CSE446: Blockchain & Cryptocurrencies

Lecture – 16: Ethereum - 4

Inspiring Excellence

Agenda

- Ethereum Transactions
- Gas and Gas price
- Ethereum Block

Ethereum DoS protection

- EVM code is “Turing complete", e.g., has loops and may not halt
- In order to ensure safety, a type of DoS protection, the concept of fee is used
- Ethereum utilises a “pay as you execute” model
  - Meaning, you must pay for every single execution of operation within your smart-contract
- Users need to pay for both computation and storage of data
- This payment is collected by the respective miner

Ethereum gas and gas price

- Each transaction contains a “gas limit”, and a “gas price” which is the price the sender will pay per unit of gas, denominated in ETH (currency)
- Along with sufficient ETH to pay the fee
- But, how much do you need to pay?
- It depends on how much unit of work, i.e. code execution, is carried out or the amount of data stored in the network
  - The unit of work in Ethereum is defined with the concept of Gas
  - Ether buys GAS to fuel the EVM, like hour(s) of labour

Ethereum gas and gas price

- Each operation in Ethereum utilises some amount of gas
- For example:
  - If you add two numbers, 3 gas is used
  - If you multiply two numbers, 5 gas is used
  - One hash call requires 30 gas
- Ethereum also uses gas for contract creation
- By calculating total operations, we can estimate how much gas might require
- It is difficult to know the exact gas required for contract execution
  - So, set a higher limit using the notion of gasLimit, that implies the maximum amount of gas you are willing to buy

Ethereum gas and gas price

- But, still we don't know how much Ether we have to pay the miners
- That is determined by Gas Price, which is like the hourly labour rate: 500 taka/hour
- Fee for computation in Ether = gasLimit x gasPrice;
  - gasLimit (gas) and gasPrice are specified in the transaction
  - The calculated Fee is placed in Escrow
- Gas price is determined by the sender
- If gasPrice is too low, no miner will bother about processing a transaction

Ethereum gas and gas price

- If the fee is less than required, gas runs out before a transaction is completed
  - State is reverted back to the previous state
  - Fee is consumed from the escrowed Ether
- If execution completes
  - consumed gas x gasPrice is paid to the miner
  - Remaining gas (gasLimit – consumed gas) x gasPrice is returned back to the caller
- The higher the gas price you set, the sooner your transaction gets mined
  - Most miners tend to choose transactions with higher fee
- If price is relatively low, your transaction will eventually get included in the block but you might have to wait for a bit

Ethereum gas and gas price

	Fuel	Fee
General	Every operation in the EVM consumes a **pre-defined amount of gas**: not changeable by user. Every transaction has a **user-specified startGas**	Every transaction has a **user-specified gas price** (current default is 0.02μETH per gas)
At start of transaction	Originator **should** provide enough fuel: startGas. remainingGas = startGas	Originator **must** pay for all the fuel. startGas × gas price = Ether placed in escrow
Each operation	remainingGas is decreased by operation's gas consumption	Deferred until unsuccessful or successful transaction
Unsuccessful transaction	remainingGas is zero and there are operations remaining. This causes an Out of Gas exception and all operations are undone	All the escrowed Ether is paid to miner
Successful transaction	All remainingGas is refunded to originator	(startGas – remainingGas) × gas price = fee paid to miner remainingGas × gas price = refund

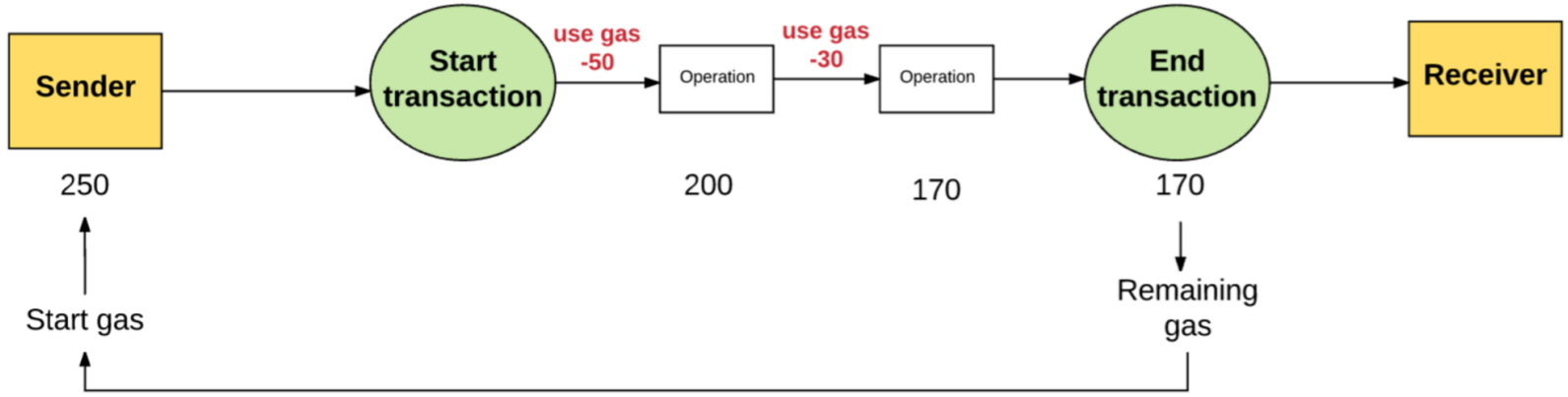
Source: https://media.consensys.net/2016/06/23/ethereum-gas-fuel-and-fees/

Ethereum gas and gas price

- Gas limit: Max no. of computational steps the transaction is allowed.
- Gas Price: Max fee the sender is willing to pay per computation step.

