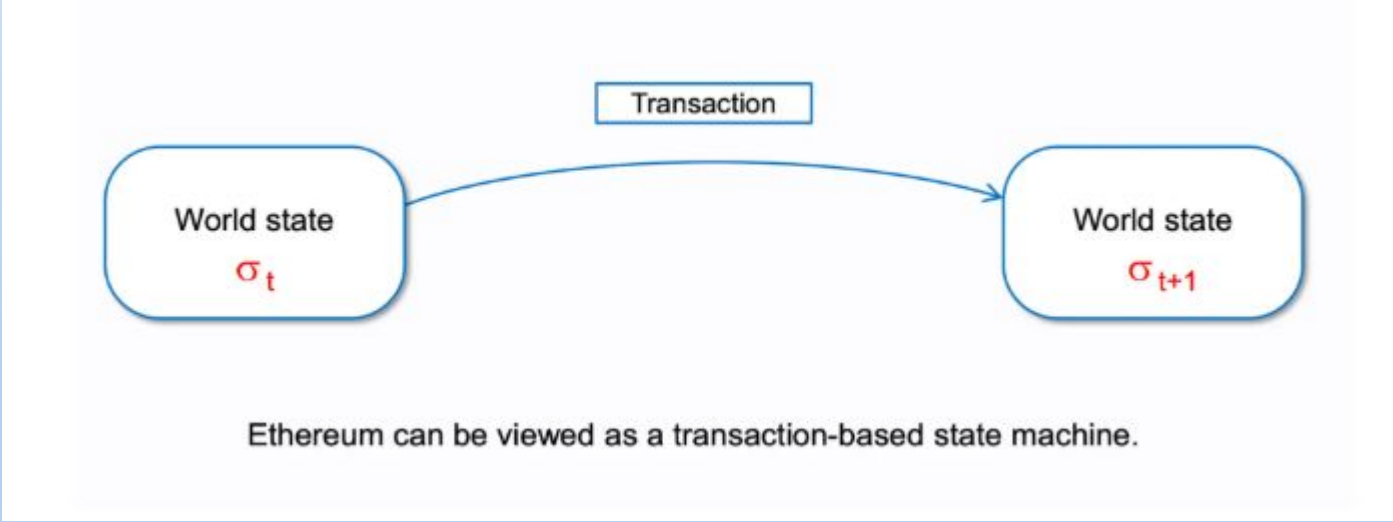
ETHEREUM TRANSACTION

Transactions

- **Ethereum = A Transaction-Based State Machine**
- **The Ethereum computer lives and breathes transactions**. They are the only vehicles that can actually change any state in the computer, as show in the diagram below.

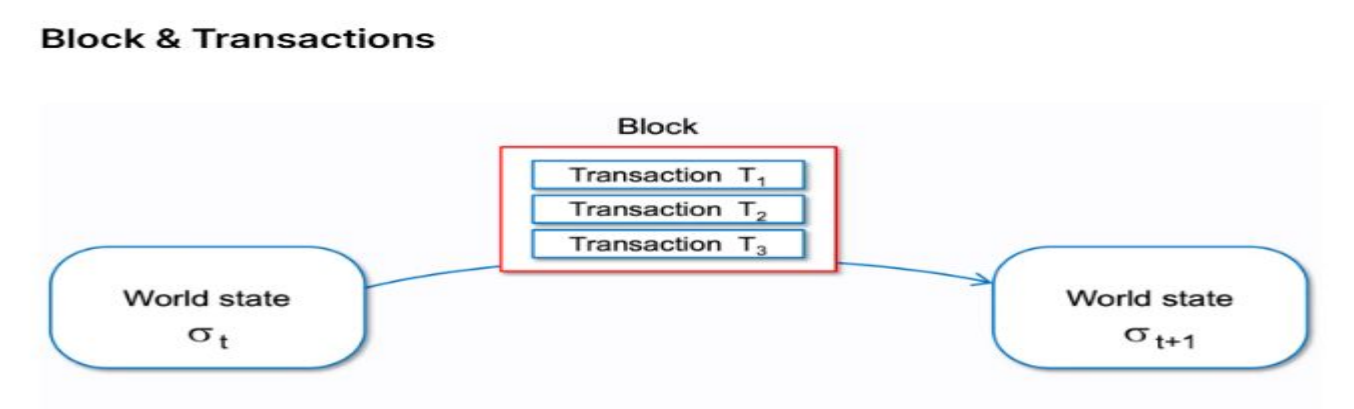


Ethereum can be viewed as a transaction-based state machine.

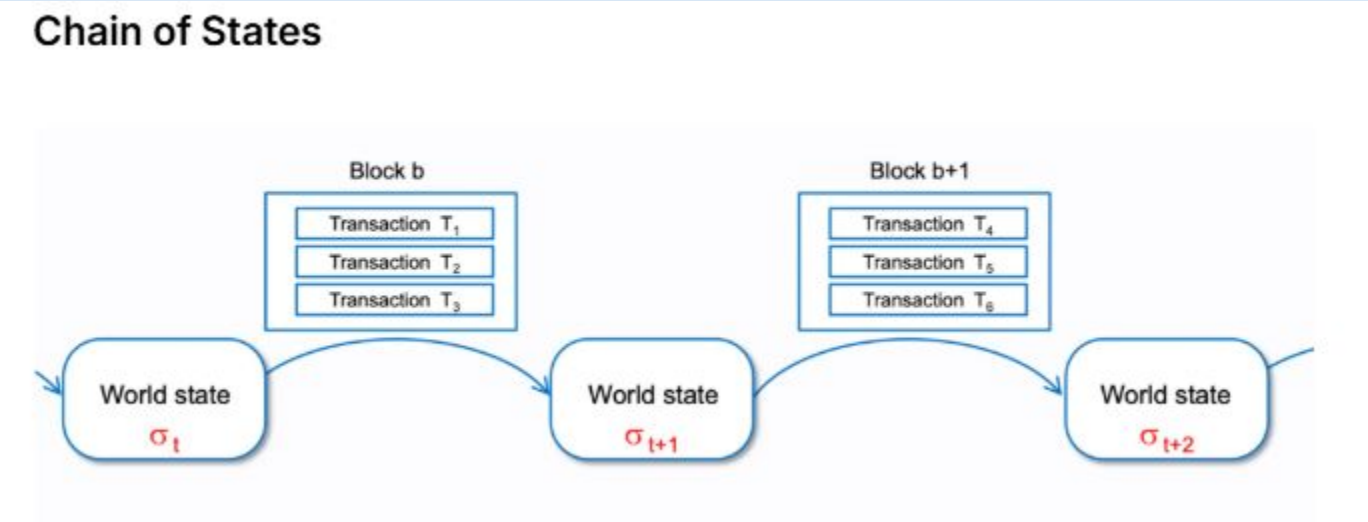
What is a Transaction? (Ethereum)

