

Block Chain

Dr. Anil Kumar K.M

Associate Professor,

Dept of CS&E

Sri Jayachamarajendra College of Engineering
JSS Science And Technology University, Mysore

Disclaimer:

- This Instructional material is used as teaching aid for the instructor. It is an abstract of contents from the prescribed syllabus.
- The learner's are strongly recommended to refer to the prescribed Text Books, reference Books and other Literatures for enhancing their learning.

Acknowledgements

- The instructor acknowledges the use of Contents/Figures/ Statistics/ Concepts from the works of the respected authors (both in print and online) for only teaching purpose.
- This material is prepared by referencing contents from the following resources:
 1. Blockchain Technology Overview, NIST publications, 2018.
 2. Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming.
 3. Beginning Block Chain- Bikarmaditya Singhal et al
 4. IBM blockchain for dummies- Manav Gupta
 5. Web/ Blogs (Professional Block Chain sites)

What is Block Chain?

| | |
|------------------------|--|
| A Database | A list of records / transactions, like a ledger, that keeps growing as more entries are added; |
| Which is Distributed | Copies of the entire database are stored on multiple computers on a network, syncing within minutes / seconds; |
| adjustably Transparent | Records stored in the database may be made visible to relevant stakeholders without risk of alteration; |
| highly Secure | Malicious actors (hackers) can no longer just attack one computer and change any records; |
| and Immutable | The mathematical algorithms make it impossible to change / delete any data once recorded and accepted. |

- Value

- Trust

- Truth

- Secure

What is Block Chain?

Blockchain is a secure series or chain of timestamped records stored in a database that a group of users manages who are a part of a decentralized network.

Blockchain is a decentralized or distributed ledger where each node in the network has access to the data or records stored in a blockchain.

The encryption of all the important data records in the blockchain is done using cryptographic techniques. This ensures the security of the data in the blockchain. [Web]

Characteristics of Block chain

- Ledger – Append only
- Secure – Cryptographically secure
- Shared- Multiple participants
- Distributed – Scaling of nodes

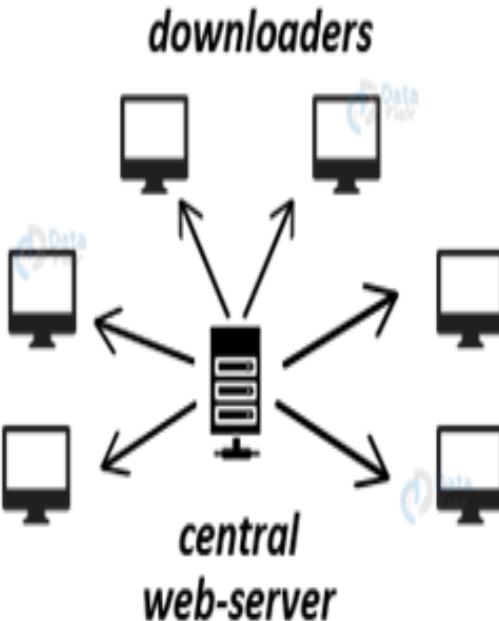
Features of Block Chain

- Peer-to-peer Network
- Decentralized
- Incorruptible and Immutable

Block chain Basics – Key Elements

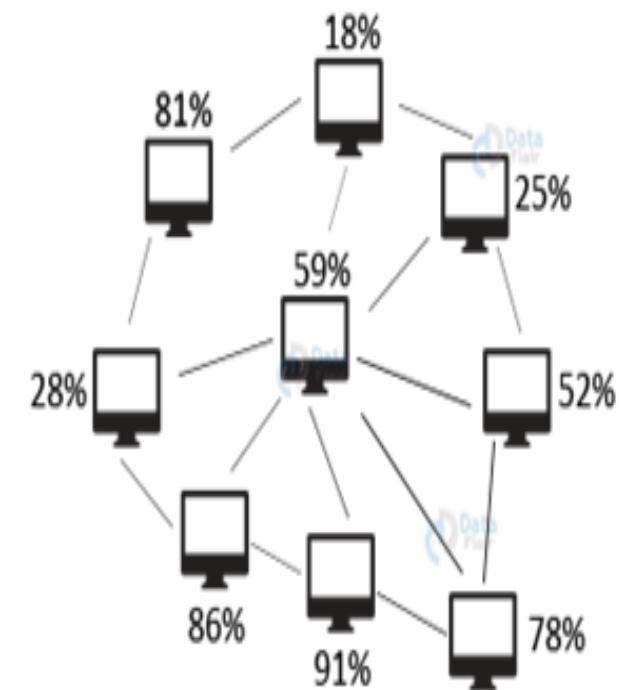
- Peer-to-peer Network
- Distributed Ledger System
- Key Cryptography
- Hashing
- Proof-of-Work
- Merkle Tree and Merkle Root

Traditional Centralized Downloading



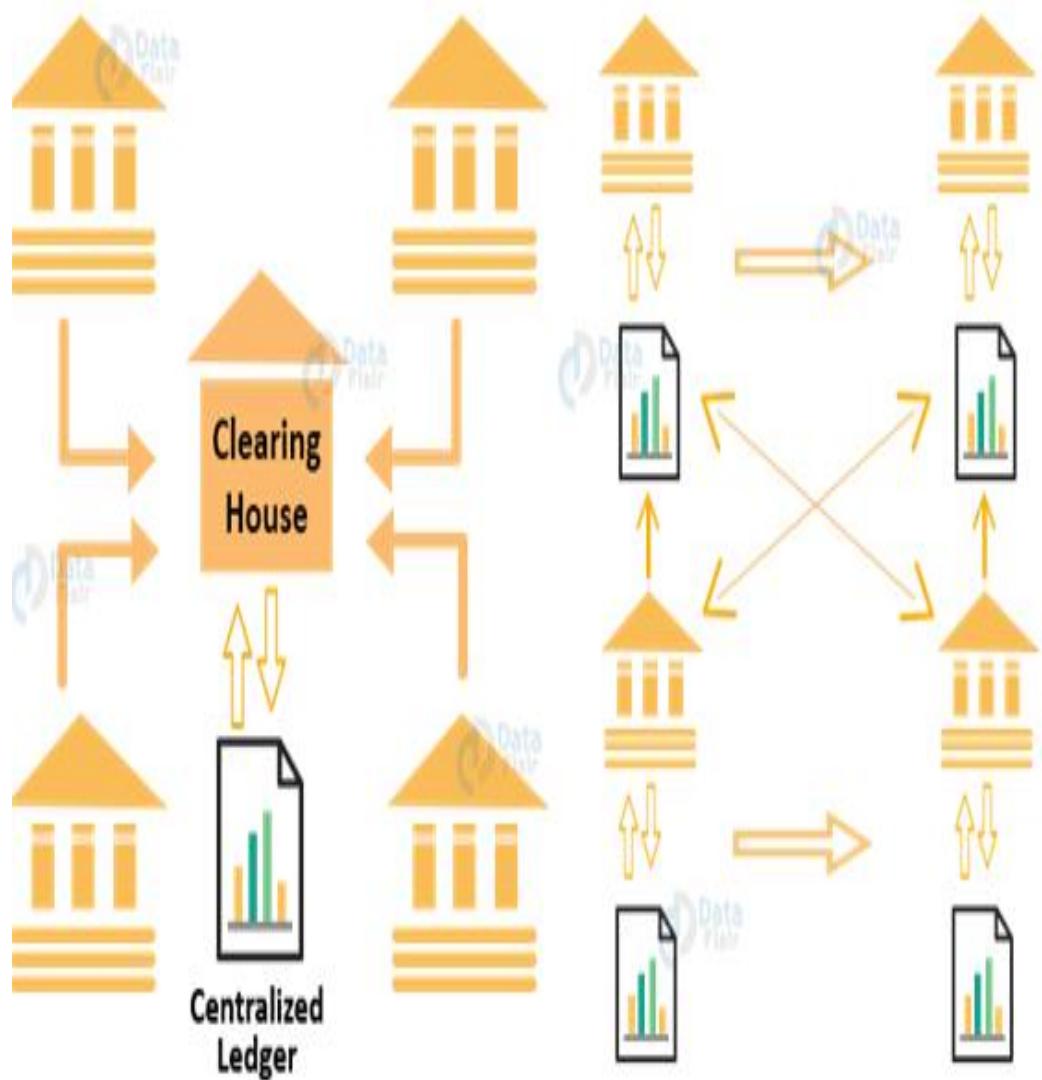
- Slow
- Single Point of Failure
- High Bandwidth Usage for Server

Decentralized Peer-to-Peer Downloading

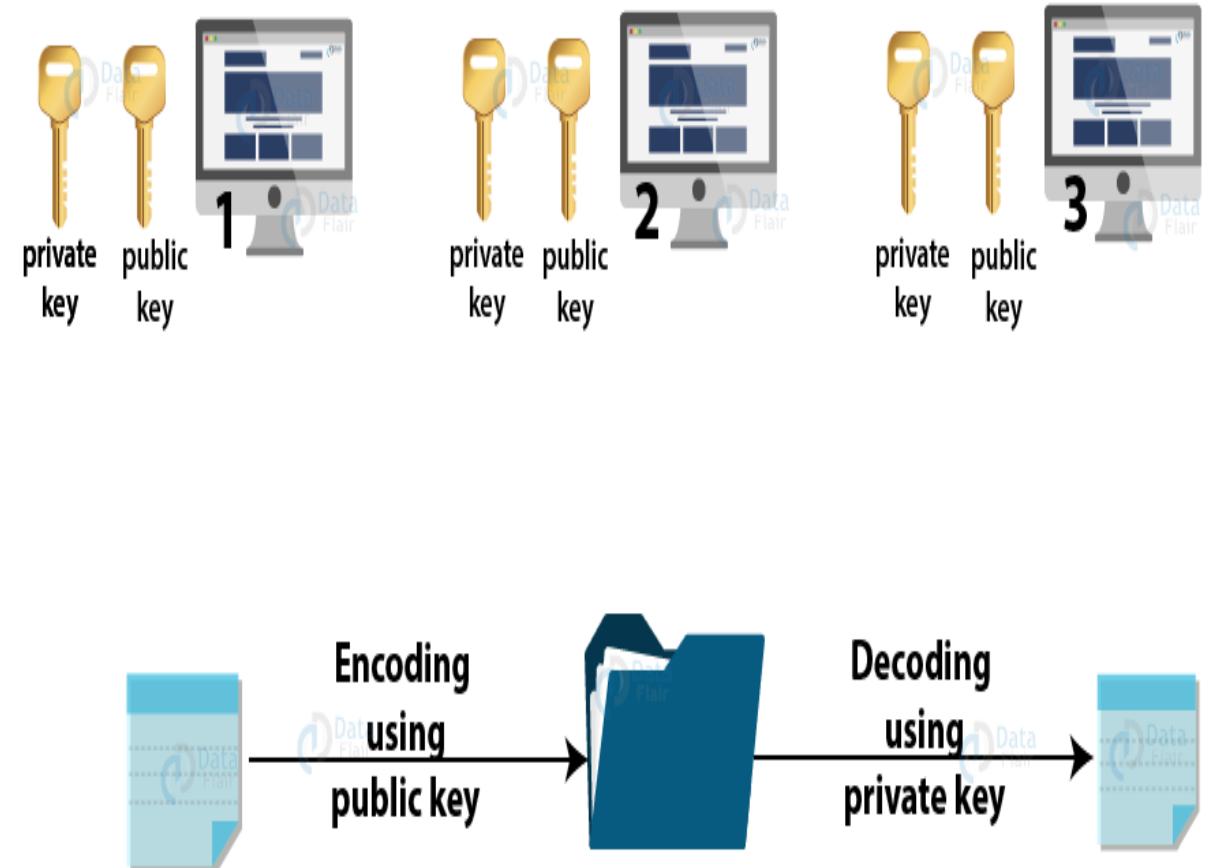


- Fast
- No Single Point of Failure
- All Downloaders are Uploaders also

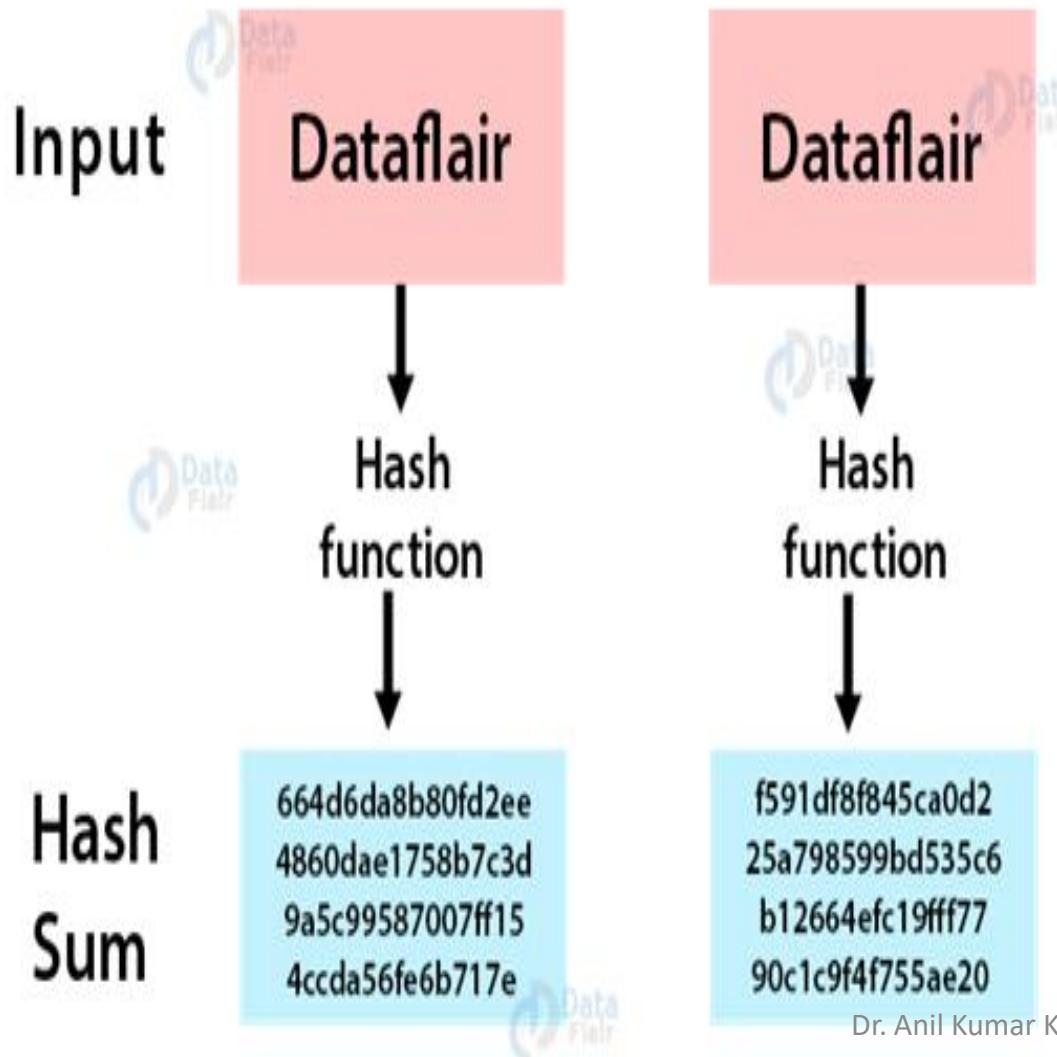
Distributed Ledger System in Blockchain



Cryptography



Hashing Process in Blockchain



Few Facts

- Recently, Walmart was able to reduce a food tracing process from 6 days to about just 2 seconds by leveraging blockchain
- Blockchain is expected to disrupt the Banking and financial industry soon. Experts claim banks could save **\$8-12 billion annually** by leveraging blockchain technology

Who/What is
Gartner?



By 2023, blockchain will support the global movement and tracking of **\$2 trillion of goods and services** annually.

Gartner

BLOCKCHAIN HISTORY

1990s

The concept of distributed computing has been around since 1990

2011 – 2012

The deployment of cryptocurrency in application related to cash

2013 – 2014

Financial markets and applications using blockchain beyond cash transactions

2015 – 2016

Permissioned blockchain network solutions

Origin

Transactions

Contracts

Application

2009 Satoshi Nakamoto created bitcoin and introduced the concept of a blockchain to incarnate a decentralized ledger maintained by anonymous consensus

Currency transfer and digital payments systems

Smart contracts

Market consolidation and further sub-developments

2009

2012 – 2013

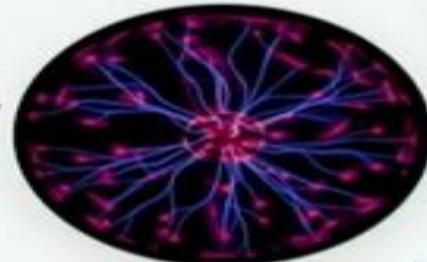
2014 – 2015

2016 – 2017

Blockchain

THE TECHNOLOGICAL REVOLUTION

Social Web



Internet of Things



The Cloud

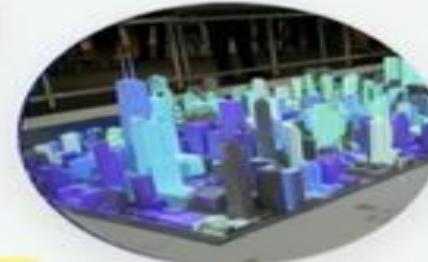


Mobility



Rectangular Slip

Big Data



Machine Learning

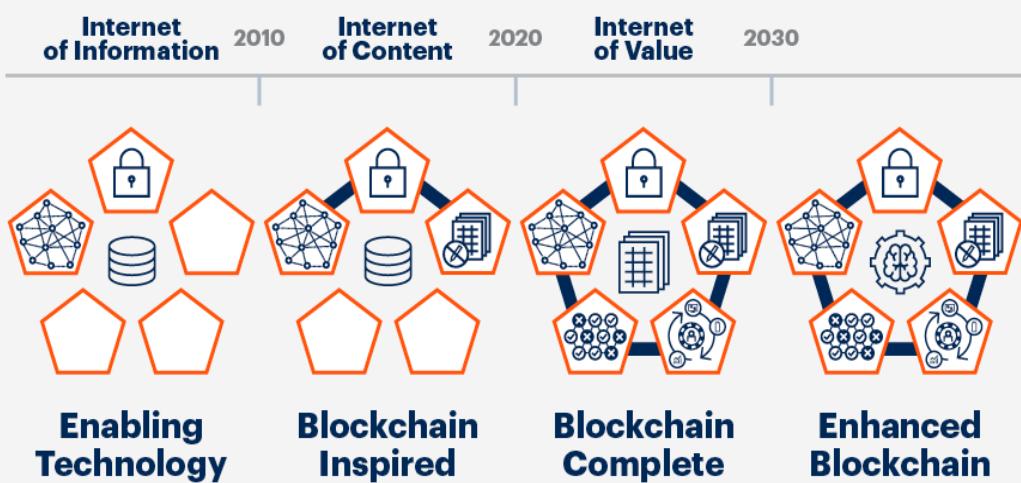


Drones & Robotics



Is it a Hype ???

The Gartner Blockchain Spectrum, which began with emergence in 2008, predicts **maturity around 2025:**



Source: Gartner
© 2019 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner®



By 2023, blockchain will support the global movement and tracking of **\$2 trillion of goods and services** annually.

Source: Gartner
© 2019 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner®

Why Blockchain???

TOP EMERGING JOBS

- Blockchain developer
- AI specialist
- JavaScript developer
- Robotic process automation consultant
- Back-end developer
- Growth manager
- Site reliability engineer
- Customer success specialist
- Full-Stack engineer
- Robotics engineer (software)
- Cybersecurity specialist
- Python developer
- Digital marketing specialist
- Front-end engineer
- Lead generation specialist



Wikipedia

"Open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way" wikipedia.org

- ✓ Open
- ✓ Distributed
- ✓ Ledger
- ✓ P2P
- ✓ Permanent



Source: wikipedia.org

A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography.

– Wikipedia



Prelude for Block chain

- Cryptography exists from 1900 onwards
- Chaum published the idea of anonymous electronic money in a 1983 paper
- Chaum started the company [DigiCash](#) in 1989 with "ecash" as its trademark. (It went bankruptcy in 1998)
- In 1998 Nick Szabo's paper named [GOD protocol](#)

•The core ideas behind blockchain technology emerged in the late 1980s and early 1990s. In 1989, Leslie Lamport developed the Paxos protocol, and in 1990 submitted the paper *The Part-Time Parliament* to *ACM Transactions on Computer Systems*; the paper was finally published in a 1998 issue. The paper describes a **consensus model** for reaching agreement on a result in a network of computers where the computers or network itself may be unreliable.



