Blockchain

Blockchain components, Blocks, transactions,
DLT, consensus, proof of work, mining,
Public vs Private Blockchain

Dr. Sarwan Singh
NIELIT Chandigarh

Agenda

- Introduction to Bitcoin
- Mining, solving puzzle
- Blockchain components - Blocks, transactions, DLT
- consensus, proof of work
- Public vs Private Blockchain

References

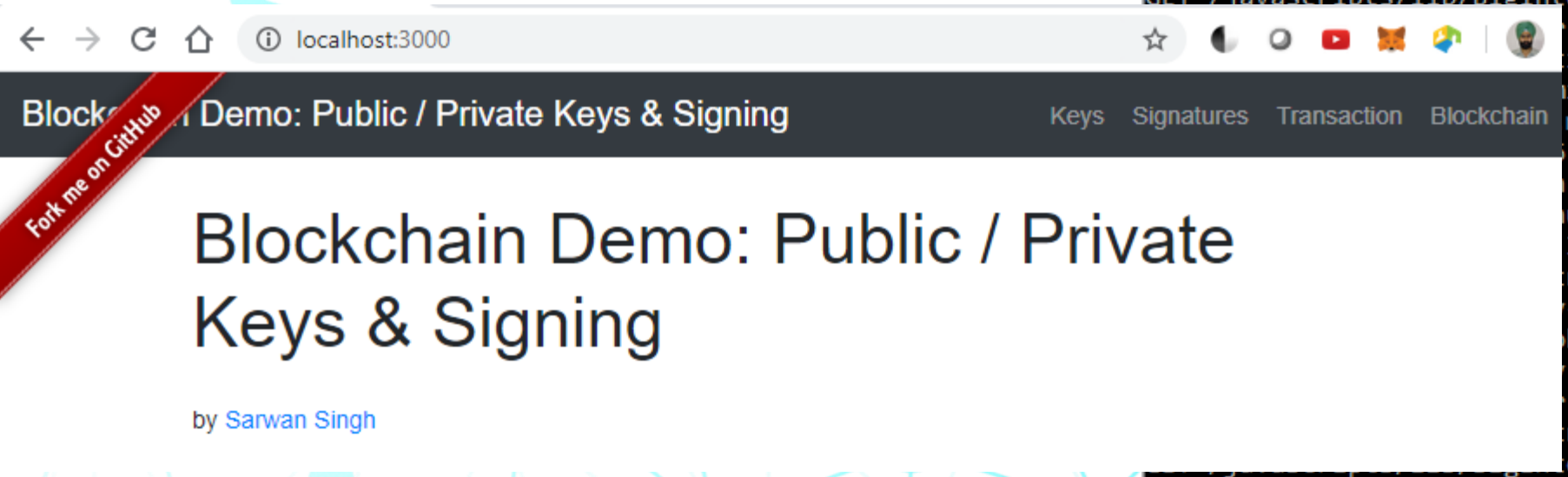
- Medium.com - Blockchain
- “Blockchains Architecture, Design and Use Cases” Prof. Sandip Chakraborty, Prof. Praveen Jayachandran IIT Kharagpur
- IBM Blockchain Labs
- “Conceptualizing blockchains: characteristics & applications” 11th IADIS International Conference Information Systems 2018
- blockgeeks, appinventiv
- consensys.net – Blockchain application areas, use-case
- flatworldbusiness.wordpress.com
- 101blockchains.com

