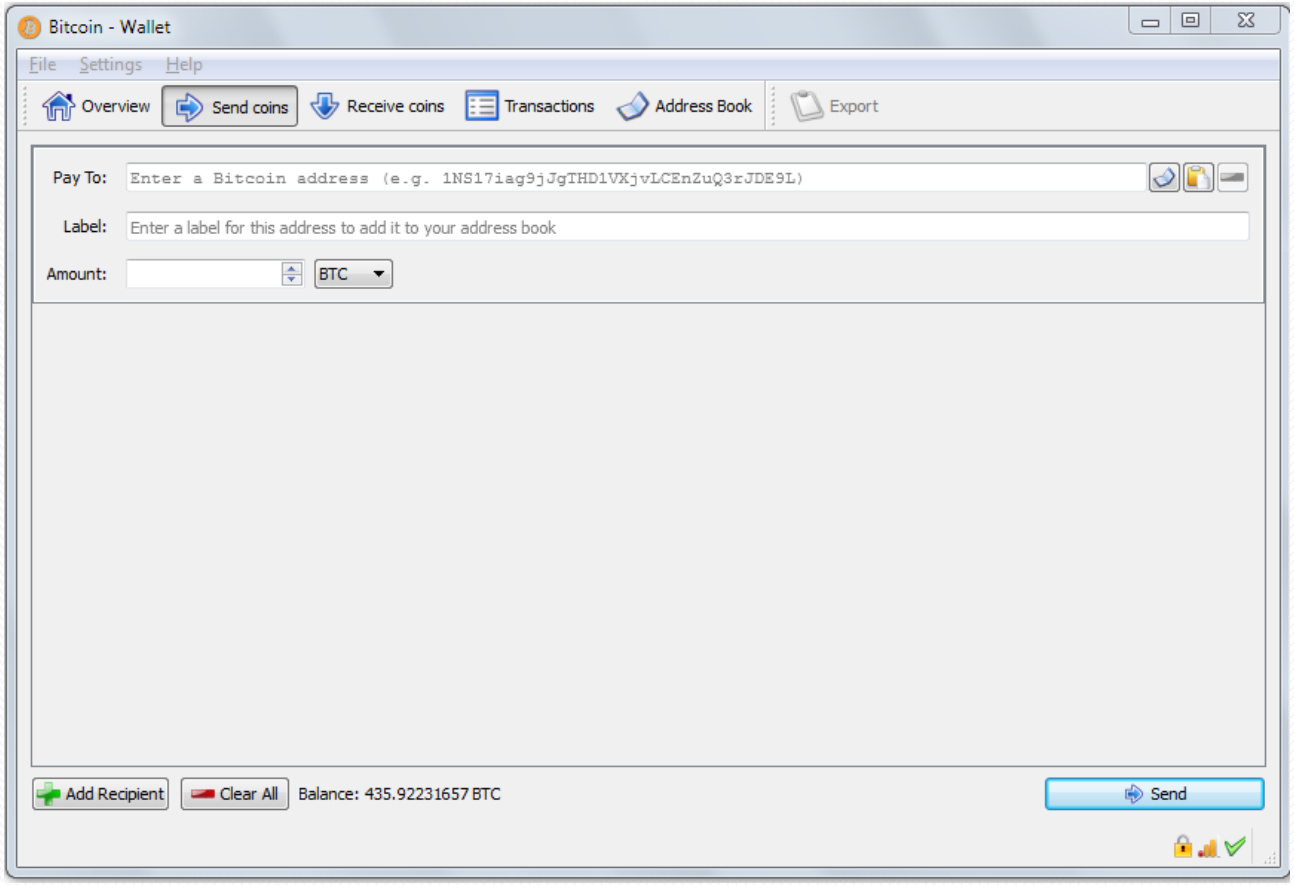
Bitcoin is *the first!*

- Plenty of physical currencies
  - Gold, silver, seashells, rocks…
- Plenty of centralized digital currencies
  - PayPal, WebMoney, e-gold, DigiCash, LR, WoW gold, SLL, EVE isk…
- Bitcoin is the world's first decentralized digital currency
- Invented in 2008 by "Satoshi Nakamoto" (pseudonym)

How to use?

