Lecture 2

Bitcoin

Bitcoin Keys

private key → public key → Address

Bitcoin Keys

Private :
9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08

Public :
045f81956d5826bad7d30daed2b5c8c98e72046c1ec8323da3364454761 83fb7ca54ba511b8b782bc5085962412e8b9879496e3b60bebee7c36987d1d5848b9a50

Address :
1HKqKTMpBTZZ8H5zcqYEWYBaaWELrDEXeE

Bitcoin Keys

Live example…

Bitcoin Transactions

No sender, No Receiver,
Only Inputs and outputs

Transactions

Live example…

Sample Transaction

- Alice has **0.1 BTC** (she got that from Joe)
- She wants to buy a cup of coffee from Bob for **0.015 BTC**

Sample Transaction

- Alice will take the **0.1 BTC** she has and will send:
  - 0.0150 to Bob
  - 0.0845 to herself (the change)
  - 0.0005 as a transaction fee
- Total Transaction Input: 0.1 BTC
- Total Transaction Output: 0.1 BTC
- Let us look at this transaction in more detail:

Transactions - Behind the Scenes


```
{
  "version": 1,
  "locktime": 0,
  "vin": [
    {
      "txid": "7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
      "vout": 0,
      "scriptSig" : "3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f0
      "sequence": 4294967295
    }
  ],
  "vout": [
    {
      "value": 0.01500000,
      "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHEC
    },
    {
      "value": 0.08450000,
      "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHEC
    }
  ]
}
```


This is the actual Transaction !!

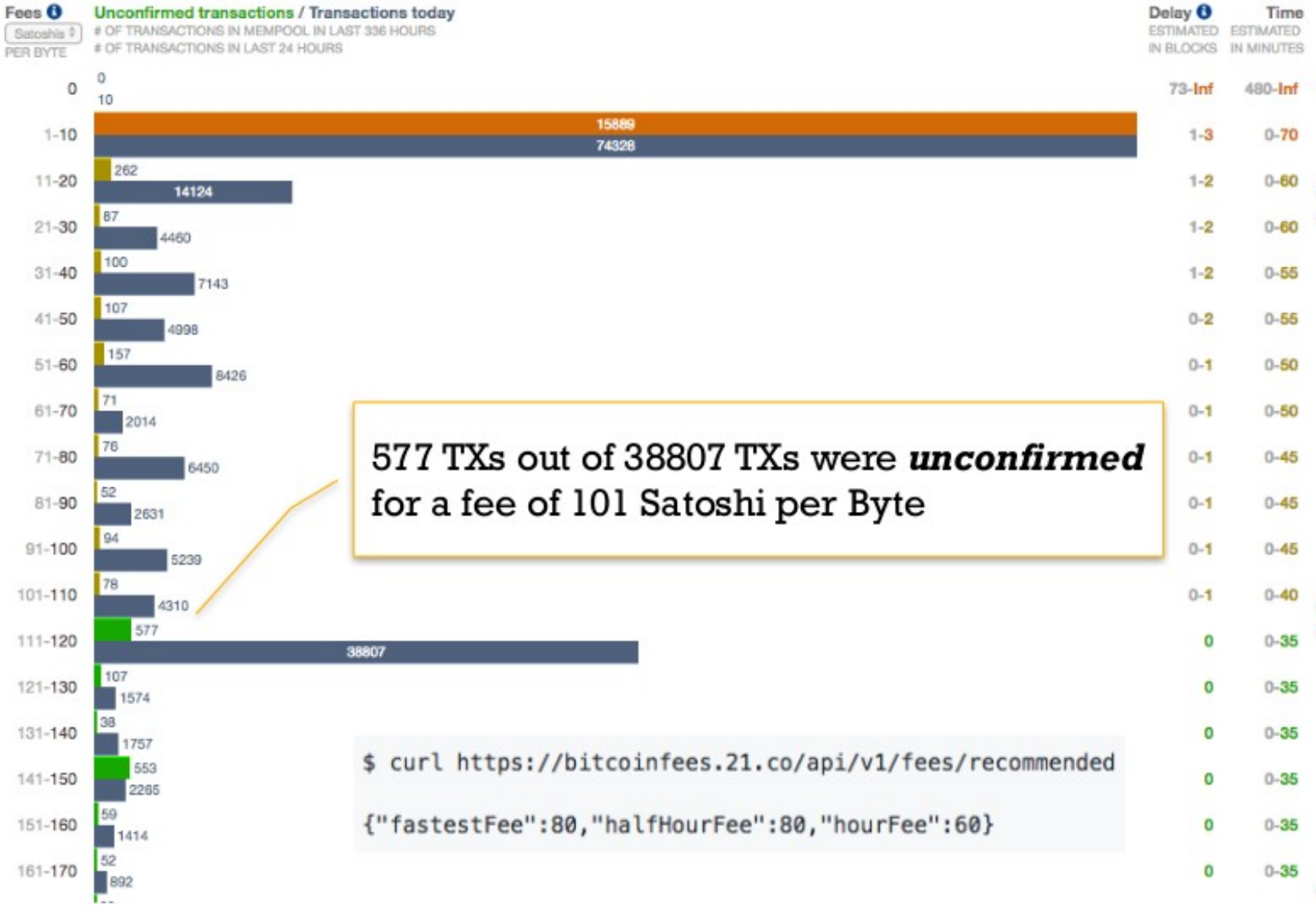
No sender, No Receiver, No Fees!!!

Only Inputs and outputs, lets' take them one by one.

Transaction Fees

- Who decides the fee?
  - Supply and demand
  - The issuer of the TX proposes a fee
  - The miners (more in ch10) *might* be interested to mine it (record it in the blockchain)
  - The TX size in bytes is also a factor, for the same fee a larger TX might be dumped
- What happens to dumped transactions?
  - They can take longer to process
  - They might stall forever if the fee is too low.

Transaction Fee Estimation



577 TXs out of 38807 TXs were ***unconfirmed*** for a fee of 101 Satoshi per Byte

```
$ curl https://bitcoinfees.21.co/api/v1/fees/recommended

{"fastestFee":80,"halfHourFee":80,"hourFee":60}
```

