# When Peer-to-Peer Meets Money:
## an introduction to Bitcoin

Tom Van Cutsem


Vrije
Universiteit
Brussel

# Disclaimer

- I'm a computer scientist
- I'm not a cryptographer
- I'm not an economist
- I own bitcoins

# What is Bitcoin?

- A virtual currency
- Technically, a **cryptocurrency**
- Digital “money”
- Digital gold

# Talk outline

- What makes Bitcoin unique?
- Why Bitcoin?
- How are Bitcoins created?
- Under the Hood
- Bitcoin in practice

# Virtual currencies are not new

- Linden Dollars (Second Life), WoW Gold (World of Warcraft), Interstellar Kredits (EVE Online), ...
- Facebook Credits
- cryptographic “e-cash” systems since the ‘80s
- ...

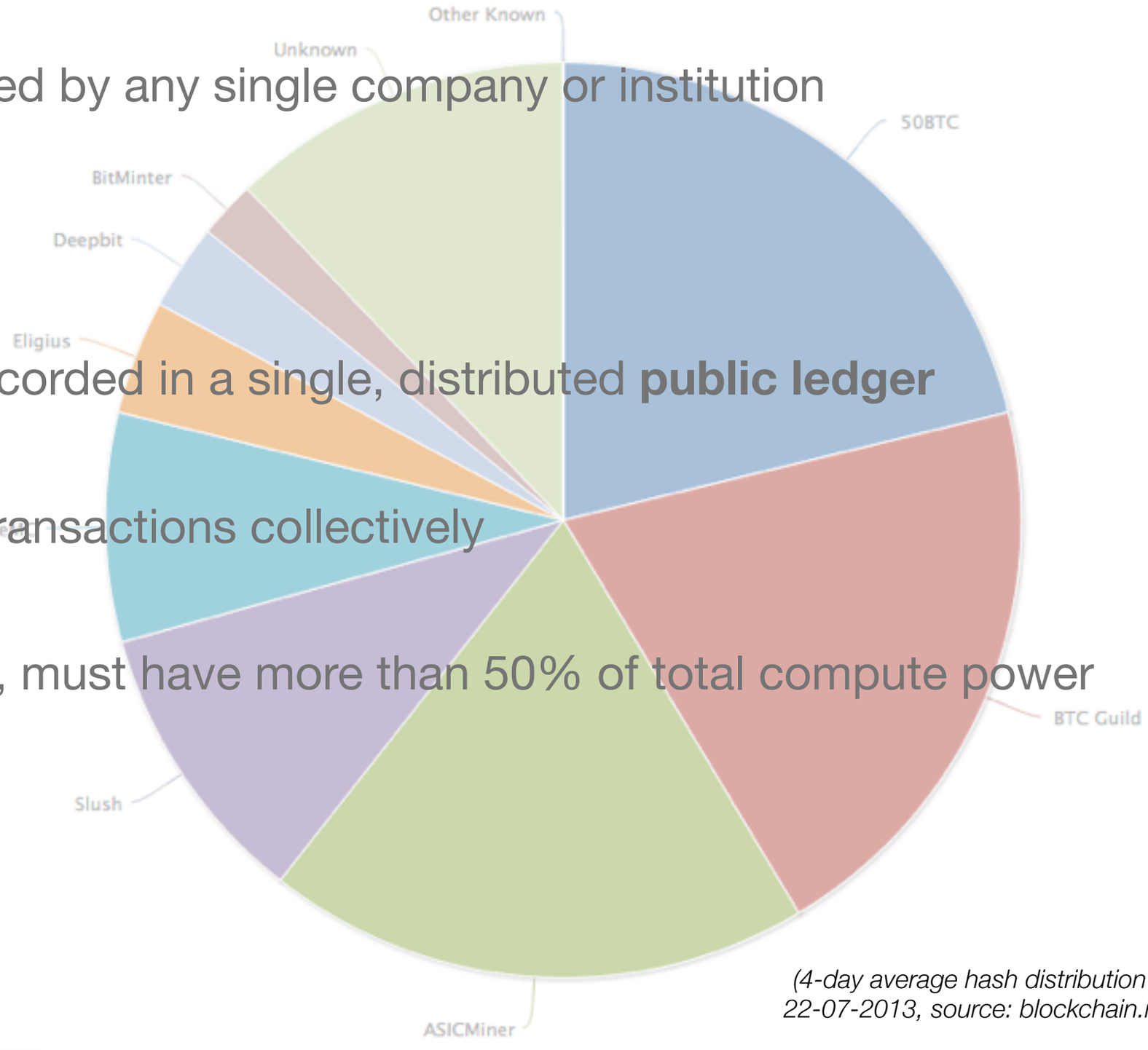
# What makes Bitcoin unique?

# Fiat money and most virtual currencies are centralized

- Fiat money:
  - Central Bank
    - Central source of supply (“the mint”)
    - Central “clearing house”
      - Makes it easy to verify double spending
- Single point of trust

# Bitcoin is decentralized

- Not issued or controlled by any single company or institution
- “Peer-to-peer”
- All transactions are recorded in a single, distributed **public ledger**
- The network verifies transactions collectively
- To attack the network, must have more than 50% of total compute power




*(4-day average hash distribution on 22-07-2013, source: blockchain.info)*

# BTC Strengths

- **No central point of trust**, no central authority
- Transactions are typically carried out within **minutes** (compare to banking transactions taking days)
- Transaction **fees are low** to non-existent. This makes BTC suitable for **micro-transactions**.
- Transactions are **irreversible**
- **Limited supply**: controlled inflation
- **No borders**: works the same across the planet

# BTC Weaknesses

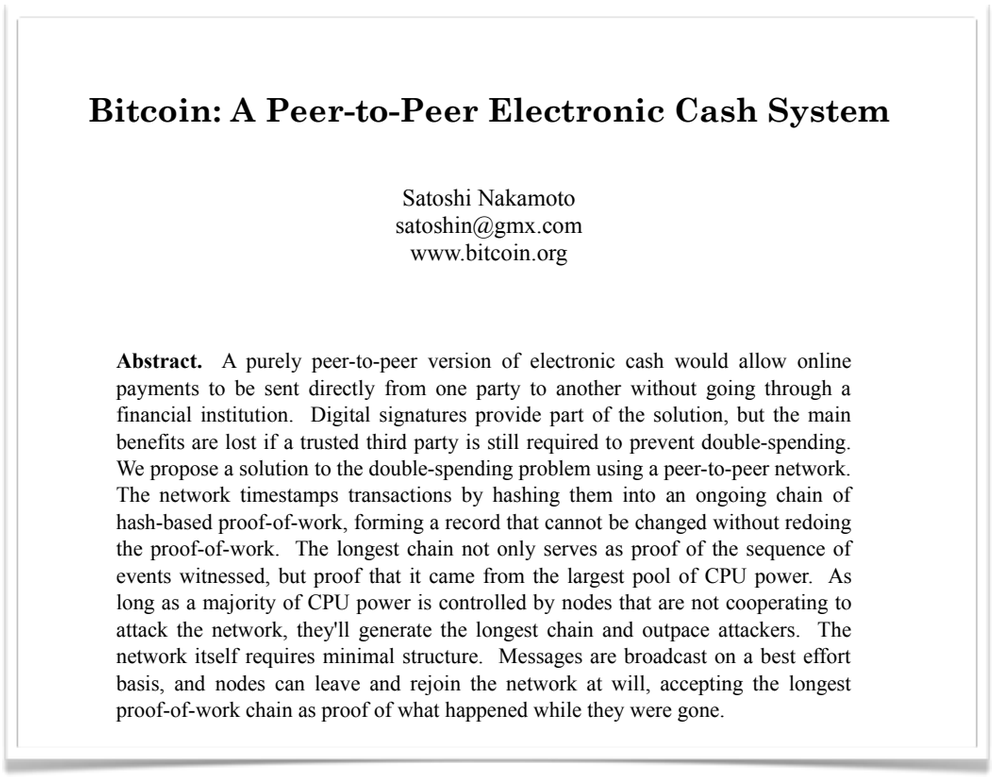
- Technical Risks:
  - **Relies on crypto**graphic algorithms not being broken
  - If a **single party controls > 50%** of compute power in the network, it can steal back its own spent coins
    - > 50% control does not allow attacker to generate Bitcoins out of thin air, or to stop or revert other people's transactions

# BTC Weaknesses

- Non-technical Risks:
  - Not sure **who invented it**. Have to place trust in the network.
  - **End-user** is **responsible for safe-keeping** of his/her coins
    - blockchain.info slogan: “be your own bank”
      - Reliance on online wallets reintroduces third-party risk
    - **Lack of** a **legal framework** (taxation, ...)
    - Governments cannot manipulate the currency, but can **coerce companies** that serve as entry-point into the Bitcoin economy

# Who is behind Bitcoin?

- 2008 white paper by “Satoshi Nakamoto”
- Doubtful this person really exists
- Today, Bitcoin codebase maintained as an open source project on GitHub



**Bitcoin: A Peer-to-Peer Electronic Cash System**

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

**Abstract.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

# Why Bitcoin?

# Why Bitcoin's increasing popularity is timely

- Growing distrust with Governments' monetary policy
  - e.g. FED's Quantitative Easing policy
- EUR, USD, etc. are "floating" currencies, not redeemable for any commodity
- Growing financial unrest, capital controls
  - E.g. Cyprus bail-in