Digital Time Stamping

Haber, Stuart; Stornetta, W. Scott (January 1991). "How to time-stamp a digital document". *Journal of Cryptology*. 3 (2): 99–111



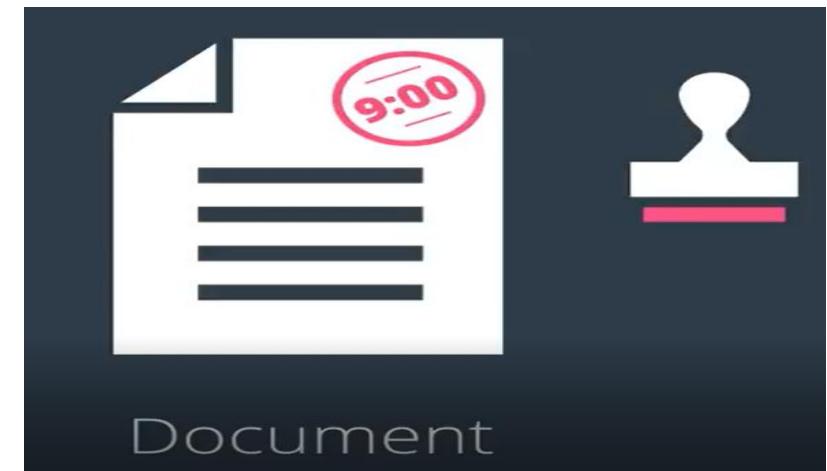
Stuart Haber



W. Scott Stornetta



The technology was originally described in 1991 and was intended to timestamp digital documents to avoid backdate or tempering of any records



BIRTH OF BITCOIN

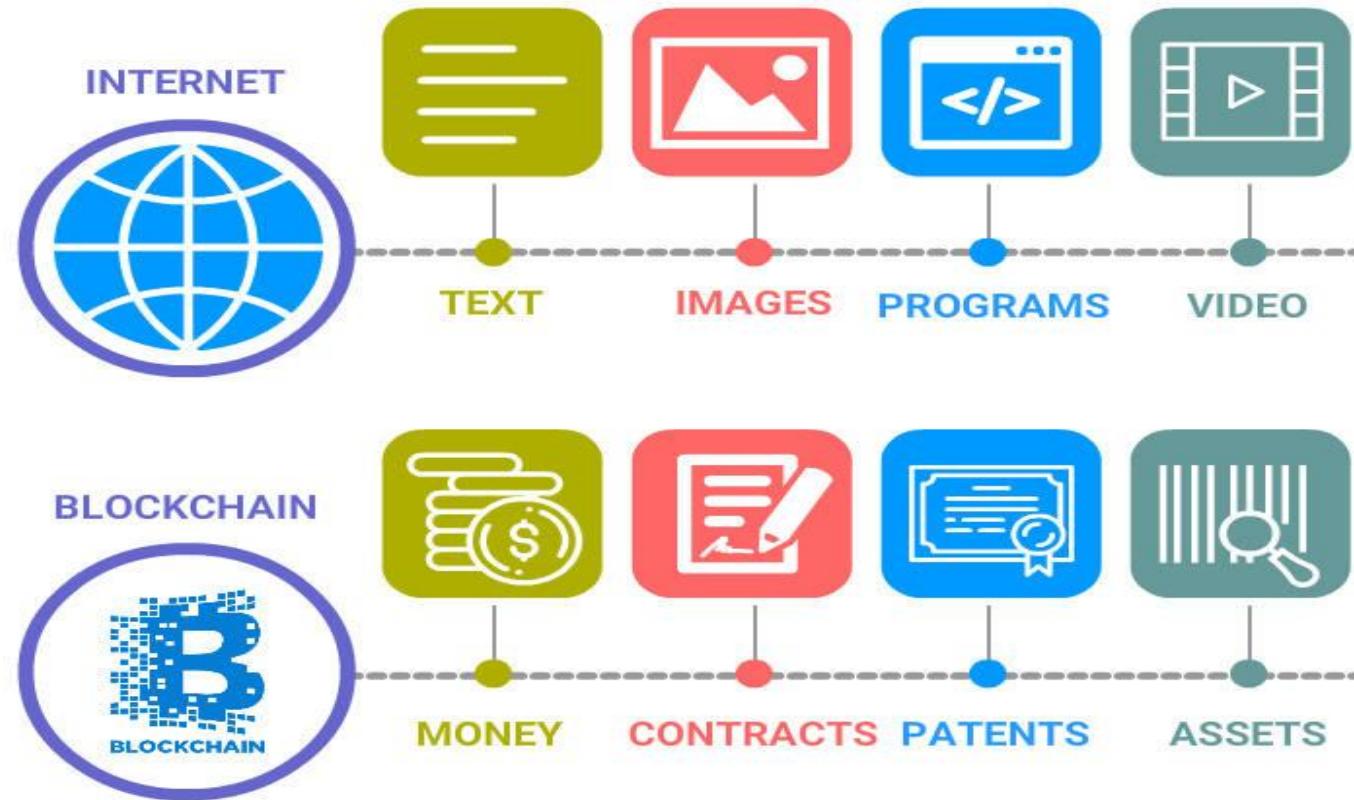
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest



Internet & Blockchain

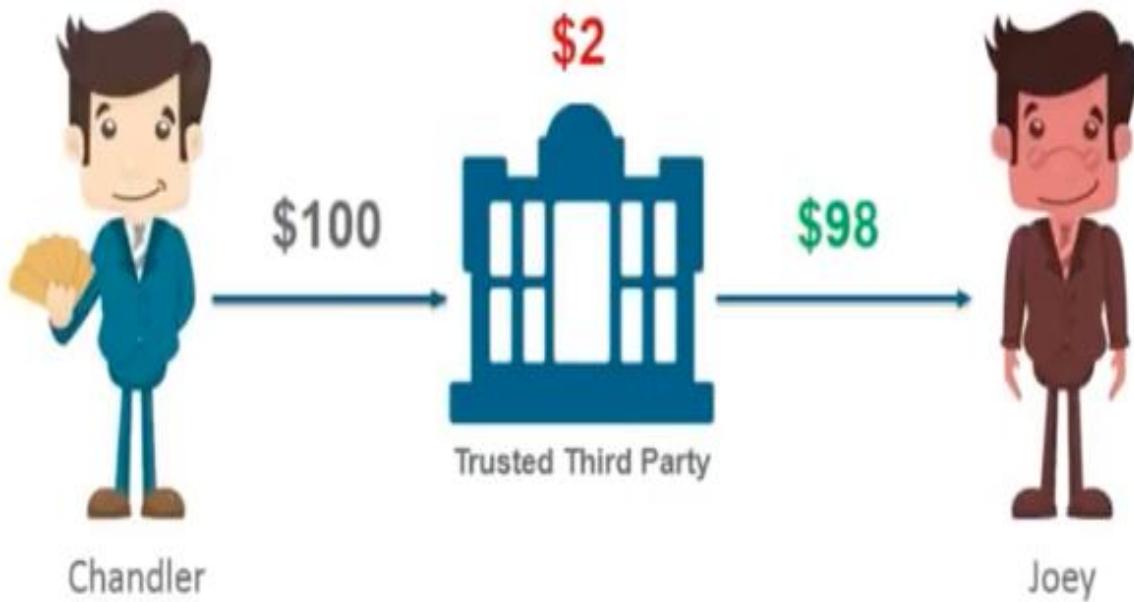


“What the internet did for communications, blockchain will do for trusted transactions.” - Ginni Rometty,
CEO of IBM

Why Block chain ?? (Bank)

Centralization

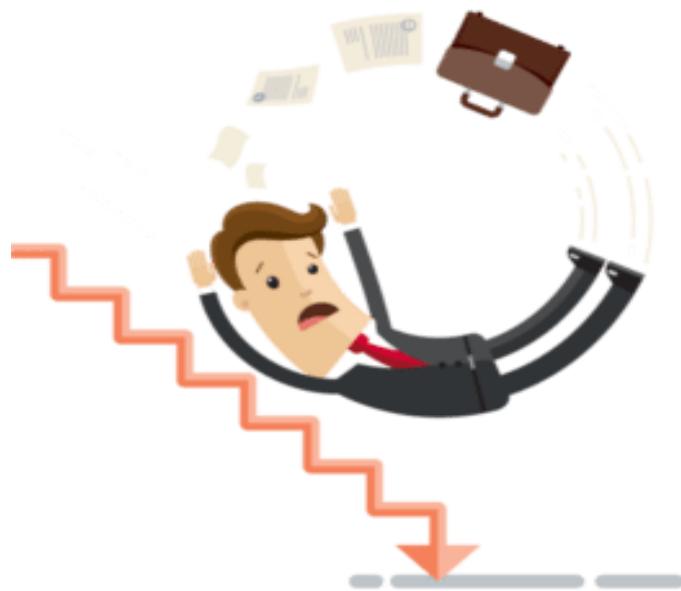
High Transaction Fees



JPMorgan Chase, Bank of America and Wells Fargo

alone earned more than \$6 billion from ATM and overdraft fees in 2015 (SNL Financial and CNNMoney Report)

Why Blockchain ??(Bank scenario)



In 2007-08 financial crisis up to the tune of \$11 Trillion (\$11,000,000,000,000) worldwide.

Why Blockchain ??(Bank scenario)



Reserve Bank of India

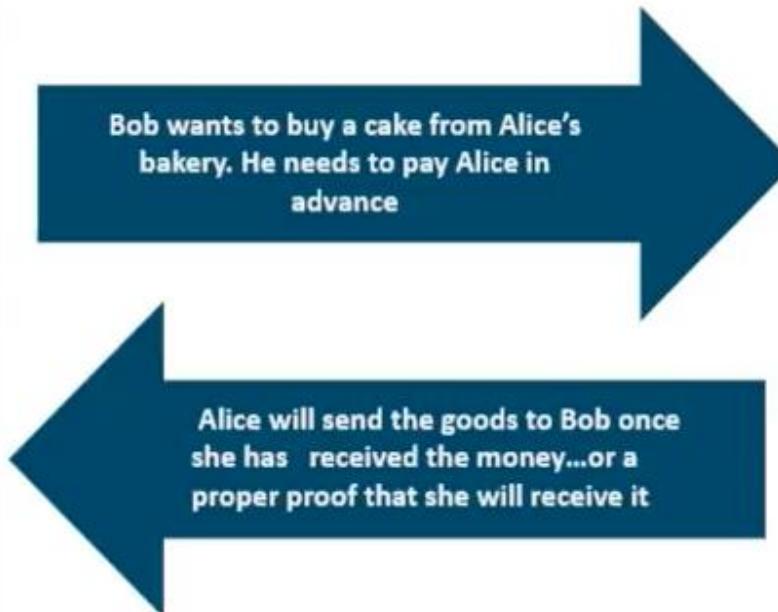


US Federal Reserve

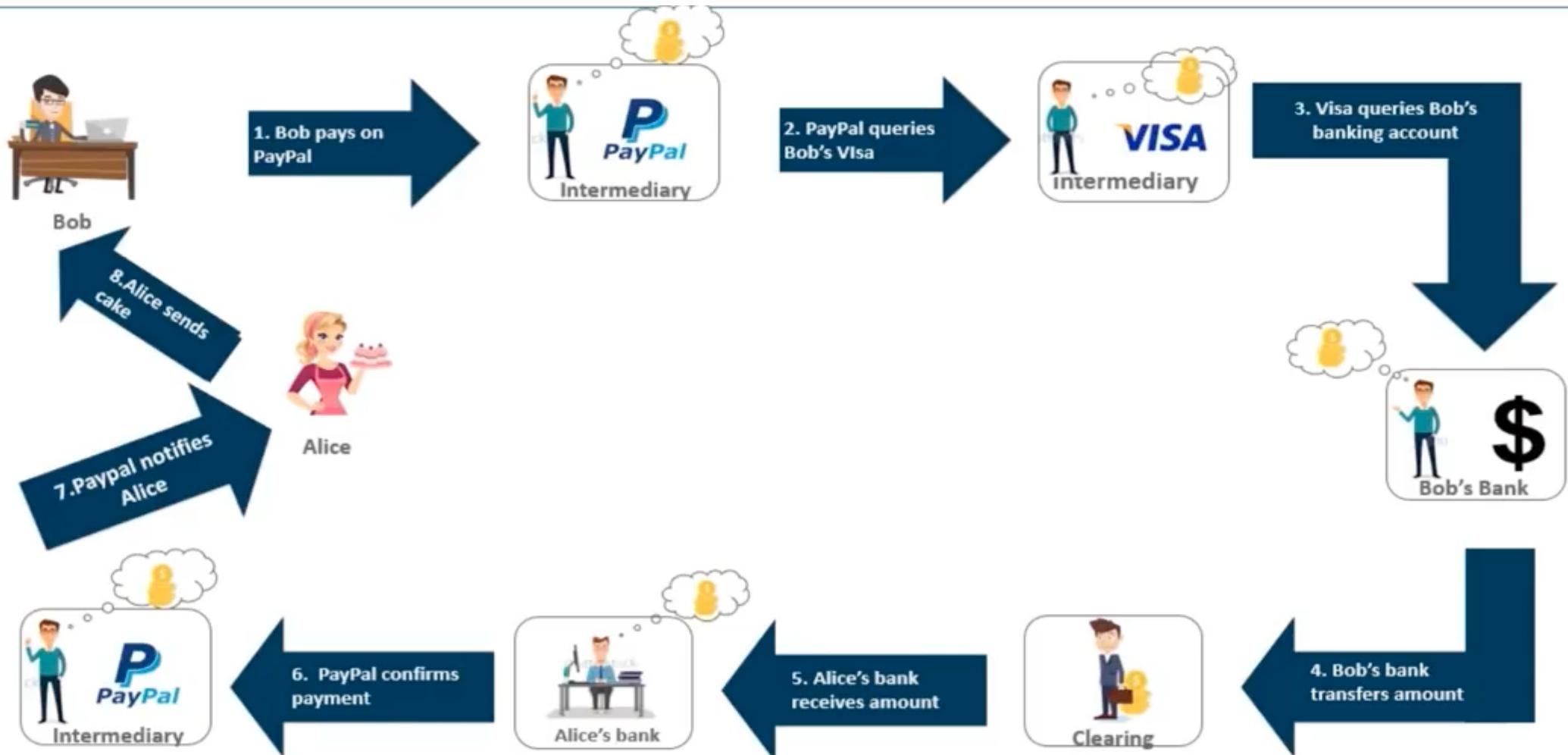
Banks have become synonymous with crises and crashes due to depression and fractional reserve banking



Traditional Transaction



Traditional Transaction flow

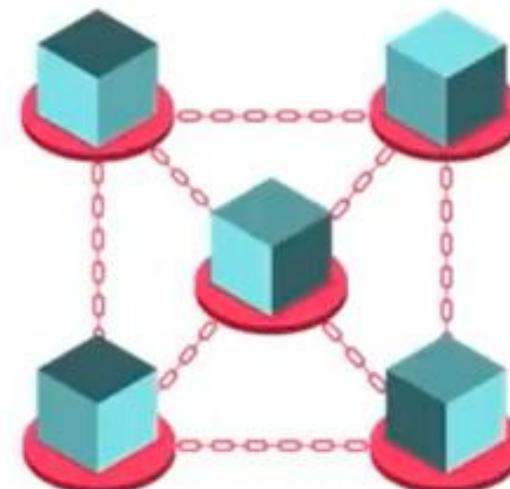


How Blockchain works??

Decentralization



Bob



Alice

Bob want's to send money to Alice over a blockchain network!

Blockchain Transaction flow!!!!

Bob wants to send
money to Alice



The transaction is
represented online as a block

Every block is broadcast to
every party on the network



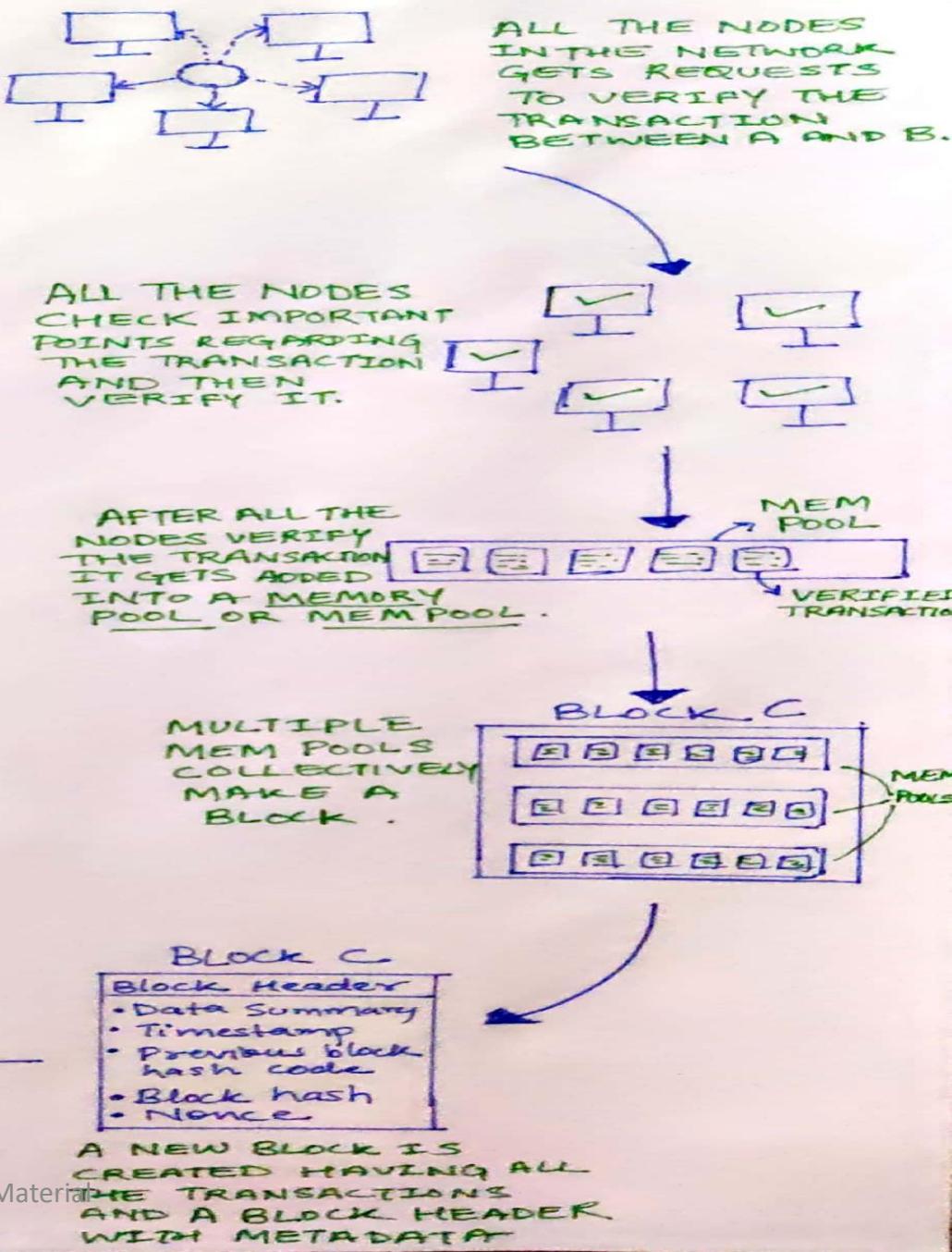
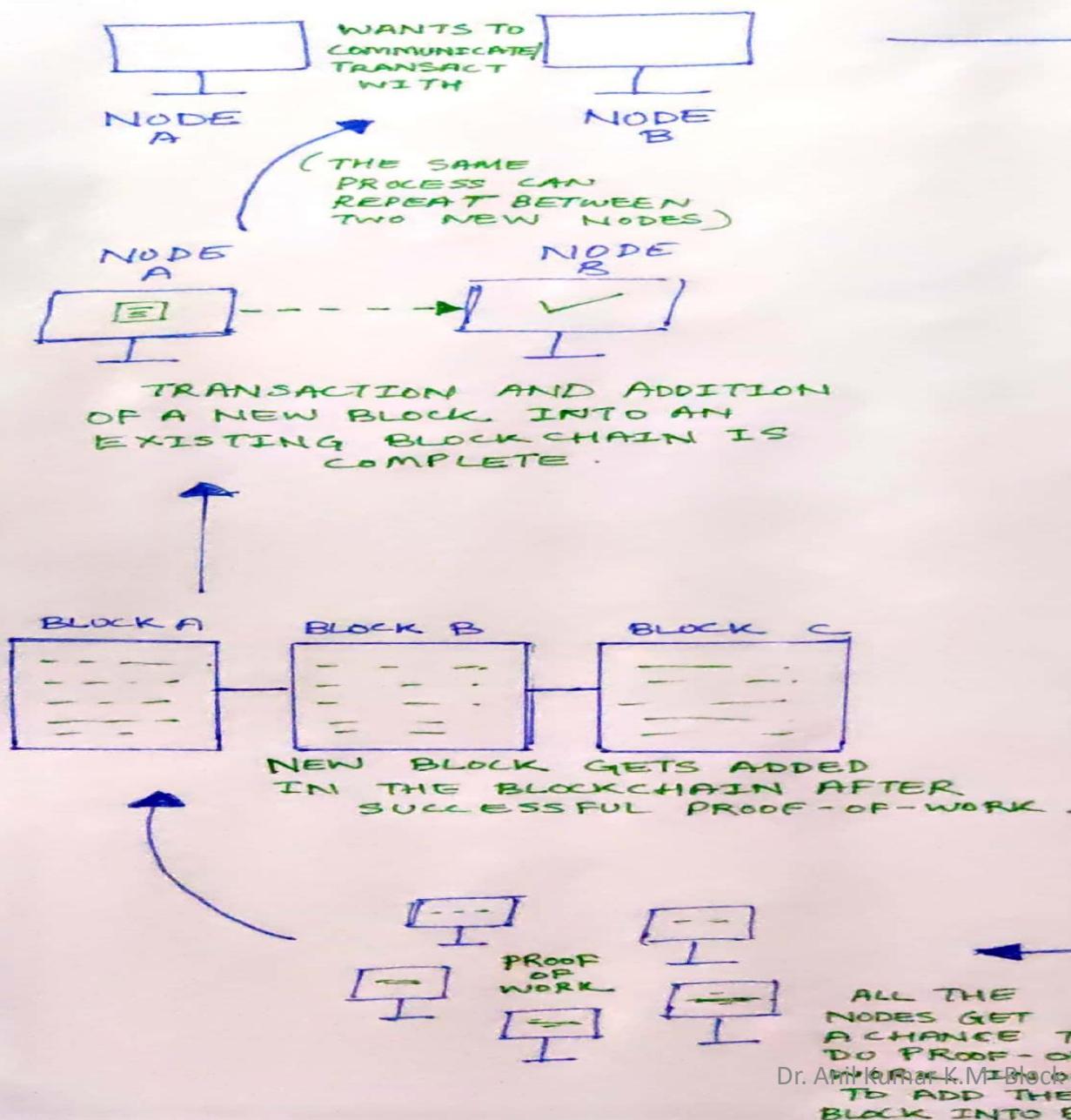
Those in the network approve
the validity of the transaction

After validation the block is
added to the chain



Alice receives her
money from Bob

HOW BLOCKCHAIN WORKS?