~10 min at best

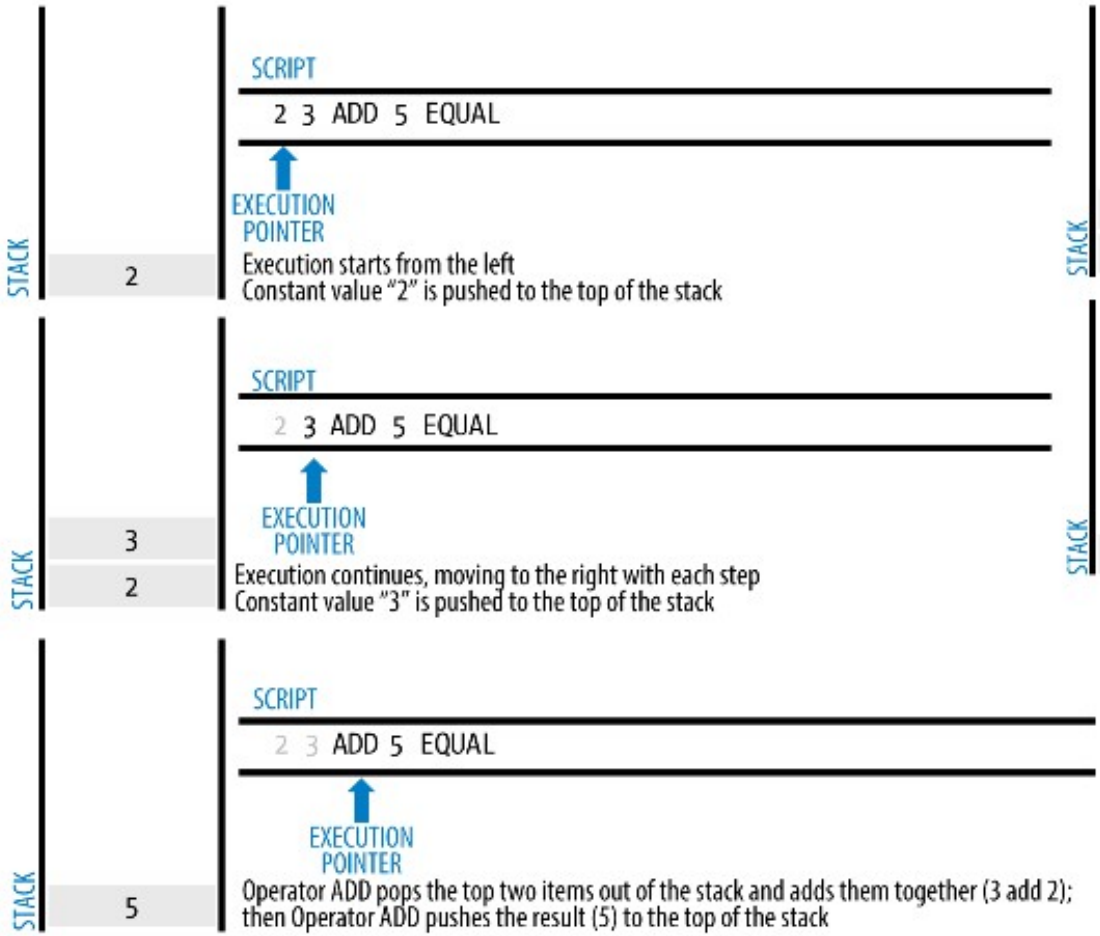
Transactions

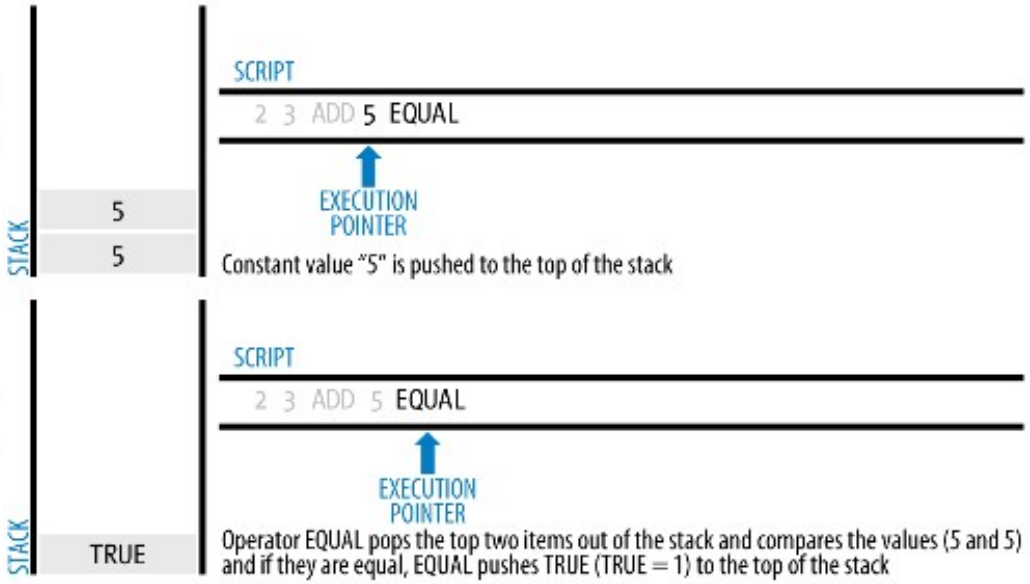
The “Script” Language

Script Construction (Lock + Unlock)

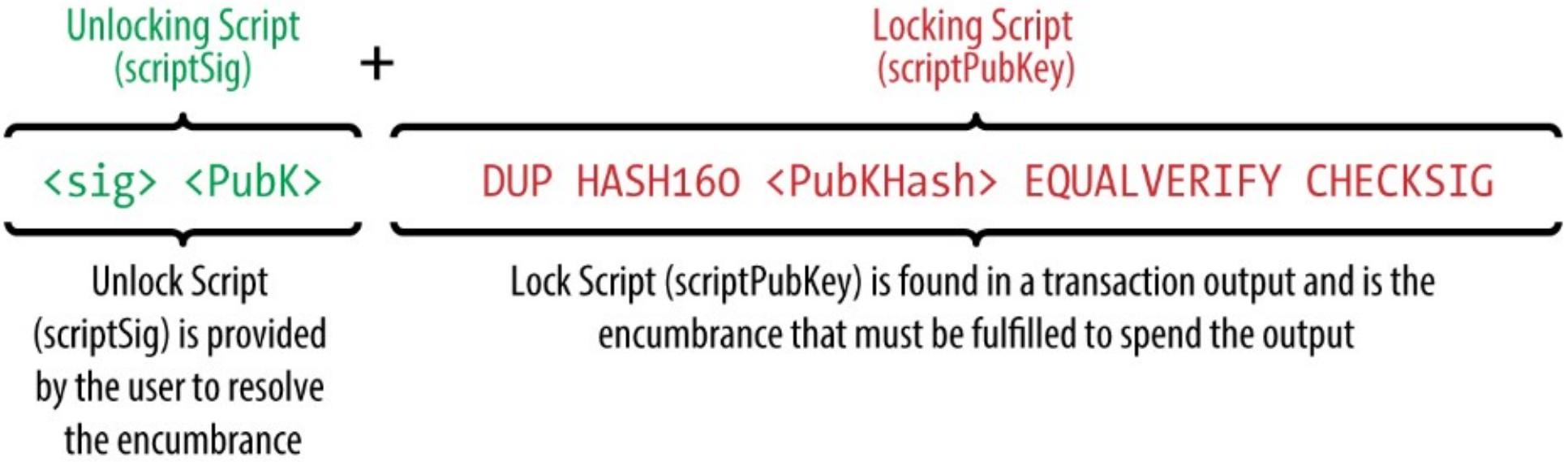
Locking Script: **3 OP_ADD 5 OP_EQUAL**

Unlocking Script: **2**





P2PKH Script



More Advanced Topics

- Security issues with P2SH
- Timelocks & nSequence (relative time)
- Scripts with flow control (if/then/else)
- Segregated Witness, P2WPKH, **P2WSH**

```
IF
  IF
    2
  ELSE
    <30 days> CHECKSEQUENCEVERIFY DROP
    <Abdul the Lawyer's Pubkey> CHECKSIGVERIFY
    1
  ENDIF
  <Mohammed's Pubkey> <Saeed's Pubkey> <Zaira's Pubkey> 3 CHECKMULTISIG
ELSE
  <90 days> CHECKSEQUENCEVERIFY DROP
  <Abdul the Lawyer's Pubkey> CHECKSIG
ENDIF
```