**FIGURE 1 – 2013 USD/BTC EXCHANGE RATE**


*(source: The Genesis Block, 2013 Mid-year review)*

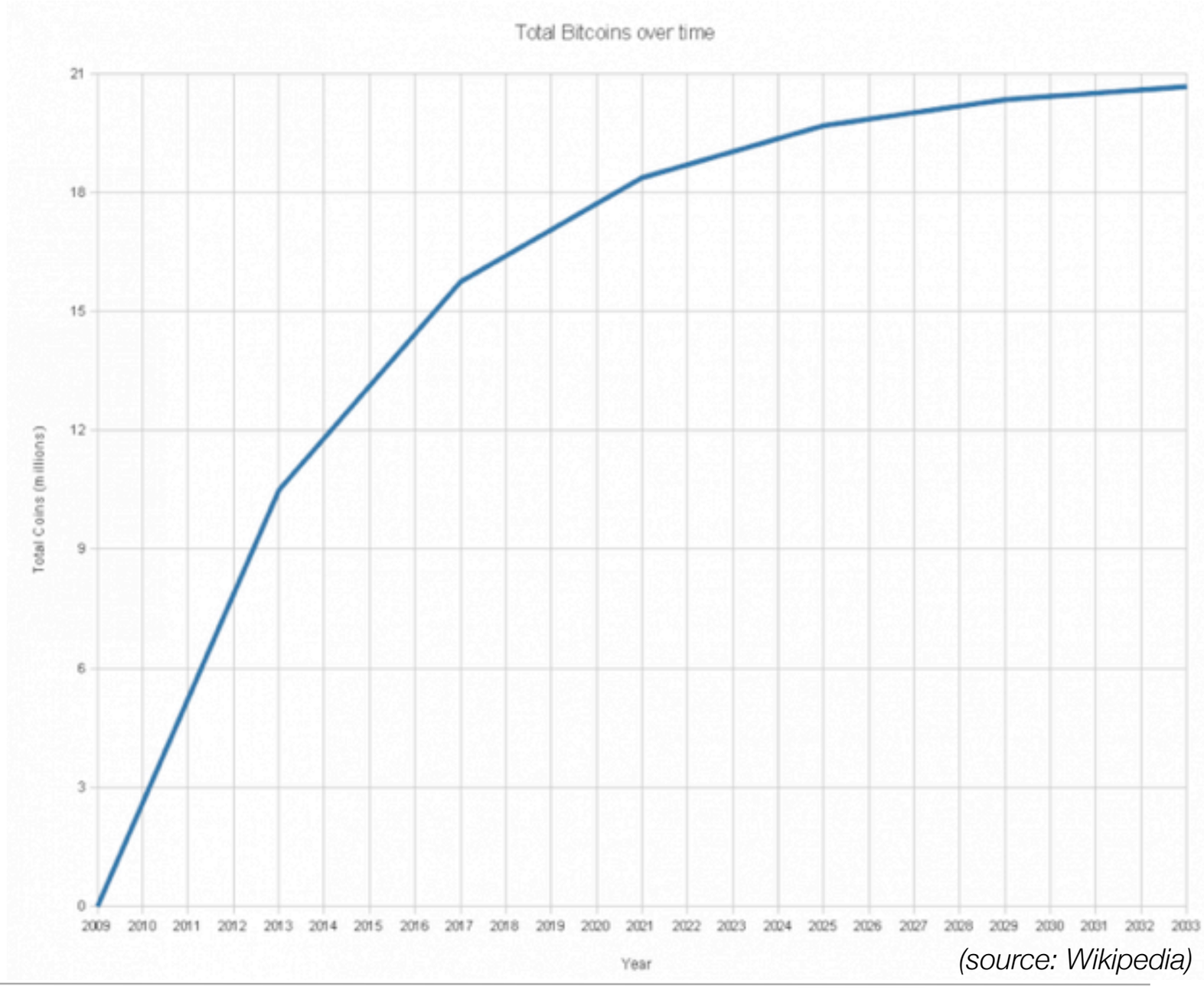
# How are Bitcoins created?

# Bitcoin mining

- Computers that aid in processing transactions get a “reward”
  - **Incentive** to become part of the network and help transaction processing
  - Analogy with mining gold.
- New bitcoins born by solving a cryptographic puzzle
- Limit on inflation: the “reward” is halved every 4 years

# Bitcoin mining: pre-determined issuance schedule

- Asymptotic limit of 21 Million bitcoins (to be hit around 2140)
- Approx. 12 Million mined so far

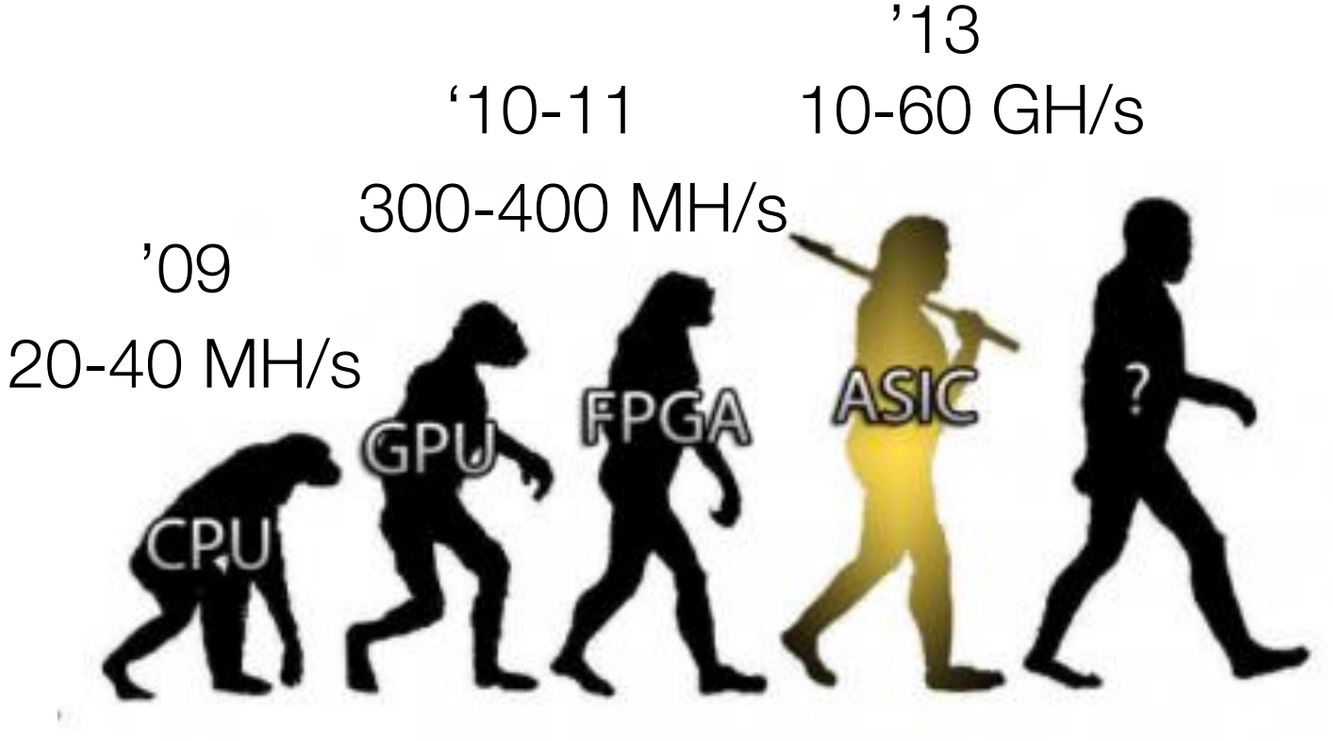



*(source: Wikipedia)*

# Mining Rigs


# From CPU to ASIC Mining

- Mining speed measured in (SHA-256) “hashes per second”

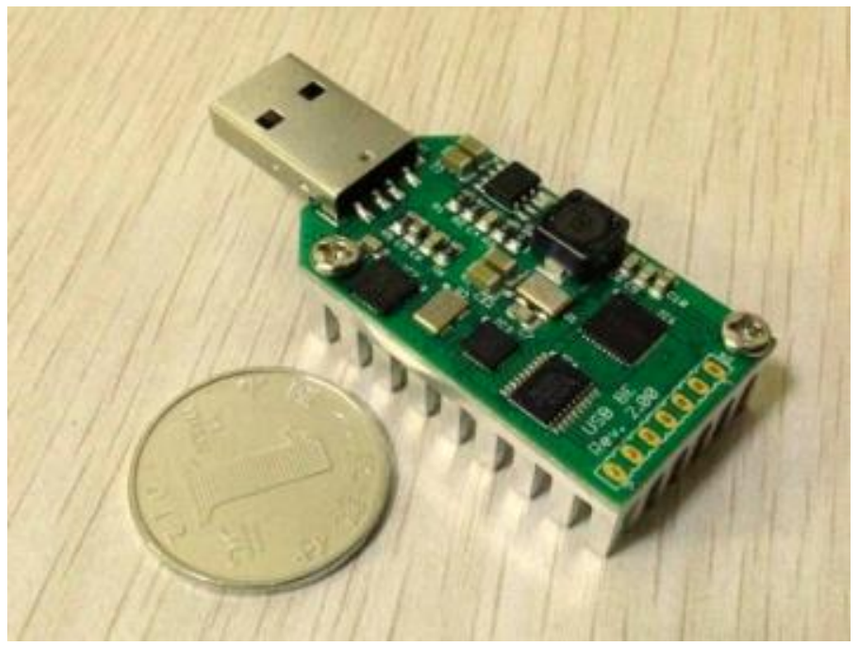



*(source: The Genesis Block, 2013 Mid-year review)*

*ASIC = Application-Specific Integrated Circuit*

# ASIC Miners

- Companies exist that sell dedicated chips (ASICMiner, Butterfly Labs, Avalon)
- E.g. USB miner achieving 330 MH/s
- Cost: 0.89 BTC



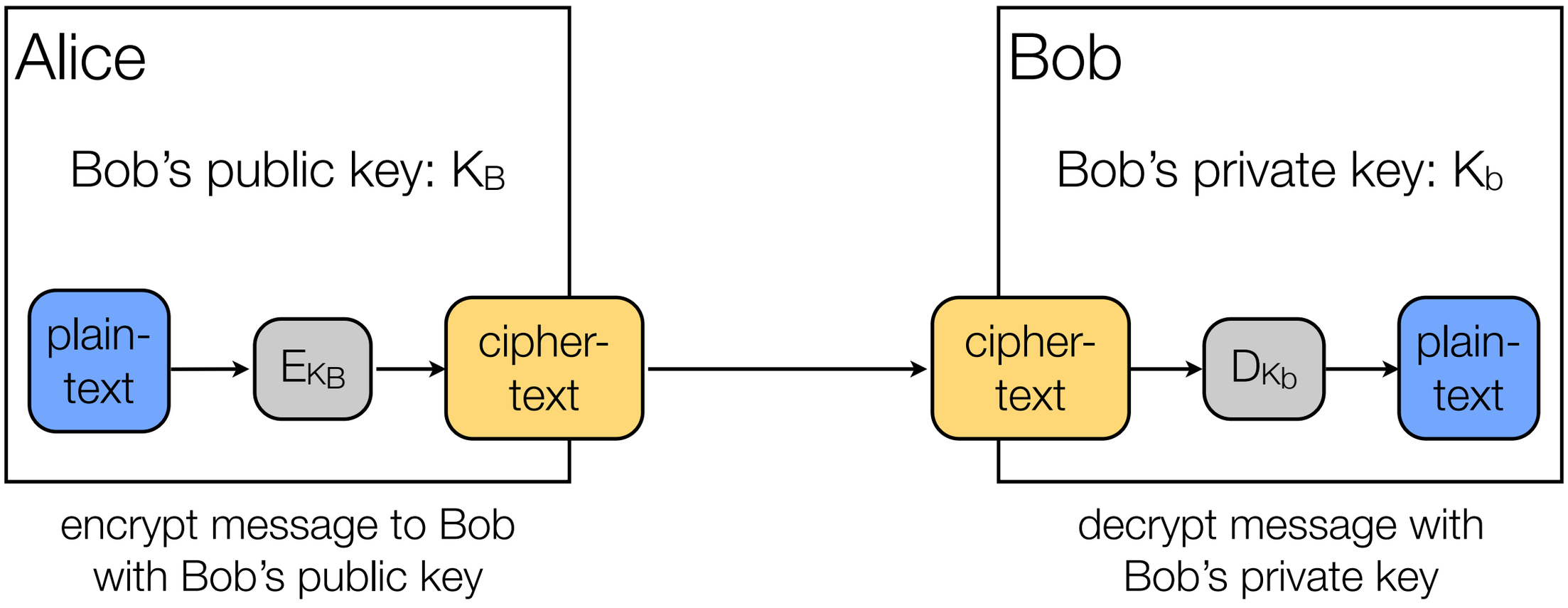
# Under The Hood

# Bitcoin addresses

- To create a bitcoin address, generate a new public/private key pair
- (hash of) public key serves as an “address” or “account number”
- Access to public key allows you to query the account balance
- Access to private key allows you to spend