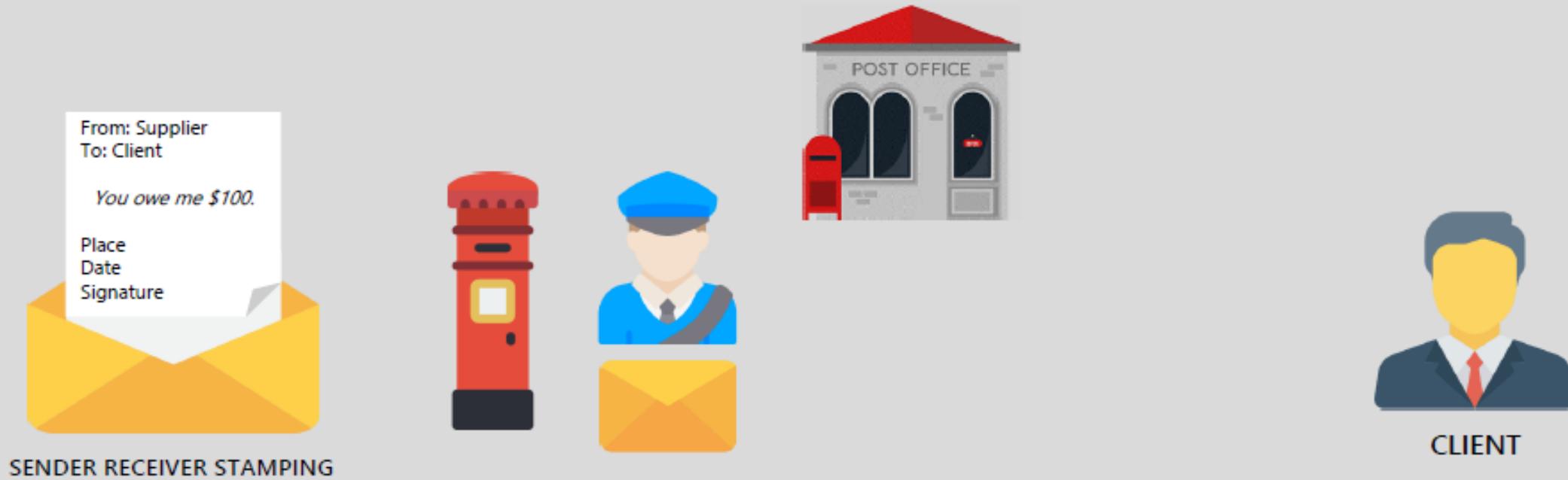


ABC of Blockchain: What can you do with it?

Let's use an example:

I am a supplier that delivered goods to a client.

To get paid I send a letter to the client asking for the due (es. \$100).



ABC of Blockchain: What can you do with it?

Question: How do I know if (and when) the letter has been delivered?

Answer: Ask the client!

Question: What if I don't trust what my client says?

Answer: Ask the post office!

Question: I don't want an intermediary. No other options are possible?

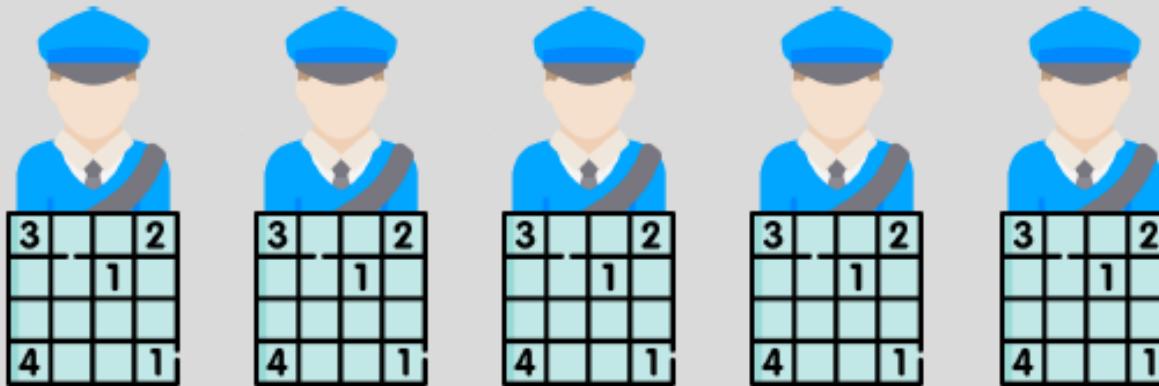
Answer: You can ask directly to the mail carrier!

Example

Which mail carrier should I ask?

How to get the information WITHOUT asking the post office?

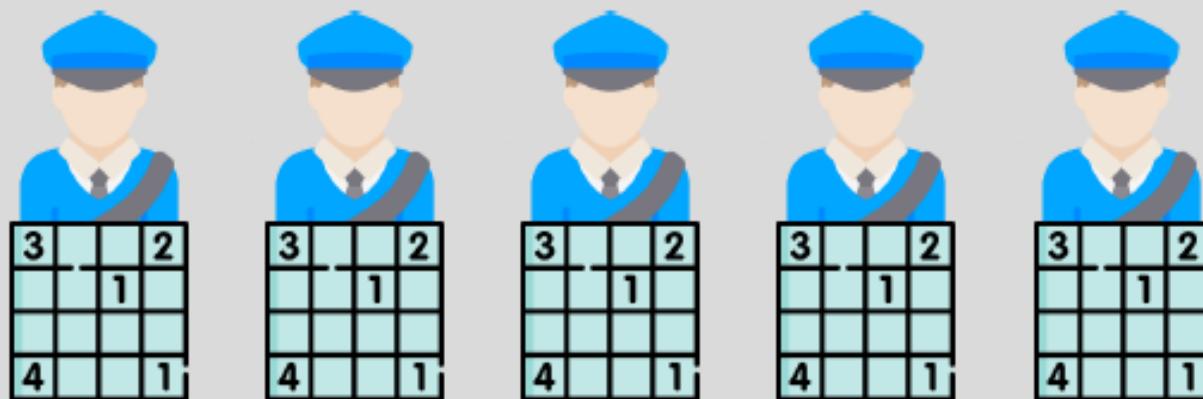
Solution: The mail carriers compete to decide who will make the delivery.



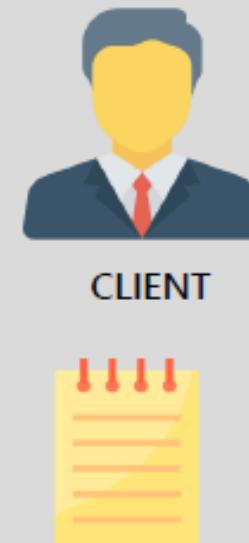
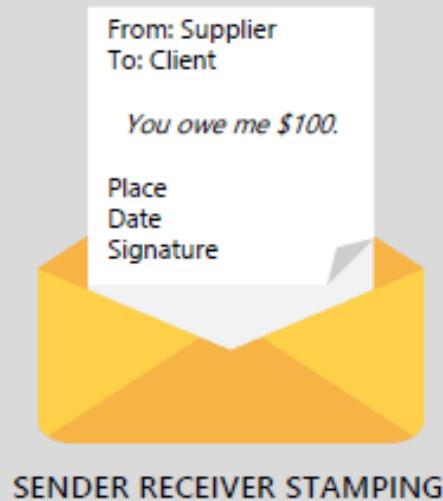
Eventually someone will resolve the Sudoku



The winner makes the delivery.



The winner makes the delivery

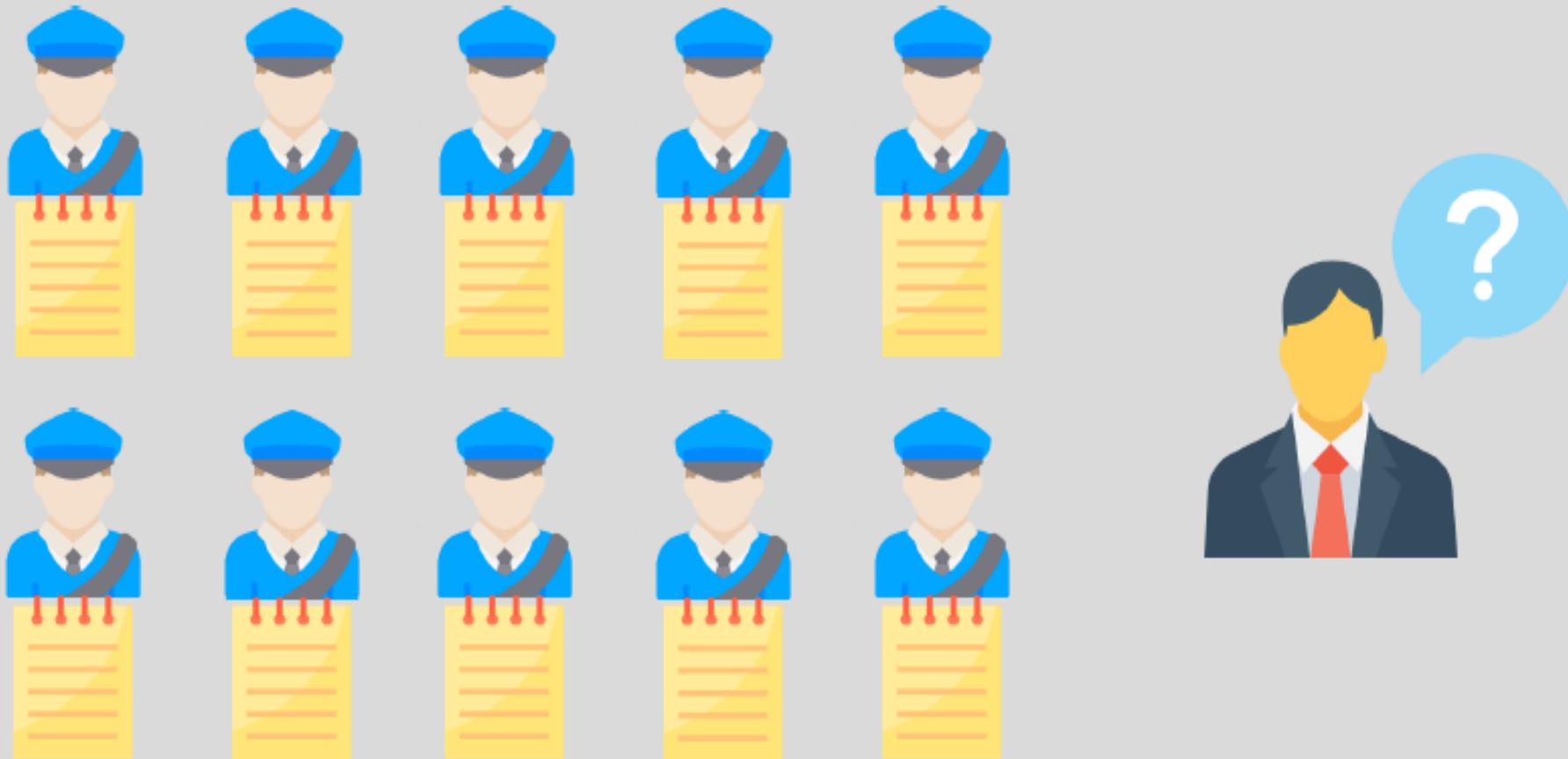


And writes the data of the delivery on his registry.

Copy of the registry is given to all the other mail carriers



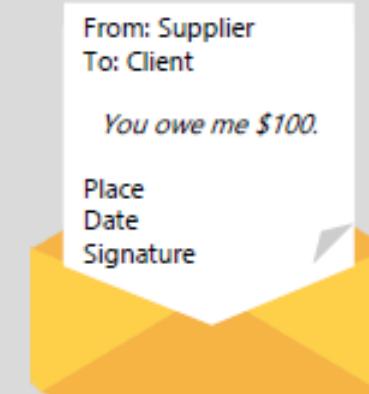
Question: Which mail carrier can I ask?



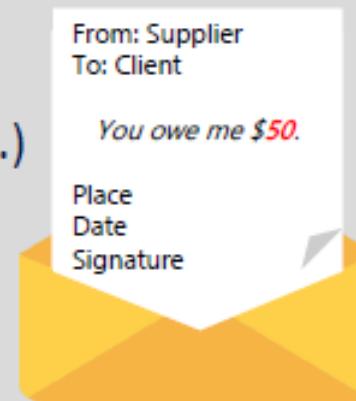
Answer: Ask **any** mail carrier!

So is everything set?

Not really. What happens if against my original letter...



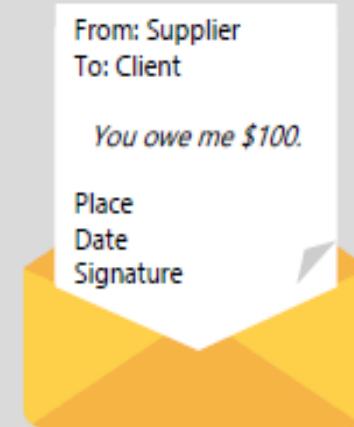
... the client presents a different letter?
(warning: the client has maliciously altered the information...)



As of today, I must ask (again) an intermediary to fix the issue.

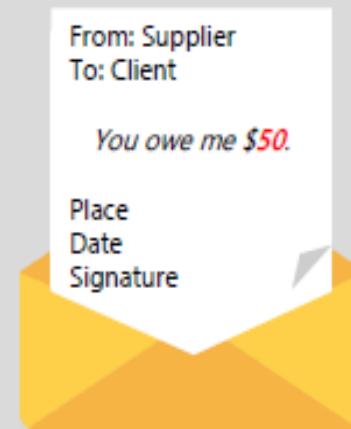
The solution

The original letter is assigned a «special» code



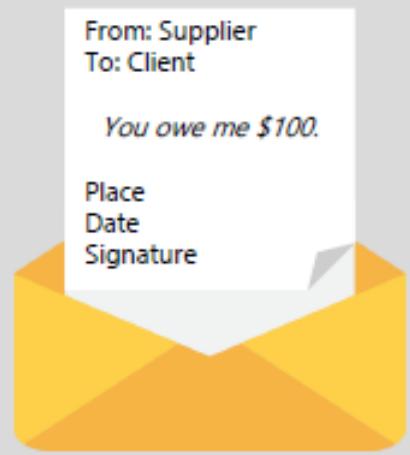
Cf23df207d99a74fbe169K3e5a035e633b65d94

The malicious letter is assigned a different
«special» code



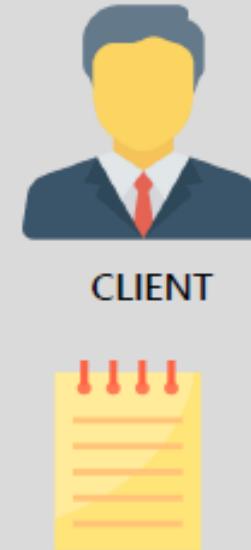
Ar03vt\$97de9a7sfbr257f3r8b099e824wf5a18

The winner mail carrier delivers the letter with the Cf23df207d99a74fbe169K3e5a035e633b65d94 code



SENDER RECEIVER STAMPING

Cf23df207d99a74fbe169K3e5a035e633b65d94



Cf23df207d99a74fbe169K3e5a035e633b65d94

All mail carriers have now a copy of the letter with the code
Cf23df207d99a74fbe169K3e5a035e633b65d94



They match

Cf23df207d99a74fbe169K3e5a035e633b65d94

with

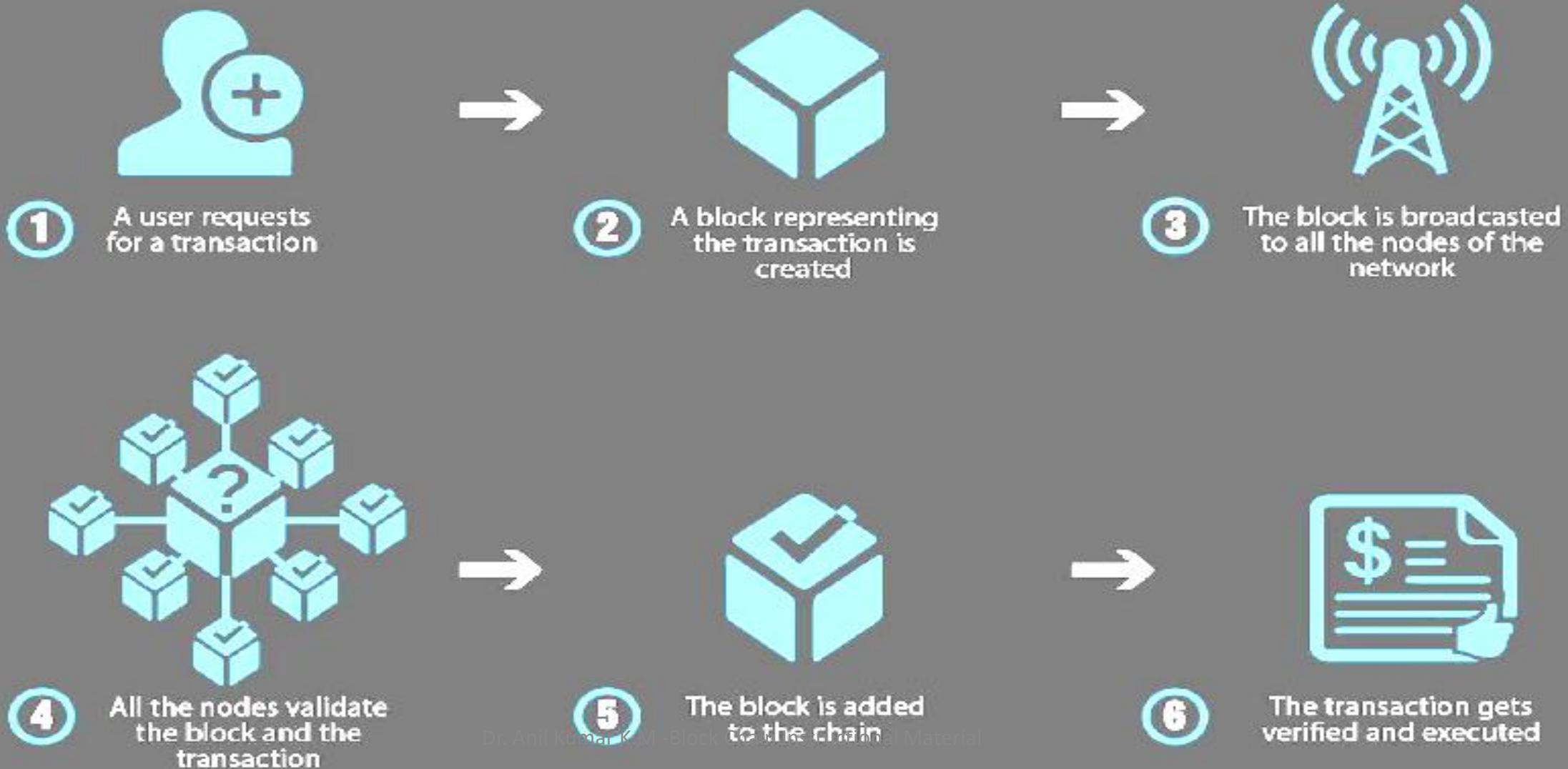
Ar03vt\$97de9a7sfbr2 3r8b099e824w£5a18



The «mail carriers system» acknowledges the original as the
valid letter

Step by Step Working of Blockchain

How Does a Blockchain Work: A Step-by-Step View



Potential Business features of Block chain

Improving profitability and quality

- Automation using smart contracts / algorithms
- Traceability of all historical transactions
- Speed and efficiency of transactions by eliminating intermediaries
- Enhanced security by encryption of data at the stage of dissemination
- Prevents tampering as any tampering may leave behind trail