Bitcoin Blockchain

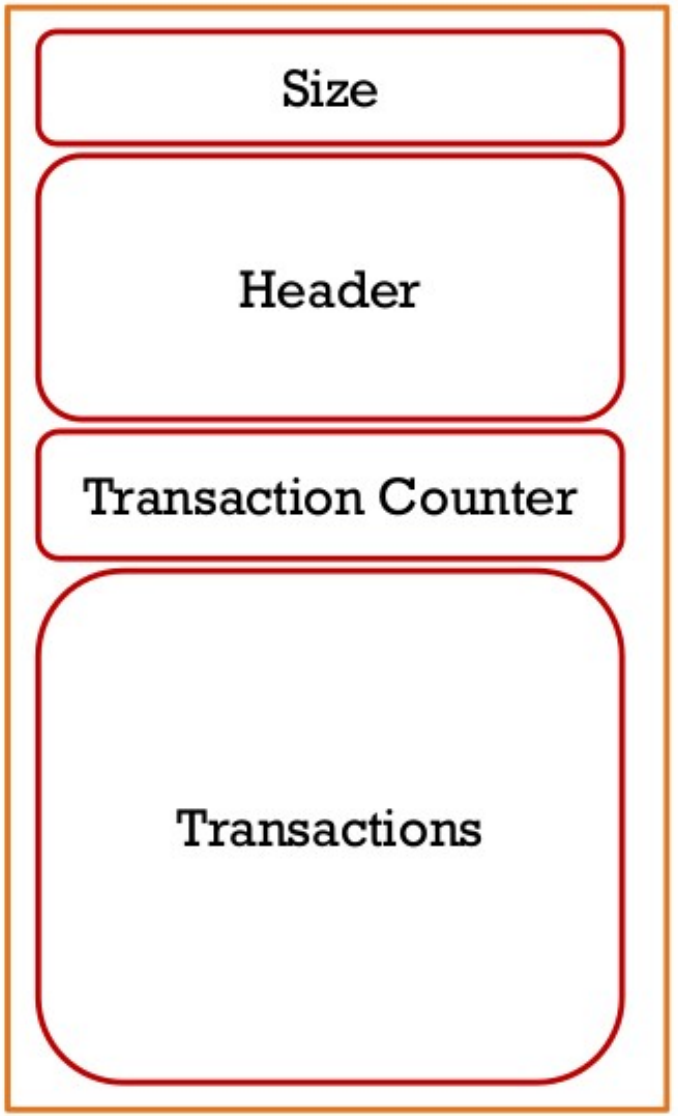
let me explain…

Bitcoin Blockchain

The Block is
A collection of trancations

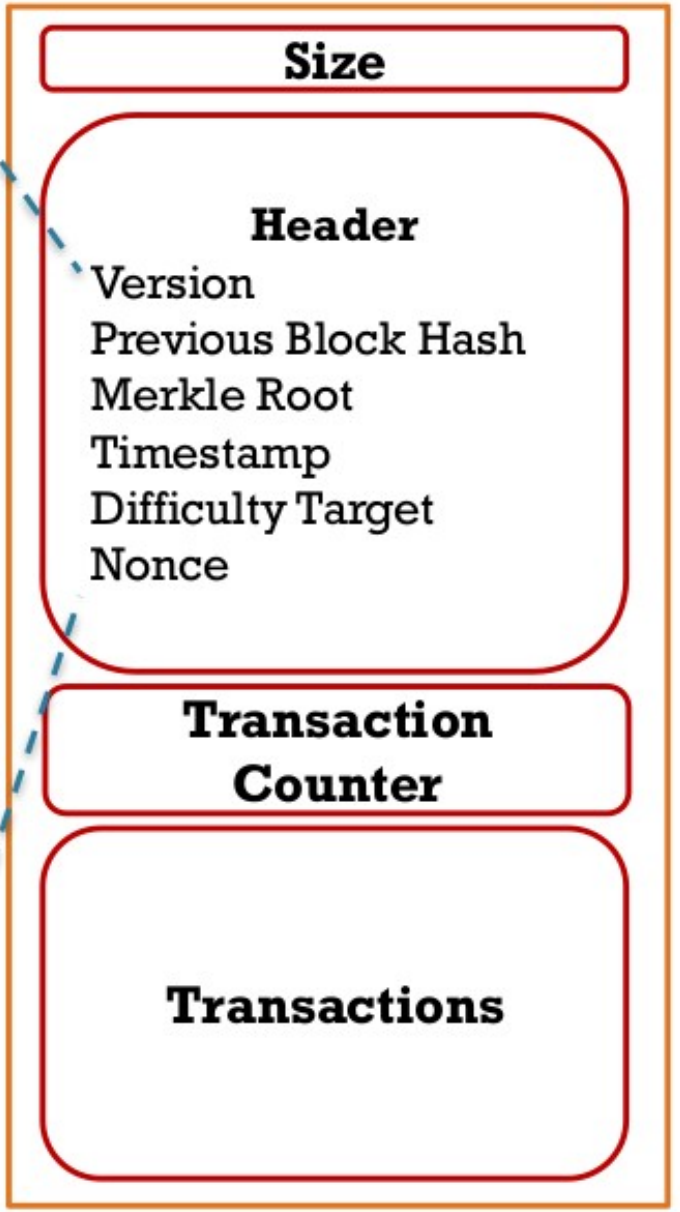
Structure of a Block

Size	Field	Description
4 bytes	Block Size	size of block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1–9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block



Block Header

# Bytes	Field	Description
4	**Version**	A version number to track software/protocol upgrades
32	**Previous Block Hash**	A reference to the hash of the previous (parent) block in the chain
32	**Merkle Root**	A hash of the root of the merkle tree of this block's transactions
4	**Timestamp**	The approximate creation time of this block (seconds from Unix Epoch)
4	**Difficulty Target**	The Proof-of-Work algorithm difficulty target for this block
4	**Nonce**	A counter used for the Proof-of-Work algorithm



Block Identifiers:
Block Header Hash & Block Height

Block Header Hash

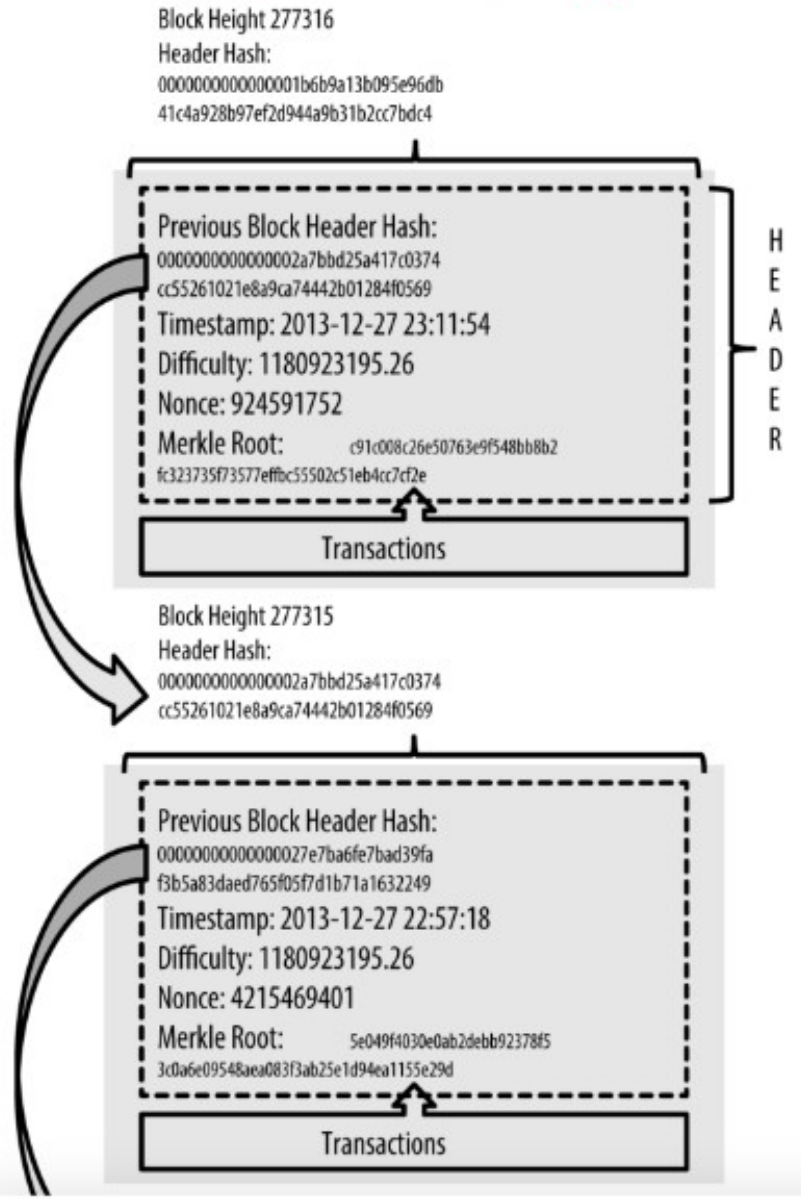
- Not written anywhere inside a block or its header
- Appears only in the "parent/previous" field of its child

Block Height

- Can be used to identify a block
- Also Not written anywhere

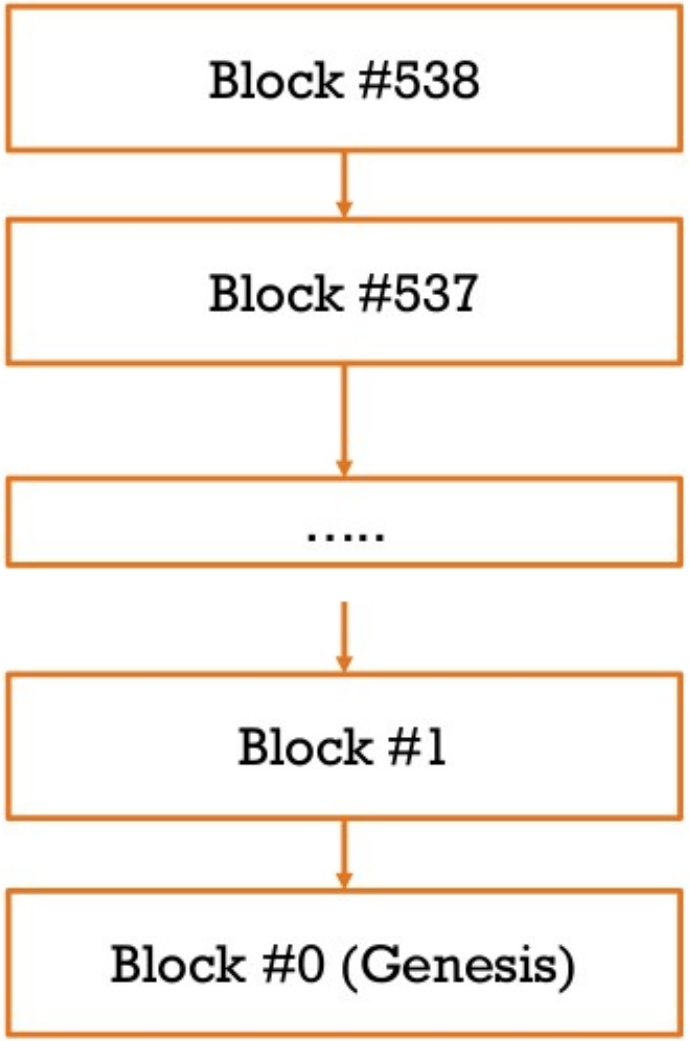
Uniqueness

- When 2 blocks compete for the tip of the blockchain
  - Hash is a unique id of the block
  - Height is not



