

Bitcoin

Precursor #1: Ledgers

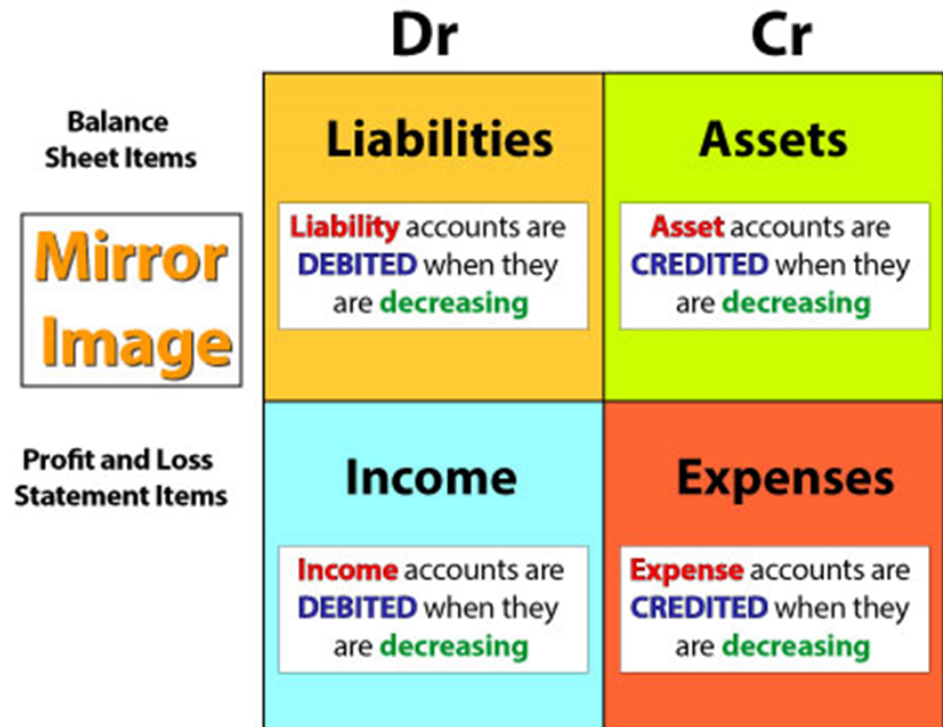
Ledgers

- At the beginning of written history (~3000 BC, Mesopotamia)
 - Believed to be used to record barley transactions, and payments
 - Reduces errors to make system more trustworthy
 - Recorded on papyrus scrolls or clay



Double-entry book-keeping

- Managing accounts so that any debit has an equal and offsetting credit amount.
 - Pacioli, da Vinci circa 1494 as monetary systems begin to take hold in Europe
 - Ensures integrity of ledger and keeps it from an invalid state
- Parts
 - Original records (transactions)
 - Classification (organized per account and placed into a single ledger)
 - Summary (profit and loss)
- Modern example
 - A company's balance sheet



But...

- Ledger is centralized
- Implicit trust in the person managing it



- Enron, Arthur Andersen 2001

Enron and the Fall of Arthur Andersen

May 26, 2006 · 4:00 PM ET

Heard on [All Things Considered](#)



SCOTT HORSLEY



- Lehman Brothers 2008

Report: Lehman Brothers Used "Accounting Gimmick" To Hide The Size Of Its Balance Sheet

John Carney Mar. 11, 2010, 6:09 PM

Lehman Brothers was cooking the books far more than we ever imagined, if the allegations of bankruptcy court examiner Anton Valukus can be trusted.

Centralized book-keeping and trust

- If developed nations can't get it right, how can anyone else?
- Even if book-keeper is trustworthy, what if the ledger is hacked or deleted?
 - Adversaries or disgruntled insiders tampering with the ledger
- Motivates the need for a ledger that is...
 - Shared (for transparency)
 - Replicated and managed in a decentralized manner (for availability)
 - Authenticated, append-only, and tamper-resistant (for integrity)

Precursor #2: Currencies

Currency

- Direct settlement via *untraceable* exchange of money for goods/services
- ~3,000 B.C. in Egypt
 - Revolves around precious metals (e.g. gold) and agricultural products (barley)
 - Adopted by many ancient civilizations (e.g. Greek)
- In the US, gold/silver made into legal tender via Mint and Coinage Act of 1792
 - Establishes fixed price between gold and US dollar
 - US Mint buys and sells gold and silver at a value of 15:1
- In 1862, unable to pay debts using gold/silver, US adopts paper money as legal tender
 - Establishes a "fiat" currency for the first time in the US
 - e.g. not convertible on demand at a fixed rate

- In 1900, gold standard established and paper dollars issued to represent US gold reserves
- Extended internationally with Bretton Woods Agreements (1944)
 - WW II wreaks havoc on gold standard
 - Create gold exchange standard where price of gold fixed to the US dollar (\$35 for ounce of gold)
 - Helps make US a global superpower

Issues with currencies

- Gold standard provides stability in monetary supply via scarcity of gold
 - But perhaps not flexibility to react to problematic economic situations since supply of currency unchanged (John Maynard Keynes)
- Nixon 1971
 - Drops gold standard in financial fallout of Vietnam war
 - Government can now control supply of currency to manipulate value
 - Many believe this was problematic
 - Contributed to double-digit inflationary period in late 1970s

Digicash (1982)

- Secure, anonymous digital cash proposed by David Chaum
 - Want the benefits of on-line transactions without the drawback of transactions being traceable
 - Credit card transactions provide a paper-trail
- Model
 - Users obtain digital currency from bank
 - Spend it in a manner not traceable by bank
 - Done via blind signatures
 - <http://www.hit.bme.hu/~buttyan/courses/BMEVIHIM219/2009/Chaum.BlindSigForPayment.1982.PDF>

High-level operation

- Bank uses its private key s to sign anything
 - Anything signed is worth \$1
- Payer with an account at the bank creates a single \$1 note, blinds it, gets it signed by the bank who debits payer \$1
- Payer gets back blind and signed note, unblinds it, and provides it to the payee.
- Payee (also with an account at the bank) sends note to bank who validates its signature and updates the Payee's balance. Bank has no idea that the note is from Payer

Cryptographic primitives

- s' is the signing function of the bank (e.g. its private key)
 - s is the inverse of s' such that $s(s'(x)) = x$
- Special commuting (blinding) function c the payer applies
 - $c'(s'(c(x))) = s'(x)$
- Redundancy check r for ensuring x has been chosen with specific properties..
 - r is used to effectively check the integrity of c
 - Checks for sufficient redundancy in x to make search for valid signatures impractical in c

Digicash mechanism

- Payer randomly chooses x s.t. $r(x)$ holds for $C(x)$
- Gives $C(x)$ to the bank to sign
- Bank signs $C(x)$ and returns $S'(C(x))$ to payer
 - Debits payer's account \$1
 - Payer can not lose $S'(C(x))$ since it's a live \$1 note!
- Payer computes $C'(S'(C(x)))$ to yield $S'(x)$
- Payer checks that $S'(x)$ is valid by applying bank's public key to get x back via $S(S'(x))$
- Payer makes a payment to payee by providing $S'(x)$
- Payee forms $r(S(S'(x)))$ and stops if false
- Payee forwards $S'(x)$ to bank
 - Note that the bank has never seen x before since it was given as $C(x)$ so it does not know the payer involved! (This is the magic)
- Bank forms $r(S(S'(x)))$ and stops if false
- Bank checks note against a comprehensive list of cleared notes and stops if it is a double-spend, otherwise adds note to list
- Bank adds \$1 to payee

Hashcash (1997)

- Defense against email spam and DoS attacks developed by Adam Back
 - Computational digital postage on e-mail messages
 - Solution to a difficult proof-of-work puzzle used as postage
 - Find any x where $\text{SHA}(x \parallel \text{message}) < Y$
 - Effectively the proof-of-work function used in Bitcoin
 - Leverages pre-image resistance, avalanche effect of hash function

Precursor #3: Decentralized networks

Napster (1999)