Ethereum gas and gas price

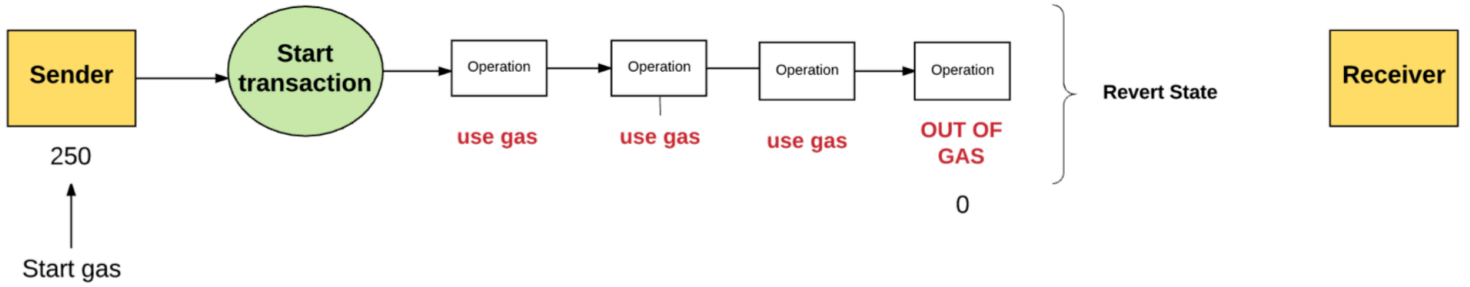
The sender is refunded for any unused gas at the end of the transaction.



From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum gas and gas price

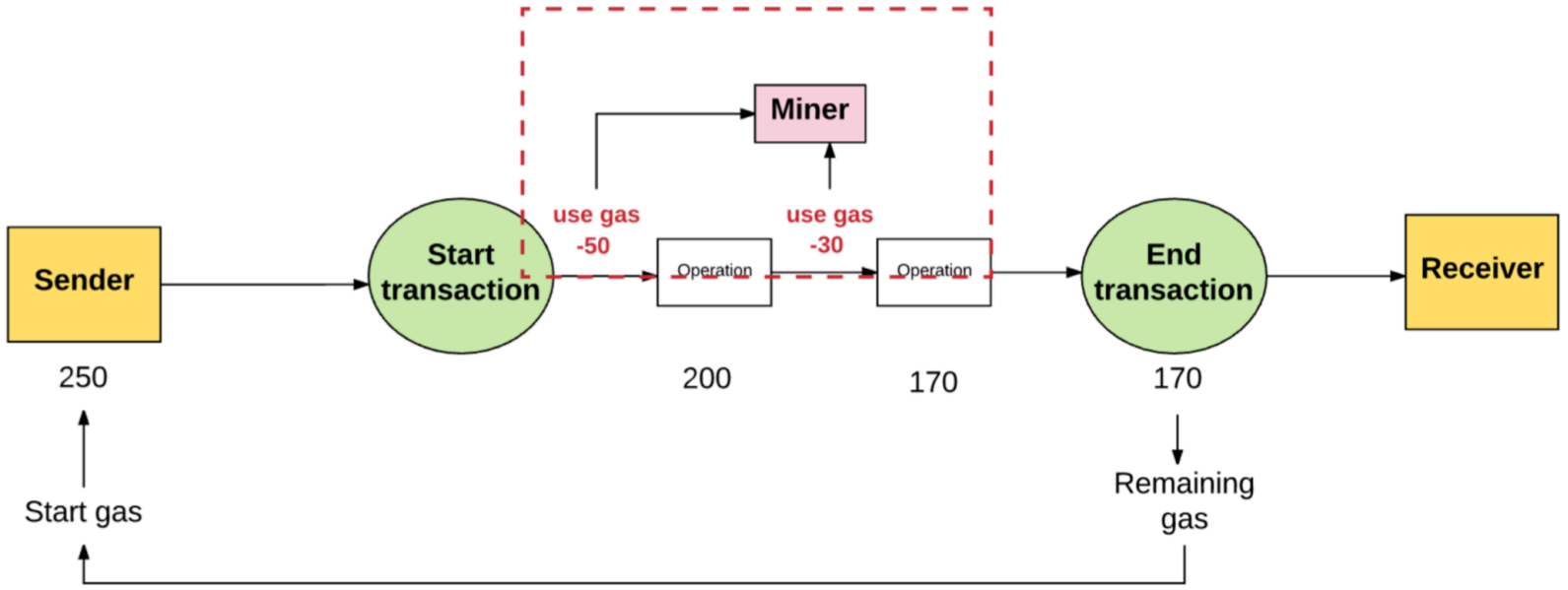
If sender does not provide the necessary gas to execute the transaction, the transaction runs “out of gas” and is considered invalid.