The Blockchain Data Structure

- Blocks refer to the previous block in the chain.
- "**height**" refers to distance from first block,
- "**tip**"/"head" refer to the most recently added block.
- "Genesis" block is the first block ever created i.e. #0



Bitcoin

Mining

Mining

Just Creating New **BLOCK**

Bitcoin Mining

Sameh El-Ansary, PhD

Bitcoin Economics and Currency Creation

Date	Mined Block Reward
2009	50 BTC
2012	25 BTC
2016	12.5 BTC
	…
2140	1 Satoshi (10^{-8} BTC)

```
$ python max_money.py
Total BTC to ever be created: 2099999997690000 Satoshis
```

Mining

- Mining *secures the bitcoin system* and enables the emergence of network-wide *consensus without a central authority*.
- The reward of newly minted coins and transaction fees is an incentive scheme that aligns the actions of miners with the security of the network, while simultaneously implementing the monetary supply

1. Verification of Transactions

1. To spend coin, the wallet software:
   a. Collects UTXO(s) as input
   b. Unlocks the UTXO(s) using private keys
   c. Creates new UTXO(s) spendable by others, (possibly incl. sender)
   d. Puts inputs and outs in a new TX
2. The wallet sends the new TX to a bitcoin node (SPV) or locally (full-node)
3. The receiving node verifies the transaction
4. Sends to all neighbors if valid
5. The process continues recursively

Transaction Verification Rules

- Format correct.
- Inputs and outputs not empty.
- 100< Size in bytes < MAX_BLOCK_SIZE.
- Each vout,& sum(vout) < less than 21m coins and > *dust* threshold
- Each vin & sum(vin) < less than 21m coins and > *dust* threshold
- A matching transaction in the pool, or in a block in the main branch, must exist.
- For each input:
  - if the referenced vin exists in any other transaction in the pool, the transaction must be rejected.
  - look in the main branch and the transaction pool to find the referenced output transaction. If the output transaction is missing for any input, this will be an orphan transaction. Add to the orphan transactions pool, if a matching transaction is not already in the pool.
  - if the referenced output transaction is a coinbase output, it must have at least COINBASE_MATURITY (100) confirmations.
  - the referenced output must exist and cannot already be spent.
  - The unlocking scripts for each input must validate against the corresponding output locking scripts.
- Sum(vin) > sum(vout).
- Transaction Fee > minRelayTxFee.
- None of the inputs have hash=0, N=–1 (coinbase transactions should not be relayed).
- nLocktime is equal to INT_MAX, or nLocktime and nSequence values are satisfied according to MedianTimePast.
- The number of signature operations (SIGOPS) contained in the transaction is less than the signature operation limit.
- The unlocking script (scriptSig) can only push numbers on the stack, and the locking script (scriptPubkey) must match IsStandard forms (this rejects "nonstandard" transactions).

2. Aggregating Txs into Blocks

- After validating transactions, a bitcoin node will add them to the *memory pool*, or *transaction pool*, where transactions await until they can be included (mined) into a block
- If the node is a miner:
  - Adds a coinbase Tx
  - Adds a merkle root
  - Solves the PoW puzzle

Coinbase Transaction


```
{
    "hex" : "0100000001000000000000000000000000000000000000000000000000000000000000000000ffffffff0f03443b0
    "txid" : "d5ada064c6417ca25c4308bd158c34b77e1c0eca2a73cda16c737e7424afba2f",
    "version" : 1,
    "locktime" : 0,
    "vin" : [
        {
            "coinbase" : "03443b0403858402062f503253482f",
            "sequence" : 4294967295
        }
    ],
    "vout" : [
        {
            "value" : 25.09094928,
            "n" : 0,
            "scriptPubKey" : {
                "asm" : "02aa970c592640d19de03ff6f329d6fd2eecb023263b9ba5d1b81c29b523da8b21OP_CHECKSIG",
                "hex" : "2102aa970c592640d19de03ff6f329d6fd2eecb023263b9ba5d1b81c29b523da8b21ac",
                "reqSigs" : 1,
                "type" : "pubkey",
                "addresses" : [
                    "1MxTkeEP2PmHSMze5tUZ1hAV3YTKu2Gh1N"
                ]
            }
        }
    ]
}
```


Coinbase Data

Block Reward +Fees

Address of the miner

Mining the Block

- Given that all miners want the reward
- We need a way to make sure only 1 single miner wins
- The technique for solving that is know as Proof of Work
- If a miner shows it can solve a puzzle of a certain **difficulty target** *before* others, she wins the block

Proof Of Work

- The miner needs to adds a nonce to the block
- The miner keeps changing the nonce until the **hash of the block is a number less that the target**
- With the example target, a single miner processing 1 trillion hashes per second (1 terahash per second or 1 TH/sec) would only find a solution once every 8,496 blocks or once every 59 days, on average

Bitcoin Mining

Live example…

Retargeting to Adjust Difficulty

- Q: Difficulty controls the time needed for successfully mining a block, so who sets it?
- A: The nodes have been designed so the whole network collaboratively adjusts the difficulty such that with one purpose, controlling the supply of new coin
  - Practically, for Satoshi's design to work, a block has to be mined every 10 minutes on average
- Ok, but how?

Decentralized Difficult Retargeting

- Every 2,016 blocks, each nodes retargets the Proof-of-Work
- New Target = Old Target * (Actual Time of Last 2016 Blocks / 20160 minutes)
- To avoid volatility, the adjustment is never more than a factor of 4 up or down
- Now solely based on cost of electricity
- What if a node does not follow the rules? Her block will be rejected by others who adjusted

Successfully Mining the Block

- One a solution is found, the miner sends the block to all peers
- Peers validate the block and if indeed successful,
  - Non-miners pass it on
  - Miners cease their mining and immediately take on the next block