Public-Private key Demo

- npm start

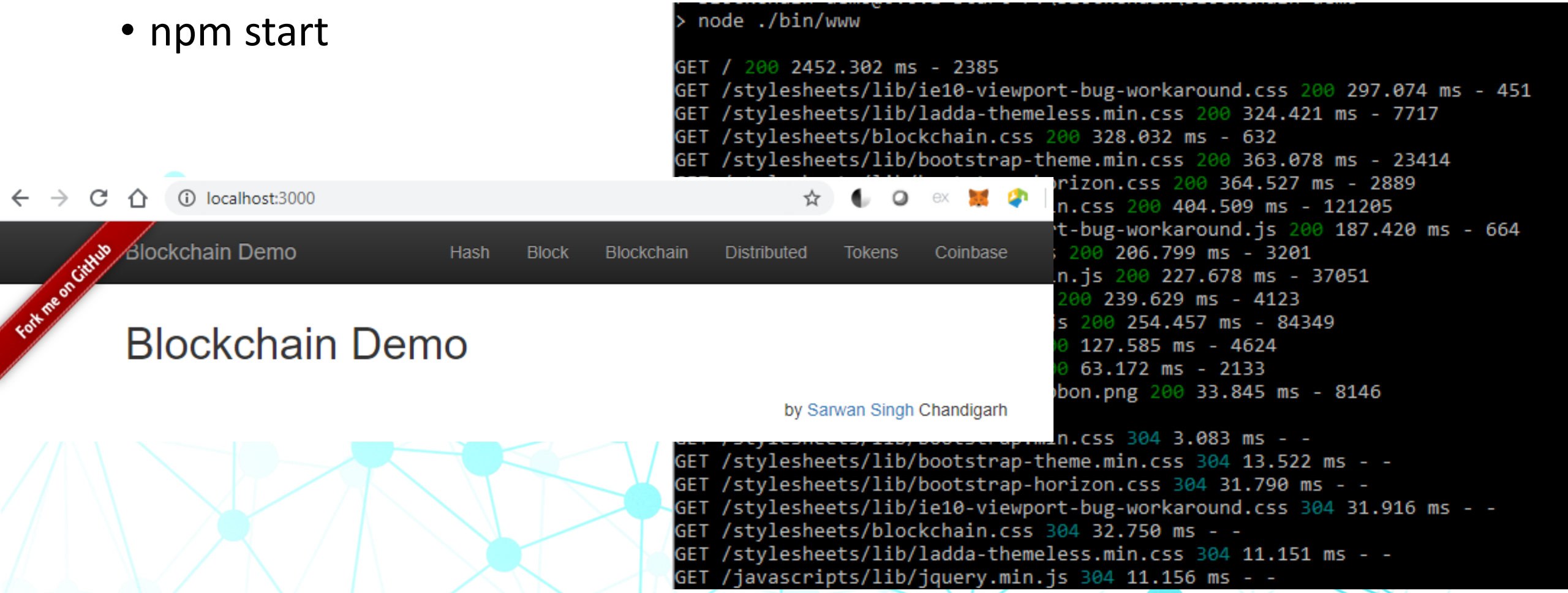
```
> node ./bin/www

GET / 200 1326.684 ms - 2246
GET /stylesheets/lib/bootstrap.min.css 304 9.460 ms - -
GET /stylesheets/public-key-private-key.css 304 18.051 ms - -
GET /javascripts/lib/jquery.min.js 304 19.106 ms - -
GET /javascripts/lib/popper.min.js 304 19.214 ms - -
GET /javascripts/lib/bootstrap.min.js 304 5.085 ms - -
GET /javascripts/lib/js.cookie.min.js 304 5.466 ms - -
GET /javascripts/lib/BigInteger.min.js 304 5.862 ms - -
.js 304 1.583 ms - -
ic.js 304 2.753 ms - -
in.js 304 3.166 ms - -
min.js 304 8.798 ms - -
.js 304 9.332 ms - -
.js 304 0.544 ms - -
ub-ribbon.png 304 0.829 ms - -
3392
rap.min.css 304 1.214 ms - -
-private-key.css 304 9.930 ms - -
kie.min.js 304 4.610 ms - -
.min.js 304 5.365 ms - -
.min.js 304 6.507 ms - -
rap.min.js 304 14.942 ms - -
eger.min.js 304 16.101 ms - -
GET /javascripts/lib/Buffer.js 304 5.175 ms - -
GET /javascripts/lib/elliptic.js 304 6.061 ms - -
GET /javascripts/lib/spin.min.js 304 9.276 ms - -
```



Blockchain Demonstration

- npm start



Bitcoin

- **Bitcoin** is a decentralized digital currency that enables instant payments to anyone, anywhere in the world. Bitcoin uses peer-to-peer technology to operate with no central authority: transaction management

  and money issuance are carried out collectively by the network.
- The original Bitcoin software by Satoshi Nakamoto was released under the MIT license. Most client software, derived or "from scratch", also use open source licensing.
- Bitcoin is designed around the idea of using cryptography to control the creation and transfer of money, rather than relying on central authorities.
- Bitcoins have all the desirable properties of a money-like good. They are portable, durable, divisible, recognizable, fungible, scarce and difficult to counterfeit.

Bitcoin Transactions are …

- **Permissionless** and **borderless**. The software can be installed by anybody worldwide.
- **Do not require any ID** to use. Making it suitable for the unbanked, the privacy-conscious, computers or people in areas with underdeveloped financial infrastructure.
- Are **censorship-resistant**. Nobody is able to block or freeze a transaction of any amount.
- **Irreversible** once settled, like cash. (but consumer protection is still possible.)
- **Fast**. Transactions are broadcasted in seconds and can become irreversible within an hour.
- Online and available **24 hours a day, 365 days per year**.