- The changes are reverted.
- None of the gas is refunded to the sender.

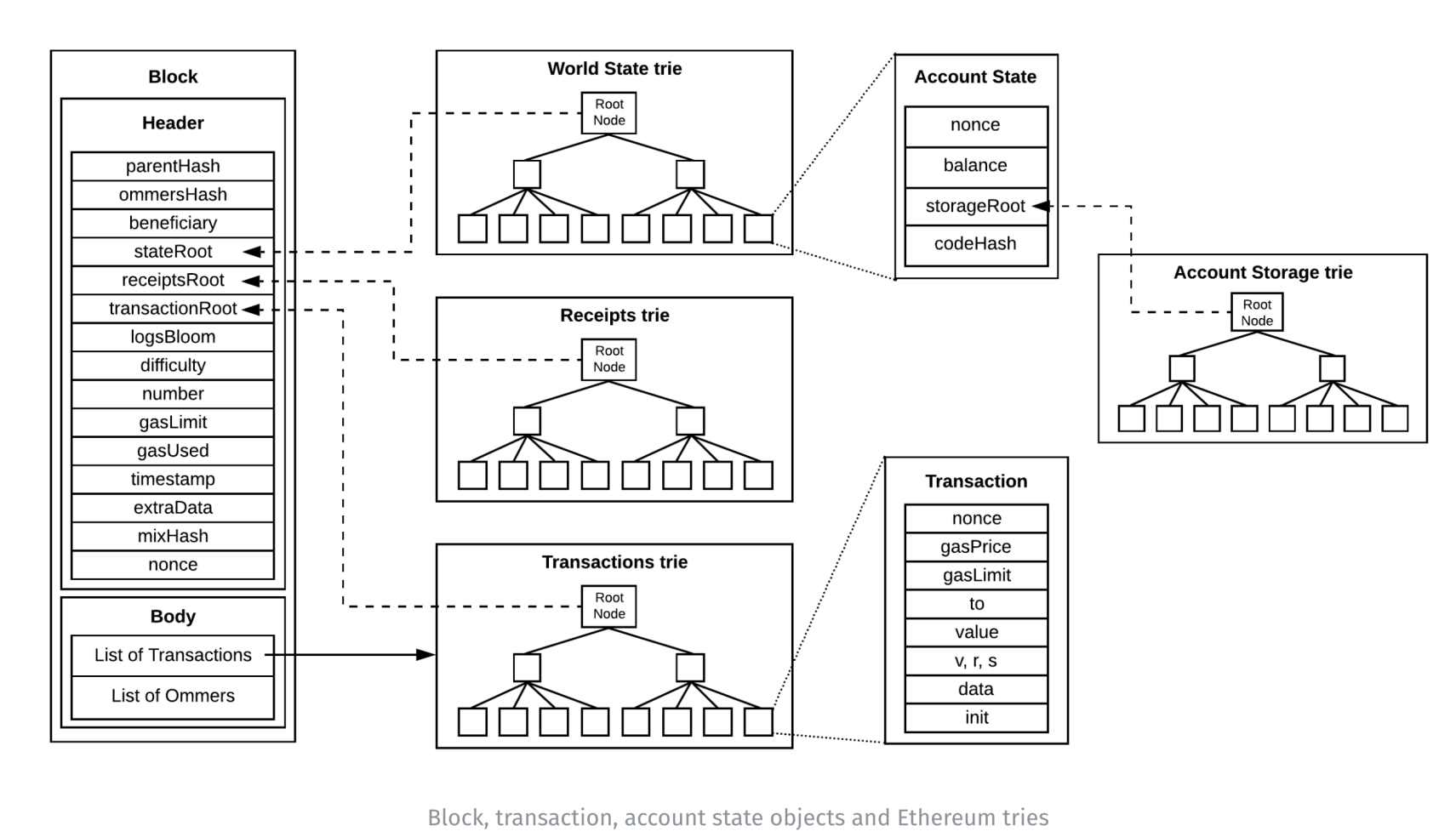
Ethereum gas and gas price

All the money spent on gas by the sender is sent to the miner's address.



