

The background is a dark, abstract digital space filled with numerous glowing, multi-colored lines (red, blue, green, yellow) that appear to be data streams or network connections. In the center, there is a large, three-dimensional wireframe cube. The edges of this cube are highlighted with a bright orange-yellow glow, and small clusters of similar glowing particles are scattered around it, particularly near the top and bottom vertices.

# BCDV 1001

# Intro to Blockchain

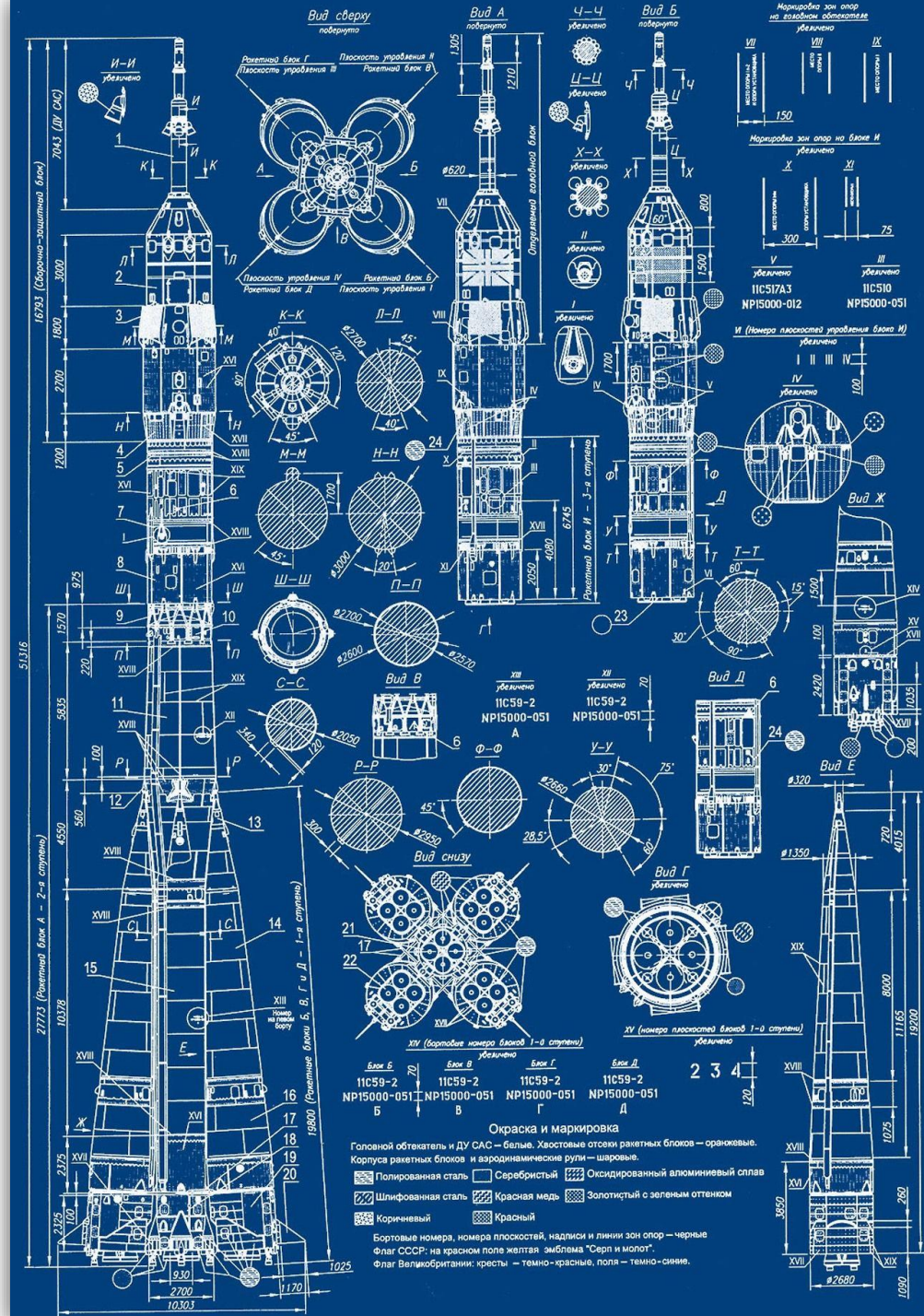
[Week 02 - Lesson 02]

Prof: Djordje (George) Petrovic  
[djordje.petrovic@georgebrown.ca](mailto:djordje.petrovic@georgebrown.ca)



# Overview

- Lab Test 02
- Lab Exercise 02
- Recap of previous class
- Closer look at block creation – mining PoW
- Ethereum
- Ether (ETH)
- ERC tokens
- Smart Contract

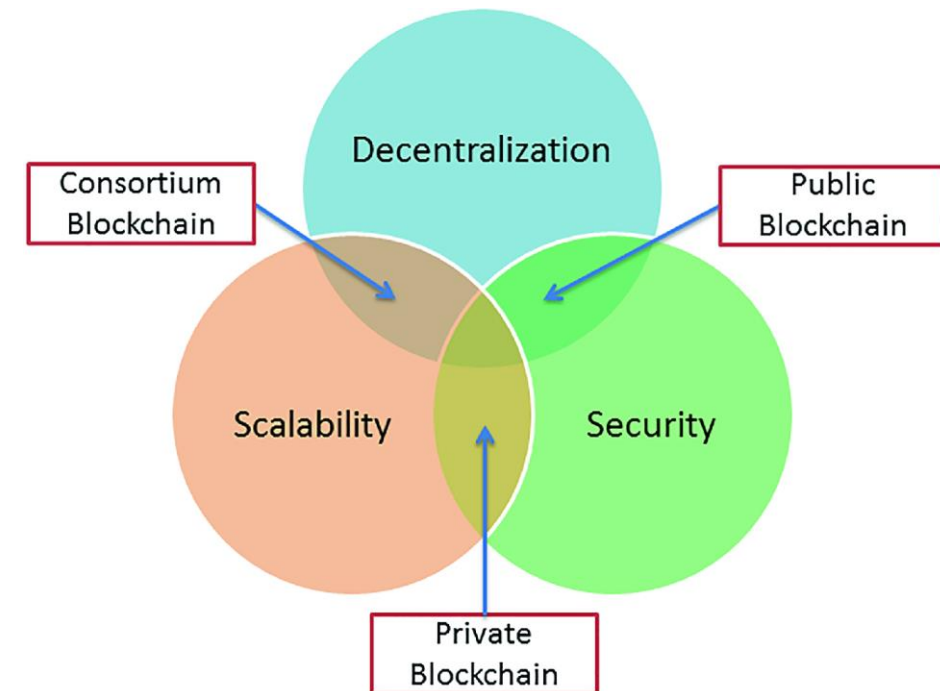


# Assignment Update

- **Assignment Mark** = Presentation (/w deck) + Written document;
- Presentation – **no more than 10 slides**, it's a pitch deck;
- Written document – no more than **~5-7 pages**;
- Labs, Assignment, Presentation and deck is group work - One submission and same mark for all.
- Test and Final exam is individual work – individual marks

