Bitcoin and the Blockchain

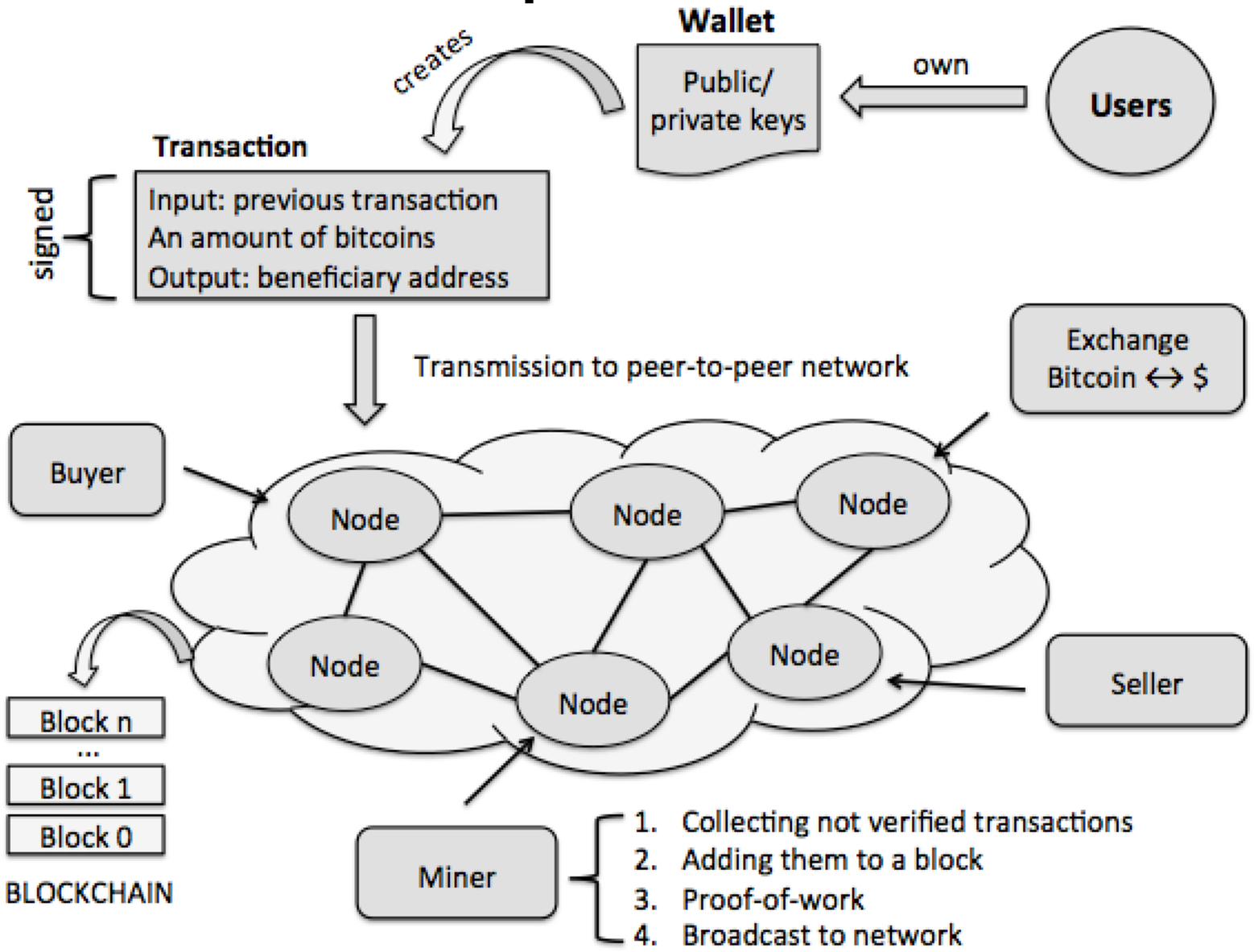
- Bitcoin is a cryptocurrency
  - “Bitcoin” can refer to:
    - Bitcoin (uppercase) - the protocol, software, and community
    - bitcoins (conventionally lowercase) - the unit
- “blockchain” - umbrella term for the datastructure shared among the nodes (the “public ledger”)
- Nodes *asymptotically* reach an agreement on the content of the blockchain
  - Inconsistency (“forks”, i.e. different evolutions) may happen, but are resolved by a “majority vote”, where majority is in terms of computational power (Satoshi agreement)

Bitcoin: The Protocol

- **Distributed public ledger** of **transactions**
- shared with peer-to-peer technology
- Transactions specify the **ownership transfer** of a native digital scriptural asset
- a “digital token” that can be exchanged, but not duplicated
- keeps records of each and every transaction forever
- It could replace any processing central authority with decentralized peer-to-peer cryptographically secure equivalent

What is Blockchain? Basic concepts

- **Transactions:** Transfers of bitcoin from input **addresses** to output **addresses**
- **Blocks:** Timestamped collection of transactions.
- **Miner:** Agent which validates transactions and puts them into blocks
- **Blockchain:** The entire series of blocks 'chained' together
  - Miners compete to add blocks, the “winner” is compensated with bitcoins



Basic Concepts - Identity in Bitcoin

- Bitcoin is really **pseudonymous**, not anonymous
- More similar to «anonymous bank accounts» in fiscal paradises…
- …but whose transactions are visibile to everyone!
- **All transactions are transparent to everybody's inspection.**
- The bitcoin address does not provide direct information about the bitcoin owner
- **Perfect persistent public account history:** the public ledger is forever
- Exchange must identify customers, so if you lost anonymity, they can track all transactions, from the beginning.

Basic Concepts - Identity in Bitcoin

- The pseudonym is generated randomly directly by the user (not issued by any central authority) in “wallets” using **Asymmetric Cryptography**
- Two mathematically linked keys perform opposite digital signature functions:
  - The private (secret) key, used to generate the signature
  - The public key, used by anyone to verify the signature
- The private key cannot be derived from the public one
- A bitcoin address is derived from a public key, but the public key cannot be derived from the address

  Private key -> public key -> bitcoin address
- The corresponding private key allows spending from that address

Basic Concepts - Identity in Bitcoin

- A user can generate as many keys (and addreses) he wants
- Could two users generate the same keys (and addresses)?
- No, in practice, because the amount of keys is HUGE
  - Example Address: 3EnQkjmt3Pv2Uyk8gG736xYKen9efED5LQ
  - 2^{160} possible addresses (1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976 addresses)
  - Grains of sand on earth: 2^{63}

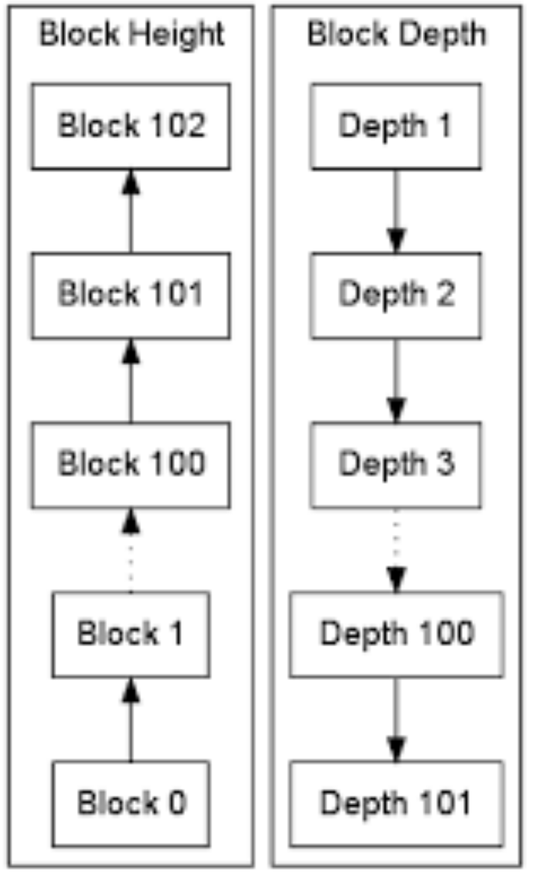
Basic Concepts - Blocks + Blockchain

Blocks

- Contains an ordered bunch of transactions
  - Timestamps the transactions, are immutable
- Each block references a previous block
- Each block has height and depth (confirmations)
  - Currently 612k blocks…and counting

Blockchain

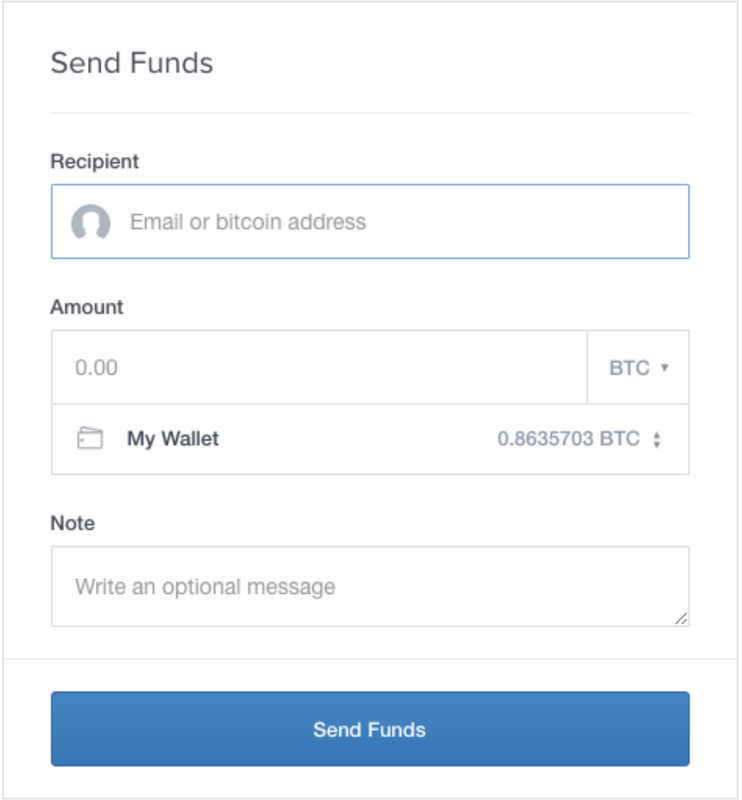
- The entire series of blocks 'chained' together
- All nodes participating to the blockchain have a complete copy of the blockchain



Block Height Compared To Block Depth

A Bitcoin Transaction - Basic Version

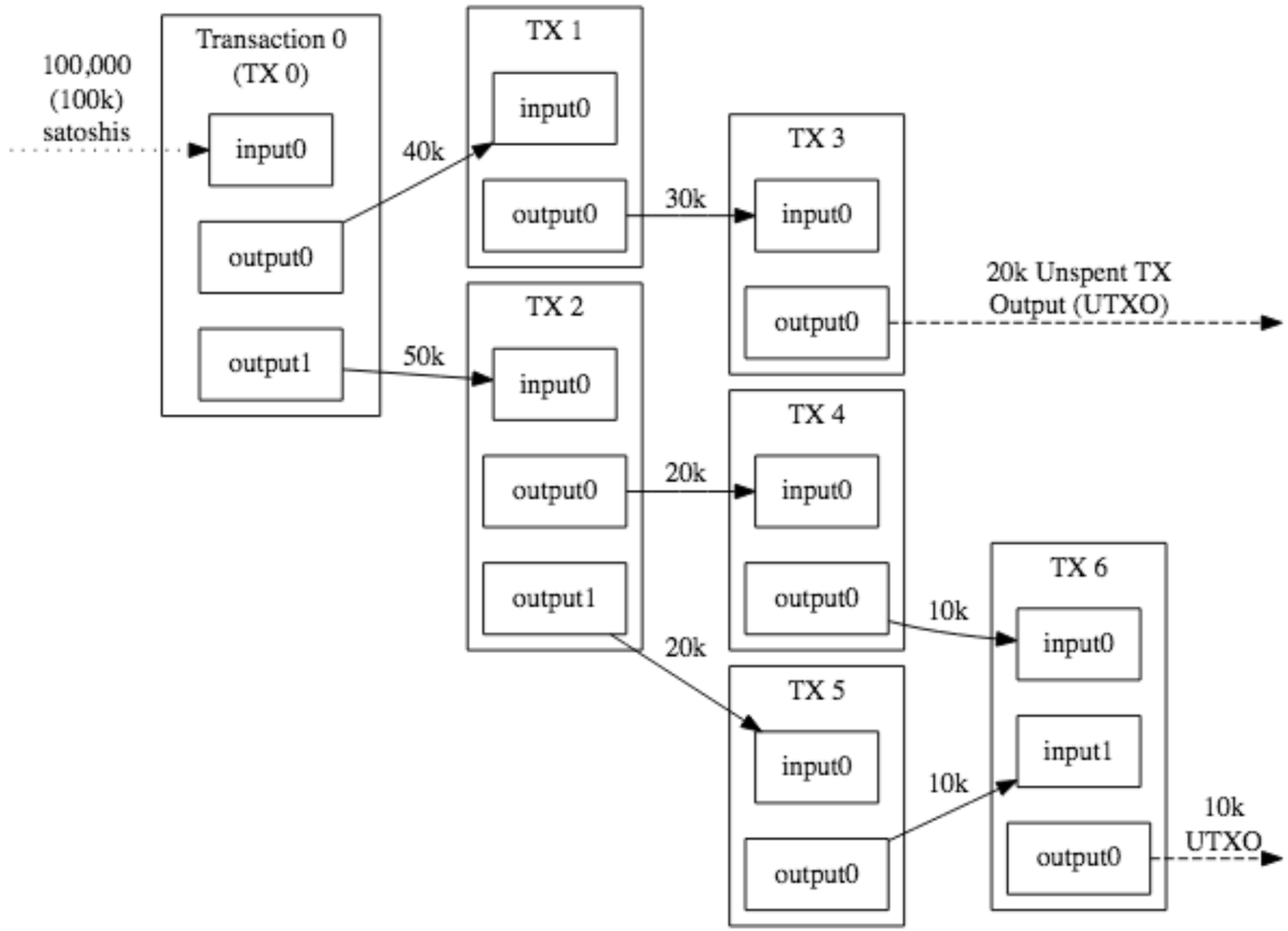
- Bitcoin exists as software
  - Transactions are conducted through wallet software
  - Wallet creation generates a Bitcoin address
  - This is my Bitcoin address (in case you wanna give me a tip): 3EnQkjmt3Pv2Uyk8gG736xYKen9efED5LQ
- To receive money, you share your address
  - Sender specifies address and amount
- The transaction is broadcast to the network, where "miners" verify it and add it to the transaction history
- Once validated, the transaction is stored forever
- Possibly a note can be added to the transaction (and stored as well)



