Blockchain Basics

What is a Blockchain?

Hashing

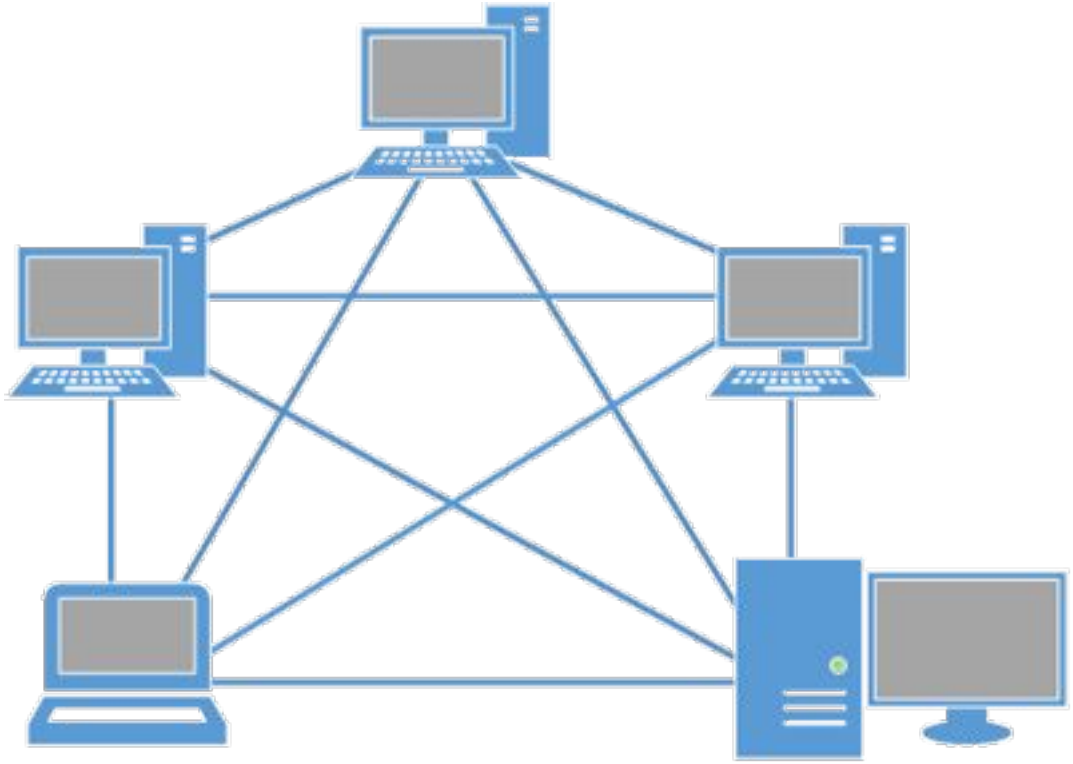
```
$ python hash_example.py

I am Satoshi Nakamoto0 => a80a81401765c8eddee25df36728d732...
I am Satoshi Nakamoto1 => f7bc9a6304a4647bb41241a677b5345f...
I am Satoshi Nakamoto2 => ea758a8134b115298a1583ffb80ae629...
I am Satoshi Nakamoto3 => bfa9779618ff072c903d773de30c99bd...
I am Satoshi Nakamoto4 => bce8564de9a83c18c31944a66bde992f...
I am Satoshi Nakamoto5 => eb362c3cf3479be0a97a20163589038e...
I am Satoshi Nakamoto6 => 4a2fd48e3be420d0d28e202360cfbaba...
I am Satoshi Nakamoto7 => 790b5a1349a5f2b909bf74d0d166b17a...
I am Satoshi Nakamoto8 => 702c45e5b15aa54b625d68dd947f1597...
I am Satoshi Nakamoto9 => 7007cf7dd40f5e933cd89fff5b791ff0...
I am Satoshi Nakamoto10 => c2f38c81992f4614206a21537bd634a...
I am Satoshi Nakamoto11 => 7045da6ed8a914690f087690e1e8d66...
I am Satoshi Nakamoto12 => 60f01db30c1a0d4cbce2b4b22e88b9b...
I am Satoshi Nakamoto13 => 0ebc56d59a34f5082aaef3d66b37a66...
I am Satoshi Nakamoto14 => 27ead1ca85da66981fd9da01a8c6816...
I am Satoshi Nakamoto15 => 394809fb809c5f83ce97ab554a2812c...
I am Satoshi Nakamoto16 => 8fa4992219df33f50834465d3047429...
I am Satoshi Nakamoto17 => dca9b8b4f8d8e1521fa4eaa46f4f0cd...
I am Satoshi Nakamoto18 => 9989a401b2a3a318b01e9ca9a22b0f3...
I am Satoshi Nakamoto19 => cda56022ecb5b67b2bc93a2d764e75f...
```

How does a normal payment system work?

Account	A	B	C	D
Initial	75	60	10	100
Transfer	-50	+50		
Transfer		+15	+15	
Transfer	-25			+25
Transfer		-40	+40	
~~Transfer~~			~~+200~~	~~-200~~
Transfer	+50	-50		

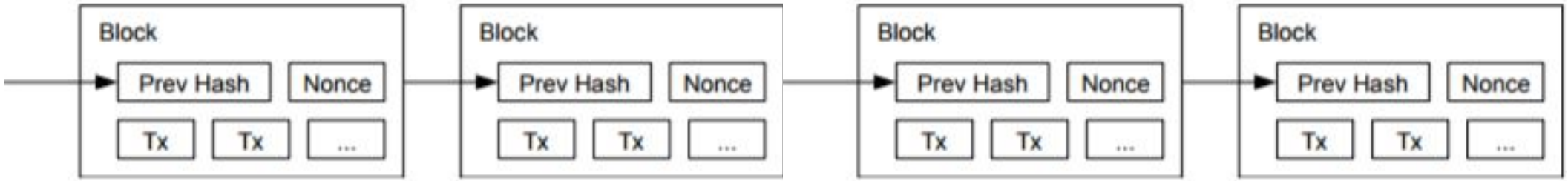
P2P Network



What does the network we want have to have?