# Recap

- **Public vs Private** (permission-less vs permissioned – *who is able to write the data on the blockchain*)
- **Open vs Closed** (*who is able to read the data*)
- Blockchain Trilemma
- Blockchain issue – **scalability**
- Consensus Mechanism
- Use Case – **business network players**



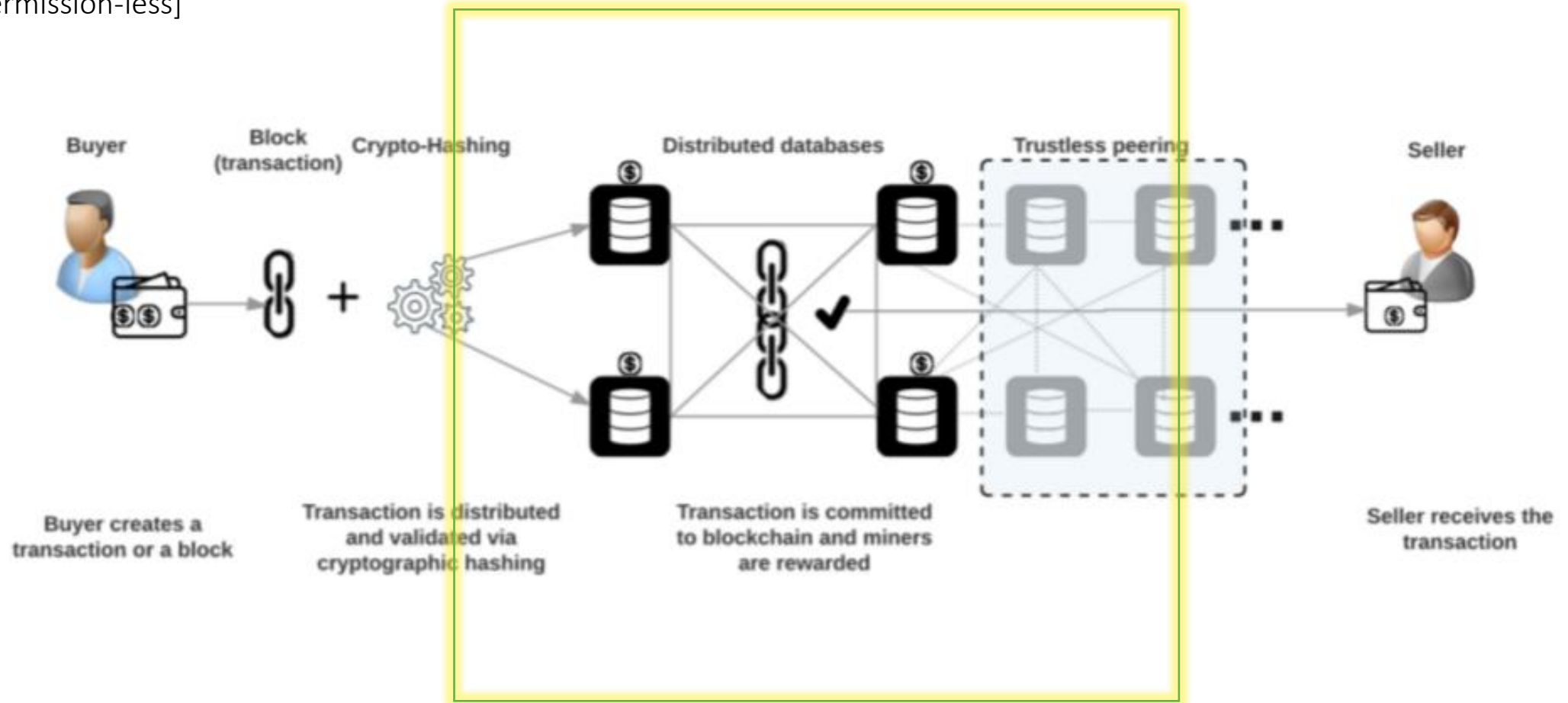
# Public Blockchain

[permission-less]

- **Anyone is allowed to participate** in the network
- Anyone can indulge in the network as a node and can conduct transactions.  
**Ledger are not maintained by anyone** and is completely opened to anyone.
- **Network is completely decentralized**, user can download the software and track the record.
- **Data is shared among participants**, and anyone can see and participate in the ongoing transactions.
- However, the **distributed consensus mechanism is being maintained** in order to reach the final decision.
- Transactions are carried **without the need of a 3<sup>rd</sup> party**, therefore eliminates the third party;
- **Each transaction need to be verified by thousands of participants**, therefore the verification process becomes **time-consuming**.