Stored Bitcoins

Bitcoin can also be a store of value, some have said it is a "swiss bank account in your pocket".

- Cannot be printed or debased. **Only 21 million bitcoins will ever exist**.
- Have **no storage costs**. They take up no physical space regardless of amount.
- Are **easy to protect and hide**. Can be stored encrypted on a hard disk or paper backup.
- Are in your **direct possession** with no counterparty risk. If you keep the private key of a bitcoin secret

- Bitcoin uses public-key cryptography, peer-to-peer networking, and proof-of-work (consensus model) to process and verify payments. Bitcoins are sent (or signed over) from one address to another

- A Bitcoin address is a pseudonymous identity that is used to send and receive bitcoins.
- An address can be described as the hash of an EC (Elliptic Curve) public key and the accompanying private key is used to produce ECDSA (Elliptic Curve Digital Signature Algorithm) signatures to authorize payments.

Bitcoin mining - solving puzzle

The Data

- This is the hash of the latest block (shortened to 30 characters):

  0000000000001adf44c7d69767585 (13 leading zeros)

The hashes of a few valid transactions waiting for inclusion (shortened)

5572eca4dd4

db7d0c0b845

the hash of one special transaction which gives 25BTC (the current reward) to yourself

916d849af76

Bitcoin mining - solving puzzle

Building the next block:

use a gross approximation of what a new block might look like (the real one uses binary format). It contains the hash of the previous block and the hashes of those 3 transactions:

0000000000001adf44c7d69767585 -- 5572eca4dd4 --db7d0c0b845 --916d849af76 --

mining by hand! Our goal is to complete this block with a nonce (a piece of garbage) such that the hash of the new block starts with 13 zeros

(Bitcoin uses double sha256 but we use md5 for ease)

try with nonce=1 {nonce can be any random hexadecimal value},

"0000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--1" | md5sum

Output : 8b9b994dcf57f8f90194d82e234b72ac Not accepted

Bitcoin mining - solving puzzle

Building the next block …

try with nonce=1,
"0000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--1" | md5sum
Output : 8b9b994dcf57f8f90194d82e234b72ac Not accepted

try with nonce=2,
"0000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--2" | md5sum
Output : 5b7ce5bcc07a2822f227fcae7792fd90 Not accepted

Bitcoin mining - solving puzzle

Building the next block …

try with nonce=16,
"00000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--16" | md5sum
Output : 03b80c7a34b060b33dd8fbbece79cee3 Not accepted

try with nonce=208,
"00000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--208" | md5sum
Output : 0055e55df5758517c9bed0981b52ce4a Not accepted

If you finally find a hash that has 13 leading zeroes… you're a winner!
Other miners will now build upon your block, you've just got 25BTC

Bitcoin mining - solving puzzle

Building the next block …

try with nonce=934,224,175,
"00000000000001adf44c7d69767585--5572eca4dd4-db7d0c0b845-916d849af76--934224175" | md5sum
Output : 000000000000005633dd8fbbece79cee3 accepted
But to solve the puzzle it took 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.

Bitcoin mining - solving 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
- completed in 10 minutes, 14 seconds

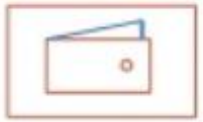
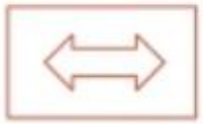
Core Blockchain Architecture Components:

- **Node** — user or computer within the blockchain
- **Transaction** — smallest building block of a blockchain system
- **Block** — a data structure used for keeping a set of transactions which is distributed to all nodes in the network
- **Chain** — a sequence of blocks in a specific order
- **Miners** — specific nodes which perform the block verification process
- **Consensus**— a set of rules and arrangements to carry out blockchain operations

