

Intro to Ethereum

Marisa Paryasto

25 Nov 2019

What is Ethereum

Ethereum is an [open source](#), [public](#), [blockchain](#)-based [distributed computing](#) platform and [operating system](#) featuring [smart contract](#) (scripting) functionality. It supports a modified version of [Nakamoto](#) consensus via transaction-based state transitions. Ether is a [cryptocurrency](#) generated by the Ethereum platform and used to compensate mining nodes for computations performed. Each Ethereum account has an ether balance and ether may be transferred from one account to another.

Ethereum provides a decentralized [virtual machine](#), the Ethereum Virtual Machine (EVM), which can execute scripts using an international network of public nodes. The virtual machine's [instruction set](#), in contrast to others like [Bitcoin Script](#), is thought to be [Turing-complete](#). "Gas", an internal transaction pricing mechanism, is used to mitigate [spam](#) and allocate resources on the network.

- Wikipedia

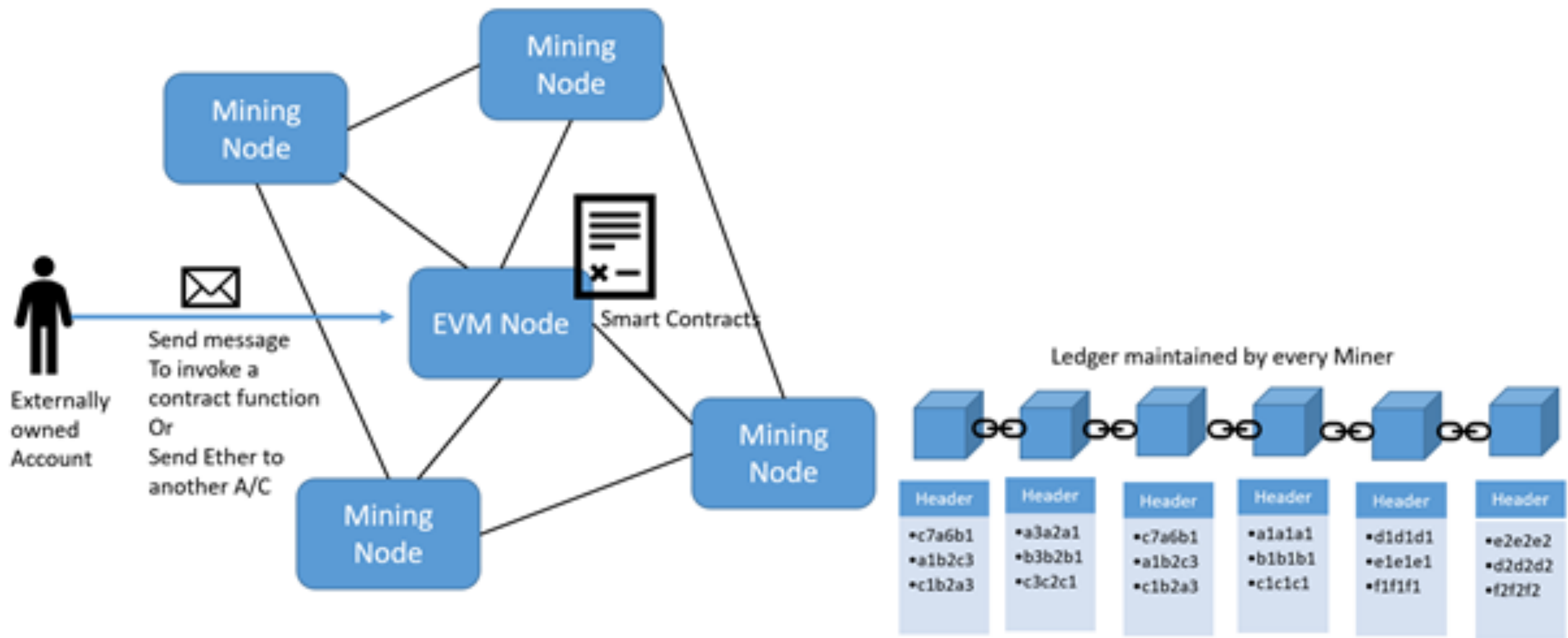
Ether

Ether is the currency of Ethereum. Every activity on Ethereum that modifies its state costs Ether as fee and miners who are successful in generating and writing a block in chain are also rewards Ether. Ether can easily be converted to dollars or other traditional currencies through Crypto-exchanges.