# Public Blockchain

[permission-less]





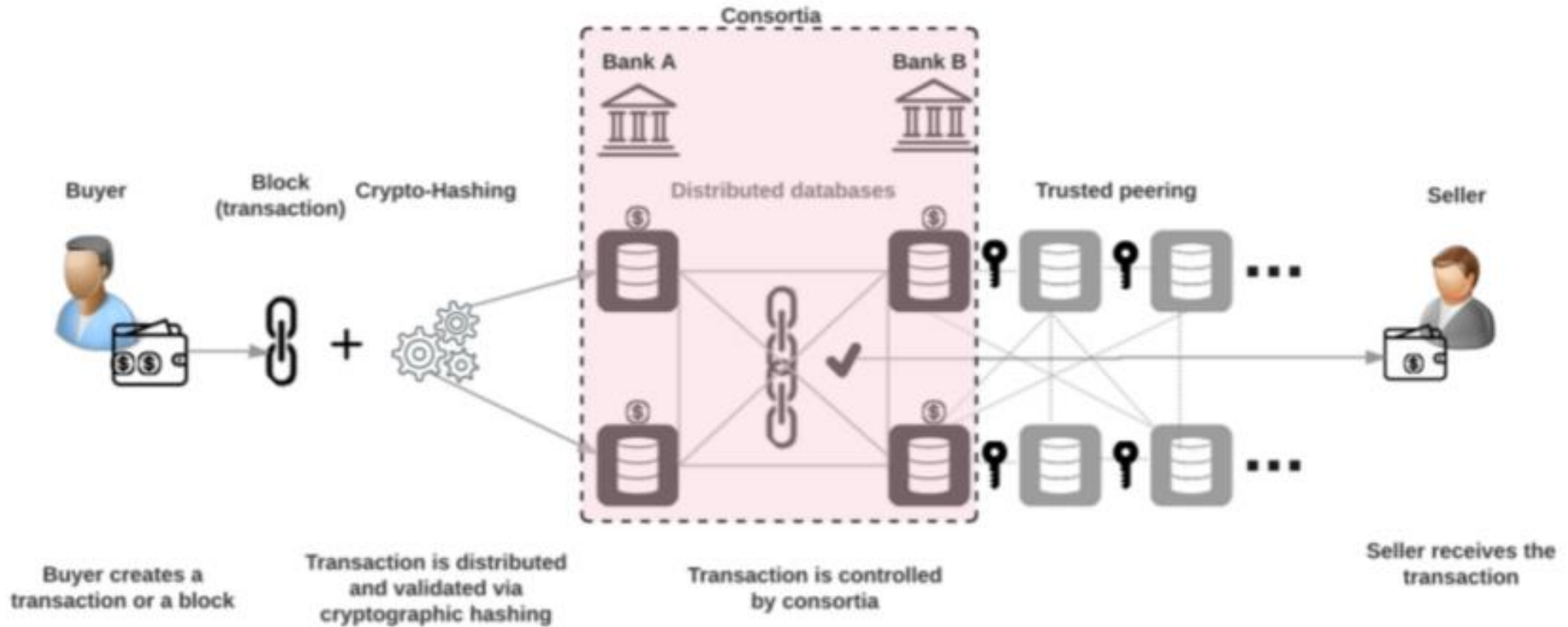
# Private Blockchain

[permissioned]

- **Opened to a group of 'entities'** or consortium that has **agreed to share the ledger** with each other.
- Mutually agreed to **trust the validators** in the network (trusted and certified).
- Organization that runs the blockchain is given the authority to allow participants to **read a particular transaction** as all transaction are not necessary for participants to be read.
- Private network gives **more privacy** to the members.

# Private Blockchain

[permissioned]





# Consortium Blockchain

[Hybrid Model]

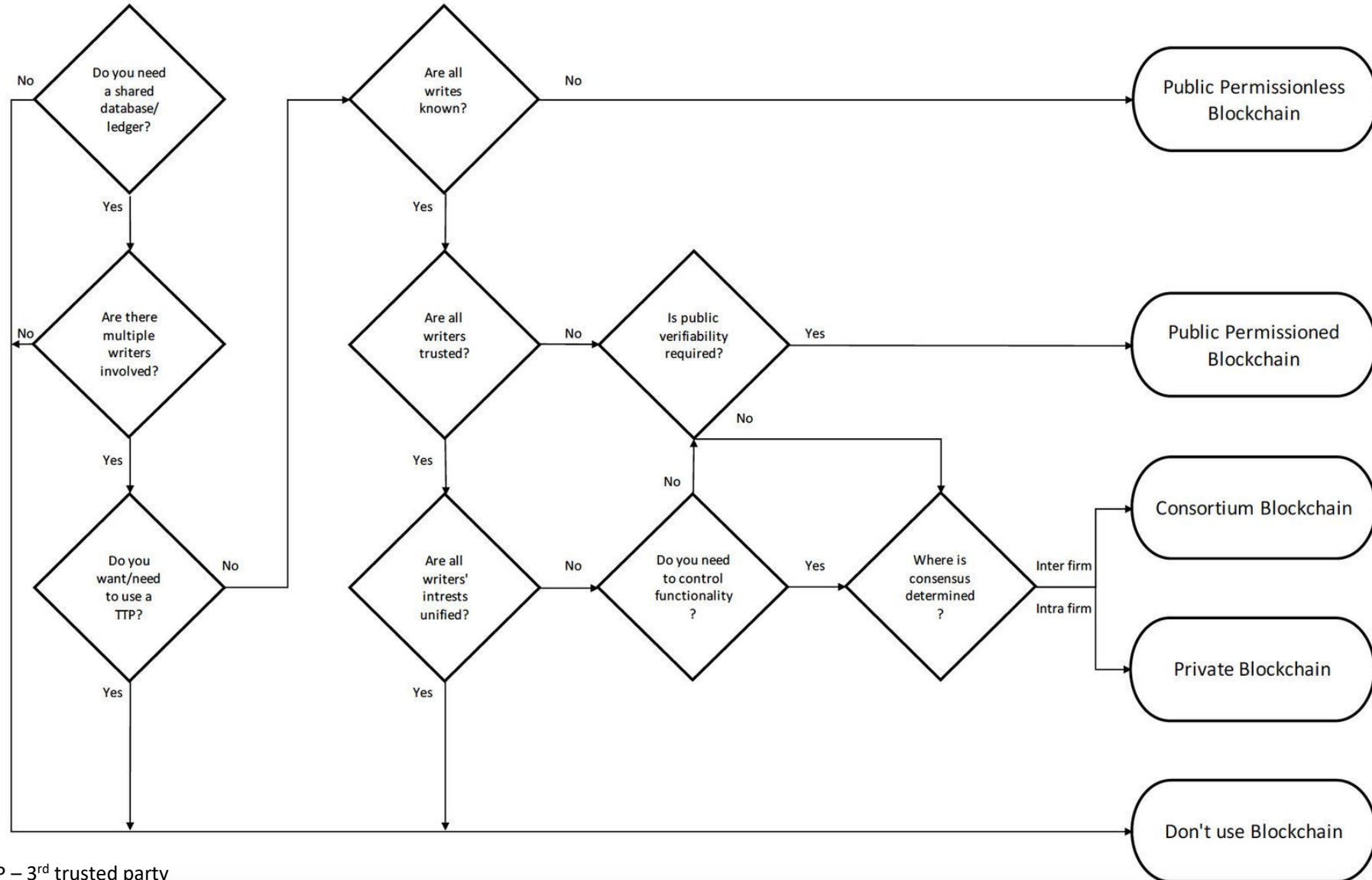
	PUBLIC Blockchain	PRIVATE Blockchain	CONSORTIUM/ENTERPRISE/ FEDERATED/HYBRID Blockchain
ACCESS	<ul style="list-style-type: none"><li>• <b>Anyone</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Single</b> Organization</li></ul>	<ul style="list-style-type: none"><li>• <b>Multiple</b> selected organizations</li></ul>
PARTICIPANTS	<ul style="list-style-type: none"><li>• Permissionless</li><li>• Anonymous</li></ul>	<ul style="list-style-type: none"><li>• Permissioned</li><li>• Known Identities</li></ul>	<ul style="list-style-type: none"><li>• Permissioned</li><li>• Known identities</li></ul>
SECURITY	<ul style="list-style-type: none"><li>• Consensus mechanism</li><li>• Proof of Work/</li><li>• Proof of Stake</li></ul>	<ul style="list-style-type: none"><li>• Pre-approved participants</li><li>• Voting/multi-party consensus</li></ul>	<ul style="list-style-type: none"><li>• Pre-approved participants</li><li>• Voting/multi-party consensus</li></ul>
TRANSACTION SPEED	<ul style="list-style-type: none"><li>• Slow</li></ul>	<ul style="list-style-type: none"><li>• Lighter and faster</li></ul>	<ul style="list-style-type: none"><li>• Lighter and faster</li></ul>