# Public Key Crypto 101: Communication

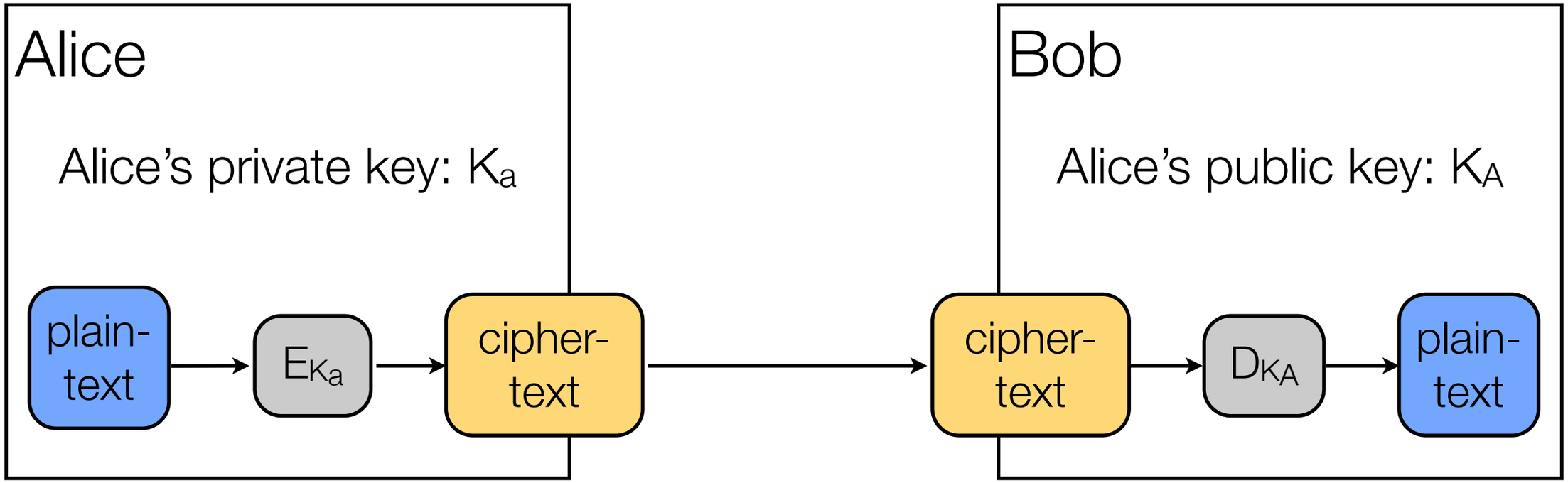
- When Alice wants to send a confidential (encrypted) message to Bob:



# Public Key Crypto 101: Digital Signatures

- Encrypting a message with a private key is the same as signing it!

- If Bob can decrypt the message with $K_A$, he knows it could only have been encrypted with $K_a$, i.e. that it was sent by Alice

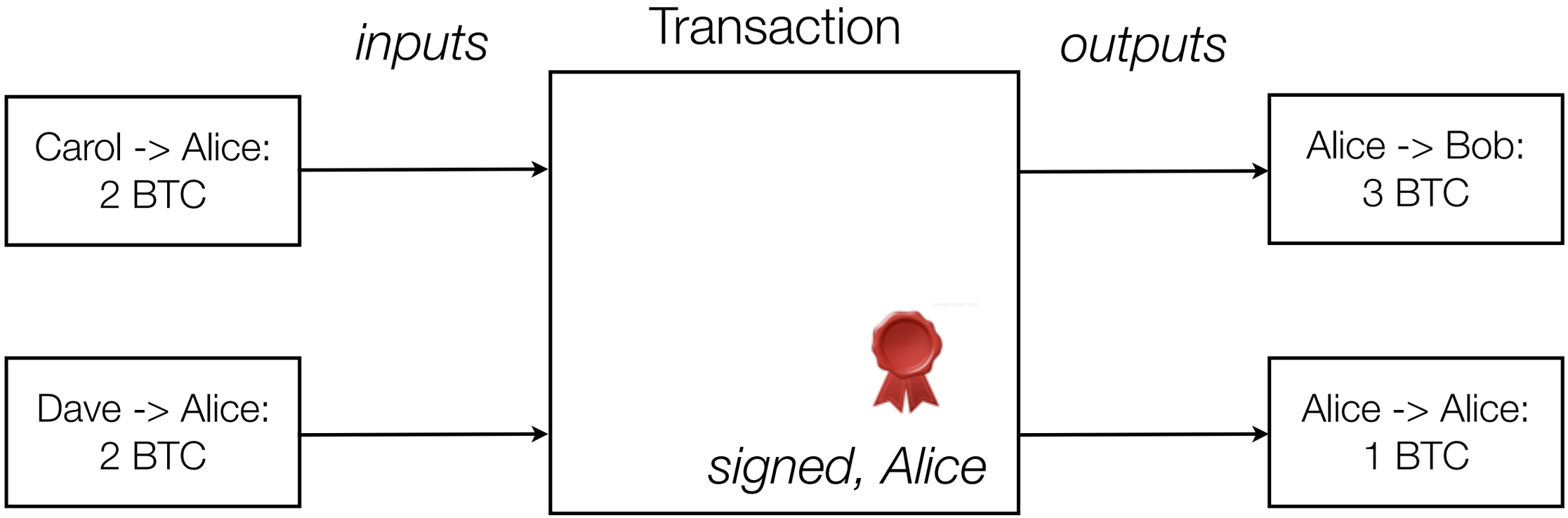


# Where are Bitcoins stored?

- Your coins “reside” implicitly in prior transactions that designate your public key as a beneficiary

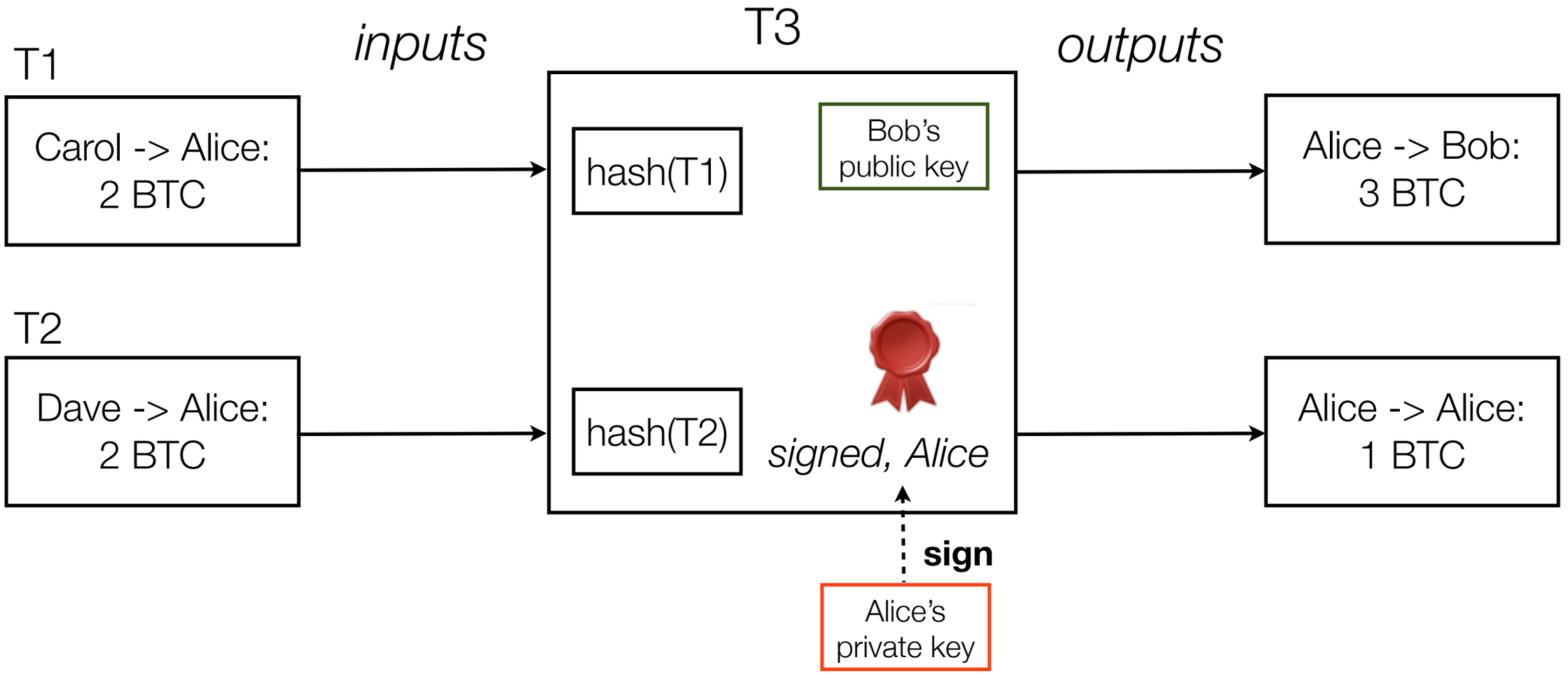
# Example

- Alice wants to pay Bob 3 BTC
- She “owns” 4 BTC by proving that she previously received 2 BTC from Carol and 2 BTC from Dave