Gas

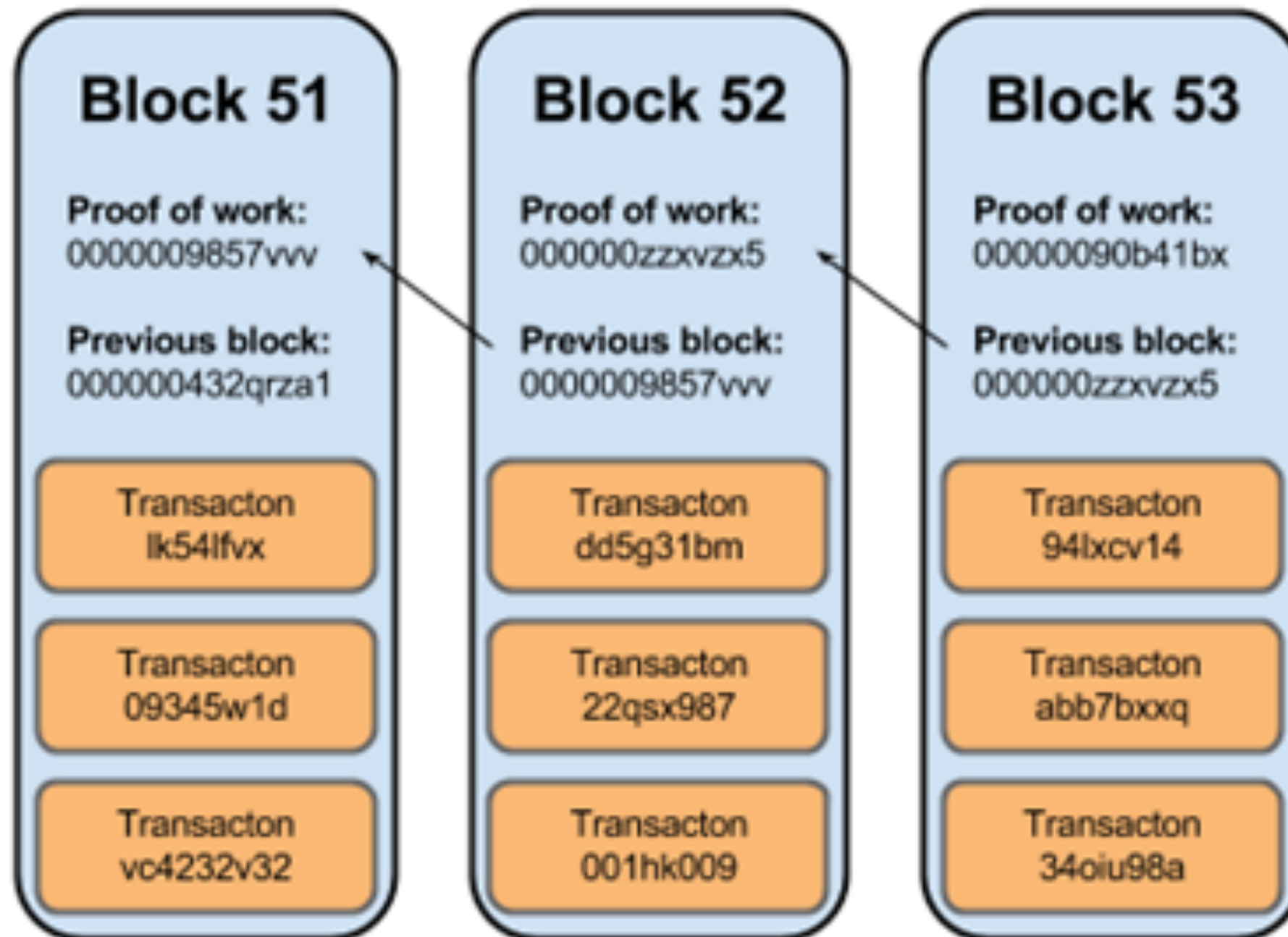
Ether is paid as fees for any execution that changes state in Ethereum. Ether is traded on public exchanges and its price fluctuates daily. If Ether is used for paying fees, then the cost of using the same service could be very high on certain days and low on other days. People will wait for price of Ether to fall to execute their transactions. This is not ideal for a platform like Ethereum. Gas helps in alleviating this problem. Gas is the internal currency of Ethereum. The execution and resource utilization cost is predetermined in Ethereum in terms of Gas units. This is also known as Gas Cost. There is also Gas price that can be adjusted to lower price when price of Ether increases and higher price when price of Ether decreases. For example, to invoke a function in a contract that modifies a string will cost Gas which is pre-determined, and Users should pay in terms of Gas to ensure smooth execution of this transaction.