- Everyone must be able to check for fraudulent transactions
  - Thus, everyone must be able to access all past data with transactions having some kind of timestamp
- Past data must be immutable meaning that people should not be able to change transactions that already exist
  - Immutability prevents people from sending transactions to someone and then reversing the transaction later fraudulently
- All nodes must have a way to agree on what new transactions will be added to the ledger
  - Nodes must be able to achieve **consensus** on what transactions will be added
  - Everyone relevant for consensus must know what transactions will be included

Blockchain

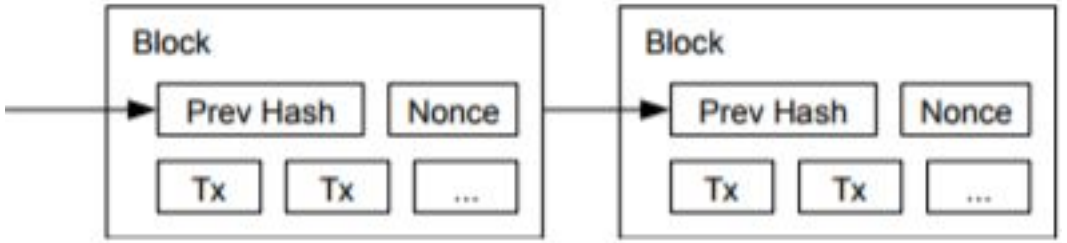

Block
Prev Hash
Nonce
Tx
Tx
...
Block
Prev Hash
Nonce
Tx
Tx
...
Block
Prev Hash
Nonce
Tx
Tx
...
Block
Prev Hash
Nonce
Tx
Tx
...

Distributed Ledger

Account	A	B	C	D
Initial	75	60	10	100
Transfer	-50	+50		
Transfer		+15	+15	
Transfer	-25			+25
Hash				
Transfer		-40	+40	
Transfer			+100	-100
Transfer	+50	-50		
Hash				
Transfer	-30			+30
Transfer	-20	+20		
Transfer		+65	-65	
Hash				
Transfer			-50	+50
Transfer	+20	-20		
Transfer		+30	-30	

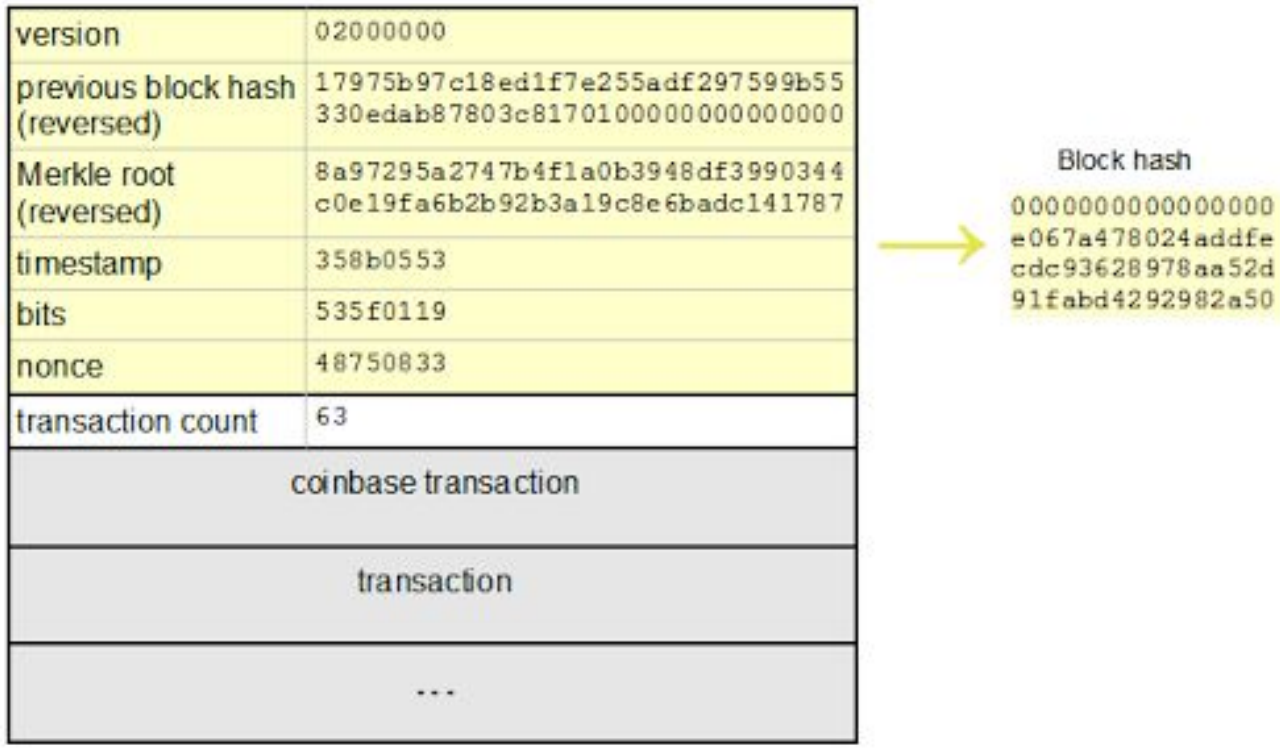
Blockchain

- Validatable by everyone
  - Timestamped and ordered transactions
- Immutability
  - Changing past blocks invalidates all future blocks
- Nodes must have a way to come to consensus on what transactions to add



Proof of Work

- Hash the block repeatedly while incrementing the nonce until you get an output with a desired characteristic (such as a certain number of zeroes)
- Ideally, we would create blocks based on what the majority wants but if the majority was based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it.*



*taken from the bitcoin whitepaper

https://bitcoin.org/bitcoin.pdf

What happens if the network “forks”?

