BlockChains, Cryptocurrencies, Payment Systems

Sanjiv R. Das

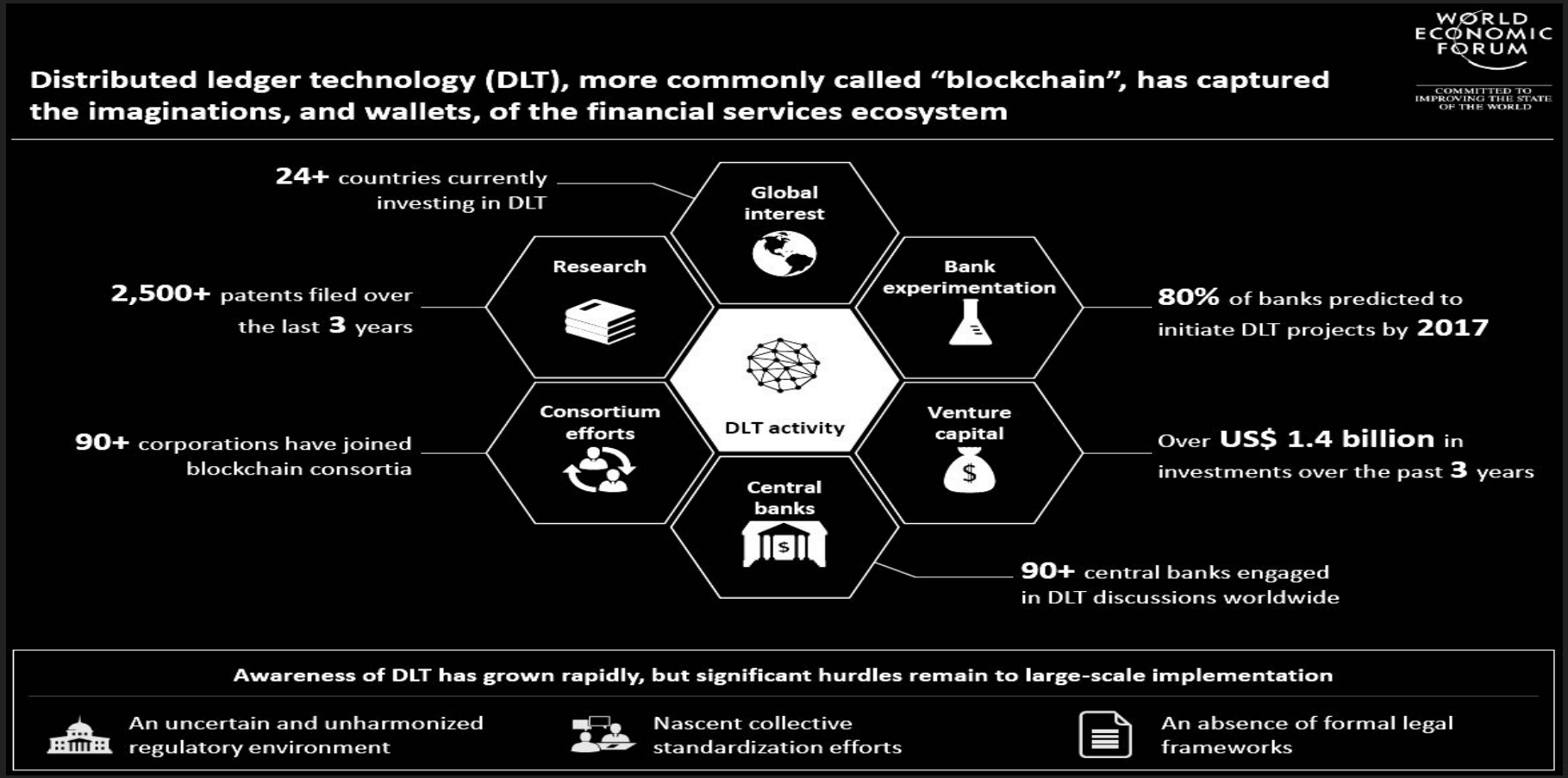
ISB | 2018 | FinTech

Getting Started with Python

- Python tutorial for absolute beginners: http://stackabuse.com/python-tutorial-for-absolute-beginners/
- Instance, class, and static methods: https://realpython.com/blog/python/instance-class-and-static-methods-demystified/
- Jupyter notebook: GetStarted_Python_in_Finance.ipynb

Distributed Ledgers

http://www3.weforum.org/docs/WEF_The_future_of_financial_infrastructure.pdf



Distributed ledger technology has great potential to drive simplicity and efficiency through the establishment of new financial services infrastructure and processes

COMMITTED TO IMPROVING THE STATE OF THE WORLD

The following six key value drivers for DLT were identified through the in-depth examination of nine use cases from across financial services.

Value drivers

1. **Operational simplification**
   DLT reduces / eliminates manual efforts required to perform reconciliation and resolve disputes
2. **Regulatory efficiency improvement**
   DLT enables real-time monitoring of financial activity between regulators and regulated entities
3. **Counterparty risk reduction**
   DLT challenges the need to trust counterparties to fulfil obligations as agreements are codified and executed in a shared, immutable environment
4. **Clearing and settlement time reduction**
   DLT disintermediates third parties that support transaction verification / validation and accelerates settlement
5. **Liquidity and capital improvement**
   DLT reduces locked-in capital and provides transparency into sourcing liquidity for assets
6. **Fraud minimization**
   DLT enables asset provenance and full transaction history to be established within a single source of truth

Applications of distributed ledger technology will differ by use case, each leveraging the technology in different ways for a diverse range of benefits

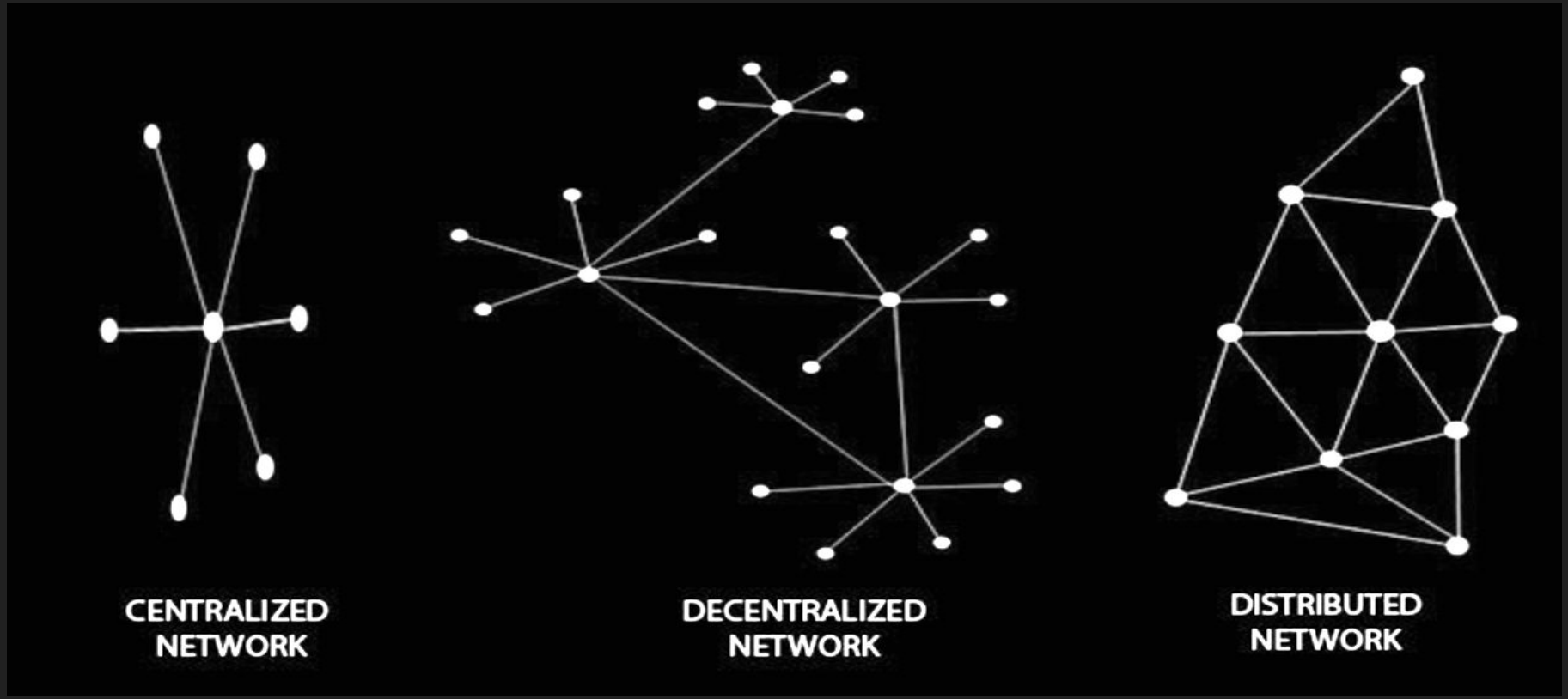
Examples of DLT value drivers and benefits

Use case	Value driver	Benefits
Trade finance	Operational simplification	Enables real-time multi-party tracking and management of letters of credit, and enables faster automated settlement
Automated compliance	Regulatory efficiency improvement	Provides faster and more accurate reporting by automating compliance processes that draw on immutable data sources
Global payments	Settlement time reduction	Enables the near real-time point-to-point transfer of funds between financial institutions (FIs), removing friction and accelerating settlement
Asset rehypothecation	Liquidity and capital improvement	Provides market participants with an improved line of sight into assets, enabling improved risk evaluation and decision-making

What is a Blockchain?

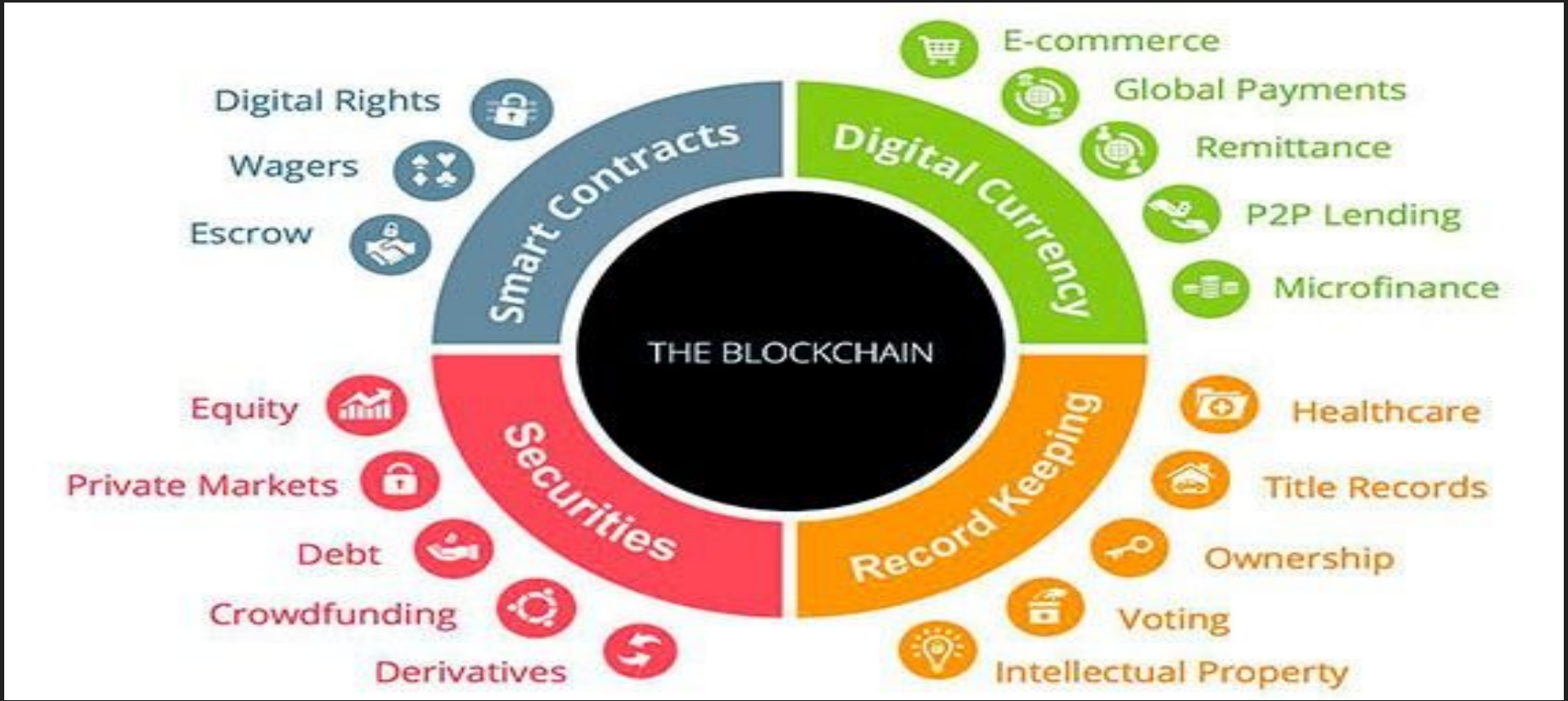
- A digital ledger, maintaining chronological transactions as a public record, i.e., a database, secured and maintained by a large number of nodes worldwide.
- The DIST acronym:
  - D: Decentralized, for validation. Or, Distributed.
  - I: Immutable, once recorded on the blockchain, it cannot be reversed by anyone.
  - S: Secure, meant to be tamper proof.
  - T: Trusted, only valid transactions will enter the blockchain, i.e., there is no double spending. (Trustless.)
- A blockchain is a peer-to-peer (p2p) network. This is the key benefit of the blockchain, it eliminates the middleman. The savings are potentially huge!
- "WTF is The Blockchain? The ultimate 3500-word guide in plain English to understand Blockchain."
  - https://hackernoon.com/wtf-is-the-blockchain-1da89ba19348
  - https://hackernoon.com/understanding-ethereum-a-complete-guide-6f32ea8f5888

Distributed Networks



http://www.datasciencecentral.com/profiles/blogs/introduction-to-blockchains-amp-what-it-means-to-big-data-1

BlockChain Use Cases



https://hackernoon.com/a-rant-about-blockchains-2235b96d64cf

Cryptocurrency

- A medium of exchange.
- A store of value.
- Open: anyone on the internet can transact in this form of currency.
- Privacy and anonymity: to some extent for now, because no regulation or disclosure requirements exist, but users are not entirely untraceable.
- Should satisfy DIST.
- One might consider interpreting CCs as commodities, not currencies, as they are a class of financial asset, more than a medium of exchange.
- Nice writeup on a price analysis of cryptocurrencies: https://blog.patricktriest.com/analyzing-cryptocurrencies-python/
- Physical coins, do not have history or memory of transactions, so cannot be used as a record of contracting.
- Original article: Nakamoto, Satoshi (24 May 2009). "Bitcoin: A Peer-to-Peer Electronic Cash System": https://bitcoin.org/bitcoin.pdf

Bitcoin

- Example: Bitcoin (BTC), attributed to Satoshi Nakamoto (anonymous).
- Latest price: https://www.coindesk.com/price/
- Original paper: https://bitcoin.org/bitcoin.pdf

https://hackernoon.com/all-you-need-to-know-about-cryptocurrencies-an-overview-for-the-savvy-investor-bdc035b14982

A Transaction

- A is sending bitcoin to B in payment.
- The transaction details and an identifier (for the coin) are appended and hashed using the sender's digital signature to create a "message". The digital signature uses strong cryptography and makes the transaction secure.
- This converts a transaction record of arbitrary length into a fixed length.
- Many transactions are collected into a single block. This is also combined with the hash from the previous block to create a current hashed transaction block. The new transaction block has the last block, current block, and a sequence of numbers.
- Bitcoin miners add a number (nonce) to the hashed block and create another hash, and if this hash has several leading zeros then the block is "solved" and the miner gets newly issued BTC as a reward.