Blockchain and Ethereum Architecture

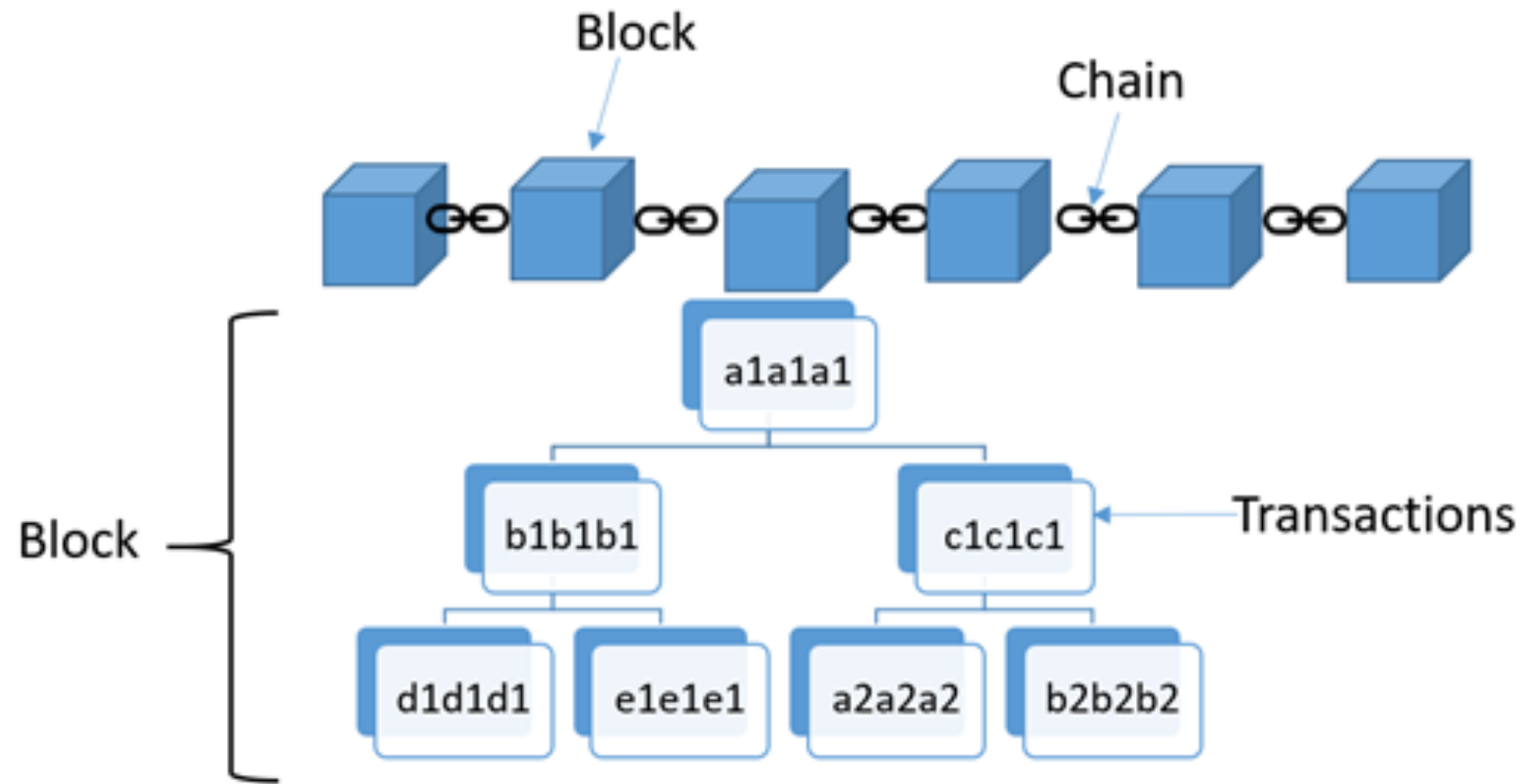


*<https://medium.com/coinmonks/https-medium-com-ritesh-modi-solidity-chapter1-63dfaff08a11>

How are blocks related to each other



How are transactions and blocks related to each other



Nodes

There are two types of nodes in Ethereum.

- **Ethereum virtual machines** and
- **Mining Nodes**

It is to be noted that this distinction is made to clarify concepts of Ethereum. In most scenarios' there is no dedicated EVM machines instead all nodes acts as miner as well as EVM node.

Ethereum Virtual Machines (EVM)

Think of EVM as the execution runtime for Ethereum network. EVM's are primarily responsible to provide a runtime that can execute code written in smart contracts. It can access accounts — contract and externally owned, its own storage data. It does not have access to ledger but has limited information about current transaction.

EVM are the execution component in Ethereum. The purpose of EVM is to execute the code in smart contract line by line. However, when a transaction is submitted, the transaction is not executed immediately instead is it pooled in a transaction pool. These transactions are not yet executed and not yet written to the Ethereum ledger. EVM nodes are similar to Mining nodes however they do not do mining.

Mining Nodes

A miner is responsible for writing transactions to the Ethereum chain. A miner job is very similar to that of an accountant. As an accountant is responsible for writing and maintain the ledger similarly, a Miner is solely responsible for writing transaction to Ethereum ledger. A miner is interested in writing transactions to ledger because of the reward associated with it. Miners get two types of reward — reward for writing a block to the chain and cumulative gas fees from all transactions in the block. There are generally many miners available within a blockchain network each trying and competing to write transactions. However only one miner can write the block to the ledger and rest will not able to write the current block and determination of a miner who will write the block happens using a challenge. The challenge is given to every node and every miner tries to solve the puzzle using its compute power. The miner who solves the puzzle first write block containing transactions to ledger and also receiver 5 ether as reward. Every Mining node maintains its own instance of Ethereum ledger and the ledger is same ultimately across all miners. It is job of miners to ensure that their ledger is updated with latest blocks.

Mining Nodes

There are three important functions performed by Miners or Mining Nodes.

- Mine or create a new block with transaction and write the same to Ethereum Ledger
- Advertise and send a newly mined block to other miners.
- To accept new blocks mined by other miners and keep its own ledger instance up-to-date

Mining Nodes refers the nodes that belong to Miners. These nodes are part of the same network where EVM is hosted. At some point of time, the miners would create a new Block, collect all transaction from transaction pool and adds them to the newly created block. Finally, this Block is added to the chain. There are additional concepts like consensus, solving of target puzzle before writing the block and will be explained in section “How mining works”.