# Private vs Public – Final thoughts

- Public has **miners**; Private has **validators**;
- Public miners are rewarded, validators maybe;
- Private users are known (**KYC**), while in public users are **anonymous** – question of identity and understanding who the user is and to what information should they have access to;
- **Consensus Mechanism** in public is energy consuming (expensive);
- Private is much **quicker and secure** as only limited by number of people are taking part in the network.
- *There's a perception that public blockchain platforms like Ethereum can't be used to build permission scenarios or to control access to data.  
The truth is that they can, they just don't give you all the built-in tools that you can find on a private or permission blockchain platform. But basically, you can always use these open public platforms to build a permission solution, you just need to be aware that it's upon you, your architects and your developers to create that permissioning model, and that all starts with some kind of identity management system.*
- *A lot of people get the impression that they compete with one another, but they really don't. They just serve to provide different types of solutions.*

# Private vs Public – Final thoughts









[DIAGRAM]



\*TTP – 3<sup>rd</sup> trusted party

© Exterkate & Wagenaar 2018

# Blockchain Components

Ledger		contains the current world state of the ledger and a Blockchain of transaction invocations
Smart Contract		encapsulates business network transactions in code. transaction invocations result in gets and sets of ledger state
Consensus Network		a collection of network data and processing peers forming a Blockchain network. Responsible for maintaining a consistently replicated ledger
Membership		manages identity and transaction certificates, as well as other aspects of permissioned access
Events		creates notifications of significant operations on the Blockchain (e.g. a new block), as well as notifications related to smart contracts. Does not include event distribution.
Systems Management		provides the ability to create, change and monitor Blockchain components
Wallet		securely manages a user's security credentials
Systems Integration		responsible for integrating Blockchain bi-directionally with external systems. Not part of Blockchain, but used with it.



# Blockchain Components

[MORE]

COMPONENT	DETAILS
LEDGER	<ul style="list-style-type: none"><li>• Contains all the record of the transaction and participants.</li><li>• 2 types of ledgers (public and private).</li></ul>
SMART CONTRACT	<ul style="list-style-type: none"><li>• Like a traditional document that defines rules and regulations (events) but it automatically enforces those obligations. (i.e. vending machine)</li><li>• Reduces fraud and illegal activities.</li></ul>
CONSENSUS MECHANISAM	<ul style="list-style-type: none"><li>• Consensus refers to aggregate decision or agreement between different people over a group of rules and law which run a blockchain.</li></ul>
MEMBERSHIP	<ul style="list-style-type: none"><li>• All participants on the blockchain network are given permission to use the it by assigning a unique identity.</li></ul>
EVENTS	<ul style="list-style-type: none"><li>• Useful as it gives notifications of all the important operations on the blockchain network.</li></ul>
SYSTEM MANAGEMENT	<ul style="list-style-type: none"><li>• A system management or manger define, create modify or remove the rules (broad responsibility).</li><li>• Give authority to make sure that error-free operations happen on the blockchain.</li></ul>
WALLET	<ul style="list-style-type: none"><li>• Each member of the blockchain is being given a wallet which securely maintains all records of the user.</li><li>• Types of wallets: hardware, paper, desktop, mobile, and web.</li></ul>
SYSTEM INTEGRATION	<ul style="list-style-type: none"><li>• Oracle, AI, IoT, BaaS, etc.</li></ul>



# Additional Material

- Do you need blockchain? (paper) – [[LINK](#)]
- BBC Panorama – Just how do cryptocurrencies work (public, PoW) (3min) [[LINK](#)]
- BBC Newsnight - How does bitcoin mining work? (7min) [[LINK](#)]
- Proof-of-Work vs Proof-of-Stake vs Proof-of-...
  - Consensus: Proof-of-.... [[LINK](#)]
  - PoW vs PoS by Vitalik [[LINK](#)]
  - Understanding Consensus Mechanisms [[LINK](#)]
  - Practical Byzantine Fault Tolerance (PBFT) used by Hyperledger (will cover later)