- A network fork can occur if there are two valid blocks that both exist simultaneously
- Nodes will choose whichever chain ends up being longer (and thus the one that has more computational power)
- 51% of computing power (minimum) required to fork the network or to reverse and overtake a previous block

Why not vote?

- Requires centralization and a level of trust in other nodes
  - Ripple
- Sybil attacks
- Censorship
- Need to require more effort for an attacker to censor the network

Characteristic for choosing next block creator

- Computational power
  - Anyone can participate if they have a CPU or GPU, the more hardware they have the higher the chance of winning
  - Proof-of-work, solve puzzles in return for a chance to create the next block
  - 1 CPU = 1 Vote
  - Downside is that it requires significant amount of power and hardware
- Stake
  - 1 Coin = 1 Vote
  - The more of a coin you have, the higher the chance your chance of being next block creator
  - Can end up with people that control significant amount of the network just getting richer
  - Referred to as proof-of-stake

How would you set up a local payment system?

- Easiest thing to do would be to designate a central treasurer to keep track of everyone’s balances in the form of a ledger
- They would keep a record of all past transactions in order to check incoming transactions for validity
    - Leads to fees, higher sending times, lower security (from depending on a central authority) and decreases privacy
- Want some kind of decentralized alternative

What should the ideal consensus be?

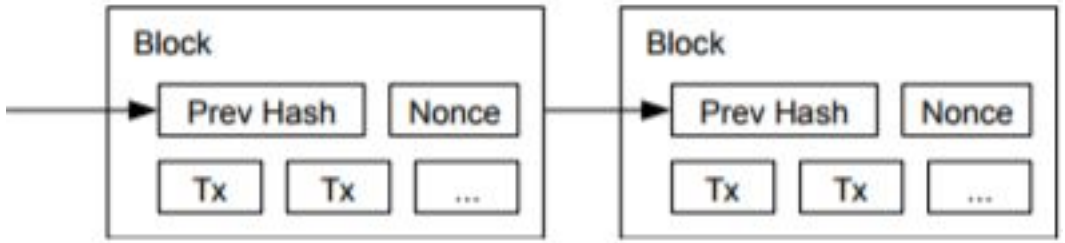
- Reject transactions that are invalid
- Add all valid transactions (that do not conflict with each other)
  - Must have some constraint to prevent the size of the chain from becoming too big for anyone to store
- Everyone should be able to make transactions
  - Transaction propagation (gossip protocol)

Easiest Option: Vote!

Block Creation Reward

- To motivate users to attempt to create the next block, there are two rewards:
  - Coinbase transaction
    - Transaction to user at the beginning of each block that gives them a set reward
  - Transaction fees
    - To motivate block creators to actually include transactions (since a block with no transactions is valid) there is a fee that is included when you send a transaction
      - The person that actually creates the block receives this fee

Cryptocurrencies



+

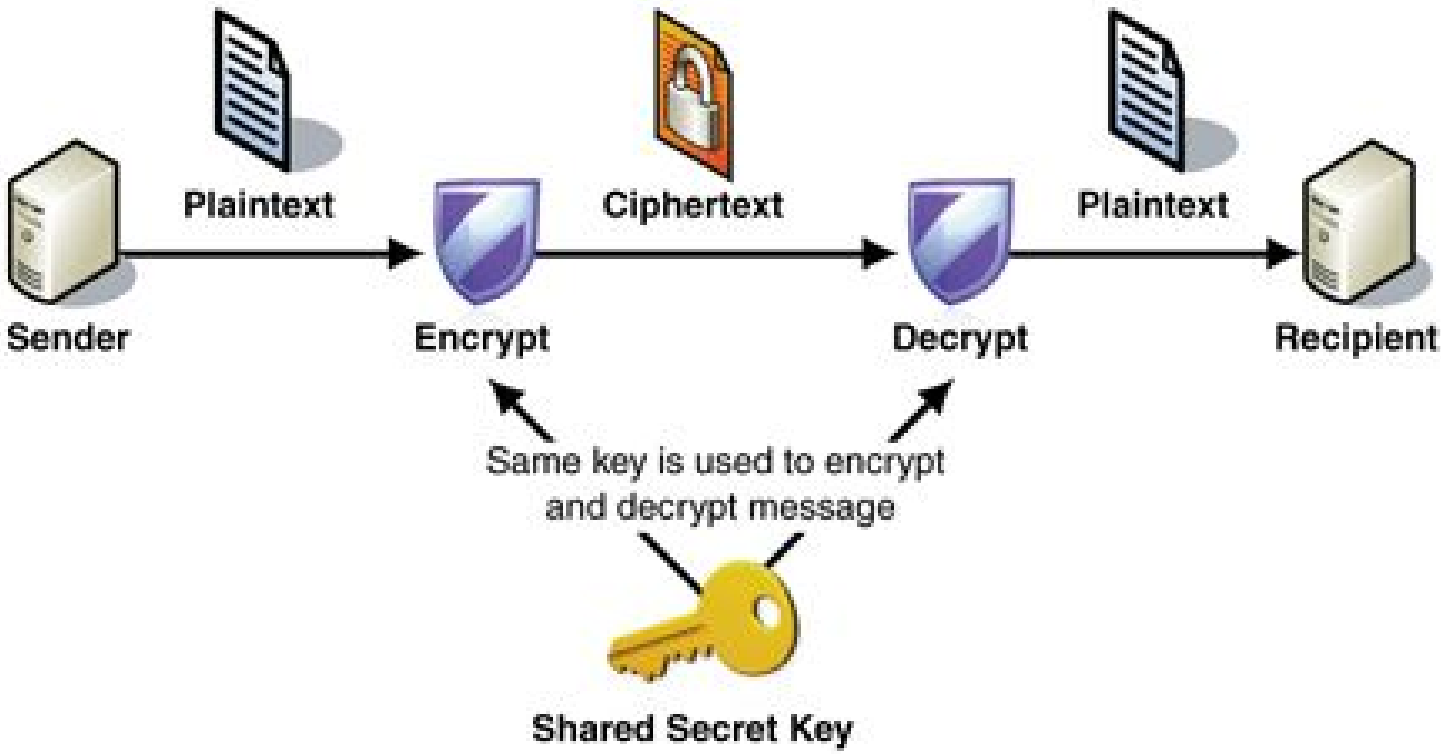
Consensus Algorithm

Summary

1.) Users broadcast their transactions to the network who give these to other nodes.
2.) Nodes receive these transactions and include them in their next block and attempt to find a nonce.
3.) When a miner succeeds in finding a proof-of-work, they broadcast their block to the network. Nodes on the network verify that all transactions are valid and then broadcast it to other nodes till all nodes accept the block.
4.) Miners then repeat the process and use the previous block hash in their new block.