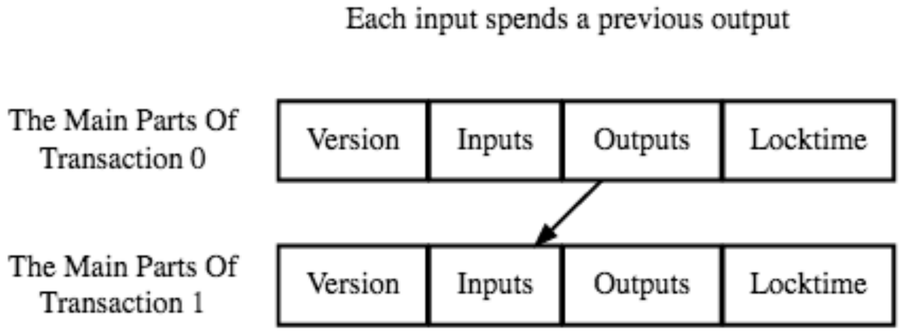
Coinbase interface

Basic Concepts - Transactions

- Maps inputs addresses to output addresses
  - Outputs can only be spent once
- Typical tx: one input, two outputs
- Fees are implicit



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin



Each output waits as an Unspent TX Output (UTXO) until a later input spends it

Transaction View information about a bitcoin transaction

447cb6623db32b5f28c94ac10551802075f053208fe995204a145197e2904bb9

3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v (44,000 BTC - Output)	3LrLWTSdd69oZVVQ6dtWaAAaBLn7N3rRjz - (Spent)	333.33328889 BTC
	3QkXtcSWJA9w77eCujnMBKDWFe7F7zwxTg - (Spent)	333.33328889 BTC
	3Qd7hXZoZ1iyXZznrbdUwUQBxHMmujdqhJ - (Spent)	333.33328889 BTC
	3ECJwvx9VgfotcUuEJMVNvmWnTGVMk179L - (Spent)	333.33328889 BTC
	3BuQmbmdce3e31GEovq5SgowLdfMgJzLDE - (Spent)	333.33328889 BTC
	3NwKLjJjzXSnBFQWokXRgBG3JeuF3bsnfE - (Spent)	333.33328889 BTC
	3GEaT8ZRXELcjMSFvGro6eZcC5S1LSLZuN - (Spent)	333.33328889 BTC
	35DVAzDtZDKAU94kFT9sxoscnuLCTxgwYc - (Spent)	333.33328889 BTC
	3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v - (Unspent)	38,000 BTC
	35mwqShnStDro6uEB4bmsgbyBo8en6Byfm - (Spent)	333.33328889 BTC
	39pvSqfNcUosc8RGVWxyzKM3ny96a3uSkW - (Spent)	333.33328889 BTC
	39QNJSgQg5JnBXAtbF8ezkDn72VqWdPZPJ - (Spent)	333.33328889 BTC
	3L9qAGBQLbXkFAB2GpijnJXPScSVjuiJio - (Spent)	333.33328889 BTC
	37WSkANPVUQ8uuktf8hv671CejRtBtQ4tJ - (Spent)	333.33328887 BTC
	3EEwPZZ6pYRJSotCz9RBoVYPRnoWyGWEka - (Spent)	333.33328889 BTC
	3C4ABC7iPcAAKBh6SJXfvUSDBew3abCtw3 - (Spent)	333.33328889 BTC
	3HpQozfTzoXAsHf87m2mwJXUQ14LVtLgK4 - (Spent)	333.33328889 BTC
	337RfngTLRTpU7RT9sKWQWDdmfcdmWnugi - (Spent)	333.33328889 BTC
	3P2eoKr3vAeZhJcTzon3VFkv5r7DqSXW9G - (Spent)	333.33328889 BTC
		43,999.9992 BTC

Summary	
Size	1055 (bytes)
Received Time	2016-08-30 11:45:03
Included In Blocks	427512 (2016-08-30 11:51:09 + 6 minutes)
Confirmations	854 Confirmations
Relayed by IP	5.39.93.85 (whois)
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	44,000 BTC
Total Output	43,999.9992 BTC
Fees	0.0008 BTC
Estimated BTC Transacted	333.33328887 BTC
Scripts	Hide scripts & coinbase

Basic Concepts - UTXO analogy

UTXOs stands for "Unspent Transaction Outputs"

- Global set of unspent bitcoins
- "I'm spending THIS bitcoin," not "I'm spending A bitcoin."

Analogous to real estate (land), art objects, or Rai stones of the Yap Islands

- Rai stones are carved and placed somewhere, then never moved
- Instead: common and shared agreement on change of ownership
- The same for land lots: a land lot does not move, only the ownership changes (Cf. *Grundbuch)*

The Innovation of Satoshi Nakamoto

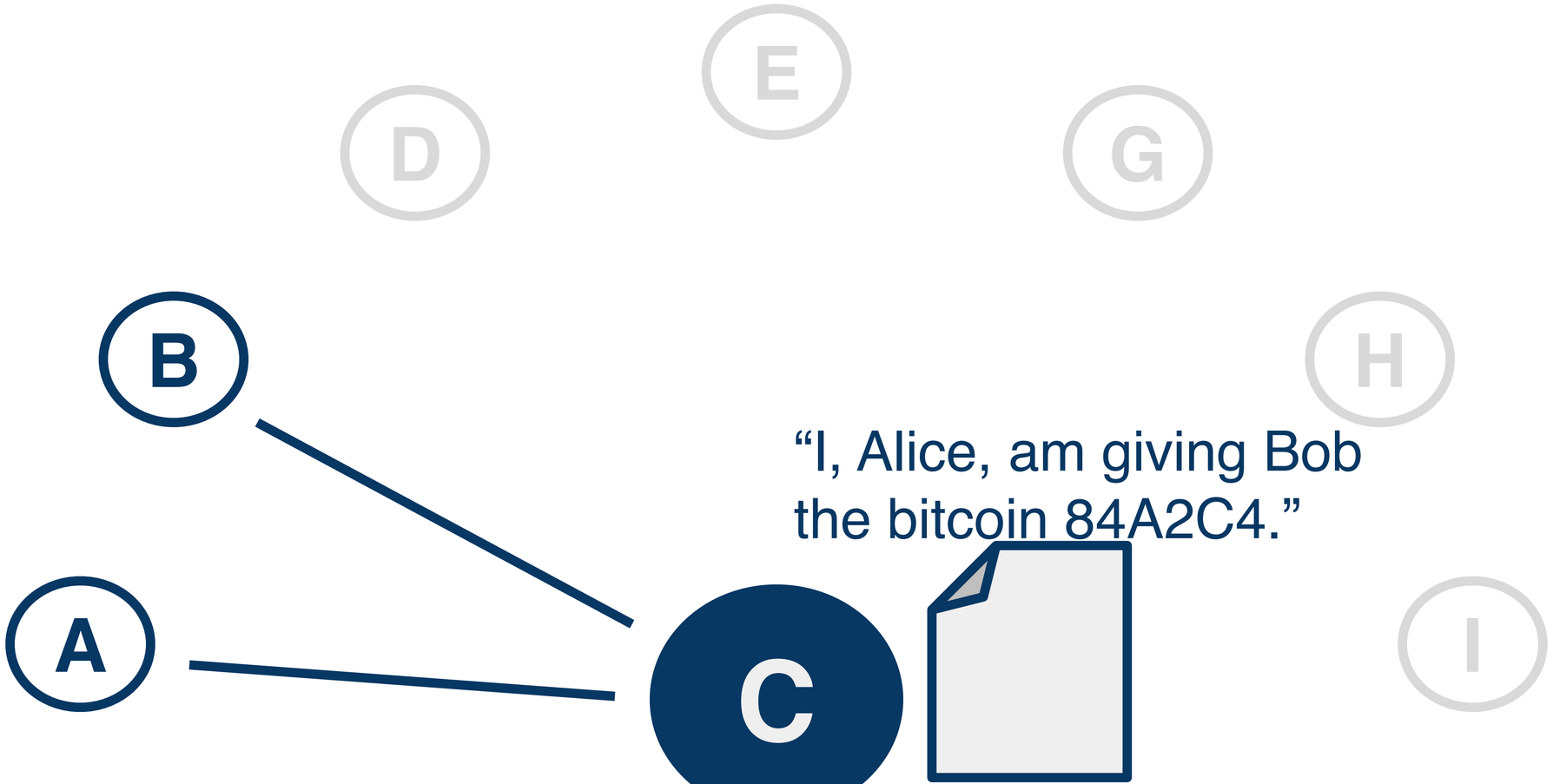
Bitcoin was created by Satoshi Nakamoto in 2009

- First ever decentralized, trustless system for transactions
  - A low cost financial system that only requires an internet connection
- Nakamoto solved the Double Spending problem
  - Prevent someone from spending the same asset twice
  - Solution? The blockchain + Proof of Work

Avoiding double spending: centralised ledger

In a centralized solution, a central bank manages transactions and balances

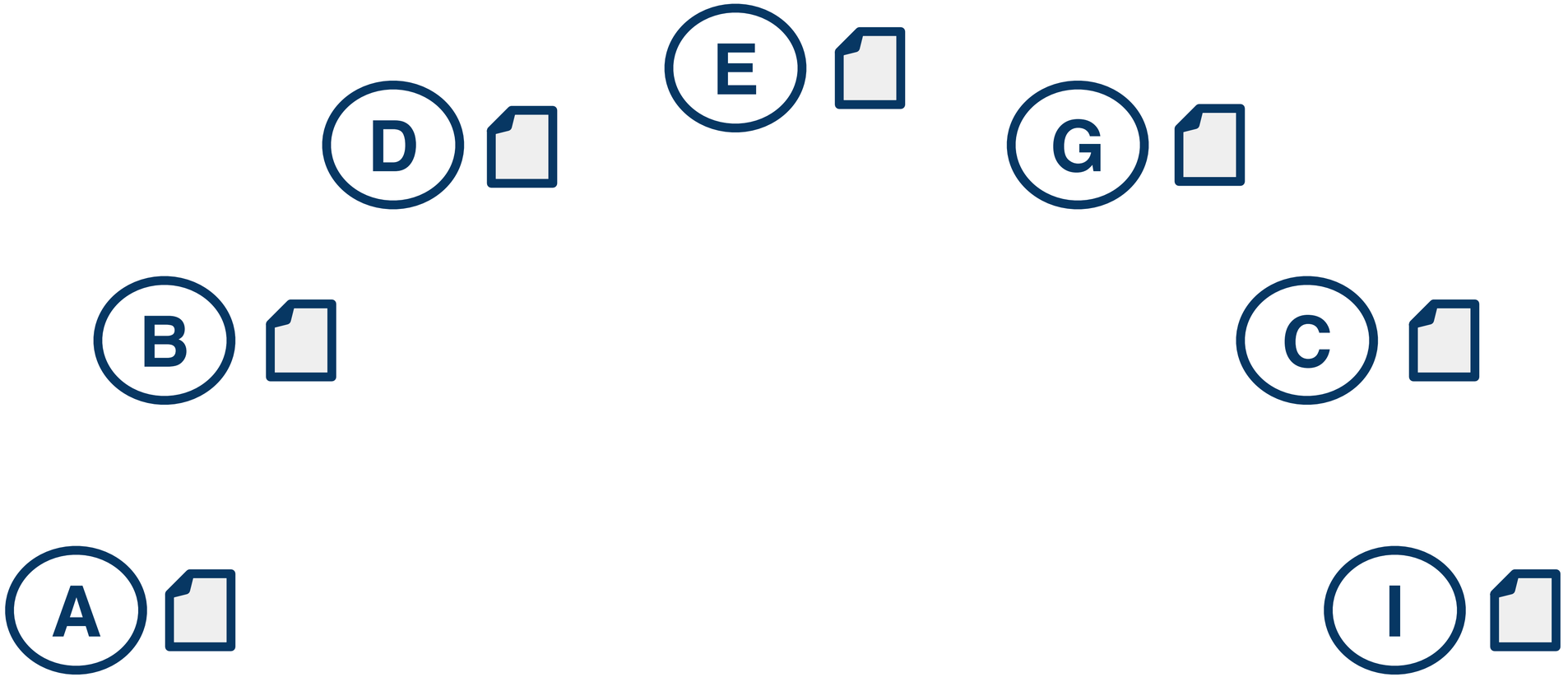
* We have to trust the bank
* Bottleneck and SPOF