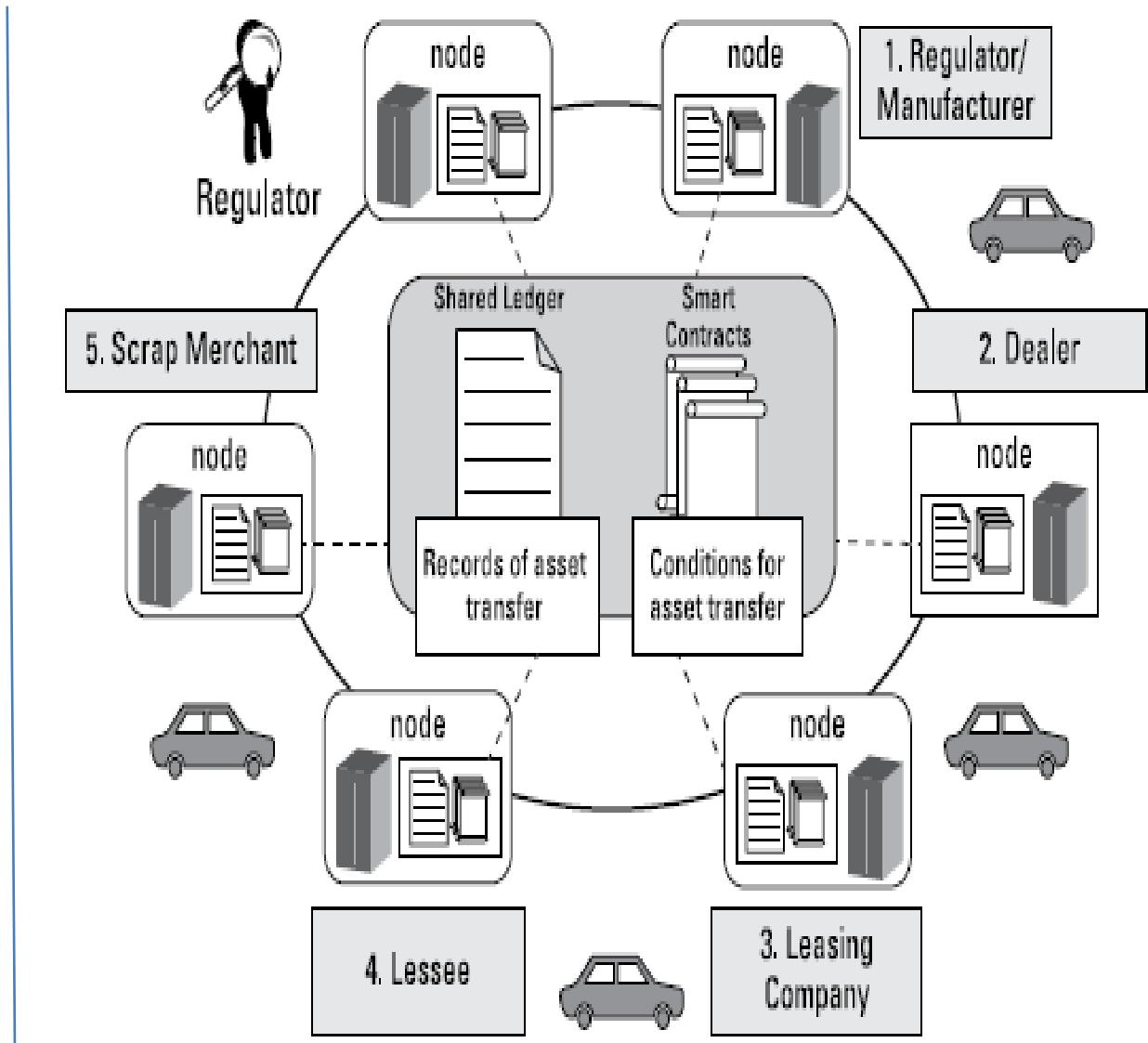
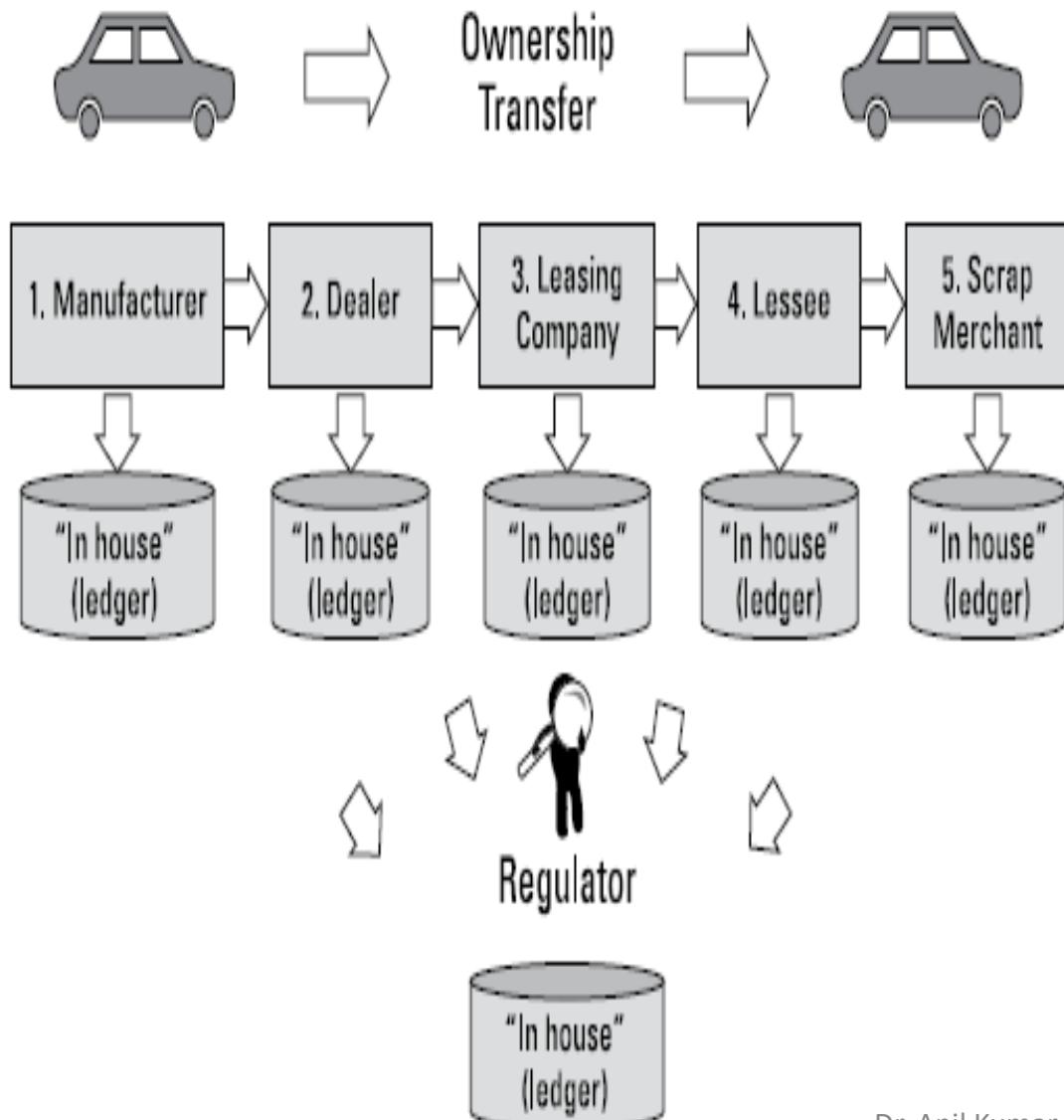
Increasing transparency

- Distributed ledger
- Provides a comprehensive picture: all stakeholders see the same information to which they have access
- Availability of multiple copies of the shared data

Reinventing products and processes

- Transparent and predefined rules which facilitates creation of new products / processes through a decentralized model
- Tokenization / Digital Assets which are physical objects with a unique digital representation that enable digital ownership, management and transfer

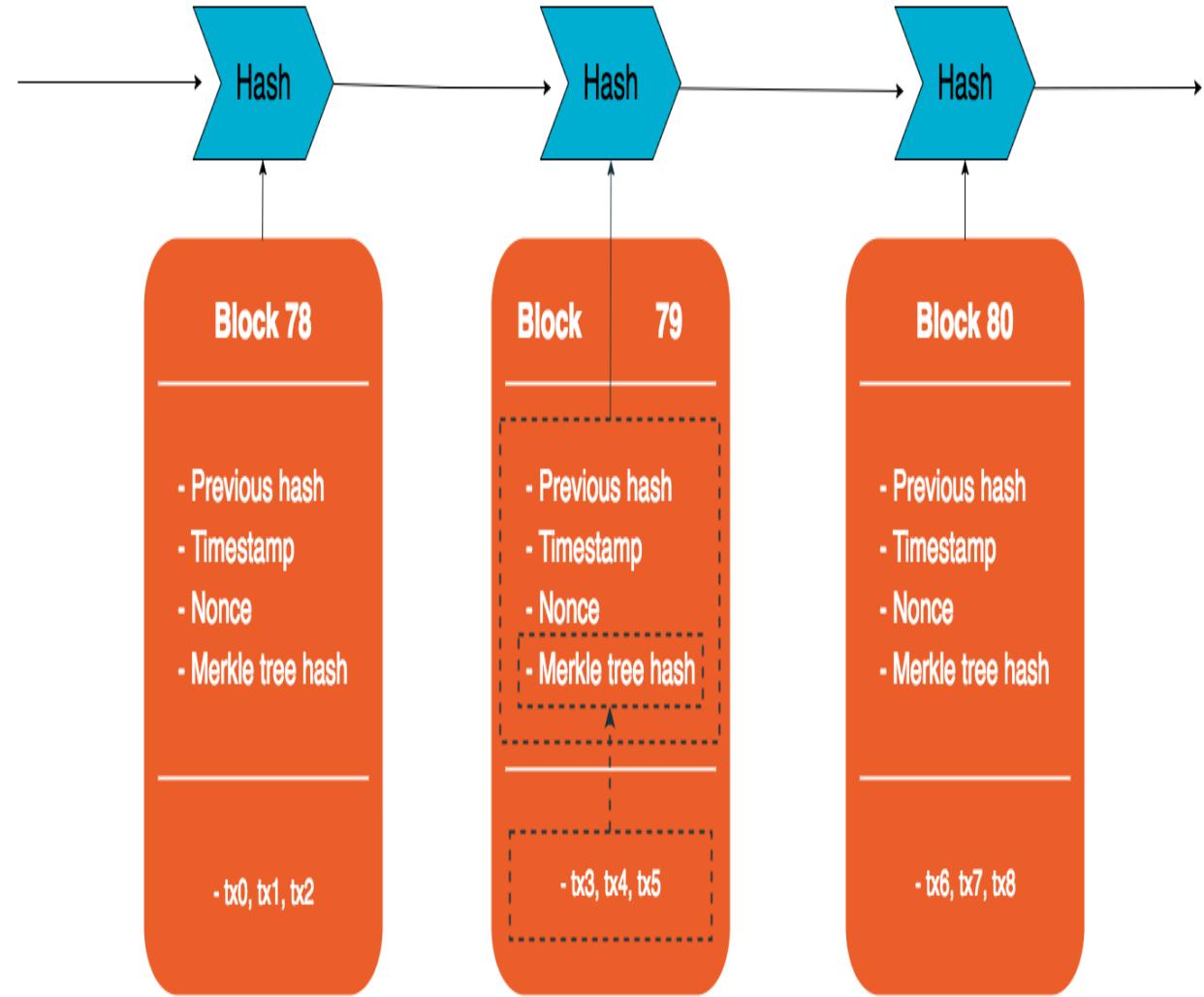
Example : Car Ownership without and with Block Chain



- Anatomy of a Block
 - Nodes and Network
 - Full Nodes & Light Nodes
 - Miners :
 - Nodes (find the valid block, store a copy and distribute it to other nodes) – Powerful Computers
 - Basic Operations
 - Validation of Transactions
 - Gathering transactions for a Block
 - Broadcasting a valid transaction and block
 - Consensus on next block creation
 - Chaining of blocks
- Role of participants
-
- ```
graph TD; A[Role of participants] --> B[Transaction]; A --> C[Miner]
```

## Block:

- The current version of the block
- Previous block header hash as a reference to the parent block
- An encrypted hash of all transactions taking place in this block known as Merkle root hash
- Time of the block creation
- nBits
- Nonce – any random number that is assigned by a block creator that can be changed as and when required



The remaining part in a block consists of transactions set by the data miner to include in the block.

- All the transactions taking place within a blockchain network stay in the blockchain as a **flat file or in database**. A particular set of transactions together forms a block and that block gets added into the blockchain.
- An interesting thing to note is how to efficiently store a huge number of transactions in a block? If we take all the transactions as it is and keep it in a block, the block size will get unmanageably large.
- Merkle tree and root are a solution to this problem. Merkle trees structure the data in such a way that at the end of it there is only one root representing the entire tree.

| BLOCK               |                               |                                                                          |
|---------------------|-------------------------------|--------------------------------------------------------------------------|
| Field               | Field Size                    | Description                                                              |
| Block header        | 80 bytes                      | Contains hash, previous block hash, timestamp, nonce, etc type metadata. |
| Block size          | 4 bytes                       | Shows the size of the whole block.                                       |
| Transaction counter | 1-9 bytes                     | Shows the total number of transactions contained in the block.           |
| Transactions        | Variable (at least 400 bytes) | Contains all the transactions of the block.                              |

Transactions Hashed in Merkle Tree

# How Block chain establish Trust

- └ Validation
  - Miners get 6.26 BTC per Block ( May 2020)
  - Miner get 12.5 BTC per Block (Earlier)
- └ Verification
  - 1 Bit coin = 7,66,308.13 Indian Rupee  
(22.9.2020)
- └ Consensus
- └ Immutable recording

# Categories of Block chain

- Public Block Chain

Example: Bitcoin, Ethereum, Litecoin

- Private

Example: voting, supply chain management, digital identity, asset ownership, etc.

- Permissioned (Consortium)

Example: banks, government organizations

- More than one manages the block chain network

- Hybrid Block Chain

Example: dragonchain  
<https://dragonchain.com/>

## Which is Better Private or Public or Hybrid?

# Categories of Block Chain (contd..)

- Permissionless:
  - They are decentralized ledger platforms open to anyone for publishing blocks.
  - They don't need permission from any authority for publishing
  - They are often open source software, freely available to anyone who wishes to download them
  - Anyone has the right to publish blocks
  - Malicious users may attempt to publish blocks in a way that subverts the system
  - To prevent this, permissionless blockchain networks often utilize a multiparty agreement or 'consensus' system

## Permissioned block chain

- Permissioned blockchain networks are ones where users publishing blocks must be authorized by some authority (be it centralized or decentralized)
- it is possible to restrict read access and to restrict who can issue transactions
- Permissioned blockchain networks may be instantiated and maintained using open source or closed source software

# Block Chain Components

## 1. Cryptographic Hash Function:

Hashing is a method of applying a cryptographic hash function to data, which calculates a relatively unique output (called a message digest, or just digest) for an input of nearly any size

important security properties are:

- a. They are preimage resistant. [This means that they are one-way; it is computationally infeasible to compute the correct input value given some output value]
- b. They are second preimage resistant. [cryptographic hash functions are designed so that given a specific input, it is computationally infeasible to find a second input which produces the same output]
- c. They are collision resistant. [This means that one cannot find two inputs that hash to the same output.]

## 2. Cryptographic Nonce

- A cryptographic nonce is an arbitrary number that is only used once.
- A cryptographic nonce can be combined with data to produce different hash digests per nonce:  $\text{hash}(\text{data} + \text{nonce}) = \text{digest}$
- Only changing the nonce value provides a mechanism for obtaining different digest values while keeping the same data.
- This technique is utilized in the (proof of work) consensus model

## 3. Transactions

- Transaction represents an interaction between parties.
- With cryptocurrencies, for example, a transaction represents a transfer of the cryptocurrency between blockchain network users.
- For business-to-business scenarios, a transaction could be a way of recording activities occurring on digital or physical assets

### 3. Asymmetric Key Cryptography

#### 4. Addresses and Address Derivation:

- Some blockchain networks make use of an address, which is a short, alphanumeric string of characters derived from the blockchain network user's public key using a cryptographic hash function, along with some additional data (e.g., version number, checksums).
- Most blockchain implementations make use of addresses as the "to" and "from" endpoints in a transaction.
- These Addresses are shorter than the public keys and are not secret.
- One method to generate an address is to create a public key, applying a cryptographic hash function to it, and converting the hash to text:
  - public key → cryptographic hash function → address

- Each blockchain implementation may implement a different method to derive an address.
- For permissionless blockchain networks, which allow anonymous account creation, a blockchain network user can generate as many asymmetric-key pairs, and therefore addresses as desired, allowing for a varying degree of pseudo-anonymity.
- Addresses may act as the public-facing identifier in a blockchain network for a user, and oftentimes an address will be converted into a QR code

## 5. Private Key Storage :

- Users must manage and securely store their own private keys
- Instead of recording them manually, they often use software to securely store them.
- This software is often referred to as a wallet.

- The wallet can store private keys, public keys, and associated addresses.
- It may also perform other functions, such as calculating the total number of digital assets a user may have.
- If a user loses a private key, then any digital asset associated with that key is lost, because it is computationally infeasible to regenerate the same private key.
- If a private key is stolen, the attacker will have full access to all digital assets controlled by that private key

## 6. Ledgers

- It is a collection of transactions
- Throughout history, pen and paper ledgers have been used to keep track of the exchange of goods and services.
- In modern times, ledgers have been stored digitally, often in large databases owned and operated by a centralized trusted third party

## Difference between Centrally Owned Ledger and Block Chain Network

- Centrally owned ledgers may be lost or destroyed; a user must trust that the owner is properly backing up the system
  - A blockchain network is distributed by design, creating many backup copies all updating and syncing to the same ledger data between peers
- Centrally owned ledgers may be on a homogeneous network, where all software, hardware and network infrastructure may be the same. Because of this characteristic, the overall system resiliency may be reduced
  - blockchain network is a heterogeneous network, where the software, hardware and network infrastructure are all different. Because of the many differences between nodes on the blockchain network, an attack on one node is not guaranteed to work on other nodes

- Centrally owned ledgers may be located entirely in specific geographic locations (e.g., all in one country). If network outages were to occur in that location, the ledger and services which depend on it may not be available.
  - A blockchain network can be comprised of geographically diverse nodes which may be found around the world. Because of this, and the blockchain network working in a peer-to-peer fashion, it is resilient to the loss of any node, or even an entire region of nodes.
- The transactions on a centrally owned ledger are not made transparently and may not be valid; a user must trust that the owner is validating each received transaction.
  - A blockchain network must check that all transactions are valid; if a malicious node was transmitting invalid transactions, others would detect and ignore them, preventing the invalid transactions from propagating throughout the blockchain network.

- The transaction list on a centrally owned ledger may not be complete; a user must trust that the owner is including all valid transactions that have been received.
  - A blockchain network holds all accepted transactions within its distributed ledger. To build a new block, a reference must be made to a previous block – therefore building on top of it. If a publishing node did not include a reference to the latest block, other nodes would reject it.
- The transaction data on a centrally owned ledger may have been altered; a user must trust that the owner is not altering past transactions
  - A blockchain network utilizes cryptographic mechanisms such as digital signatures and cryptographic hash functions to provide tamper evident and tamper resistant ledgers

## 7. Blocks

## 8. Chaining of Blocks

Blocks are chained together through each block containing the hash digest of the previous block's header, thus forming the blockchain.

If a previously published block were changed, it would have a different hash.

This in turn would cause all subsequent blocks to also have different hashes since they include the hash of the previous block.

This makes it possible to easily detect and reject altered blocks

## 9. Consensus Models /Mechanism And trust Frame Work

- Proof – of –Work Consensus Model
- Proof –of- Stake Consensus Model
- Round Robin Consensus Model
- Proof of Authority/Proof of Identity Consensus Model
- Proof of Elapsed Time Consensus Model



## Proof – of –Work Consensus Model

- In the proof of work (PoW) model, a user publishes the next block by being the first to solve a computationally intensive puzzle.
- The solution to this puzzle is the “proof” of their performed work.
- The puzzle is designed such that solving the puzzle is difficult, but checking that a solution is valid is easy.
- This enables all other full nodes to easily validate any proposed next blocks, and any proposed block that did not satisfy the puzzle would be rejected.
- A common puzzle method is to require that the hash digest of a block header be less than a target value.
- Publishing nodes make many small changes to their block header (e.g., changing the nonce) trying to find a hash digest that meets the requirement. For each attempt, the publishing node must compute the hash for the entire block header.