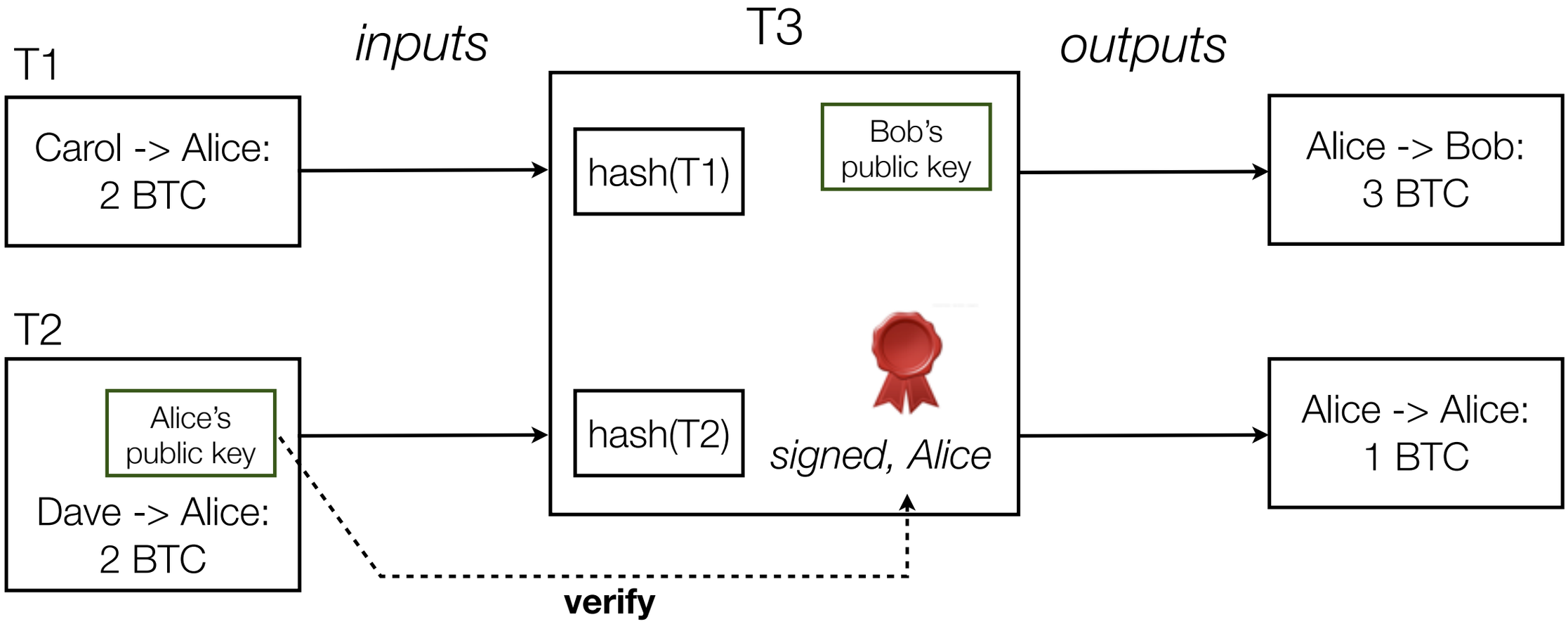
# Example

- To transfer ownership, Alice includes in T3 the hashes of input transactions and the public key of the next owner
- Alice digitally signs the transaction



# Example

- Bob (or anyone else) can verify T3 by verifying Alice's signature, based on the public key found in the input transactions T1 and T2

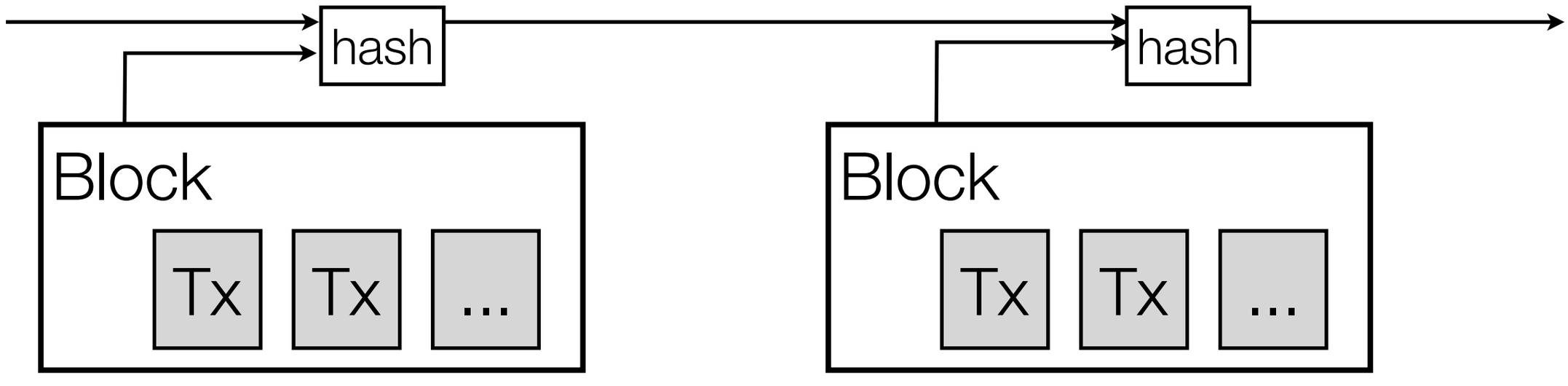


# The double spending problem

- How does Bob know the received coin has not been spent before?
- Bob must be able to check that previous owners did not sign any earlier transactions.
- Solution: make all transactions **public** so that everyone can verify what transactions happened first and detect double spending.
- All participants must **agree on a single history** of the order in which transactions were made
  - This is a hard problem in distributed systems, also known as **consensus**!

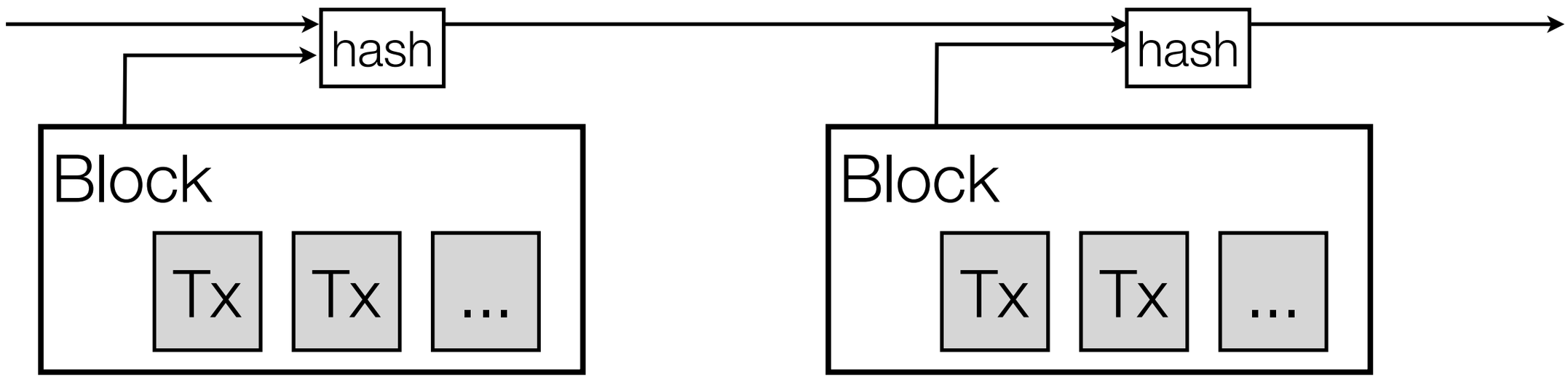
# Solution: timestamp server

- Timestamp server hashes a block of transactions to be timestamped and widely publishes the hash
- The timestamp proves that the data must have existed at the time, in order to get into the hash
- Each timestamp includes the previous timestamp in its hash, forming a chain
- This is called the **blockchain**



# The Blockchain

- The blockchain is Bitcoin's **transaction ledger**, publicly recording *all* transactions
- Benefit of chaining: changing a single block would require changing all blocks after it as well



# Distributing the timestamp server: proof-of-work

- Problem: if anyone can easily produce a valid block, there is little hope that the network will end up working on a *single* blockchain
- More likely, would end up with a quickly growing *tree* of blocks
- Solution: use “proof-of-work”
  - Make it really hard to produce a valid block (as in: need a lot of compute time)
  - Once a valid block is found, it is trivial prove that it is indeed valid
  - The generated block is its own proof of the work invested to generate it

# Distributing the timestamp server: proof-of-work

- The proof-of-work involves scanning for a value *v* such that hash(*v*) begins with a number of zero bits *n*.
- Average work required is $O(2^n)$
- Done by incrementing a number in the block until a value is found that gives the block's hash the required zero bits.
- Difficulty is adjusted dynamically such that on average, only one block is generated every 10 minutes

# Proof-of-work: example

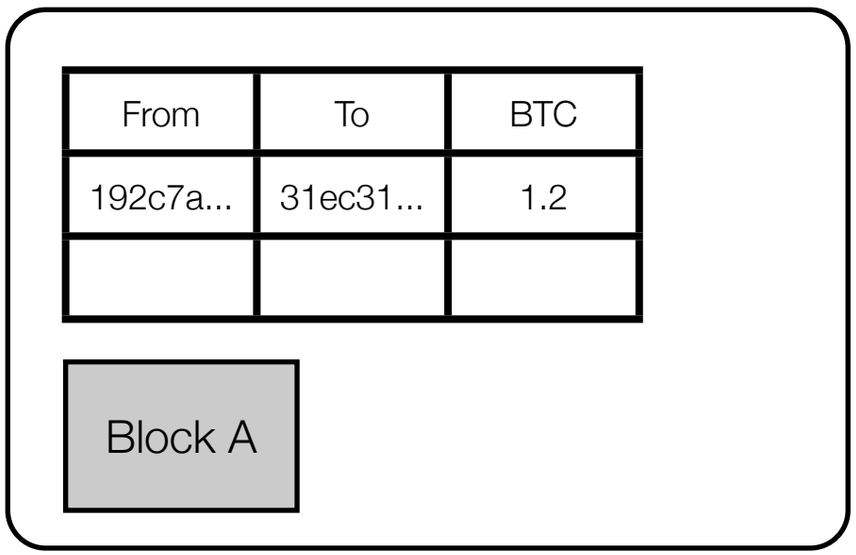
- Transaction in block: transfer 10 BTC from address a1 to address a2.
- Target difficulty: at least 3 zeroes.
- hash(“a1->a2:10_0”) = 1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64
- hash(“a1->a2:10_1”) = e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
- hash(“a1->a2:10_2”) = ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7
- ...
- hash(“a1->a2:10_**5142**”) = **0000**c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9