Avoiding double spending: decentralised ledger

Decentralization: Making everyone the bank

Every node has a complete copy of transactions

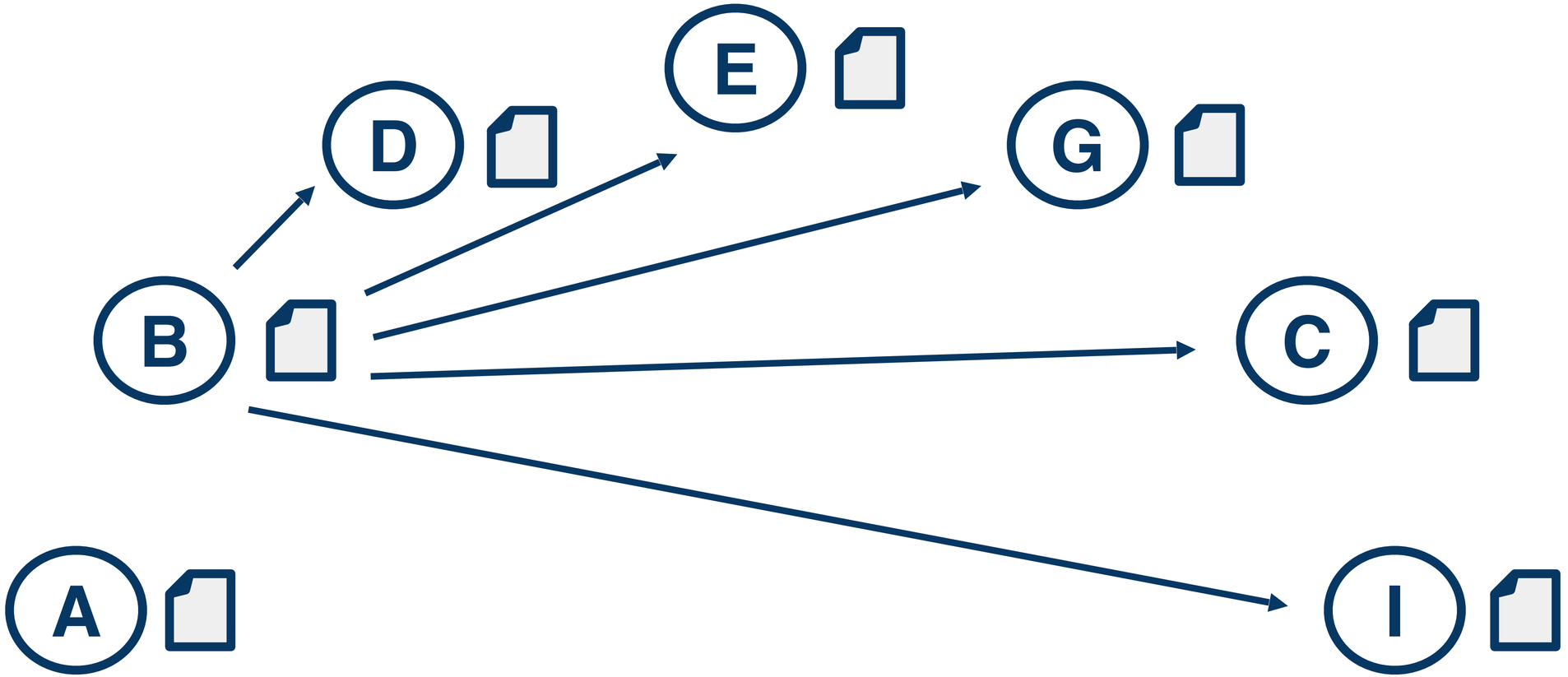


Avoiding double spending: decentralised ledger

Avoiding double spending: decentralised ledger

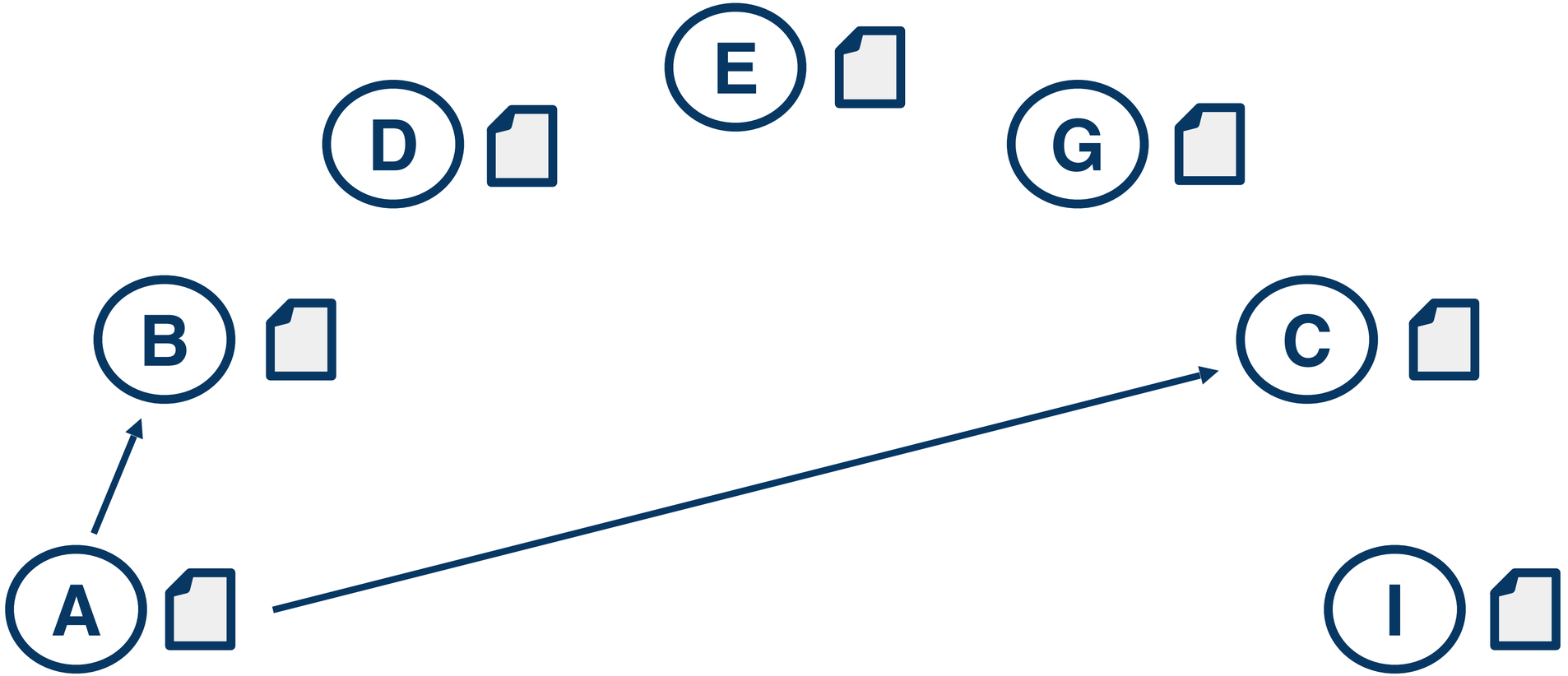
Bob announces the transaction to the world

Every node updates its copy of the ledger



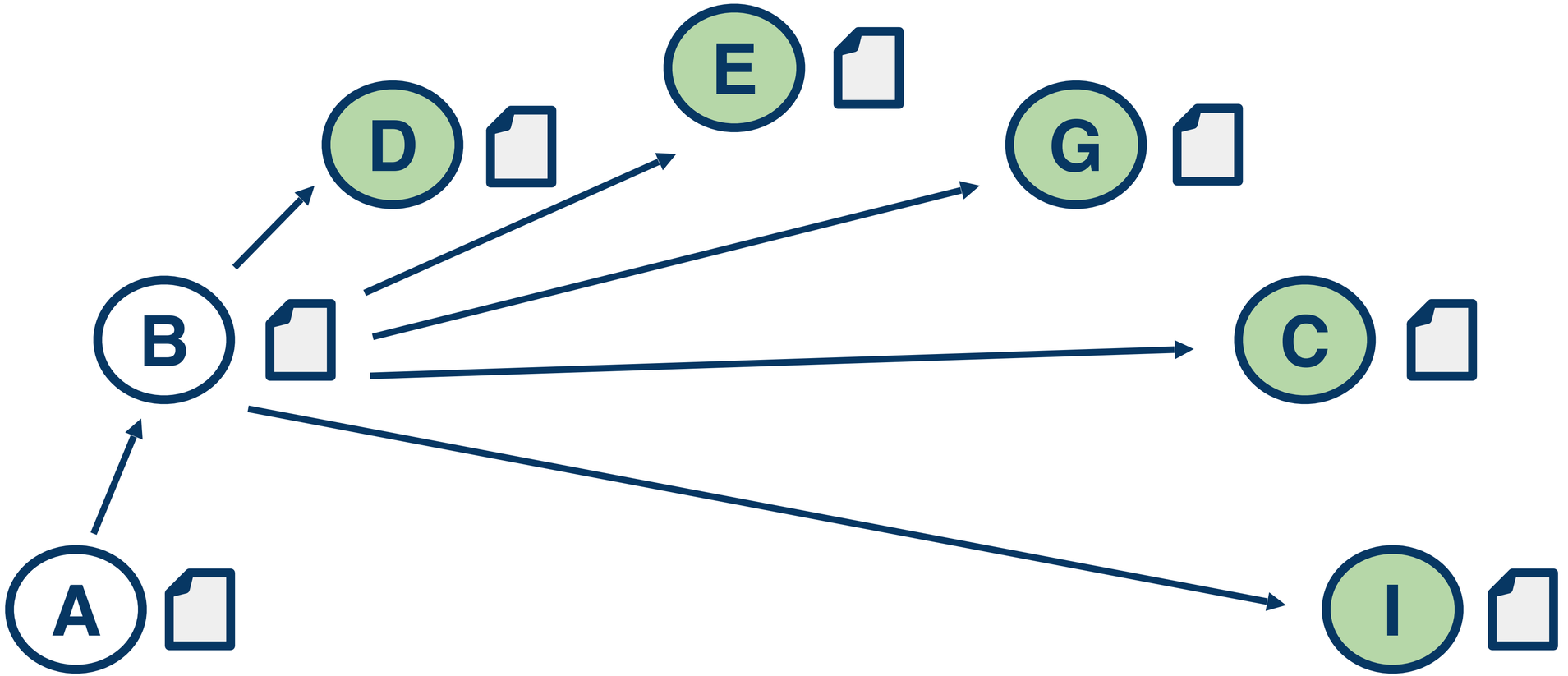
Avoiding double spending: decentralised ledger

But what if Alice double spends the same bitcoin on Bob and Charlie?



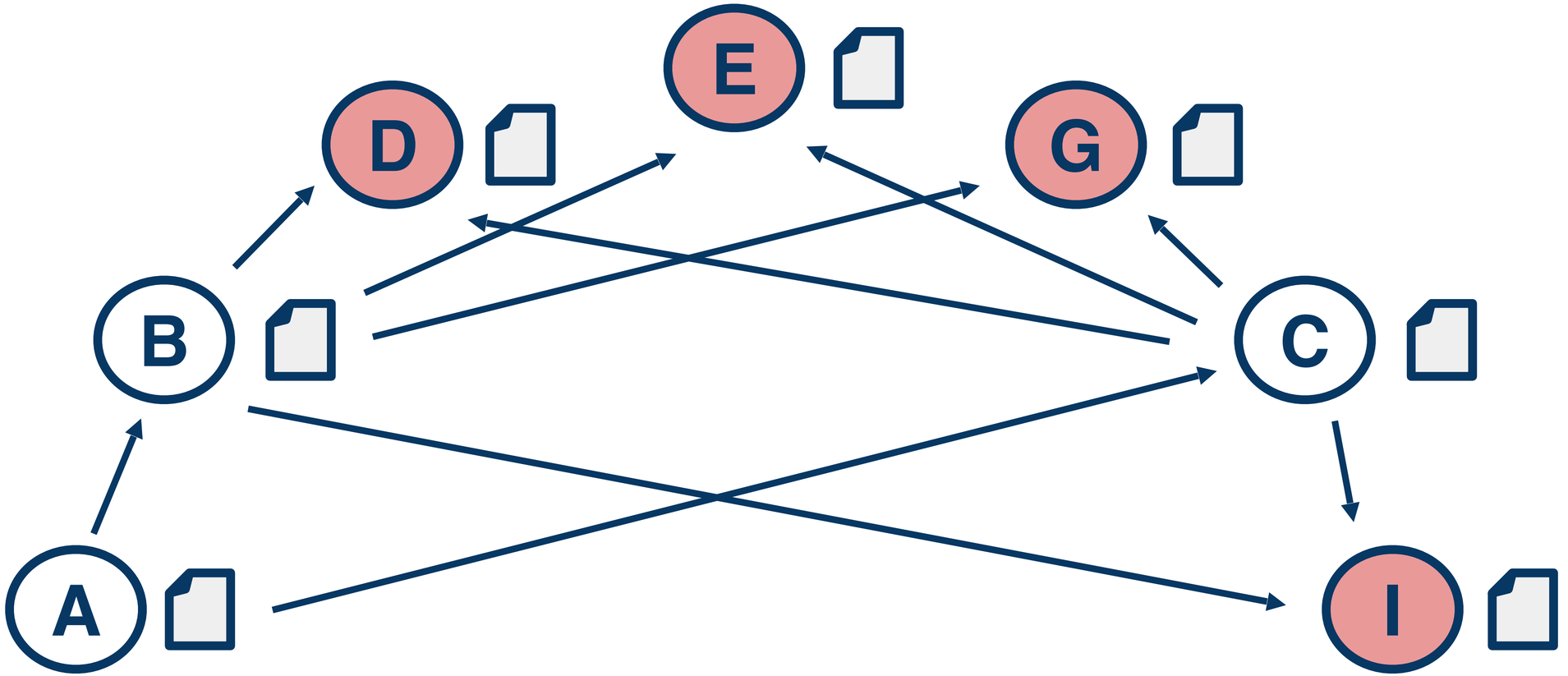
Avoiding double spending: decentralised ledger