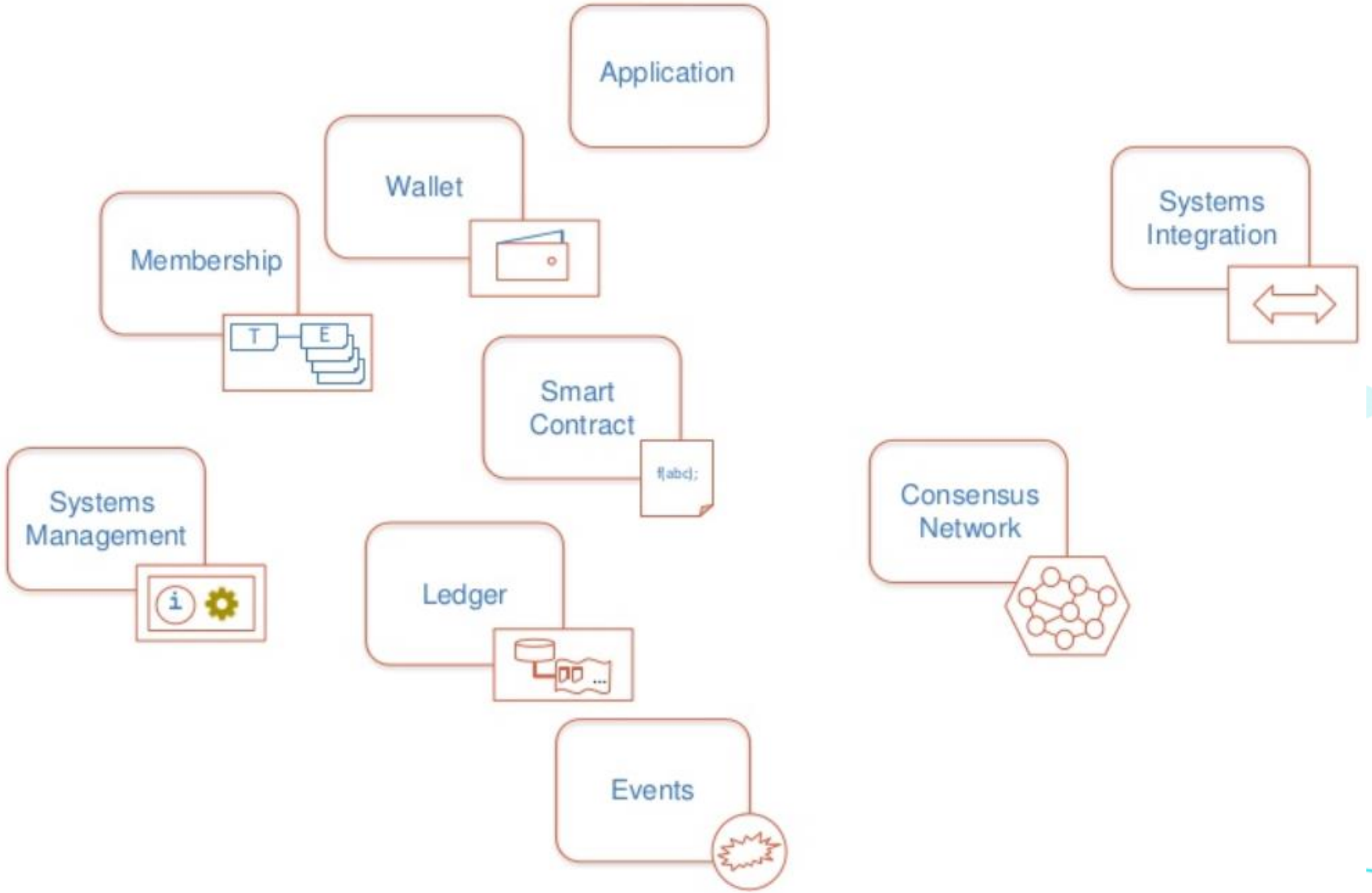
Actors in Blockchain solution

Blockchain Architect	Responsible for the architecture and design of the blockchain solution
Blockchain User	The business user, operating in a business network. This role interacts with the Blockchain using an application. They are not aware of the Blockchain.
Blockchain Regulator	The overall authority in a business network. Specifically, regulators may require broad access to the ledger's contents.
Blockchain Developer	The developer of applications and smart contracts that interact with the Blockchain and are used by Blockchain users.
Blockchain Operator	Manages and monitors the Blockchain network. Each business in the network has a Blockchain Network operator.
Membership Services	Manages the different types of certificates required to run a permissioned Blockchain.
Traditional Processing Platform	An existing computer system which may be used by the Blockchain to augment processing. This system may also need to initiate requests into the Blockchain.
Traditional Data Sources	An existing data system which may provide data to influence the behavior of smart contracts.