From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum block



Block, transaction, account state objects and Ethereum tries

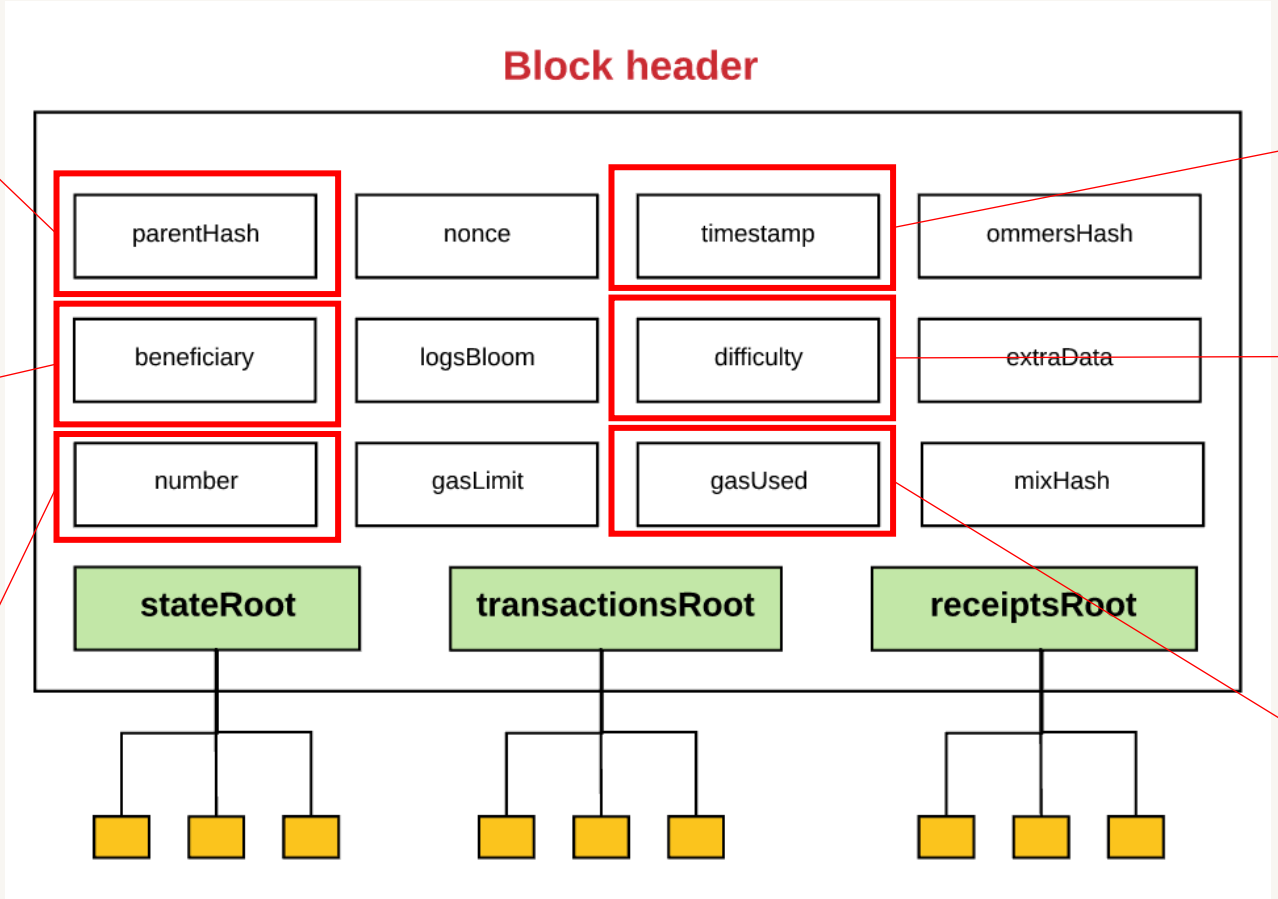
https://miro.medium.com/max/4800/1*BVvVSUIj4A9fmmVBNs4WhQ.png

