Diving into the Ethereum chain this week. Tl;dr: its whitepaper, in particular, focusing on the history as well as use-cases for a built-in Turing complete programming language primarily to provide the implementation and use of smart contracts. Also the tree constellation, as well as the future for Ethereum with ETH 2.0 coming up.

Smart contracts are used to encode arbitrary state transitions functions. Where Bitcoin tries to build a (seemingly) dedicated monetary system, Ethereum wants to extend the functionality to support multiple use-cases. These take the form of decentralized applications on-chain with the possibility to interact.

The Ethereum network is fueled and secured by the coin Ether. These coins are spent on different transactions both as payment to the network, and the provider of some goods. Coins can be stored in private wallets (with users) or in contracts (think code).

The Ethereum contract is not executed in one physical space, rather it is executed as part of the state transition function. Meaning that when a block is appended to the chain by the validators, the state transition function is executed, and so are smart contracts in the same block.

Where Bitcoin tries to be a dedicated decentralized monetary system, without TTP, Ethereum tries to be the same with more functionality but ends up needing some form of TTP. This might be a semi-decentralized solution (ShellingCoin), or it might be with the help of dedicated oracles. This is, however, specific for those contracts dependent on some data as a trigger or validation.

There are several great use-cases for a chain like Ethereum such as savings, insurance, data feed, cloud computing, gambling, and prediction markets

The tree constellation used in Ethereum is more complicated than in Bitcoin. It’s similar but different. Each transaction as I understand it contains three Patricia trees: the transaction, receipts, and state. The constellation allows for two main properties: ease of verifying data integrity - as the single hash value (root) is representative of all the transaction data if any of the nodes change, so does the root. Also, the ease to verify the inclusion of a specific transaction (Merkle proof) - this enables SPV (simple payment verification), no need to hold a copy of the entire chain -> light client (does only need to receive the block header)

ETH 2.0 is a process planned for some time and is to be acted out in several steps over the upcoming years. One of the main change is the move away from PoW to PoS, also several scaling solutions is investigated.