- Install open-source client software
- Software generates “addresses”, which are like bank account numbers (e.g. 1BBsbEq8Q29JpQr4jygjPof7F7uphqyUCQ)
- To receive bitcoins, let the sender know your address
- To send bitcoins, specify receiving address and amount, and click “send”

How to use?


Bitcoin - Wallet
File
Settings
Help
Overview
Send coins
Receive coins
Transactions
Address Book
Export
Pay To:
Enter a Bitcoin address (e.g. 1NS17iag9jJgTHD1VXjvLCEnZuQ3rJDE9L)
Label:
Enter a label for this address to add it to your address book
Amount:
BTC
Add Recipient
Clear All
Balance: 435.92231657 BTC
Send

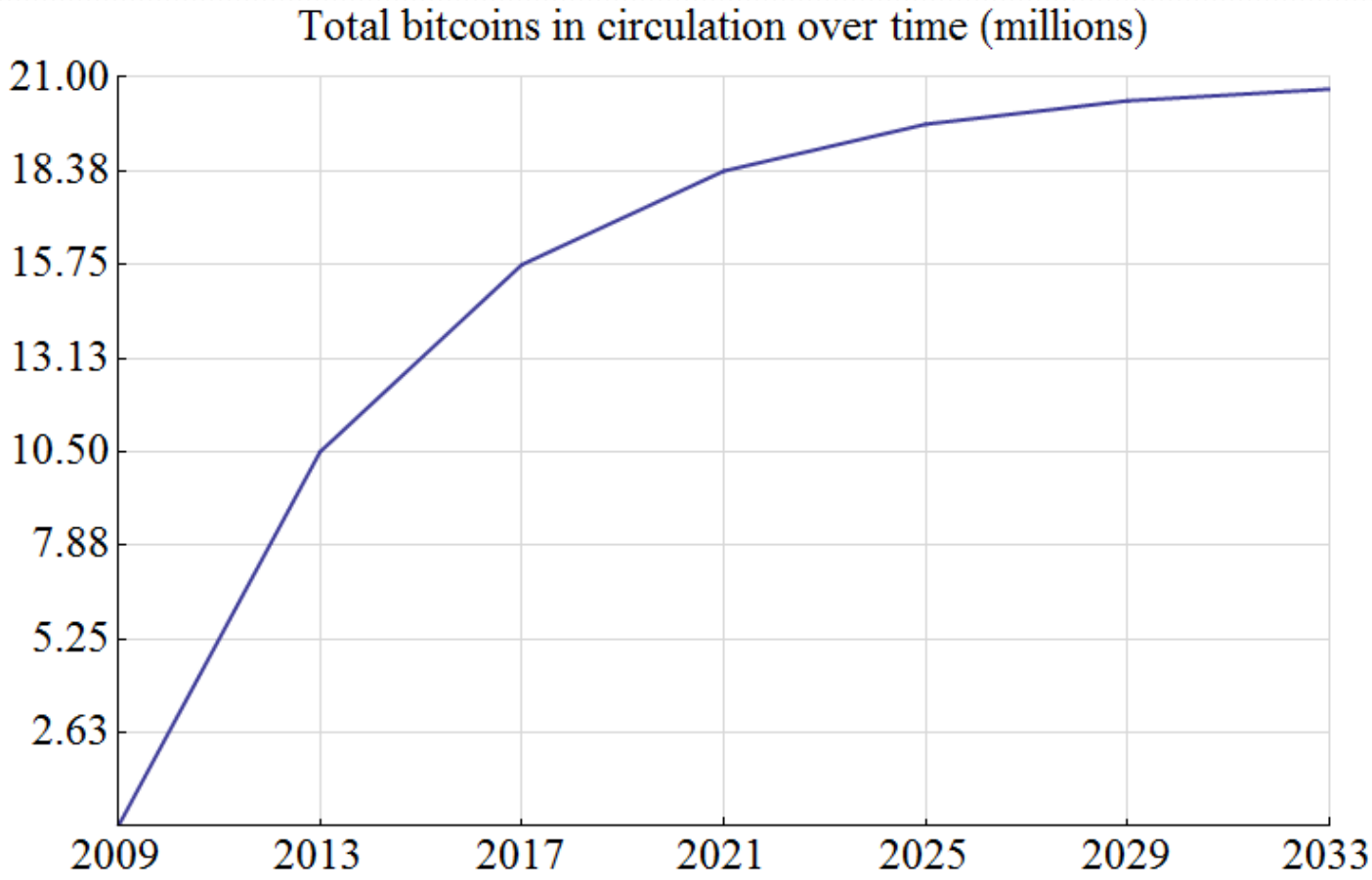
Why?