Ethereum block header

A hash of the parent block's header

The account address that receives the fees for mining this block

The count of the current block



The time stamp of this block's inception

The difficulty level of this block

The sum of the total gas used by transactions in this block

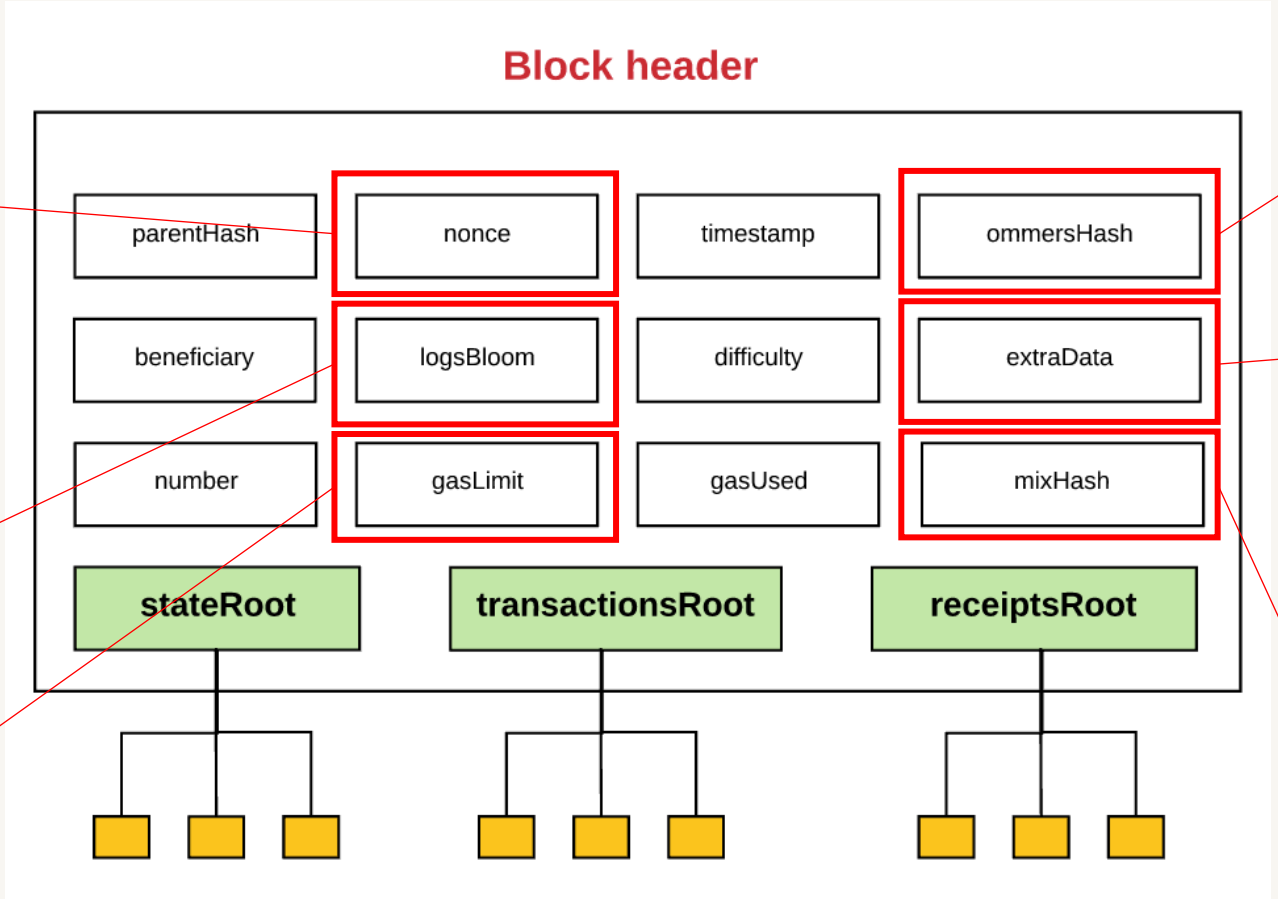
From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum block header

A value that, when combined with the mixHash, proves that this block has carried out enough computation

A Bloom Filter(data structure) that consists of log information

The current gas limit per block



A hash of the current block's list of ommers

Extra data related to this block

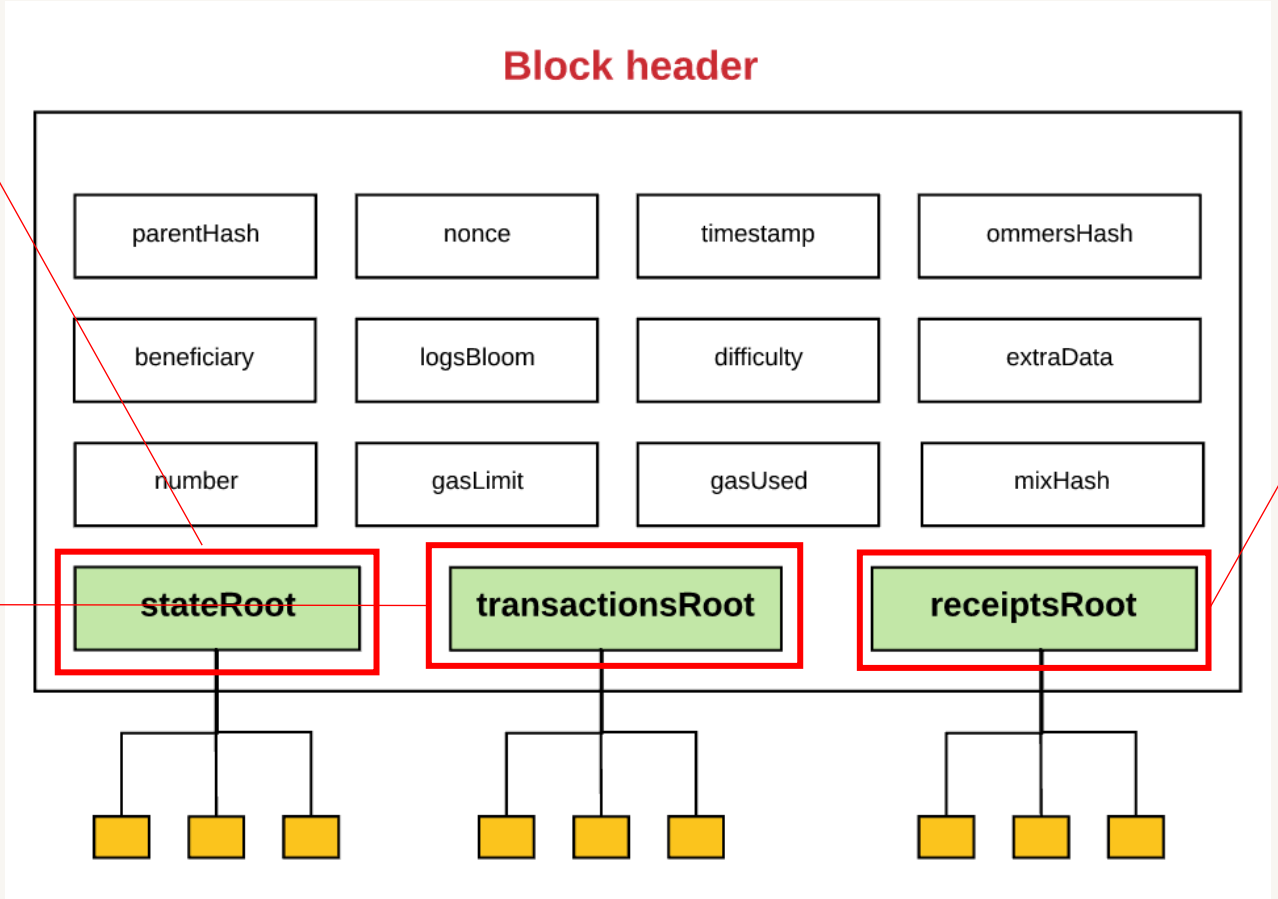
A hash that, when combined with the nonce, proves that this block has carried out enough computation