Addresses and Transactions Using Public-Key Cryptography

Symmetric Key Cryptography

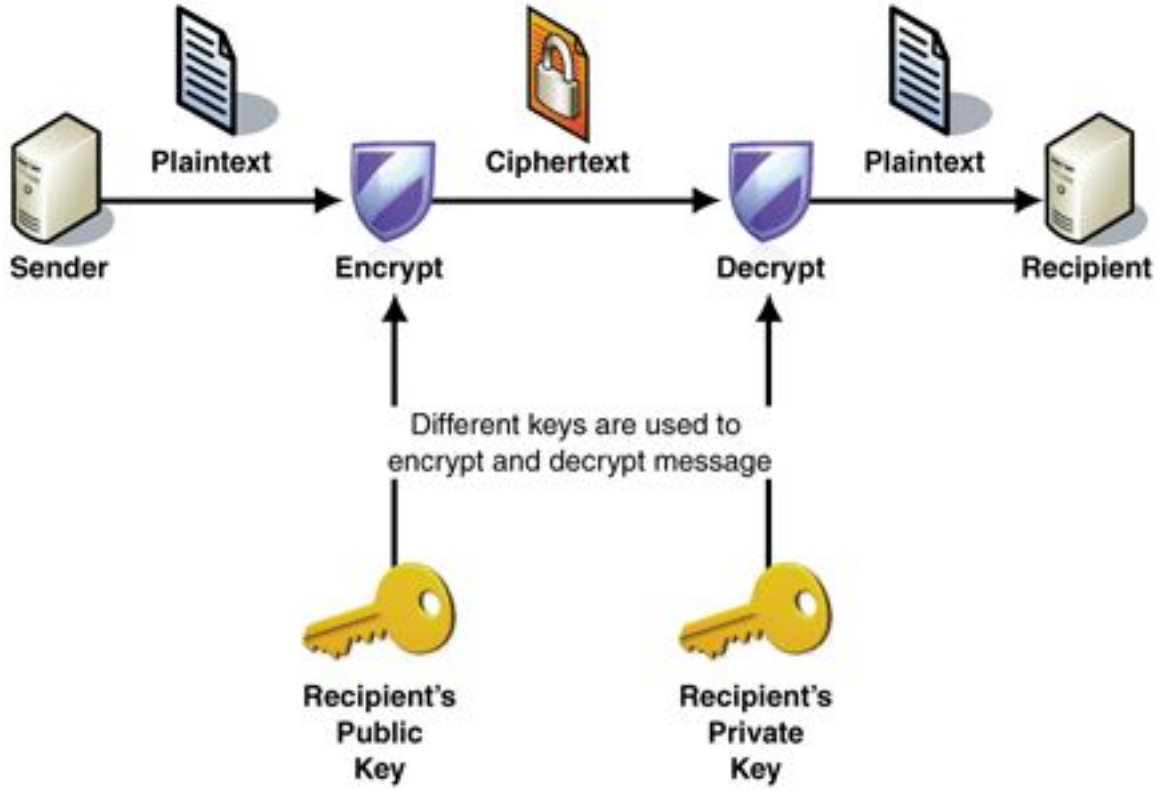


Public/Private Key Cryptography

Private Key: A secret key that you don’t share

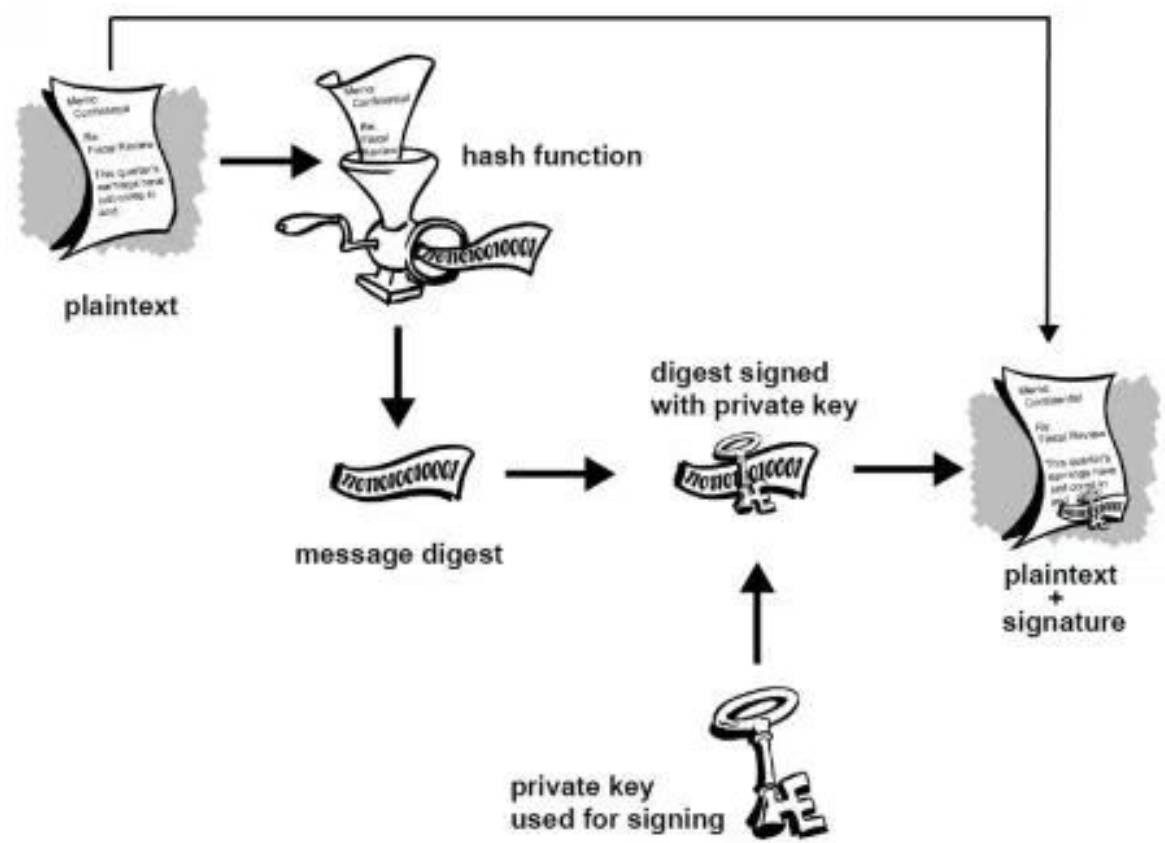
Public Key: Key that you share with others

