

Dependable Distributed Systems Notes

José Duarte

June 18, 2020

1 Blockchains

A blockchain is an immutable ledger for recording ordered transactions, maintained within a distributed network of mutually untrusting peers.

In a blockchain transactions are grouped into blocks, each block is linked (chained) to the previously mined block.

In Bitcoin a block can be thought as an header, containing meta-information and the list of transactions, these blocks have the limit size of 1MB. The information contained in the header is the following:

- **Magic Number** - size of 4 bytes.
- **Block** - size of 4 bytes.
- **Version** - size of 4 bytes.
- **Previous Block Hash** - size of 32 bytes.
- **Merkle Root** - size of 32 bytes.
- **Timestamp** - size of 4 bytes.
- **Difficulty Target** - size of 4 bytes.
- **Nonce** - size of 4 bytes.
- **Transaction Count** - size of 1 to 9 bytes.
- **Transaction List** - size of up to 1 megabyte.

1.1 Ownership

To prove ownership Bitcoin uses public key cryptography, for a transfer between A and B of value m we have the following entry in the log:

$$[K_{pub}A, K_{pub}B, m]_{K_{priv}A} : Sign_{K_{priv}A}(T)$$

This way Bitcoin allows anyone with the full list of transfers to verify what user X owns. Furthermore, the use of public key cryptography allows users to remain anonymous, being able to create a new key pair whenever needed.

1.2 Consensus

In Bitcoin one has multiple miners concurrently trying to generate the next block N , whenever one is able to achieve said goal, it makes the new block public, broadcasting it to the network. The other miners then will stop trying to generate block N and move to trying to generate block $N + 1$.

1.2.1 Confirmation

Consider the block N , transactions on such block are required to be confirmed, the process of confirmation is done by mining blocks ahead, that is, blocks $N + 1$, $N + 2$, etc.

Transactions are typically considered to be *confirmed* when 6 ($N + 6$) or more confirmations have been received. For newly minted Bitcoins the number of confirmations required is around 100 ($N + 100$).

1.2.2 Proof-of-Work

The intuition behind Proof-of-Work (POW) is the following, for a given goal, that goal needs to be hard to achieve but easy for others to verify that such goal has been achieved.

Bitcoin uses the Hashcash POW, a summary of the POW is as follows: For the content **Hello, World!** the miner will append a nonce within the interval $[0 - 2^{240}[$, then the miner will hash the content with the appended nonce until the result starts with a given number of zeros.

This is easy to validate by other miners, just send them the content, nonce and candidate hash, they can easily hash the content and nonce to check if it matches the candidate.

1.3 Service Planes

Blockchains are complex systems and thus have a lot of moving parts, these moving parts can be defined as service planes and serve different purposes.

- **Network** - P2P Networking, Ordered Transaction.
- **Ledger** - Decentralized Logging.
- **Transaction Management** - Transaction Message Format and Verification.
- **Consensus** - POW Mechanism.
- **Cryptography** - Public Key Digital Signatures.
- **Storage** - Persistence and Data-Structures.
- **Access Control** - Permission Management.
- **Participation & Membership** - Role Management and "Gatekeeping".
- **Governance** - Node Behavior.

A block can also have other kinds of tools such as:

- External APIs & Integration Facilities
- Deployment and Operation Services
- Tools

1.4 Blockchain Procurement & Analysis

Before taking on any kind of blockchain one must evaluate the goals of the task at hand, trying its best to match them with the available blockchains. Some common picking parameters are:

- Kind of Network
- Language support
- Popularity, Activity, Community, Documentation
- Scalability and Performance
- Readiness for Deployment
- Virtualization and Isolation Support

1.5 Permissionless vs. Permissioned

1.5.1 Permissionless

Permissionless (or public) blockchains are available for anyone to join, in general a participant is only required to have an (anonymous) identifies, these blockchains do not have access control. Bitcoin and Ethereum are examples of existing permissionless blockchains.