From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum block header

The hash of the root node of the state tree

The hash of the root node of the tree that contains all transactions listed in this block



The hash of the root node of the tree that contains the receipts of all transactions listed in this block

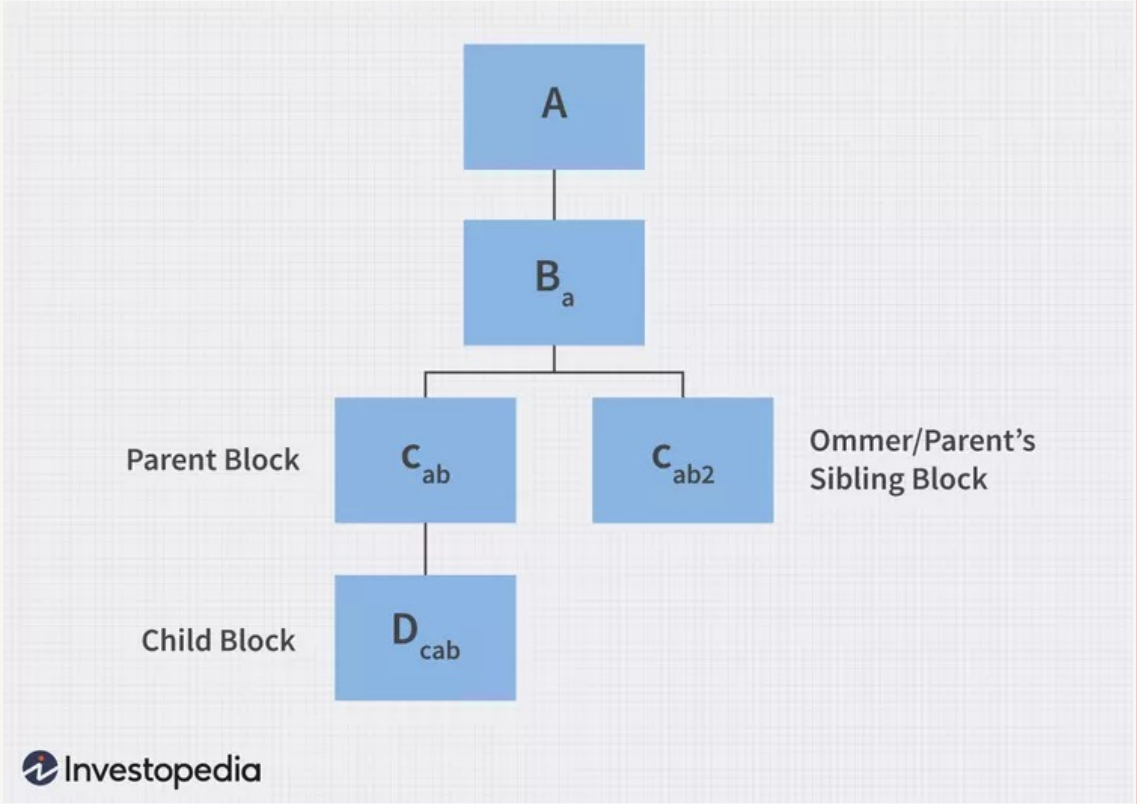
Ethereum ommers

- It is possible for two blocks to be created simultaneously by a network
- When this happens, a fork happens and eventually one block is left out
- This leftover block is called an ommer block
- In the past, they were called uncle blocks
  - referring to the familial relationships used to describe block positions within a blockchain
- In Bitcoin, there is no reward for this omner block
  - Ethereum provides a minimum amount of reward to the omner miner