- An Ethereum transaction refers to an action initiated by an EOA (externally-owned account), in other words an account managed by a human, not a contract.

- For example, if Bob sends Alice **1 ETH**, Bob's account must be debited and Alice's must be credited. **This state-changing action takes place within a transaction.**



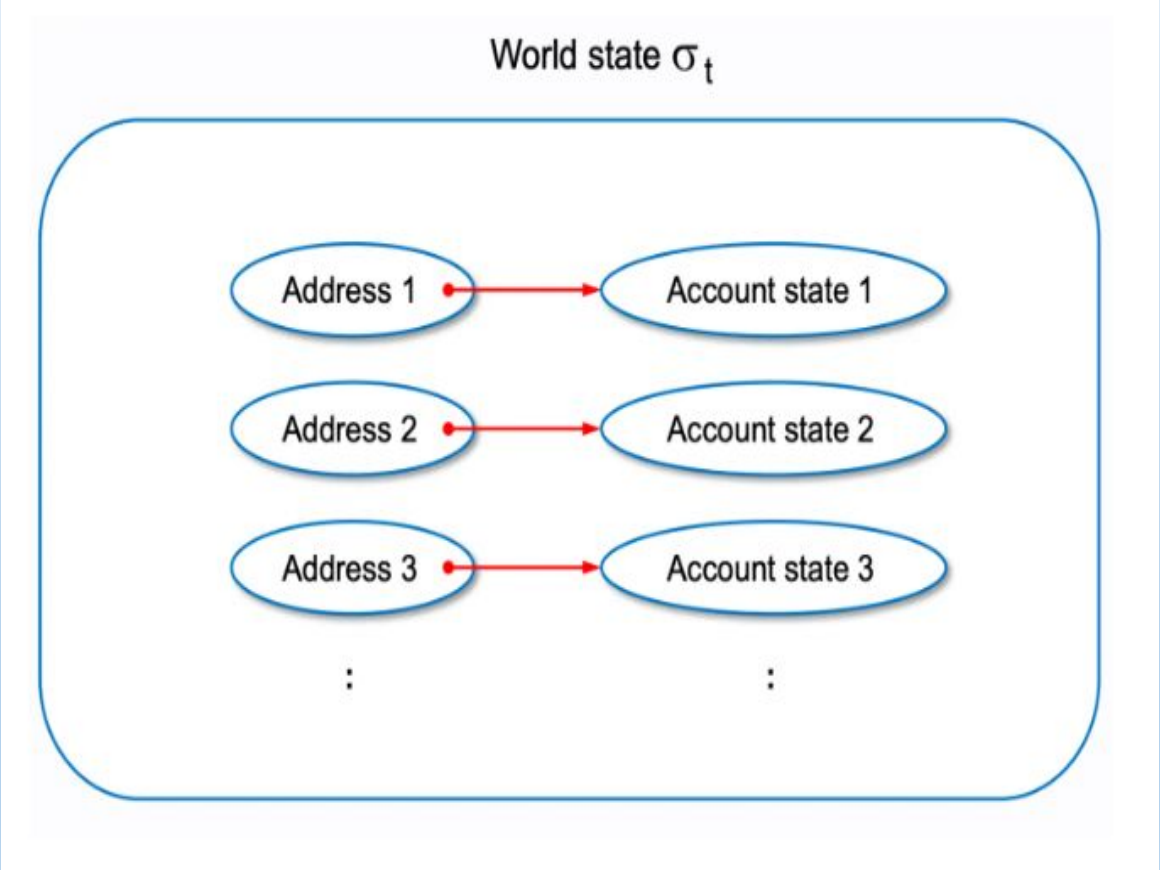
- **Transactions** are collected into blocks. A **block** is a package of data (in the form of transactions.)



- If you focus on how the global singleton world state of Ethereum changes after each block, Ethereum can be seen as a chain of states.