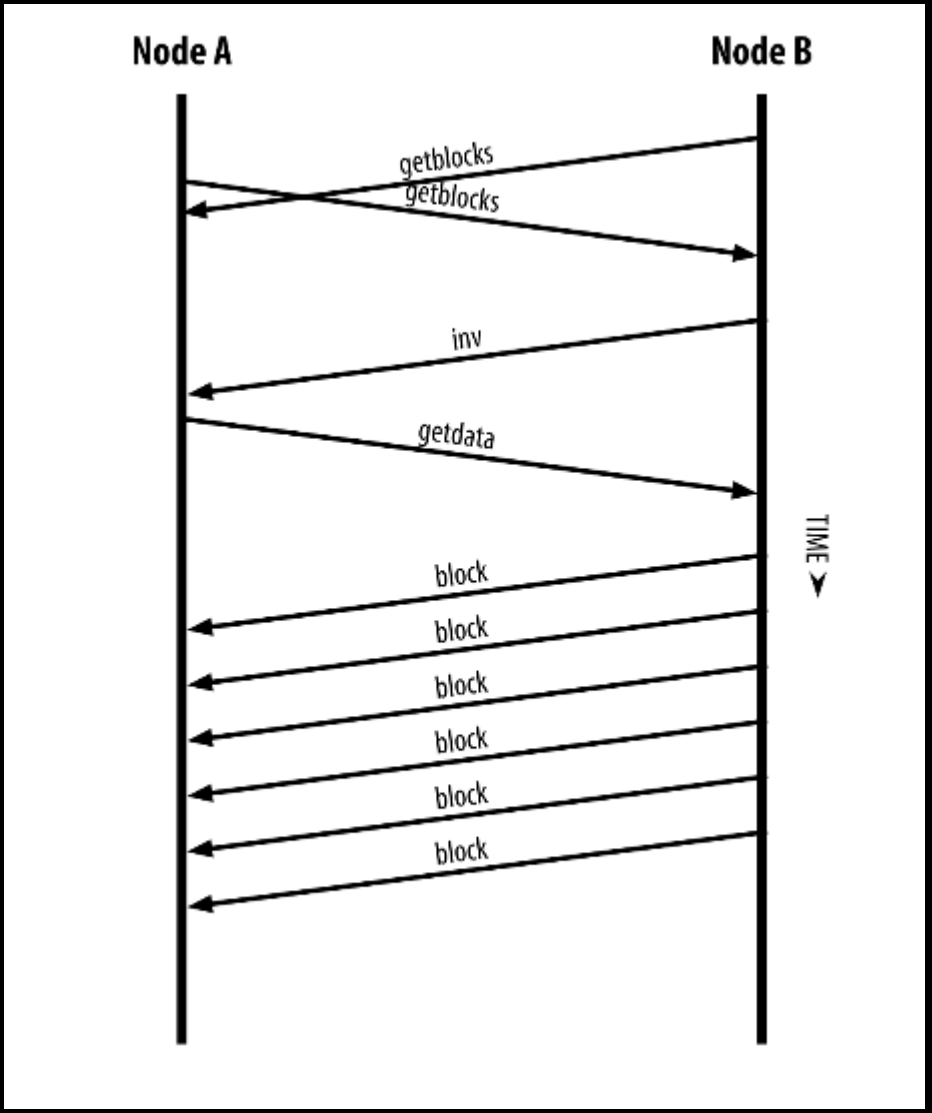
Validating a New Block

- The block data structure is syntactically valid
- The block header hash is less than the target (enforces the Proof-of-Work)
- The block timestamp is less than two hours in the future (allowing for time errors)
- The block size is within acceptable limits
- The first transaction (and only the first) is a coinbase transaction
- All transactions within the block are valid using the transaction checklist discussed earlier

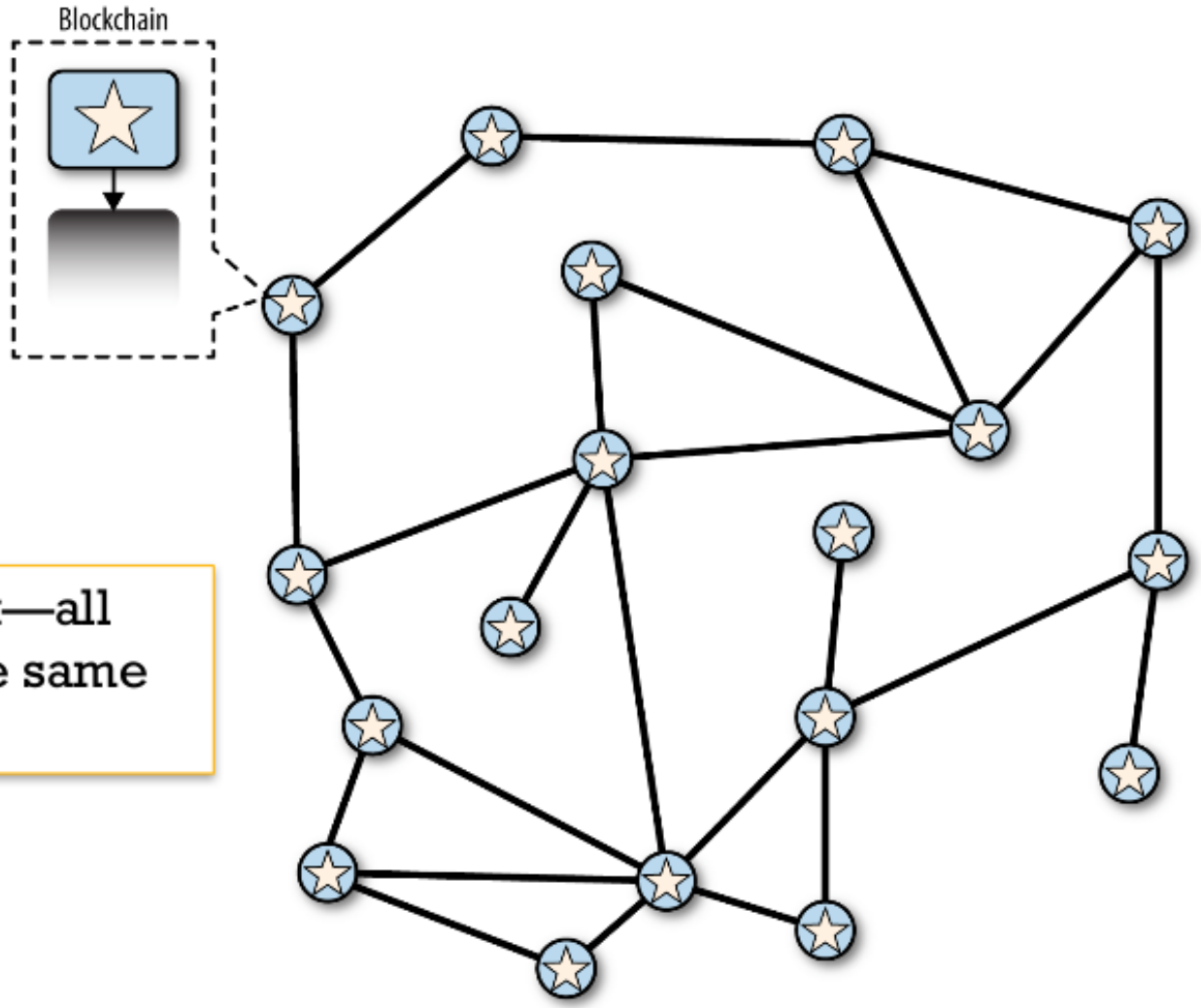
Bitcoin

P2P Network

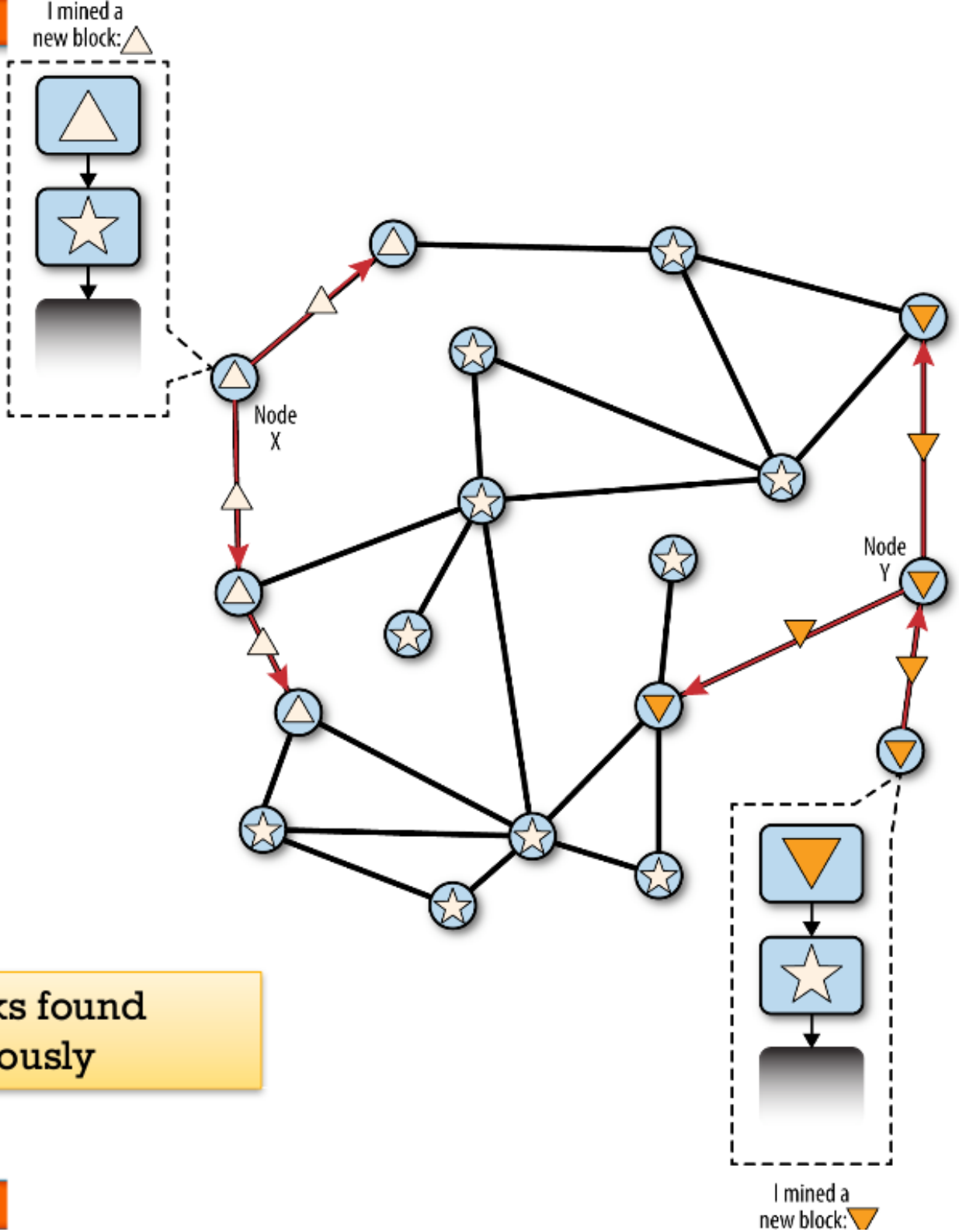
P2P Network


Node A
Node B
getblocks
getblocks
inv
getdata
block
block
block
block
block
block
TIME