A Transaction

- Because it takes several random trials (work) to solve a block, this is compute intensive, but the BTC protocol sets the number of required leading zeros to a specific number to manage the rate at which BTC are issued. Currently the number of leading zeros in the hash is 17 (of a hash length of 256), and a block is solved approximately every 10 minutes.

- Once a block is solved, the transactions are recorded on the blockchain. The number the miner added to make the hash contain the required number of leading zeros is the Proof of Work (POW) required to earn the reward BTCs.

- Proof of Stake: to do

Cryptographic Hash Function Properties

- Computational efficiency. Fast to calculate.
- Collision resistance. It should be hard to find two messages that give the same hash or "digest".
- One-way function. It hides information about the original inputs. You cannot reverse engineer the inputs from the hash.
- Well-distributed. The hash function should generate hash values that are widely distributed, so that it is not easy to find a set of values on the number line where hash values are concentrated, which would make random guessing easier.
- High perturbation variance. Small changes in the bytes of the input message should result in large changes in the resultant hash function. The hash function should be random.

Hash Function Protocols

- MD5: message digest 5 (128 bit algorithm).
- SHA256: secure hashing algorithm, 256 bits. Fixed length.
- Message --> Hash Function --> Digest, tag, hash, fingerprint.

```
import uuid
import hashlib

M = "Investment Management"
print(hashlib.md5(M.encode()).hexdigest())
print(hashlib.sha256(M.encode()).hexdigest())
print(hashlib.sha256(M.encode()))
```

```
897c91670daad7cba272861918d6e06a
f4b67ffa0f010f4c62afafd12708203340d68f3791c83338c0ec06622b8285d1
<sha256 HASH object @ 0x7f3547887558>
```

```
M2 = "InvestmentManagement"
print(hashlib.md5(M2.encode()).hexdigest())
print(hashlib.sha256(M2.encode()).hexdigest())
```

```
0c81dc598a9a859c9ce1ac21ba9a53da
48272b27a8f51f10a890f316007d0f1e2cc7a8cdacd11e464c3e2d9229e21375
```

Digital Signatures

- Digital signature schemes (DSS), e.g., RSA.
- Each user has a signing (private) key *SK* for signing the message *M*.
- User also releases a verification (public) key *VK* to decrypt the message.
- Document + Identity, ensures that only one person can create this digital combination. This creates a signed message
  - *SM = f(M,SK)*
  - where *f(.)* is a hash function.
- Verification function:
  - *g(M,SM,VK) ---> TRUE*
- The current algorithm is the Elliptic Curve Digital Signature Algorithm (ECDSA). See https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm

Example

- Sender Alice (*A*) has *SK(A), VK(A)*, and receiver Bob (*B*) has *SK(B), VK(B)*.

- Suppose *A* has received BTC from past transactions, i.e., from 45 from *C* with *VK(C)*, and 35 from *D* with *VK(D)*, total 80 BTC. These came with digests *d(C), d(D)*.

- Of these 80 BTC, *A* pays *B* 60 BTC, then the combination of *d(C)*, *d(D)*,*VK(B)=60*, *VK(A)=19*, *txn=1* is sent to the P2P network for mining. *A* will pay a 1 coin txn fee which is shown above. Totally *d(C) + d(D) = 80* are inputs of the transaction, and this corresponds to the sum of outputs of the transaction, to *A*, *B*, and the winner of the POW competition for that block.

Double Spending

- When a sender tries to send the same BTC to two or more receivers.
- Not easily possible, because of a clear sequence of timestamps, and two receivers cannot both get priority across blocks. But within a block it could happen, so a block's value is restricted to $1,000,000, which is the maximum loss.
- Transactions are not considered fully verified for 1-2 hours, after which they are so deep in the ledger that constructing another fork to double spend is prohibitively expensive.
- Requires a fork in the blockchain, where the same BTC appears in both branches. However, in the end the branch with the highest POW is accepted and the other one is rejected.
- An attacker must put in the double transaction and then proceed to win the mining competition to approve it, which is too costly and improbable to make it worth it.
- Finally, it assumes that no one else will notice the second propagating fork, which is also hugely unlikely.
- https://www.cryptocoinsnews.com/bitcoin-mining-costs-large-fair/

Size of the Blockchain

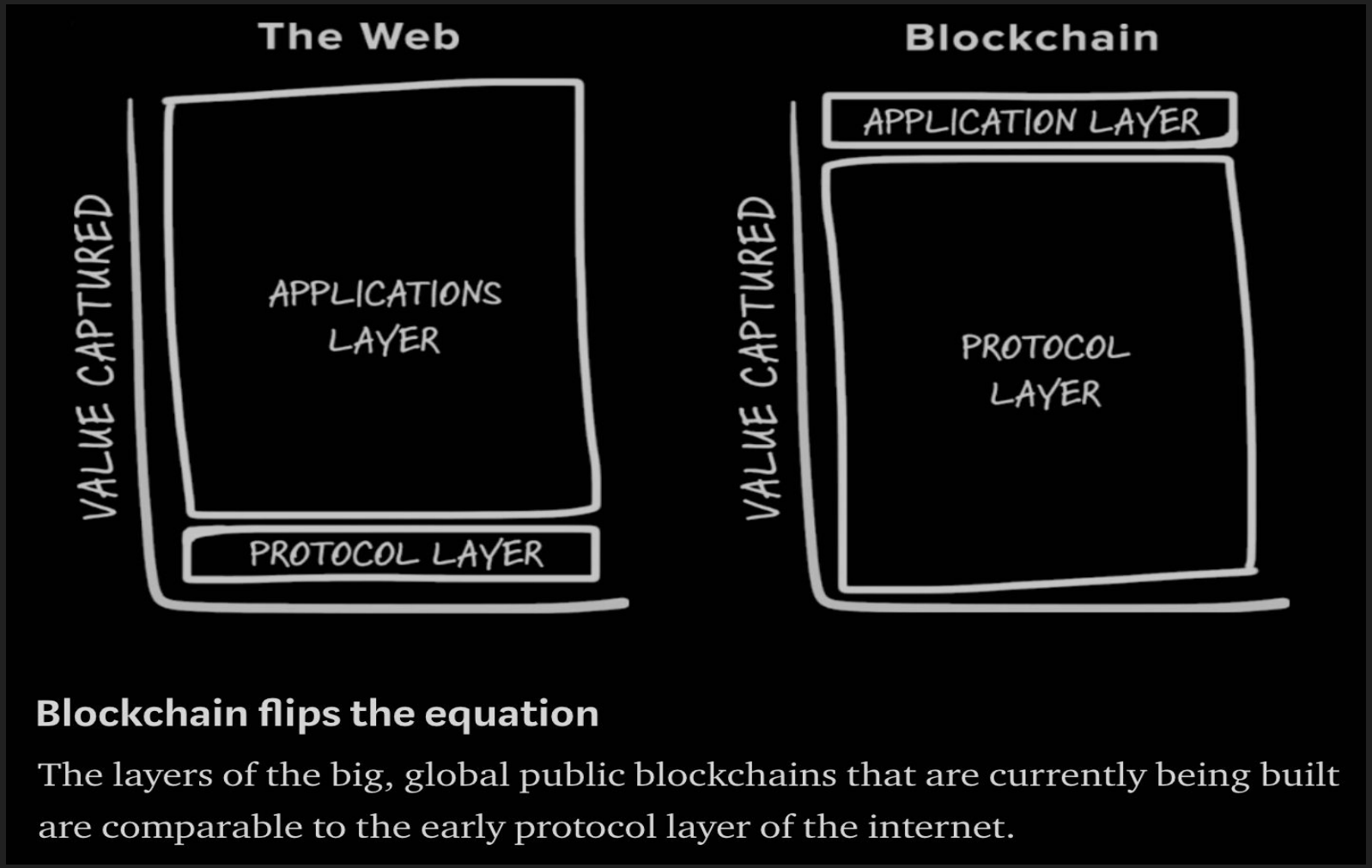
- The blockchain keeps a record of all blockchains since the beginning, so the same coin appears against multiple transactions.
- Pruning the blockchain means keeping only the unspent version of the BTC and not the history for that BTC. In the previous example, it would mean keeping the BTC record for transactions *A=19*, *B=60*, *Miner=1*. These are called UTXO (unspent transaction outputs).
- Storage of the blockchain may be compressed to save space.
- Simplified Verification Protocol. (i) Request verification of a random sampling of transactions, not all. (ii) Verify only block headers.
- Segmented Witness (SegWit2x).
  https://medium.com/@jimmysong/segwit2x-what-you-need-to-know-about-the-2mb-hard-fork-27749e1544ce

Proof of Work (POW) Protocol

- Solving the problem should be hard, verification should be easy. The problem should be in the NP class, i.e., problems that cannot be solved in polynomial time (class P), but may be verified in polynomial time. Note that $P \subseteq NP$, but one of the most important open problems in computer science is whether $P=NP$.
- Take the challenge string c and add to it a proof string p, submit the sum to a hash function $f(c,p)$ to get a hash:
    - $000 \ldots 000xxx…….xxx$
      where the first n bits are zeros.
- The number of leading zeros is the "difficulty" of the POW. Currently $n=17$ out of 256 bits.
- We need approximately 2^n tries of p to get one hash with n zero first bits. Note that $2^{40} \sim 1,000,000,000,000$. Of course, this is easy to verify.
- The BTC system adjusts n over time to make the rate of issuance of BTC constant. Set to solving one block every 10 minutes, which is the latency to confirm a transaction.
- The POW solves what is known as the Byzantine General's problem, where everyone has to agree on a consensus mechanism. In BTC, everyone agrees to pay the miner who does the most work because, barring some luck involved, the person who does the most work usually wins.
    - https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/The-Byzantine-Generals-Problem.pdf

Blockchain: Attributes

- Consensus
- Tamper-proof
- Authentication through better security
- Digital
- The code (technology) is constantly being updated and improved.
- Scalability? Ethereum 5 tps, Bitcoin 7 tps for small txns, 3 tps for big. VISA 2,000 tps, peak 50K tps. The fix? "segregated witness" (SegWit, https://en.wikipedia.org/wiki/SegWit). Ethereum has an upgrade (https://www.coindesk.com/metropolis-today-shifting-plans-ethereums-next-big-upgrade/).
- Proof of stake vs proof of work. http://www.investopedia.com/terms/p/proof-stake-pos.asp
- Reference: https://medium.com/future-crunch/blockchain-is-a-new-model-that-makes-the-existing-model-obsolete-8671ee6dd252



Blockchain flips the equation

The layers of the big, global public blockchains that are currently being built are comparable to the early protocol layer of the internet.

https://medium.com/future-crunch/blockchain-is-a-new-model-that-makes-the-existing-model-obsolete-8671ee6dd252

Bitcoin Supply

- At most 21,000,000 BTC may be issued. No more reward for augmenting the blockchain thereafter.
- Miners then get transaction fees but no new BTC. So, may not need POW, just verify and update the chain, which will save a lot of electricity.
- Fractional coins are allowed. The smallest unit is 10^{-8} BTC, also known as a Satoshi.
- Reward to miners decreases over time. Every 210,000 blocks generated the reward is halved, which happens in about 4 years.
- By 2140 the entire supply of BTC will be generated.
- Number of transactions per second in BTC is 1.5. Visa is 4000 tps, PayPal is 125 tps.

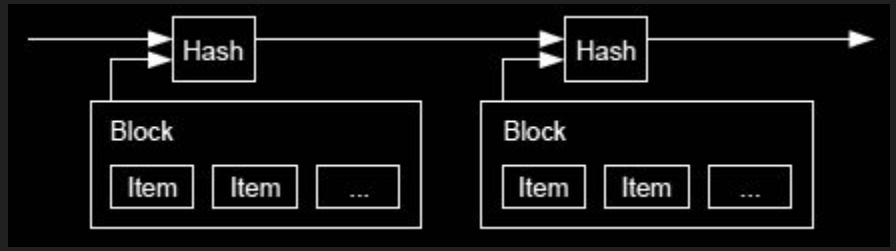
Security

- The original precursor of Bitcoin was Hashcash (http://www.hashcash.org/). Originally designed by Adam Back to counter spam.
- This was the original proof of work algorithm. Because now sending email requires POW, it is costly to spam, and this makes it less likely to be used.
- Security protocols were also provided by Hashcash.
- Back also created sidechains, to promote fungibility of coins. https://gendal.me/2014/10/26/a-simple-explanation-of-bitcoin-sidechains/
- Private blockchains will enhance security, through privileged nodes, and also speed up verification through simplified POW, but at the cost of anonymity and decentralization. https://hbr.org/2017/03/how-safe-are-blockchains-it-depends
- Staggering estimates (\$950 million) of lost bitcoin, not because of the blockchain, but poor wallet security.

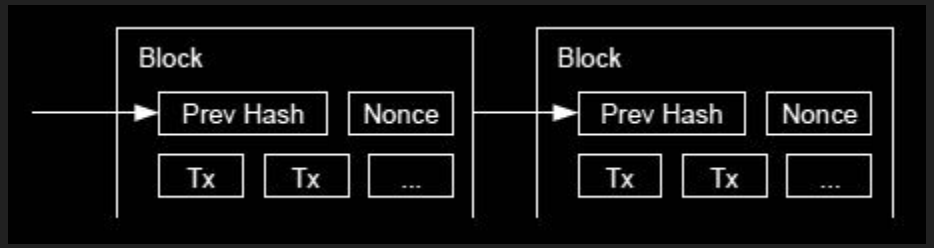
Mining: Costs and Rewards

- Consumes hardware (GPUs) and electricity in server farms. Miners worldwide generate 10^{18} hashes or one exahash every second. With existing hardware, the computation of a billion hashes consumes between 0.1 to 1 Joule of energy. Hence, roughly a billion watts (1GW/sec) are used globally every second to create a valid POW. That's $50,000 per hour in electricity costs. Cost per year ~ $400 million.
- Biggest mining farm in China: https://www.nytimes.com/2017/09/13/business/bitcoin-mine-china.html?mcubz=0&_r=0
- Given a processing rate of 10,000 transactions per hour, the cost per transaction is $5.
- Current value of winning is 12.5 BTC = $50,000 (at 1 BTC ~ $4,000). Per hour people win $300,000 in value.
- January 2009, reward 50 BTC. 2017 reward 12.5 BTC.
- Calibrated so that roughly 2016 blocks are solved every 2 weeks. This is the same as 1 block per 10 minutes, and 6 x 24 x 14 = 2016 (blocks/hr times hours times days).
- Difficulty adjusted by re-specifying the number of leading zeros required.
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2801048

Technical details of Bitcoin



Chain transactions so that the longest chain is the correct one, this obviates double-spending. Forks are disregarded.