# Reminder - things to know (research)

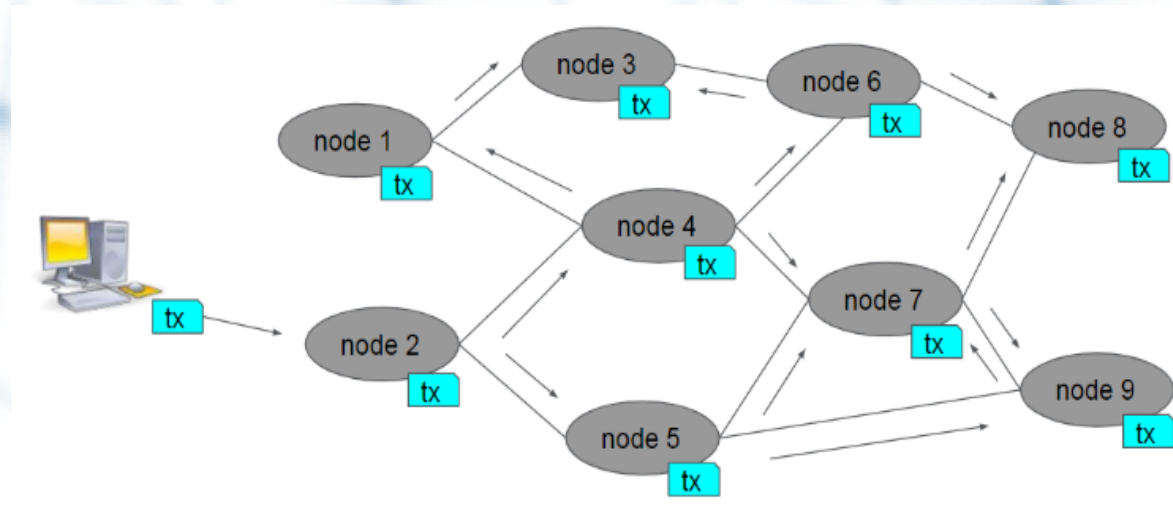
- Merkel Root
- Difficulty target/level
- Hash (SHA-256)
- Hashing Power
- Mem-pool
- Blockchain Header
- Mining
- nonce



# Block Creation – Transaction Network Propagation

[PoW - the 10 minute heart beat]

- George send a transaction/funds to Ana.
- The node checks if it is valid, and propagates (send) to other nodes if it is – and so on and so forth until all nodes receive the transaction.



# Block Creation – From Transaction to Blocks

[PoW - the 10 minute heart beat]

- From a bitcoins node perspective, the node receives a transaction (TX) which goes into a memory pool (**mempool**). *Note: not all mempools are same, each node has a different mempool.* Node 1 can have 10 transactions and at the same time Node 2 will have 15 transactions. *Nodes are miners in this case (miners are computers on the network, computers are owned by individuals, groups of people, or companies).*
- It keeps receiving transactions from all other users on the network...
- At some point it will decide if it will group these TXs into a block.
- It will add the **coinbase** TX which rewards 6.25 BTC to a node's BTC address.
- Finally, it adds the block's header containing important information.



# Block Creation – Mining

[PoW - the 10 minute heart beat]

## STEP 01

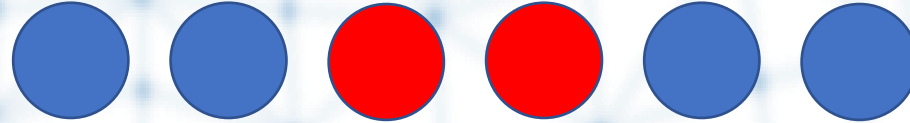
- After a node (miner) created a block it will attempt to make it final by propagating it to all other nodes in the network. Multiple nodes will receive the same transaction and will create blocks; nodes choose which TXs to include. They can create and propagate a block at any time.
- A very **difficult computational problem** needs to be solved in order to accept a block as valid. The process of finding the solution requires work (**Proof-of-Work >> PoW**) and is called mining.

## STEP 02

- The PoW puzzle is to compute a cryptographic hash of the new block that we want to create which should be **less than a given number**. Since a hash is random it will take several attempts to find a proper has but other nodes will verify with only one attempt.

# Block Creation – Mining

[PoW - the 10 minute heart beat]



## STEP 03

- Puzzle's difficulty automatically adjusts so that it requires approximately 10 minutes to solve. As more miners join the network the block will be created faster. The difficulty of the puzzle is then increased to require ~10 minutes again. (Adjustment happens every 2016 blocks, approximately 2 weeks, if each block takes 10 minutes to mine).

## STEP 04

- Additionally, the transactions fees of all the TXs in a block are also awarded to the miner that creates the new block.
- The header of a block contains, previously created block. A block always contains a coinbase tx transaction which is used to pay the mining reward to the miner. The mining reward is available to the miner after 100 confirmations.



# INSIDE BITCOIN'S BLOCKCHAIN

## HEADER

The block header is hashed twice to create the fingerprint which is referred to in the next block.

<b>Technical data</b> Includes a Magic ID, a version number (to specify which set of protocol rules this block conforms to), the size of this block.	<b>Previous block hash</b> 2x SHA256 hash of previous block header (excluding magic ID & block size). This is the link that creates the chain of blocks.
<b>Merkle Root</b> Distills all the transactions in the block into a single hash.	<b>Timestamp</b> Approximate timestamp of when the block was created. Used to figure out mining difficulty re-targets i.e if the network is making blocks too quickly or too slowly.
<b>Difficulty target</b> Related to mining and how hard it is to successfully mine the block	<b>Nonce</b> A random number. One of the things you can change when mining to create different hashes, while searching for a suitable hash.

## TECHNICAL DATA

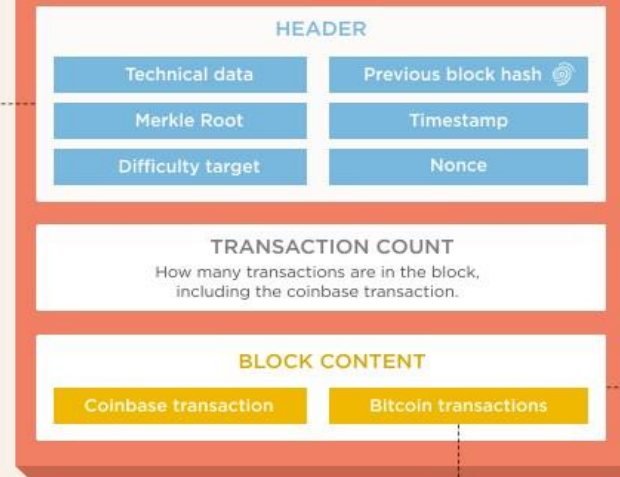
<b>Version number</b> Can be used for specifying which set of protocol rules this transaction confirms to.	<b>Transaction lock time</b> Something which may be used in future for "future dating" a transaction, like writing a post-dated cheque.
<b>Input count</b> How many inputs are in this transaction.	<b>Output count</b> How many outputs does this transaction create.