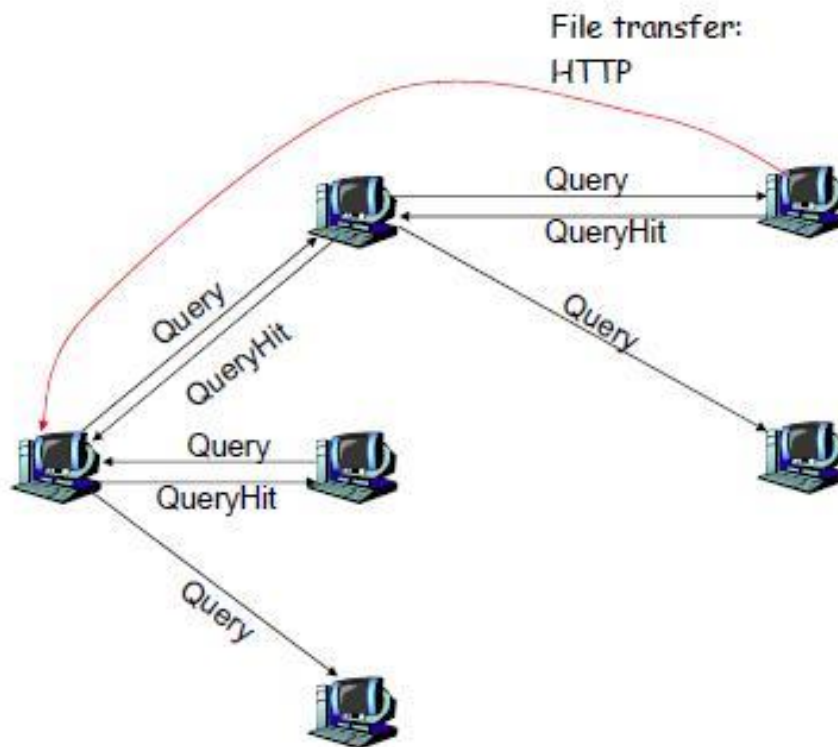
- P2P file sharing system developed by Shawn Fanning
 - One of the first decentralized applications on the Internet where users participate in system
 - Central registry maintains metadata on peers and files they have
 - Peers store actual copies of files
 - But, centralization of registry makes "censorship" trivial



The image shows a screenshot of a news article from The Washington Post. The header features the newspaper's logo, the tagline "Democracy Dies in Darkness", a hamburger menu icon on the left, and a user profile icon on the right. The main headline is "Napster Ordered to Shut Down" in a large, bold, serif font. Below the headline, the author is listed as "By James V. Grimaldi" and the date as "July 27, 2000". The first paragraph of the article reads: "A federal judge today ordered Napster Inc. to halt the operations of its phenomenally popular song-sharing Internet service, saying the upstart company was allowing copyrighted music to be pirated online."

Gnutella (2000)

- Alternative to centralized registry
 - Peers form an overlay network and are largely equal to each other
 - Queries broadcast throughout network (hop-limited)
 - Can not be shut down
 - Unless one does a wholesale block of its ports (which can be easily moved to 80)
 - Both protocol and source code are open-source



BitTorrent (2001)

- File-sharing application for large files written by Bram Cohen
 - Creates a P2P network on-demand per file being distributed
 - Nodes with entire copy of file called "seeds"
 - Altruistically allow others to copy parts of file
 - Nodes downloading a file allow others to download parts it already has
 - Eliminates free-loading, creates much higher transfer rates
 - Censorship-resistant
 - Difficult to shut down all seeds once a torrent is established
 - Results in MPAA going after search-engines for finding torrents instead of individuals holding seeds (e.g. PirateBay)

Blockchains and cryptocurrencies

Goals

- Decentralized trust
- Tamper-resistant ledger of transactions
 - (e.g. append-only, ordered log of authentic immutable transfers)
- Highly available and replicated
- Low overhead
 - Computational resources
 - Network bandwidth
 - Transaction latency
 - Transaction costs
- Anonymity (?)

BitGold (1998)

- Proposal for first decentralized blockchain for digital currency by Nick Szabo (never implemented)
 - Mechanics
 - Participant solves cryptographic puzzle to generate currency
 - Solution is sent to a byzantine fault-tolerant registry for acceptance
 - Registry assigns solution/ currency to the public-key of solver
 - Accepted solution becomes part of the next puzzle (creating a chain)
 - Majority of parties in registry must accept new solution before next puzzle can be undertaken (limits inflation)
 - System does not depend on a trusted central authority to generate currency
 - Trivia: Szabo eventually coined the term "smart contract"

RPOW (1999)

- Re-usable Proof-of-Work developed by Hal Finney similar to BitGold, but implemented
 - <https://github.com/NakamotoInstitute/RPOW>
 - Mechanics
 - Participant solves puzzle of a given difficulty and signs solution (referred to as a token) with private key
 - Publishes token to a server that registers it to public-key of participant
 - Participant can then transfer token to another participant by signing a transfer order to the recipient's public key
 - Server then registers token to public-key of recipient
 - Trusted third party prevents double-spending
- Trivia
 - Finney the receiver of the first Bitcoin transaction from Satoshi
 - Lived for 10 years in a town where a Dorian Satoshi Nakamoto lived.
 - Died of ALS in 2014

Bitcoin (2009)

Genesis block on Jan 3 2009 from Satoshi Nakamoto

(an alias)

- Public dataset available in GCP BigQuery
- ~\$500,000 block reward 10/2020

Block #0

Summary	
Number Of Transactions	1
Output Total	50 BTC

Hashes

Hash	00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Previous Block	00
Next Block(s)	00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048
Merkle Root	4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b



Transactions

SPONSORED
Crypto Credit

4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b

(Size: 204 bytes) 2009-01-03 18:15:05

No Inputs (Newly Generated Coins)



1A1zP1eP5QGefi... (Genesis of Bitcoin) - (Unspent)

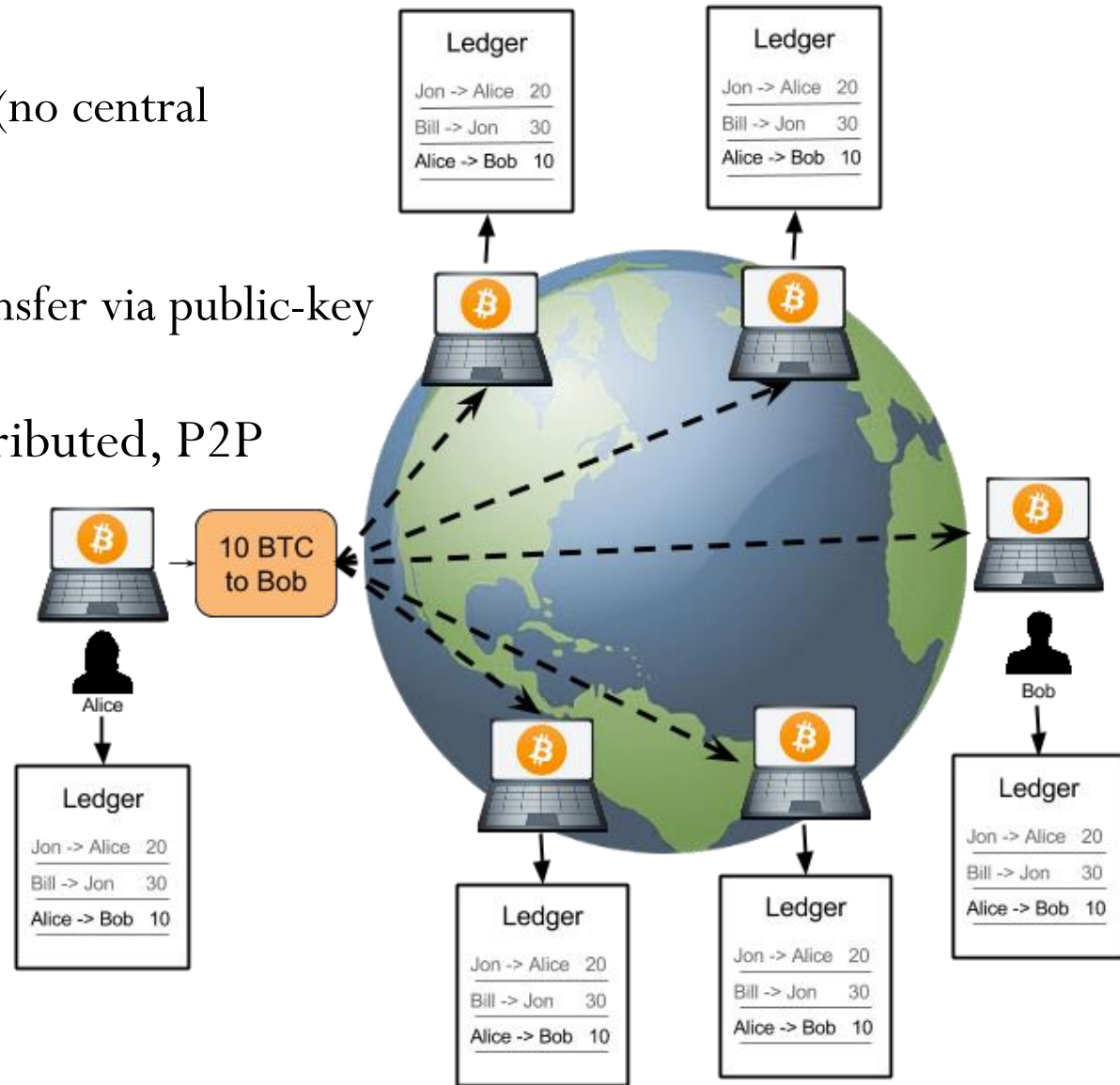
50 BTC

CoinBase

04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e636566c6c6f72206f6e206272696e6b206f666207365636f6e64206261696c6f72
(decoded) □□□□□EThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks

Basic model

- Takes ideas from...
 - Decentralized systems (no central authority)
 - BitGold (hash-chains)
 - RPOW (ownership transfer via public-key crypto)
- Builds a consistent, distributed, P2P ledger of transactions



Main innovations

- Add Nakamoto distributed consensus
 - Consensus based on majority of participants accepting the longest chain of blocks
 - Constructing chain requires CPU resources
- Add restriction on amount of currency
 - Like gold standard
 - Supply fixed via cryptographic properties
 - Unlike fiat currency whose supply is controlled by central authority

Nakamoto consensus and FLP/CAP

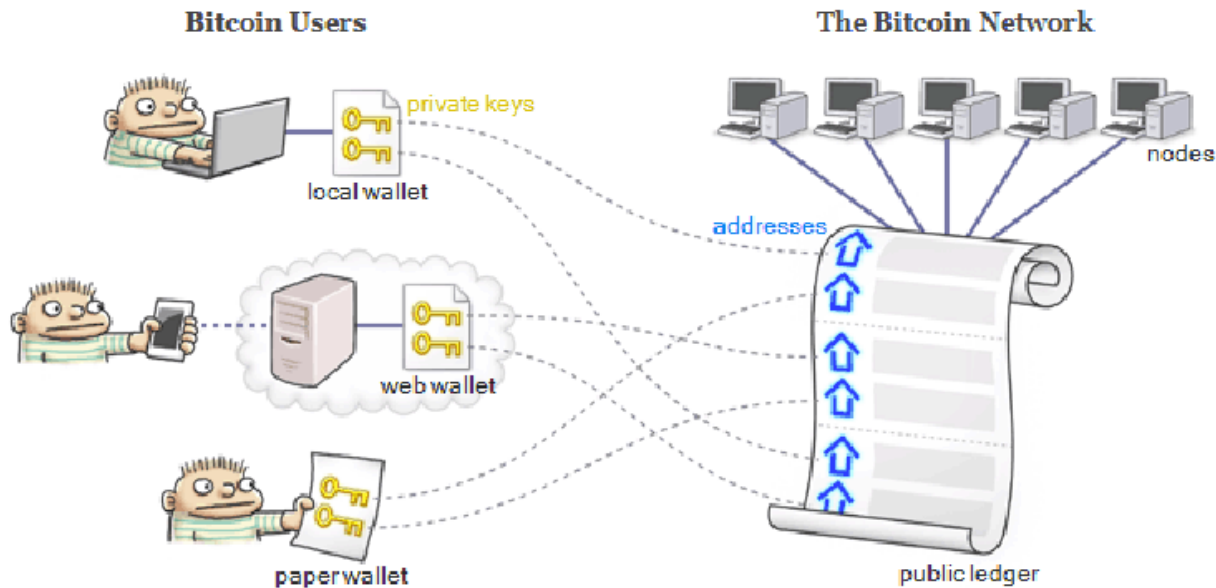
- "Consensus impossible in asynchronous network with deterministic protocol"
 - "Support *eventual consistency* in a mostly synchronous network with a randomized protocol"
- Tight synchrony ensures strong consistency
- Upon partition, compromise consistency temporarily to support availability. (CAP theorem)
 - Partition causes the blockchain to fork
 - Multiple chains created from forking point
 - Reconciled on reconnection by invalidating shorter chains
 - *Longest-chain is always accepted by Bitcoin nodes*
 - Valid, accepted transactions on shorter chain become invalid (e.g. fall off the ledger)
 - Not acceptable for many financial institutions who would rather lose availability rather than consistency in a partition (recall CAP theorem)











1. Transaction model

- <https://anders.com/blockchain/tokens>
- Transactions recorded, but not balances
 - Must replay transaction log to determine if a user can spend \$ in a transactions
 - Notion of Unspent transaction outputs (UTXOs)

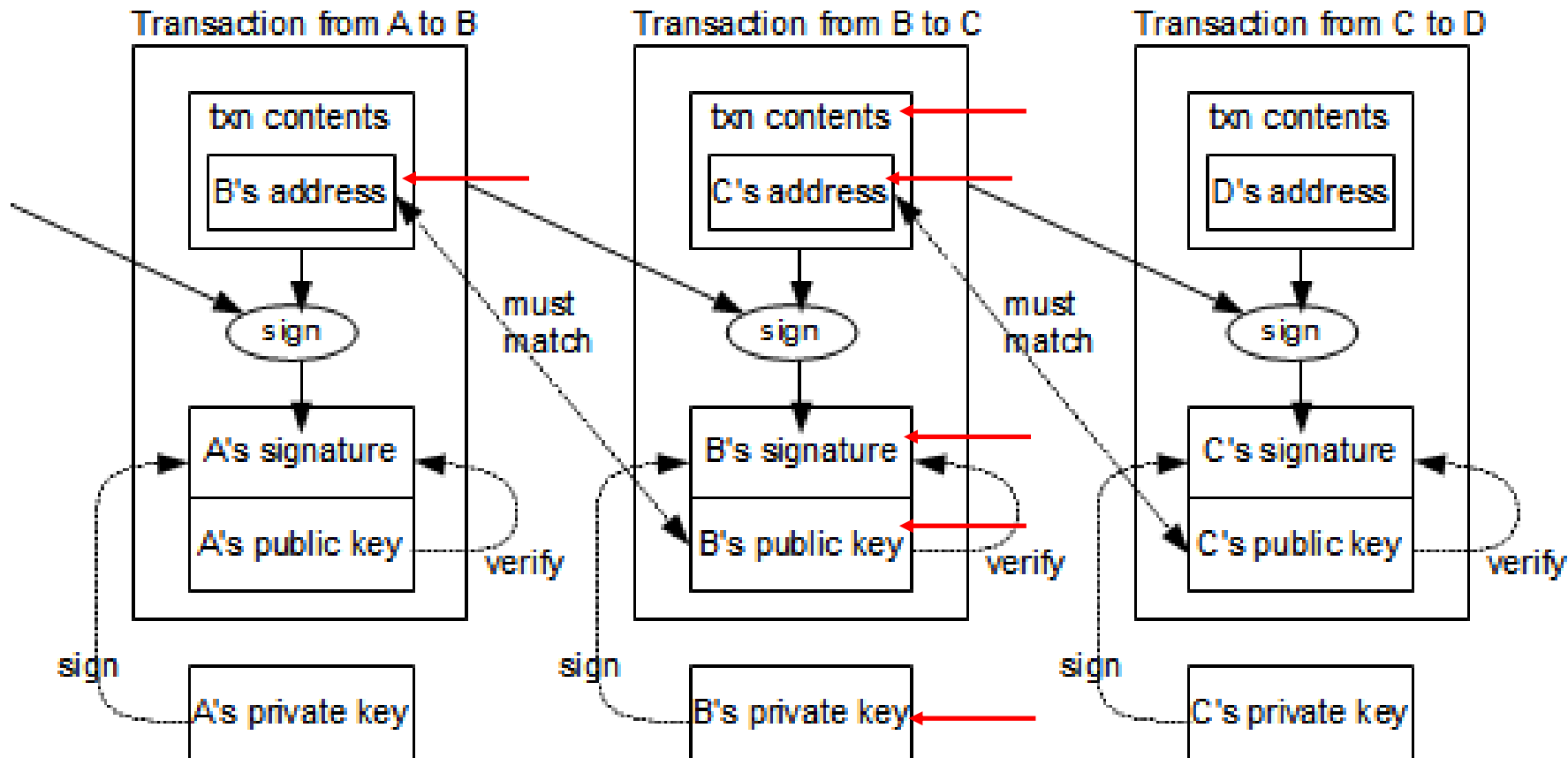
Mechanics

- Wallets with public-key as an address
 - Don't hold "Bitcoin" as in other digital cash systems, but rather corresponding **private key** to sign transactions
 - Have access to unspent currencies for corresponding **public-key addresses** indicated in Bitcoin ledger



- New transactions created by wallet, signed by private key  _S and sent into network for execution
- All nodes use wallet addresses (e.g. public-key of sender  _S) to verify signature on transaction
- Creating a transaction
 - Private key and public key of sender  _S  _S
 - Public key of recipient  _R
 - Use  _S to sign transaction (Send X amount from  _S to  _R)
 - Broadcast to full nodes for inclusion in ledger
- Full nodes use  _S to validate transaction as a candidate to be included in next block
 - Must validate via UTXOs before accepting (e.g. unspent transaction outputs where  _S is the recipient address)