How Does Mining Work

- The miner hashes all the transactions in the block, these hashes are further combined in pairs to generate a new hash. The process continues until there is just one hash for all transactions in the block. The hash is referred as Root transaction hash or Merkel Root transaction hash. This hash is added to the block header.
- The miner also identifies the hash of the previous block. The previous block will become parent to the current block and its hash will also be added to the block header.
- The miner in similar way calculates the State and Receipts transaction root hashes and add them to the block header.
- A nonce and timestamp is also added to the block header. The mining process starts where the miner keeps changing the nonce value and try to find a hash that will satisfy as an answer to the given puzzle. It is to be kept in mind that everything mentioned here is executed by every miner in the network.

Eventually, one of the miner would be able to solve the puzzle and advertise the same to other miners in the network. The other miners would verify the answer and if found correct would further verify every transaction while accept the block and append the same to their ledger instance.

This entire process is also known as Proof of Work wherein a miner provides proof that is has worked on computing the final answer that could satisfy as solution to the puzzle.

Accounts

Accounts are main building block for Ethereum ecosystem. It is the interaction between accounts that Ethereum wants to store as transaction in its ledger. Ethereum supports two types of accounts. Each account has a balance property that returns the current value stored in it.

Transaction

A transaction is an agreement between a buyer and seller, a supplier and a consumer or a provider and a consumer that there would be exchange of assets, products or services in lieu of currency, crypto-currency or some other asset either in present or in future. Ethereum helps in executing transaction. There are four types of transaction that can be executed in Ethereum.

1. **Transfer of Ether from one account to another.** The accounts can be externally owned accounts or contract account. Following are the possible cases

- a. An externally owned account sending ether to another externally owned account in a transaction.
- b. An externally owned account sending ether to a contract account in a transaction.
- c. A contract account sending ether to another contract account in a transaction.
- d. A contract account sending ether to an externally owned account in a transaction

2. **Deployment of Smart contract** — An externally owned account can deploy a contract using a transaction in Ethereum virtual machine.

3. **Using or invoking a function within a contract** — Executing a function in a contract that changes state are considered as transactions in Ethereum. If executing a function does not change state, it does not require a transaction. A transaction has some important properties related to it.

From Account property denotes the account that is originating the transaction and represents an account who is ready to send some gas or ether. We will consider concepts related to Gas and Ether later section in this chapter. From account can be externally owned or a contract account.

To account property refers to an account that is receiving ethers or benefits in lieu of an exchange. In case of transaction related to deployment of contract, the To field is empty. It can be externally owned or a contract account. Value refers to the amount of ether that is transferred from one account to another.

Input refers to the compiled contract bytecode and is used during contract deployment in EVM. It is also used for storing data related to smart contract function calls along with its parameters.

Transaction

Blockhash refers to the hash of Block to which this transaction belongs to.

BlockNumber is the block in which this transaction belongs to.

Gas refers to amount of gas supplied by sender which executing this transaction

GasPrice refers to the price per gas the sender was willing to pay in Wei (Wei is explained in section related to Ether). Total Gas is computed at Gas units * Gas price