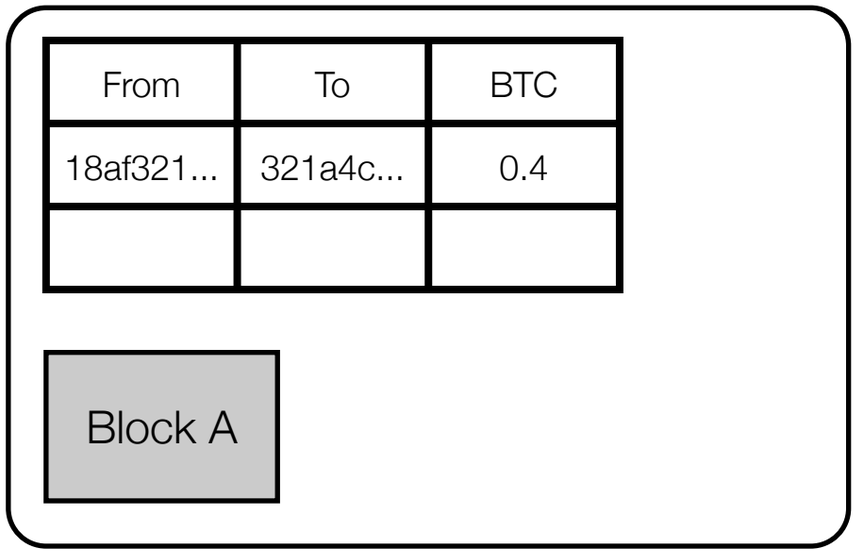
# Proof-of-work

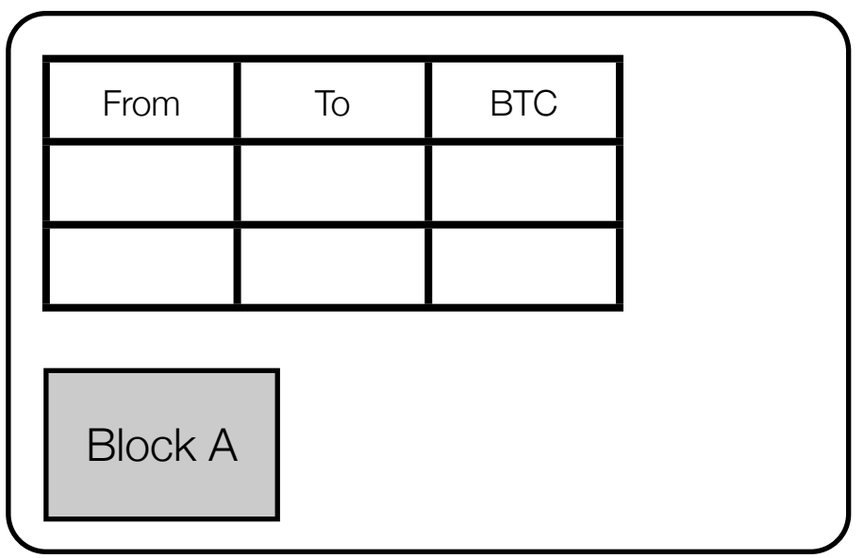
- Proof-of-work solves the problem of deciding the *majority vote*
  - One IP address one vote? Problem: attacker may issue multiple IPs
  - Bitcoin: roughly “one CPU, one vote”:
- The majority decision is represented by the **longest chain**, which has the greatest proof-of-work effort invested in it.
- If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.

# The Bitcoin network

1. New transactions are broadcast to all nodes.

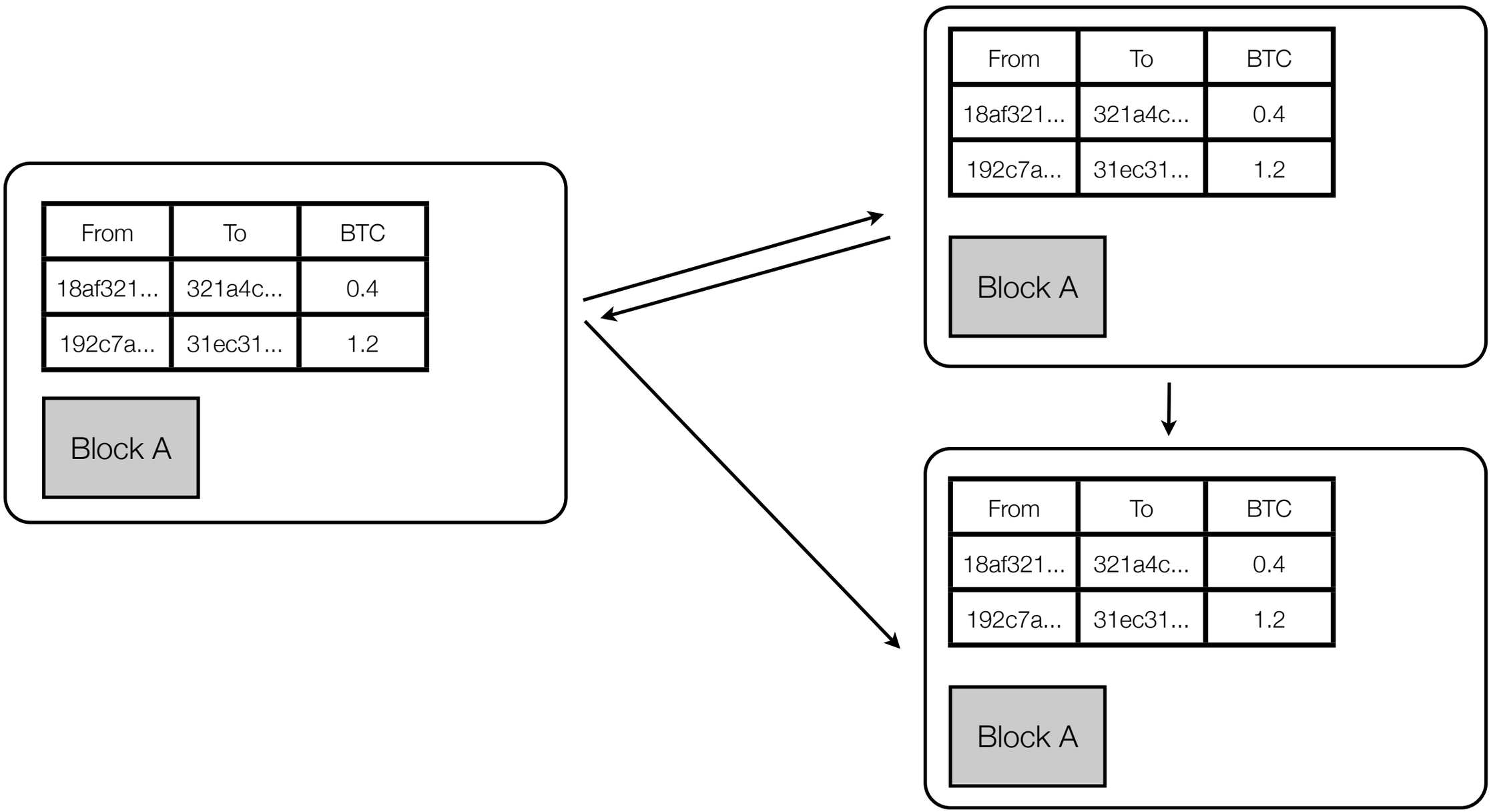


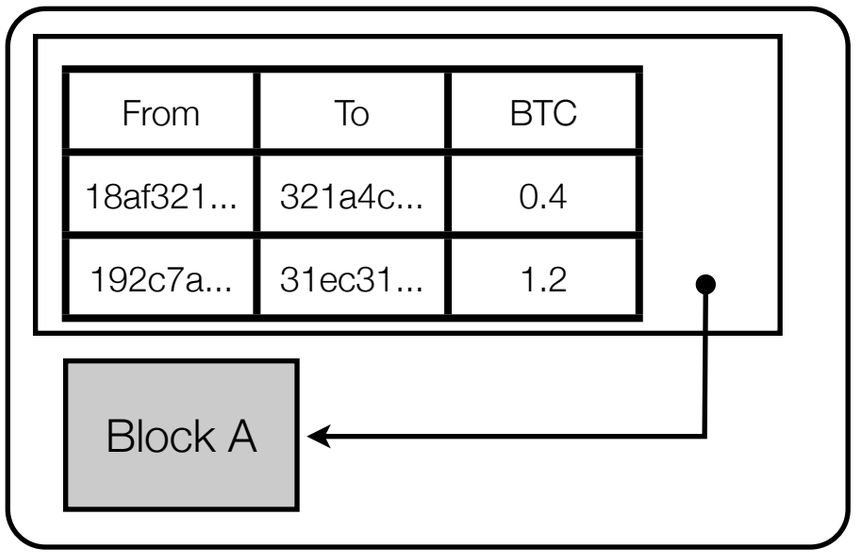
# The Bitcoin network

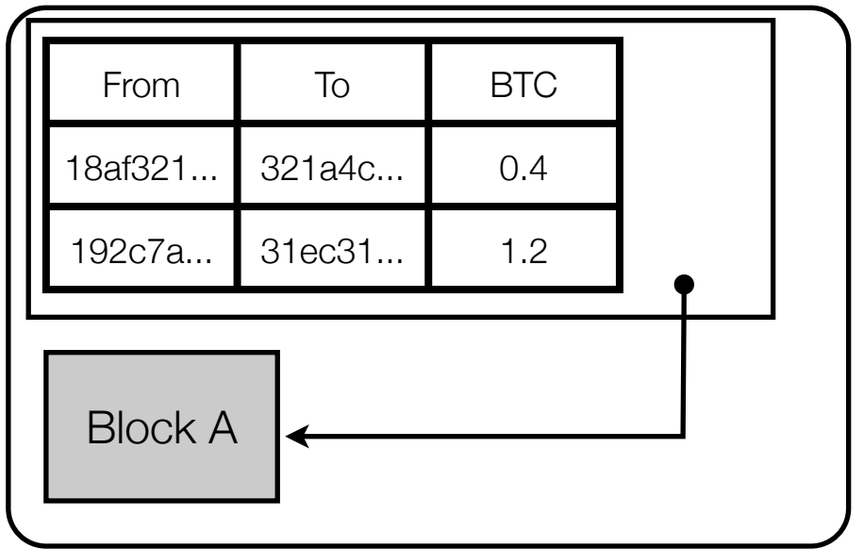
1. New transactions are broadcast to all nodes.