## INPUT

<b>(Technical) Input script length</b> How much data is in the input.	<b>(Technical) Sequence number</b> Not really used.
<b>Previous transaction hash &amp; index</b> This identifies where the coins are coming from, by specifying an output from a previous transaction.	<b>Script data</b> This is where you "prove" you own the coins and you are allowed to spend it, by signing with the private key of the address that the bitcoins are in.

## BLOCK

Blocks are the units of the blockchain, like pages of transactions in a ledger.



## TRANSACTION

Each transaction is a bitcoin payment



## INPUT

<b>(Technical) Input script length</b>
<b>(Technical) Sequence number</b>
<b>Previous transaction hash &amp; index</b>
<b>Script data</b>

## OUTPUT

<b>(Technical) Output script length</b>
<b>Amount</b>
<b>Output script</b>



www.bitsonblocks.net



Designed by www.fractalphia.com

## BLOCK CONTENT

<b>Coinbase transaction</b> The bit where you get to pay yourself the mining reward (currently 25 BTC) plus the fees from the transactions included in the block. It's a special transaction where there are no 'inputs' or 'from' addresses.	<b>Bitcoin transactions</b> This is the main payload of the block. Contains bitcoin payments. Transaction Transaction Transaction Transaction
---	--

## FOLLOWING THE MONEY

### Bank accounts vs cryptocurrencies

Bank accounts mix money up. When you pay someone, you don't specify "use those pounds which I earned from my salary" or "use those pounds which I received for my birthday". Money is treated equally once it hits your account, and is untraceable.

On the other hand, with cryptocurrencies, you need to specify exactly which incoming deposits you are spending. This makes every transaction traceable, right back to the creation of the coins.

### Inputs and Outputs

Every bitcoin transaction references some incoming deposits as inputs, and spends them entirely as new outputs, with change returned to one of your addresses.

This is like paying £43.50 by taking three £20 banknotes from your wallet and creating two new banknotes: £43.50 and £16.50. You hand over the £43.50 banknote and keep the £16.50 banknote. You can then spend the £16.50 later in one go. The other person can spend the £43.50 later in one go.

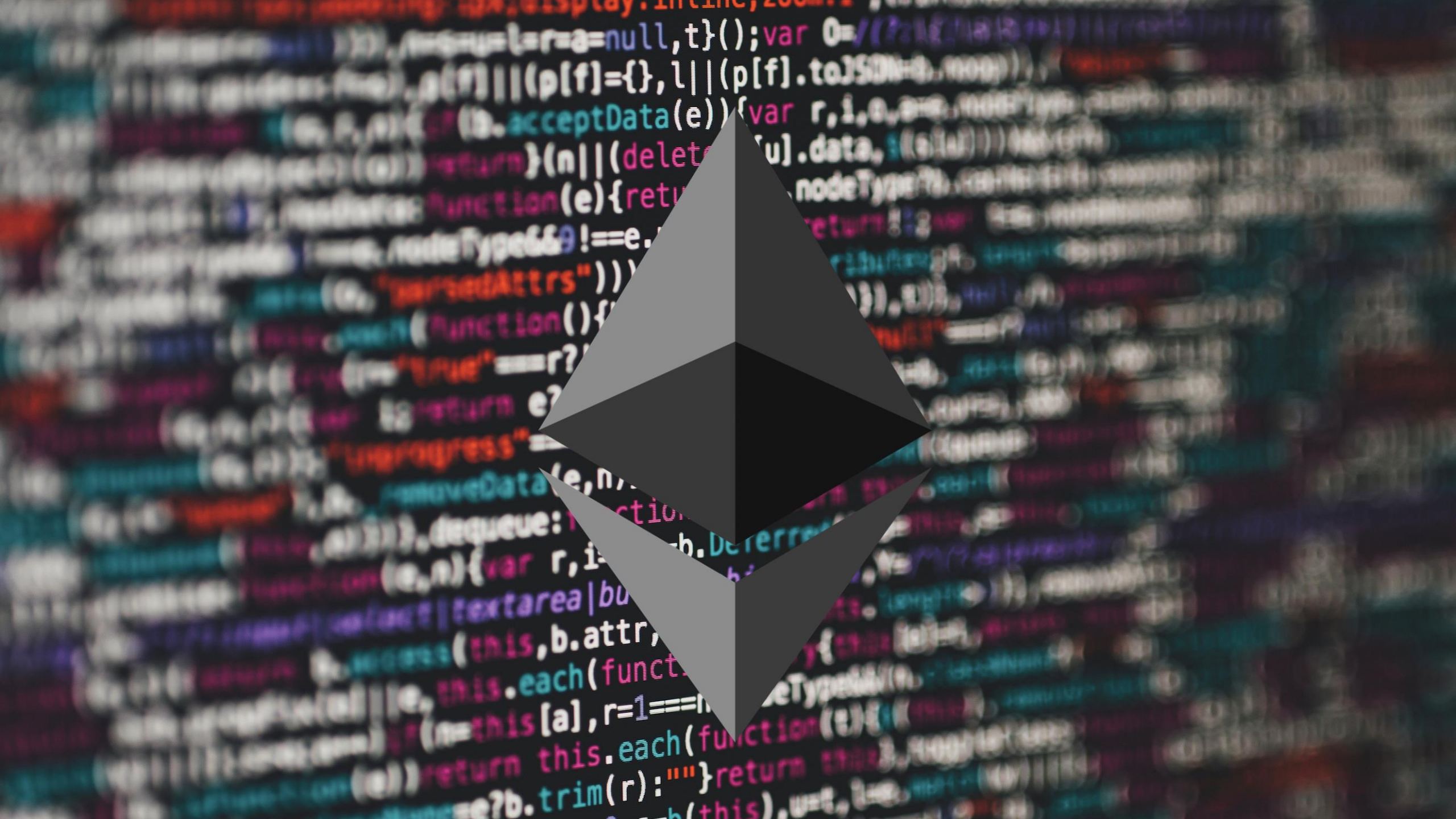
**Inputs:** 3 x £20

**Outputs:** £43.50 (payment), £16.50 (change)

## OUTPUT

<b>(Technical) Output script length</b> How much data is in this output	<b>Amount</b> How many bitcoins (actually, Satoshis) are being sent.
<b>Output script</b> Who (which address/es) are the bitcoins being sent to? Which signatures are needed to re-spend these coins?	