Ethereum World State

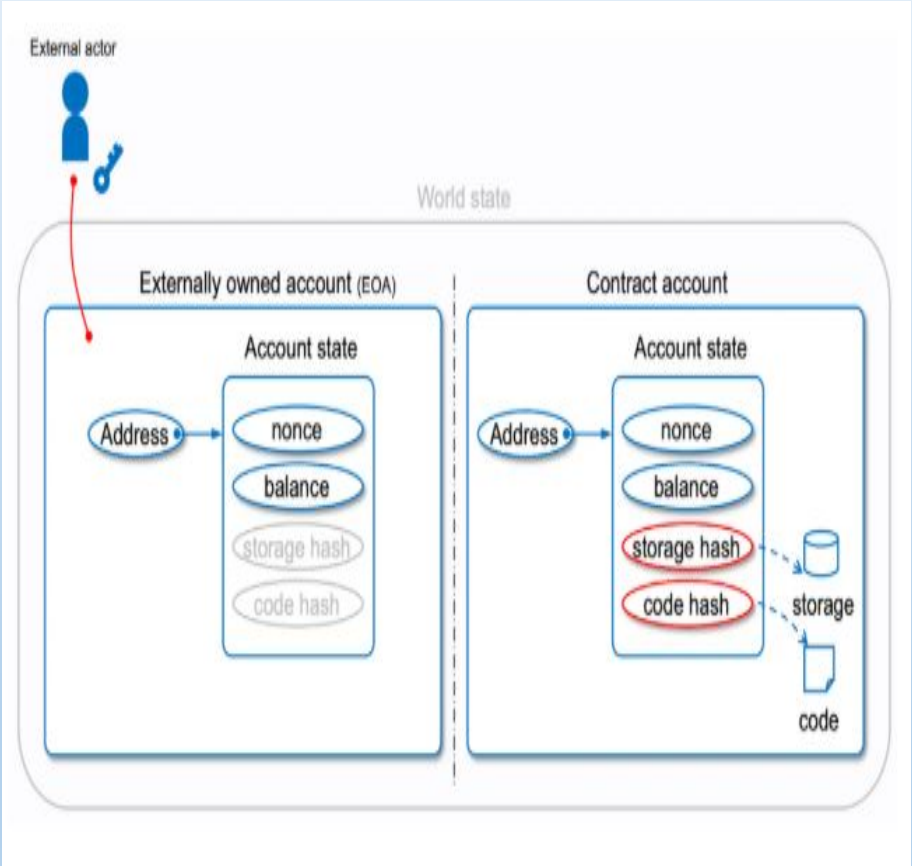
- There is only one single world state and that world state is changed by blocks packed full of data in the form of transactions.
- **Ethereum world state** is simply a mapping between Ethereum addresses and their account state.
- Accounts are simply ledger entries that are indexed via a public address into the world state. Query the world state by providing it an Ethereum address, and the world state will return that address's account state (balance, nonce, smart contract code & state if applicable).



So An Account Can Be A Smart Contract?

There are two types of accounts in Ethereum:

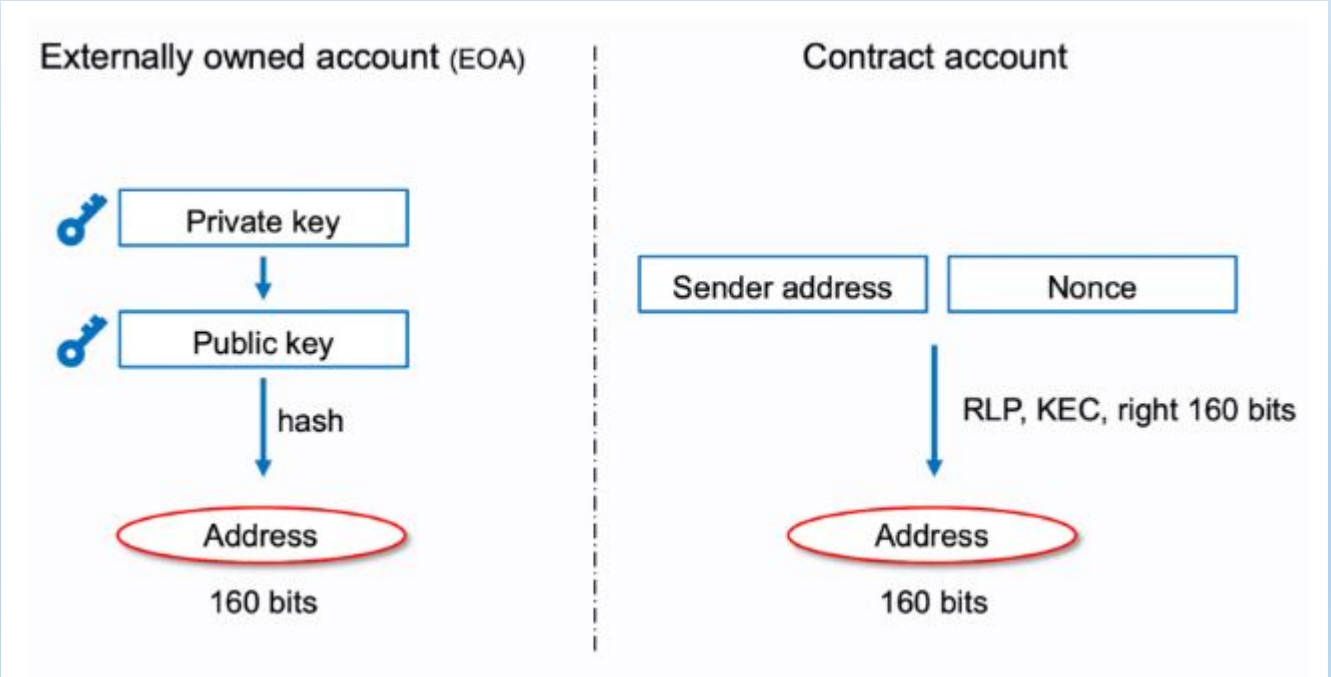
- **EOA**: This is an account directly controlled by a private key
  - An EOA cannot contain EVM code
- **Contract account**: This is an account that does NOT have a private key
  - As seen in the diagram, this account contains two extra properties on its state:
    - **storage hash**: contains the root hash of a Merkle patricia trie that holds any state relevant to this smart contract account (ie. variable values, owners, etc)
    - **code hash**: bytecode representation of skeleton code



How Are The Account Public Addresses Determined?