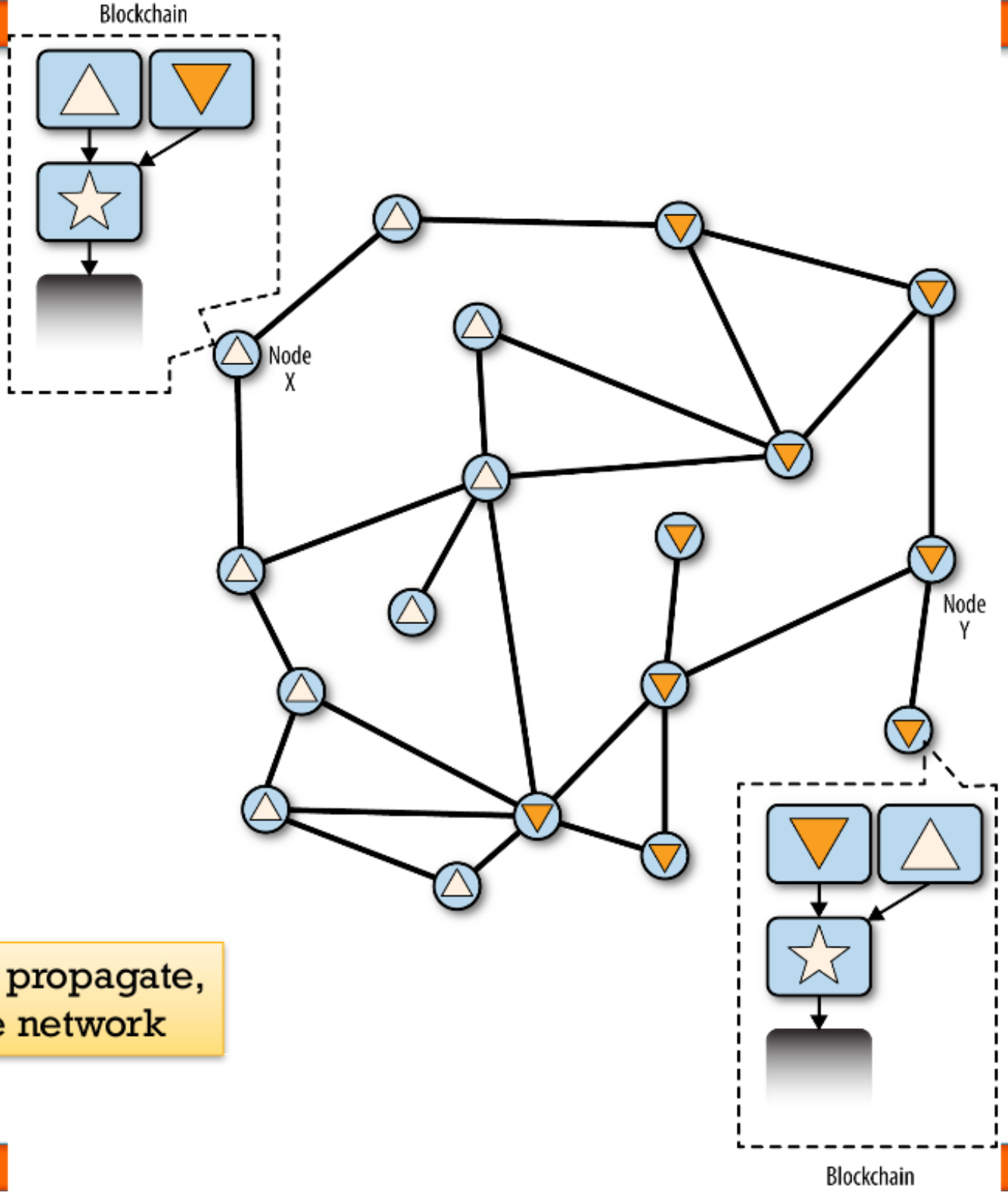
Blockchain Forks



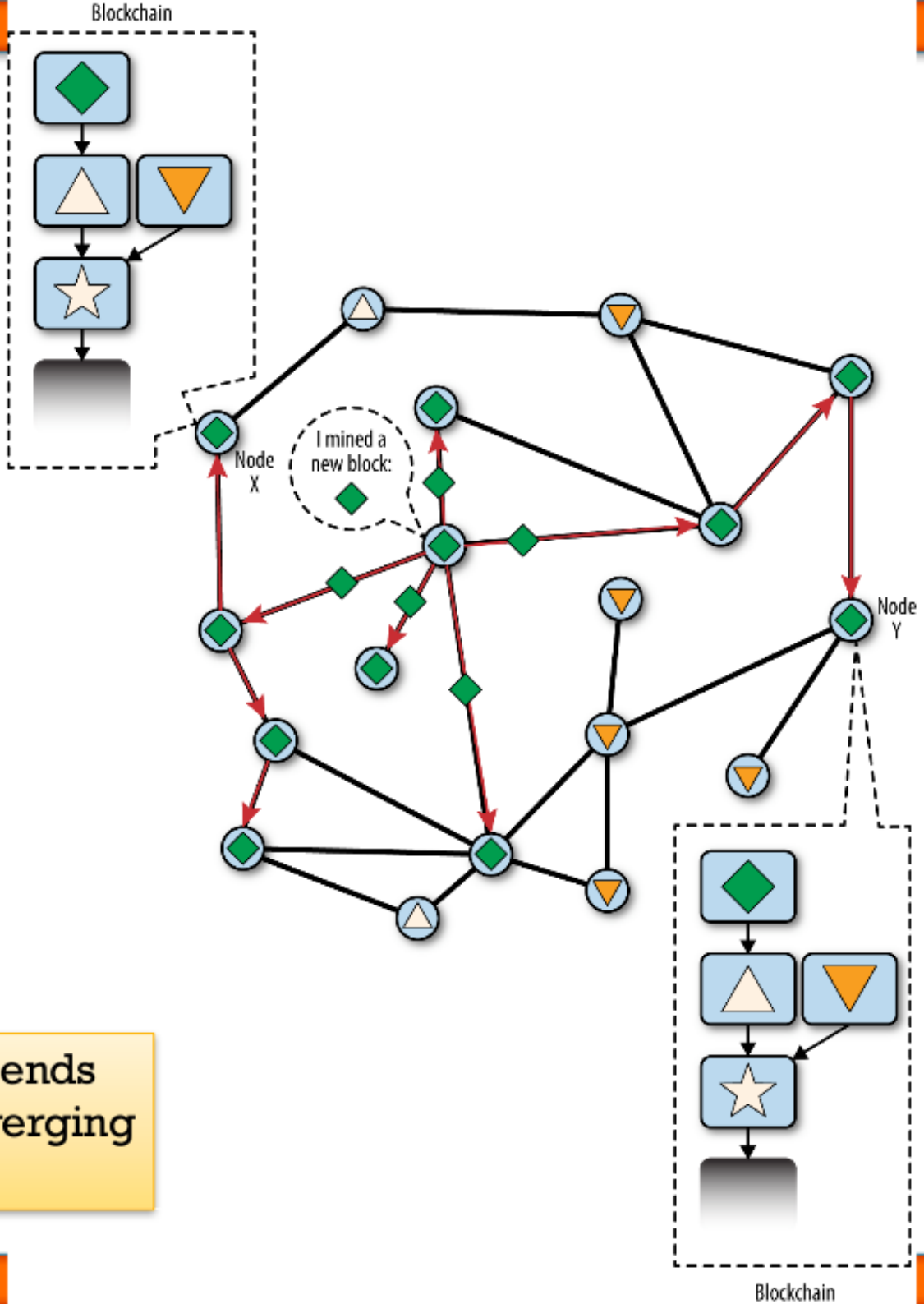
Before the fork—all nodes have the same perspective