https://www.investopedia.com/terms/u/uncle-block-cryptocurrency.asp

Ethereum ommers

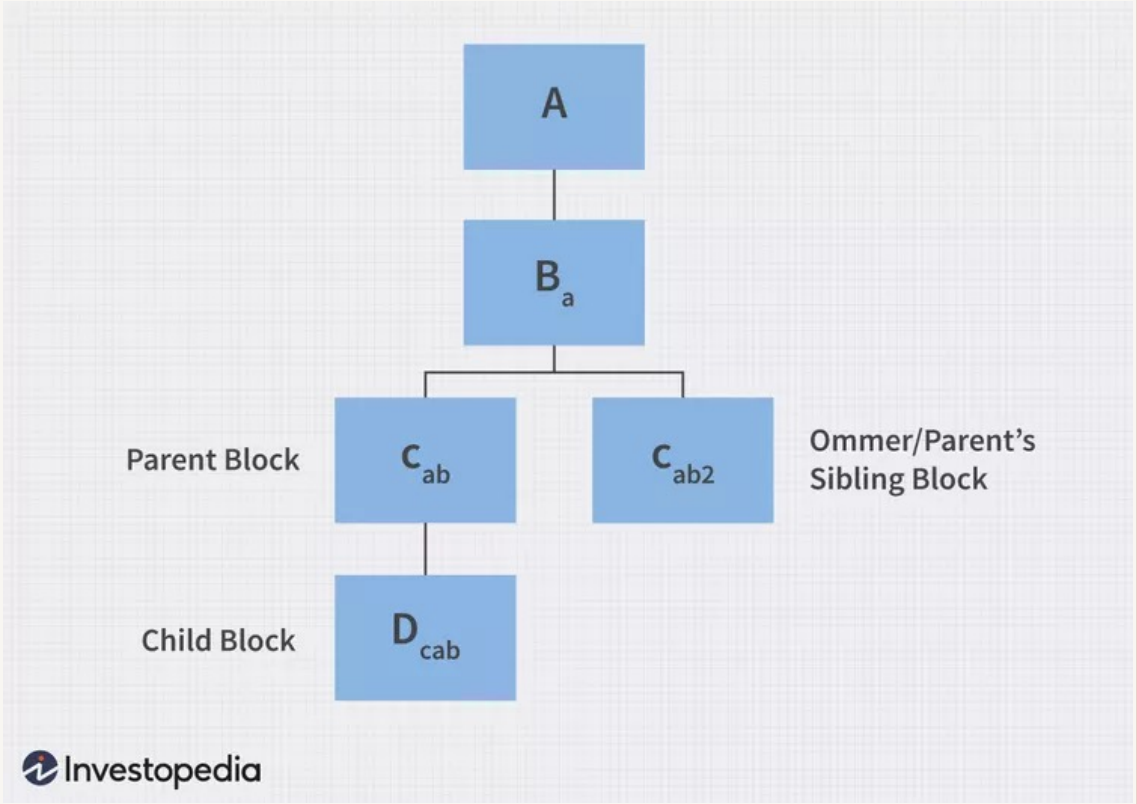
- An ommer is a block whose parent is equal to the current block's parent's parent
- Block times in Ethereum are around 15 sec
  - This is much lower than that in Bitcoin (10 min)
- This enables faster transaction



https://www.investopedia.com/thmb/Bf315VzUb4nwViDSK36Piv55D1Q=/750x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/INV-uncle-block-cryptocurrency-fdc913eee3bb40aebba7e499bfceeada.jpg

Ethereum ommers

- But there are more competing blocks, hence a higher number of orphaned blocks
- The purpose of ommers is to help reward miners for including these orphaned blocks
  - Compensating the miners for their computation