- Hashing the block header many times becomes a computationally intensive process. The target value may be modified over time to adjust the difficulty (up or down) to influence how often blocks are being published
- Adjustments to the difficulty target is done to ensure that no entity can take over block production.
- When a user receive completed block from another user, they need to discard it and proceed.

As an example, consider a puzzle where, using the SHA-256 algorithm, a computer must find a hash value meeting the following target criteria (known as the difficulty level):

**SHA256("blockchain" + Nonce) = Hash Digest starting with "000000"**

SHA256("blockchain0") = 0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab938 (not solved)

SHA256("blockchain1") = 0xdb0b9c1cb5e9c680dff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10 (not solved)

...

SHA256("blockchain10730895") = 0x**000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d67587** (solved)

- To solve this puzzle, it took 10,730,896 guesses (completed in 54 seconds on relatively old hardware, starting at 0 and testing one value at a time).
- In this example, each additional “leading zero” value increases the difficulty. By increasing the target by one additional leading zero (“0000000”), **the same hardware took 934,224,175 guesses to solve the puzzle (completed in 1 hour, 18 minutes, 12 seconds)**:
- There is currently no known shortcut to this process
- Publishing nodes must expend computation effort, time, and resources to find the correct nonce value for the target.
- Publishing nodes normally attempt to solve this computationally difficult puzzle to claim a reward of some sort (usually in the form of a cryptocurrency offered by the blockchain network).
- The prospect of being rewarded for extending and maintaining the blockchain is referred to as a reward system or incentive model.

- Once a publishing node has performed this work, they send their block with a valid nonce to full nodes in the blockchain network.
- The recipient full nodes verify that the new block fulfills the puzzle requirement, then add the block to their copy of the blockchain and resend the block to their peer nodes.
- In this manner, the new block gets quickly distributed throughout the network of participating nodes.
- Verification of the nonce is easy since only a single hash needs to be done to check to see if it solves the puzzle
- For many proof of work based blockchain networks, publishing nodes tend to organize themselves into “pools” or “collectives” whereby they work together to solve puzzles and split the reward.
- This is possible because work can be distributed between two or more nodes across a collective to share the workload and rewards.
- Splitting the example program into quarters, each node can take an equal amount of the nonce value range to test.

- Node 1: check nonce 0000000000 to 0536870911
- Node 2: check nonce 0536870912 to 1073741823
- Node 3: check nonce 1073741824 to 1610612735
- Node 4: check nonce 1610612736 to 2147483647

The following result was the first to be found to solve the puzzle:

SHA256("blockchain1700876653") =

**0x0000003ba55d20c9cbd1b6fb34dd81c3553360ed918d07acf16dc9e75d7c7f1**

- This is a completely new nonce, but still one that solved the puzzle. It took 90,263,918 guesses (completed in 10 minutes, 14 seconds).
- Dividing up the work amongst many more machines yields much better results, as well as more consistent rewards in a proof of work model.

- “Sybil Attack” – a computer security attack (not limited to blockchain networks ) where an attacker can create many nodes (i.e., creating multiple identities) to gain influence and exert control.
- The proof of work model combats this by having
- Amount of computational power (hardware, which costs money)
- Mixed with a lottery system
- Network identities

<https://www.youtube.com/watch?v=-EKhIBUQjcA>

## Proof of Stake Consensus Model

- The proof of stake (PoS) model is based on the idea that the more stake a user has invested into the system, the more likely they will want the system to succeed, and the less likely they will want to subvert it.
- Stake is often an amount of cryptocurrency that the blockchain network user has invested into the system (through various means, such as by locking it via a special transaction type, or by sending it to a specific address, or holding it within special wallet software).
- Once staked, the cryptocurrency is generally no longer able to be spent.
- Proof of stake blockchain networks use the amount of stake a user has, as a determining factor for publishing new blocks.
- Thus, the likelihood of a blockchain network user publishing a new block is tied to the ratio of their stake to the overall blockchain network amount of staked cryptocurrency
- With this consensus model, there is no need to perform resource intensive computations (involving time, electricity, and processing power) as found in proof of work. Since this consensus model utilizes fewer resources, some blockchain networks have decided to forego a block creation reward;

The reward for block publication is then usually the earning of user provided transaction fees.

Blockchain network uses the stake and can vary. There are four approaches: **random selection** of staked users, **multi-round voting**, **coin aging systems** and **delegate systems**. Regardless of the exact approach, users with more stake are more likely to publish new blocks.

1. When the choice of block publisher is a **random choice** (sometimes referred to as *chain-based proof of stake*), the blockchain network will look at all users with stake and choose amongst them based on their ratio of stake to the overall amount of cryptocurrency staked. So, if a user had 42% of the entire blockchain network stake they would be chosen 42 % of the time; those with 1 % would be chosen 1 % of the time.
2. When the choice of block publisher is a **multi-round voting system** (sometime referred to as Byzantine fault tolerance proof of stake) there is added complexity. The blockchain network will select several staked users to create proposed blocks. Then all staked users will cast a vote for a proposed block. Several rounds of voting may occur before a new block is decided upon. This method allows all staked users to have a voice in the block selection process for every new block.
3. When the choice of block publisher is through a **coin age system** referred to as a coin age proof of stake, staked cryptocurrency has an age property. After a certain amount of time (such as 30 days) the staked cryptocurrency can count towards the owning user being selected to publish the next block. The staked cryptocurrency then has its age reset, and it cannot be used again until after the requisite time has passed.

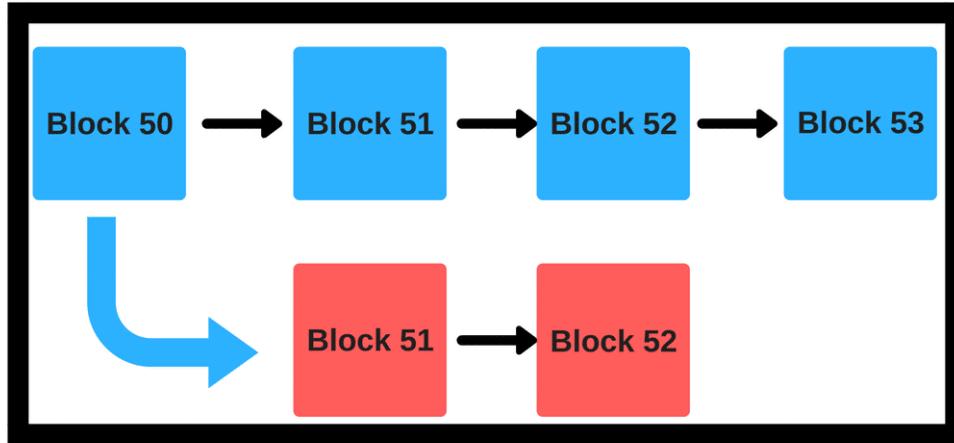
This method allows for users with more stake to publish more blocks, but to not dominate the system – since they have a cooldown timer attached to every cryptocurrency coin counted towards creating blocks

4. When the choice of block publisher is through a delegate system, users vote for nodes to become publishing nodes – therefore creating blocks on their behalf.

- Blockchain network users' voting power is tied to their stake so the larger the stake, the more weight the vote has.
- Nodes who receive the most votes become publishing nodes and can validate and publish blocks.
- Blockchain network users can also vote against an established publishing node, to try to remove them from the set of publishing nodes.
- Voting for publishing nodes is continuous and remaining a publishing node can be quite competitive.
- The threat of losing publishing node status, and therefore rewards and reputation is constant so publishing nodes are incentivized to not act maliciously.
- Additionally, blockchain network users vote for delegates, who participate in the governance of the blockchain.
- Delegates will propose changes, and improvements, which will be voted on by blockchain network users.

It is worth noting that a problem known as “nothing at stake” may arise from some proof of stake algorithms

What is nothing at stake ?



In the diagram above we have the main chain (blue) which has been mined till block #53. However, there is a parallel branch originating from block #50 (red). What will happen if some malicious miners get together and keep mining on the red chain until it overtakes the blue one? All the transactions that have taken place in block 51, 52, and 53 will be instantly null and voided.

In a Proof of Stake system, this risk can be mitigated

Suppose malicious miner Alice wants to mine on the red chain. Even if she dedicates all of her hash power to it, she won't get any other miner to join her on the new chain. Everyone else will still continue to mine on the blue chain because it is more profitable and risk-free to mine on the longer chain.

Now, remember, POW is extremely expensive resource-wise. It makes no sense for a miner to waste so many resources on a block that will be rejected by the network anyway. Hence chain splits are avoided in a proof of work system because it will be extremely expensive.

However, things look a little different when you bring in Proof of Stake. If you are a validator, then you can simply put your money in both the red chain and blue chain without any fear of repercussion at all. No matter what happens, you will always win and have nothing to lose, despite how malicious your actions maybe.

This is called the “Nothing at Stake” problem, and this is something that Block chain / Ethereum had to address. They needed a protocol that could implement POS and mitigate the “Nothing at Stake” problem - **Enter Casper Protocol**

## **Round Robin Consensus Model**

- Round Robin is a consensus model that is used by some permissioned blockchain networks.
- Within this model of consensus, nodes take turns in creating blocks.
- Round Robin Consensus has a long history grounded in distributed system architecture.
- To handle situations where a publishing node is not available to publish a block on its turn, these systems may include a time limit to enable available nodes to publish blocks so that unavailable nodes will not cause a halt in block publication. This model ensures no one node creates the majority of the blocks. It benefits from a straightforward approach, lacks cryptographic puzzles, and has low power requirements

- Since there is a need for trust amongst nodes, round robin does not work well in the permissionless blockchain networks used by most cryptocurrencies.
- This is because malicious nodes could continuously add additional nodes to increase their odds of publishing new blocks. In the worst case, they could use this to subvert the correct operation of the blockchain network.

### **Proof of Authority/Proof of Identity Consensus Model**

*“It takes 20 years to build a reputation and five minutes to ruin it. If you think about that, you’ll do things differently.”*

- The proof of authority (also referred to as proof of identity) consensus model relies on the partial trust of publishing nodes through their known link to real world identities.

- Publishing nodes must have their identities proven and verifiable within the blockchain network (e.g., identifying documents which have been verified and notarized and included on the blockchain).
- The idea is that the publishing node is staking its identity/reputation to publish new blocks.
- Block chain network users directly affect a publishing node's reputation based on the publishing node's behavior.
- Publishing nodes can lose reputation by acting in a way that the blockchain network users disagree with, just as they can gain reputation by acting in a manner that the block chain network users agree with.
- The lower the reputation, the less likelihood of being able to publish a block.
- Therefore, it is in the interest of a publishing node to maintain a high reputation. This algorithm only applies to permissioned block chain networks with high levels of trust.

## **Proof of Elapsed Time Consensus Model**

Within the proof of elapsed time (PoET) consensus model, each publishing node requests a wait time from a secure hardware time source within their computer system.

The secure hardware time source will generate a random wait time and return it to the publishing node software.

Publishing nodes take the random time they are given and become idle for that duration. Once a publishing node wakes up from the idle state, it creates and publishes a block to the blockchain network and the entire process starts over.

This model requires ensuring that a random time was used, since if the time to wait was not selected at random, a malicious publishing node would just wait the minimum amount of time by default to dominate the system. This model also requires ensuring that the publishing node waited the actual time and did not start early.

These requirements are being solved by executing software in a trusted execution environment found on some computer processors (such as Intel's Software Guard Extensions<sup>5</sup>, or AMD's Platform Security Processor<sup>6</sup>, or ARM's TrustZone).

Verified and trusted software can run in these secure execution environments and cannot be altered by outside programs.

A publishing node would query software running in this secure environment for a random time and then wait for that time to pass.

After waiting the assigned time, the publishing node could request a signed certificate that the publishing node waited the randomly assigned time.

The publishing node then publishes the certificate along with the block.

| Name                    | Goals                                                                                                                                                                                                                        | Advantages                                                                                                                                   | Disadvantages                                                                                                                                                                                                                                                                                                         | Domains                                                     | Implementations                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------|
| Proof of work<br>(PoW)  | To provide a barrier to publishing blocks in the form of a computationally difficult puzzle to solve to enable transactions between untrusted participants.                                                                  | Difficult to perform denial of service by flooding network with bad blocks.<br><br>Open to anyone with hardware to solve the puzzle.         | Computationally intensive (by design), power consumption, hardware arms race.<br><br>Potential for 51 % attack by obtaining enough computational power.                                                                                                                                                               | Permissionless cryptocurrencies                             | Bitcoin, Ethereum, many more   |
| Proof of stake<br>(PoS) | To enable a less computationally intensive barrier to publishing blocks, but still enable transactions between untrusted participants.                                                                                       | Less computationally intensive than PoW.<br><br>Open to anyone who wishes to stake cryptocurrencies.<br><br>Stakeholders control the system. | Stakeholders control the system.<br><br>Nothing to prevent formation of a pool of stakeholders to create a centralized power.<br><br>Potential for 51 % attack by obtaining enough financial power.                                                                                                                   | Permissionless cryptocurrencies                             | Ethereum Casper, Krypton       |
| Delegated PoS           | To enable a more efficient consensus model through a "liquid democracy" where participants vote (using cryptographically signed messages) to elect and revoke the rights of delegates to validate and secure the blockchain. | Elected delegates are economically incentivized to remain honest<br><br>More computationally efficient than PoW                              | Less node diversity than PoW or pure PoS consensus implementations<br><br>Greater security risk for node compromise due to constrained set of operating nodes<br><br>As all delegates are 'known' there may be an incentive for block producers to collude and accept bribes, compromising the security of the system | Permissionless cryptocurrencies<br><br>Permissioned Systems | Bitshares, Steem, Cardano, EOS |

## Comparison of Consensus

| Name                         | Goals                                                                                                                                | Advantages                                                                                                                                                    | Disadvantages                                                                                                                                                                                                                                     | Domains                                          | Implementations                                                              |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------|
| Round Robin                  | Provide a system for publishing blocks amongst approved/trusted publishing nodes                                                     | Low computational power.<br>Straightforward to understand.                                                                                                    | Requires large amount of trust amongst publishing nodes.                                                                                                                                                                                          | Permissioned Systems                             | MultiChain                                                                   |
| Proof of Authority/Identity  | To create a centralized consensus process to minimize block creation and confirmation rate                                           | Fast confirmation time<br>Allows for dynamic block production rates<br>Can be used in sidechains to blockchain networks which utilize another consensus model | Relies on the assumption that the current validating node has not been compromised<br>Leads to centralized points of failure<br>The reputation of a given node is subject to potential for high tail-risk as it could be compromised at any time. | Permissioned Systems, Hybrid (sidechain) Systems | Ethereum Kovan testnet, POA Chain, various permissioned systems using Parity |
| Proof of Elapsed Time (PoET) | To enable a more economic consensus model for blockchain networks, at the expense of deeper security guarantees associated with PoW. | Less computationally expensive than PoW                                                                                                                       | Hardware requirement to obtain time.<br>Assumes the hardware clock used to derive time is not compromised<br><br>Given speed-of-light latency limits, true time synchronicity is essentially impossible in distributed systems [13]               | Permissioned Networks                            | Hyperledger Sawtooth                                                         |

## 51% Attack

- A **51% attack** is used to **describe the unfortunate event** that a group or single person gains more than 50% of the total mining power.
- If that happened in a Proof of Work blockchain like Bitcoin, it would **allow the person to make changes** to a particular block. If this person was a criminal, they could alter the block for their gain.
- A recent example of a 51% attack happened against the [\*\*Verge blockchain\*\*](#), which allowed the hacker to walk away with 35 million XVG coins. At the time of the attack, this amounted to a **real-world value of \$1.75 million!**
- When using a **Proof of Stake** consensus mechanism, it **would not make financial sense** to attempt to perform a 51% attack. For this to be achieved, the bad actor would need to stake at least 51% of the total amount of cryptocurrency in circulation. The only way they could do this is to purchase the coins on the open market.
- If they decided to buy an amount this substantial, then the *real-world value of the coin would increase along the way*. As a result, they would **end up spending significantly more** than they could gain from the attack.

# Block Chain Interoperability

- Two Way Program

## When to Use Block chain ?

- Database
- Multiple Writer
- Unknown/ untrusted
- Third party

# Blockchain and Database

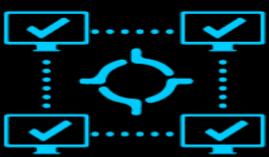
No one has the central authority.



Modifying data or asset is nearly impossible.



All the data or activity is out in the open for everyone to see.



Cuts down the excessive costing.



Blockchains are slow.



Suited for an organization where users don't trust each other.



Selected groups of individuals have authoritative control.



Data or assets can be easily changed.



All the data or transactions are hidden from each other.



Implementing process is costly.



Databases are comparatively faster.



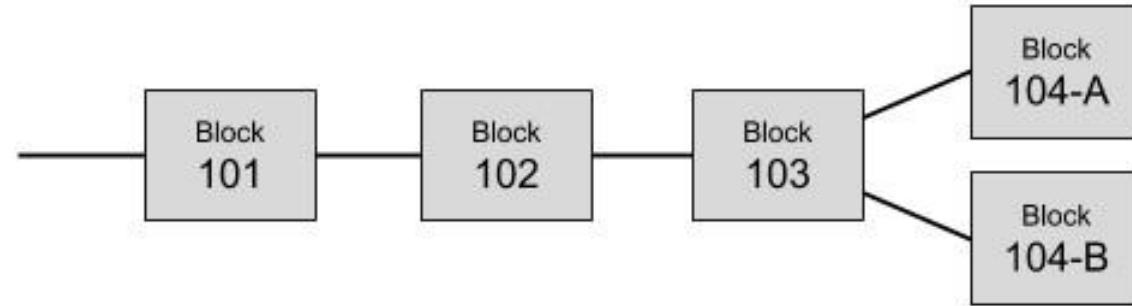
Suited for an organization where there is mutual trust.



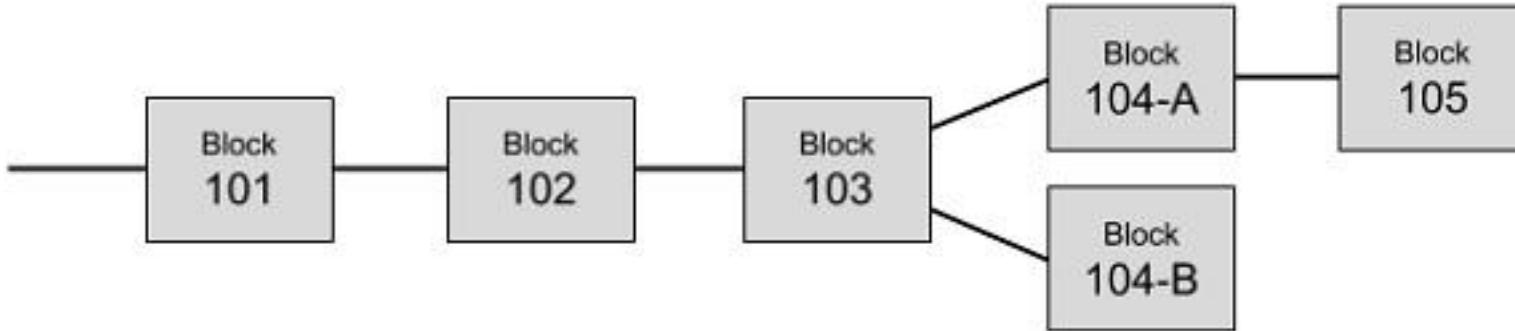
# Ledger Conflicts and Resolutions

What are conflicts?

- It is possible, that the two different miners solve the Proof-of-Work at the same time and thus add their blocks to the last known block in the chain.



- Now, we have two branches after Block 103. Both the branches are valid. So the next mined block may be added in either of the branches. Suppose, the miner adds the newly mined block to Block 104-A, the branch containing Block 104-A will be longer than the branch containing Block 104-B.



In Bitcoin architecture, the longest branch always wins and the shorter ones are purged. So the Block 104-B has to be purged. Before purging this block, all transactions in this block will be returned to the transaction pool so that they are mined and added to some future block. This is how the conflicts are resolved and only one single chain of blocks is maintained by the system.

# Forking

- Forking is a term that refers to a situation where a cryptocurrency or token project needs to make technical updates to its own code.
- These updates will either be applied to the backend of a project with no major changes in service, or they will fundamentally change the scope of the original project.
- Forking implies any divergence(divison) in Blockchain- temporary or permanent. Very simply, forking is said to happen when a Blockchain splits into two branches.
- It can happen as a result of a change in consensus algorithm or other software changes.
- Depending on the nature of change, the fork can be categorized into Hard Fork and Soft Fork

## **Hard Fork**

A hard fork is a permanent divergence from the previous version of the Blockchain, and nodes running previous versions will no longer be accepted by the newest version.

A hard fork is a radical change to the protocol that makes previously valid blocks or transactions invalid.

Any transaction on the forked (newer) chain will not be valid on the older chain.

All nodes and miners will have to upgrade to the latest version of the protocol software if they wish to be on the new forked chain.