- No need for 3rd party
- Easy to send and receive money
- Almost no fees
- No single point of failure
- Secure
- Limited supply – no built-in long-term monetary and price inflation
- No chargebacks
- International
- Usable by weak/small countries
- Pseudonymous
- Public ledger
- Advanced applications

Quantitative data

- No more than 21 million bitcoins will ever exist
- So far about 11 million bitcoins have been created
- Each bitcoin is currently worth roughly $20
  - Started at roughly half a cent, all-time high $32
- Bitcoin amounts can be specified with 8 decimal places
  - 2.1 quadrillion atomic units
- Monetary inflation rate is stepwise decaying exponential
  - Creation rate is cut in half roughly every 4 years

Inflation schedule



Historic price chart

Analogies

- Bitcoin is to money what…
  - Email is to communication
  - The WWW is to publishing
  - Social networks are to socializing
- Bitcoin is an open source currency
  - You can look under the hood
  - You can hack it (but you can't crack it)
- Bitcoin is a startup currency

The Bitcoin Blockchain

How does Bitcoin work?

Meni Rosenfeld
Bitcoil

Public key cryptography

- Every user has a private key and a public key (numbers)
  - Everyone knows user's public key
  - Private key is the user's secret, never shared with anyone
- Public key is uniquely determined by the private key
- Virtually impossible to compute private key from public key
- Can be used for encryption and digital signatures

Digital signatures

- User wants to send a message and prove that he wrote it
- Takes message and private key and performs a computation to create a signature
- Recipient compares the signature against the message and the user's known public key
- Only the user who possesses the private key can sign messages, does not need to share the private key
- Examples: RSA, ECDSA

Hash functions

- Example: SHA-256

- Takes arbitrary data and transforms it to a 256-bit number
  - Integer from 0 to 115792089237316195423570985008687907853269984665640564039457584007913129639935
  - Usually expressed as hexadecimal string
  - IG46Us2X7EKc4Cn3 => `6fe47cd49392e511dac5ef335aaf3b...`
  - IG46Us2X7EKc4Cn4 => `3a9ee39ea060e2f94d5f9e1346430a...`
  - Even the tiniest change can alter the hash in ways you can't imagine

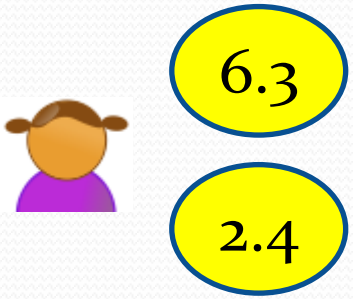
- The hash of random data is essentially a random number
  - If highest possible hash is M, has probability X/M to be less than X

Bitcoin system components

- A transaction structure for specifying and changing ownership
- A p2p network for propagating, verifying and storing transaction data
- A proof-of-work system (hashing, “mining”) for:
  - Synchronizing transactions
  - Determining initial distribution of coins