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



Consensus Model

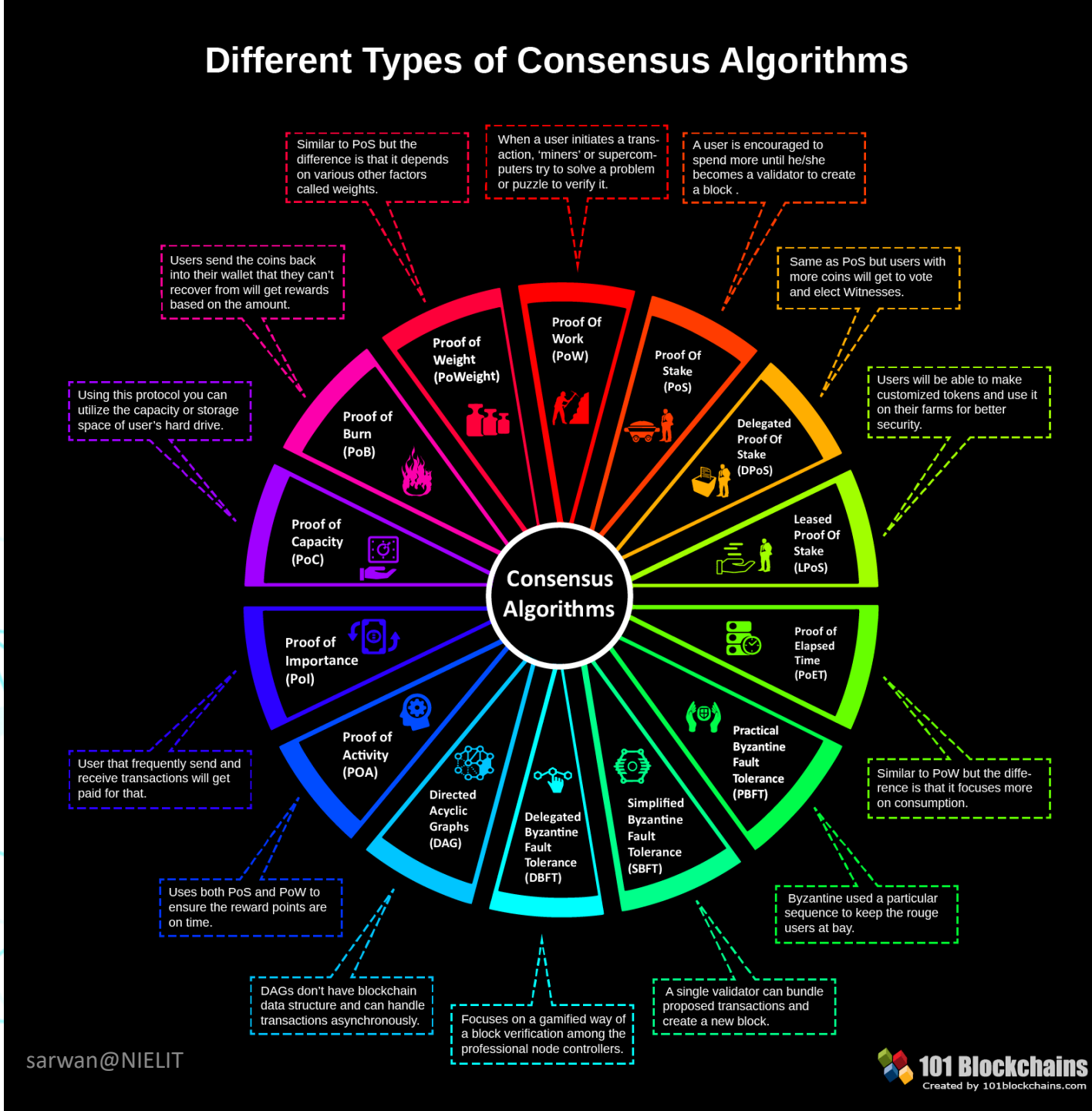
- As blockchain is a decentralized peer-to-peer system with no central authority, which makes it a corruption free but this may be problematic in some ways:
  - How to make decisions ? *"when/how new block will be added"*
  - How to execute some job ? *"who will add new block"*
- In a centralized system, central server do this job or has this role as leader.
- In a blockchain, as there is no leader as such decisions are taken on the basis of consensus.

"Consensus decision-making is a group decision-making process in which group members develop, and agree to support a decision in the best interest of the whole. Consensus may be defined professionally as an acceptable resolution, one that can be supported, even if not the "favorite" of each individual. Consensus is defined by Merriam-Webster as, first, general agreement, and second, group solidarity of belief or sentiment."

wiki

Consensus Models

- A key aspect of blockchain technology is determining which user publishes the next block.
- This is solved through implementing one of many possible consensus models.
- Consensus models enables a group of mutually distrusting users to work together.