- B uses private key to sign transaction to C
 - Indicates public key from which UTXOs are transferred from
 - Indicates public key for C where UTXOs are to be transferred to
- All nodes verify B's signature on transaction
- Examine ledger for prior UTXO sent to B to validate B has access
- If so, add to transaction pool for inclusion



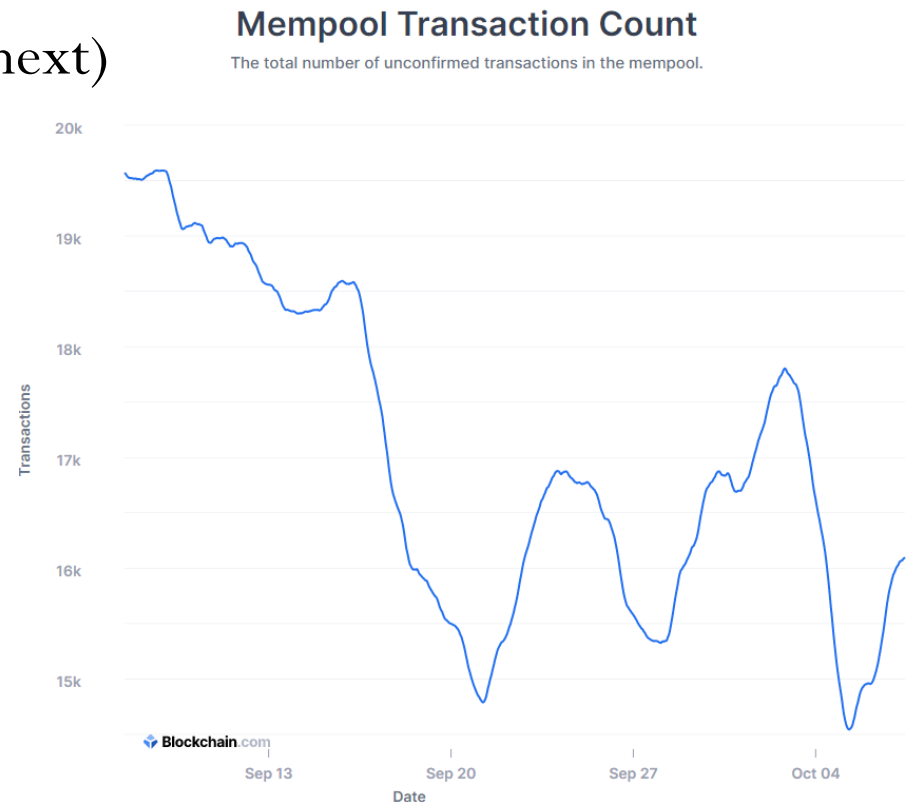
Bitcoin public/private keys

- BIP39-HD Wallets (Bitcoin Improvement Plan) standard
 - Library of 2048 short words
 - 24 words randomly selected to generate private key
 - $2048^{24} = 2^{264}$ to brute-force
 - Words hashed to create root private key
 - ECDSA produces public key
 - Public key is your address

2. Transaction processing

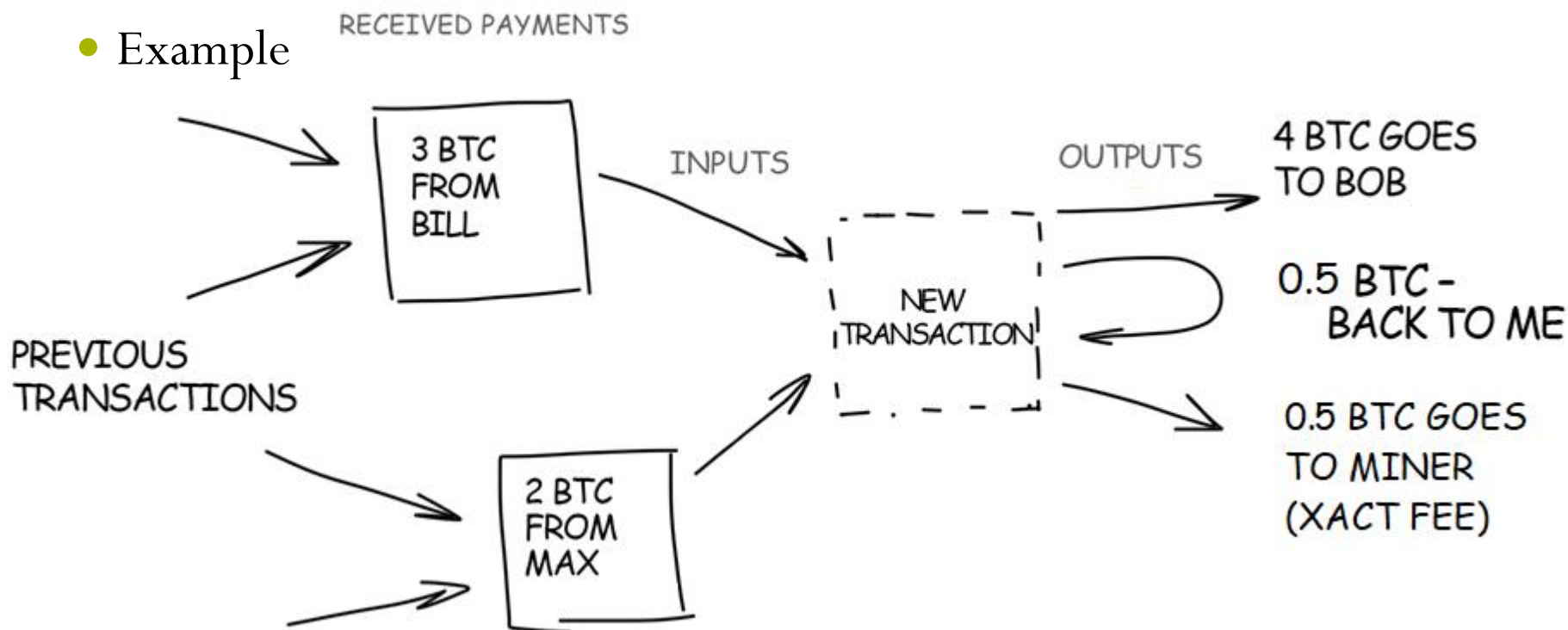
- Transactions sent to a "Mempool" within full-nodes
- Miners examine Mempool to select transactions for candidate block and validate each
- Construct proposed block and begins solving the proof-of-work puzzle

- Mempool can grow large and has a 2 week timeout (blockchain.info)
 - Transactions eventually time out and are dropped if not included in a block within 2 weeks
 - Not ideal for financial transactions!
- <https://www.blockchain.com/charts/mempool-count?timespan=60days>
- How are transactions selected? (next)