To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash, rather than newspaper or Usenet posts.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

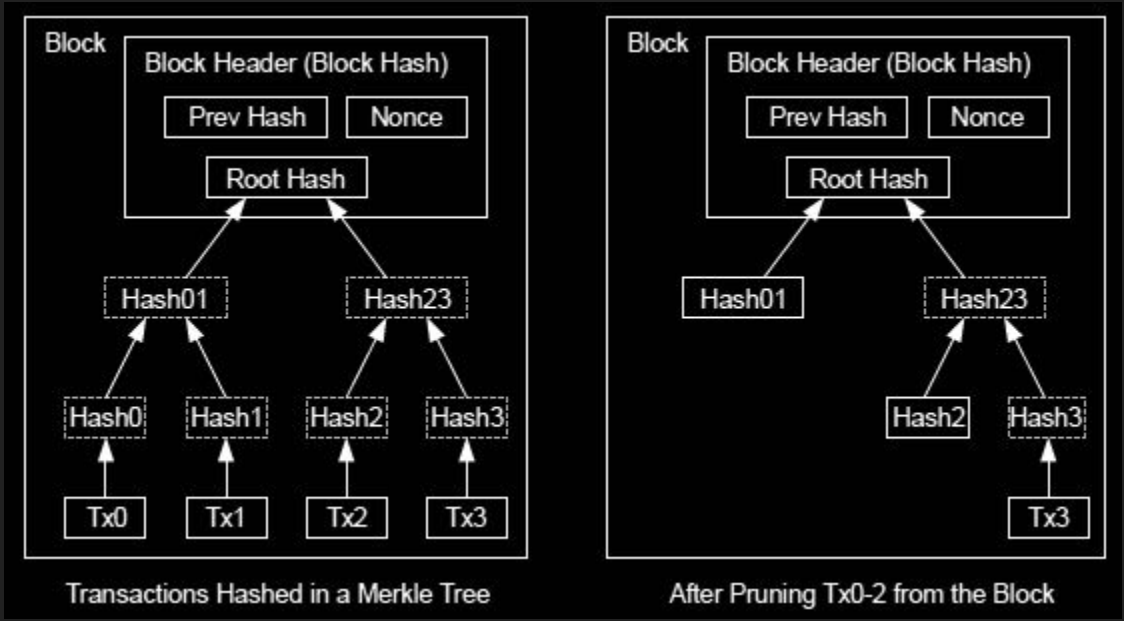
The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

Running the network

The steps to run the network are as follows:

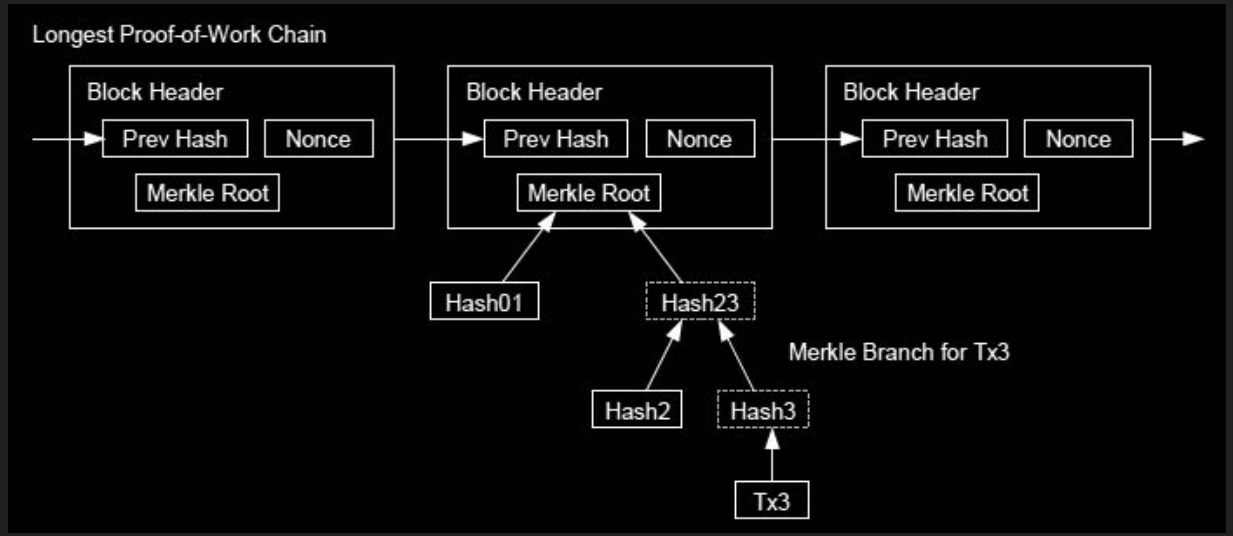
1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
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.
7. Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of- work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.
8. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

Saving disk space from a very long chain



- Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space.
- To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree, with only the root included in the block's hash.
- Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

Simplified Payment Verification



- It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in.
- He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.
- As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker.

Probability of a malicious attack

- The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem.
- Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. The probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain is as follows:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

- Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

Ethereum

- Founder Vitalik Buterin, who wanted to have a programmable blockchain platform. https://www.youtube.com/watch?v=TDGq4aeevgY
- Ethereum: a software platform for blockchains to deploy decentralized applications. (BaaS: Blockchain as a Service.)
- It is a blockchain network, more general than Bitcoin.
- Miners earn Ether, which can be used for transacting on Ethereum, or paying transaction fees.
- Current price of Ether: https://www.coindesk.com/ethereum-price/
- Detailed note: https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369

Smart Contracts

- Supports smart contracts so that they can be settled anonymously.
- May be used to create trading exchanges where smart settlement occurs, but systemic risk may still be managed.
- Real estate applications: title verification, settlement, shared equity in properties, and liquid trading of real estate assets. Decentralized App = DApp
- The technical platform software is known as the Ethereum Virtual Machine (EVM), and it enables any centralized service to be implemented in a decentralized manner. For example, see programming language Solidity. https://blockgeeks.com/guides/how-to-learn-solidity/
- EVM fee is called “gas”.
- The Future: (a) Private blockchains; (b) Consortia blockchains like https://www.r3.com/.

Reference: https://blockgeeks.com/guides/what-is-ethereum/

Ethereum (a programmable blockchain)

- Might be the main protocol driving the “internet of value” (https://hackernoon.com/understanding-ethereum-a-complete-guide-6f32ea8f5888)
- Blockchain is “trustless” - transmittal does not require a trusted third party.
- Smart Contract: (i) , (iii) immutable, (iv) no need for controlling amming language, (vi) So smart contracts
- Ethereum Virtual M of all nodes. Runs user submitted code (sm called “gas” paid in Ether.
- Token is a digital pr demand. Can be created on Ethereu
- ICOs are funded by nd speculative profits for investors.
- DAOs: organization run on smart contracts; it is a DApp in which the developer does not play a role any more. [DApp = decentralized app]

Ethereum: Technical Details

See: https://ethereum.github.io/yellowpaper/paper.pdf

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

EIP-150 REVISION (759dccd - 2017-08-07)

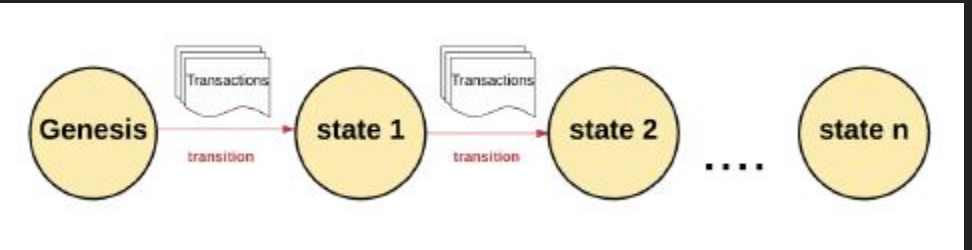
DR. GAVIN WOOD
FOUNDER, ETHEREUM & ETHCORE
GAVIN@ETHCORE.IO

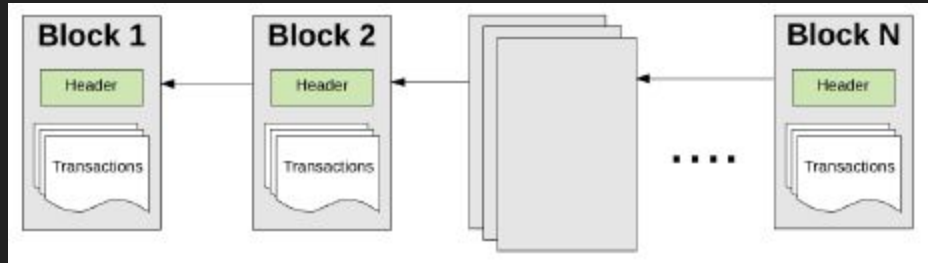
Abstract. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated its utility through a number of projects, not least Bitcoin. Each such project can be seen as a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactional singleton machine with shared-state.

Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resources, each with a distinct state and operating code but able to interact through a message-passing framework with others. We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

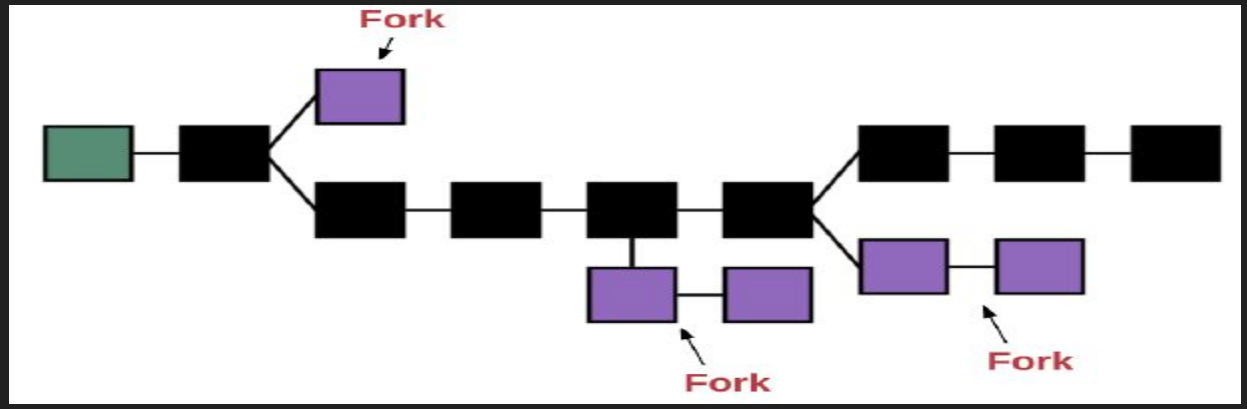
Technical Definitions

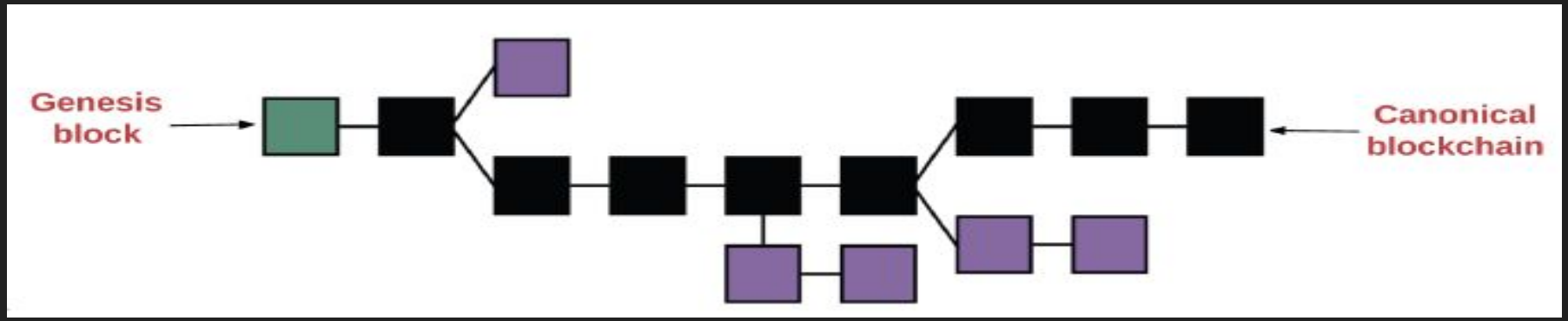
- Blockchain: cryptographically secure transactional singleton machine with a shared state. See: https://ethereum.github.io/yellowpaper/paper.pdf
- State + new information = New state
- Begin with “genesis” state.
- Block = groups of transactions.
- Proof of work = validation of a block
- Ether = reward
- KECCAK-256 = Ethereum hashing protocol
- Wei = ETH x 10^{-18} (1 gwei = 1,000,000,000 Wei)
- Ethereum is a Turing complete language.



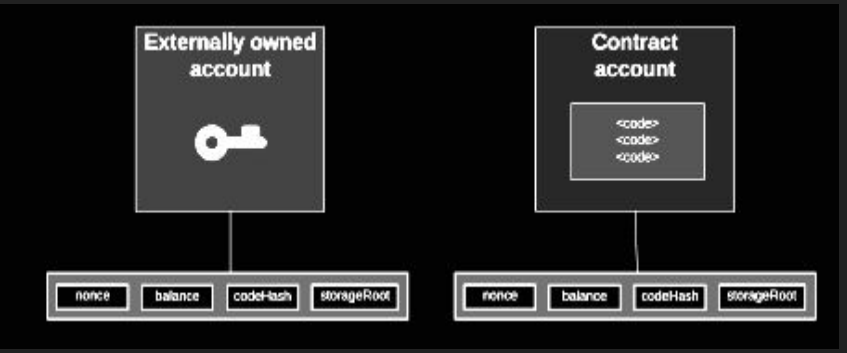


GHOST Protocol = Greedy Heaviest Observed Subtree (in case of a fork) - check for the highest block number

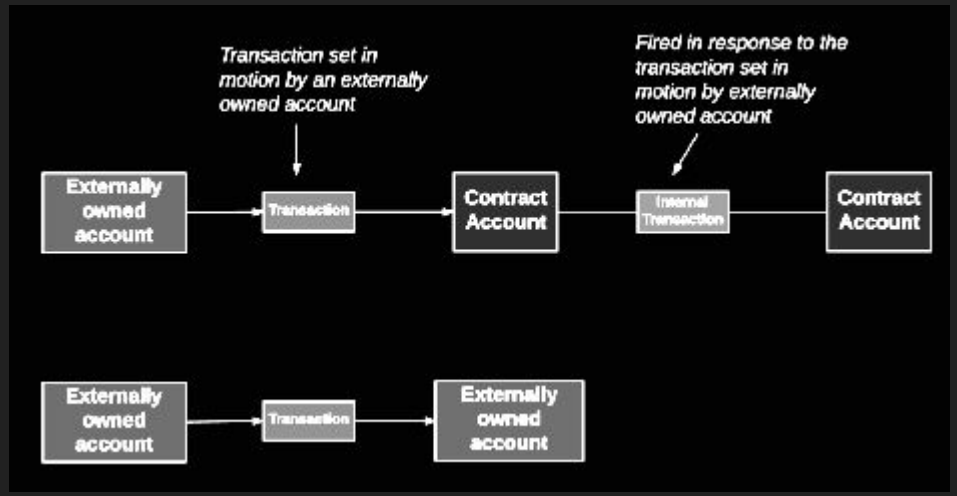




Accounts in Ethereum



- External accounts can initiate transactions.
- Contract accounts can only respond to transactions through an activation of its contract code.
- External to external is usually a value transfer.