1.5.2 Permissioned

Permissioned blockchains are run among a set of known, verifiable and identified participants, they are also called *Consortium Blockchains* or *Private Blockchains*. This model targets a network where participants have a common goal and interact securely but do not fully trust each other. Nodes may also have different roles in the network.

The base idea behind permissioned blockchains is that to participate a party needs to fulfill a group of requirements. These requirements are such as having an authorized ID and being accepted by other nodes.

Traditionally these blockchains run traditional byzantine consensus algorithms. This works because nodes are previously known and there is membership and access control.

1.6 Ethereum

Like Bitcoin, Ethereum is a public blockchain, however it has several extra features such as smart contracts and the possibility for usage of Proof-of-Stake (POS). Casper will implement two rounds of voting (the *prepare* and *commit* phases), furthermore Casper will slash bad validators. The Ethereum blockchain is also faster than Bitcoin and the reward is different.

1.6.1 Block

An Ethereum block consists of three main elements, the block header, a transaction list and an Ommer list.

Ommer List The Ommer List contains all included Uncle blocks. Uncle blocks are valid solutions to the POW that do not make the main chain. This has the objective of decreasing centralization of the network and rewarding work. If a given Uncle block is included in a main block, 1/32 of the main block miner's reward is given to the Uncle block miner.

1.6.2 Nodes

Ethereum nodes validate all transactions and new blocks, they operate in a P2P fashion and each contain a copy of the entire blockchain. There also exist light clients, these will only execute validations, being mainly used for account balance verification.

1.6.3 Accounts & Wallets

There exist two kinds of accounts on the Ethereum blockchain, these are external owned accounts (EOA) and contract accounts. Both kinds consist of a key pair and allow for interaction with the blockchain.

External Account An external account has an associated nonce, balance, hash code and root. The hash code is the hash of the associated account code (i.e. program code). The root is the root hash of the account associated tree.

Contract Account Contract accounts can store and execute code, they have an associated nonce and balance, as well as an hash code and root.

Wallet Wallets are a set of one or more external accounts, these can be used to store and transfer *ether*.

1.6.4 Casper Protocol

The Casper protocol was proposed as a way to improve on the POW of the Ethereum blockchain, it is a smart contract that implements and monitors POS.

Proof-of-Stake The creator of the next block is chosen by some combination of randomness and a stake (e.g. a number of coins). When compared with traditional POW, POS consumes less energy and can improve throughput conditions.

Validators Validators are nodes that have two voting functions, *prepare* and *commit*, the votes are then weighed by the amount the voter staked. Each validator can only vote once.

Protocol Validators will select a pending block to be prepared, to do so, the validator will send a *prepare* vote¹ to the network, staking a certain amount of coins. When 2/3 of the network agree on the block, voting

¹Prepare votes reference the last prepared block as well as the last committed block.

moves on to the *commit* phase. Finally, the prepared block is again required to be voted² by 2/3 of the network, finalizing the block.

1.6.5 Smart Contracts

Smart Contracts are computer protocols intended to digitally facilitate, verify or enforce the negotiation or performance of a contract. They allow the performance of credible transactions without third parties, as verifiable, trackable and irreversible transactions.

Smart contracts comply with the following properties:

- They are executable code.
- The source language is Turing complete.
- They function like an external account.
 - Holding funds.
 - Being able to interact with other accounts as well as contracts.
 - Contain code.
- Smart contracts can be called through transactions.

1.6.6 Contract Execution

In Ethereum the contracts run in the Ethereum Virtual Machine (EVM), there source is written in Solidity. Every full-node on the blockchain processes every transaction and stores the entire state.

Smart contracts are subject to the halting problem, that is, it is impossible to know if they terminate. To deal with this, Ethereum uses the concept of Gas.

Gas Gas is the exchange value used to run code on the blockchain. Each contract is required to provide a maximum gas usage, stopping infinite loops from running. Gas has its own price and market.

²Commit votes reference the last prepare block.