Asymmetric Key Cryptography



http://ehindistudy.com/2015/10/01/symmetric-and-asymmetric-key-cryptography-in-hindi/

Digital Signatures

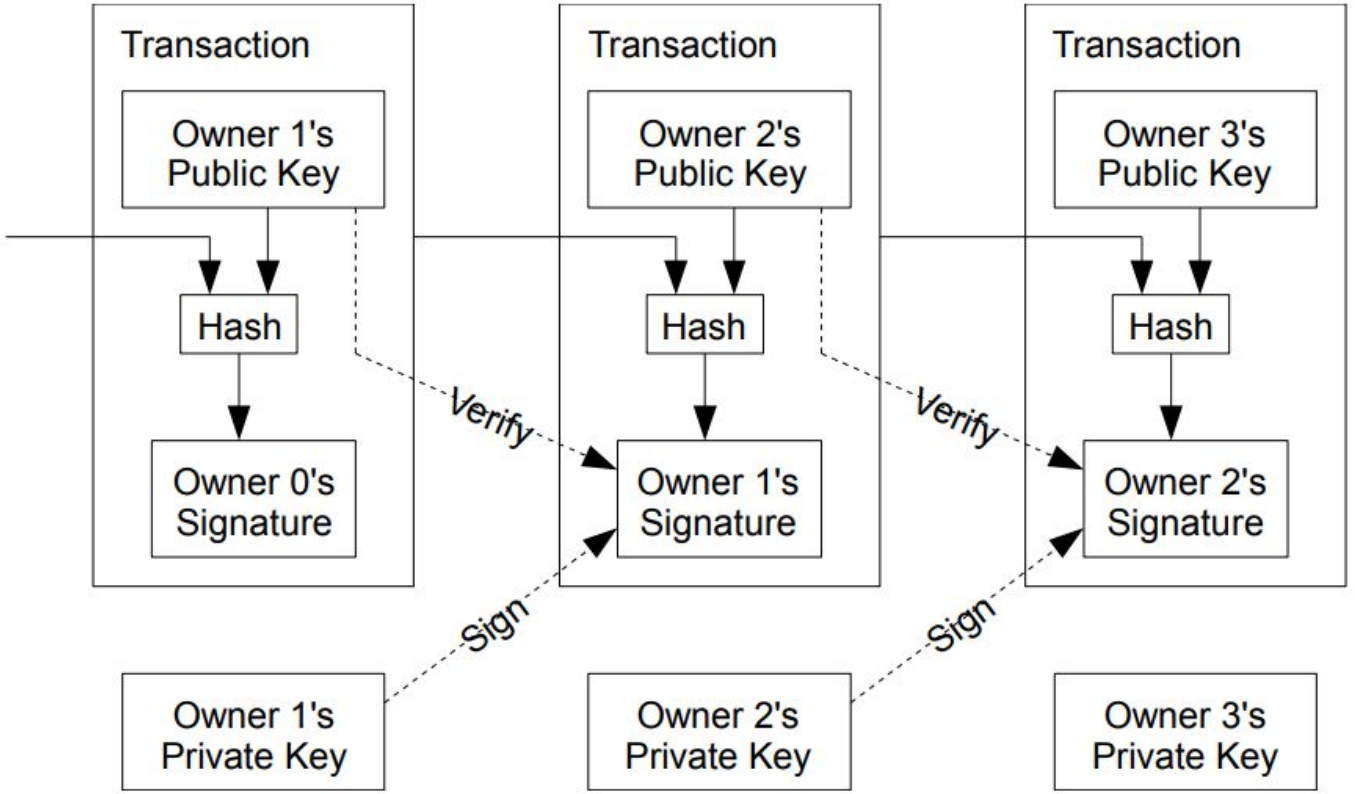


Addresses

- Instead of identifying people by identity, we identify individuals by their public key
- Once you generate a private/public key pair you can generate a cryptocurrency address
- If people want to send you cryptocurrency they send it to this address

Transactions

“We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.”



https://chrispacia.wordpress.com/2013/09/07/bitcoin-cryptography-digital-signatures-explained/