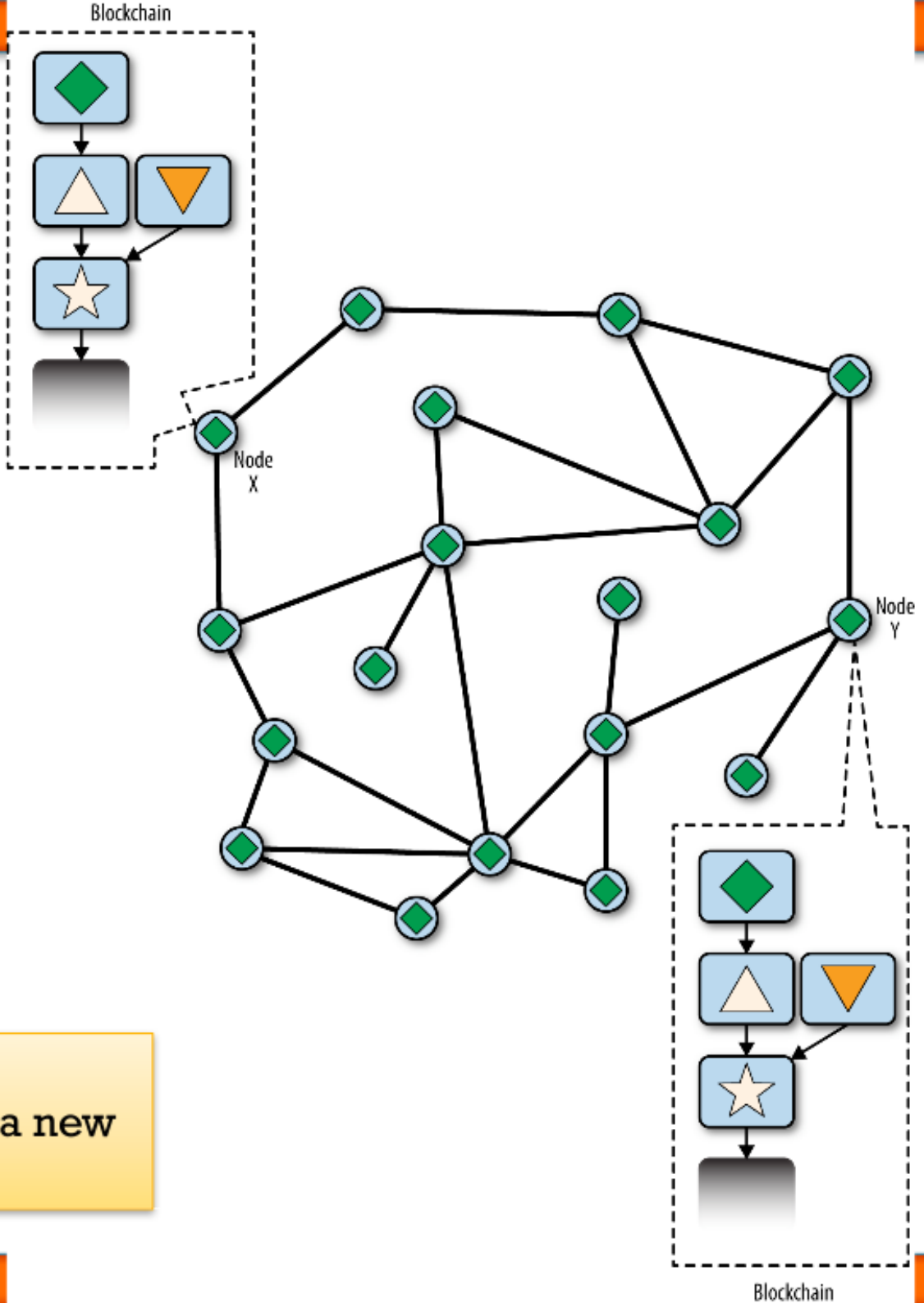
Two blocks found simultaneously



Two blocks propagate, splitting the network



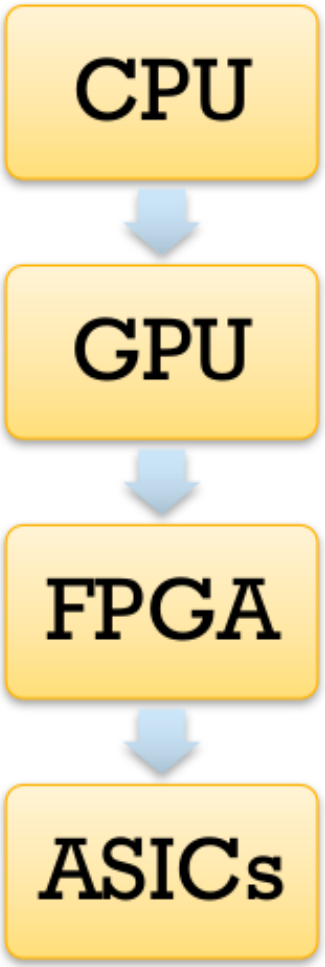
A new block extends one fork, reconverging the network



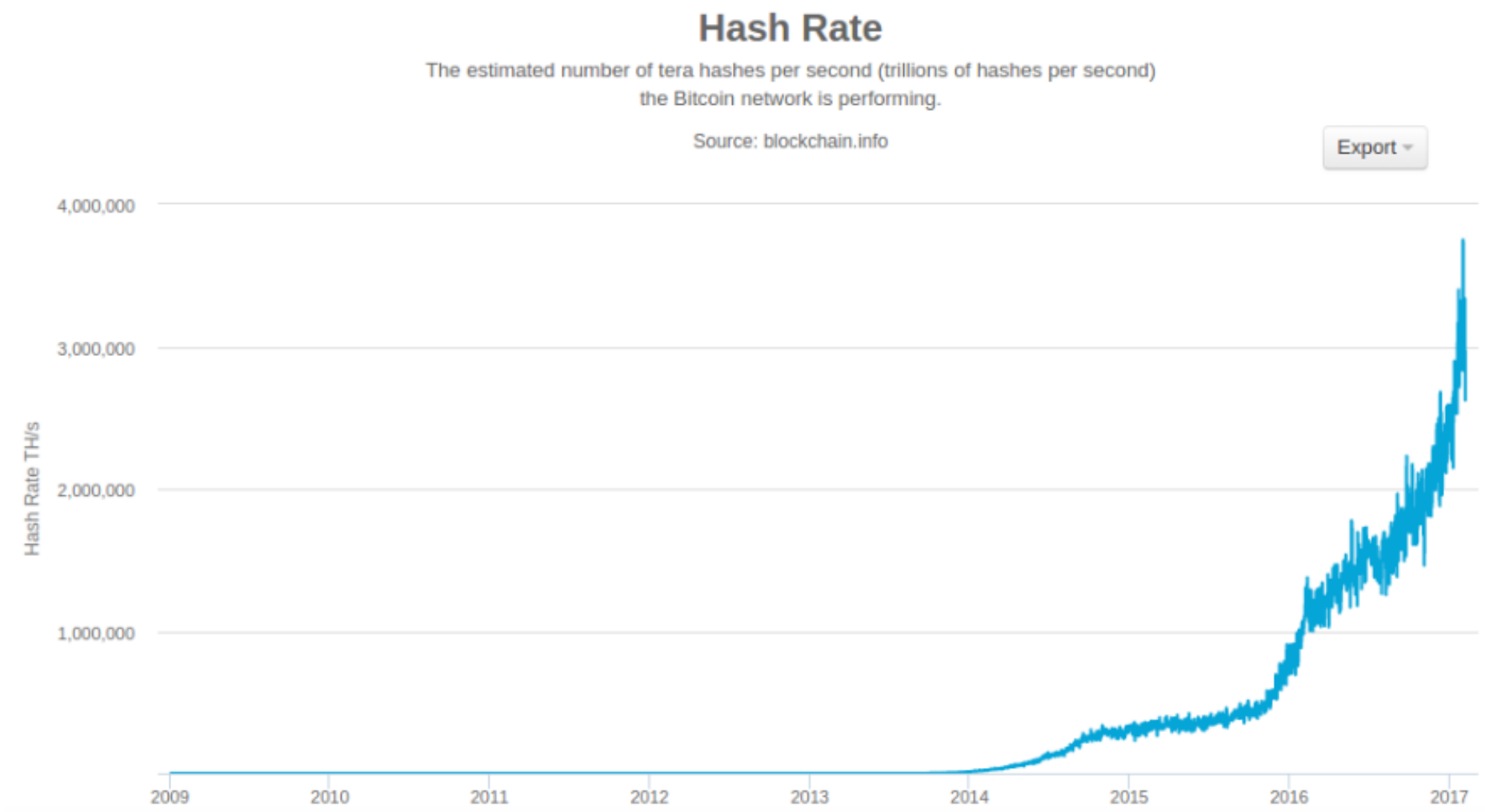
The network reconverges on a new longest chain