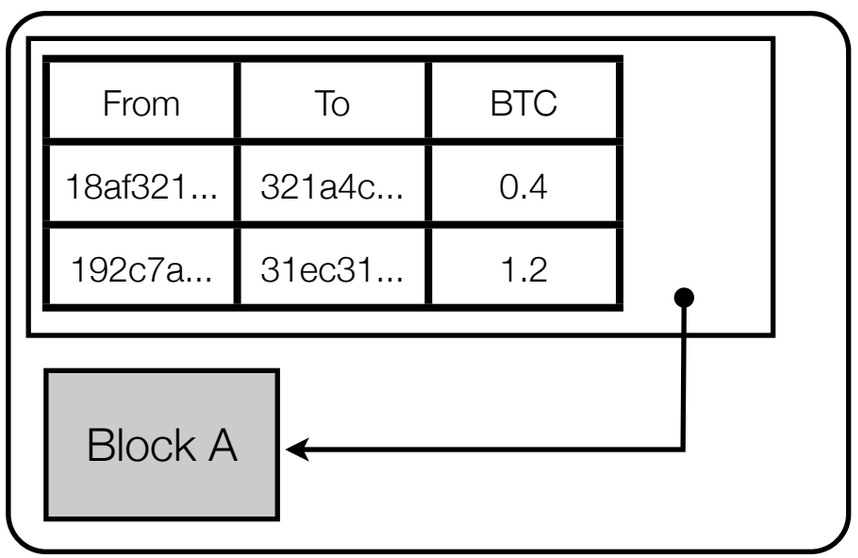


# The Bitcoin network

## 2. Each node collects new transactions into a block.

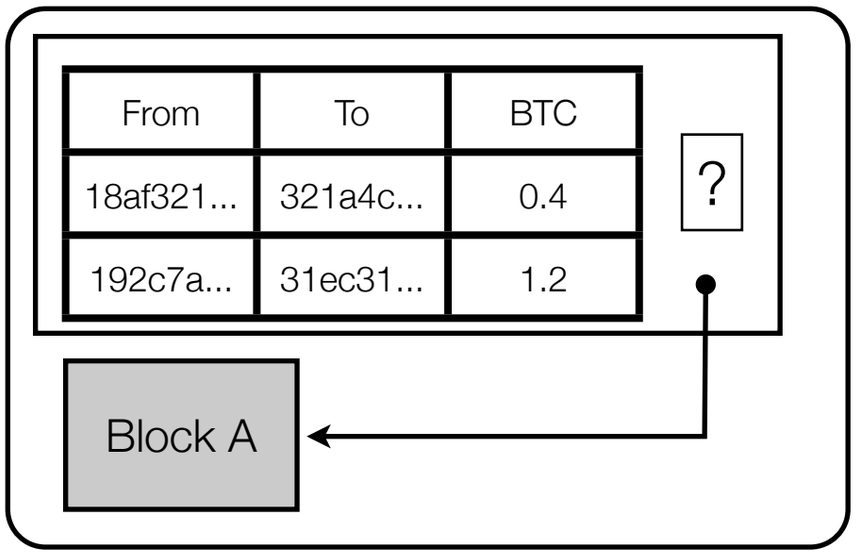


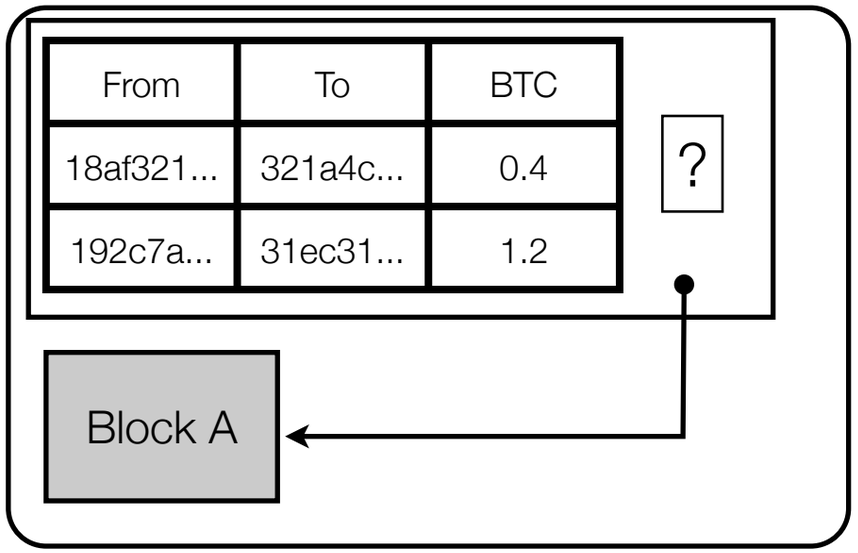



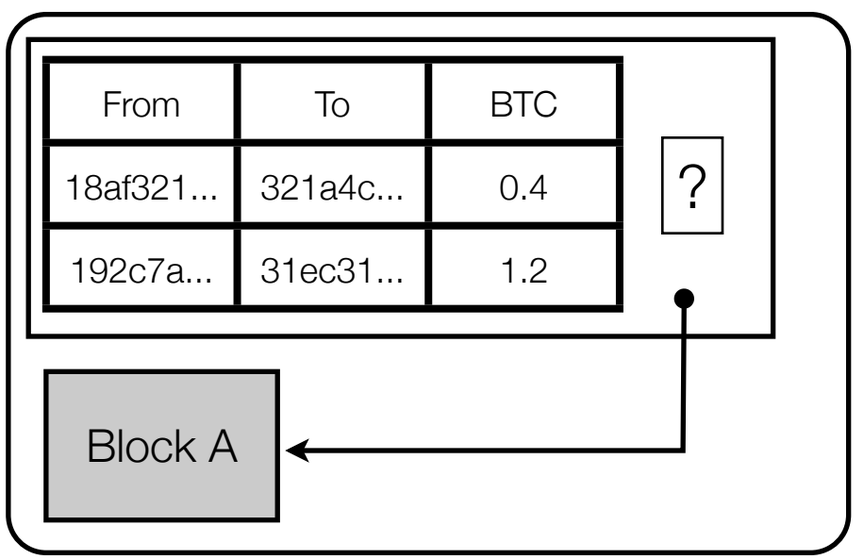



# The Bitcoin network

3. Each node works on finding a difficult proof-of-work for its block.

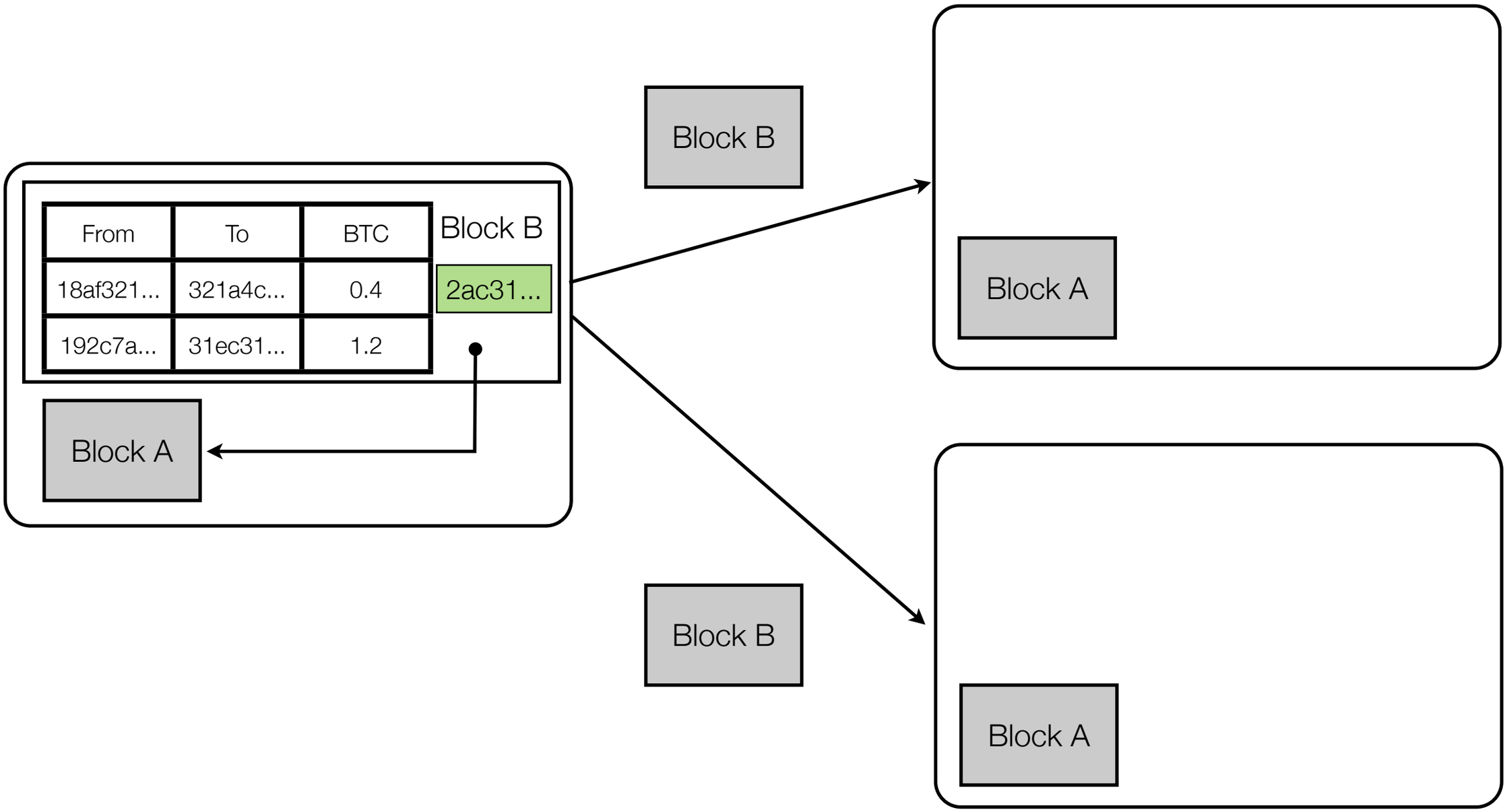


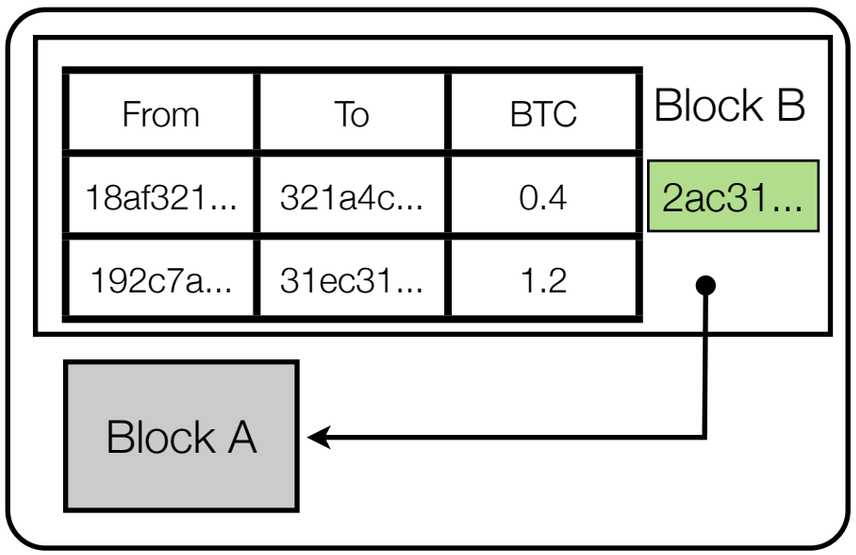
# The Bitcoin network

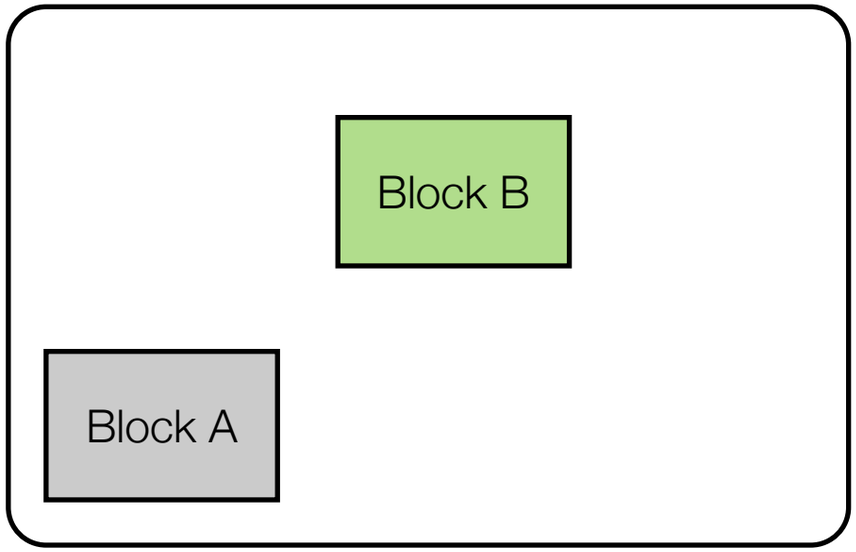