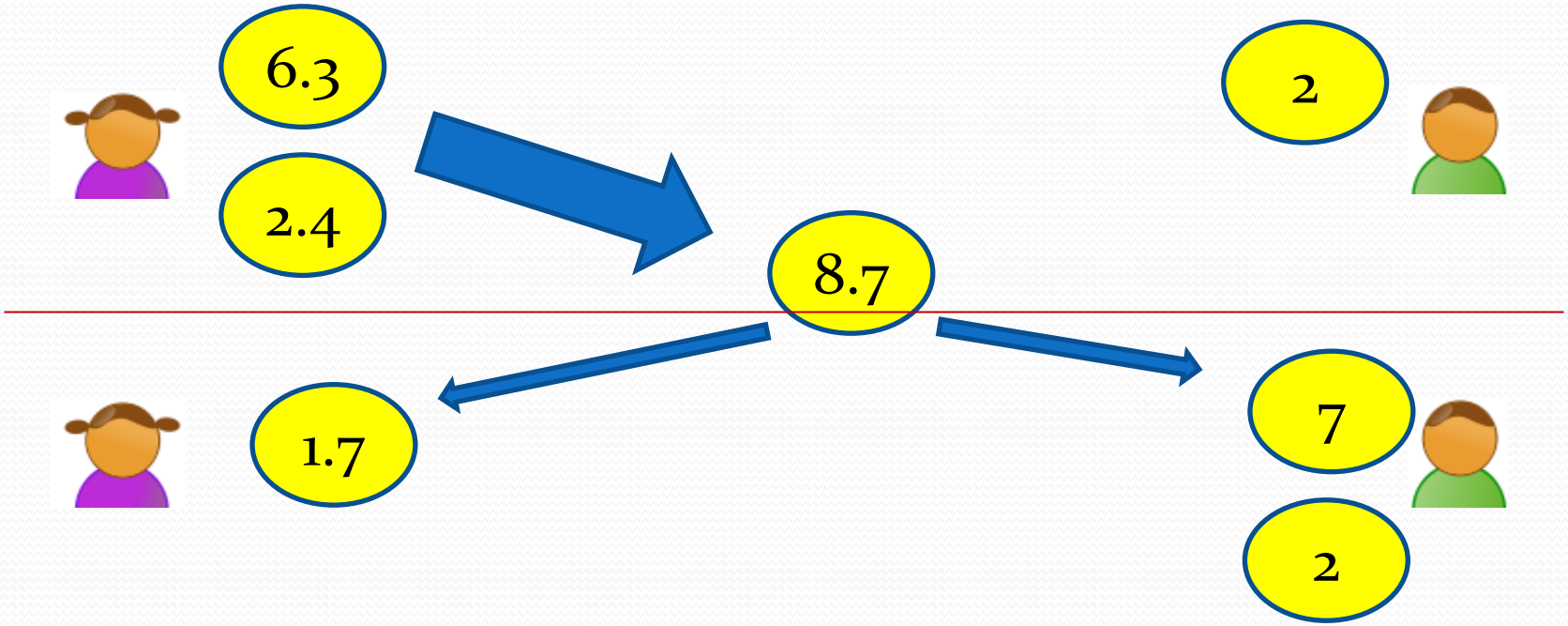
Coins

- The fundamental building block of Bitcoin is a “coin”
- A coin is characterized by:
  - Unique ID
  - Quantity (denomination) – arbitrary number with 8 decimal places