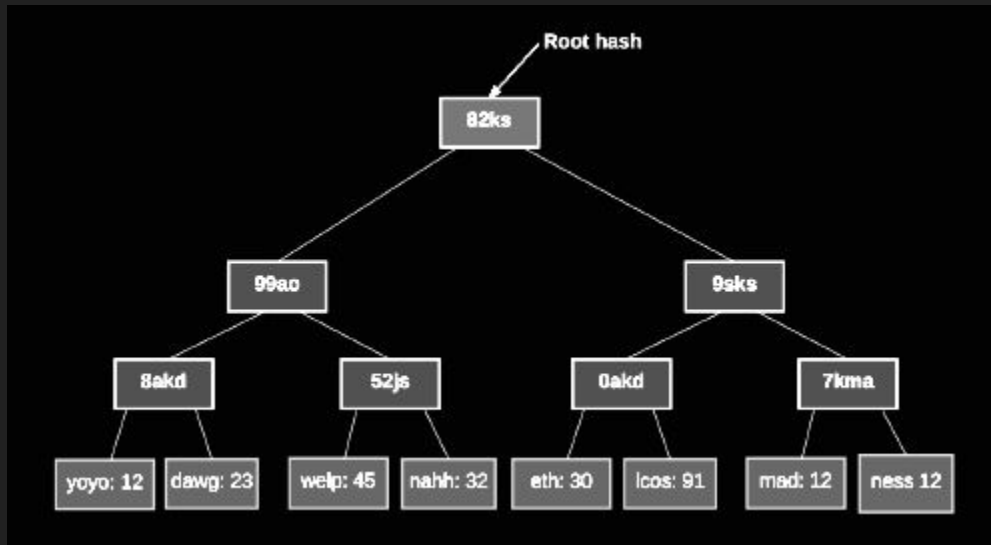


Account State:

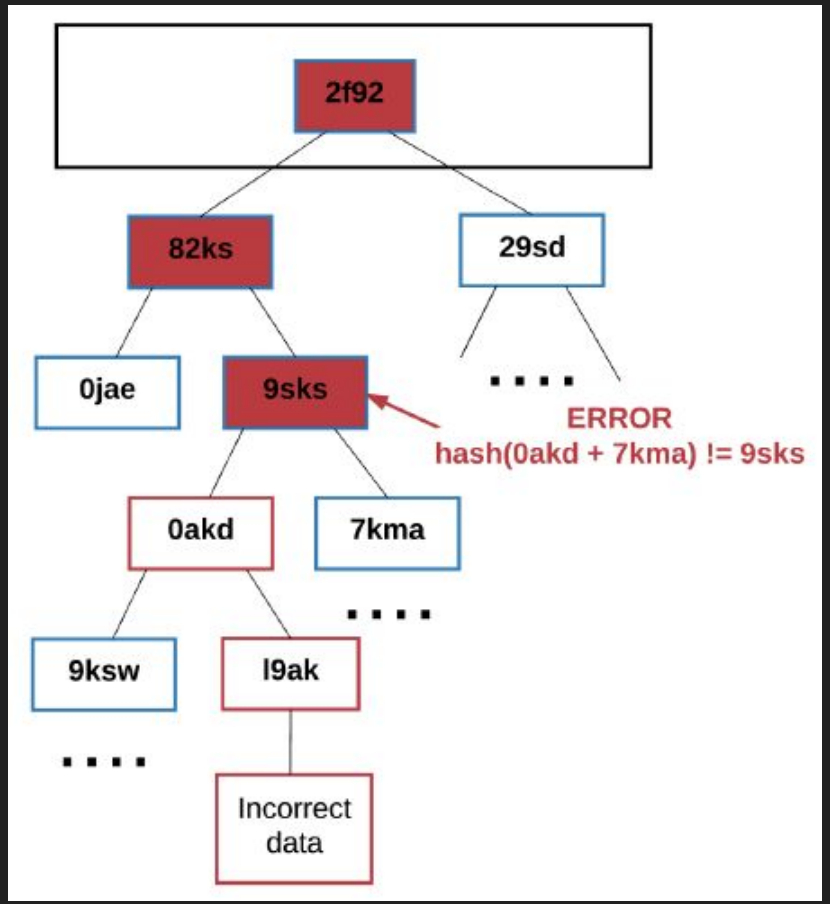
1. Nonce = #txns by external account or #contracts in contract account.
2. Balance (in Wei).
3. storageRoot = hash of root node of a Merkle tree.
4. codeHash = hash of the EVM (Ethereum Virtual Machine) code of the account.

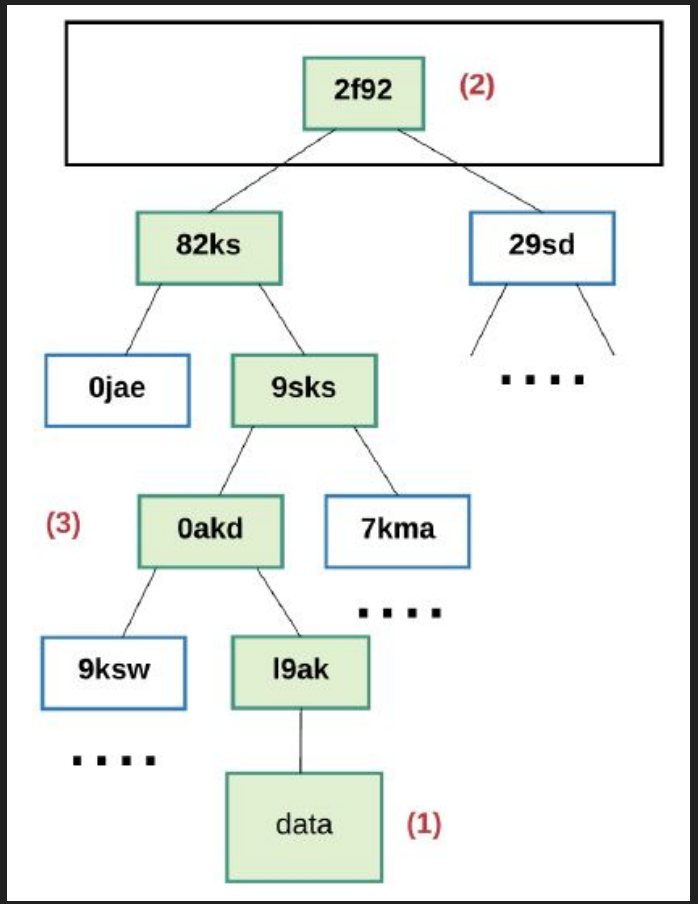
Merkle Patricia Tree

- Keeps the entire global state of Ethereum by hashing all accounts.
- Every child node has a key, which tells you how to go from the root node to the leaf node.
- Merkle tree types: State, Transactions, Receipts.
- Trees reduce data storage as the hash is enough; good for light clients.
- Full nodes and light nodes.
  (Full blockchain vs headers only)



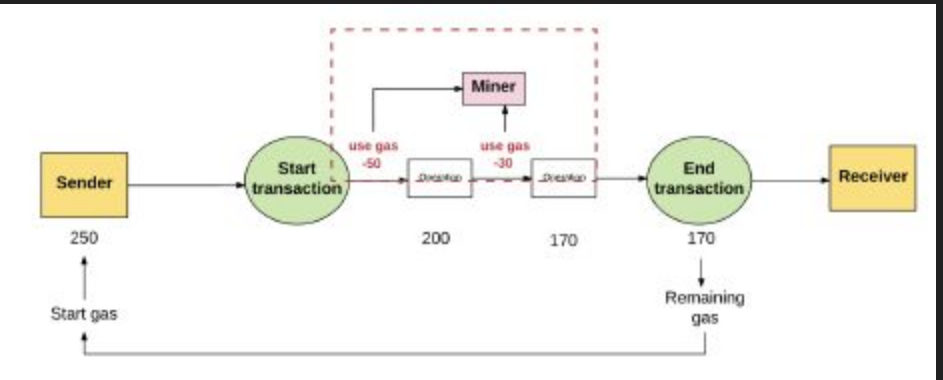
Merkle Proof

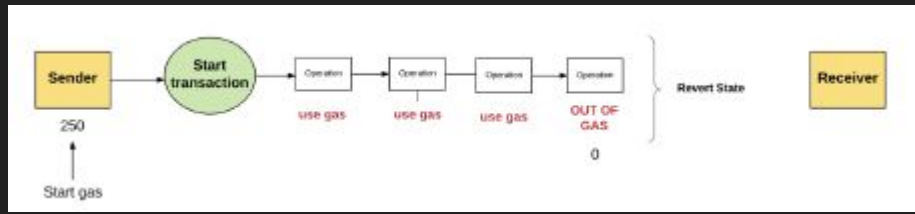




Gas

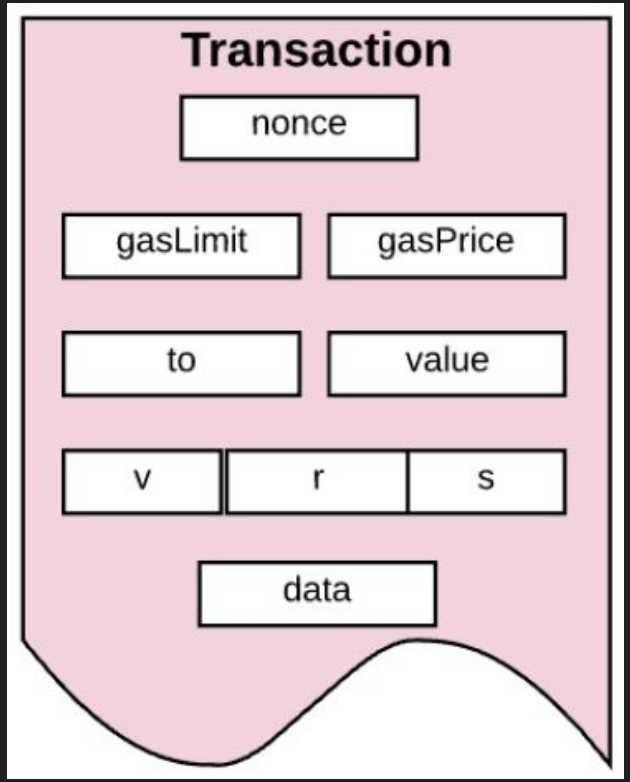
- Gas is the fee to be paid for a transaction. Quoted in gwei per unit of gas. Hence the “g” added to wei.
- Set gas limit in number of units, which expresses the willingness to do the transaction and also determines its priority.
- If insufficient gas available, all gas provided is confiscated by the miner.
- Miners can state minimum gas.
- Fees defray miner costs and also prevent rogue agents from implementing infinite loops.





Transaction components

- nonce: a count of the number of transactions sent by the sender.
- gasPrice: the number of Wei that the sender is willing to pay per unit of gas required to execute the transaction.
- gasLimit: the maximum amount of gas that the sender is willing to pay for executing this transaction. This amount is set and paid upfront, before any computation is done.
- to: the address of the recipient. In a contract-creating transaction, the contract account address does not yet exist, and so an empty value is used.
- value: the amount of Wei to be transferred from the sender to the recipient. In a contract-creating transaction, this value serves as the starting balance within the newly created contract account.
- v, r, s: used to generate the signature that identifies the sender of the transaction.
- init (only exists for contract-creating transactions): An EVM code fragment that is used to initialize the new contract account. init is run only once, and then is discarded. When init is first run, it returns the body of the account code, which is the piece of code that is permanently associated with the contract account.
- data (optional field that only exists for message calls): the input data (i.e. parameters) of the message call. For example, if a smart contract serves as a domain registration service, a call to that contract might expect input fields such as the domain and IP address.



Internal txns (contract account generated) do not have a gas limit.

Blocks

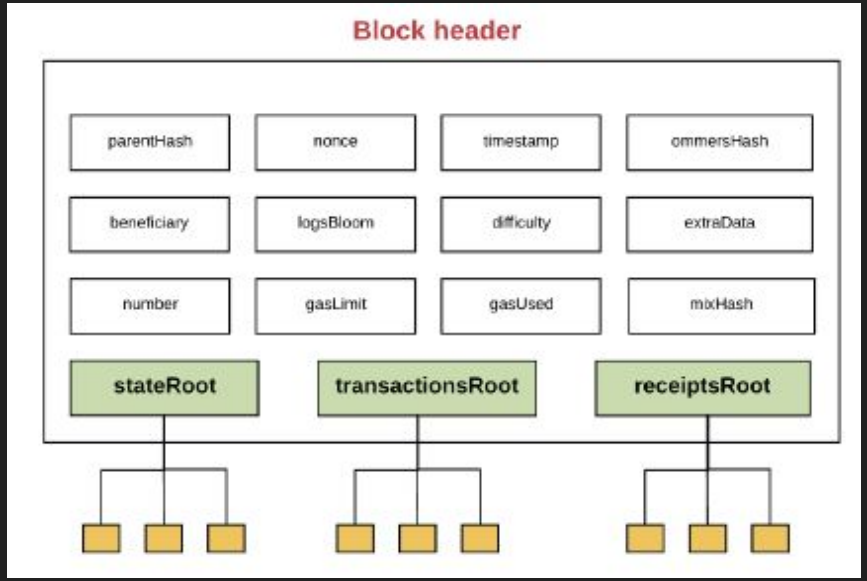
- the block header
- information about the set of transactions included in that block
- a set of other block headers for the current block's Ommers

Ommer = block whose parent is equal to current block's parent's parent.

Blocktime ~ 15 seconds (vs BTC ~10 minutes). Results in orphaned blocks, so they need to be included by miners, and they are rewarded for Ommers.