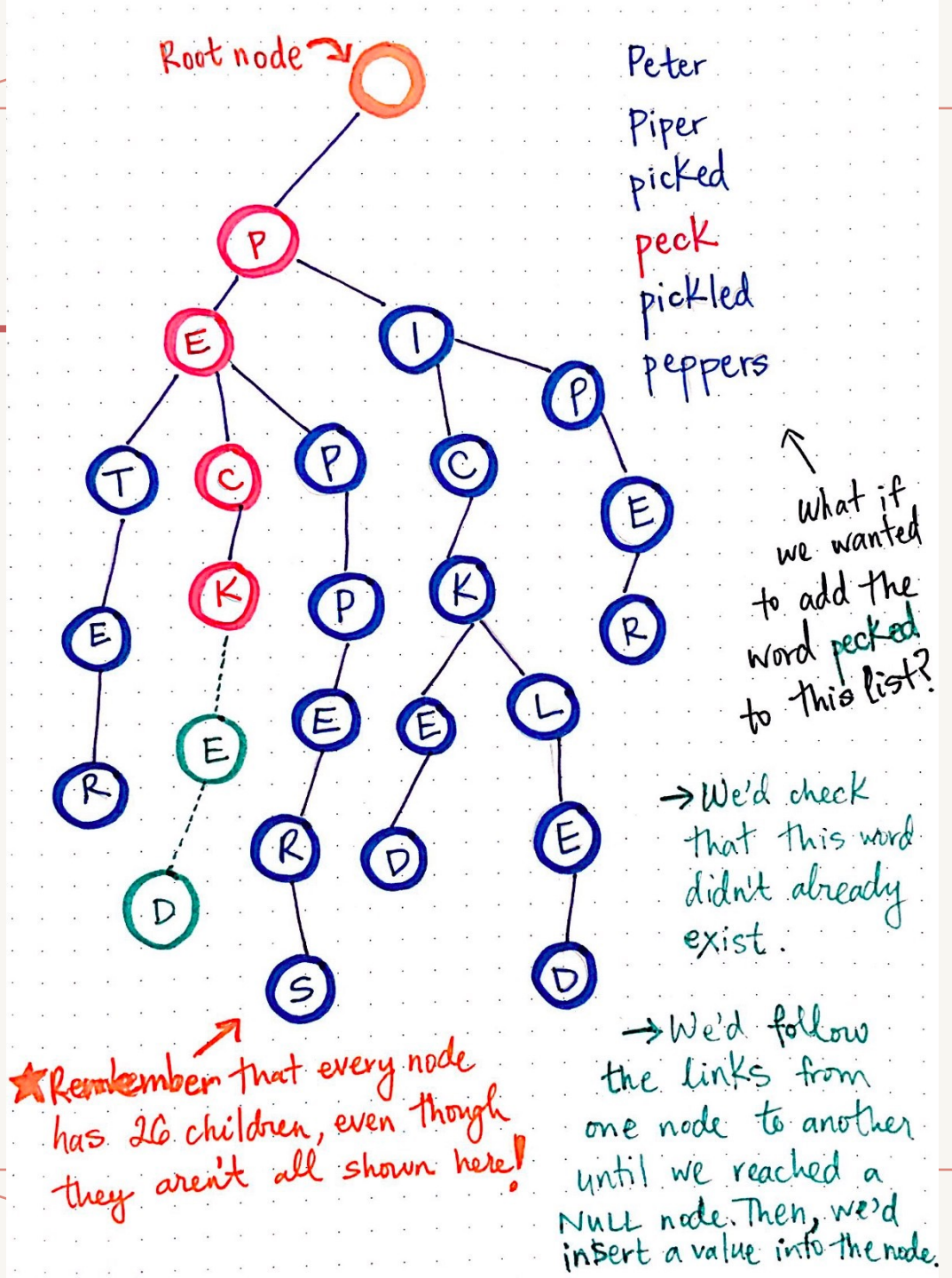


https://www.investopedia.com/thmb/Bf315VzUb4nwViDSK36Piv55D1Q=/750x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/INV-uncle-block-cryptocurrency-fdc913eee3bb40aebba7e499bfceeada.jpg

Trie

A **trie** is a tree-like data structure wherein the nodes of the tree store the entire alphabet, and strings/words can be retrieved by traversing down a branch path of the tree.

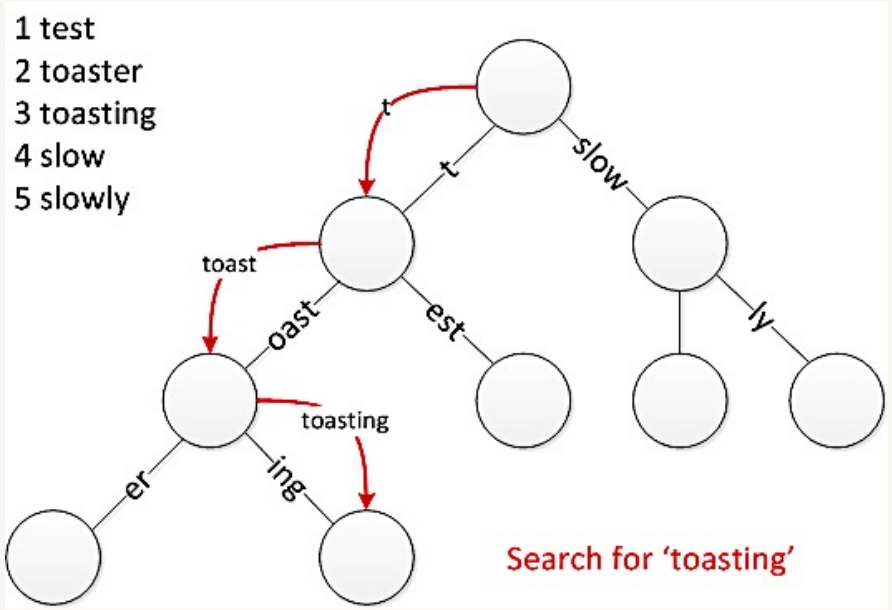
https://cdn-images-1.medium.com/max/1600/1*rkanFIU4G_tmuC939_txhA.jpeg



https://cdn-images-1.medium.com/max/1200/1*sZOrNXzlQICVv5ePpav1-g.jpeg

Merkle Patricia trie

- A Patricia (Practical Algorithm To Retrieve Information Coded In Alphanumeric) trie is a binary radix trie
  - binary choice at each node when traversing the trie
- Modified in Ethereum with the concept of Merkle Patricia trie
  - the root node becomes a cryptographic fingerprint of the entire data structure, just like a Merkle tree



https://i.stack.imgur.com/d2w07.png

Question?