Objective of Consensus mechanism

- Agreement Seeking
- Collaborative
- Cooperative
- Egalitarian
  - each and every vote has equal weightage
- Inclusive
- Participatory

Byzantine Generals Problem

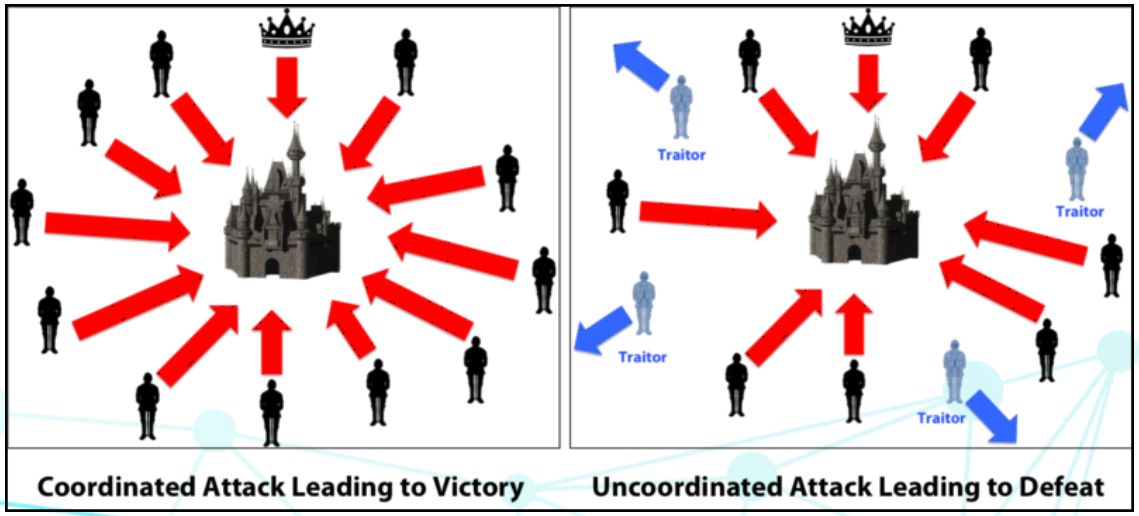
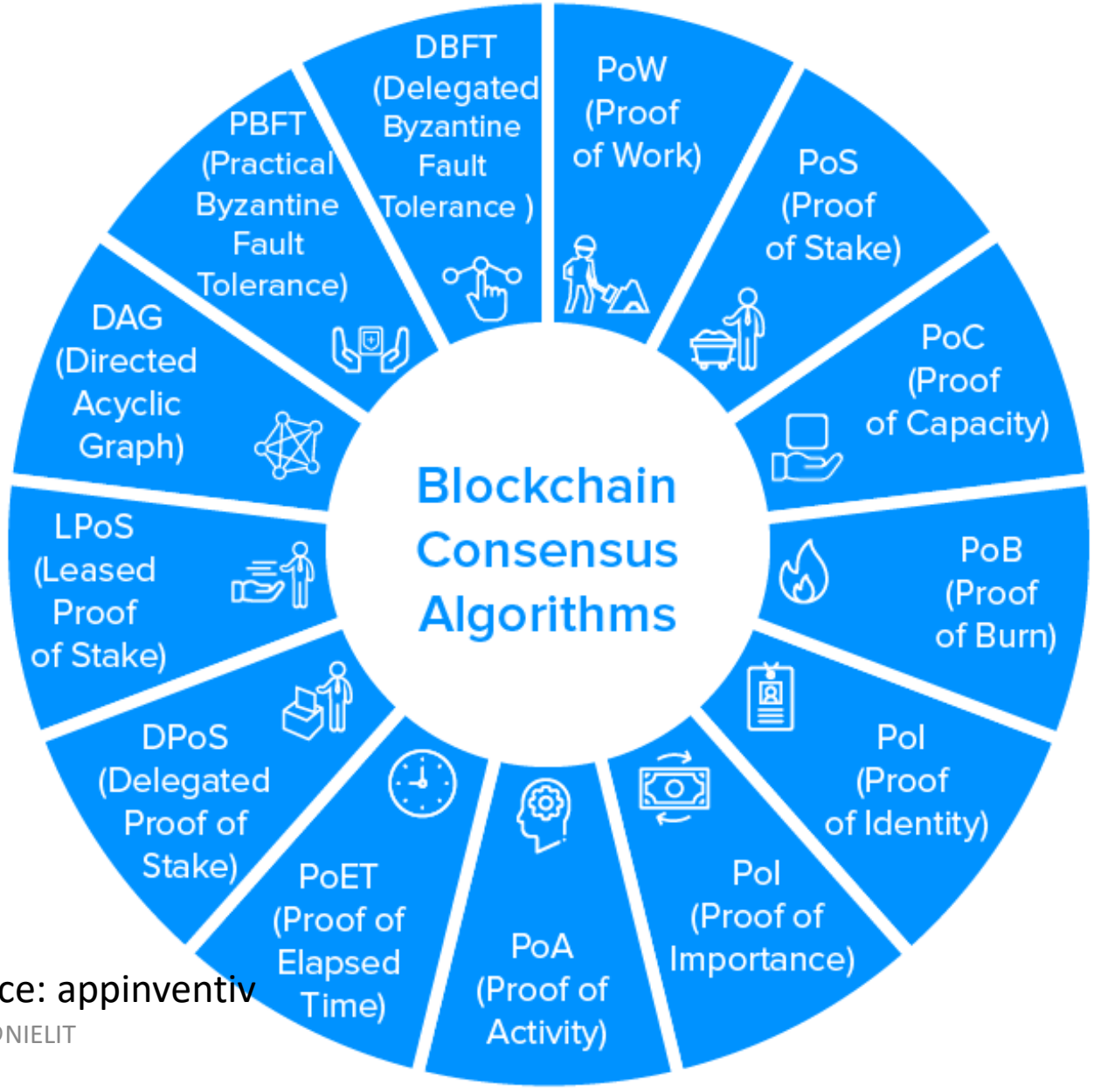