Block Header

1. parentHash: a hash of the parent block's header (this is what makes the block set a "chain")
2. ommersHash: a hash of the current block's list of ommers bene□ciary: the account address that receives the fees for mining this block
3. stateRoot: the hash of the root node of the state trie (recall how we learned that the state trie is stored in the header and makes it easy for light clients to verify anything about the state)
4. transactionsRoot: the hash of the root node of the trie that contains all transactions listed in this block
5. receiptsRoot: the hash of the root node of the trie that contains the receipts of all transactions listed in this block
6. logsBloom: a Bloom filter (data structure) that consists of log information
7. difficulty: the difficulty level of this block
8. number: the count of current block (the genesis block has a block number of zero; the block number increases by 1 for each each subsequent block)
9. gasLimit: the current gas limit per block gasUsed: the sum of the total gas used by transactions in this block
10. timestamp: the unix timestamp of this block's inception extraData: extra data related to this block
11. mixHash: a hash that, when combined with the nonce, proves that this block has carried out enough computation
12. nonce: a hash that, when combined with the mixHash, proves that this block has carried out enough computation



Logs

To keep track of transactions and messages. Features:

1. the logger's account address,
2. a series of topics that represent various events carried out by this transaction, and
3. any data associated with these events.
4. Logs are stored in a bloom filter, which stores the endless log data in an efficient manner.

A **Bloom filter** is a space-efficient probabilistic data structure, conceived by Burton Howard **Bloom** in 1970, that is used to test whether an element is a member of a set. https://en.wikipedia.org/wiki/Bloom_filter

Transaction Receipt

Generated for every transaction. Contains:

1. the block number
2. block hash
3. transaction hash
4. gas used by the current transaction
5. cumulative gas used in the current block after the current transaction has executed
6. logs created when executing the current transaction

Block Difficulty

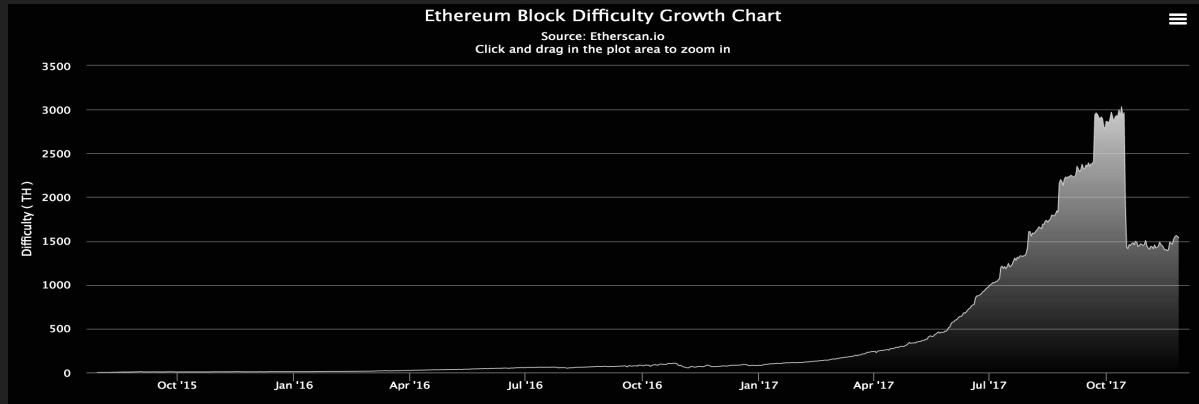
POW algorithm requires calculating the “nonce”, e.g., a hash that contains a specified number of leading zeros (n).

Genesis block had a difficulty of 131,072

Difficulty adjusted to solve a block every 15 seconds.

How difficulty adjustment occurs:
https://dltlabs.io/how-difficulty-adjustment-algorithm-works-in-ethereum/

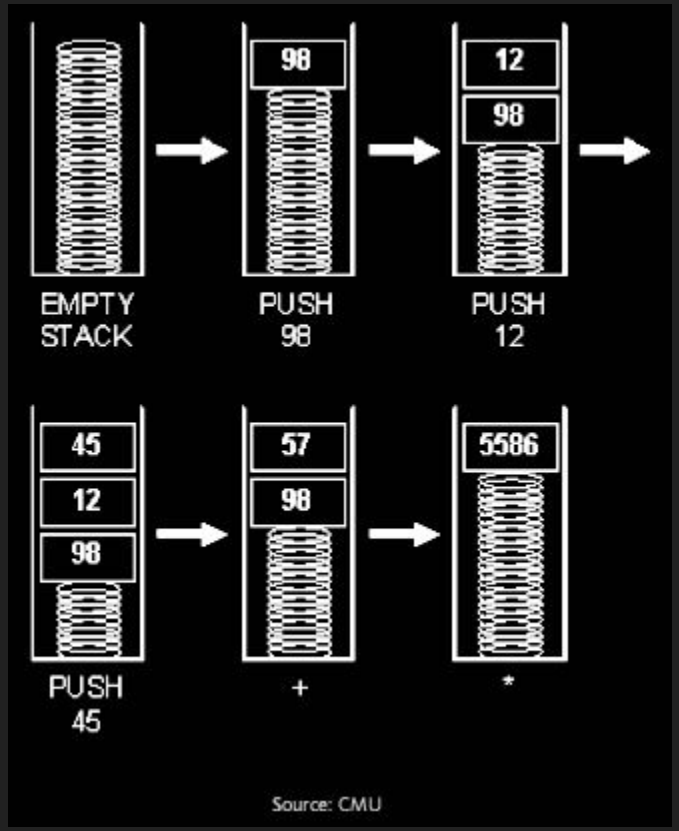
https://etherscan.io/chart/difficulty



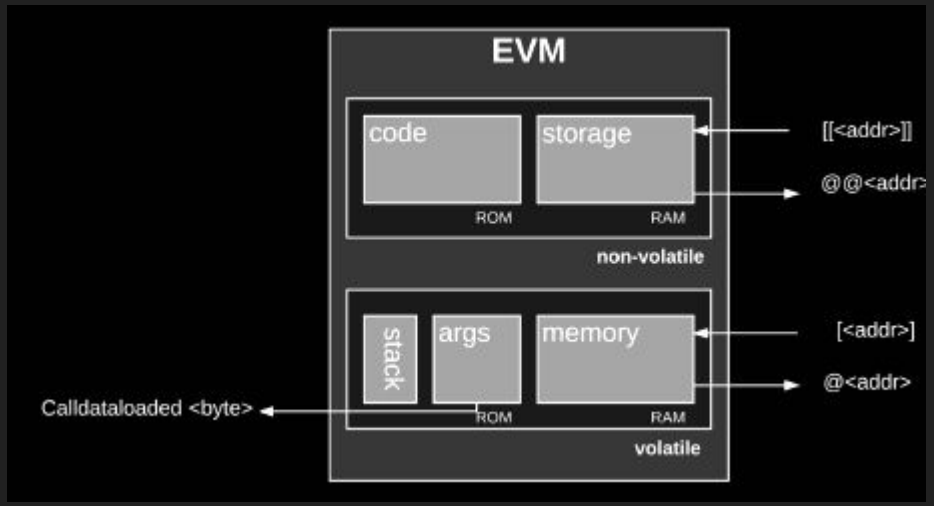
Transaction and Contracts

- The transaction must be a properly formatted RLP. “RLP” stands for “Recursive Length Prefix” and is a data format used to encode nested arrays of binary data. RLP is the format Ethereum uses to serialize objects.
- Valid transaction signature.
- Valid transaction nonce. Recall that the nonce of an account is the count of transactions sent from that account. To be valid, a transaction nonce must be equal to the sender account’s nonce.
- The transaction’s gas limit must be equal to or greater than the intrinsic gas used by the transaction. The intrinsic gas includes: (i) predefined gas of 21,000; (ii) gas fee for data; (iii) additional 32,000 for contract-creating txn.
- If all the above are satisfied, then the various steps of the transaction are executed.
- When txn goes through into a mined block, the state is updated.
- Same process with some additional details to create a contract account.

Ethereum Virtual Machine (EVM)



- EVM is a stack-based vm, uses a LIFO stack to hold temporary values.
- The size of each stack item in the EVM is 256-bit, and the stack has a maximum size of 1024.
- EVM bytecode language
- Contracts written in a high-level language Solidity



Decentralized Autonomous Organization (DAO)

- DAO is a DApp with no remaining developer role.
- Established on the Ethereum blockchain by https://slock.it/.
- $150,000,000 venture fund invested in by thousands. Crowd-sourced VC funding.
- July 2017: SEC rules that tokens issued by DAO are securities and are regulated by federal securities law.
- SEC: Securities, for regulatory purposes, include any "investment of money in a common enterprise with a reasonable expectation of profits to be derived from the entrepreneurial or managerial efforts of others."
- "Money" includes cash and ether.
- Cynical viewpoint: cryptocurrency players are learning the old rules of money in a new setting.

DAOs and DApps:

https://coinmarketcap.com/tokens/views/all/

1174 Cryptocurrencies / 5825 Markets Market Cap: **$172,925,914,247** / 24h Vol: **$5,108,851,225** / BTC Dominance: **54.3%**

Cryptocurrency Market Capitalizations

Market Cap ▾ | Trade Volume ▾ | Trending ▾ | Tools ▾ | Search Currencies

All Tokens

Market Cap: All | Price: All | Volume (24h): All

All ▾ | Coins ▾ | Tokens ▾ | USD ▾ | ← Back to Top 100

#	Name	Platform	Market Cap	Price	Circulating Supply	Volume (24h)	% 1h	% 24h	% 7d
1	OmiseGO	Ethereum	$770,738,741	$7.84	98,312,024	$20,151,600	0.53%	-2.45%	4.57%
2	Tether	Omni	$434,763,607	$0.994742	437,061,677	$213,104,000	-0.01%	-0.42%	-0.56%
3	Ardor	Nxt	$253,739,878	$0.253994	998,999,495	$1,852,580	1.83%	0.10%	37.58%
4	EOS	Ethereum	$231,430,399	$0.565704	409,101,577	$6,057,900	-0.48%	-0.95%	1.09%
5	Augur	Ethereum	$210,359,600	$19.12	11,000,000	$886,163	-0.02%	4.93%	5.41%
6	TenX	Ethereum	$195,958,417	$1.87	104,661,310	$1,757,100	-0.03%	-4.53%	-1.19%
7	SALT	Ethereum	$191,692,737	$3.51	54,675,000	$5,286,490	-0.94%	21.51%	12.53%
8	MaidSafeCoin	Omni	$184,934,638	$0.408648	452,552,412	$1,673,470	-1.23%	2.77%	1.82%
9	Gas	NEO	$184,750,367	$21.49	8,595,839	$779,133	1.54%	-0.73%	-3.48%
10	Golem	Ethereum	$178,313,832	$0.214054	833,032,000	$2,243,150	0.33%	6.85%	12.79%

DAO: Problems

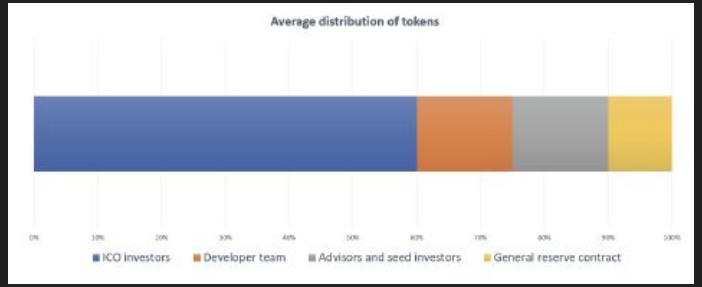
- Machine failure: bugs, downtime.
- Human failure: poorly designed contracts, when all externalities are not thought through, leading to immutable poor incentives and gaming, or outright theft. Lack of clarity in contracting eventually leads to contract failure.
- Problems require consensus resolution, which are hard to implement in a decentralized setting.

The DAO hack of 2016

- Hack exploited a flaw in DAOs code, not any flaw in Ethereum.
- Hard fork to cancel stolen tokens, and make whole the investors who lost their tokens.
- Proponents of a soft fork that retained the immutability of the blockchain fell out with the original Ethereum team, and created a new currency called "Ethereum Classic".

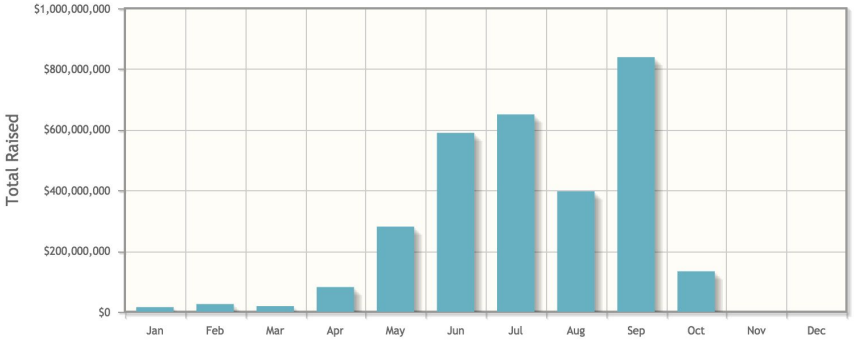
Initial Coin Offerings (ICOs)

- Tokens that are claims on a physical product, business, service, or another ICO (meta coin).
- Exist on top of Ethereum.
- Analogous to a pre-product sale.
- A store of future value.
- Is it a security to be regulated by the SEC?