Everyone verifies transactions: the first spending is accepted…



Avoiding double spending: decentralised ledger

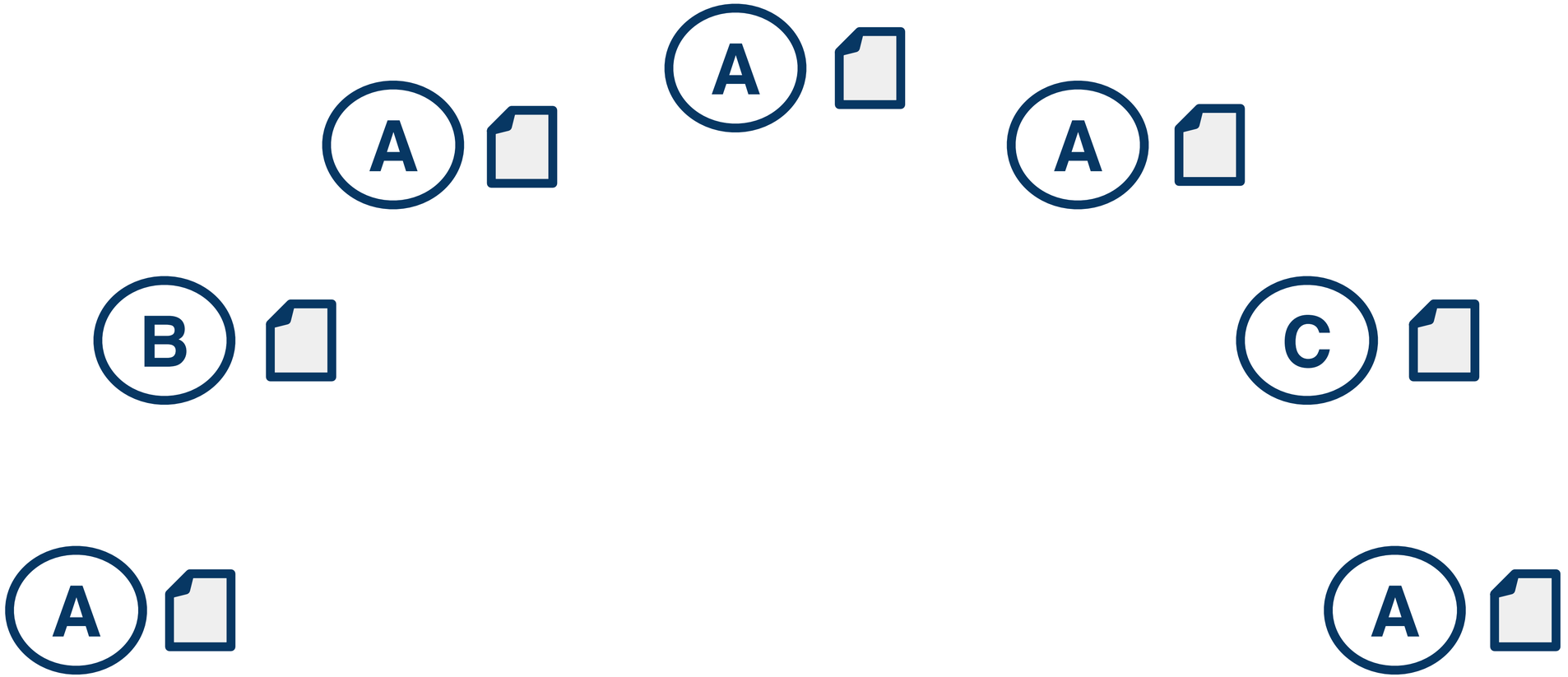
…but not the second, so Alice is prevented from double spending



Decentralised ledger is not enough…

“Sybil” attack:

Alice sets up multiple identities

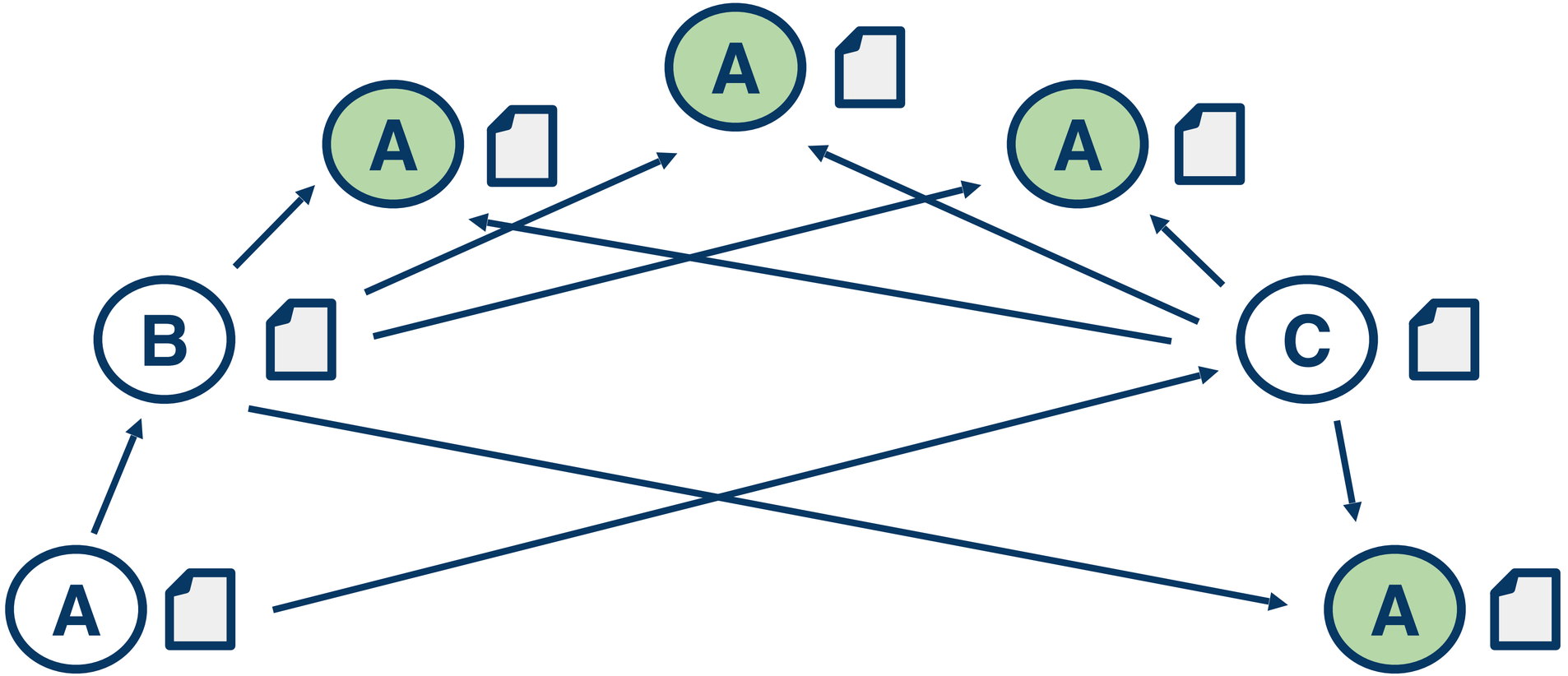


… if many nodes are byzantine

Alice double spends with her multiple identities

The fake identities confirm both spendings, so B and C are fooled!

(A's copies are byzantine)



Satoshi's solution: Proof of Work for verifying transactions

The problem is that Alice can **too easily** generate incoherent confirmations.

Satoshi's idea: **make confirmation hard**.

Let us suppose that a node, David, wants to validate and announce received a bunch of transactions to validate. To this end, he proceeds as follows.

1. David has to check his copy of the block chain to make sure the transactions are legitimate.
2. His computer has to use resources to solve a **hard mathematical puzzle**.
3. Only after the solution has been found, he can to announce the block of transactions to the network.
4. Every nodes can check (easily) that the transactions are legitimate and David has actually found a solution for the mathematical puzzle
5. If everything is fine, every node updates its blockchain (and David gains a reward)

Proof of work is a competition among nodes

- David has to solve the mathematical puzzle in order to avoid double spending
- Proof-of-work as a competition to verify transactions.
    - In Bitcoin, this is called ***mining.***
- If your computer solves the math puzzle before the others, you will verify the pending transactions (and receive some bitcoin as a reward).
- Proof-of-work prevents bad actors like Alice from double spending because it puts them in competition with everyone else trying to verify transactions.
- So as long as most of the computing power on the network is controlled by honest people, *more likely some honest node will win the race before Alice*
- Hence bad actors like Alice will have a hard time doing dishonest things.

Sketch of Bitcoin Mining - The Mining Problem

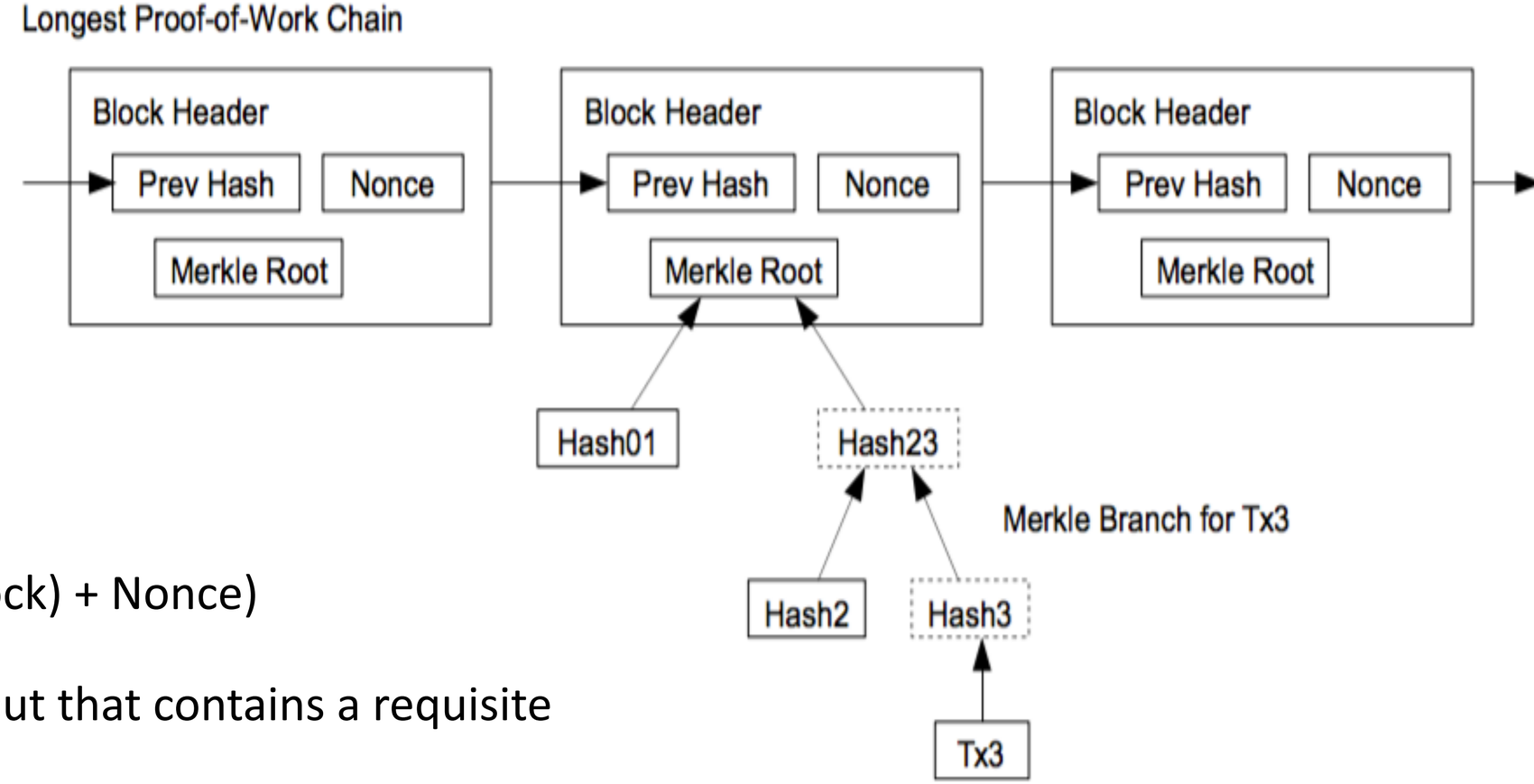
Components hashed together:

- Merkle Root ('summary' of the transactions in the block)
- Hash of previous block
- Nonce

Formally:

Output = SHA-256(Merkle Root +
SHA-256(PreviousBlock) + Nonce)

- Solution (Proof-of-work): an output that contains a requisite number of leading 0 bits
  - The number of 0 bits is the **difficulty**
  - Difficulty adjusts every every 2016 blocks (2 weeks) to maintain 1 block creation / 10 minutes

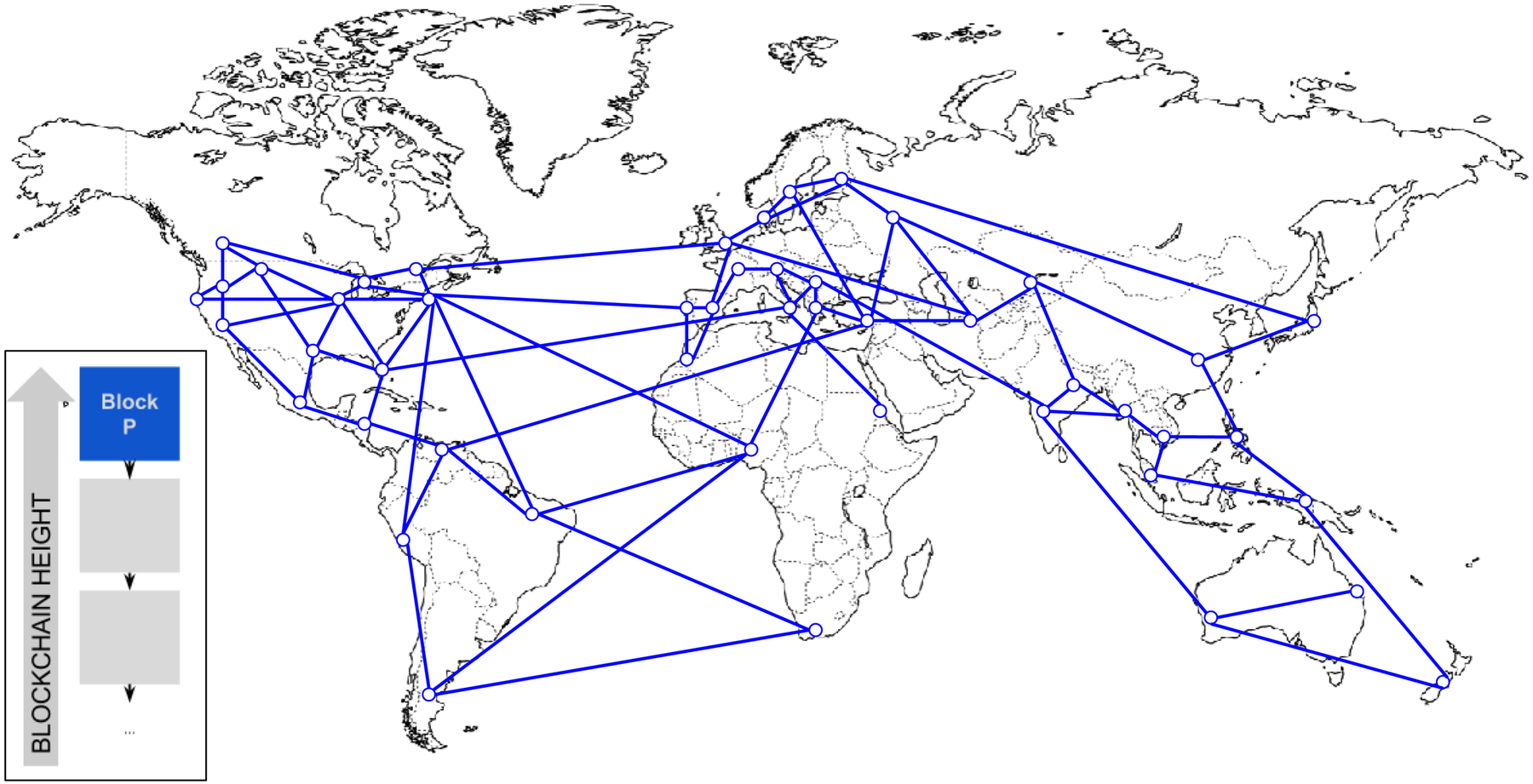


Sketch of Bitcoin Mining - Finding blocks

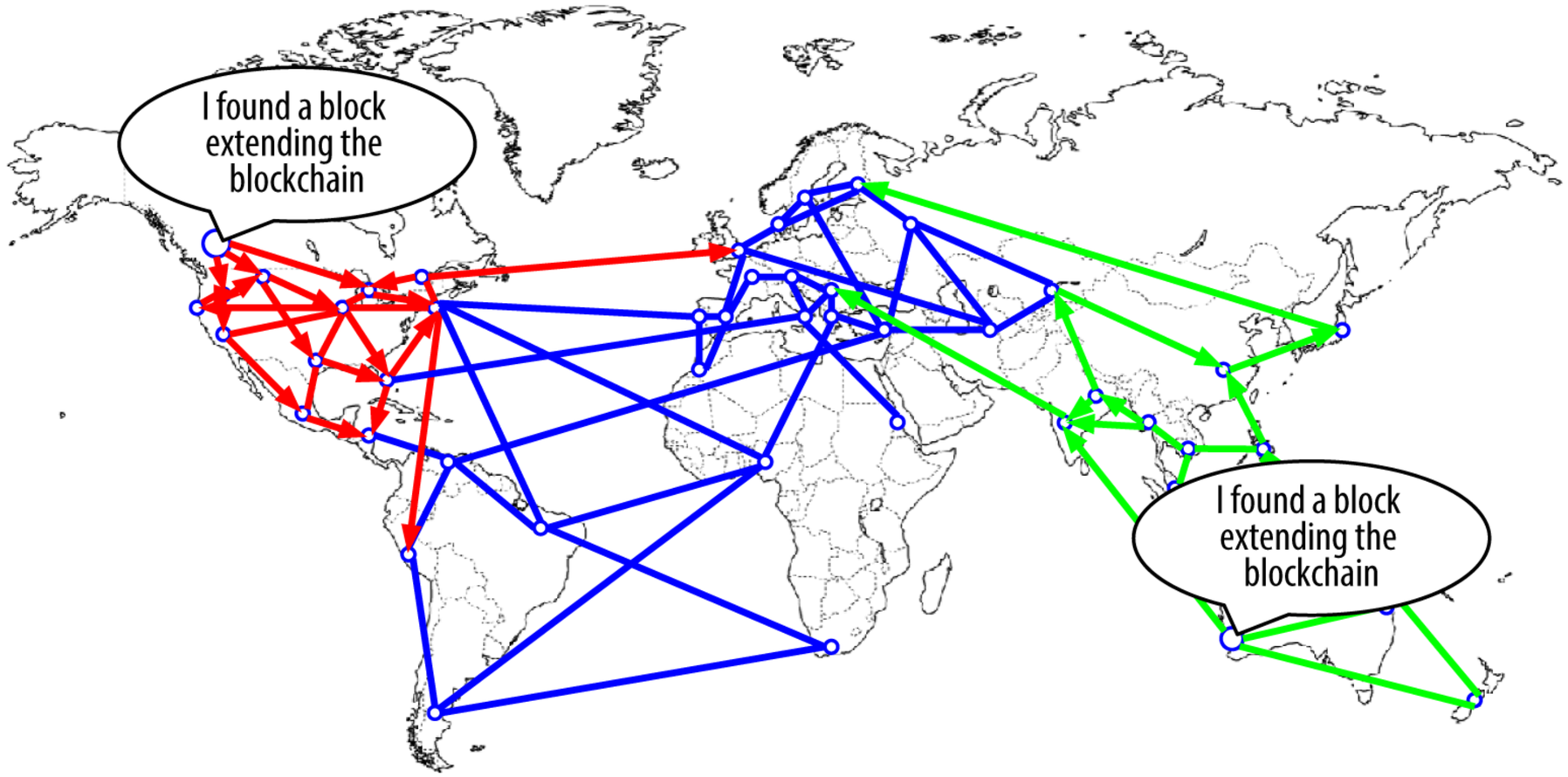
- Solving the PoW => 'found' a block; can add block to blockchain
  - Miner who found block adds "**coinbase transaction**"
    - contains mining reward (currently 12.5 BTC, will halve at 630k)
  - Miner broadcasts block
  - Other nodes verify, then add to their own copy of the blockchain
    - Verification is easy: just compute the SHA256 digest with the given nonce, and check that it gives a legit value
- This happens roughly every 10 minutes
  - Difficulty of the problem adjusted to keep block generation rate constant

A blockchain fork event: before the fork

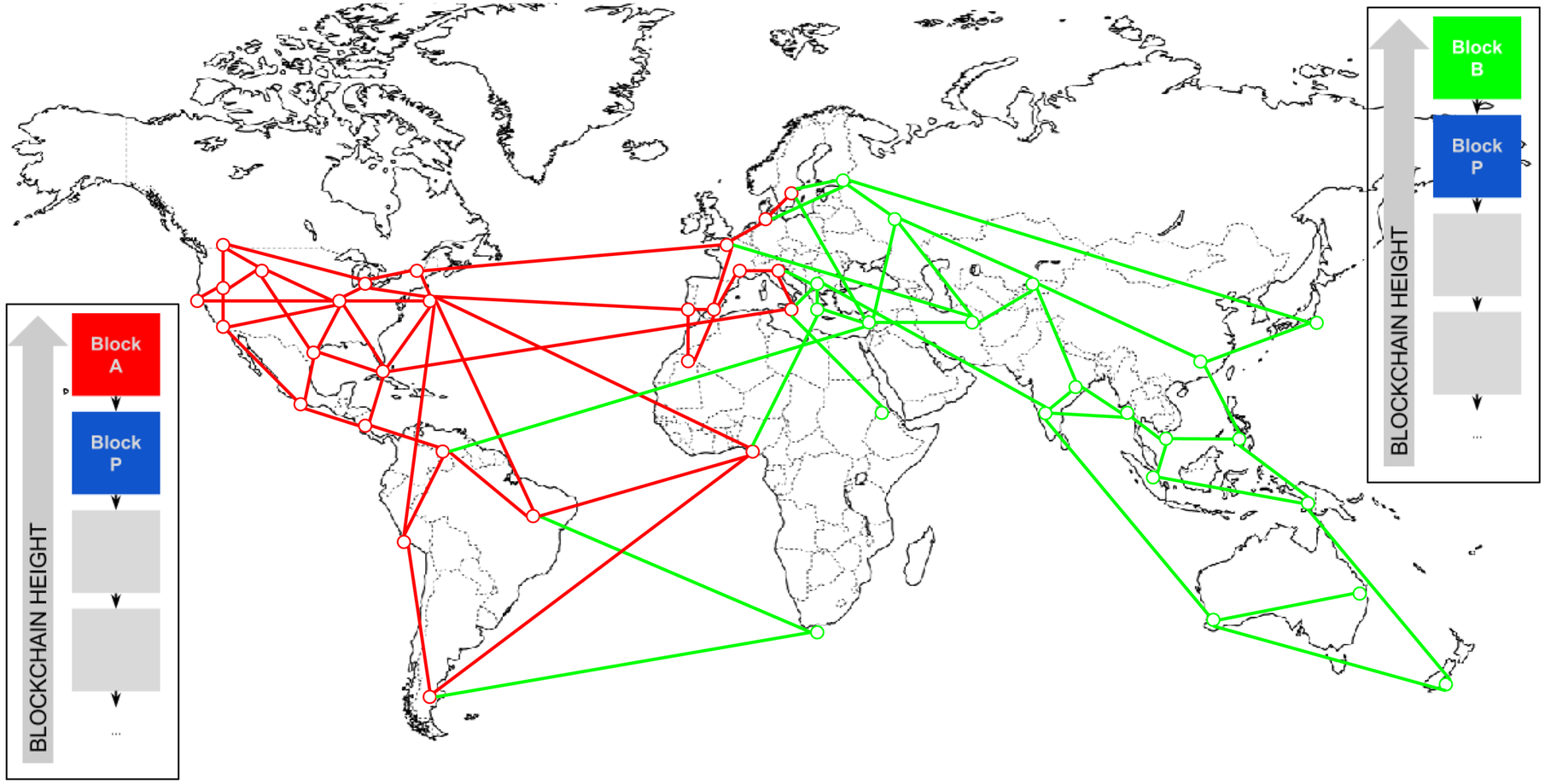
- A fork can occur when two miners publish blocks simultaneously. Such blocks are almost always in conflict.