Smart Contracts

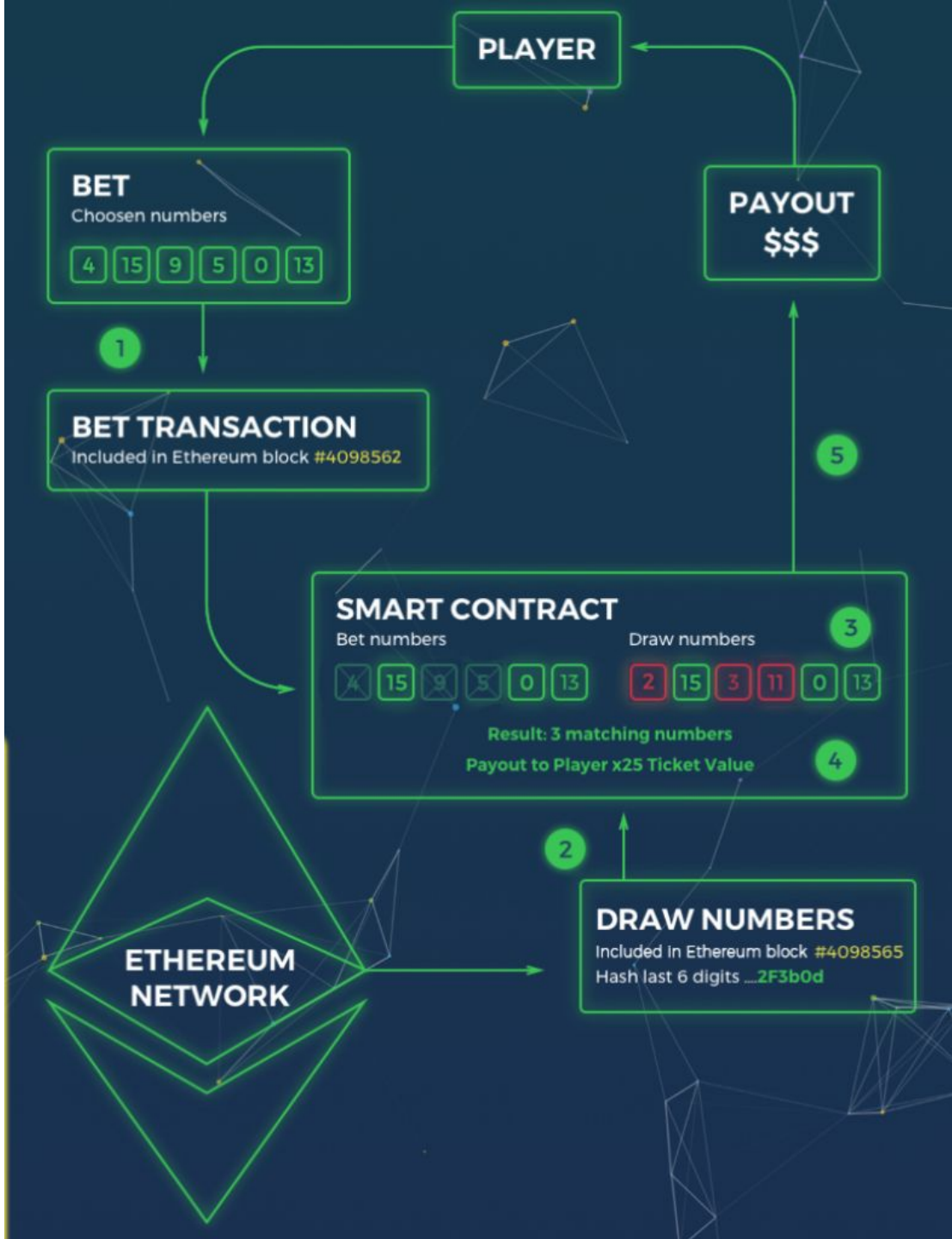

Chips
Kettle Cooked
Doritos
Cheetos
Lay's
Sensit


PLAY
calottery
This is your Lucky Dog!
Scratch & Win!!
SCRATCHERS
calottery
SCRATCHERS
calottery

Need to trust a 3rd party when they have incentive to cheat you

Smart Contract Capabilities

- Can:
  - Receive inputs (in the form of transactions)
  - Execute calculations
  - Store information
  - Send transactions (which can be inputs/messages to other contracts)
- Are autonomous and transparent
- Can interact with each other



https://itsblockchain.com/smartbillions/ https://medium.freecodecamp.org/smart-contracts-for-dummies-a1ba1e0b9575

How do smart contracts run on the blockchain and what are the constraints?

- In the case of Ethereum, smart contracts run on what is known as the Ethereum Virtual Machine (EVM)
- Each and every node runs the EVM and executes the same instructions
- Parallelization leads to massive redundancy at the cost of low performance
- Thus, the main constraint is that all functions for smart contracts have a fee (referred to as gas) which is paid in ethereum
    - This prevents people from effectively ddosing the network with smart contracts that would make the EVM too hard to run in a decent amount of time for low powered computers

http://ethdocs.org/en/latest/introduction/what-is-ethereum.html

What are smart contracts?

-
- Smart contracts are (generally) autonomous and self-executing accounts that act according to their functions
  - Can receive inputs, execute calculations, store information, send transactions (which can be inputs) to other accounts
  - Is essentially an if….then system

https://hackernoon.com/blockchain-for-non-techies-3-smart-contracts-104f77277297

Example of Smart Contract

- You could write the code for a lottery function; users would send a certain amount of funds to this account and using a random characteristic like the nonce of the next block randomly pick an entrant as the winner
- By using a smart contract with auditable code, the lottery is provably fair
- Users can create and deploy these smart contracts on platforms such as ethereum
  - People can then interact with these smart contracts using transactions to those accounts
- Smart contract code for execution is visible to everyone
- Can also create a smart contract to represent a digital token in the form of a ledger smart contract with storage

How do smart contracts run on the blockchain?

- With normal transactions, miners just add the transaction to the chain after briefly verifying them and try to mine the block
- With smart contracts, all miners must execute this same code to to get the same result and then

Nodes must verify and run

Homomorphic Encryption

Tax Returns

Homomorphic Encryption

https://crypto.stanford.edu/craig/easy-fhe.pdf

What is homomorphic encryption?