This essentially creates a fork in the Blockchain, one path which follows the new, upgraded Blockchain, and one path which continues along the old path.

Hard Fork is usually done only when there is enough support from the mining community. Only when the majority of miners give positive signal towards the upgrade or fork, the developers of the chain starts work on the upgraded code.

Typically, the support should come from 90 to 95 percent of the miners

Fork can occur because of

- Block Size
- Proof of Work
- Bitcoin Reward per block and many more

A good example of a Hard Fork was when **Bitcoin Cash** came into existence. Previously Bitcoin was the dominant player in the cryptocurrency game. But as transaction times slowed and fees started to rise, the future of Bitcoin came into question.

Generally, there are two different ways which a hard fork can occur. It can either be: Contentious or planned

A planned hard fork means that the creators/developers were all in agreement as to how the coin would be forked. When this occurs, there is no split in the chain. This is because all of the nodes have agreed to upgrade to the latest version of the coin's client that possesses the rule change.

- There will be a split in the chain if all nodes do not adhere to the new rules set by the hard fork implementation. This chain will more than likely die off without enough economic support. Therefore, planned hard forks are usually under little to no threat unless there is an invisible negative sentiment.
- If there was disagreement in the community about the path of a coin, like Bitcoin, for example, then a *contentious hard fork* would occur. The most notable example of this is the creation of Bitcoin Cash.

Hard forks pose a systemic risk to Bitcoin:

When there are two running versions of the blockchain. This is something that any coin would want to prevent at any costs for the following reasons:

- Blockchain reorganization
- TX being confirmed by the wrong chains
- Political/Social Turmoil Within the Community Regarding the “real” and the “fake”
- And many others...

Bitcoin XT was effectively destroyed. There are only 11 Bitcoin XT nodes in service.

<https://coin.dance/nodes/xt>

- Bitcoin would remain unchanged from its original vision and stay a store of value, while Bitcoin Cash would become much faster and cheaper to use currency.
- The result is there are now two completely separate types of Bitcoins you can own. Each with its own value and its own prices.

Another Example : A well-known example of a hard fork is from Ethereum. In 2016, a smart contract was constructed on Ethereum called the Decentralized Autonomous Organization (DAO). Due to flaws in how the smart contract was constructed, an attacker extracted Ether, the cryptocurrency used by Ethereum, resulting in the theft of \$50 million [15]. A hard fork proposal was voted on by Ether holders, and the clear majority of users agreed to hard fork and create a new version of the blockchain, without the flaw, and that also returned the stolen funds.

## **Soft Fork:**

- A soft fork is said to happen when a change to the software protocol keeps it backward compatible.
- What this means is that the new forked chain will follow the new rules and will also honor the old rules.
- The original chain will continue to follow the old rules.
- This kind of fork requires only a majority of the miners upgrading to enforce the new rules, as opposed to a hard fork which requires (almost) all nodes to upgrade and agree on the new version.

- New transaction types can often be added as soft forks, requiring only that the participants for e.g. sender and receiver and miners understand the new transaction type.
- This is done by having the new transaction appear to older clients as a “pay-to-anybody” transaction (of a special form) and getting the miners to agree to reject blocks including this transaction unless the transaction validates under the new rules.
- A soft fork can also occur at times due to a temporary divergence in the Blockchain when miners using non-upgraded nodes violate a new consensus rule their nodes don’t know about.

## Cryptographic Changes and Forks

- If flaws are found in the cryptographic technologies within a blockchain network, the only solution may be to create a hard fork, depending on the significance of the flaw.
- For example, if a flaw was found in the underlying algorithms, there could be a fork requiring all future clients to use a stronger algorithm.
- Switching to a new hashing algorithm could pose a significant practical problem because it could invalidate all existing specialized mining hardware.
- Hypothetically, if SHA-256 were discovered to have a flaw, blockchain networks that utilize SHA-256 would need a hard fork to migrate to a new hash algorithm.
- The block that switched over to the new hash algorithm would “lock” all previous blocks into SHA-256 (for verification), and all new blocks would need to utilize the new hashing algorithm.

- There are many cryptographic hash algorithms, and blockchain networks can make use of whichever suits their needs. For example, while Bitcoin uses SHA-256, Ethereum uses Keccak-256
- One possibility for the need to change cryptographic features present in a blockchain network would be the development of a practical quantum computer system, which would be capable of greatly weakening (and in some cases, rendering useless) existing cryptographic algorithms.
- NIST Internal Report (NISTIR) 8105, Report on Post-Quantum Cryptography provides a table describing the impact of quantum computing on common cryptographic algorithms

## Impact of Quantum Computing on Common Cryptographic Algorithms

| Cryptographic Algorithm                   | Type          | Purpose                       | Impact from Large-Scale Quantum Computer |
|-------------------------------------------|---------------|-------------------------------|------------------------------------------|
| AES                                       | Symmetric key | Encryption                    | Larger key sizes needed                  |
| SHA-2, SHA-3                              | N/A           | Hash functions                | Larger output needed                     |
| RSA                                       | Public key    | Signatures, key establishment | No longer secure                         |
| ECDSA, ECDH (Elliptic Curve Cryptography) | Public key    | Signatures, key exchange      | No longer secure                         |
| DSA (Finite Field Cryptography)           | Public key    | Signatures, key exchange      | No longer secure                         |

## Smart Contracts:

- They are logic or code that operate on block chain
- They are part of the block chain in Ethereum
- They are immutable, it should be carefully coded
- Solidity is popular to code smart contract, EVM (Ethereum virtual machine) is the computer that runs the code.
- EVM is the engine of ethereum. Gas is the fuel for the engine.

In 1994, Nick Szabo, a legal scholar, and cryptographer realized that the decentralized ledger could be used for smart contracts, otherwise called self-executing contracts, blockchain contracts, or digital contracts.

In this format, contracts could be converted to computer code, stored and replicated on the system and supervised by the network of computers that run the blockchain.

This would also result in ledger feedback such as transferring money and receiving the product or service.

As Vitalik Buterin, founder of ethereum, says that, in a smart contract approach, an asset or currency is transferred into a program “and the program runs this code and at some point it automatically validates a condition and it automatically determines whether the asset should go to one person or back to the other person, or whether it should be immediately refunded to the person who sent it or some combination thereof

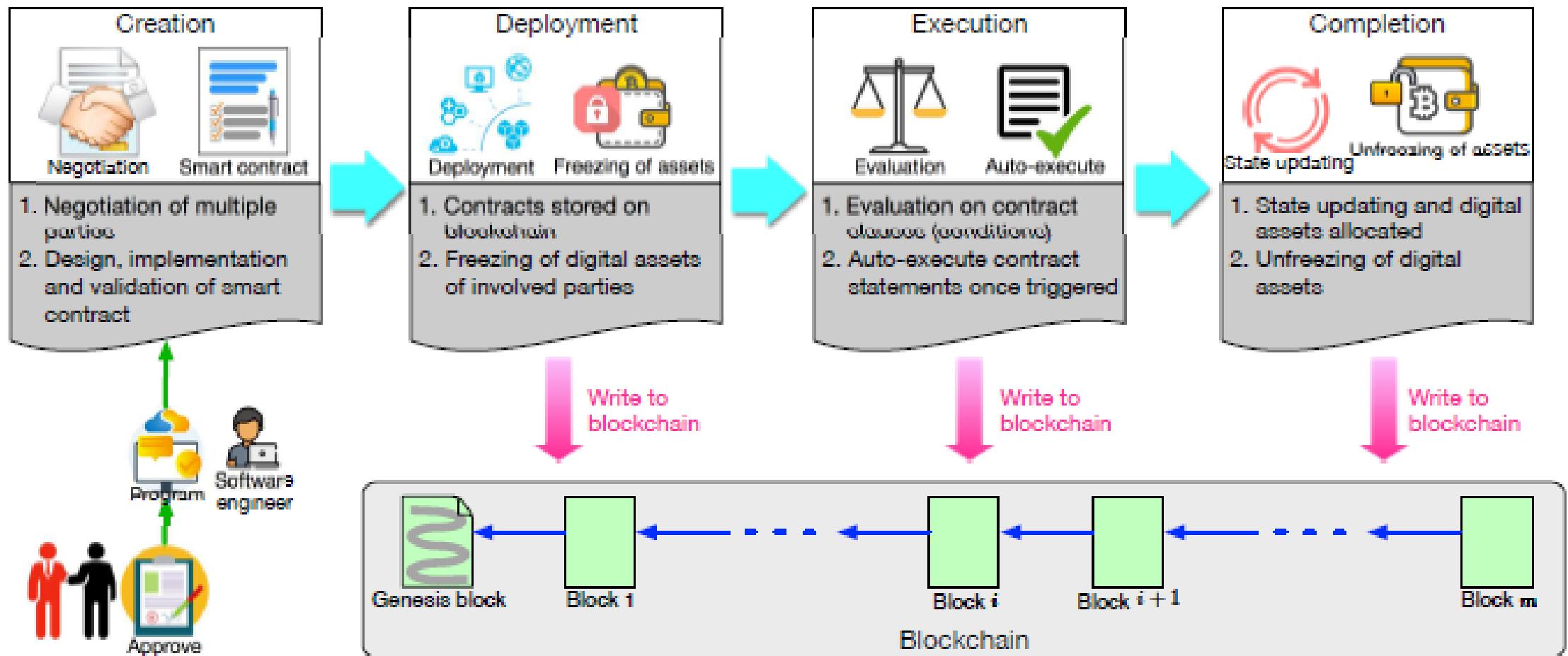
## Smart Contracts (contd..)

- Smart contracts are lines of code that are stored on a blockchain and automatically execute when predetermined terms and conditions are met.
- At the most basic level, they are programs that run as they've been set up to run by the people who developed them.
- The benefits of smart contracts are most apparent in business collaborations, in which they are typically used to enforce some type of agreement so that all participants can be certain of the outcome without an intermediary's involvement.

### How do they Work:

Smart contracts work by following simple “if/when...then...” statements that are written into code on a blockchain. A network of computers executes the actions (releasing funds to the appropriate parties; registering a vehicle; sending notifications; issuing a ticket) when predetermined conditions have been met and verified. The blockchain is then updated when the transaction is completed.

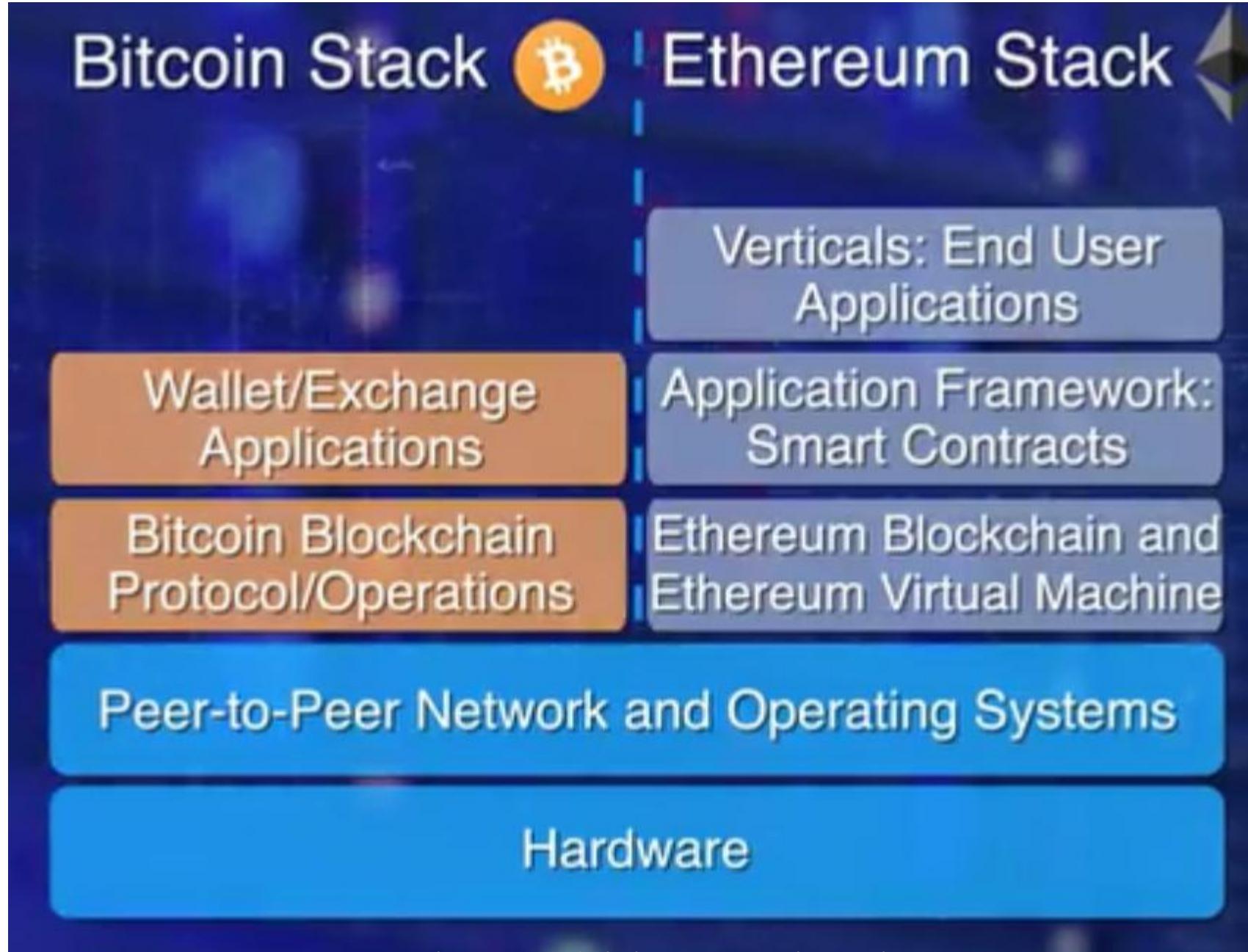
# Life Cycle of Smart Contracts

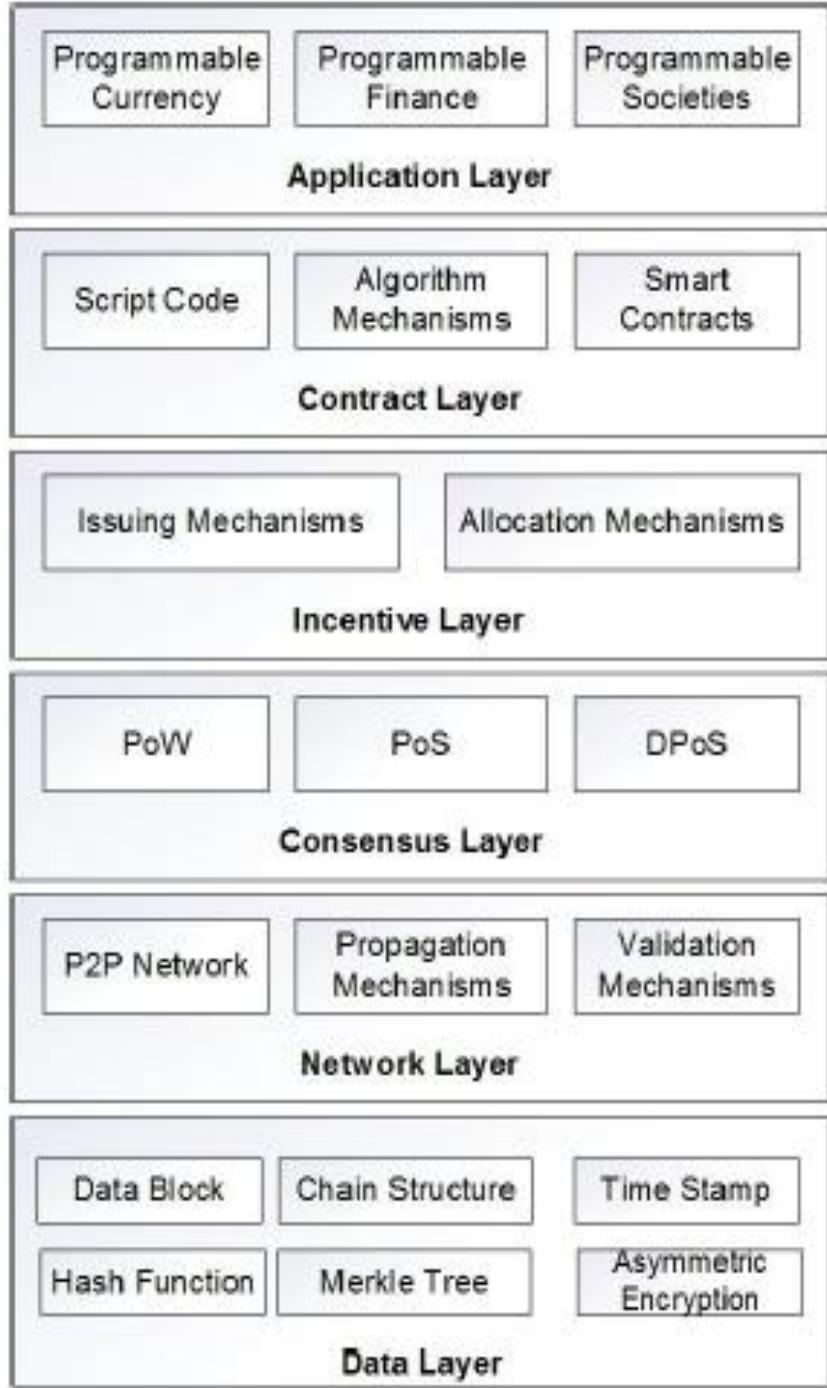


# Difference between Traditional and Smart Contracts

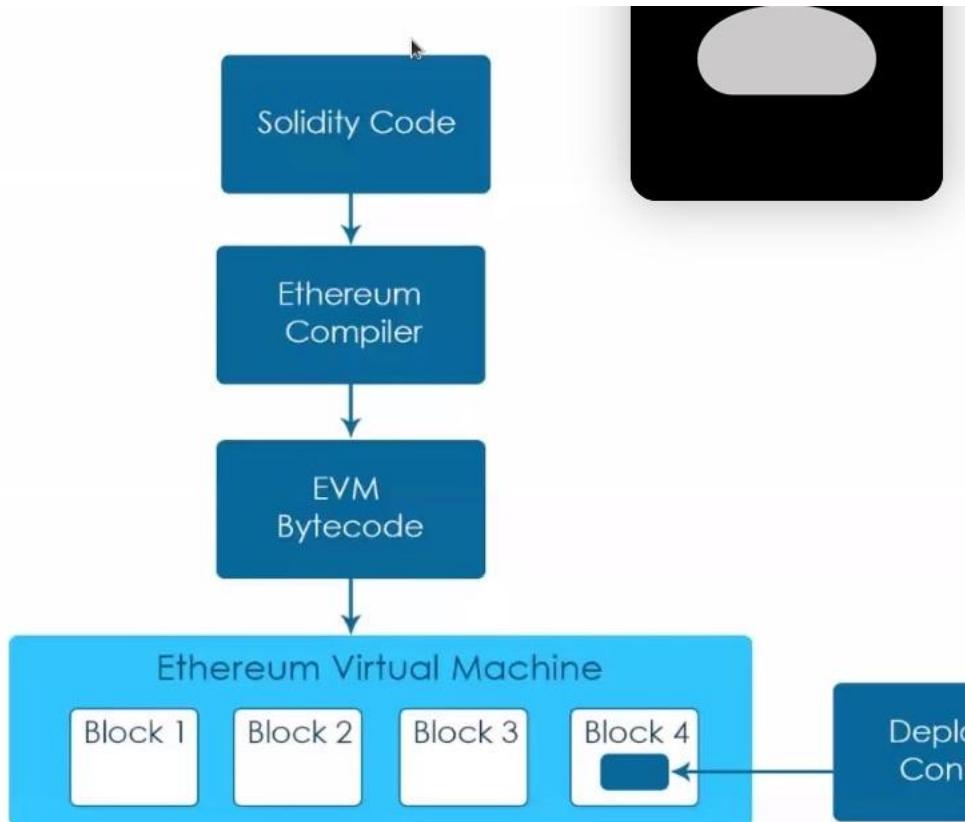
| Sl.No. | Traditional Contracts              | Smart Contracts                |
|--------|------------------------------------|--------------------------------|
| 1      | Created by legal team              | Created by Programmers         |
| 2      | Physical contracts                 | Digital contracts              |
| 3      | Legal language                     | Programming language           |
| 4      | Enforcement depends on Third party | Code is automatically executed |
| 5      | It takes days                      | It takes minutes               |
| 6      | Escrow may be necessary            | Escrow may not be necessary    |

Bit Coin  
and  
Ethereum  
Stack





\* An Overview of Smart Contract: Architecture, Applications, and Future Trends (2018)