- 46 ICOs issued in July 2017, $665 million, without registration.
- Exchanges on which ICOs are traded will likely attract SEC regulation.
- Risk of recission: an illegal securities offering, people are legally required to be repaid.
- For ICOs, see Matt Levine's articles:
  - https://www.bloomberg.com/view/articles/2017-07-26/tokens-vaults-ties-and-taxes
  - https://www.bloomberg.com/view/articles/2017-08-08/ico-risks-and-wells-fargo-mistakes
  - https://www.bloomberg.com/view/articles/2017-10-13/icos-marxism-and-credit-reports
  - https://www.bloomberg.com/view/articles/2017-10-17/indexes-insiders-and-icos

Cryptocurrency ICO Stats 2017



Totals raised are grouped by the ICO closing date and are valued using BTC exchange rate at that time. Data correct on 16th October 2017 14:00 UTC

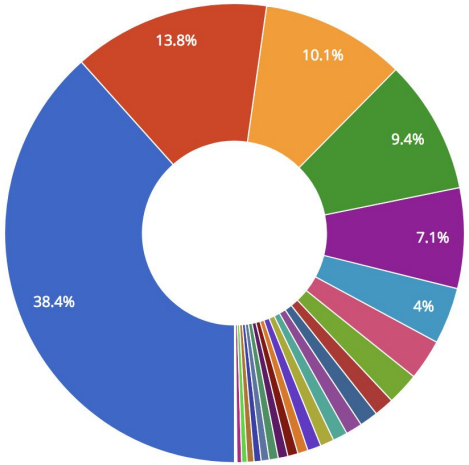
Total Raised: $3,031,120,416

Total Number of ICOs: 201

Top Ten ICOs of 2017

Position	Project	Total Raised
1	Filecoin	$257,000,000
2	Tezos	$232,319,985
3	EOS Stage 1	$185,000,000
4	Bancor	$153,000,000
5	Kin	$97,041,936
6	Status	$90,000,000
7	TenX	$64,000,000
8	MobileGO	$53,069,235
9	KyberNetwork	$48,000,000
10	MCAP	$45,192,400

ICOs by Category 2017



- Infrastructure 38.4% ($1,163,761,392.8)
- Trading & Investing 13.8% ($419,588,257.32)
- Finance 10.1% ($307,380,780)
- Data Storage 9.4% ($286,222,856)
- Payments 7.1% ($214,367,821.36)
- Gaming & VR 4.0% ($120,886,334.9)
- Gambling & Betting 2.9% ($86,825,095.92)
- Commerce & Advertising 2.3% ($68,750,549)
- Identity & Reputation 1.4% ($42,359,520)
- Art & Music 1.3% ($39,310,901)
- Real Estate 1.2% ($35,132,362)
- Events & Entertainment 1.0% ($31,302,122.5)
- Legal 1.0% ($29,368,234)
- Energy & Utilities 1.0% ($28,866,928)
- Social Network 0.7% ($21,843,495.52)
- Communications 0.7% ($21,712,418)
- Mining 0.7% ($20,647,239)
- Drugs & Healthcare 0.6% ($19,368,779.4)

1/2

https://www.coinschedule.com/stats.php

Hedge Funds on the Blockchain

- Numerai: https://medium.com/numerai/an-ai-hedge-fund-goes-live-on-ethereum-a80470c6b681
- Trading algorithms on Numerai will accept Numeraire, a cryptocurrency developed by Numerai so that the algorithms and payments will all run on the Ethereum platform and remain within the same ecosystem, and will not have to interact with Bitcoin.
- Destroying market efficiency by running machine learning algorithms on encrypted data: https://medium.com/numerai/encrypted-data-for-efficient-markets-fffbe9743ba8
- First hedge fund that gives its data away for free with structure-preserving encryption, and allows open participation by data scientists around the world.
- Homomorphic encryption. https://en.wikipedia.org/wiki/Homomorphic_encryption
  https://www.theregister.co.uk/2016/08/16/researchers_crack_homomorphic_encryption/

Hyperledger (https://www.hyperledger.org/)

- Open source blockchain technology, created by the Linux Foundation (https://en.wikipedia.org/wiki/Linux_Foundation) in December 2015.
- Not a cryptocurrency platform, only for blockchains. Not focused on cryptosecurity.
- Various projects and tools in Hyperledger: Burrow (client + EVM); Sawtooth (consensus mechanisms); Iroha (mobile); Fabric (blockchain platform); Cello (deployment toolkit); Composer (blockchain package manager); Explorer (analytics); Indy (cross platform identity management). See https://www.hyperledger.org/projects
- Technical paper: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

Bitcoin Futures

An arbitrage? Efficient (costs of storage are high) or merely an inefficiency?
https://www.bloomberg.com/view/articles/2017-12-12/bitcoin-arbitrage-and-tax-math

BTC 1-month Futures trade at $1000 more than spot! Margin is 40%.

MARGIN REQUIREMENTS – Cboe FUTURES EXCHANGE

Effective 12-06-2017

Contract	Speculative Customer Initial[1]	Customer Maintenance[2] Hedger & TPH[3] Initial Hedger & TPH Maintenance	Spread[4] Speculative Customer Initial	Spread Customer Maintenance Spread Hedger & TPH Initial Spread Hedger & TPH Maintenance
Cboe Volatility Index (VX) – Monthly Expirations[5 6]				
Dec.2017	$6,215	$5,650	See the VX table below.	See the VX table below.
Jan.2018	4,015	3,650		
Feb.2018	4,015	3,650		
Mar.2018	2,860	2,600		
Apr.2018	2,860	2,600		
May.2018	2,860	2,600		
Jun.2018	2,530	2,300		
Jul.2018	2,530	2,300		
Aug.2018	2,530	2,300		
Cboe Volatility Index (VX) – Weekly Expirations[7 8]	$6,215	$5,650	$2,750	$2,500
Cboe Russell 2000 Volatility Index (VU)[9]				
Dec.2017	$3,465	$3,150	See the VU table below.	See the VU table below.
Jan.2018	2,464	2,240		
Feb.2018	2,464	2,240		
Mar.2018	2,464	2,240		
Cboe Bitcoin (USD) Futures (XBT)				
All Contracts	44% of the current daily settlement price	40% of the current daily settlement price	1.10 (110%) x Spread Customer Maintenance	The net difference between the outright customer maintenance margin requirements on each long and short contract **plus** For each spread, a spread charge equal to 5% of the daily settlement price that is the greatest among all XBT futures contracts available for trading

The Future

- On August 13, 2017, the price of Bitcoin (BTC) surged past $4,000. This was a 20% increase over the previous week, after a plan to speed up trade execution was agreed up. As of date, it was up 300% from the beginning of the year.
- As of Sep 2017, from the start of the year, cryptocurrency market cap has grown from USD 18 billion to USD 135 billion.
- The new solution denoted SegWit2x has been a bone of contention in the BTC community. Groups opposed to the new solution have spun off with a new coin called Bitcoin Cash.
  https://www.bloomberg.com/news/articles/2017-08-14/bitcoin-surges-past-4-000-as-speed-breakthrough-to-fuel-spread
- BTC daily volatility ~5%, which is very high. The volatility of gold averages around 1.2%, while other major currencies average between 0.5% and 1.0%. Tech stocks are around 1-2%. Price stabilization CreamCoin. https://creamcoin.com/

Ref: http://hbswk.hbs.edu/item/do-bitcoin-and-digital-currency-have-a-future

WORLD ECONOMIC FORUM

Agenda Initiatives Reports Events About

The future of financial infrastructure: An ambitious look at how blockchain can reshape financial services

Overview

- **Business is growing fast and steadily :** The global payments volume is increasing at an approximate rate of 5% yearly worldwide and will reach an estimated US$ 601 billion in 2016.[1] Revenue is growing in all regions, especially in Asia where China will likely surpass Brazil as the third largest payment area after the United States and the Eurozone[2, 3]
- **Profit margins are high:** The average cost to the final customer (money sender) is 7.68% of the amount transferred
- **Newcomers are arriving:** Non-bank transactions are reaching up to 10% of the total payments volume[2]

Payments: Global Payments.....

Insurance: P&C Claims Processing.....

Deposits and Lending: Syndicated Loans.....

Deposits and Lending: Trade Finance.....

Capital Raising: Contingent Convertible ("CoCo") Bonds..

Investment Management: Automated Compliance.....

Investment Management: Proxy Voting.....

Market Provisioning: Asset Rehypothecation.....

Market Provisioning: Equity Post-Trade.....

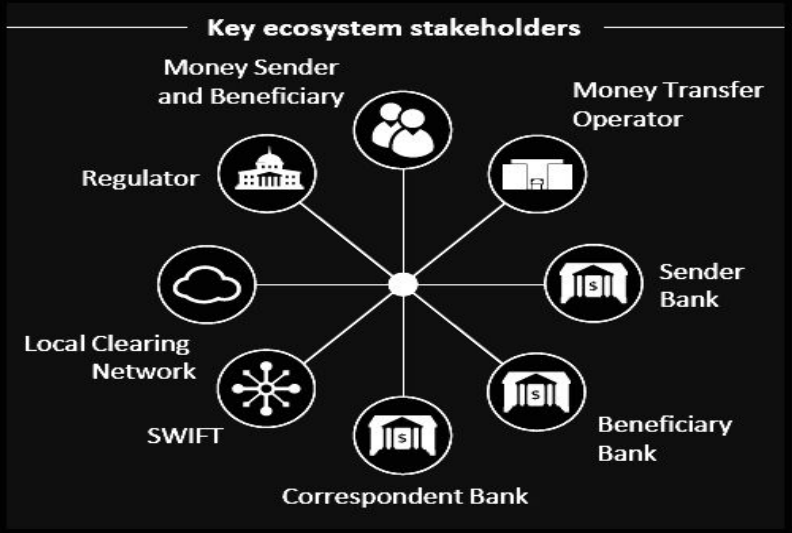
https://www.weforum.org/reports/the-future-of-financial-infrastructure-an-ambitious-look-at-how-blockchain-can-reshape-financial-services

Payment Systems

Current-state background

A payment refers to the process of transferring value from one individual or organization to another in exchange for goods, services or the fulfillment of a legal obligation. Global payments are an expansion of that concept, in which payments can be completed across geographical borders through multiple fiat currencies.

Key ecosystem stakeholders



Overview

- **Business is growing fast and steadily :** The global payments volume is increasing at an approximate rate of 5% yearly worldwide and will reach an estimated US$ 601 billion in 2016.[1] Revenue is growing in all regions, especially in Asia where China will likely surpass Brazil as the third largest payment area after the United States and the Eurozone[2, 3]
- **Profit margins are high:** The average cost to the final customer (money sender) is 7.68% of the amount transferred
- **Newcomers are arriving:** Non-bank transactions are reaching up to 10% of the total payments volume[2]

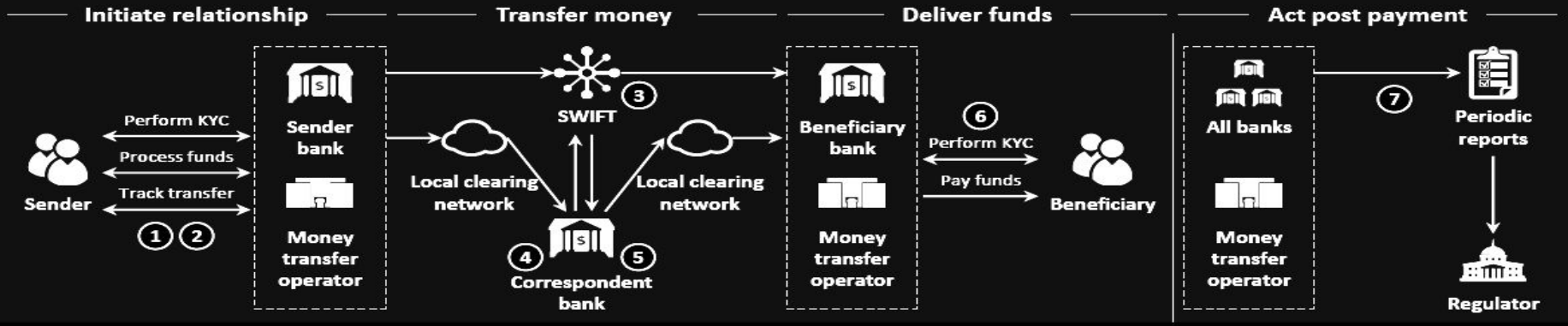
The focus of this use case is on low value–high volume payments from an individual/business to an individual via banks or money transfer operators. These transfers are more commonly known as remittances

1. Migration and Remittances Factbook 2016, World Bank, 2016.
2. Top 10 Trends in Payments in 2016, Capgemini, 2016.
3. Global Payments 2015: A Healthy Industry Confronts Disruption, McKinsey & Company, 2015.

Global Payments

Current-state pain points

WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD



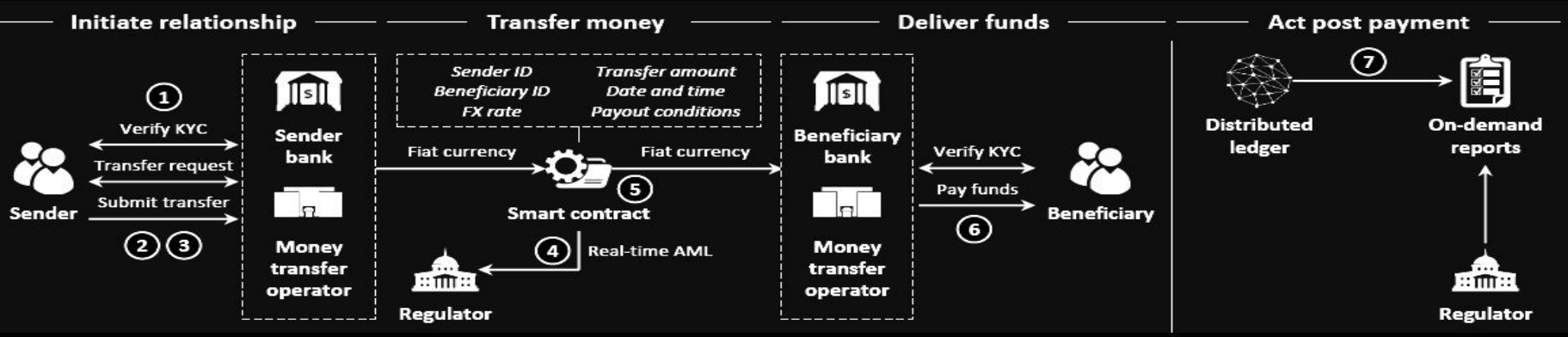
Current-state pain points

1. **Inefficient onboarding:** information about the sender and beneficiary is collected via manual and repetitive business processes
2. **Vulnerable KYC:** limited control exists over the veracity of information and supporting documentation, with various maturity levels across institutions
3. **Cost and delay:** payments are costly and time consuming depending on route
4. **Error prone:** information is validated per bank/transaction, resulting in high rejection rate
5. **Liquidity requirement:** banks must hold funds in nostro accounts, resulting in opportunity and hedging costs
6. **Vulnerable KYC:** similar to #2, limited control exists over the veracity of information and supporting documentation, with various maturity levels across institutions
7. **Demanding regulatory compliance:** due to various data sources and channels or origination, regulatory reports can require costly technology capabilities in addition to complex business processes (often supported by multiple operation teams)

Global Payments

Future-state process depiction

WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD

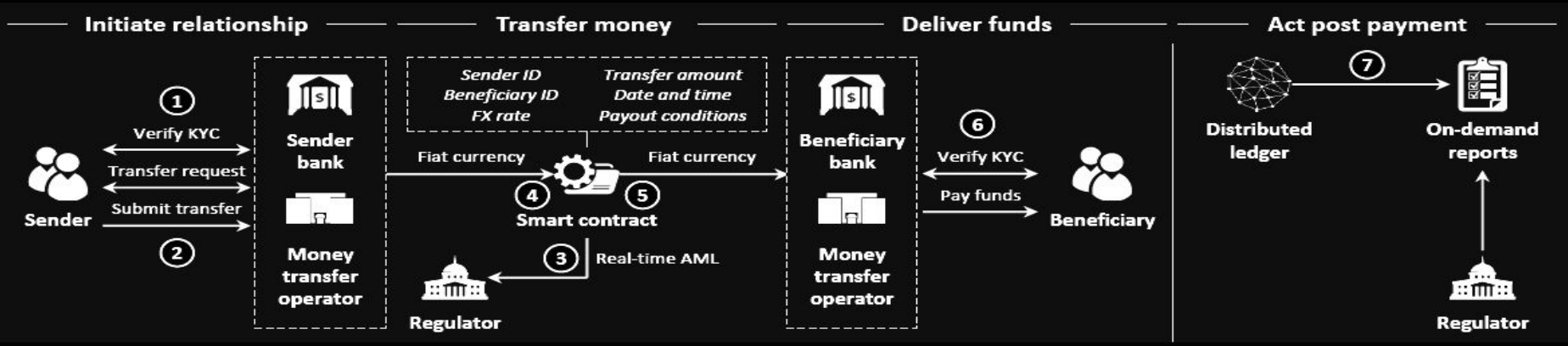


Future-state process description

1. Trust between the sender and a bank or money transfer operator is established either via traditional KYC or a digital identity profile
2. A smart contract encapsulates the obligation to transfer funds between sender and beneficiary
3. The currency conversion is facilitated through liquidity providers on the ledger
4. The regulator can monitor transactions in real time and receive specific AML alerts through a smart contract
5. A smart contract enables the real-time transfer of funds with minimal fees and guaranteed delivery without the need for correspondent bank(s)
6. Funds are deposited automatically to the beneficiary account via a smart contract or made available for pickup after verifying KYC
7. The transaction history is available on the ledger and can be continuously reviewed by regulators