# Ethereum

[Blockchain 2.0]

- What is Ethereum
- Features of Ethereum
- Smart Contracts
- Decentralized Autonomous Organizations (DAO)



# What is Ethereum



- Ethereum is an **open-source, public, blockchain-based distributed computing platform** featuring smart contract functionality. “A planetary scale computer built on blockchain technology”
- It is **not limited to crypto-currencies**; a programmable blockchain using **smart contracts**.
- A **decentralized platform that runs smart contracts**: application that run exactly as programmed **without any possibility of downtime, censorship, fraud, or third party interference.**”
- Ethereum Virtual Machine (**EVM**) – runs on Ether, pay per computation step (concept of **gas**).
- Turing-complete - any system or programming language able to compute anything computable given enough resources.



# What is Ethereum



- How is it different from bitcoin

	Bitcoin	Ethereum
<b>Concept/Usage</b>	Digital Money – be a global decentralized payment system	Smart Contract – be a decentralized super computer to power dApps
<b>Scripting Language</b>	Turing incomplete	Turing complete
<b>Release date</b>	Jan 2009	July 2015
<b>Coin Release Method</b>	Early Mining (Genesis Block Mined)	Through ICO (Presale)
<b>Average Block time</b>	~10 minutes (6/hr)	~12-15 seconds (215.83/hr)
<b>Throughput</b>	7-8 tps	15-20 tps
<b>Consensus</b>	PoW	PoW (PoS)
<b>Block Reward</b>	12.5 BTC	2 ETH

# Ethereum Features



Key factors making Ethereum popular and effective:

- **Crypto-currency:** blockchain implementation and therefore requires its internal crypto called Ether (ETH);
- **Smart contracts:** application that runs on blockchain and process all the information on the blockchain;
- **Decentralized Autonomous Organizations:** decentralized organization that does not require governing;
- **Smart Property:** digitize your property and transform without the hustles for exchanging the property or validating the documents

# Ether



- **Token** of the Ethereum blockchain, listed as **ETH** on exchanges;
- **Used to pay for transaction fees and computational services** on the Ethereum network;
- Every time a contract is executed, **Ethereum consumes tokens** (termed as **gas**) to run the computations;
- Providing too little gas will result in failed transaction;
- ETH GAS STATION, [[link](#)]

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000



# ERC standard tokens



## DIFFERENT ETHEREUM REQUEST FOR COMMENTS

### ERC 20

- Most popular token standard.
- Used in most of the ICOs.
- Fungible token standard.
- Allows the implementation of standard API within a smart contract.

### ERC 721

- A standard for non-fungible tokens.
- Allows the implementation of standard API for NFTs within a smart contract.
- Offers functionality to transfer and track NFTs.

### ERC 165

- A standard for a method, instead of tokens.
- Covers how interfaces are identified.
- States how any contract can publish the interfaces after the implementation.
- States how to detect when a contract implements ERC-165.
- Covers the way to detect when a smart contract uses any given interface.

### ERC 777

- Reduces friction in crypto transactions.
- Not in use, still in EIP phase.
- Gets rid of the double transaction verification of ERC 20.
- Lowers transaction overhead.
- Allows users to reject incoming tokens from a blacklisted address.

### ERC 223

- Prevents accidental burns of tokens, a bug in ERC 20.
- Developers can either accept or decline tokens arriving at their smart contract addresses.
- Rejected transactions will fail but won't burn the tokens.
- Not in use, still in EIP phase.

### ERC 827

- An extension of ERC 20.
- Wallets and exchanges can reuse tokens.
- Token holders can transfer token while also approving a 3rd party to spend it.
- Not in use, still in EIP phase.

### ERC 621

- An extension to ERC 20 standard
- Uses two functions - 'increaseSupply' and 'decreaseSupply'.
- Can increase or decrease the token supply.
- Not in use, still in EIP phase.

### ERC 884

- Allows companies to use blockchain to maintain share registries.
- Identity verification and mandatory whitelisting of token holders.
- Only whole value of tokens, i.e., no partial value.
- Recording of information regulators mandate.
- Not in use, still in EIP phase.

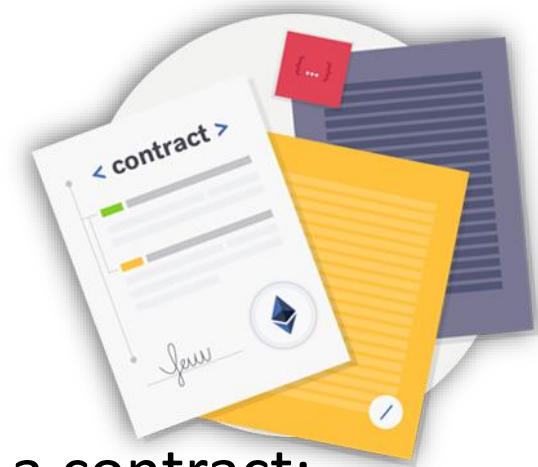
# ERC tokens



- **ERC 721** - is a non-fungible token standard that enables the development of unique tokens from the same contract address. The ERC 721 standard is popularly known as the standard for creating NFTs (non-fungible tokens) or 'nugibles'.
- **ERC 1410** - These tokens correspond to the requirement of securities to contain differing metadata yet allow a level of fungibility. To this effect, the ERC 1410 token standard enables partitioning of the token into a fungible and a non-fungible component.
- **ERC 1643** - Traditional securities are associated with several documents that relate to ownership, rights, and obligations. The ERC 1643 standard takes care of this requirement by enabling a mechanism of simplified transfer of documents for a speedy transfer of legal rights and obligations associated with the ownership of that security token.