- Allows someone to process data for you without giving away access to the data itself
- If I want someone else to add the numbers 4,000 and 5,800 for me without them knowing what the numbers are nor what the result is, homomorphic encryption can do that

Homomorphic Encryption

- Hand over an encrypted version of your data to a company
- The company can then process that data using their proprietary algorithms and give you back a result
- You can then decrypt this result and use it as you wish

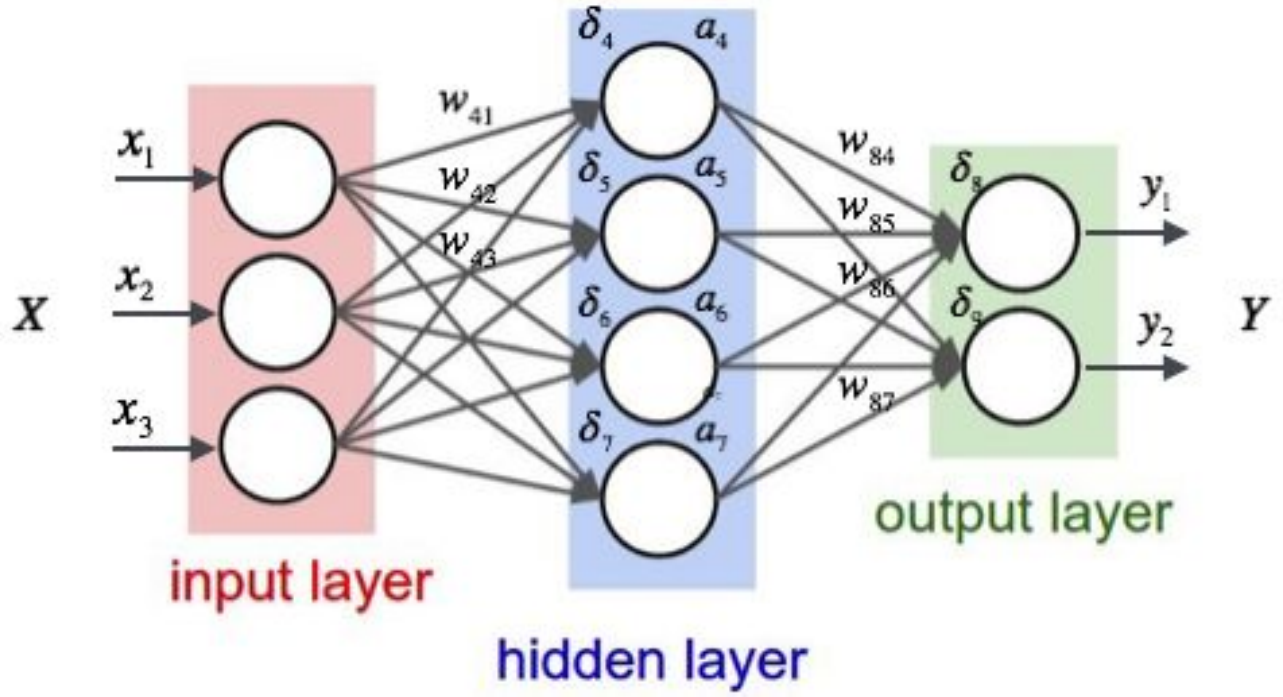
Neural Networks on Distributed Ledgers

How to compute a Neural Network on-chain

- Computing a neural network on 1 smart contract is difficult
- 1 neuron = 1 smart contract address

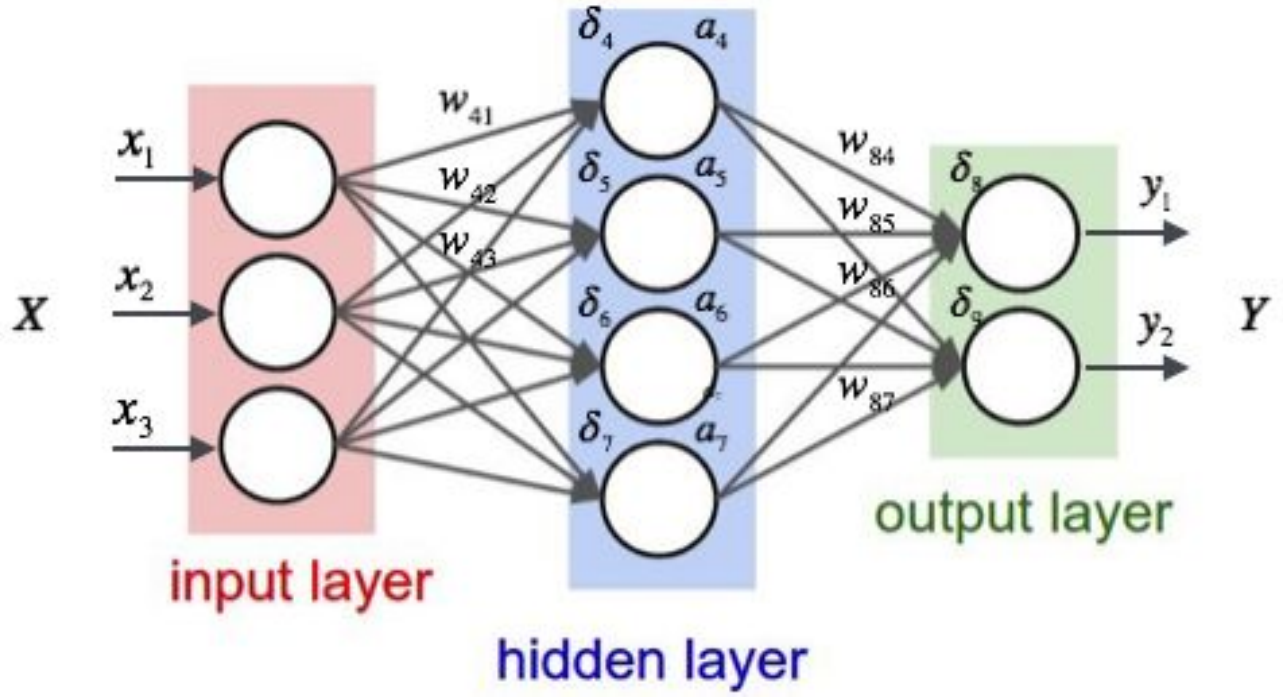
- Each neuron does relativity simple calculations (additions, multiplications)
- The weight values of each neurons & synapses are written in the smart contract
- Neuron can signal other neurons about their calculated weight values (neuron weight x synaptic weight) by transferring tokens