- If the account is an EOA, the Ethereum public address is derived from the private key.
- If the account is a smart contract, that smart contract public address is derived from the deployer address and the deployer nonce value.

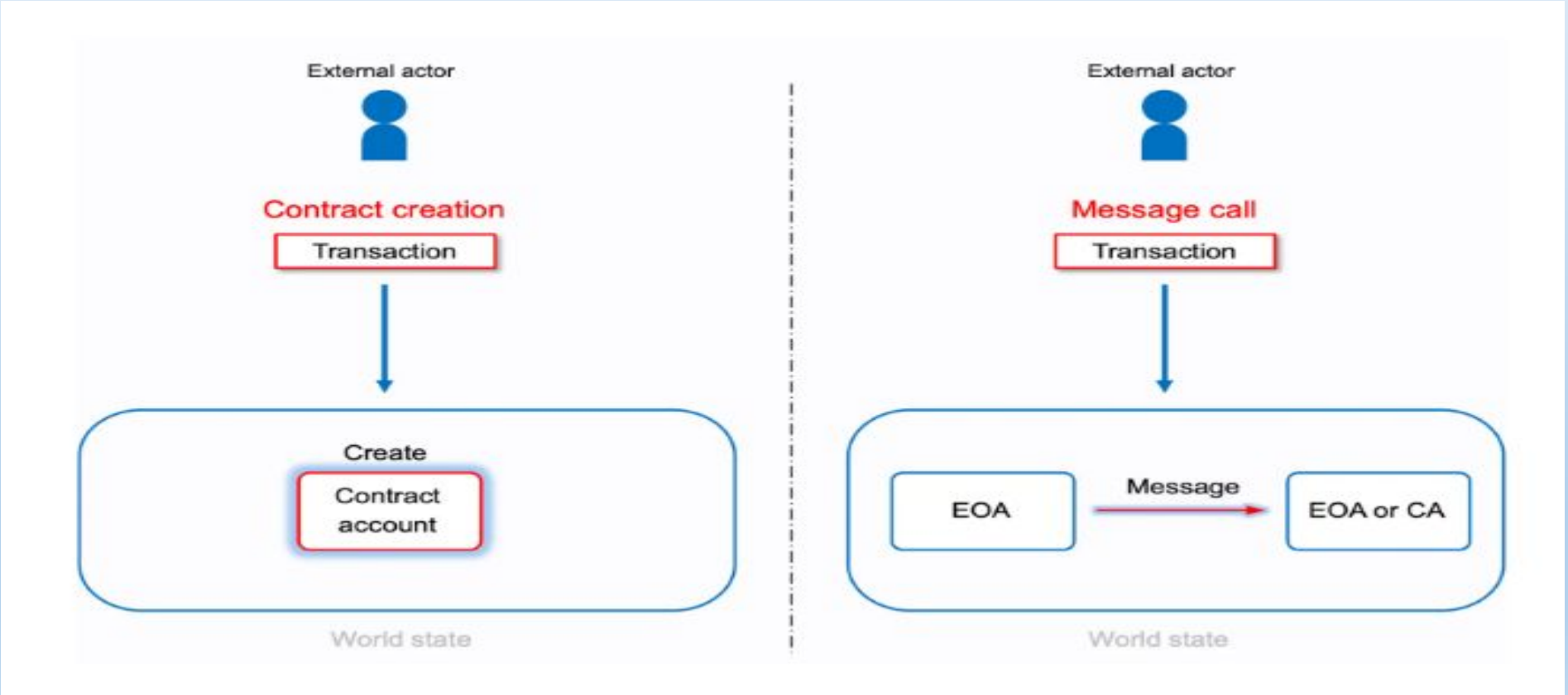
The output, regardless of whether the account is an EOA or a smart contract, is always 160 bits representing the Ethereum public address. You'll typically hear Ethereum public addresses described as 20 bytes long with a 0x appended in front.



TRANSACTION

- A **transaction** is a single cryptographically-signed instruction. It is a signal of intent from an owner of a private key that they want to change the Ethereum state in one way or another.
- Reading data from Ethereum does not require an account! Anyone can ping the Ethereum computer and read data instantly.
- But writing data requires you own a private key and some ETH (to pay for gas!)… all write operations cost gas and so you need ETH to pay for that gas. And all write operations must be signed by a private key!