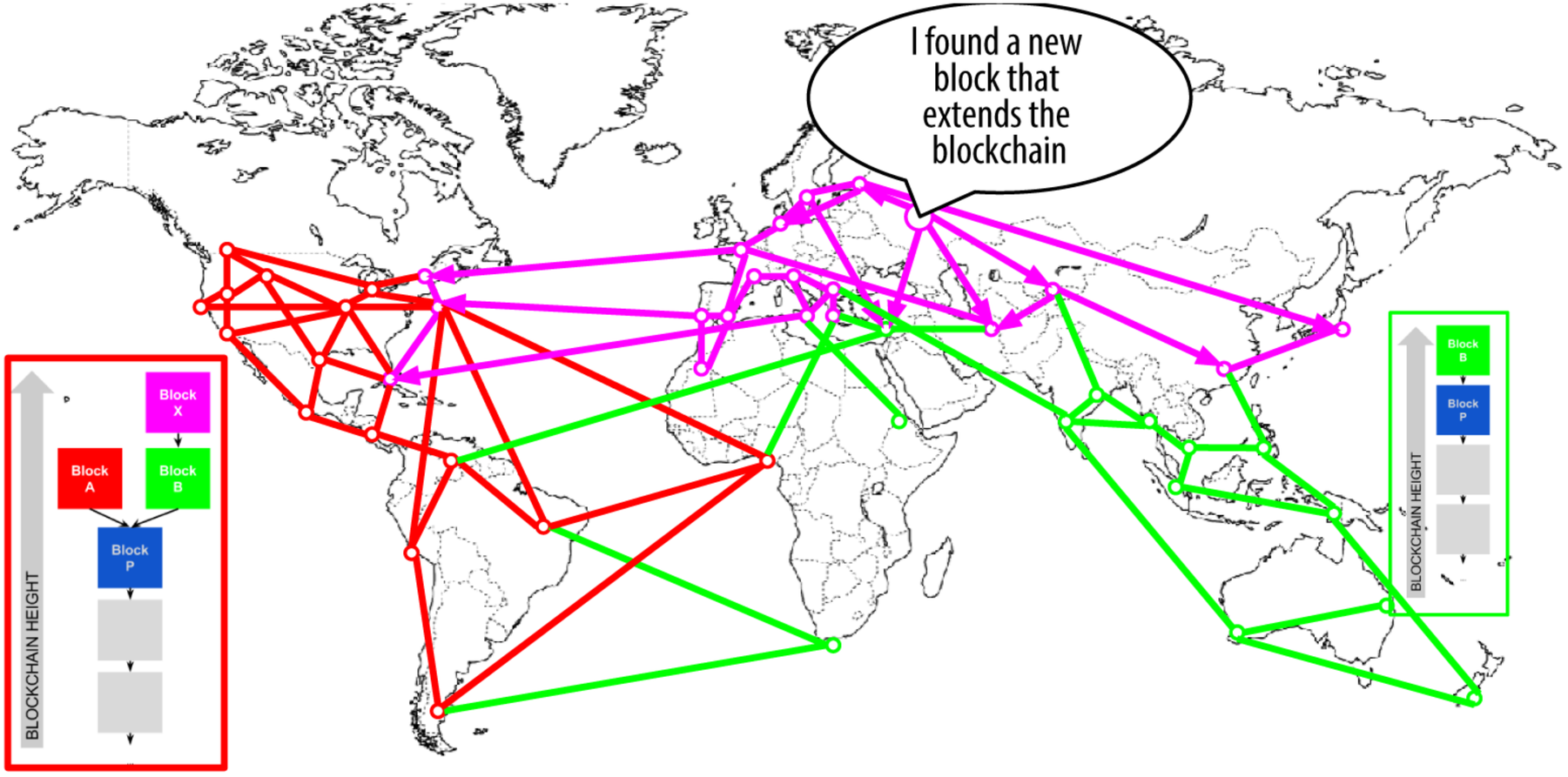
Visualization of a blockchain fork event: two blocks found simultaneously



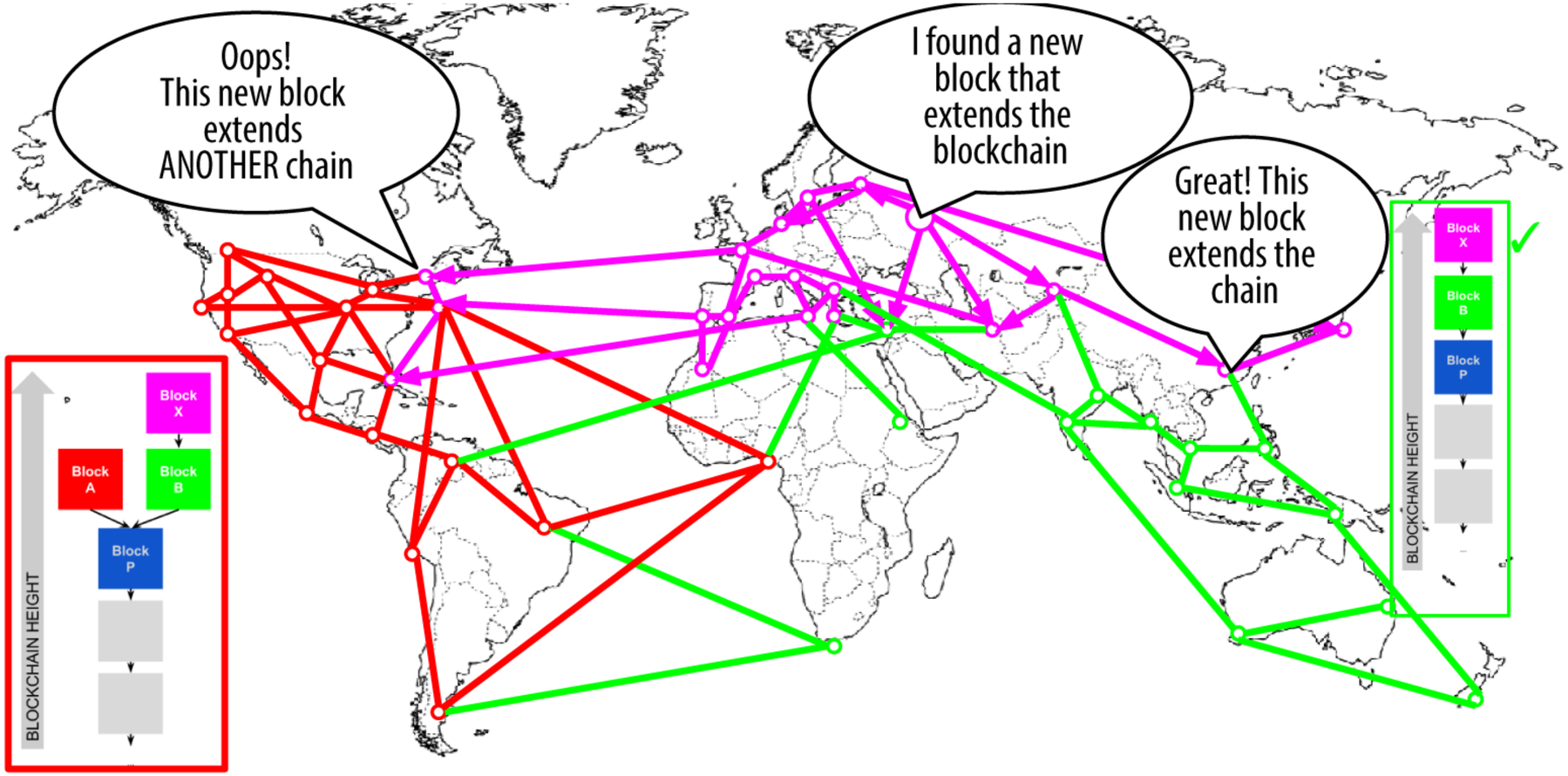
A blockchain fork event: two blocks propagate, splitting the network



Visualization of a blockchain fork event: a new block extends one fork



Visualization of a blockchain fork event: the network converges on a new longest chain



Sketch of Bitcoin Mining - 51% Attacks

Major assumption of Bitcoin:

Strictly less than 50% percent of the whole CPU power in the network is controlled by dishonest nodes

Under this assumption, the honest (CPU) majority will always form, eventually, the longest proof-of-work chain

51% Attack: Attempt to overwhelm the mining power of the network: if attacker nodes are able to assemble so much CPU power, they can «change» the history as they desires.

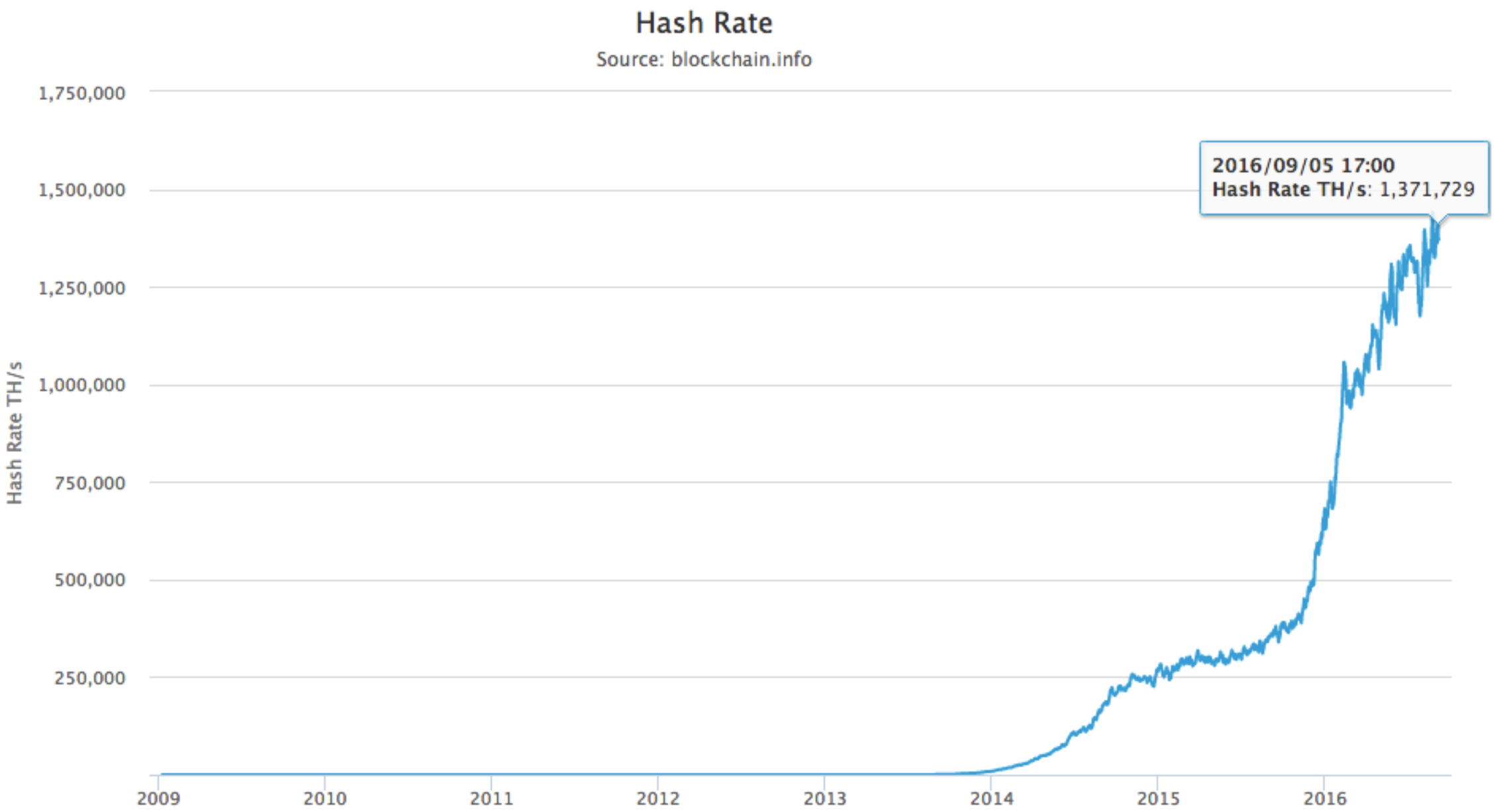
Sketch of Bitcoin Mining - 51% Attacks

However, if a group of nodes assembles so much CPU power, they have this dilemma:

1. Either use it to attack the system – but then nobody would use it anymore, and the value of bitcoins would become zero.
2. Or use it to mine bitcoins, gaining more bitcoins honestly.

If the nodes have interest IN the Bitcoin system, they are incentivized to act honestly – otherwise they would break the system.