3. Miner incentives: Fees

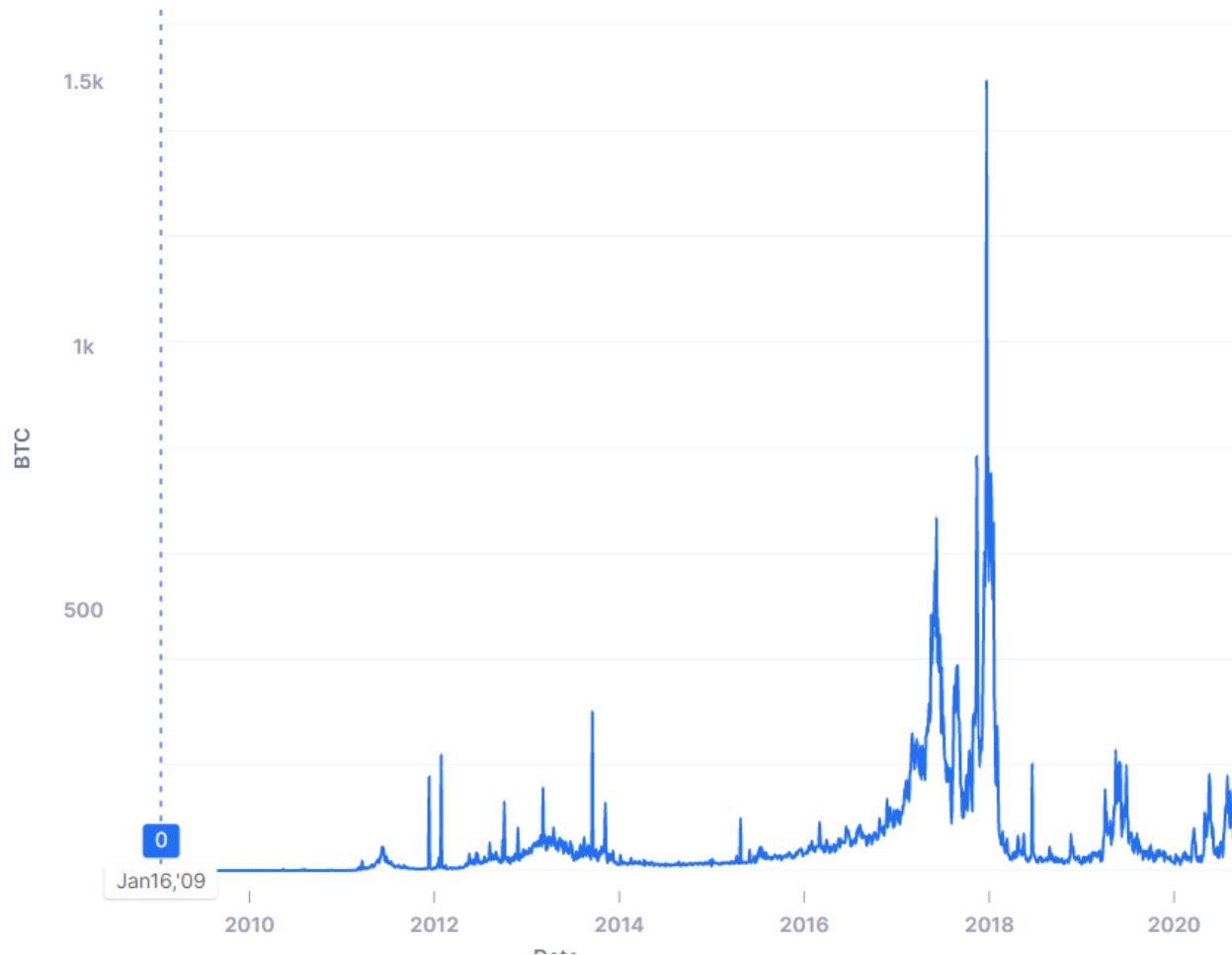
- Miner gets all transaction fees in block (specified as unallocated UTXOs in transaction)
 - Users include fees in a "bid" to get included in the next block
 - Wallet uses algorithm to guess optimal transaction fee before submitting
 - Fees automatically assigned to miner address upon successful mined block
- Example



- Leads to spikes in fees when demand is high
 - <https://www.blockchain.com/charts/transaction-fees?timespan=2years>

Total Transaction Fees (BTC)

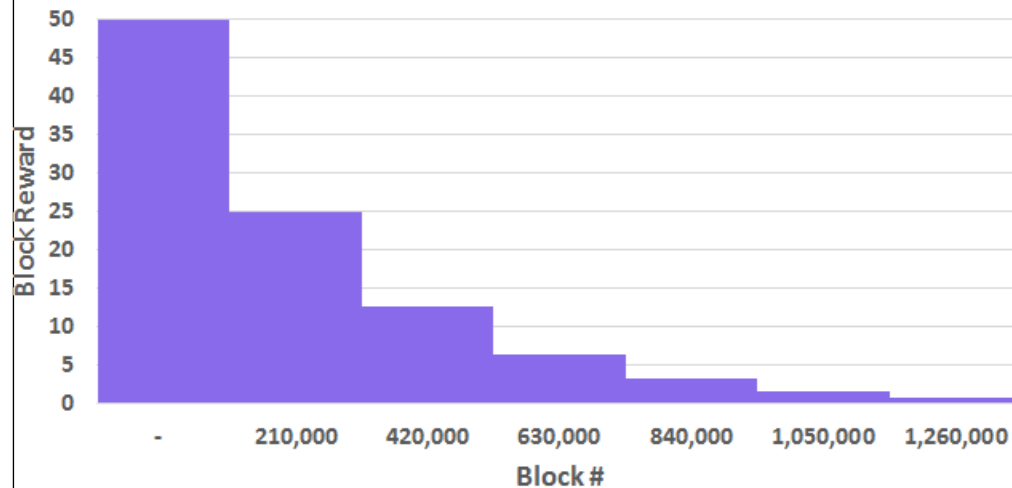
The total BTC value of all transaction fees paid to miners. This does not include coinbase block rewards.



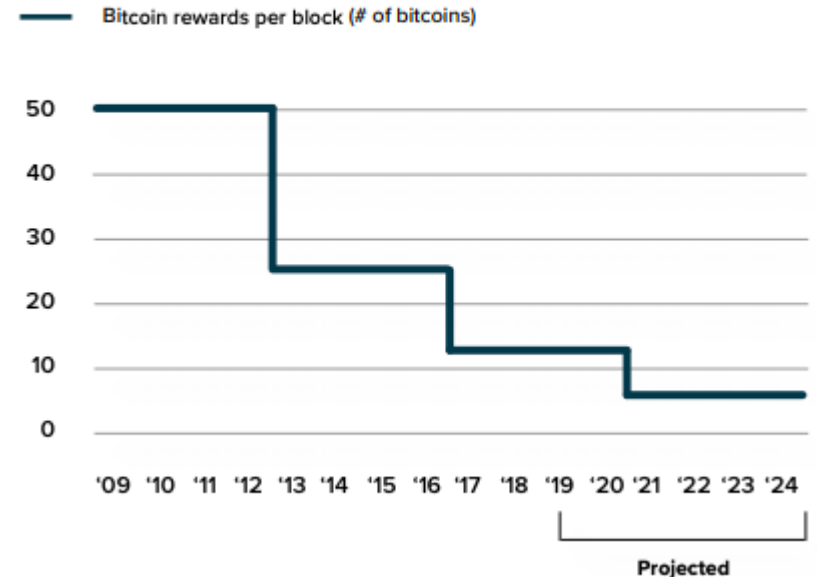
Miner incentives: Coinbase

- Miner gets block reward as first transaction in block (called the Coinbase transaction for the BTC)
 - Reward initially 50 BTC (shown in output for Block #0 earlier)
 - Halved every 210,000 blocks (~4 years) to cap supply
 - Runs out after 2147 (must rely on transaction fees afterwards)

Bitcoin
Block # vs. Block Reward



The amount of bitcoins added to the network in each block of transactions is cut in half every 210,000 blocks (roughly every four years)



- See <https://www.blockchain.com/explorer>

Block 651842 ⓘ

Hash	00
Confirmations	1
Timestamp	2020-10-08 09:49
Height	651842
Miner	ViaBTC
Number of Transactions	2,633
Difficulty	19,298,087,186,262.61
Merkle root	6e1282313fc69bf9921d
Version	0x20800000
Bits	386,831,838
Weight	3,993,510 WU
Size	1,359,831 bytes
Nonce	1,429,285,929
Transaction Volume	9487.27216357 BTC
Block Reward	6.25000000 BTC
Fee Reward	1.15637590 BTC

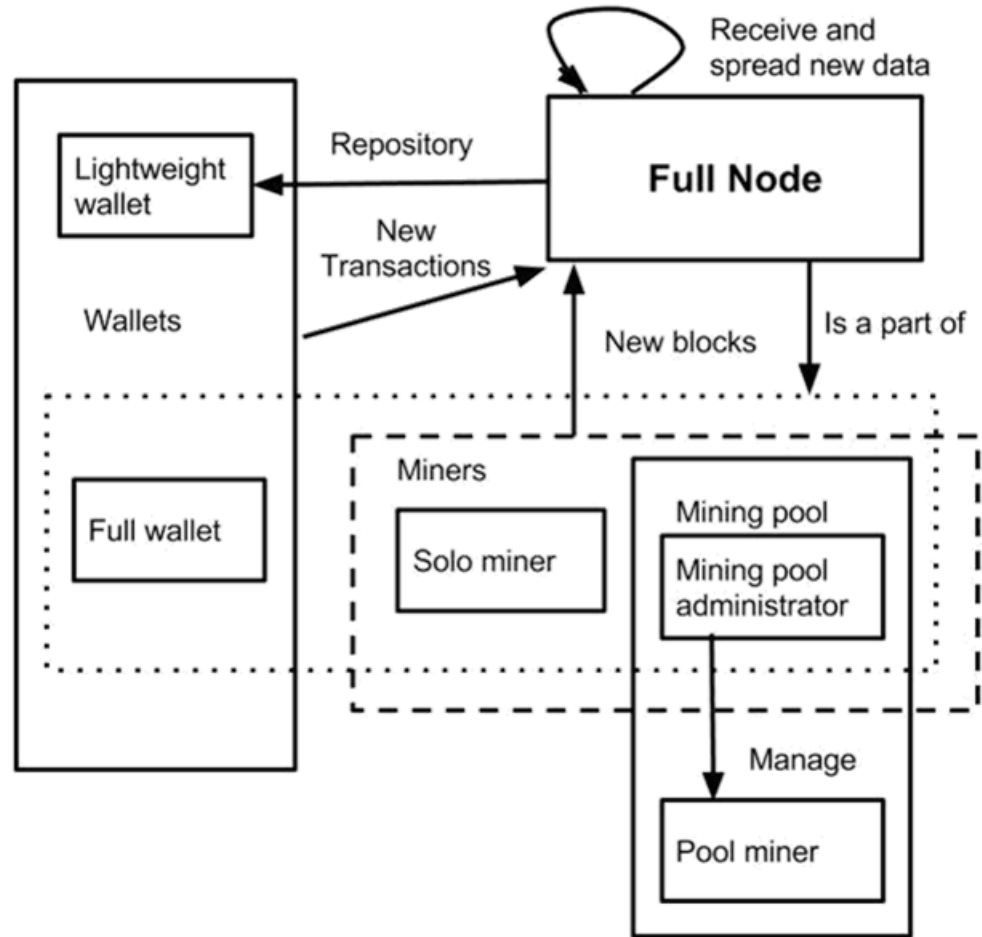


Demo

- No coinbase to determine Block #1 transactions valid!
 - <https://anders.com/blockchain/tokens>
- Coinbase given to miner who successfully mines Block #1 (anders)
 - <https://anders.com/blockchain/coinbase>
 - Initially anders, who then kicks off transactions
 - Later sophia