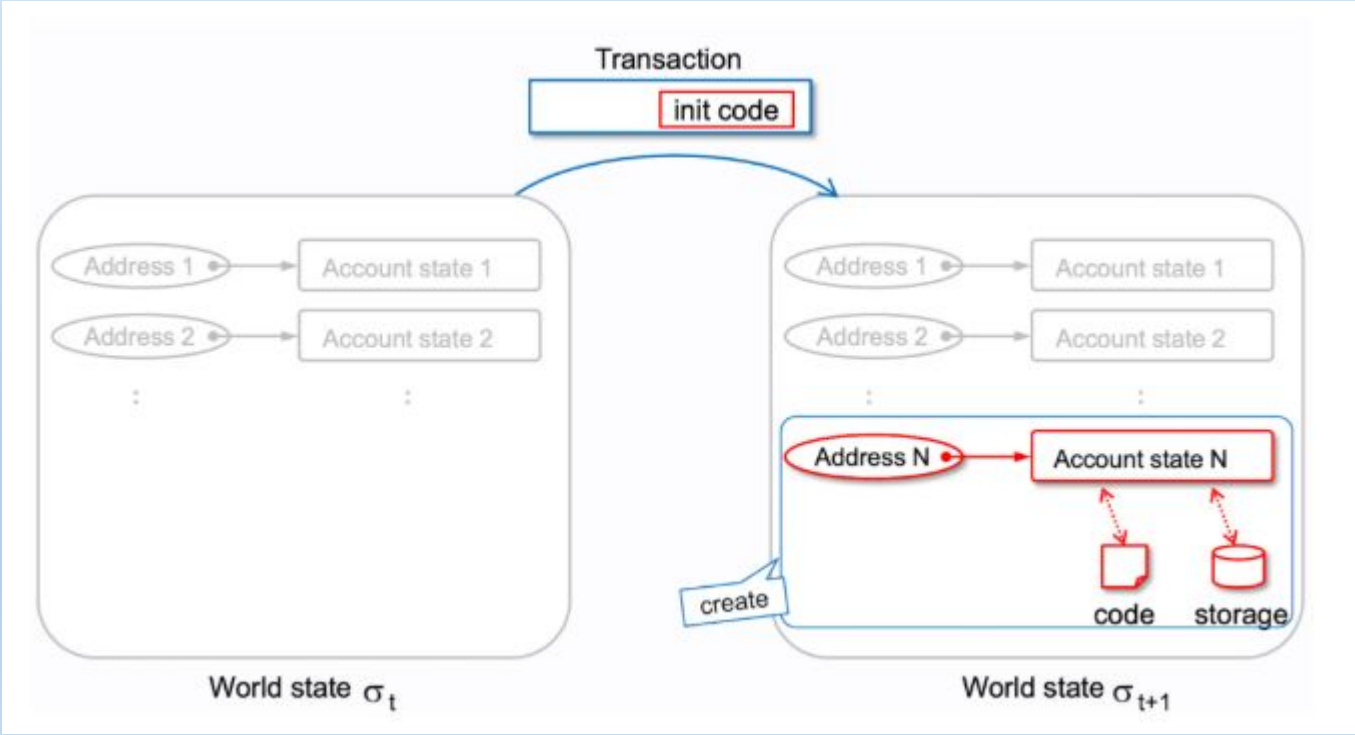
Two Types of Transactions in Ethereum



Types of Transactions(Cont..)

- **1. Contract creation: a special type of transaction that deploys a brand new smart contract**

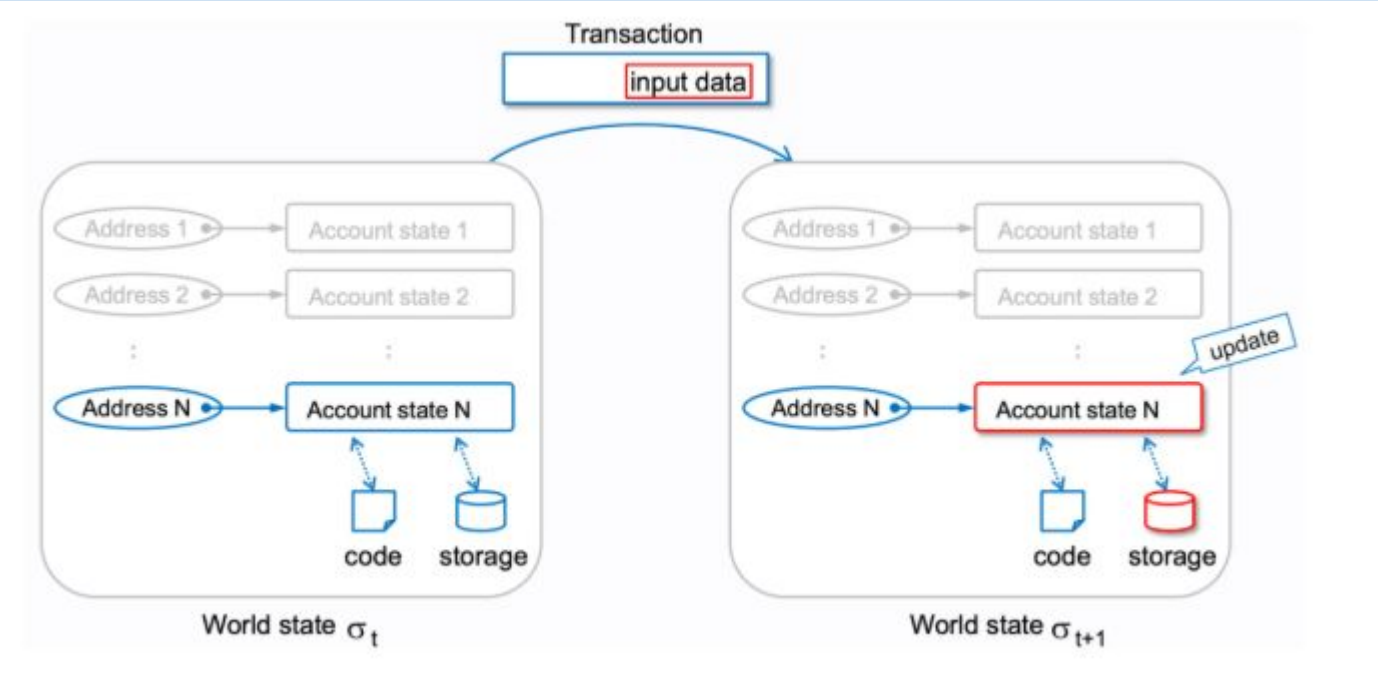
  This transaction essentially *creates* a brand new entry in the Ethereum world state



Types of Transactions(Cont..)