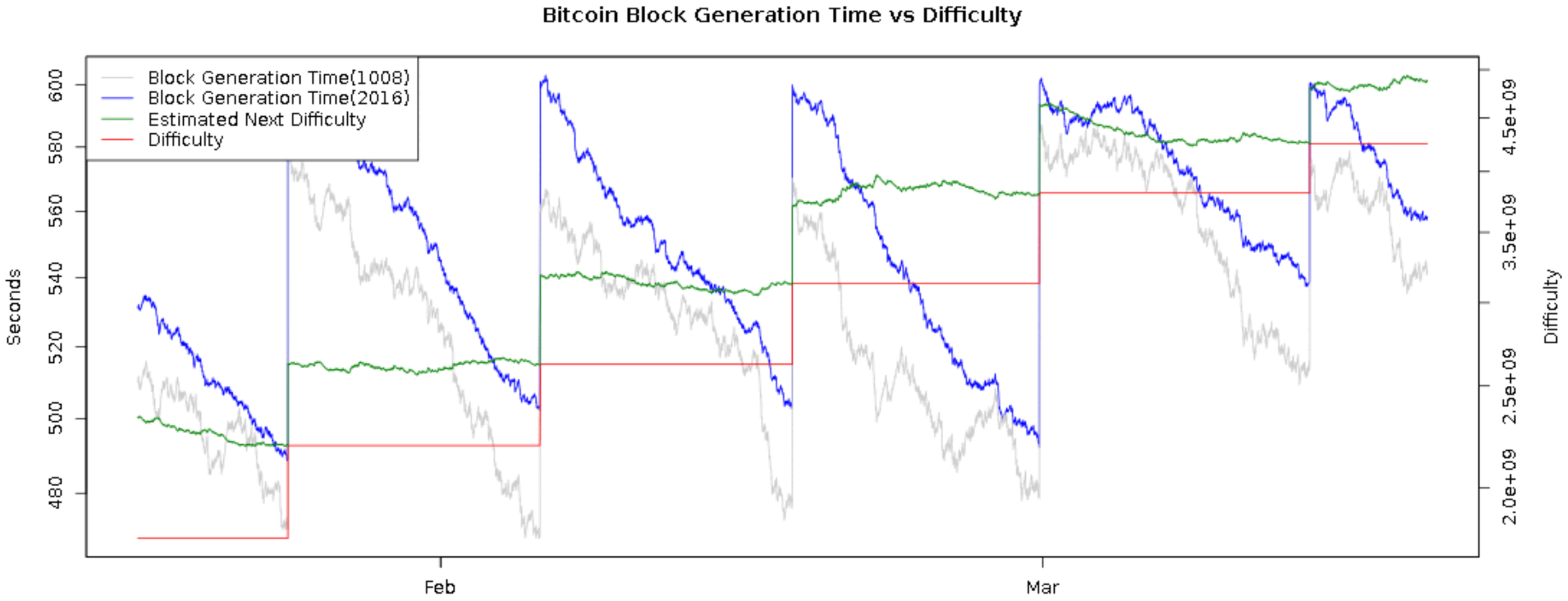
This is however a problem for altcoins – called *altcoin infanticide*
(e.g., a single Bitcoin miner, Luke-Jr., killed CoiledCoin by filling its blockchain with useless blocks much faster than honest Coiledcoin miners)


Hash Rate
Source: blockchain.info
Hash Rate TH/s
1,750,000
1,500,000
1,250,000
1,000,000
750,000
500,000
250,000
2016/09/05 17:00
Hash Rate TH/s: 1,371,729
2009
2010
2011
2012
2013
2014
2015
2016

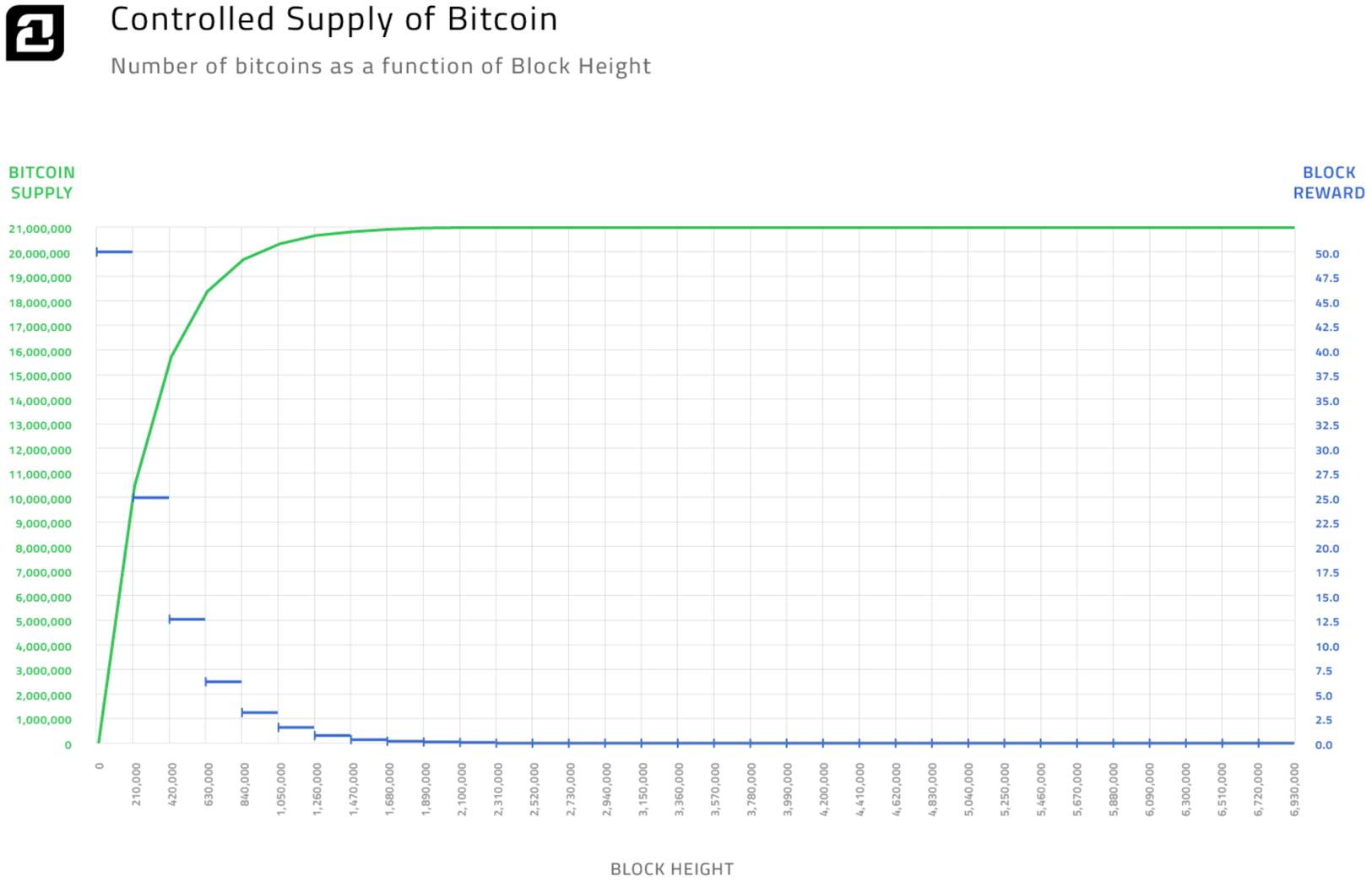
Sketch of Bitcoin Mining - Finding blocks

- Timeline + stats
  - This happens roughly every 10 minutes
    - Difficulty of the problem adjusted every 2 weeks
  - Block reward halving every 4 years
  - Bitcoin is in limited supply: 21 million bitcoins by 2141
    - Deflationary!
- 80% have been already mined
- After that, no further bitcoins will be created
- And bitcoins may go (and have got) lost - forever!
  - Owned by addresses whose private key has got lost

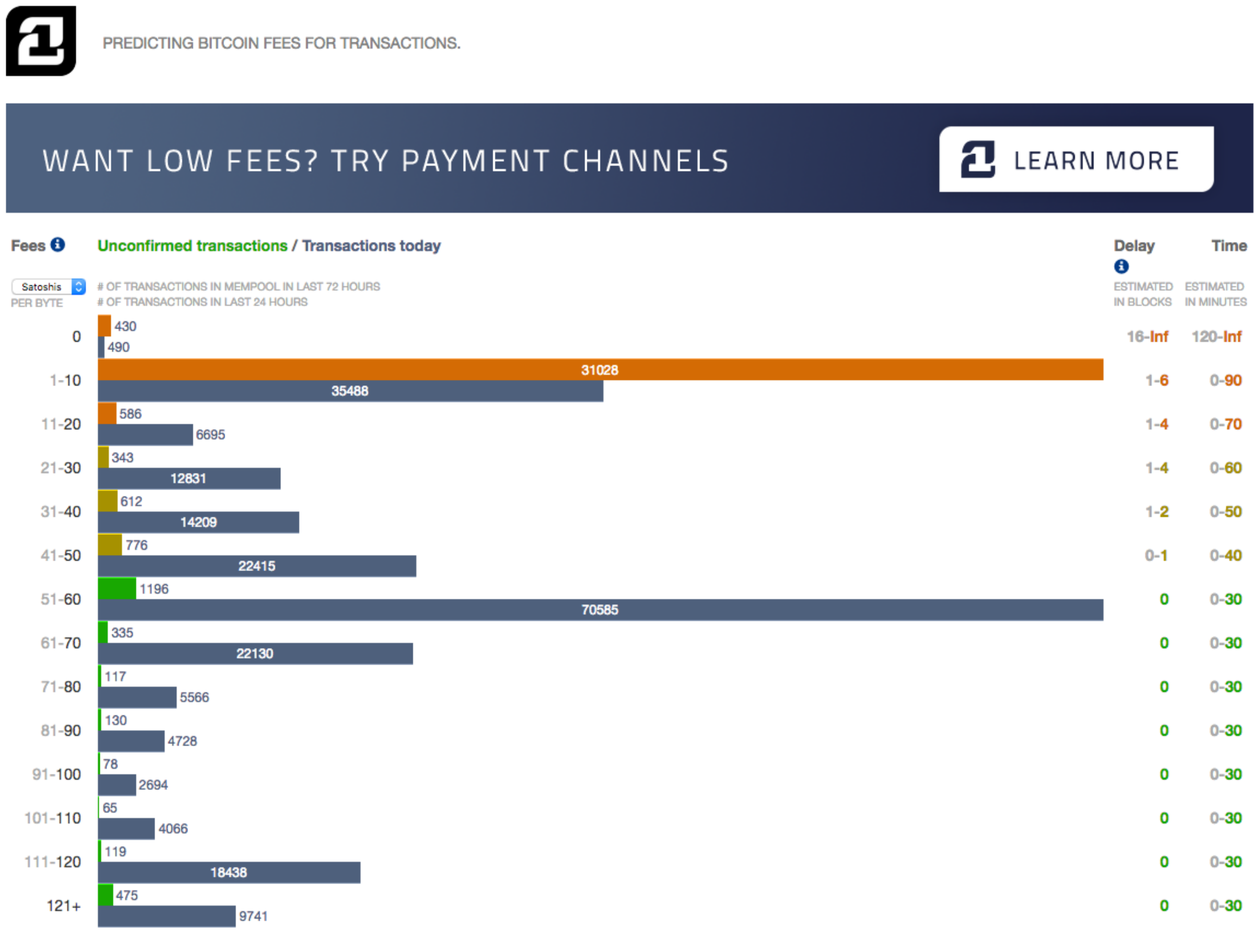
Block Reward :: Difficulty Adjustment



Block Reward :: Bitcoin Halving



Transaction Fees



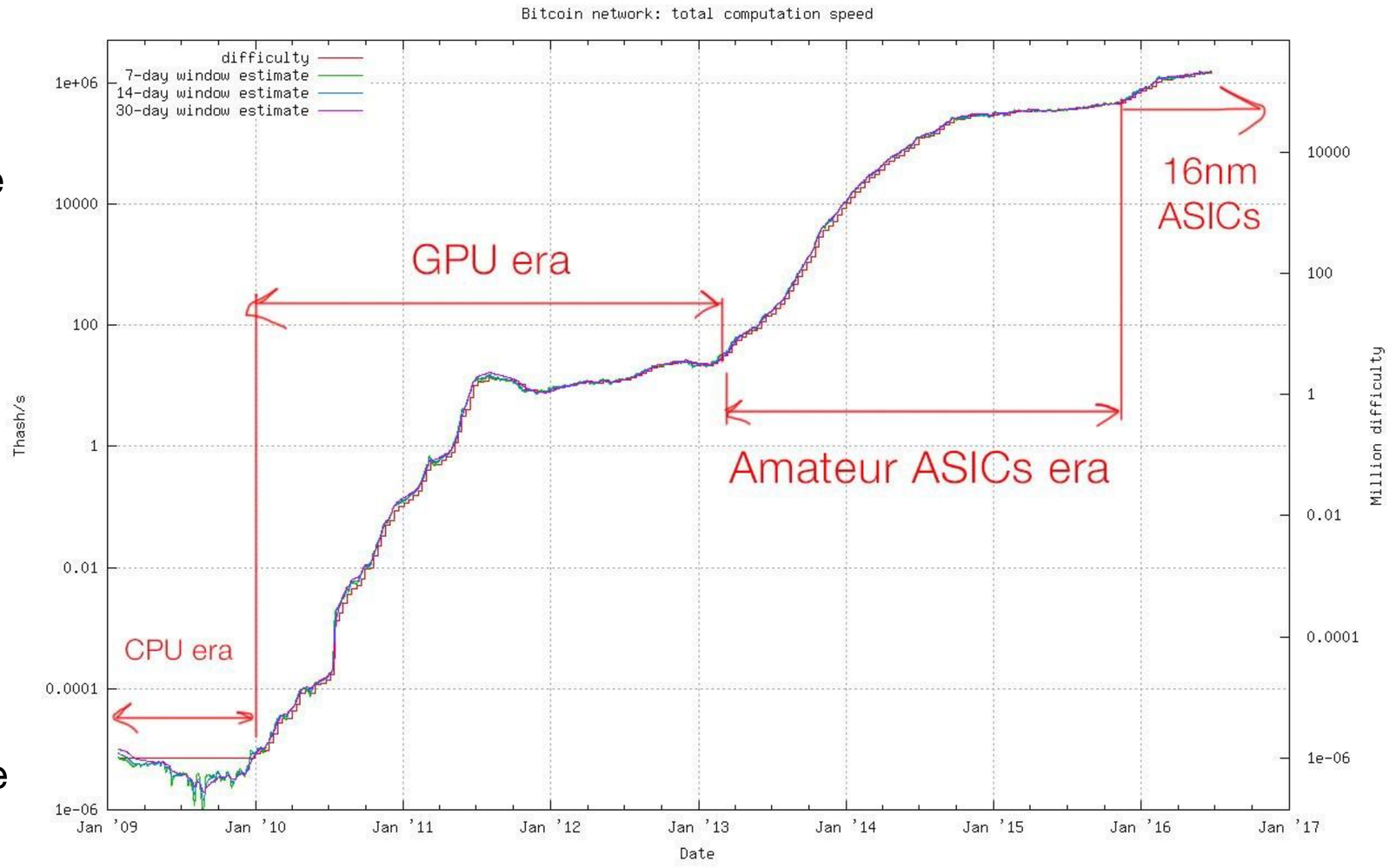
- Transaction fees are optional
- Miners tend to favor transactions with larger transaction fees
  - Incentive to include your transaction in the next block they're mining
- Primary source of revenue for miners after 2141