  - Owner



Coins

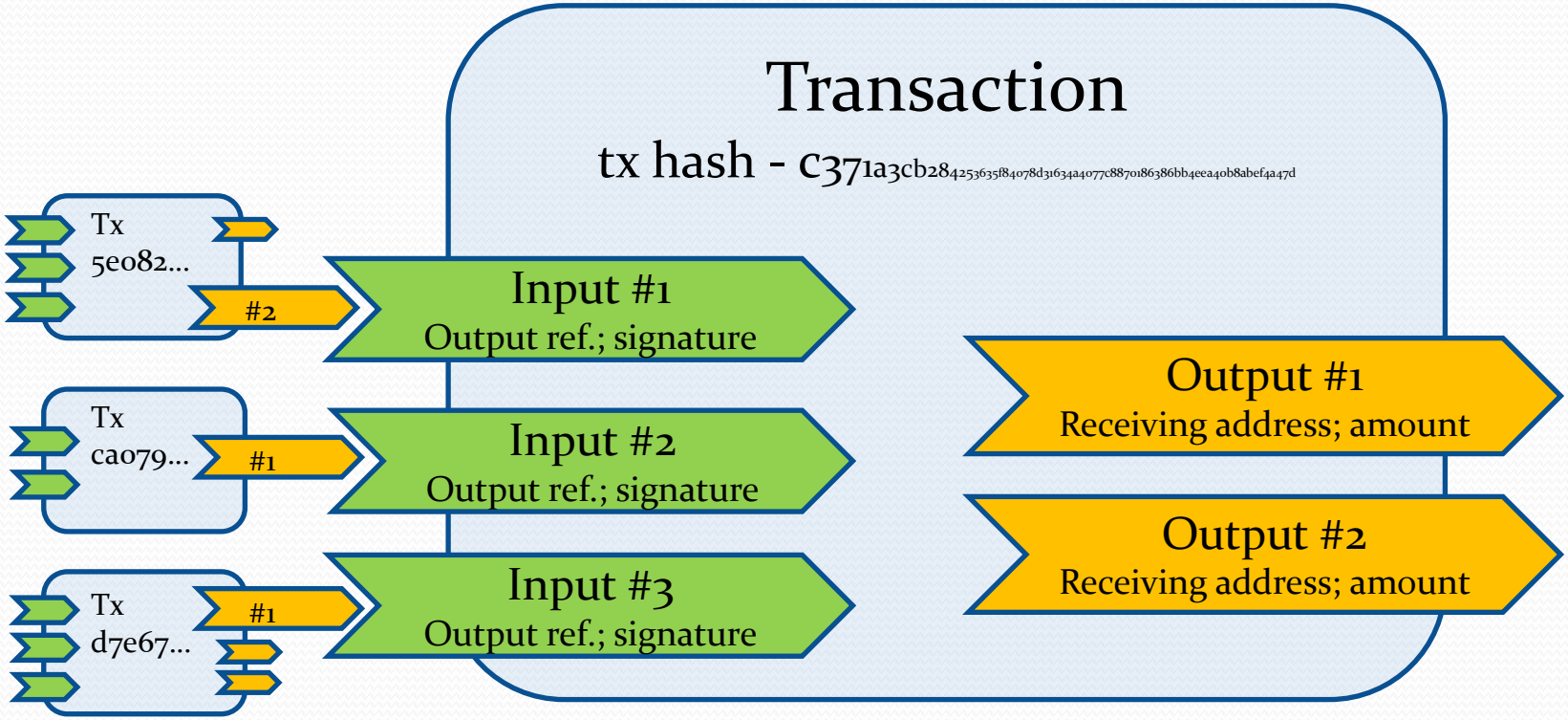
- Coins can be split and merged
- If Alice wants to send bitcoins to Bob, she will merge some of her coins and split the result between her and Bob