## Ethereum Virtual Machine

## FEATURES OF SMART CONTRACT PLATFORMS

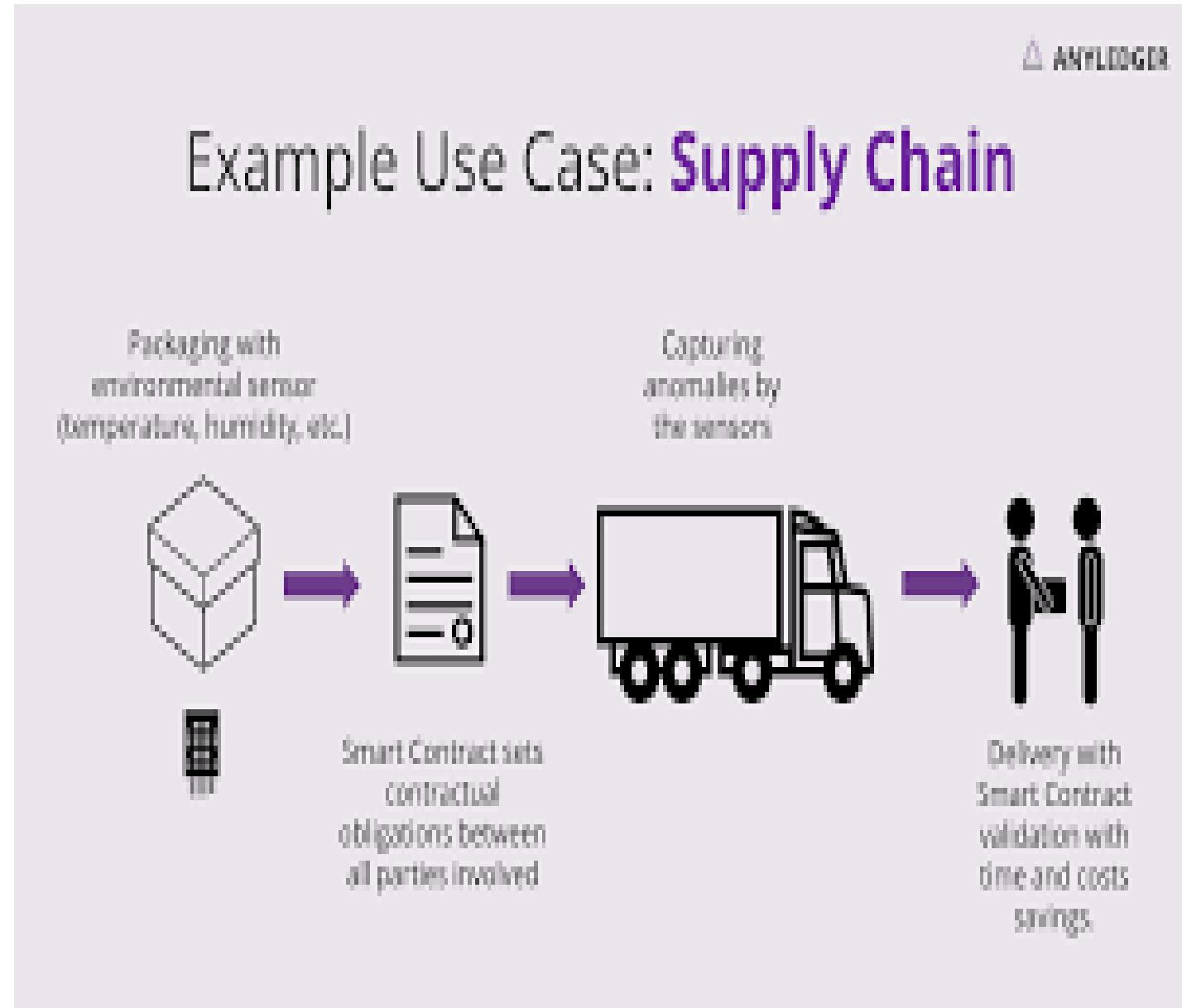
## COMPARISON OF SMART CONTRACT PLATFORMS

|               | <b>Application</b>                 | <b>Smart contract execution</b> | <b>Smart contract language</b>             | <b>Data model</b> | <b>Consensus</b>              |
|---------------|------------------------------------|---------------------------------|--------------------------------------------|-------------------|-------------------------------|
| Hyperledger   | Smart contract                     | Dockers                         | Golang, Java                               | Account-based     | PBFT                          |
| Ethereum      | Smart contract,<br>Crypto-currency | EVM                             | Solidity, Serpent, LLL                     | Account-based     | Ethash (PoW)                  |
| Eris-DB       | Smart contract                     | EVM                             | Solidity                                   | Account-based     | Tendermint (BFT)              |
| Ripple        | Crypto-currency                    | -                               | -                                          | UTXO-based        | Ripple Consensus Ledger (PoS) |
| ScalableBFT   | Smart contract                     | Haskell Execution               | Pact                                       | Account-based     | ScalableBFT                   |
| Stellar       | Smart contract                     | Dockers                         | JavaScript, Golang, Java, Ruby, Python, C# | Account-based     | Stellar Consensus Protocol    |
| Dfinity       | Smart contract                     | EVM                             | Solidity, Serpent, LLL                     | Account-based     | Blockchain Nervous System     |
| Parity        | Smart contract                     | EVM                             | Solidity, Serpent, LLL                     | Account-based     | Proof of Authority            |
| Tezos         | Smart contract,<br>Crypto-currency | Dockers                         | Tezos Contract Script Language             | Account-based     | Proof of Stake                |
| Corda         | Smart contract                     | JVM                             | Kotlin, Java                               | UTXO-based        | Raft                          |
| Sawtooth Lake | Smart contract                     | TEE                             | Python                                     | Account-based     | Proof of Elapsed Time         |

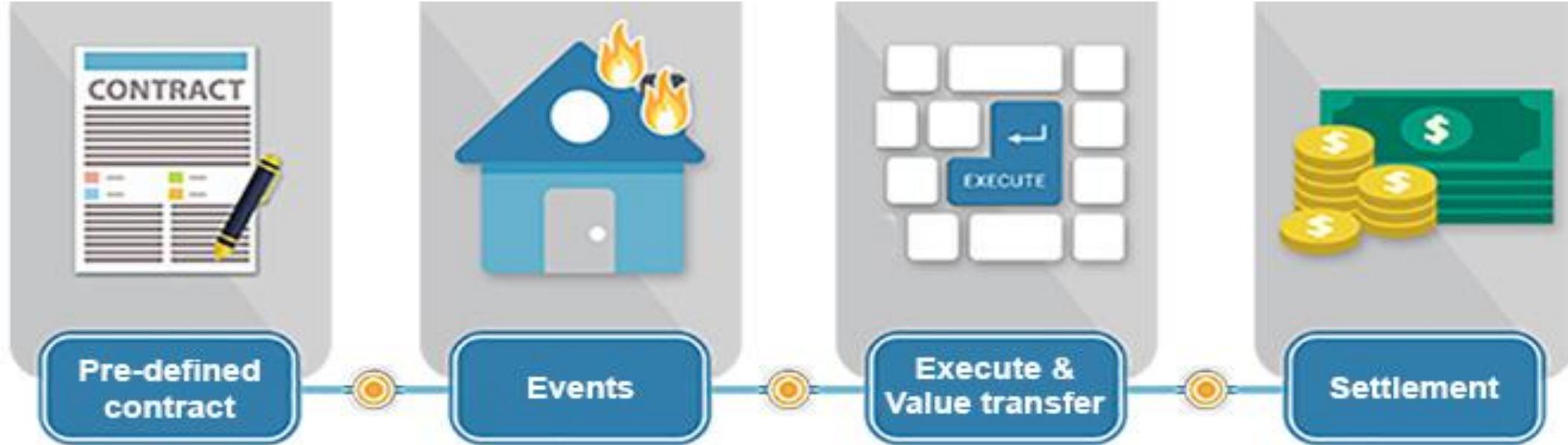
## Example : Supply Chain

Buyer B wants to buy something from Seller A, so she puts money in an escrow account. Seller A will use Shipper C to deliver the product to Buyer B. When Buyer B receives the item, the money in escrow will be released to Seller A and Shipper C.

If Buyer B doesn't receive the shipment by Date Z, the money in escrow will be returned. When this transaction is executed, Manufacturer G is notified to create items that was sold to increase supply. All this is done automatically.



## Example : Insurance Sector



- Terms of the policy are agreed by all counterparties
- These are hard coded into the smart contract and cannot be changed without all parties knowing
- Event triggers insurance policy execution
- The smart contract policy is automatically executed based on the pre-agreed terms
- Payout / other settlement completed instantly and efficiently

*Image Credit: Draglet GmbH*

**DIGITAL IDENTITY**

Provides individual identity in digital assets, removes counterfeits and also makes KYC frictionless

**FINANCIAL SECURITY**

Can be used for liability management, automatic payments, stock splits, dividends

**TRADE FINANCE**

Can be used for cross border payments, international trade

**FINANCIAL SERVICE**

Offer error-free services, automating many aspects

**FINANCIAL DATA RECORDING**

Improves data recording, accuracy, saves reporting and auditing costs

**GOVERNMENT**

Help automate operations, improves transparency and efficiency

**SUPPLY CHAIN MANAGEMENT**

Automates supply chain with visibility and transparency, leads to fewer frauds

**INSURANCE**

Automates claims and resolves disputes with proof

**CLINICAL TRIAL**

Offers cross-institutional visibility, automate data share and improves privacy

**ESCROW**

Automates escrow amount, authenticates and improves trust

**TRADING ACTIVITY**

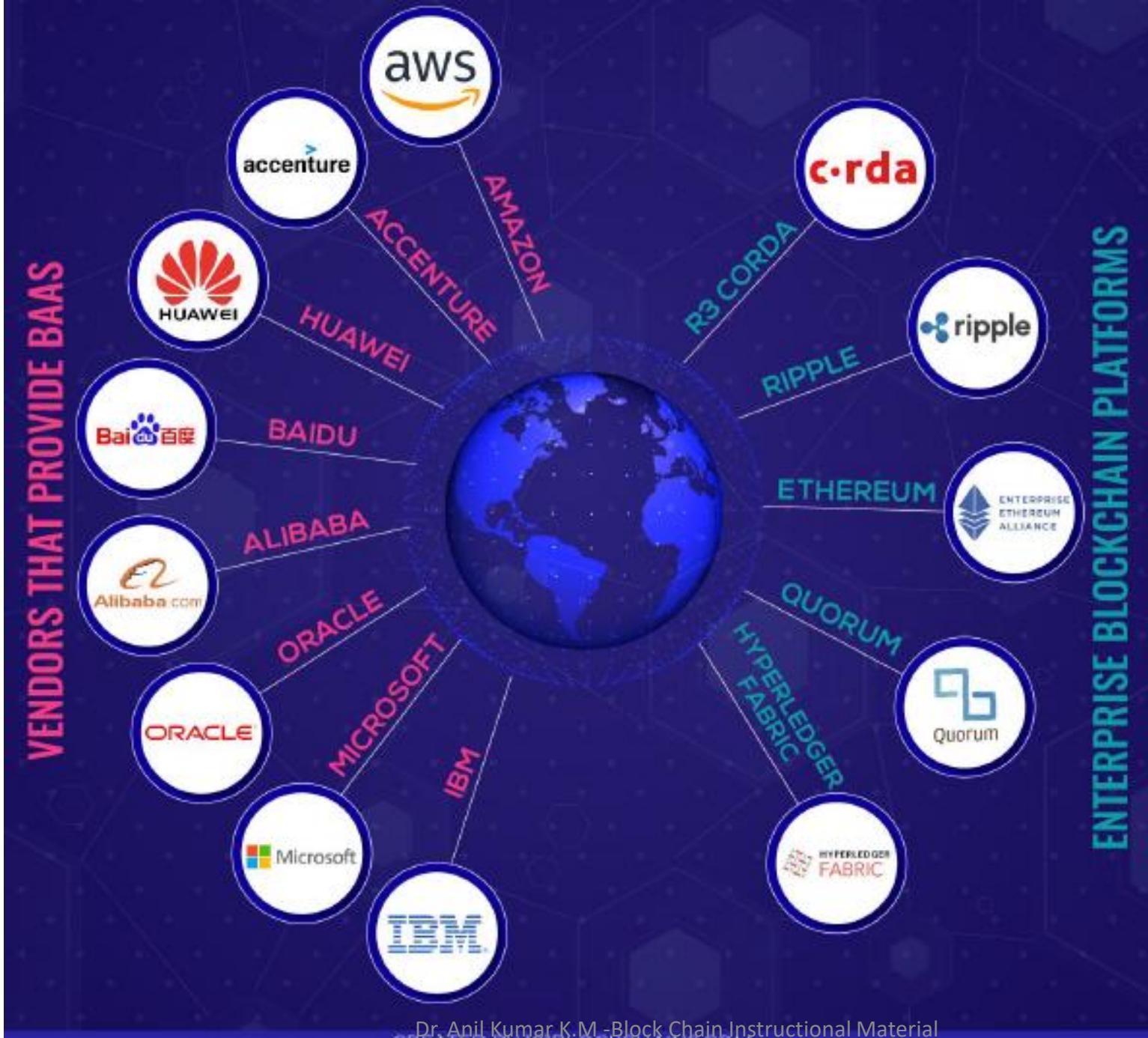
Trades can be automated without the need for intermediaries

**MORTGAGE SYSTEM**

Automates mortgage and fastens the process



|                            | Distributed Data Storage                           | Distributed Data Storage + Computing |
|----------------------------|----------------------------------------------------|--------------------------------------|
| <b>Value Token</b>         | Bitcoin (BTC)                                      | Ether (ETH)                          |
| <b>Block Time</b>          | 10 Minutes                                         | 10-20 Secs                           |
| <b>Block Size</b>          | ~1 MB                                              | Depends (~25KB)                      |
| <b>Scripting</b>           | Limited                                            | Smart Contract                       |
| <b>Economic Model</b>      | Shrinks $\frac{1}{2}$ every 2.1L Blocks in 4 years | Fixed                                |
| <b>Mining Reward</b>       | 50 > 25 > 12.5 > 6.25                              | 5 > 3 > 2                            |
| <b>Transaction Fees</b>    | Simple                                             | Complex                              |
| <b>Computational Power</b> | Distributed Storage                                | Turing Complete                      |



## URLs

<https://tools.superdatascience.com/blockchain/coinbase>

<https://blockchaindemo.io/>

Develop Use case for implementing smart contract

## What are the benefits of smart contracts?

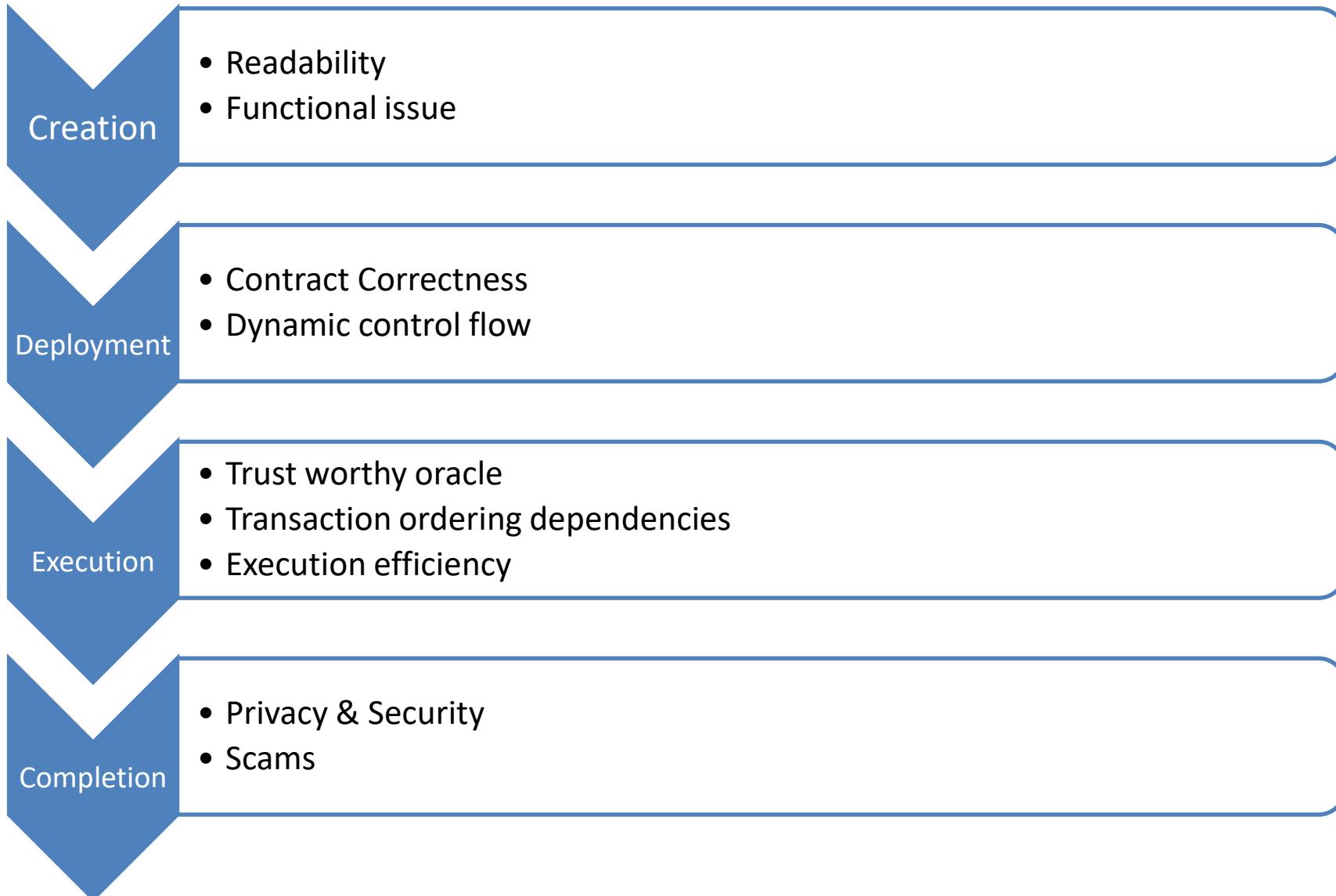
**Speed and accuracy:** Smart contracts are digital and automated, so you won't have to spend time processing paperwork or reconciling and correcting the errors that are often written into documents that have been filled manually. Computer code is also more exact than the legalese that traditional contracts are written in.

**Trust:** Smart contracts automatically execute transactions following predetermined rules, and the encrypted records of those transactions are shared across participants. Thus, nobody has to question whether information has been altered for personal benefit.

**Security:** Blockchain transaction records are encrypted, and that makes them very hard to hack. Because each individual record is connected to previous and subsequent records on a distributed ledger, the whole chain would need to be altered to change a single record.

**Savings:** Smart contracts remove the need for intermediaries because participants can trust the visible data and the technology to properly execute the transaction. There is no need for an extra person to validate and verify the terms of an agreement because it is built into the code.

# Challenges



## Readability

Most of smart contracts are written in programming languages such as Solidity, Go, Kotlin and Java. Then source codes will be compiled and executed. Therefore, in different time periods, programs have different forms of codes. How to make programs readable in each form remains a big challenge.

## Functional issues

- 1) Re-entrancy means that the interrupted function can be safely recalled again. Malicious users may exploit this vulnerability to steal digital currency.
- 2) Overcharging. Smart contracts can be overcharged due to features like dead code, expensive operations in loops consisting of repeated computations.

## Contract correctness

Once smart contracts have been deployed on block chains, it is nearly impossible to make any revisions. Therefore, it is of vital importance to evaluate the correctness of smart contracts before the formal deployment.

## Dynamic Control Flow

Despite the fact that the deployed smart contracts are immutable, the control flow of smart contracts is not guaranteed to be immutable. In particular, a smart contract can interact with other contracts (e.g., transferring funds to the contract or creating a new contract). The control flow of smart contract needs to be designed carefully when developing the contract. The interaction of smart contracts can result in an increased number of interconnected contracts over time. Therefore, how to predict the contract behaviours becomes challenging. In addition, most of existing methods pay attention to the detection of potential dynamic control flow problems in programs while the reliability of the execution environment is not always ensured. Therefore, it is also significant to check whether the execution environment is reliable.

# Execution

## Trustworthy Oracle

Smart contracts cannot work without real-world information. For example, an Eurobet (i.e., a soccer betting smart contract) needs to know the result of European Cup. However, a smart contract is designed to run in a sandbox isolating from the outside network. In a smart contract, an oracle plays a role of an agent who finds and verifies real-world occurrences and forwards this information to the smart contract. Thus, how to determine a trustworthy oracle becomes a challenge.

## Execution Efficiency

Smart contracts are serially executed by miners. In other words, a miner will not execute another contract until the current contract is completed. The execution serialization essentially limits the system performance. However, it is challenging to execute smart contracts concurrently due to the shared data between multiple smart contracts. In the meantime, how to inspect the contract data without prescribed interface is also important to improving the smart contract execution efficiency as it removes the need to redeploy a new contract.

### Privacy and Security

Most current smart contract and blockchain platforms lack of privacy-preserving mechanisms, especially for transactional privacy. In particular, the transaction records (i.e., the sequence of operations) are disseminated throughout the whole blockchain networks. Consequently, all the transactions are visible to everyone in the networks. Although some blockchain systems utilize pseudonymous public keys to improve the anonymity of the transactions, most transaction data (such as balances) are still publicly visible. it is possible to obtain useful information from the transaction data based on the transactional graph analysis.