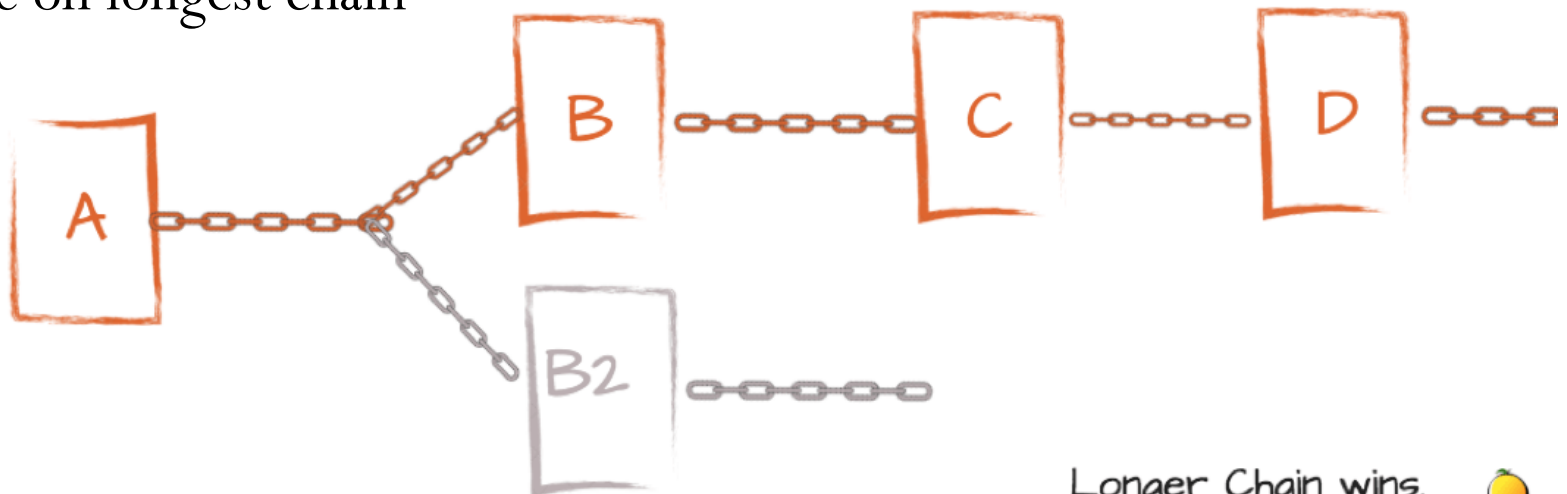
3. Mining details


- Miners (often organized as a pool) solve a PoW puzzle based on
 - Hash of prior block
 - Hash of proposed block containing transactions selected
 - Finds the nonce that results in a partial hash collision with difficulty specified algorithmically
- As soon as a miner solves the puzzle for a proposed block, it is broadcast
 - All full nodes validate block and its solution
 - Immediately accept it and move onto next block



Mining details

- Blocks with invalid transactions or bad hashes rejected (along with reward)
 - Miners responsible for verifying transactions before solving puzzle
 - Blocks must obey rules of the game (protocol)
- Longest chain wins
 - Can only profit by mining off of latest block!
 - Orphan blocks fall off chain (as do their coinbase!)
 - No one wants to mine a block that falls off the chain, so miners always mine on longest chain

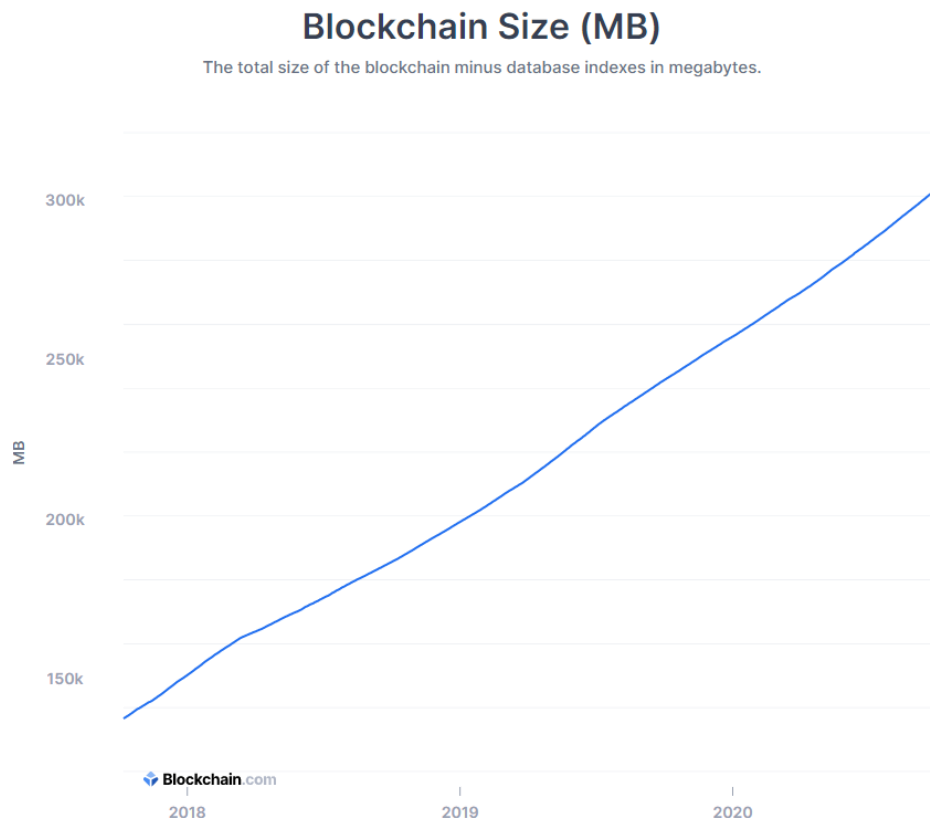


Longer Chain wins.
Shorter is "orphaned" 

- Leads to the notion of "confirmations" and "block depth"
 - Number of blocks that have reconfirmed your block as part of chain
 - Versus block height (# of blocks from genesis block)
 - Typically must wait 3-4 confirmations to ensure no orphans
 - 40 minute transaction delay!