Global Payments
Future-state benefits

WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD



Future-state benefits

1. **Seamless KYC:** leveraging the digital profile stored on DLT establishes trust and authenticates the sender
2. **FX liquidity capabilities:** through smart contracts, foreign exchange can be sourced from participants willing to facilitate the conversion of fiat currencies
3. **Real-time AML:** regulators will have access to transaction data and can receive specific alerts based on predefined conditions
4. **Reduced settlement time:** cross-border payments can be completed in real time
5. **Cost savings:** with fewer participants, the improved cost structure can generate value
6. **Seamless KYC:** leveraging the digital profile stored on DLT establishes trust and authenticates the beneficiary
7. **Automated compliance:** the regulator will have on-demand access to the complete transaction history over the ledger

Global Payments
Conclusion

WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD

Summary

- **Real-time settlement:** enabling banks can fulfil and settle international money transfers in real time, while increasing profitability via a reduction in liquidity and operations costs
- **Reduced fraud:** transparent and immutable data on DLT can reduce fraudulent transactions to a fraction of what they are today
- **Development of digital obligations:** smart contracts can be used to capture obligations among FIs in order to ensure that appropriate funds are exchanged, eliminating operational errors

Outlook

- SWIFT is implementing a "Global Payments Innovation Initiative" to facilitate global payments with transparent fees and same-day funds delivery but this initiative does not employ DLT
- Currently, the adoption of DLT for global payments by incumbent banks is limited, although concrete initiatives are occurring in North America and Europe across retail and wholesale banking
- Opportunities exist for regulators to assess and promote the viability of prototypes and future implementations within current regulatory frameworks

Key takeaways

- **Challenge correspondent banks:** DLT has the potential to disrupt the role of dedicated banks that act as gateways to international fund transfers
- **Allow direct interaction between sender and beneficiary banks:** DLT can give direct access to most if not all relevant destinations for adopting banks and money transfer operators
- **Enable micropayments:** DLT can make low-value transactions more feasible to FIs as cost structures are modified

Unanswered questions

- **Initiatives:** Will retail and wholesale banking initiatives merge towards common DLT implementation despite competing interests?
- **Volatility:** Is there a role for cryptocurrencies as a bridge asset to facilitate FX?
- **SWIFT:** What role will SWIFT play in enabling DLT-based global payments?

Crypto Portfolios

Download data and form portfolios.
https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory

For all data on various cryptocurrencies: https://coinmarketcap.com/

See also: https://www.coindesk.com/

See the Jupyter notebook: CryptoPortfolios.ipynb

Start up Python and load libraries

Cryptocurrency Portfolios

Sanjiv Das

```
%pylab inline
import pandas as pd
import datetime as dt
```

```
Populating the interactive namespace from numpy and matplotlib
```

Read in the data

```
df = pd.read_csv("CryptoData/bitcoin_price.csv")
df = df[["Date","Close"]]
```

```
cc = ["dash","ethereum","iota","litecoin","monero","nem","neo","numeraire","omisego","qtum","rip
for ccy in cc:
    fn = "CryptoData/" + ccy + "_price.csv"
    temp = pd.read_csv(fn)
    temp = temp[["Date","Close"]]
    temp.rename(columns={'Date':'Date','Close':ccy})
    df = pd.merge(df,temp,how='outer',on='Date')
```

```
df.columns = append(['Date','bitcoin'],cc)
df = df.iloc[::-1]
df.head()
```

	Date	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum	ripple	stratis	wave
1654	Apr 28, 2013	134.21	NaN	NaN	NaN	4.35	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1653	Apr 29, 2013	144.54	NaN	NaN	NaN	4.38	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Apr														

```
df.tail()
```

	Date	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum	ripple
4	Nov 03, 2017	7207.76	278.59	305.71	0.393126	56.18	87.99	0.171728	26.82	12.48	6.73	10.38	0.208133
3	Nov 04, 2017	7379.95	274.68	300.47	0.368643	55.04	87.30	0.170845	26.49	11.47	6.45	10.05	0.203750
2	Nov 05, 2017	7407.41	273.17	296.26	0.350084	54.75	86.35	0.180309	26.38	10.95	6.29	10.13	0.202055
1	Nov 06, 2017	7022.76	275.68	298.89	0.373939	55.17	102.92	0.186642	26.32	10.78	6.48	10.44	0.205990
0	Nov 07, 2017	7144.38	293.38	294.66	0.383096	61.30	99.76	0.180246	26.23	10.60	6.33	11.21	0.210354

```
df.shape
```

```
(1655, 15)
```

```
df.describe()
```

	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum
count	1655.000000	1363.000000	824.000000	148.000000	1655.000000	1266.000000	952.000000	425.000000	138.000000	117.000000	168.000000
mean	826.285541	39.117070	69.098058	0.495890	10.325734	12.927654	0.043495	7.576292	25.027174	7.028875	10.442679
std	1119.053510	82.250150	112.587105	0.201230	14.675906	25.705279	0.083407	12.361736	13.803336	3.374683	3.251350
min	68.430000	0.314865	0.434829	0.158688	1.160000	0.223522	0.000086	0.080181	9.760000	0.384906	3.880000
25%	264.135000	2.630000	6.460000	0.379881	3.080000	0.519464	0.000172	0.146916	14.160000	6.190000	8.192500
50%	453.380000	5.680000	11.390000	0.452168	3.850000	1.495000	0.003574	0.213843	23.105000	7.850000	10.965000
75%	744.820000	11.950000	50.032500	0.566460	9.965000	11.825000	0.011657	8.110000	33.862500	9.200000	12.105000
max	7407.410000	399.850000	401.490000	1.050000	86.040000	145.400000	0.337213	47.490000	101.830000	12.870000	19.150000

Convert Prices to Returns

```
dfret = df
dfret.drop(["Date"],axis=1)
dfret = dfret[append("bitcoin",cc)].pct_change()
dfret.head()
```

	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum	ripple	stratis	waves
1654	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1653	0.076969	NaN	NaN	NaN	0.006897	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1652	-0.038328	NaN	NaN	NaN	-0.018265	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1651	-0.158345	NaN	NaN	NaN	-0.116279	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1650	-0.100692	NaN	NaN	NaN	-0.113158	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
dfret.describe()
```

	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum
count	1654.000000	1362.000000	823.000000	147.000000	1654.000000	1265.000000	951.000000	424.000000	137.000000	116.000000	167.000000
mean	0.003325	0.009019	0.009152	0.002386	0.004017	0.006344	0.010932	0.017331	-0.000738	0.031244	0.010701
std	0.043022	0.109106	0.080629	0.103884	0.075260	0.081263	0.093891	0.140291	0.158851	0.157865	0.126414
min	-0.233713	-0.373474	-0.728042	-0.312441	-0.401857	-0.314917	-0.297534	-0.406989	-0.307915	-0.256024	-0.363484
25%	-0.010981	-0.026144	-0.025832	-0.059373	-0.017377	-0.032727	-0.034321	-0.041465	-0.081890	-0.048503	-0.054516
50%	0.001951	-0.001870	-0.001341	-0.008595	0.000000	-0.001451	0.000000	-0.004139	-0.015525	0.004142	-0.004329
75%	0.018551	0.029840	0.033410	0.064416	0.016338	0.036466	0.041783	0.047573	0.029536	0.076389	0.057587
max	0.429680	2.562864	0.510344	0.453795	1.290954	0.794340	0.785764	1.228137	1.231646	0.741284	0.727856

Correlations of Returns

```
round(dfret.corr(),4)
```

	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum	ripple	stratis	waves
bitcoin	1.0000	0.2660	0.2564	0.5251	0.6255	0.3386	0.3274	0.2384	0.2572	0.3840	0.4689	0.2821	0.3012	0.3934
dash	0.2660	1.0000	0.2326	0.5538	0.1904	0.2341	0.2147	0.1973	0.2787	0.3969	0.4247	0.0239	0.3321	0.2752
ethereum	0.2564	0.2326	1.0000	0.6278	0.1753	0.2205	0.1375	0.2065	0.3753	0.5336	0.3811	0.0500	0.2840	0.3012
iota	0.5251	0.5538	0.6278	1.0000	0.3939	0.4690	0.5949	0.3026	0.4212	0.5156	0.5128	0.4366	0.6189	0.5735
litecoin	0.6255	0.1904	0.1753	0.3939	1.0000	0.2331	0.2166	0.2448	0.1960	0.2835	0.3815	0.2516	0.2399	0.2692
monero	0.3386	0.2341	0.2205	0.4690	0.2331	1.0000	0.1506	0.0504	0.1894	0.2585	0.3479	0.0714	0.2745	0.2611
nem	0.3274	0.2147	0.1375	0.5949	0.2166	0.1506	1.0000	0.1718	0.3522	0.4508	0.4992	0.0979	0.2970	0.2081
neo	0.2384	0.1973	0.2065	0.3026	0.2448	0.0504	0.1718	1.0000	0.1766	0.5504	0.4164	0.0445	0.1525	0.2403
numeraire	0.2572	0.2787	0.3753	0.4212	0.1960	0.1894	0.3522	0.1766	1.0000	0.3185	0.2784	0.2581	0.3665	0.3893
omisego	0.3840	0.3969	0.5336	0.5156	0.2835	0.2585	0.4508	0.5504	0.3185	1.0000	0.4469	0.3549	0.4791	0.4482
qtum	0.4689	0.4247	0.3811	0.5128	0.3815	0.3479	0.4992	0.4164	0.2784	0.4469	1.0000	0.3420	0.5670	0.5956
ripple	0.2821	0.0239	0.0500	0.4366	0.2516	0.0714	0.0979	0.0445	0.2581	0.3549	0.3420	1.0000	0.1184	0.0767
stratis	0.3012	0.3321	0.2840	0.6189	0.2399	0.2745	0.2970	0.1525	0.3665	0.4791	0.5670	0.1184	1.0000	0.3613
waves	0.3934	0.2752	0.3012	0.5735	0.2692	0.2611	0.2081	0.2403	0.3893	0.4482	0.5956	0.0767	0.3613	1.0000

Covariances of Returns

```
cv = dfret.cov()
round(cv,4)
```

	bitcoin	dash	ethereum	iota	litecoin	monero	nem	neo	numeraire	omisego	qtum	ripple	stratis	waves
bitcoin	0.0019	0.0011	0.0007	0.0027	0.0020	0.0010	0.0010	0.0013	0.0021	0.0032	0.0030	0.0011	0.0013	0.0013
dash	0.0011	0.0119	0.0011	0.0038	0.0012	0.0014	0.0011	0.0018	0.0029	0.0041	0.0037	0.0002	0.0025	0.0015
ethereum	0.0007	0.0011	0.0065	0.0042	0.0008	0.0014	0.0011	0.0019	0.0039	0.0054	0.0033	0.0004	0.0021	0.0018
iota	0.0027	0.0038	0.0042	0.0108	0.0030	0.0039	0.0049	0.0054	0.0068	0.0081	0.0056	0.0028	0.0066	0.0051
litecoin	0.0020	0.0012	0.0008	0.0030	0.0057	0.0011	0.0012	0.0024	0.0021	0.0030	0.0036	0.0017	0.0019	0.0016
monero	0.0010	0.0014	0.0014	0.0039	0.0011	0.0066	0.0011	0.0005	0.0025	0.0035	0.0036	0.0005	0.0028	0.0020
nem	0.0010	0.0011	0.0011	0.0049	0.0012	0.0011	0.0088	0.0021	0.0045	0.0058	0.0050	0.0008	0.0028	0.0017
neo	0.0013	0.0018	0.0019	0.0054	0.0024	0.0005	0.0021	0.0197	0.0035	0.0108	0.0089	0.0008	0.0023	0.0026
numeraire	0.0021	0.0029	0.0039	0.0068	0.0021	0.0025	0.0045	0.0035	0.0252	0.0052	0.0046	0.0026	0.0061	0.0054
omisego	0.0032	0.0041	0.0054	0.0081	0.0030	0.0035	0.0058	0.0108	0.0052	0.0249	0.0072	0.0036	0.0071	0.0059
qtum	0.0030	0.0037	0.0033	0.0056	0.0036	0.0036	0.0050	0.0089	0.0046	0.0072	0.0160	0.0031	0.0086	0.0069
ripple	0.0011	0.0002	0.0004	0.0028	0.0017	0.0005	0.0008	0.0008	0.0026	0.0036	0.0031	0.0077	0.0016	0.0007
stratis	0.0013	0.0025	0.0021	0.0066	0.0019	0.0028	0.0028	0.0023	0.0061	0.0071	0.0086	0.0016	0.0128	0.0031
waves	0.0013	0.0015	0.0018	0.0051	0.0016	0.0020	0.0017	0.0026	0.0054	0.0059	0.0069	0.0007	0.0031	0.0078

An Important Check!!

Check Cov Matrix is Positive Definite, else fix it

```
#Check if positive definite
cv = cv.as_matrix()
all(linalg.eigvals(cv) > 0)
```

```
False
```

```
#Fix the CV matrix
from statsmodels.stats.correlation_tools import cov_nearest
cvpd = cov_nearest(cv)
print(all(linalg.eigvals(cvpd) > 0))
```

```
True
```

```
cv = cvpd
pd.DataFrame(cvpd)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0.001851	0.001063	0.000757	0.002595	0.002008	0.000969	0.001071	0.001372	0.002021	0.003118	0.002779	0.001076	0.001343	0.001352
1	0.001063	0.011904	0.001112	0.003626	0.001216	0.001419	0.001179	0.001887	0.002859	0.003984	0.003507	0.000229	0.002486	0.001536
2	0.000757	0.001112	0.006501	0.004024	0.000788	0.001413	0.001121	0.002012	0.003863	0.005220	0.003062	0.000400	0.002138	0.001822
3	0.002595	0.003626	0.004024	0.010792	0.002998	0.003782	0.004602	0.005038	0.006901	0.008210	0.005878	0.002668	0.006220	0.004812
4	0.002008	0.001216	0.000788	0.002998	0.005664	0.001091	0.001207	0.002407	0.002149	0.003031	0.003568	0.001680	0.001843	0.001539
5	0.000969	0.001419	0.001413	0.003782	0.001091	0.006604	0.001093	0.000583	0.002428	0.003355	0.003405	0.000489	0.002790	0.001982