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.

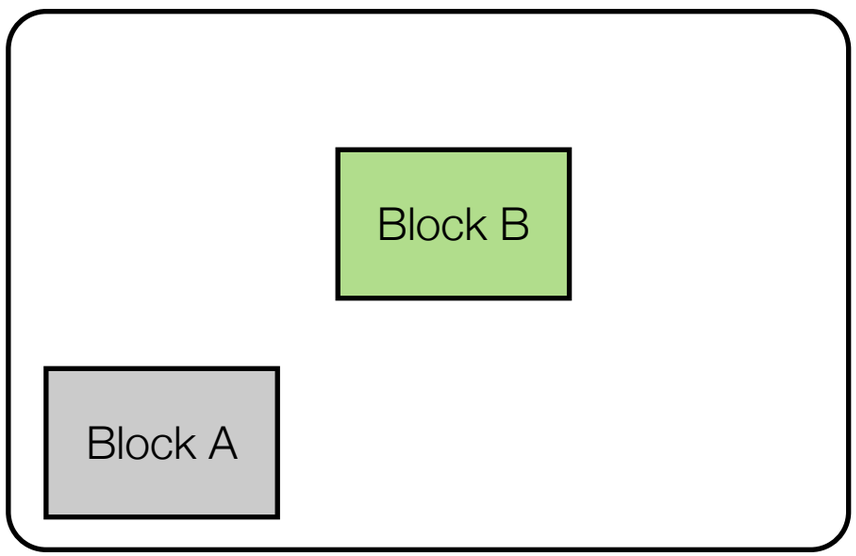


# The Bitcoin network

5. Nodes accept the block only if all transactions in it are valid and not already spent.

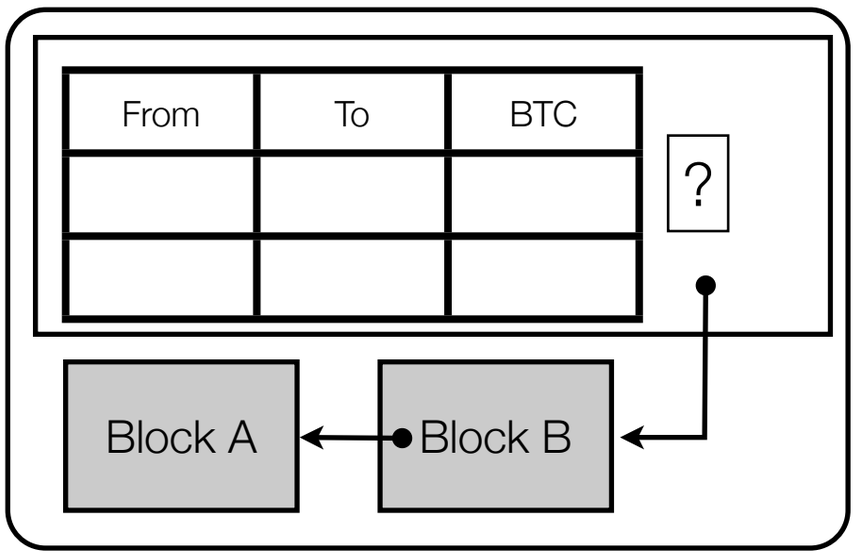


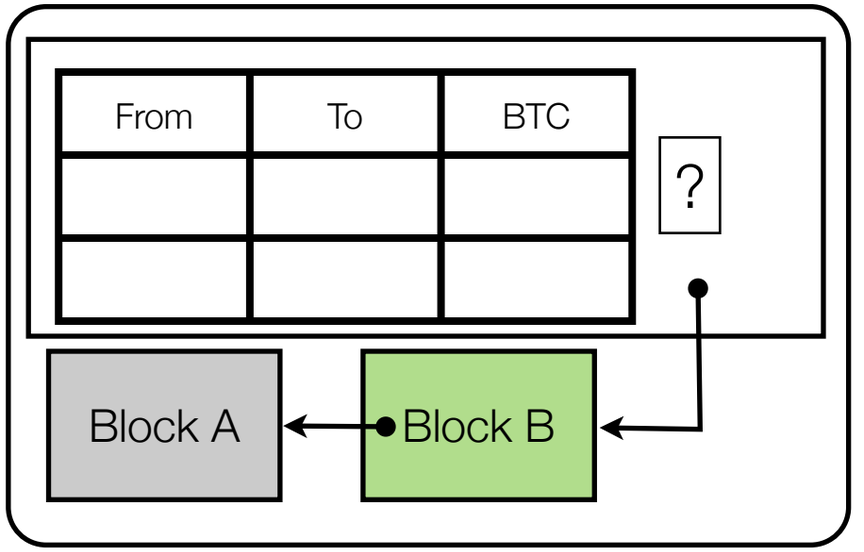



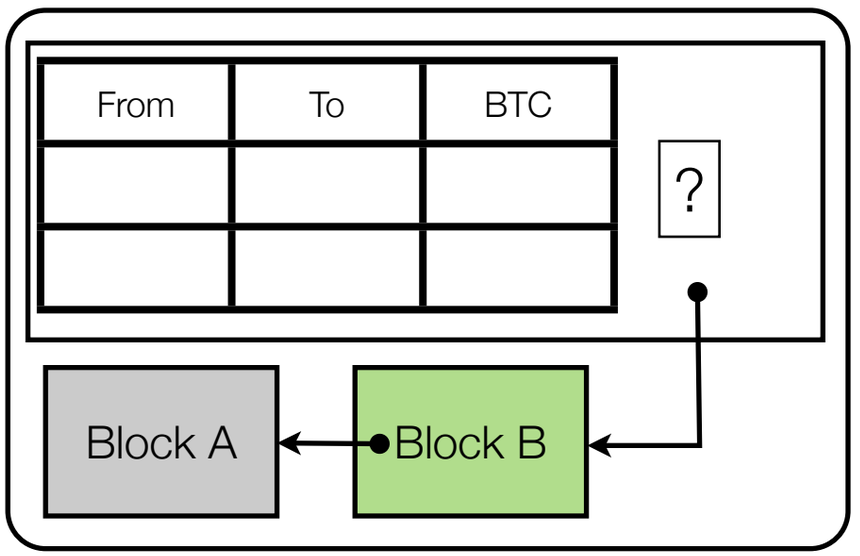



# The Bitcoin network

6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.



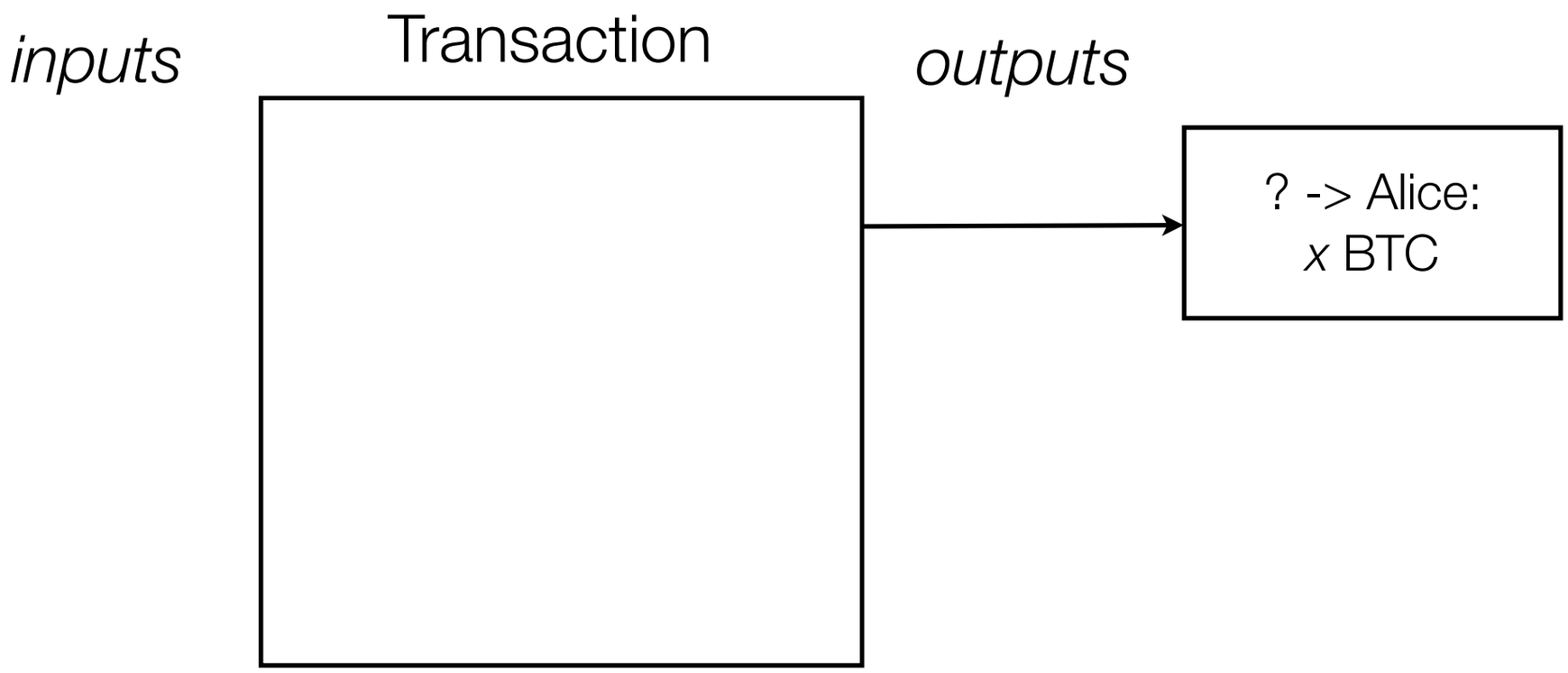
# How Bitcoin solves the consensus problem