# Smart Contracts









- Idea from 1990s by Nick Szabo;
- Computerized transaction protocol that executes the terms of a contract;
- Most popular scripting language – Solidity, Serpent;
- Tools for writing & deploying smart contracts:
  - ✓ **Mist Browser** – It is a tool to browse and use dApps. It is a separate browser that can be used to browse dApps and interact with them.
  - ✓ **Truffle Framework** – Truffle is a popular development framework for Ethereum. It has built-in smart contract compilation, linking, deployment, and binary management.
  - ✓ **Metamask – MetaMask** is a bridge that allows one to visit the distributed web of tomorrow in their browser today. It allows users to run Ethereum dApps right in their browser without running a full Ethereum node.
  - ✓ **Remix Browser** – Remix is a web browser based IDE that allows users to write Solidity smart contracts, then deploy and run the smart contract.

# Smart Contracts



## Traditional contracts

## Smart contracts

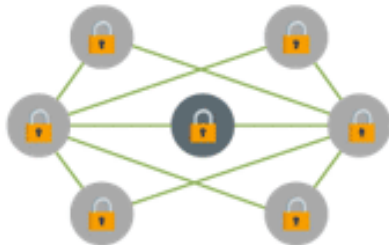
 1-3 Days	Minutes 
 Manual remittance	Automatic remittance 
 Escrow necessary	Escrow may not be necessary 
 Expensive	Fraction of the cost 
 Physical presence (wet signature)	Virtual presence (digital signature) 
 Lawyers necessary	Lawyers may not be necessary 

# Smart Contracts

## Physical Contracts



Blockchain/permissioned ledger,  
programming & encryption

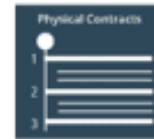


Transacting parties  
Individuals or Institutions



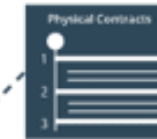
## Smart Contracts

Lower operational  
Overheads & costs leading  
To economical financial  
products



### Smart Contracts

A Software program  
on the distributed  
Ledger, allowing an  
immutable & Verifiable  
records of all Contracts &  
Transactions

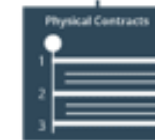


Banks, Insurers,  
Capital Markets

Act as custodians of assets,  
validators & authorities of all  
contracts & transactions

Faster, simpler &  
hassle-free processes,  
Reduced settlement times

Reduced administration  
& service costs Owing  
to automation & ease  
of compliance & reporting



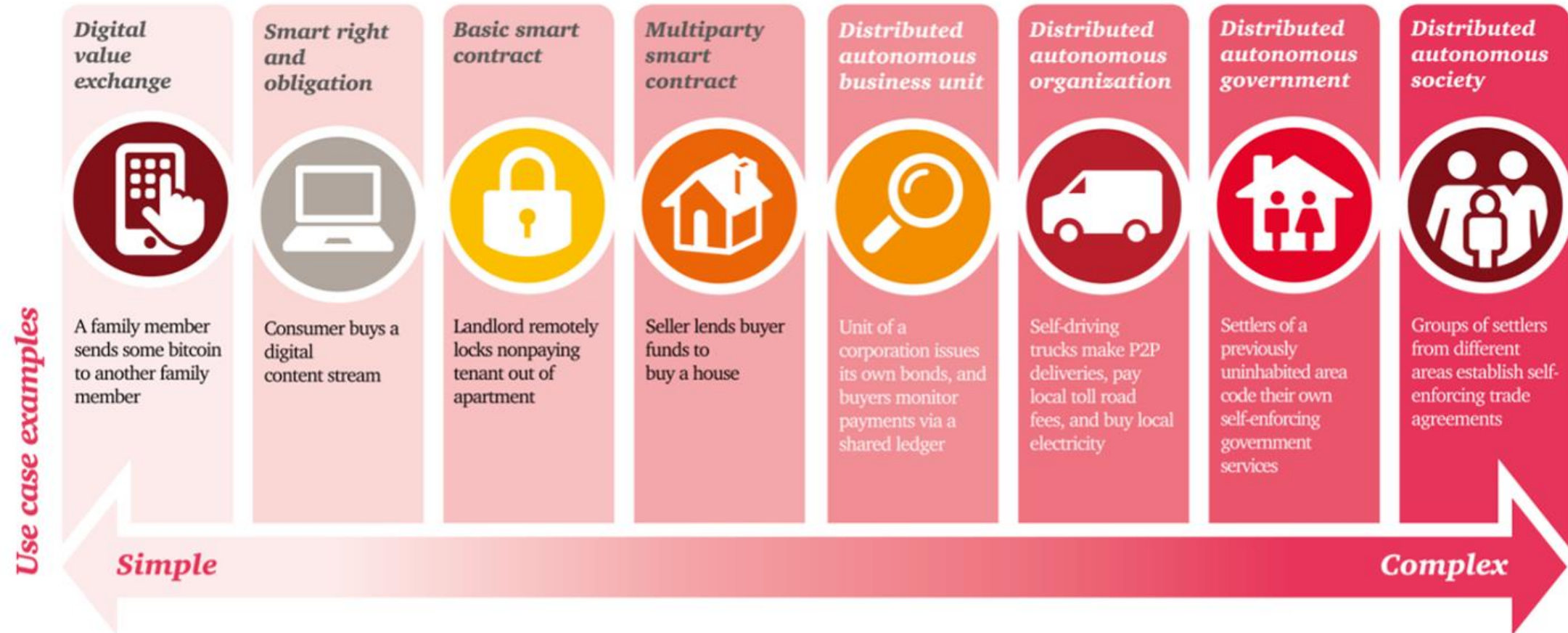
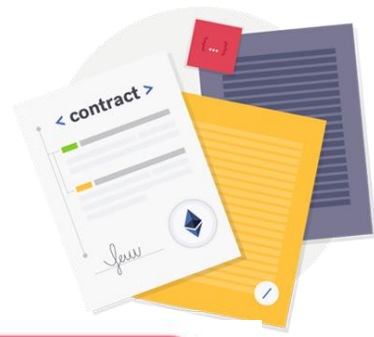
Regulators/Auditors

Central authorities that keep a tab on the system with a  
wide ranging read-access to blockchain



# Smart Contracts

[FROM SIMPLE TO COMPLEX]



# DAO - Decentralizes Autonomous Organizations

- Organizations that exists **autonomously** on a blockchain and are **governed by its protocol**;
- It is a corporation whose **bylaws are written entirely in code**;
- **No central control or authority**;
- Humans are only playing a part of a bug/code fixes, otherwise they are just participants;