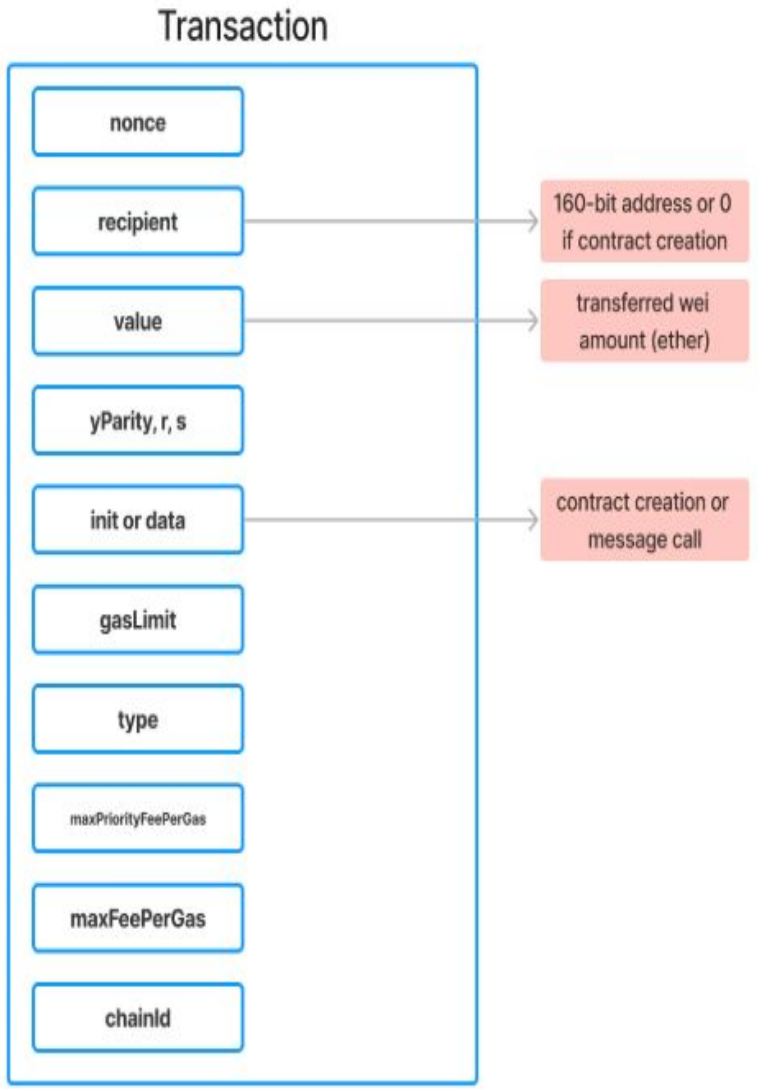
- **2. Message call: a transaction initiated by an EOA that interacts with either another EOA or a smart contract**

  This transaction does NOT create a new entry in the world state, it just *updates* an existing entry in the Ethereum world state.