Smart contract systems also have their inherent software vulnerabilities, which are susceptible to malicious attacks.

In addition, smart contracts run on top of blockchain systems which are also suffering from system vulnerability. For example, it is reported in literature that attackers exploited Border Gateway Protocol (BGP) routing scheme to intercept messages in block chains. It can cause high delay of message broadcasting and also hijack the traffic of a subset of nodes, thereby stealing digital currency.

### Scams

As a new technology, blockchain and smart contracts are vulnerable to malicious attacks initiated by scams. The detection of scams is of great importance especially for contract users since it enables them to terminate their investments at an early phase to avoid the unnecessary loss.

Ponzi scheme is a classical fraud which promises high return rates with little risk to investors. It pays the older investors with new investors' funds. But if there is no enough circulating money, the scheme unravels those posteriors who consequently lose their money.

Smart contract Honeypot implies that the vulnerable-looking contracts contain hidden traps.

## **Blockchain Limitations and Misconceptions:**

- There is a tendency to overhype and overuse most nascent technology. Many projects will attempt to incorporate the technology, even if it is unnecessary.
- This stems from the technology being relatively new and not well understood, the technology being surrounded by misconceptions, and the fear of missing out.
- Blockchain technology has not been immune to this.

## **Immutability:**

- Most publications on blockchain technology describe blockchain ledgers as being immutable. However, this is not strictly true.
- They are tamper evident and tamper resistant which is a reason they are trusted for financial transactions.
- They cannot be considered completely immutable, because there are situations in which the blockchain can be modified.
- We will look at different ways in which the concept of immutability for blockchain ledgers can be violated

The chain of blocks itself cannot be considered completely immutable. For some block chain implementations, the most recently published, or ‘tail’ blocks are subject to being replaced (by a longer, alternative chain with different ‘tail’ blocks).

Most blockchain networks use the strategy of adopting the longest chain (the one with the most amount of work put into it) as truth when there are multiple competing chains. If two chains are competing, but each include their own unique sequence of tail blocks, whichever is longer will be adopted.

However, this does not mean that the transactions within the replaced blocks are lost – rather they may have been included in a different block or returned to the pending transaction pool.

This degree of weak immutability for tail blocks is why most block chain network users wait several block creations before considering a transaction to be valid.

For permissionless blockchain networks, the adoption of a longer, alternate chain of blocks could be the result of a form of attack known as a 51 % attack.

For this, the attacker simply garners enough resources to outpace the block creation rate of rest of the blockchain network (holding more than 51 % of the resources applied towards producing new blocks). Depending on the size of the blockchain network, this could be a very cost prohibitive attack carried out by state level actors.

This attack is not technically difficult (e.g., it is just repeating the normal process of the blockchain implementation, but with selected transactions either included or omitted, and at a faster pace), it is just expensive.

For permissioned blockchain networks, this attack can be mitigated.

There is generally an owner or consortium of blockchain network users who allow publishing nodes to join the blockchain network and remove publishing nodes from the blockchain network, which gives them a great amount of control.

There is less likely to be competing chains since the owner or consortium can force publishing nodes to collaborate fairly since non-cooperating publishing nodes can simply have their privileges removed.

There are likely additional legal contracts in place for the blockchain network users which may include clauses for misconduct and the ability to take legal action.

While this control is useful to prevent misconduct, it means that any number of blocks can be replaced through legitimate methods if desired by the owner or consortium.

## **Users Involved in Blockchain Governance:**

The governance of blockchain networks deals with the rules, practices and processes by which the blockchain network is directed and controlled.

A common misconception is that blockchain networks are systems without control and ownership. The phrase “no one controls a blockchain!” is often exclaimed.

This is not strictly true. Permissioned blockchain networks are generally setup and run by an owner or consortium, which governs the blockchain network.

Permissionless blockchain networks are often governed by blockchain network users, publishing nodes, and software developers. Each group has a level of control that affects the direction of the blockchain network's advancement.

Software developers create the blockchain software that is utilized by a blockchain network. Since most blockchain technologies are open source, it is possible to inspect the source code, and compile it independently.

it is even possible to create separate but compatible software as a means of bypassing pre-compiled software released by developers.

However, not every user will have the ability to do this, which means that the developer of the blockchain software will play a large role in the blockchain network's governance.

These developers may act in the interest of the community at large and are held accountable. For example, in 2013 Bitcoin developers released a new version of the most popular Bitcoin client which introduced a flaw and started two competing chains of blocks..

The developers had to decide to either keep the new version (which had not yet been adopted by everyone) or revert to the old version.

Either choice would result in one chain being discarded—and some blockchain network user's transactions becoming invalid.

The developers made a choice, reverted to the old version, and successfully controlled the progress of the Bitcoin blockchain.

This example was an unintentional fork. Developers can purposely design updates to blockchain software to change the blockchain protocol or format.

With enough user adoption, a successful fork can be created. Such forks of blockchain software updates are often discussed at length and coordinated with the involved users.

For permissionless blockchain networks, this is usually the publishing nodes. There is often a long discussion and adoption period before an event occurs where all users must switch to the newly updated blockchain software at some chosen block to continue recording transactions on the new “main” fork.

For permissionless blockchain networks, although the developers maintain a large degree of influence, users can reject a change by the developers by refusing to install updated software.

Of the blockchain network users, the publishing nodes have significant control since they create and publish new blocks.

The user base usually adopts the blocks produced by the publishing nodes but is not required to do so.

An interesting side effect of this is that permissionless blockchain networks are essentially ruled by the publishing nodes and may marginalize a segment of users by forcing them to adopt changes they may disagree with to stay with the main fork.

For permissioned blockchain networks, control and governance is driven by members of the associated owner or consortium. The consortium can govern who can join the network, when members are removed from the network, coding guidelines for smart contracts etc.

In summary, the software developers, publishing nodes, and blockchain network users all play a part in the blockchain network governance.

## Beyond the Digital

Blockchain networks work extremely well with the data within their own digital systems. However, when they need to interact with the real world, there are some issues (often called the Oracle Problem).

A blockchain network can be a place to record both human input data as well as sensor input data from the real world, but there may be no method to determine if the input data reflects real world events.

A sensor could be malfunctioning and recording data that is inaccurate. Humans could record false information (intentionally or unintentionally). These issues are not specific to blockchain networks, but to digital systems overall.

However it is reported that for blockchain networks that are pseudonymous, dealing with data misrepresentation outside of the digital network can be especially problematic.

For example, if a cryptocurrency transaction took place to purchase a real-world item there is no way to determine within the blockchain network whether the shipment took place, without relying on outside sensor or human input.

Many projects have attempted to address the ‘Oracle problem’ and create reliable mechanisms to ingest external data in a way that is both trustworthy and accurate.

For example, projects like ‘Oraclize’ provide mechanisms to take web API data and convert it into blockchain readable byte/opcode. Within the context of decentralized applications, these projects may be considered centralized as they provide single points of failure for attackers to compromise – limitation.

As a result, projects like ‘Mineable Oracle Contract’ have recently arisen to enable oracle ingestion in a way that is inspired by blockchain technology and built atop established consensus models and economic incentives.

## **Block Chain Death**

Traditional centralized systems are created and taken down constantly, and blockchain networks will likely not be different.

However, because they are decentralized, there is a chance that when a blockchain network “shuts down” it will never be fully shut down, and that there may always be some lingering blockchain nodes running – with little publishing nodes, malicious users can take control.

## **Cybersecurity**

The use of blockchain technology does not remove inherent cybersecurity risks that require thoughtful and proactive risk management. Many of these inherent risks involve a human element.

Therefore, a robust cybersecurity program remains vital to protecting the network and participating organizations from cyber threats, particularly as hackers develop more knowledge about blockchain networks and their vulnerabilities.

Existing cybersecurity standards and guidance remain highly relevant for ensuring the security of systems that interface and/or rely on blockchain networks -protecting blockchain networks from cyberattacks.

In addition to general principles and controls, there are specific cybersecurity standards with relevance to blockchain technology which already exist and are in wide use by many industries.

For instance, the NIST Cybersecurity Framework expressly states that it is “not a one-size-fits-all approach to managing cybersecurity risk” because “organizations will continue to have unique risks—different threats, different vulnerabilities, different risk tolerances—and how they implement the practices in the [Framework] will vary.”

The Framework was not designed for blockchain technology specifically, its standards are broad enough to cover blockchain technology and to help institutions develop policies and processes that identify and control risks affecting blockchain technology.

Blockchain technologies are touted as being extremely secure due to the tamper evident and tamper resistant design – once a transaction is committed to the blockchain, it generally cannot be changed. However, this is only true for transactions which have been included in a published block.

Transactions that have not yet been included in a published block within the blockchain are vulnerable to several types of attacks. Spoofing time or adjusting the clock of a member of an ordering service could have positive or negative effects on a transaction, making time and the communication of time an attack vector. Denial of service attacks can be conducted on the blockchain platform or on the smart contract implemented on the platform.

Blockchain networks and their applications are not immune to malicious actors who can conduct network scanning to discover and exploit vulnerabilities and launch zero-day attacks. In the rush to deploy blockchain-based services, newly coded applications (like smart contracts) may contain new and known vulnerabilities and deployment weaknesses that will be discovered and then attacked through the network just like how websites or applications are attacked today.

## Malicious User

While a blockchain network can enforce transaction rules and specifications, it cannot enforce a user code of conduct.

This is problematic in permissionless blockchain networks, since users are pseudonymous and there is no a one-to-one mapping between blockchain network user identifiers and users of the system.

Permissionless blockchain networks often provide a reward (e.g., a cryptocurrency) to motivate users to act fairly; however, some may choose to act maliciously if that provides greater rewards.

The largest problem for malicious users is getting enough power (be it a stake in the system, processing power, etc.) to cause damage.

Once a large enough malicious collusion is created, malicious mining actions can include:

- Ignoring transactions from specific users, nodes, or even entire countries.
- Creating an altered, alternative chain in secret, then submitting it once the alternative chain is longer than the real chain. The honest nodes will switch to the chain that has the most “work” done (per the blockchain protocol). This could attack the principle of a blockchain network being tamper evident and tamper resistant
- Refusing to transmit blocks to other nodes, essentially disrupting the distribution of information (this is not an issue if the blockchain network is sufficiently decentralized).

- While malicious users can be annoyances and create short-term harm, blockchain networks can perform hard forks to combat them.
- In addition to there being malicious users of the network, the administrators of the infrastructure for permissioned blockchain networks may also act maliciously.
- For example, an infrastructure administrator may be able (depending upon the exact configuration) to take over block production, exclude certain users from performing transactions, rewrite block history, double spend coin, delete resources, or re-route or block network connections.

## No Trust

- Another common misinterpretation comes from people hearing that there is no “trusted third party” in a blockchain and assuming blockchain networks are “trustless” environments.
- While there is no trusted third party certifying transactions in permissionless blockchain networks (in permissioned systems it is less clear, as administrators of those systems act as an administrator of trust by granting users admission and permissions), there is still a great deal of trust needed to work within a blockchain network:

- There is trust in the cryptographic technologies utilized. For example, cryptographic algorithms or implementations can have flaws.
- There is trust in the correct and bug free operation of smart contracts, which might have unintended loopholes and flaws.
- There is trust in the developers of the software to produce software that is as bug-free as possible.
- There is trust that most users of the blockchain are not colluding in secret. If a single group or individual can control more than 50 percent of all block creation power, it is possible to subvert a permissionless blockchain network. However, generally obtaining the necessary computational power is prohibitively expensive.
- For blockchain network users not running a full node, there is trust that nodes are accepting and processing transactions fairly.

### **Resource usage:**

For blockchain networks utilizing proof of work, there are many publishing nodes expending large amounts of processing time and, more importantly, consuming a lot of electricity.

The proof of work consensus model is designed for the case where there is little to no trust amongst users of the system. It ensures that publishing nodes cannot game the system.

A major concern surrounding the proof of work consensus model is its use of energy in solving the puzzles.

The amount of energy used is often not trivial; for example, some estimate that currently the Bitcoin blockchain network uses around the same amount of electricity as the entire country of Ireland.

It has also been speculated that the Bitcoin blockchain network will consume as much electricity as the entire country of Denmark by 2020.

Software and hardware will continue to improve, resulting in more efficient puzzle solving (reducing the amount of electricity utilized). However, blockchain networks are also still growing, resulting in harder puzzle difficulty.

An additional strain on resources occurs whenever a new full node is created; the node must obtain (usually through downloading) most of or all the blockchain data (Bitcoin's blockchain data is over 175 gigabytes and growing as of this writing). This process uses a lot of network bandwidth.

### **Inadequate Block Publishing Rewards**

A potential limitation is the risk of inadequate rewards for publishing a block.

The combination of increased competition, increased computational resources needed to have meaningful contributions to pools of publishing nodes, and highly volatile market prices in the cryptocurrency market creates the risk that the expected return for any given cryptocurrency may be less than the power costs needed to run publishing node software.

Cryptocurrencies that are not able to consistently and adequately reward publishing nodes risk delays in publishing blocks and processing transactions.

These delays could therefore reduce confidence in the cryptocurrency, reducing its market value further.

It could then become increasingly less attractive for publishing nodes to contribute to that cryptocurrency's publishing efforts.

Even worse, such weakened cryptocurrencies open themselves up to being attacked by nodes with large amounts of resources that may maliciously alter the blockchain or deny service to users attempting to submit transactions.

## Application Considerations

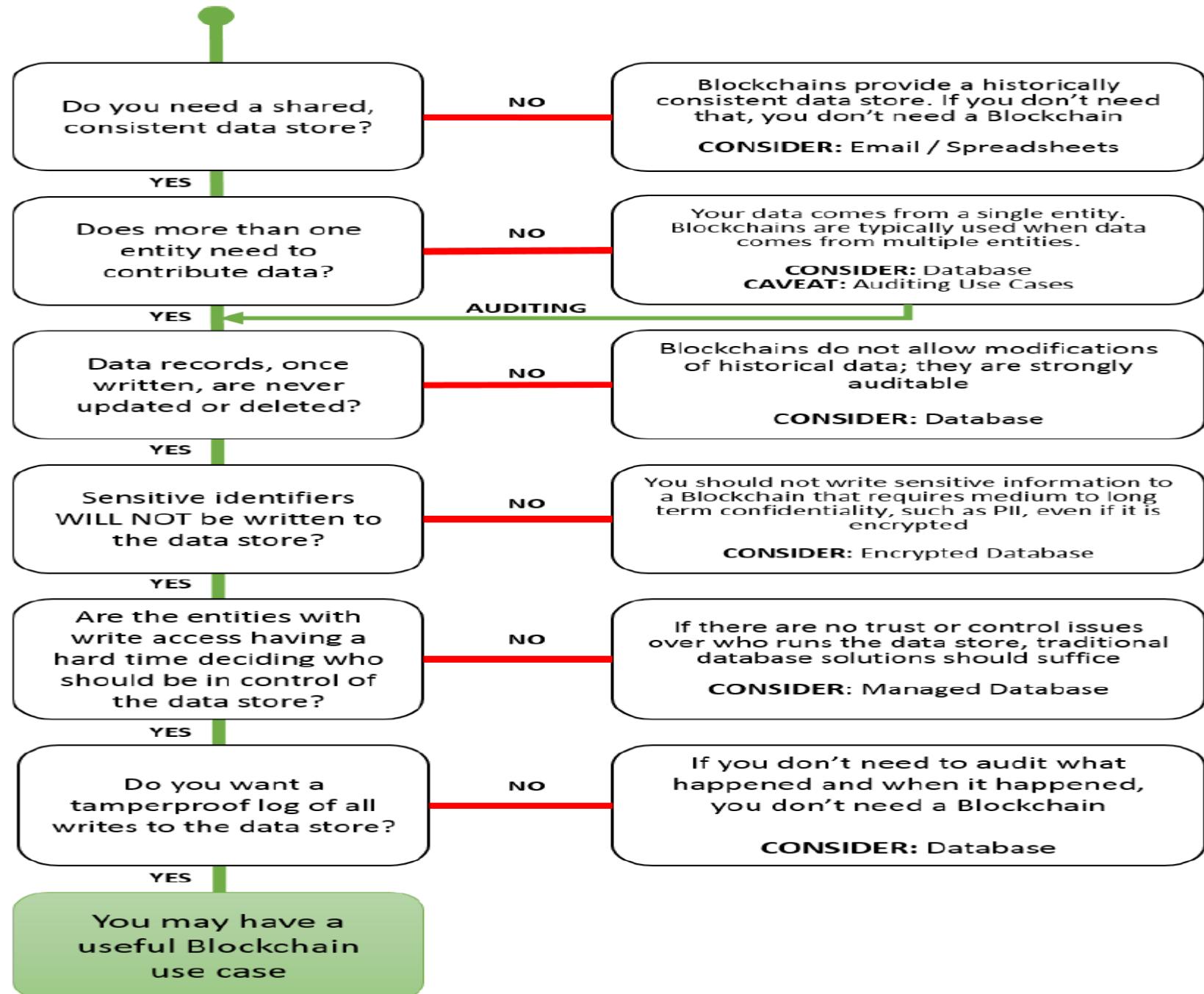
- Blockchain technology is still new, a lot of organizations are looking at ways to incorporate it into their businesses.
- The fear of missing out on this technology is quite high, and most organizations approach the problem as “we want to use blockchain somewhere, where can we do that?” which leads to frustrations with the technology as it cannot be applied universally.
- A better approach would be to first understand blockchain technology, where it fits, and then identify systems (new and old) that may fit the blockchain paradigm.

Blockchain technology solutions may be suitable if the activities or systems require features such as:

- Many participants
- Distributed participants
- Want or need for lack of trusted third party
- Workflow is transactional in nature (e.g., transfer of digital assets/information between parties)
- A need for a globally scarce digital identifier (i.e., digital art, digital land, digital property)

A need for a decentralized naming service or ordered registry

- A need for a cryptographically secure system of ownership
- A need to reduce or eliminate manual efforts of reconciliation and dispute resolutions
- A need to enable real time monitoring of activity between regulators and regulated entities
- A need for full provenance of digital assets and a full transactional history to be shared amongst participants
- There are articles and advice from several different sectors – federal government, academia, technical publications, technology websites, and software developers to help determine if a blockchain is suitable for a particular system or activity, and which kind of blockchain technology would be of most benefit.
- United States Department of Homeland Security (DHS) Science & Technology Directorate has been investigating blockchain technology and has created a flowchart to help one determine whether a blockchain may be needed for a development initiative



## Additional Blockchain Considerations

- When deciding whether to utilize a block chain, one must take into consideration additional factors and determine if these factors limit one's ability to use a block chain or a particular type of block chain

### Data Visibility

- Permission block chain
- Permission less Block Chain

**Full transactional history :** Some blockchain networks provide a full public history of a digital asset – from creation, to every transaction it is included in. This feature may be beneficial for some solutions, and not beneficial for others.

**Fake Data Input** – Since multiple users are contributing to a blockchain, some could submit false data, mimicking data from valid sources (such as sensor data). It is difficult to automate the verification of data that enters a blockchain network. Smart contract implementations may provide additional checks to help validate data where possible.

**Tamper evident and tamper resistant data** – Many applications follow the “CRUD” (create, read, update, delete) functions for data. With a blockchain, there is only “CR” (create, read). There are methods that can be employed to “deprecate” older data if a newer version is found, but there is no removal process for the original data. By using new transactions to amend and update previous transactions, data can be updated while providing a full history.

## **Transactions Per Second :**

Transaction processing speed is highly dependent on the consensus model used.

Currently transactions on many permissionless blockchain networks are not executed at the same pace as other information technology solutions due to a slow publication time for blocks (usually in terms of seconds, but sometimes minutes).

Thus, some slowdown in blockchain dependent applications may occur while waiting for data to be posted. One must ask if their application can handle relatively slow transaction processing?

**Compliance** – There are many compliance considerations with regards to legislation and policies when regarded to block chain.

- In some cases, Certain countries may limit the type of data that can be transferred across its geographic boundary.
- In other instances, certain legislation may dictate that the “first write” of financial transactions must be written to a node which is present within their borders.
- In any of these cases, a public, permissionless chain may be less appropriate, with a permissioned or hybrid approach required to satisfy regulatory needs.
- An additional example of laws and regulations are for any blockchain network which manages federal records. Federal records are subject to many laws and regulations.
- Federal agencies themselves must follow specific federal guidelines when utilizing blockchain technology

**Permissions** –there are concerns around the permissions themselves

does the permissions within the system is good enough for specific roles that users may need to perform actions within the system .

in case of permissioned block chain - who can administer permissions? Once permissions are administered to a user, can they easily be revoked etc.

**Node Diversity** – A blockchain network is only as strong as the aggregate of all the existing nodes participating in the network.

If all the nodes share similar hardware, software, geographic location, and messaging schema then there exists a certain amount of risk associated with the possibility of undiscovered security vulnerabilities.

This risk is mitigated through the decentralization of the network of heterogeneous devices, which may be defined as “the non-shared characteristics between any one node and the generalized set”