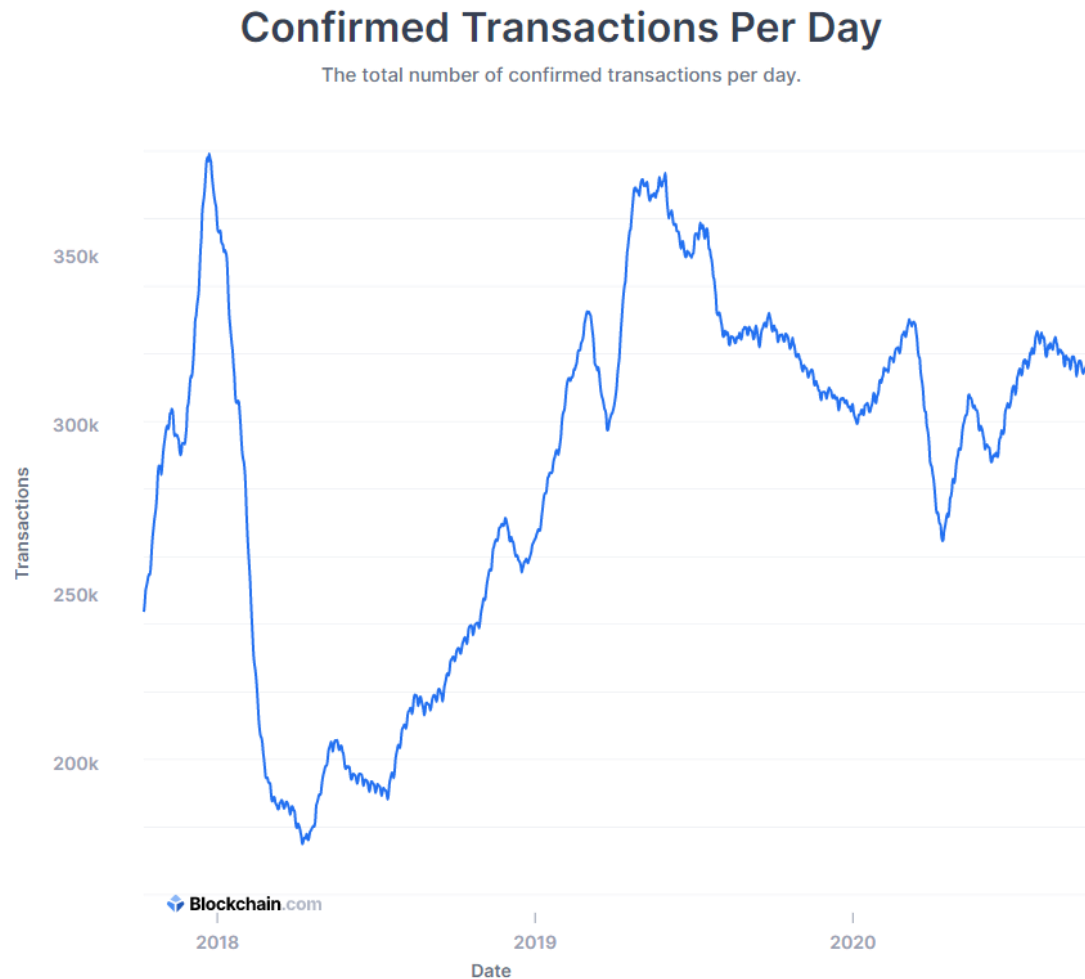
4. Design for decentralization

- Designed so anyone can participate (like BitTorrent/Gnutella)
 - 1 block (1 MB) every 10 minutes
- Reason #1: Size of full-node grows linearly
 - Currently around 300GB and can be stored on a Raspberry Pi!
 - <https://www.blockchain.com/charts/blocks-size>



- But, limits transaction rate

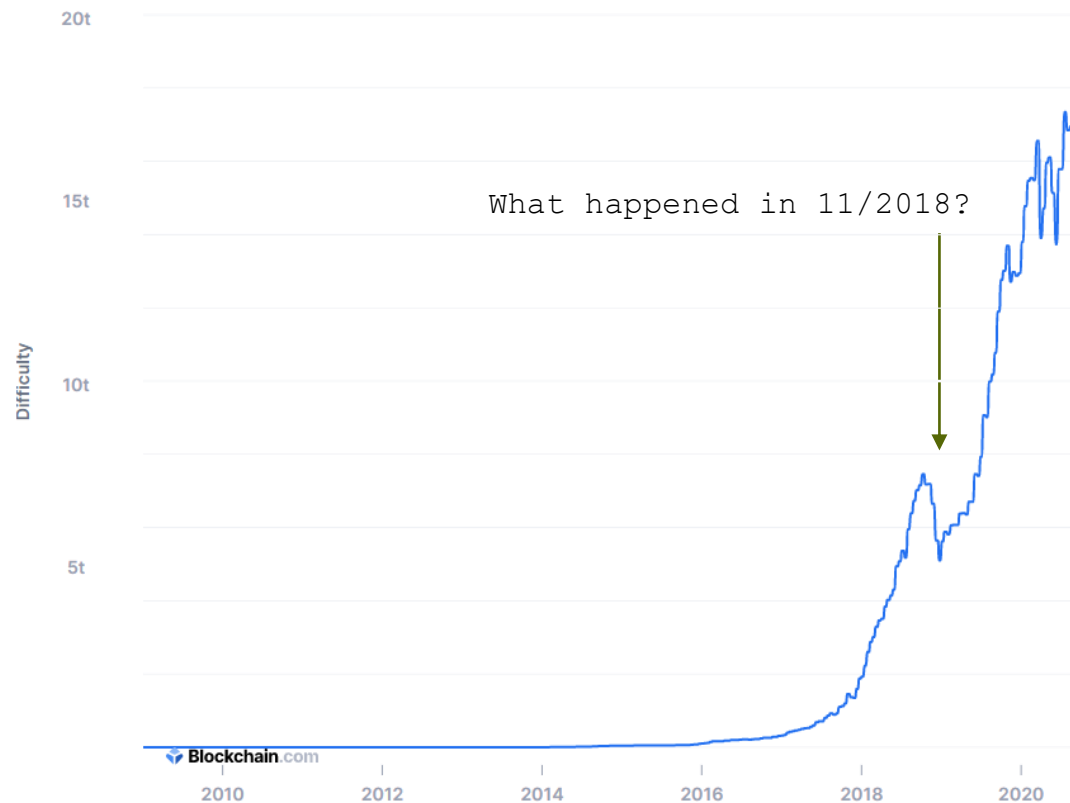
- <https://www.blockchain.com/charts/n-transactions?timespan=5years> (per-day)
- Currently supports ~ 7 transactions per second
- Compare to Visa network of 2k transactions per second average and 50k transactions per second peak



- Reason #2: Control rate that blocks are added to maintain consistency
 - Propagation time for replicating blocks \ll Creation time between new blocks
 - Solves double-spending problem by parameterizing proof-of-work difficulty to ~ 10 -minutes (Parameter reset every 2 weeks based on averaging)
 - Implemented via rule in software
 - <https://www.blockchain.com/charts/difficulty?timespan=all>

Network Difficulty

A relative measure of how difficult it is to mine a new block for the blockchain.



- Cryptocurrency winter sees profitability plummet

Bitcoin is down more than 80% from last year's high, nearing its worst-ever bear market

- Despite bitcoin's relatively short 10-year existence, it's already on its third bear market plunge of 80 percent or more.
- Bitcoin has seen downturns as deep as 92 percent in 2011, and about 84 percent leading up to 2015.
- In dollar terms though, 2018 has been by far the worst rout, with \$700 billion in market cap knocked off the cryptocurrency market.