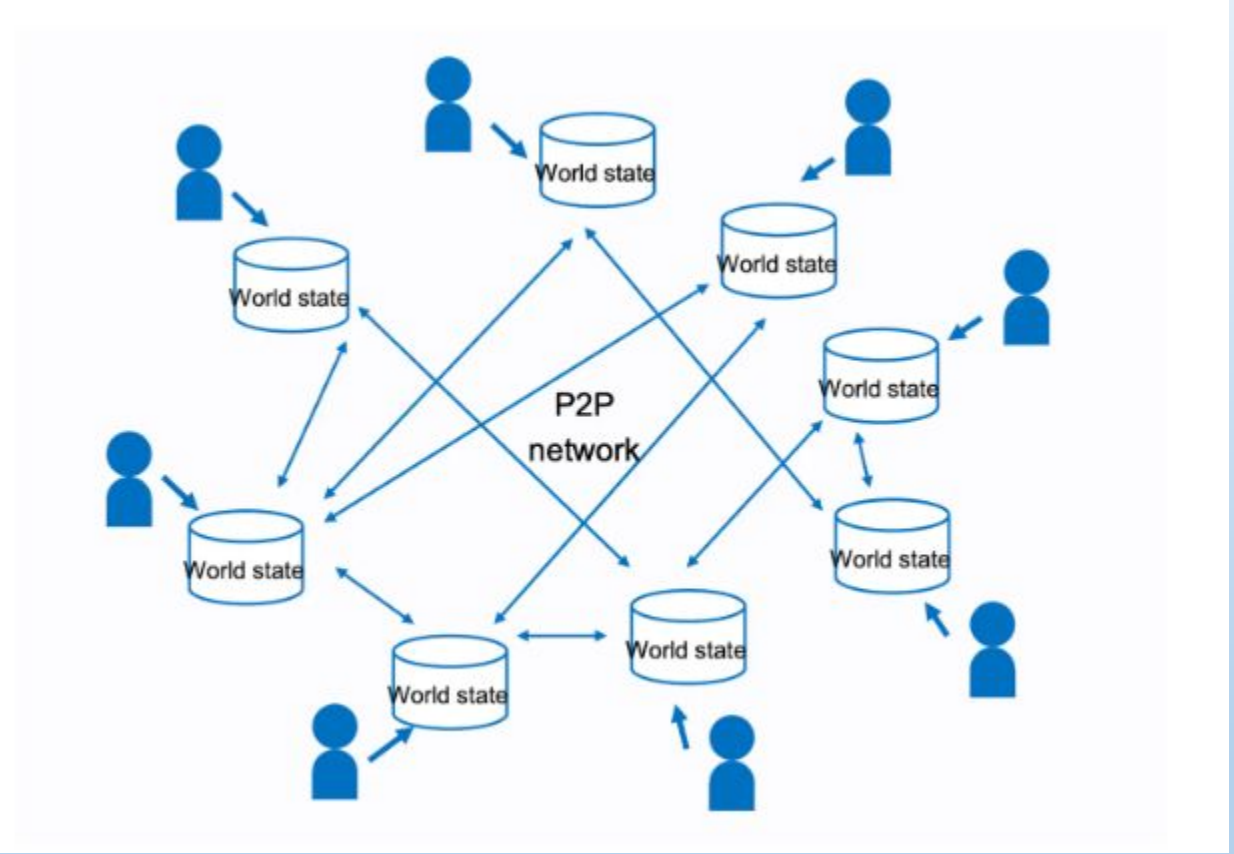
Ethereum Transaction Architecture

- `nonce`: index, gets incremented every time transaction gets mined
- `recipient`: the receiving address (if an externally-owned account, the transaction will transfer value. If a contract account, the transaction will execute the contract code)
- `value`: amount of ETH to transfer from sender to recipient (in WEI, a denomination of ETH)
- `yParity, r, s` (aka: digital signature): signature components
- `init or data`: typically referred to as "calldata", `0` if just a typical ETH transfer
- `gasLimit`: maximum amount of gas units that can be consumed
- `type`: type `0` for legacy (pre-EIP-1559) or type `2` for EIP-1559-compatible txs
- `maxPriorityFeePerGas` (aka: minerTip): the maximum amount of gas to be included as a tip to the validator
- `maxFeePerGas`: the maximum amount of gas willing to be paid for the transaction (inclusive of `baseFeePerGas` and `maxPriorityFeePerGas`)
- `chainId`: in order to protect against replay attacks on other EVM chains, each transaction must now include a specific id per chain. Mainnet is `0`. Göerli is `5`. You can check other chain ids here: https://chainlist.org/

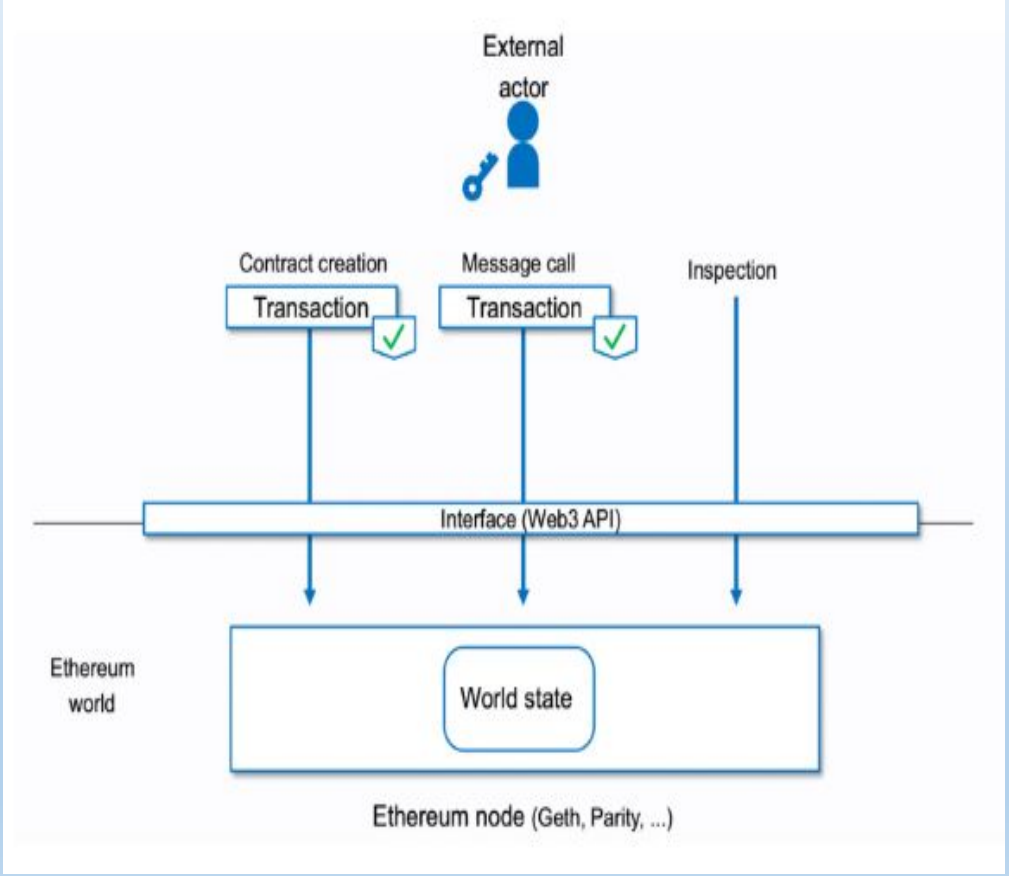


Network

- Everyone keeps a copy of the latest Ethereum world state.
- When a new block packed full of state-changing transactions gets mined, the block is independently verified by every node in the peer-to-peer network.
- If the block is truthful (no double spending transactions, all digital signatures check out, etc) then the node adds that block to their own local version of the blockchain as the latest representation of the Ethereum world state.
- This happens every time a block is mined: worldState n transaction to n+1.



- The Ethereum network is made up of nodes decentralized all over the world. Each of these nodes performs verification on any new block of transactions.
- We've learned that in order to interact with one of these nodes constituting the Ethereum P2P network, one must send a JSON-RPC request (in order to read data) or send a *signed* JSON-RPC request (in order to write data, which means you are changing some state in the Ethereum computer).



As an EOA, you have three routes to interact with the Ethereum computer via a connection to an Ethereum node:

- **Contract creation**: As an EOA, you deploy a new smart contract to the Ethereum computer *via a special transaction* (signed JSON-RPC request)
- **Message call**: As an EOA, you either send some ETH to another EOA or interact with a smart contract in some way *via a transaction* (signed JSON-RPC request)
- **Inspection**: Any user can make read queries to any Ethereum nodes, no account needed. Try out the eth_getbalance method.