Hash refers to the hash of transaction

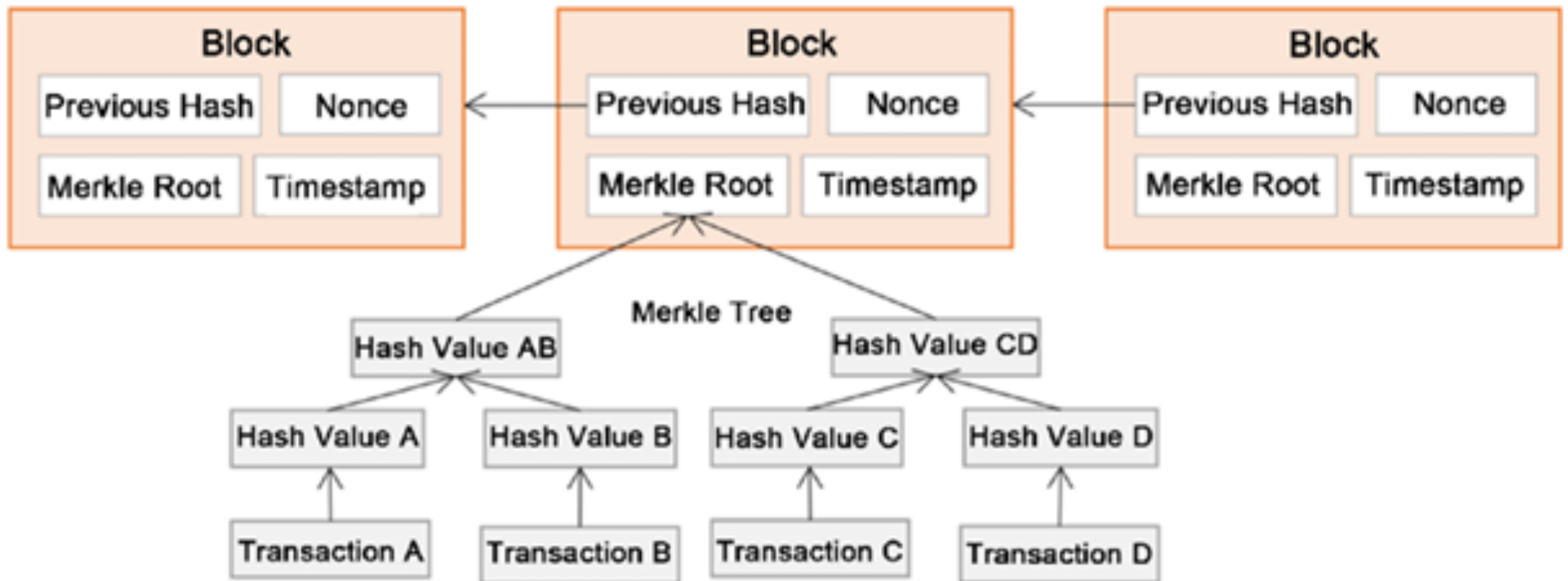
Nonce refers to number of transactions made by the sender prior to current transaction.

TransactionIndex refers to the serial number of current transaction in block

Value refers to amount of ether transferred in Wei

V, R and S relates to digital signature and signing of transaction.

How Does Mining Work



Block

Block is an important concept in Ethereum. Block are containers for transaction. A block contains multiple transactions. Each block has different number of transactions based on Gas limit. Gas limit will be explained in detail in later sections. The blocks are chained together to form blockchain. Each Block has a parent and it stored the hash of parent block in its header. Only the first Block known as Genesis block does not have a parent.

```
{ difficulty: BigNumber { s: 1, e: 5, c: [ 135070 ] },  
  extraData: '0xd783010702846765746885676f312e398777696e646f7773',  
  gasLimit: 4011042861,  
  gasUsed: 43406,  
  hash: '0xba93a91df520c7565e80347346e47b83a41d473a33352d1cf7e689c30b305ba5',  
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000',  
  miner: '0xa57de277ede9c1521f51f6989ed2497a5b9c1926',  
  mixHash: '0x4e80de770c329aebbc2e9e861190784f57b7f9910bbb049534828052284f32e4',  
  nonce: '0x655dee191333922c',  
  number: 70,  
  parentHash: '0x27d3dbc34614f88583f29ea1b7546e83563e55638b1d0a258de04f4912f42aaf',  
  receiptsRoot: '0x5dff465dd85c4ad02c71ec4099284ecaf91ed9bb600f7903978386059c16fb8d',  
  sha3Uncles: '0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347',  
  size: 742,  
  stateRoot: '0xb15363a8958d218eff295b5e877517ee4243f1b681d987793c5ec30a04bc4592',  
  timestamp: 1511421241,  
  totalDifficulty: BigNumber { s: 1, e: 6, c: [ 9302609 ] },  
  transactions:  
    [ '0x6b65b86462e6aa89d5f9469ce03b9ea21e8bf72f8c11aa72de2978f9b7a5b9fd' ],  
  transactionsRoot: '0x5aceca068d1a7ac8d8ecd8f19469ec7c687cb6ceed54e0db39b58dfd6481de9',  
  uncles: [] }
```

Block

The **Difficulty** property determines the complexity of the puzzle/challenge given to miners for this block.