Kate Rooney | @Kr00ney

Published 1:54 PM ET Mon, 26 Nov 2018 | Updated 4:11 PM ET Mon, 26 Nov 2018

HOME » NEWS

More than 10,000 Bitcoin miners shut down mining due to markets crash

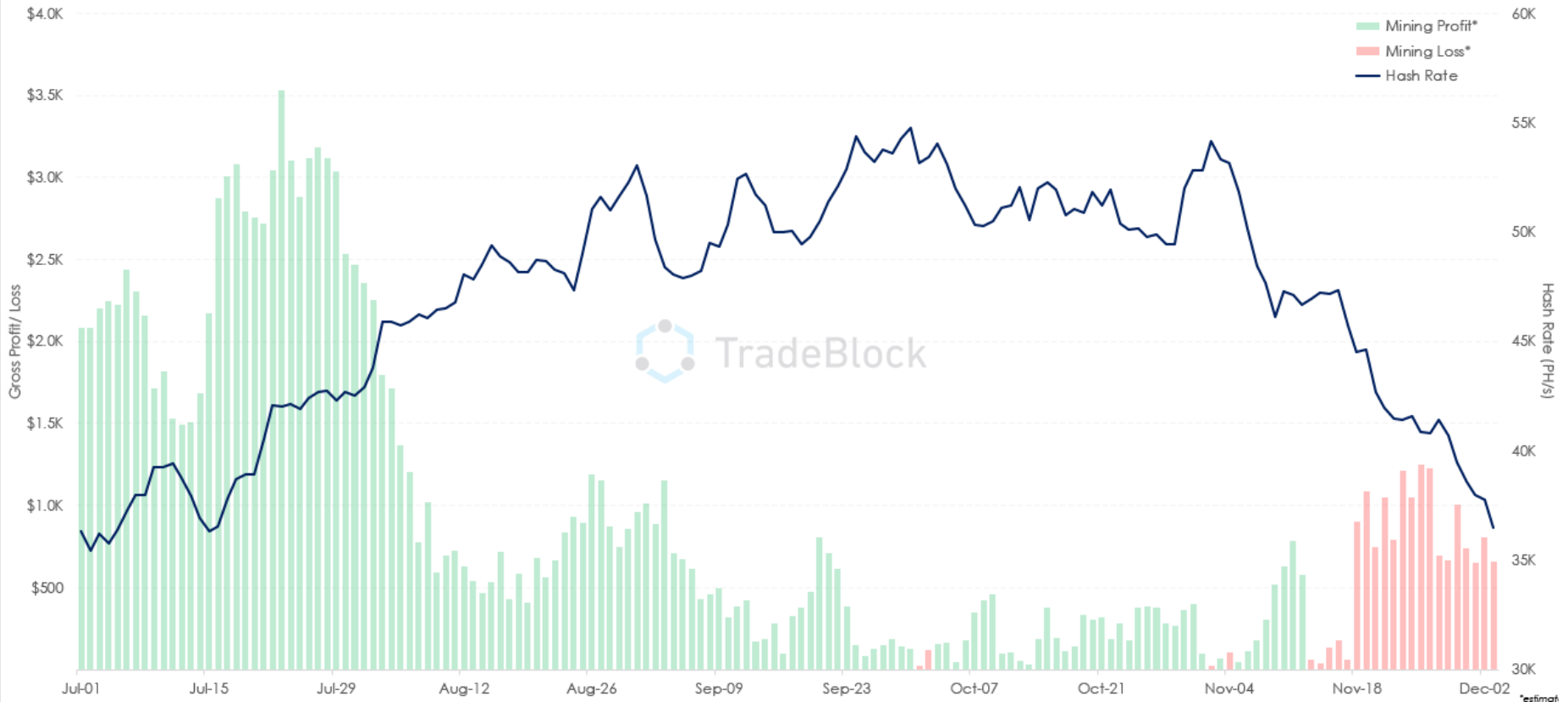


Amrit Mirchandani POSTED ON NOVEMBER 21, 2018

♥ 2 👁 7.9K Views 💬 0

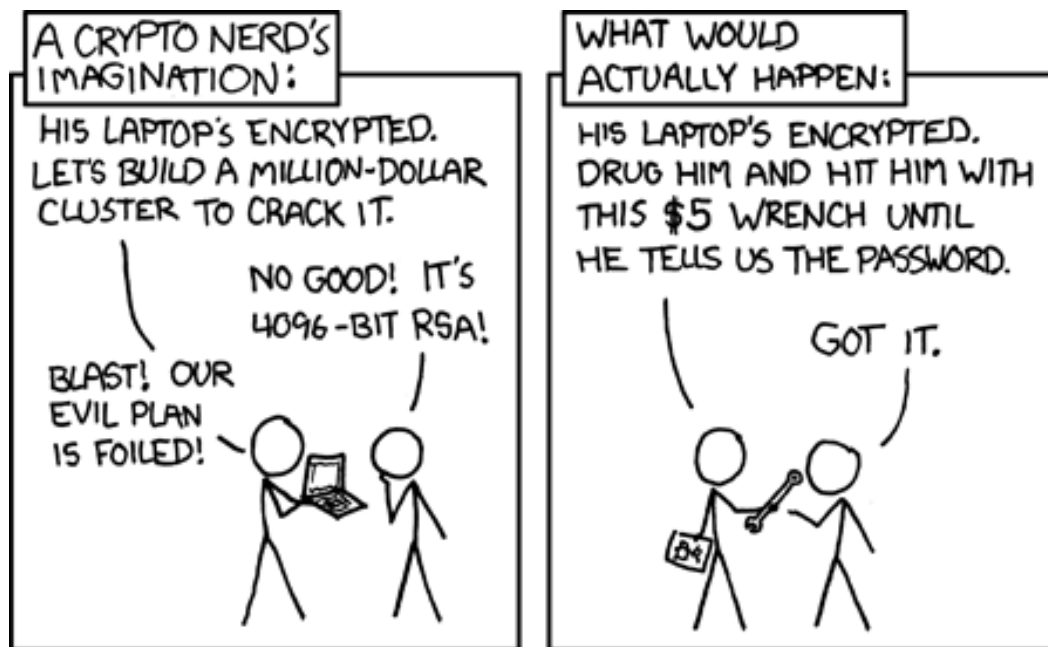
- Mining profitability chart (late 2018)

- https://research.tradeblock.com/wp-content/uploads/2018/12/20181206-Hash_Rate-Mining_Cost-1.png

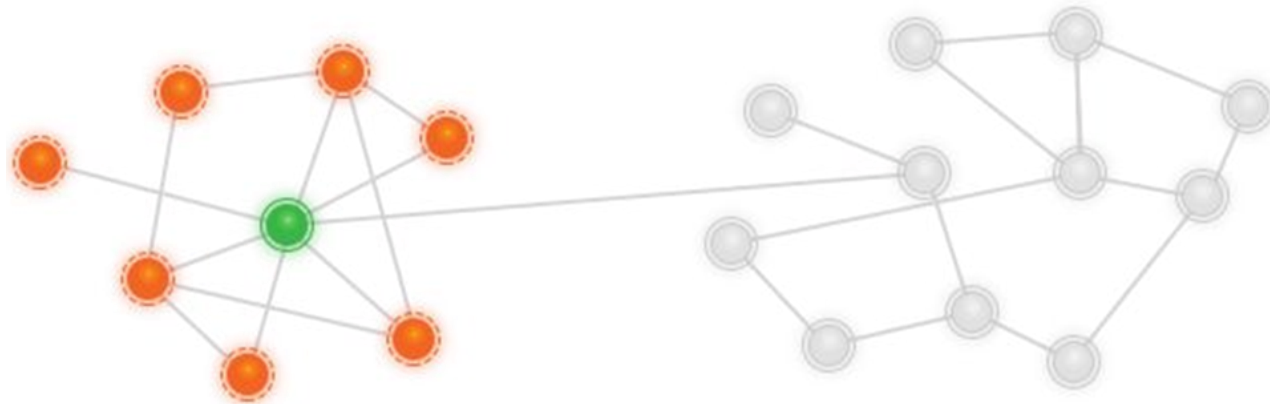


5. Security

- Authentication done via public-key cryptography with no trust required between peers
 - Accounts can not be disabled
 - User must now be responsible for securing his/her private key
 - Do you trust yourself to do this?



- Pseudo-anonymous: can trace addresses by transactions through blockchain
 - Subsequent systems attempt to improve anonymity (Zcash)
 - Many used for illegal activity (e.g. 90% of ZCash usage)
- Resists Sybil attack
 - Adversary launching multiple identities to corrupt consensus protocol
 - Ability to add blocks to blockchain determined by capacity to solve Proof-of-Work puzzles
 - Must own majority of CPU resources to subvert (51% attack..more later)



6. Scalability

- Current implementation difficult to use for small transactions
- Proposed alternatives
 - Litecoin (2011)
 - Code fork of Bitcoin
 - Uses script – sequential memory-hard puzzle (makes ASIC mining difficult)
 - Block time = 2.5 min
 - Block reward = 50 LTC halving every 4 years
 - Block size = 1MB
 - Bitcoin Cash (2017)
 - Hard fork of Bitcoin
 - Block size increased to 4MB and beyond to support lower transaction costs and faster transactions
 - No longer require that full-nodes run on embedded devices
 - Eventually forked again...

Side chains and transaction aggregation

- Transaction throughput small on Bitcoin (7 per second)
- Lightning network (2018) <https://lightning.network/>
 - Layer a transaction aggregation system on top of blockchain to reduce number of transactions (Layer 2 solutions)
 - Like opening up a tab at a bar, opening tab and settlement at end are the only things recorded
- Or create a side blockchain, then sync its blockhashes to main chain (sharding)
 - Create secondary payment network where transactions and balances summarized and committed to Bitcoin blockchain periodically
 - Reduces load on Bitcoin network, allows for higher transaction throughput
 - Hierarchical blockchains being proposed to scale transaction throughput

Hyperledger



Hyperledger

- Open-source implementations of permissioned blockchains (where participants are trusted)
 - Curated like Apache project
 - Typically for the enterprise
 - Allows enterprises to *see* code they rely upon
 - Different projects for different styles of deployments
 - Focused on adherence to regulatory compliance
 - Not possible with Bitcoin or Ethereum
 - Commonly used projects and their consensus protocols
 - https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf
 - Fabric (IBM)
 - Kafka + other consensus protocols
 - Iroha (Soramitsu/Hitachi)
 - Sumeragi
 - Indy (Sovrin)
 - RBFT
 - Sawtooth (Intel)
 - PoET

Component layers

- Consensus
 - Manage distributed agreement and ensuring correctness
- Smart contract (validation)
 - Executing code and business logic
- Communication
 - Message transport between nodes
- Data store
 - Backend storage
- Cryptography
 - Algorithms used for confidentiality, non-repudiation, authentication, etc.
- API
 - Access to blockchain

Comparison to public blockchains

	Bitcoin	Ethereum	Hyperledger Frameworks
Cryptocurrency based	Yes	Yes	No 
Permissioned	No	No	Yes (in general)*
Pseudo-anonymous	Yes	No	No
Auditable	Yes	Yes	Yes
Immutable ledger	Yes	Yes	Yes
Modularity	No	No	Yes 
Smart contracts	No	Yes	Yes
Consensus protocol	PoW	PoW	Various** 