Image Courtesy: Medium

Source : wiki

Popular Consensus models

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Byzantine Fault Tolerance (BFT)
- Direct Acyclic Graph (DAG)
- Proof of Capacity (PoC)
- Proof of Burn (PoB)
- Proof of Identity (PoI)
- Proof of Activity (PoA)
- Proof of Elapsed Time (PoET)
- Proof of Importance (PoI)



Source: appinventiv

Mining Process

In a blockchain each block contains a record of many transactions on the network.

Creating new blocks gives out a reward, also known as the "miner's fee".

Due to large number of participants, we need to maintain scarcity of the reward tokens and regulate who gets right to create the next block.

To achieve this, each participant in the network must solve a complex cryptographic problem (also known as "proof of work").

Whoever solves the problem earn the right to create the next block (and gets the reward).

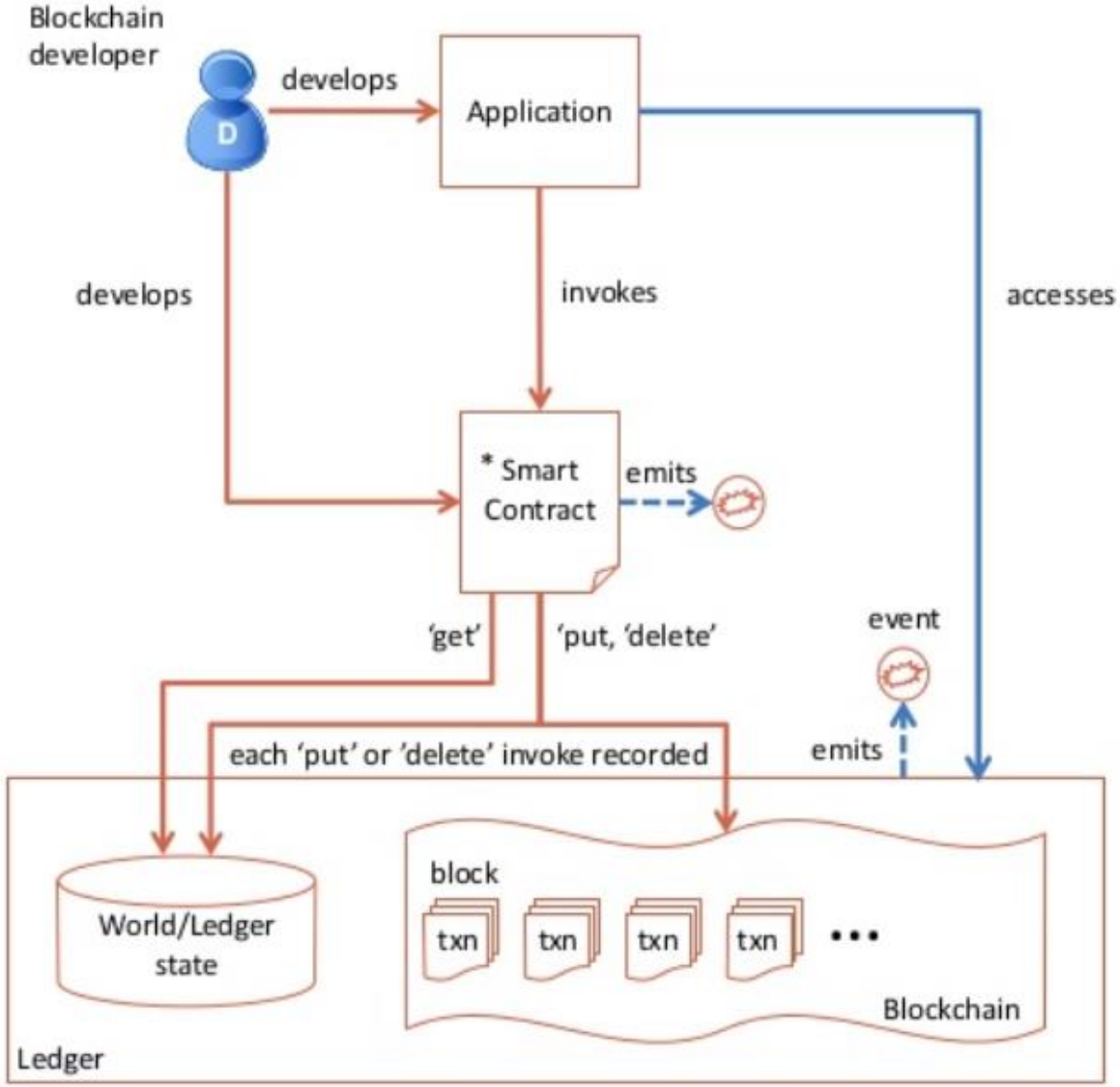
The disadvantage to this is, these problems are very resource intensive and take a substantial amount of computational power to solve.

Proof of Work (PoW)

- Developed by Satoshi Nakamoto
- It is the oldest consensus mechanism used in the Blockchain domain.
- It is also known as mining where the participating nodes are called miners.
- Working : the miners have to solve complex mathematical puzzles using comprehensive computation power.
  - They use different forms of mining methods, such as GPU mining, CPU mining, ASIC mining, and FPGA mining. And the one that solves the problem at the earliest gets a block as a reward.
- This process is not that easy.

Consensus Algorithms	Blockchain Platform	Launched Since	Programming Languages	Smart Contracts	Pros	Cons