Means and Standard Deviations

```
#Calculate the mean returns
mn = mean(dfret)
print(mn)
mn = mn.as_matrix()
```

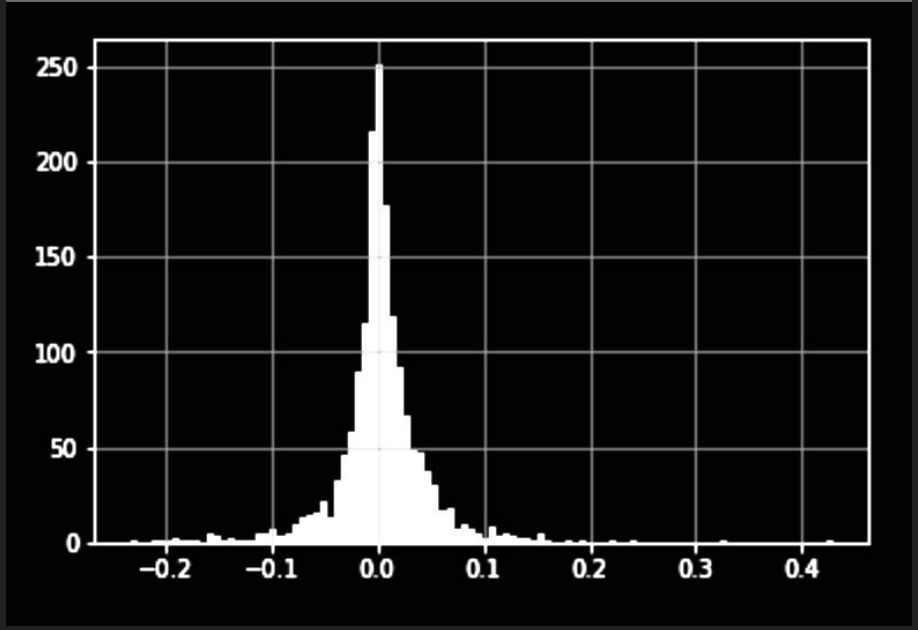
```
bitcoin      0.003325
dash         0.009019
ethereum     0.009152
iota         0.002386
litecoin     0.004017
monero       0.006344
nem          0.010932
neo          0.017331
numeraire   -0.000738
omisego      0.031244
qtum         0.010701
ripple       0.005424
stratis      0.018158
waves        0.006178
dtype: float64
```

```
#Calculate the standard deviations
stdev = std(dfret)
print(stdev)
stdev = stdev.as_matrix()
```

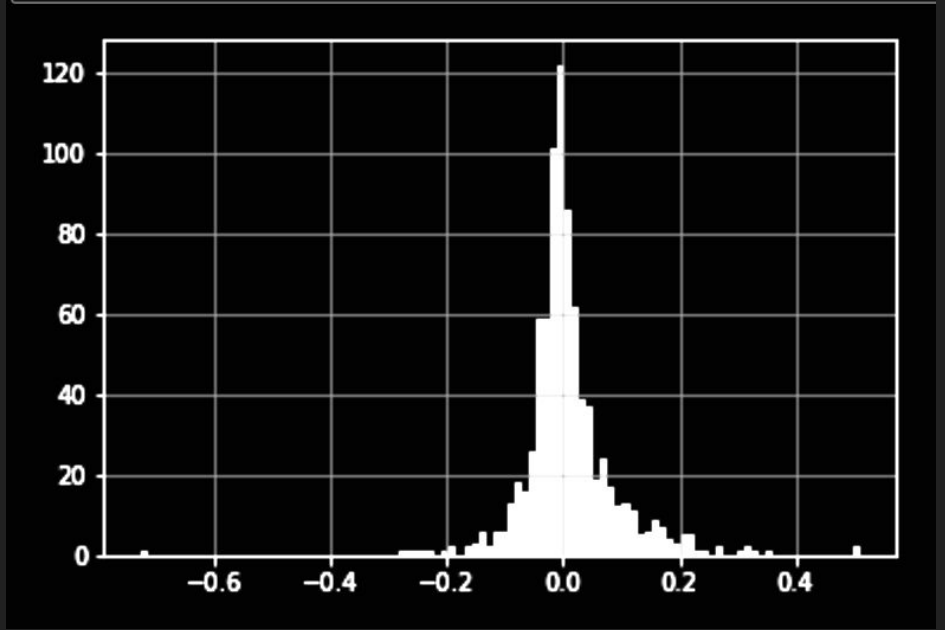
```
bitcoin      0.043009
dash         0.109066
ethereum     0.080580
iota         0.103530
litecoin     0.075237
monero       0.081231
nem          0.093842
neo          0.140126
numeraire    0.158270
omisego      0.157183
qtum         0.126035
ripple       0.087905
stratis      0.112861
waves        0.088002
dtype: float64
```

Return Distributions

```
hist(dfret["bitcoin"].dropna(),100)
grid()
```




```
hist(dfret["ethereum"].dropna(),100)
grid()
```



Optimal Portfolios

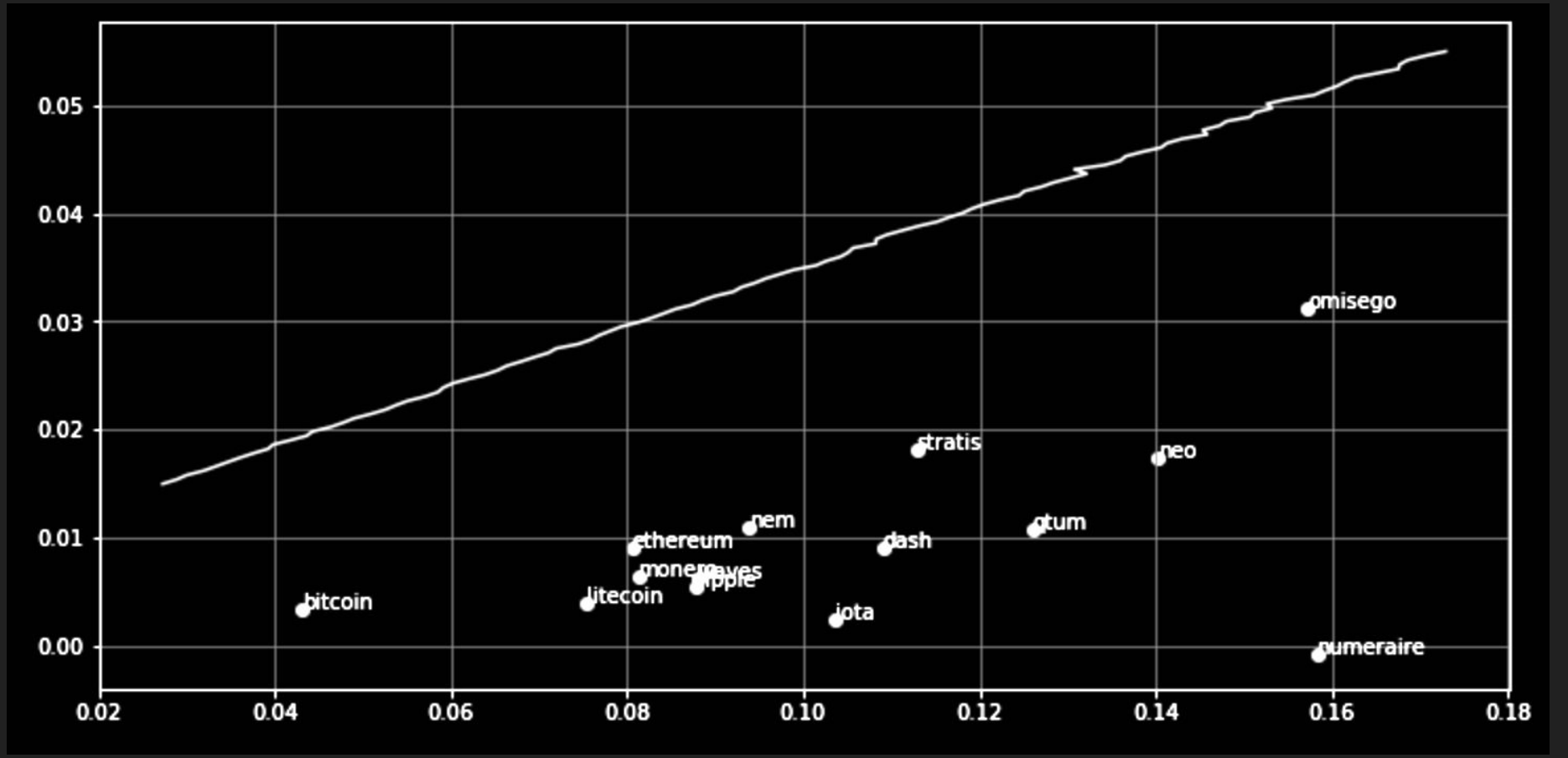
We use Markowitz Mean-Variance optimization for this.

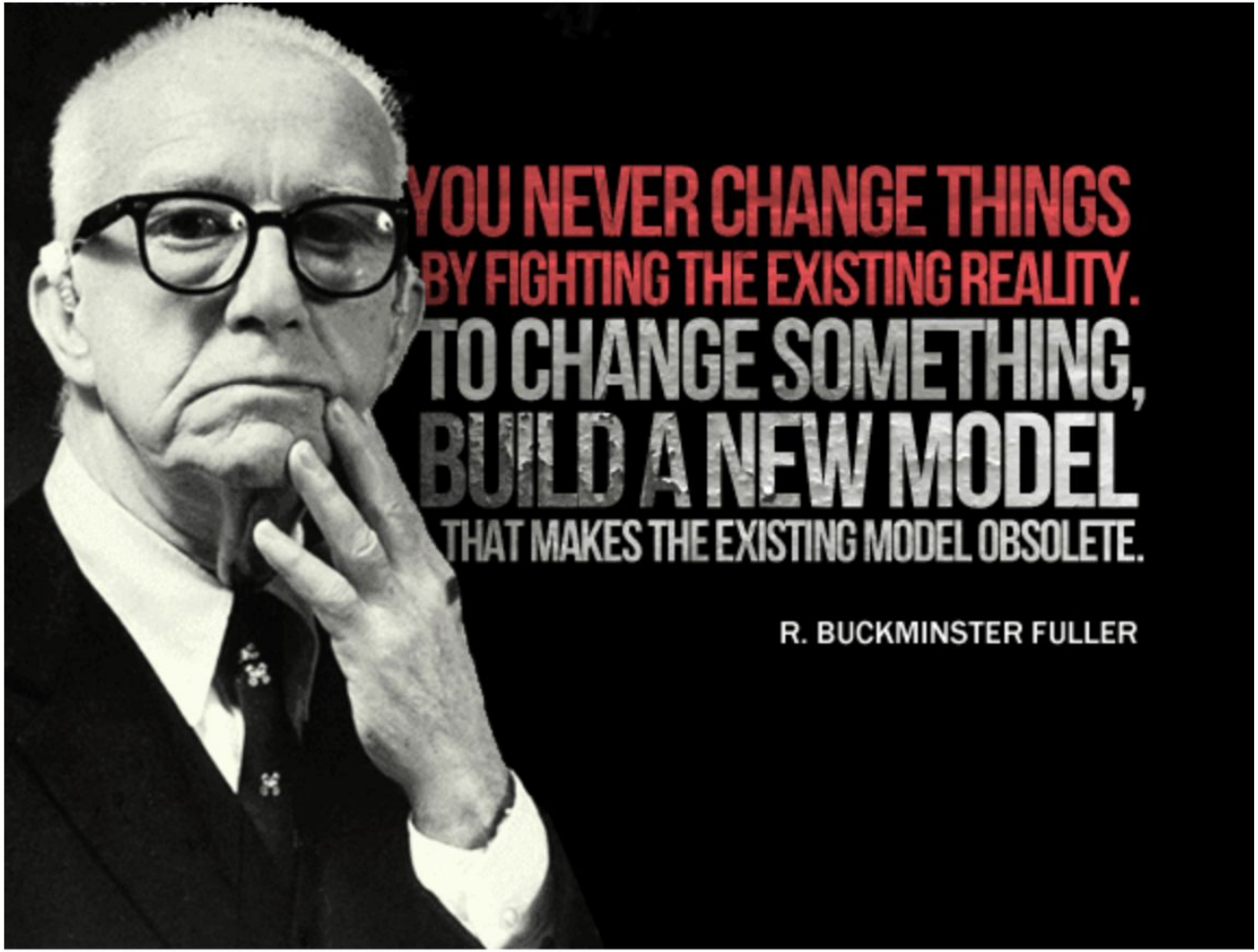
```
#Optimal portfolio solution for a given expected return E(r)
def markowitz(mu,cv,Er):
    n = len(mu)
    wuns = ones(n)
    cvinv = linalg.inv(cv)
    A = wuns.T.dot(cvinv).dot(mu)
    B = mu.T.dot(cvinv).dot(mu)
    C = wuns.T.dot(cvinv).dot(wuns)
    D = B*C - A*A
    lam = (C*Er-A)/D
    gam = (B-A*Er)/D
    wts = lam*(cvinv.dot(mu)) + gam*(cvinv.dot(wuns))
    return wts
```

Generate the efficient frontier

```
#TRACING OUT THE EFFICIENT FRONTIER
Er_vec = linspace(0.015,0.055,100)
Sig_vec = zeros(len(Er_vec))
j=0
for Er in Er_vec:
    wts = markowitz(mn,cv,Er)
    Sig_vec[j] = sqrt(wts.T.dot(cv).dot(wts))
    j=j+1

figure(figsize=(12,6))
plot(Sig_vec,Er_vec)
grid()

cn = append("bitcoin",cc)
scatter(stdev, mn)

for i, txt in enumerate(cn):
    annotate(txt, (stdev[i],mn[i]))
```

Portfolio way better than each asset




YOU NEVER CHANGE THINGS
BY FIGHTING THE EXISTING REALITY.
TO CHANGE SOMETHING,
BUILD A NEW MODEL
THAT MAKES THE EXISTING MODEL OBSOLETE.
R. BUCKMINSTER FULLER

Readings

- Python Tutorial for Absolute Beginners.pdf
- An Overview of Cryptocurrencies for the Savvy Investor.pdf
- Sarin_Value of bitcoin.pdf
- https://github.com/ethereum/yellowpaper
- WEF_The_future_of_financial_infrastructure.pdf
- ChopraPrabhalaYantri_SSRN-id2919091_BankAccountsForUnbanked.pdf
- Convergence of AI and Blockchains: https://hackernoon.com/the-convergence-of-ai-and-blockchain-whats-the-deal-60c618e3accc
- Build your own blockchain in R: http://bigdata-doctor.com/building-your-own-blockchain-in-r/