Transactions

- The owner of a coin is identified by an “address”
- Each address is associated with a private key
- To use a coin, the owner must provide a digital signature with the associated private key (ECDSA)
- The process where coins are merged and split is called a “transaction”
  - Used to move bitcoins from one owner to another

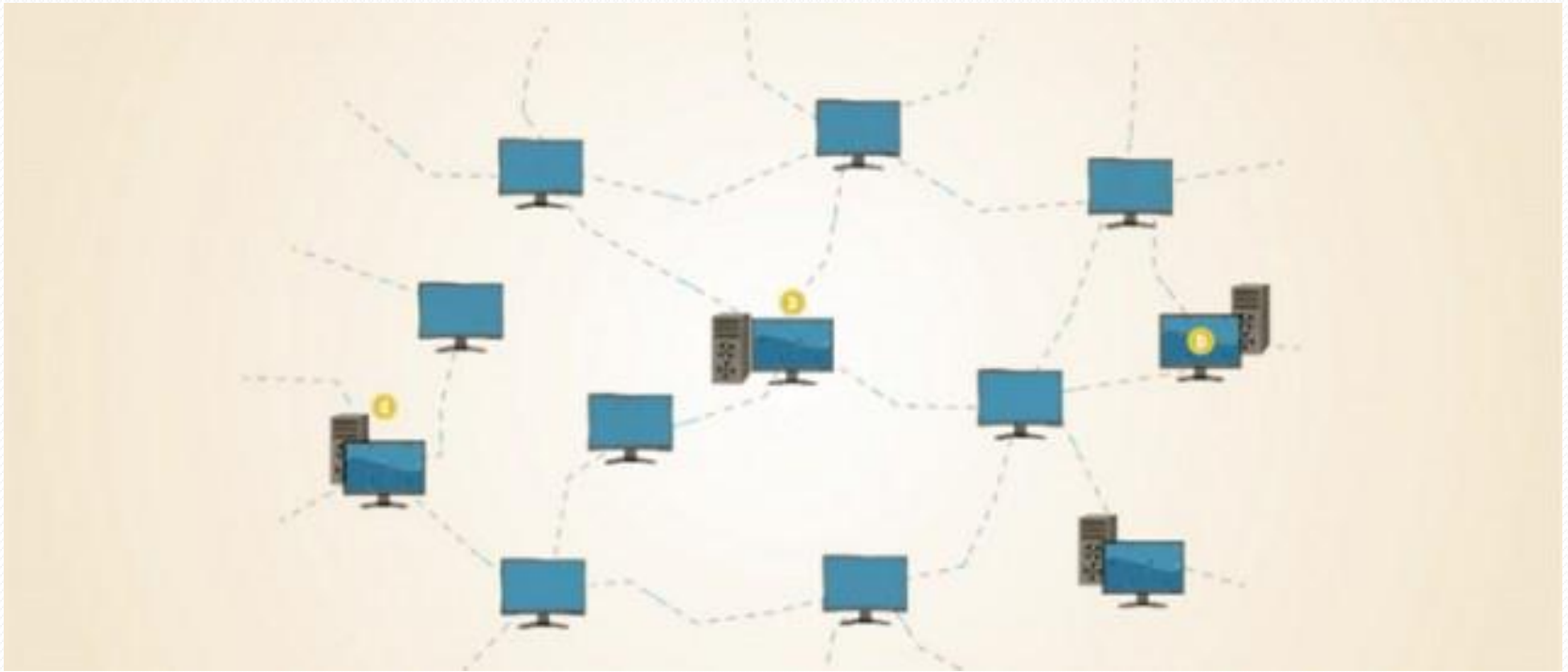
Transaction structure



Transaction structure

- A transaction can have any number of inputs and outputs
- An output specifies a receiving address and amount
- An input references a previous unspent output
- The total value of all inputs must be at least the total value of all outputs
- The transaction is identified by a hash of its data
- The hash must be signed by the private key corresponding to every input address
- An address is a hash of an ECDSA public key
- More generally, an output specifies a script with the conditions to allow spending it

The Network



Problem: Double spending

- Using the same output ("coin") to pay 2 different recipients
  - No agreement on who is the "true" recipient
  - One recipient will be out of his coins (presumably after providing some product)
  - Some way to determine order of transactions is needed
- Traditional solution: Central authority
- Naïve decentralized solutions have vulnerabilities
- The first working decentralized solution is the blockchain

Tentative solution

- Suppose there was just one coin
- Two conflicting transactions:

- Only one transaction will be accepted
- Doesn't matter which one
  - As long as everyone agrees and it won't change

Tentative solution

- Each computer in the network:
  - Chooses the transaction it thinks is correct
  - Takes the transaction hash and concatenates random data
  - Computes the hash of the result
  - If hash is less than M/D, publish the result (probability 1/D)
  - (tx hash, `c5145e94) => 0000bbe9affcf9f93b635...`
  - Repeat

- Each published result is a confirmation for the transaction
  - n confirmations prove that on average nD hashes have been computed – by nodes agreeing with this transaction