- Nodes vote with their CPU power
- Nodes accept a block by working on extending the block
- Nodes reject a block by refusing to work on it

# Mining

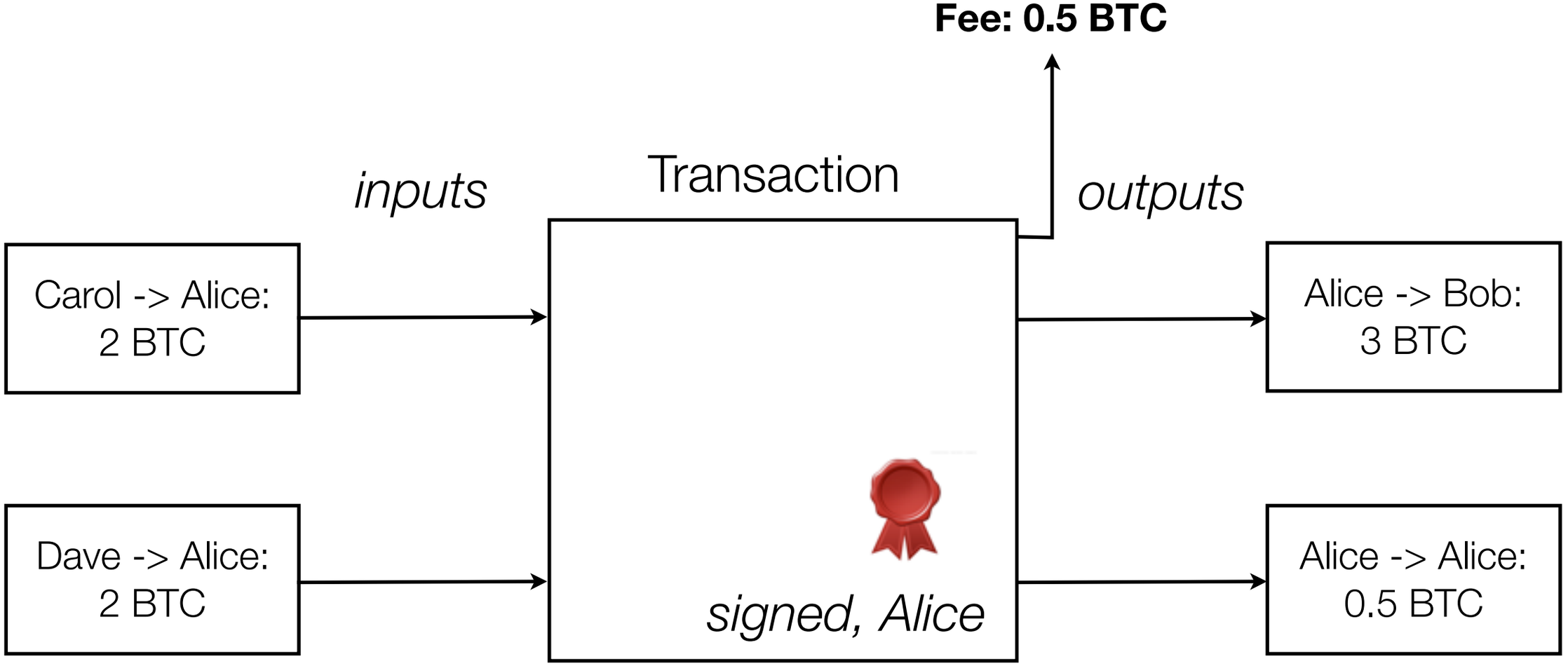
- The first transaction in a block is a special transaction that **transfers new bitcoins** to the creator of the block.

- This is the only way new Bitcoins enter circulation.



# Mining

- If output value of a transaction is less than input value, the difference is treated as a **transaction fee** added to the first transaction in the block



# Exploring the Blockchain

- E.g. blockexplorer.com or blockchain.info

**Latest blocks**[?]

| Number[?] | Hash[?] | Time[?] | Transactions[?] | Total BTC[?] | Size (kB)[?] |
|---|---|---|---|---|---|
| 247902 | 39cd5c1f60... | 2013-07-22 08:06:40 | 311 | 7908.09182124 | 124.35 |
| 247901 | 40b45b983b... | 2013-07-22 07:58:35 | 507 | 58666.0650201 | 248.839 |
| 247900 | 6e675df13c... | 2013-07-22 07:33:05 | 257 | 26001.41201723 | 116.06 |
| 247899 | 4c1e98b6ff... | 2013-07-22 07:24:25 | 157 | 24240.94784428 | 74.324 |
| 247898 | 4dc81d7079... | 2013-07-22 07:16:02 | 75 | 11619.72898059 | 32.481 |
| 247897 | 1e8fa582af... | 2013-07-22 07:13:44 | 84 | 8313.75228293 | 40.778 |
| 247896 | 2d3449a5ea... | 2013-07-22 07:11:21 | 322 | 7891.41153896 | 115.002 |
| 247895 | 1828a01a31... | 2013-07-22 06:57:58 | 46 | 1965.25524938 | 23.973 |
| 247894 | 8935c3b152... | 2013-07-22 06:56:51 | 229 | 4305.08899635 | 95.022 |
| 247893 | 68fea32837... | 2013-07-22 06:47:55 | 1 | 25 | 0.228 |
| 247892 | 41fd27a5fd... | 2013-07-22 06:47:48 | 17 | 1539.28106937 | 21.237 |
| 247891 | 40445f745a... | 2013-07-22 06:43:46 | 158 | 3343.27911845 | 65.118 |
| 247890 | fc2e8272d0... | 2013-07-22 06:33:30 | 34 | 261.103206 | 13.961 |

# The actual Blockchain

- First block known as the “Genesis block” (Jan 3rd, 2009)
- The current longest blockchain: 8+ GB

# Confirmations

- To verify whether a transaction was successful: client queries network to find out about longest chain
- Lookup block in which transaction occurred
- Every block added *after* this block is a confirmation that the network has accepted the block

# Privacy

- **Transactions are** entirely **public**: anyone can see how much Bitcoins are transferred between any 2 addresses

- Necessary to verify double-spending

- But: there is no a priori relationship between a Bitcoin address (a public key) and a user's "identity"

- Keeping your public key anonymous keeps the transactions anonymous

- In practice, not that easy to remain truly anonymous

# Bitcoin Scripts

- Bitcoin transactions may contain *scripts*
- Written in a Forth-like stack-based language. No loops (not turing-complete)
- Script = instructions that describe how BTC in a transaction can be spent
  - Normal transactions have a very simple list of instructions
- Goal: allow complex financial contracts
  - E.g. a transaction whose BTC can only be spent when signed by 10 different keys

# Bitcoin in practice

# What can you buy with it?

- Initially used for anonymously buying questionable / illegal goods
- More and more websites are accepting BTC
  - Reddit Gold, Wordpress.com store

- Some websites are proxies for other websites
  - E.g. BTCBuy allows you to buy Amazon gift cards and pay in BTC
- See https://www.spendbitcoins.com/ and https://en.bitcoin.it/wiki/Trade for a more complete list

# Adoption

- Number of transactions per day since inception in 2009:

# How much is a BTC worth?


"Bitcoin rose 722% in the first six months of 2013"
*(source: The Genesis Block, 2013 Mid-year review)*

# How much is a BTC worth?

**FIGURE 4 - LOGARITHMIC VIEW OF BITCOIN EXCHANGE RATE**


*(source: The Genesis Block, 2013 Mid-year review)*

# Exchanges

- Market places where BTC is bought/sold for EUR, USD, ...
- Typical “entry point” into the Bitcoin market
- Examples:



# Merchant Processors

- Aim to make it easy for merchants to accept BTC
- Merchant processor accepts BTC and transfers USD/EUR to the merchant
- The merchant never has to deal with BTC
- Examples:


# Software Wallets

- Download a piece of software known as a “bitcoin client”
- “Fat” clients: your computer becomes part of the Bitcoin network, requires downloading the blockchain.
  - Example: 
    Bitcoin-Qt
- “Thin” clients: only stores your wallet (public/private keys) and allows you to send/receive BTC. Does not download the blockchain.
  - Example: 
    Electrum

# Online web-based Wallets

- Store your wallet for you
- Convenient but introduces third-party risk!
- Examples: blockchain.info
  - Wallet stored encrypted on server
  - Decrypts using JavaScript on the client

# Paper Wallets

- Offline wallet. To put your bitcoins in a physical safe.
- Basically a private key printed as a QR-code


*(source: bitcointalk.org)*

# Concluding remarks

# A word of warning

- Bitcoin is a young technology
- Highly volatile price
- High risk
- Storing money in online wallet: security issues
- Don’t turn your savings money into BTC (just yet?)

# A glimpse at Money of the 21st Century?

- Bitcoin is money at the speed of the internet
- Rapidly growing list of financial services:
  - Currency exchanges
  - Offer or make loans in bitcoin
  - Buy stock in bitcoin
  - ...