Hardware Cost

- It depends a lot on the kind of hardware
- It has evolved enormously since the beginning

  1. Normal PC
  2. Graphic cards
  3. FPGA and first ASICs
  4. High technology ASICs

- Nowadays only large pools of specialised computers can have real chances of finding blocks



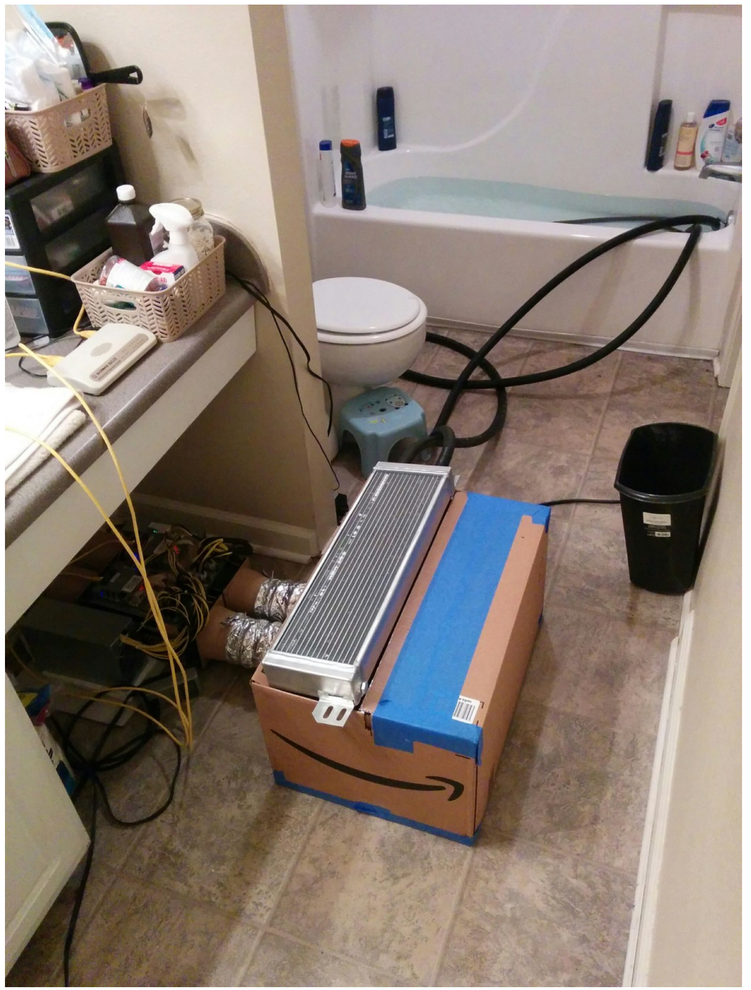
ASIC Mining

- Nowadays, mining with normal CPUs or GPUs or old ASICs is useless: lots of energy is consumed without getting any result
- Effective miners use ASICs, Application-Specific Integrated Circuits
  - VLSI circuits specifically designed to do one task: Bitcoin mining (SHA256)
  - Fastest miners around: can crunch 15 TH/s = 15*1012 H/s and more
  - Expensive
    - up $5000 each
    - but actually the price is better calculated in MH/s/$ which is around 2000-3000 MH/s per dollar
    - Consume lots of electricity: around 1.5-2 kW each, but it can be around 3.5 kW

What a Petahash miner looks like

Operating Costs

- Energy used to mine Bitcoin
  - Electricity
  - Cooling
- «A bitcoin is embodied energy»
- A miner is an electric heater that makes you money
- Bitcoin network energy usage is comparable to that of a small country
  - It is advantageous (=more profit) to run mining clusters where electricity is cheaper
  - Unfortunately, there energy is produced mostly from coal and oil, so there is a big carbon dioxide impact



Energy consumption (as of April 2019)

Description	Value
Bitcoin's current estimated annual electricity consumption* (TWh)	53.63
Bitcoin's current minimum annual electricity consumption** (TWh)	41.01
Annualized global mining revenues	$3,807,708,507
Annualized estimated global mining costs	$2,681,403,652
Current cost percentage	70%
Country closest to Bitcoin in terms of electricity consumption	Bangladesh
Total Network Hashrate in PH/s (1,000,000 GH/s)	50,761
Electricity consumed per transaction (KWh)	435
Number of U.S. households that could be powered by Bitcoin	4,965,562
Number of U.S. households powered for 1 day by the electricity consumed for a single transaction	14.72
Bitcoin's electricity consumption as a percentage of the world's electricity consumption	0,24%
Annual carbon footprint (kt of CO2)	25,473
Carbon footprint per transaction (kg of CO2)	206.85

https://digiconomist.net/bitcoin-energy-consumption

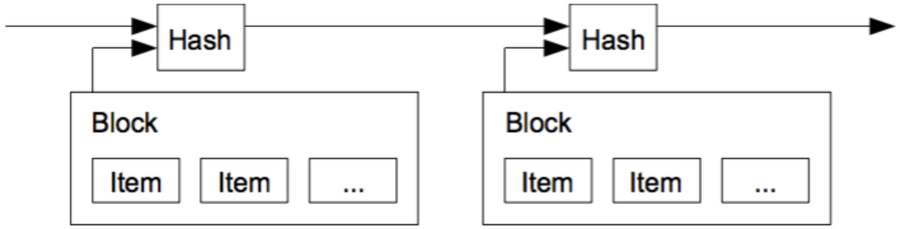
A back-on-the-envelope calculation

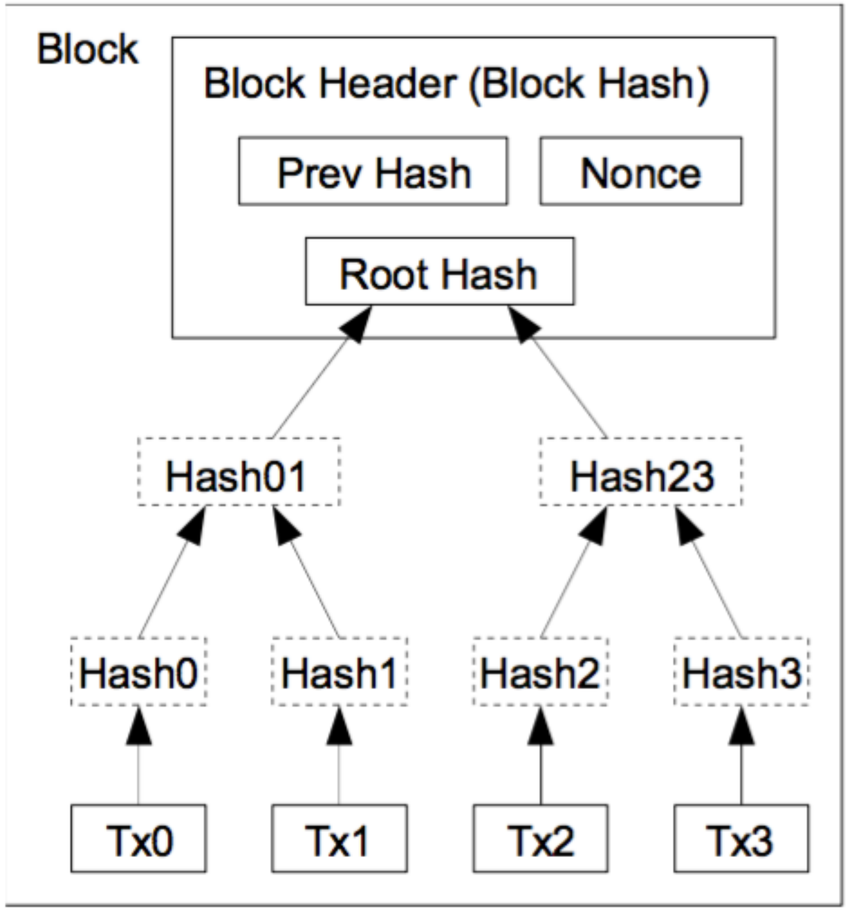
- Annualised estimated global mining costs: $2,681,403,652
- Number of blocks mined in a year: 365*24*6 = 52,560
- Number of bitcoins created per year: 54,560*12.5 = 657,000
- Production cost per bitcoin: $2,681,403,652/657000 = $4,080
- Average market price: 1 BTC = $5,014
- Revenue: $934 = 22,8%
- But of course it depends on several other factors, so YMMV.

Bringing it all together - Back to a transaction

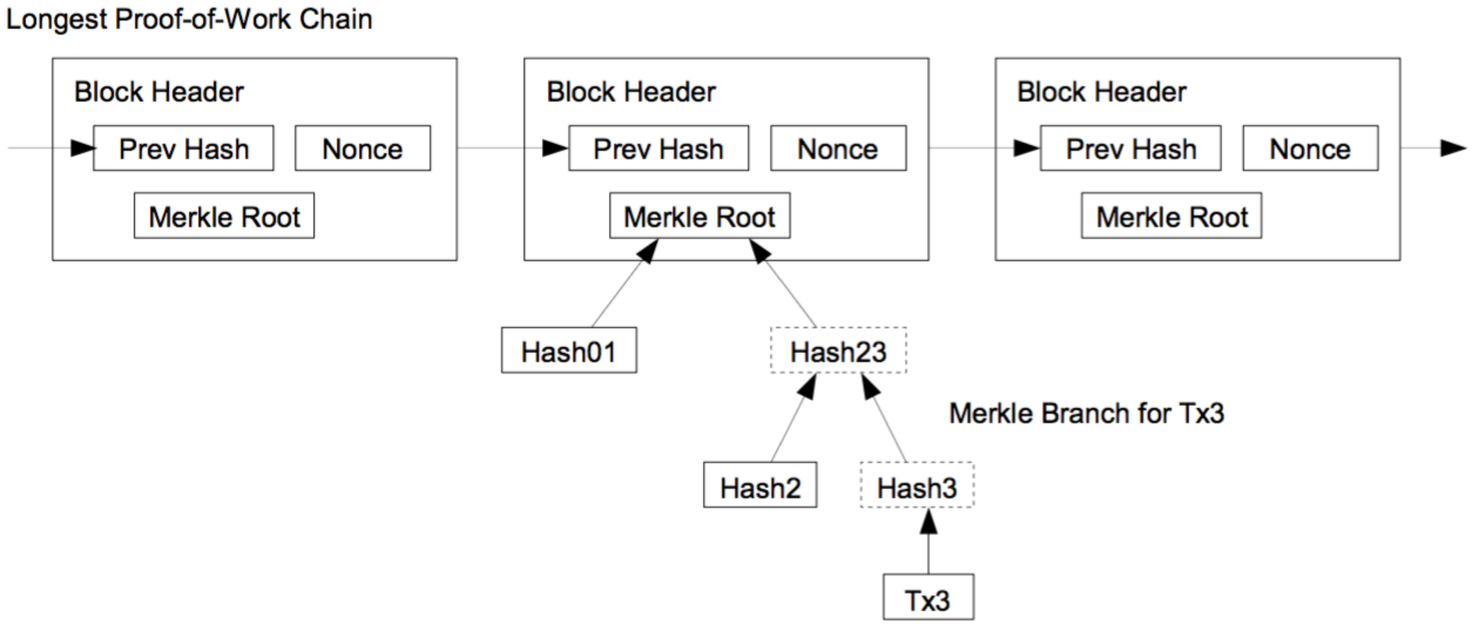
- I want to send money to Sunny
  - Sign transaction
  - Broadcast to network
- Miners receives transaction, adds to “**zero-conf pool**”
  - Verify transaction: i.e. signature matches, enough money, etc.
- Miner finds PoW, broadcasts block (with its reward inside)
  - Block propagates; others verify
- Miners work on the next problem

Bonus: Merkle Tree





Transactions Hashed in a Merkle Tree



- Makes transaction history immutable
- PoW to add chains

Source: Nakamoto 2009