GasLimit determines the maximum gas allowed. This helps in determining how many transaction can be part of the block.

GasUsed refers to the actual gas used for this block for executing all transactions in it.

Hash refers to the hash of the block.

Nonce refers to the number that helping in solving the challenge.

Miner property is the account identifier of miner also known as coinbase or Etherbase

Number is the sequential number of this block on the chain

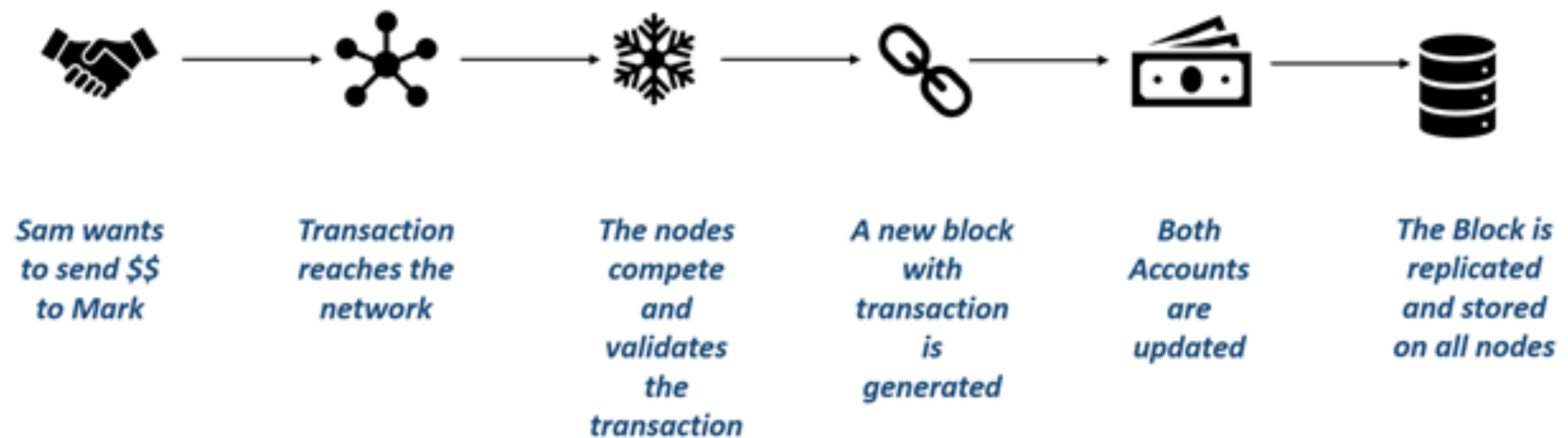
ParentHash refers to parents blocks hash.

ReceiptsRoot, StateRoot and transactionRoot refers to Merkel trees discussed during mining process.

Transactions refers to array of transactions that are part of this block.

TotalDifficulty refers to total difficulty of the chain until this block

End to End Transaction



Smart Contract

A smart contract is a contract implemented, deployed and executed within Ethereum environment. Smart contracts are digitization of the legal contracts. Smart contracts are deployed, stored and executed within the Ethereum Virtual machine. Smart contracts can store data. The data stored can be used to record information, fact, associations, balances and any other information needed to implement logic for real world contracts. Smart contracts are very similar to Object oriented classes. A smart contract can call another smart contract just like an Object-oriented object to create and use objects of another class. Think of smart contract as a small program consisting of functions. You can create an instance of the contract and invoke functions to view and update contract data along with execution of some logic

Solidity Programming Language

Solidity is an object-oriented programming language for writing [smart contracts](#). It is used for implementing smart contracts on various [blockchain](#) platforms, most notably, [Ethereum](#). It was developed by [Gavin Wood](#), Christian Reitwiessner, Alex Beregszaszi, Liana Husikyan, Yoichi Hirai and several former [Ethereum](#) core contributors to enable writing smart contracts on blockchain platforms such as Ethereum.

- Wikipedia