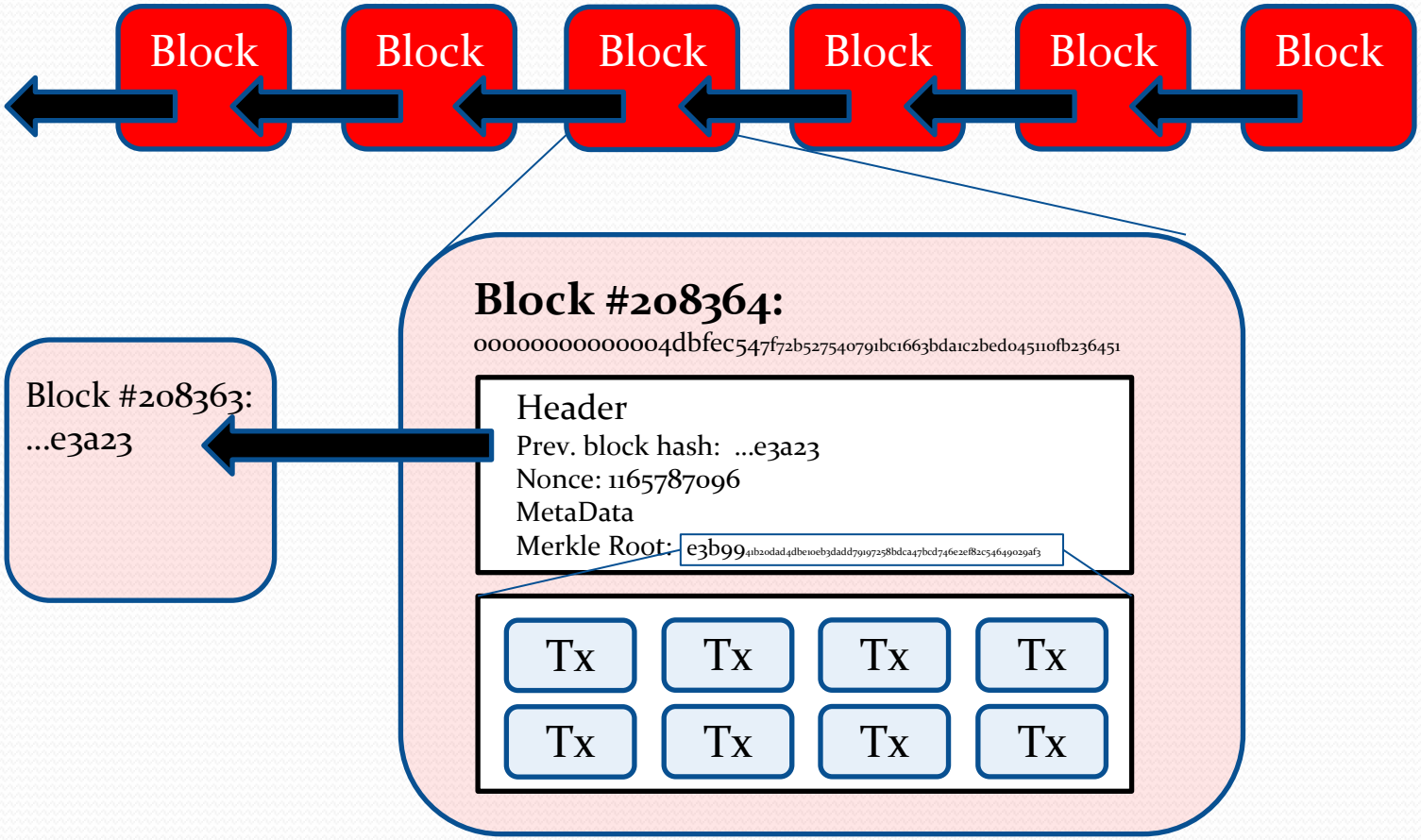
Tentative solution

- The transaction with more confirmations is considered valid
- A more widely accepted tx will get new confirmations faster
- Eventually all nodes will converge on one of the transactions
  - And continue adding more confirmations
- To switch to the other transaction, Mallory needs to compute hashes faster than everyone else combined

Solution: The blockchain

- Transactions are grouped into blocks
- Blocks are confirmed with proof of work
- A transaction is considered final if it is included in a confirmed block
- Each block references a previous block to form a chain
- In case of conflict, the transaction with more compute power spent on confirmation wins
- Attacks require having more compute power than the rest of the network

The Blockchain



Block structure

- Transactions are organized in a Merkle tree with a resulting root hash
- The block header consists of the Merkle root, the hash of the previous block, other metadata, and a nonce
- The block is identified by the SHA-256 hash of its header
- A block is valid only if its hash is lower than the target

Proof of work

- A block with given data and nonce has a very low probability of being valid
- Miners try different nonces and compute the resulting hash (billions of tries per second) until they match the target, and release the resulting block
- The existence of a block which includes a transaction proves that computational work has been done by a node which considers this transaction valid
- Each block references the previous one. Each transaction gets increasingly more powerful proof of work
- In case of competing branches, the one with the most proof of work is selected

Proof of work

- A transaction “buried” under several blocks is very hard to revert mistakenly or maliciously
- Reverting a transaction requires catching up with the computation of the honest network, which is unlikely without greater hashrate
- Any change to a transaction invalidates all proof of work
- Hash target is adjusted every 2016 blocks (roughly 2 weeks) so that on average one block is found every 10 minutes

Creation of coins

- Every block is allowed one special “generation transaction”
- A generation transaction has a single special input, and any number of outputs
- Value of input: New coins + tx fees
- New coins: $50 \cdot 2^{-\lfloor H/210000 \rfloor}$ (starts at 50 BTC per block and halves roughly once every 4 years)
- Incentivizes securing the network by hashing
- Robust way to determine initial distribution

Questions?

Thank you

- Meni Rosenfeld
- meni@bitcoil.co.il
- https://bitcoil.co.il
- 1DdrvajpK221W9dTz05cLoxMnaxu859QN6