DANN (Distributed Artificial Neural Network)



1. Receive x1 number of tokens
2. (squash)
3. Compute w41, w42, w43, w44 using my synaptic weights
4. Send w41 tokens to δ4, w42 tokens to δ5, ... etc

DANN (Distributed Artificial Neural Network)



1. A user puts in the input values (x1, x2, x3) using tokens (Homomorphic Encryption)
2. Each smart contract neurons interact with the next layer neurons by transferring tokens to them
3. The final output values comes out as the balances of the output layer neurons
4. Token redistribution to neurons with low balances (so they won't run out of tokens)

DANN

- Theoretically, this can be built on the Ethereum Network

- Slow (takes a few block times per contract)
- Expensive (every full node has to compute)

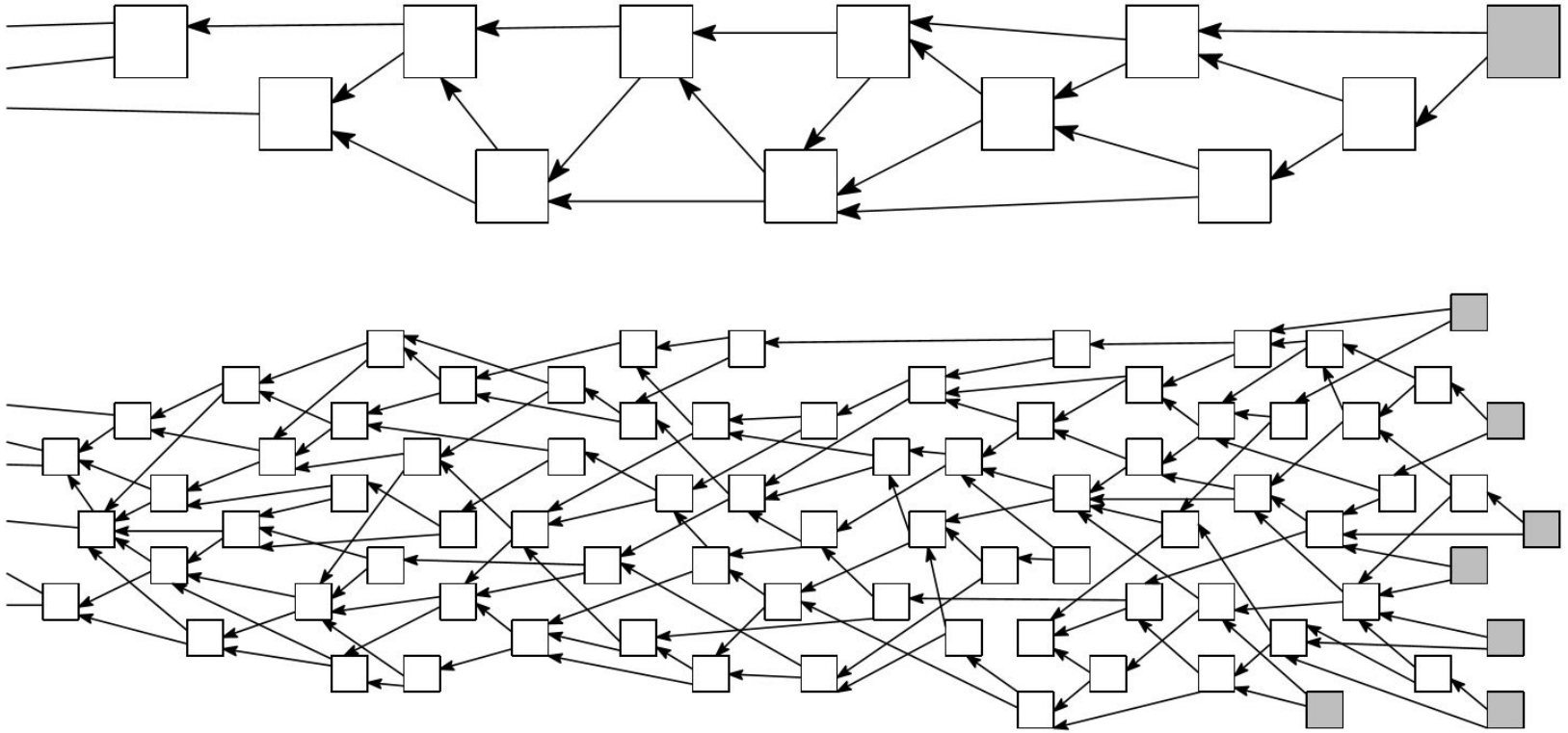
Compute Scaling

- Two types of parties: Validators (a.k.a. miners) and Users

- Public Blockchains
  - Everyone can be a validator / user -> slow
- Private Blockchains
  - Only selected people can be a validator / user -> fast
- Consortium Blockchains
  - Selected people can be validators, but everyone can be a user -> fast
  - Current (e.g. ETH) miners have tens of thousands of GPUs

Compute Scaling

- DAG (Directed Acyclic Graph)



ex) IOTA: in order of include your transaction on the ledger, you have to validate (PoW) two previous transactions
-> Transaction speed (contract computation speed) only limited to bandwidth), Asynchronous

Advantages

- Developers:
  - Write DANN code and announce it to the global network
  - Separation of code development & computation
- Users:
  - Search for AIs that they need and use them
- Miners:
  - A useful way to use computing power
  - Gain fees and block rewards

Future Use

- Global AI marketplace
- Global AI API
- AI DAOs (AIs on the network calling other AIs)
  - Autonomous AI Companies
- AGI (Maybe?)
  - Distributed asynchronous computing