Hashing Power Evolution

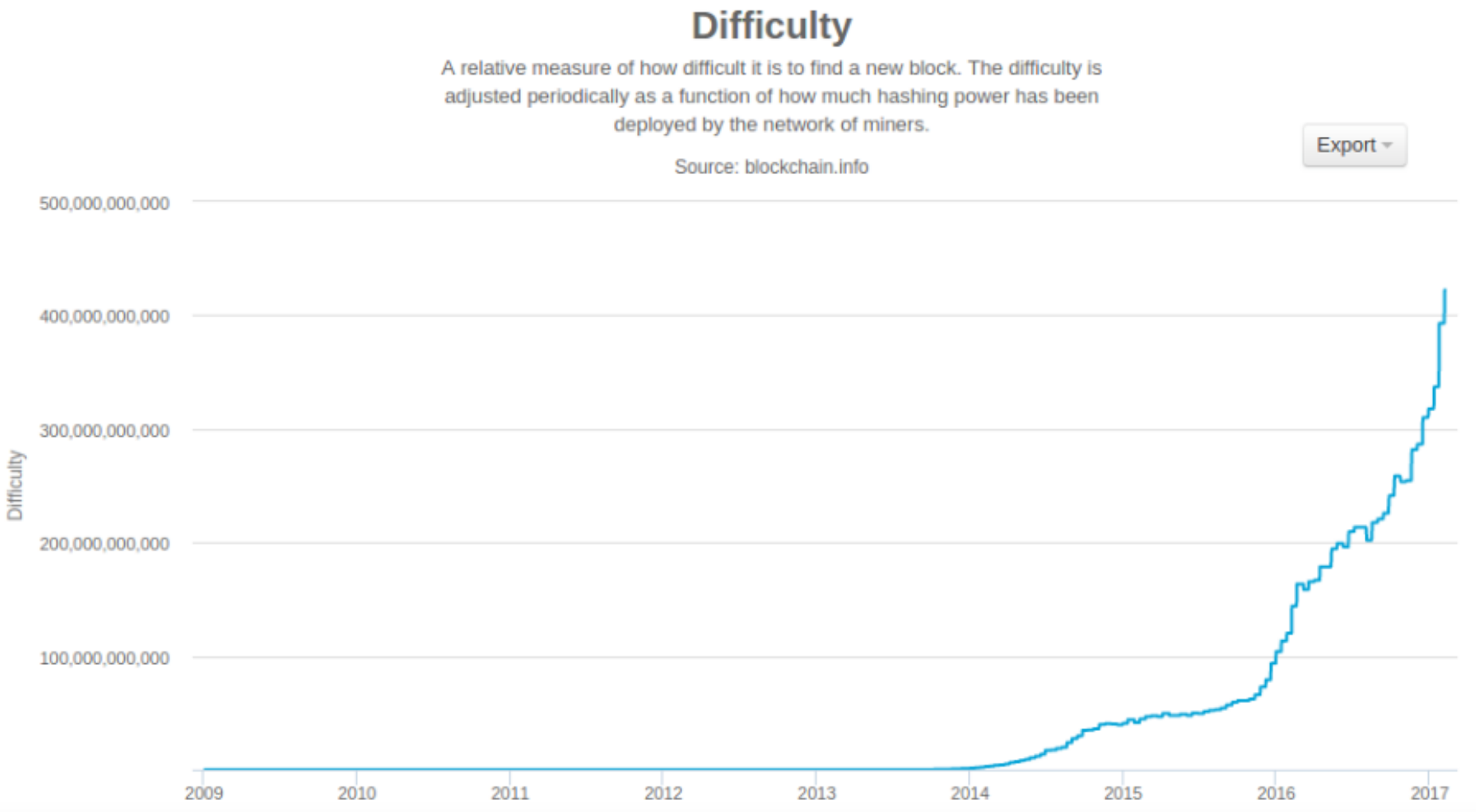
- The aggregate hashing power of the Bitcoin network
- 2009
  - 0.5 MH/sec–8 MH/sec (16× growth)
- 2010
  - 8 MH/sec–116 GH/sec (14,500× growth)
- 2011
  - 116 GH/sec–9 TH/sec (78× growth)
- 2012
  - 9 TH/sec–23 TH/sec (2.5× growth)
- 2013
  - 23 TH/sec–10 PH/sec (450× growth)
- 2014
  - 10 PH/sec–300 PH/sec (30× growth)
- 2015
  - 300 PH/sec-800 PH/sec (2.66× growth)
- 2016
  - 800 PH/sec-2.5 EH/sec (3.12× growth)



Hash Rate Evolution



Difficulty Evolution



Mining Pools

Name	Location	Size[1]	Merged Mining[2]	Reward Type	Transaction fees
AntPool	China	Large	No	PPLNS & PPS	kept by pool
BTC.com		Medium	NMC	FPPS	shared
BCMonster.com		Small	No	PPLNS	shared
Jonny Bravo's Mining Emporium		Small	No	PPLNS	shared
BitcoinAffiliateNetwork		?	NMC, DOGE	?	kept by pool
Slush Pool	Global	Medium	NMC	Score	shared
BitMinter		Small	NMC	PPLNSG	shared
BTCC Pool	China, Japan	Large	NMC	PPS	kept by pool

Solo Miner is history

Avoiding mutual fraud, by PoW evidence

Mining pools only can win by crowdsourcing hashpower and sharing profits

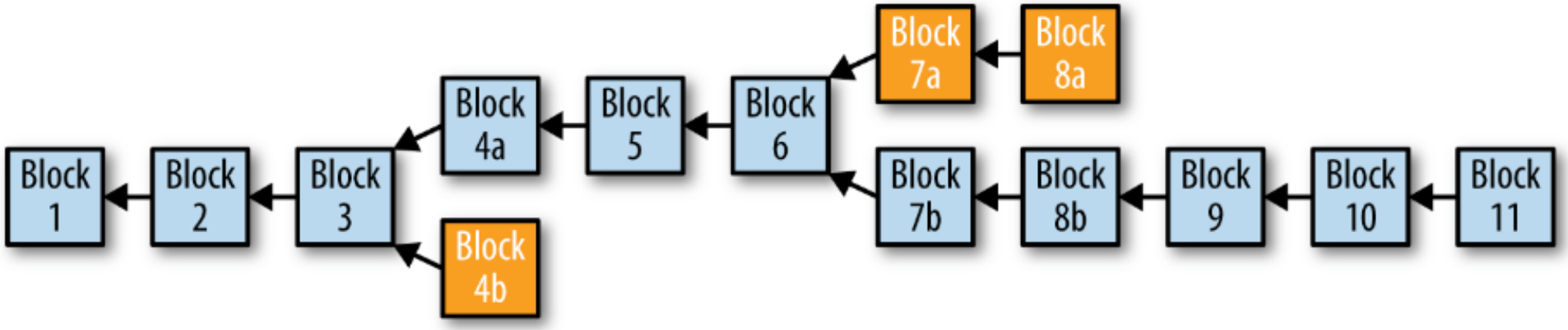
https://en.bitcoin.it/wiki/Comparison_of_mining_pools

The 51% attach

- Buy something really $$$ and pay
- The seller finds one/two confirmations gives you the goods
- Call the a group of colluding mining pools, let them quickly produce an alternate 2 blocks double spending your UTXOs to something else
- You might not need even 51%
- Practically very hard
- Evidence of pool collusion exists

Hard Forks

- Split that is never re-converged
- Due to software change bug/intentional
- Bitcoin XT, Bitcoin Classic, Bitcoin unlimited were attempts to change the bitcoin rules, but never gain wide acceptance



Soft Fork

- Not really a fork
- Rules are the same
- Extra data is inserted in the transactions to signal new meanings
- Critiqued as confusion, technical debt, irreversible upgrades,etc.

Soft Fork Signaling with Block Version

- Uses "Block version" inside blockdata
- Developer announces the change and starts generating a higher "block version"
- Agreeing nodes, would start using the new software and produce new version
- New software must accept old versions for a while, until 95% of the network has changes, then nodes start to refuse the old version