PoW	Bitcoin	2009	C++	No	Less opportunity for 51% attack Better Security	Greater energy consumption Centralization of Miners
PoS	NXT	2013	Java	Yes	Energy efficient More decentralized	Nothing-at-stake problem
DPoS	Lisk	2016	JavaScript	No	Energy efficient Scalable Increased security	Partially centralized Double spend attack
LPoS	Waves	2016	Scala	Yes	Fair usage Lease Coins	Decentralization Issue
PoET	Hyperledger Sawtooth	2018	Python, JavaScript, Go, C++, Java, and Rust	Yes	Cheap participation	Need for specialized hardware Not good for Public Blockchain
PBFT	Hyperledger Fabric	2015	JavaScript, Python, Java REST and Go	Yes	No Need for Confirmation Reduction in Energy	Communication Gap Sybil Attack
SBFT	Chain	2014	Java, Node, and Ruby	No	Good Security Signature Validation	Not for Public Blockchain
DBFT	NEO	2016	Python,.NET, Java, C++, C, Go, Kotlin, JavaScript	Yes	Scalable Fast	Conflictions in the Chain
DAG	IOTA	2015	Javascript, Rust, Java Go, and C++	In Process	Low cost network Scalability	Implementation gaps Not suited for smart contracts
POA	Decred	2016	Go	Yes	Reduces the probability of the 51% attack Equal contribution	Greater energy consumption Double signing
PoI	NEM	2015	Java, C++XEM	Yes	Vesting Transaction partnership	Decentralization Issue
PoC	Burstcoin	2014	Java	Yes	Cheap Efficient	Favoring bigger fishes

Blockchain Applications and the Ledger



Blockchain applications

Application

- focus on blockchain user business needs and experience
- Calls smart contract for interactions with ledger state
- Can access transaction ledger directly if required.
- Can process events if required.