Transaction Object Example

1. Alice sends Bob 1 ETH

```json
{
  to: "0x2c8645BFE28BEEb6E19843eE9573b7539DD5B530", // Bob
  gasLimit: "21000",
  maxFeePerGas: "30", // 28 (base) + 2 (priorityFee)
  maxPriorityFeePerGas: "2", // minerTip
  nonce: "0",
  value: "10000000000000000", // 1 ether worth of wei
  data: '0x', // no data, we are not interacting with a contract
  type: 2, // this is not a legacy tx
  chainId: 4, // this is AU, we deal only in test networks! (Göerli)
}
```

2.Alice calls a function on a smart contract

```json
{
  to: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8", // smart contract address
  gasLimit: "36000",
  maxFeePerGas: "30", // 28 (base) + 2 (priorityFee)
  maxPriorityFeePerGas: "2", // minerTip
  nonce: "1", // this is Alice's second transaction, so the nonce has increase
  value: "10000000000000000", // 1 ether worth of wei
  data: '0x7362377b0000000000000000000000000000000000000000000000000000000000000000
  type: 2, // this is not a legacy tx
  chainId: 4, // this is AU, we deal only in test networks! (Göerli)
}
```

How To Manually Construct Calldata

- Once we send a transaction that points to a smart contract, how does the contract know what specific function you intend to call? Well, all those specifics end up going in the data field of each transaction.
- Here is the algorithm to manually construct calldata:

1. Say Alice wants to call the withdrawEther() function of a faucet smart contract…
2. Alice must take the **keccak256** hash of that function signature:

3. Take the first 4 bytes (8 characters) of the hash output, which is just:7362377b

4. This function takes no arguments, so no need to append any parameter data

5. If the function took arguments, you would need to hash the entire function signature with that parameter type, for example: helloWorld(uint256)

6. Final calldata construction, padded out to 32 bytes:

```
0x7362377b0000000000000000000000000000000000000000000000000000000000000000
```

Gas

- Gas is a measurement of the cost to each operation that relates to the computational cost that the operation incurs on the network.
- So if you are making every node in the network do some kind of computationally expensive task every time they need to verify your transaction, you'll need to pay for significantly more than a simple transaction that is sending money from one individual to another.
- The Ethereum Virtual Machine has a list of operation codes with a corresponding gas cost.

List of operation codes with a corresponding gas cost.

Table

Opcode	Name	Description	Extra Info	Gas
`0x00`	STOP	Halts execution	-	0
`0x01`	ADD	Addition operation	-	3
`0x02`	MUL	Multiplication operation	-	5
`0x03`	SUB	Subtraction operation	-	3
`0x04`	DIV	Integer division operation	-	5
`0x05`	SDIV	Signed integer division operation (truncated)	-	5
`0x06`	MOD	Modulo remainder operation	-	5
`0x07`	SMOD	Signed modulo remainder operation	-	5
`0x08`	ADDMOD	Modulo addition operation	-	8
`0x09`	MULMOD	Modulo multiplication operation	-	8
`0x0a`	EXP	Exponential operation	-	10*
`0x0b`	SIGNEXTEND	Extend length of two's complement signed integer	-	5

GAS COST

We can split these operations up into several categories:

- Arithmetic (i.e. **ADD**, **DIV**, etc.)
- Information about the current context of the transaction (i.e. **TIMESTAMP** , **CALLVALUE**, etc.)
- Operations that manipulate/retrieve from **temporary memory** (i.e. **MSTORE**, **PUSH32**, etc.)
- Operations that manipulate/retrieve from **persistent memory** (i.e. **SSTORE**, **CREATE**, etc.)
- Control Flow Operations that provide us with **loops** (i.e. **JUMP**, **JUMPI**, etc.)

For example **ADD** requires **3** gas, while using **SSTORE** can require **20000** gas.

Even the operation **BALANCE** has significant costs associated with it (**700** gas) because it requires a lookup in persistent memory.

Denominations of Ether

- Ether has different denominations that are used to express smaller values, particularly when describing gas costs.
  - For example, similar to how 1 dollar is equal to 100 pennies, 1 ether is equal to 10^18 Wei (the smallest denomination of Ether) or 10^9 Gwei.

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

Example 1 :

- Rahul needs to give 1 ETH to Shubham. During the transaction, the gas boundary is 21000 units, and the gas price is 200 gwei.

*Total fee costs = Gas units (limit) * Gas price per unit*
*= 21000 * 200*
= 4,200,000 gwei (0.0042 ETH)

- When Rahul dispatched the funds, 1.0042 ETH would be subtracted from Rahul's account.
- Shubham would be credited 1.0000 ETH only as 0.0042 is the fess of Miner.
- Now **after upgradation**, increased advantages presented by the difference contain more profitable trade fee computation, typically more rapid trade inclusion, and canceling the ETH distribution by burning a portion of trade fees.
  - *Total fee after upgradation = Gas units (limit) * (Base fee + Tip)*

Example

- Shubham has to pay 1 ETH to Rahul. This process has a gas limit of 21000 units and a base fee of 100 gwei. Shubham includes a tip of 10 gwei.

*Total fee after upgradation = Gas units (limit) * (Base fee + Tip)*
*= 21000 * (100 + 10)*
= 2,310,000 gwei (0.00231 ETH).

- When Shubham sends the funds, 1.00231 ETH will be subtracted from Shubham's account.
- Rahul will be credited 1.0000 ETH and 0.00021 ETH will be received by the miner as the tip.
- A ground fee of 0.0021 ETH is ignited.

Gas Fees

- The gas fees include Base, Priority, Max, and calculating fees

1. Base fees:

- Each alliance has a ground fee which serves as a fund price.
- The ground fee is figured alone of the existing alliance and is rather specified by the alliances before it pushes trade prices better for users.
- When the block is excavated the base fee is "burned", dragging it from regulation.
- The ground fee is evaluated by instructions that compare the length of the earlier block with the marked size.
- The base fee intention rise by a max of 12.5% per block if the marked block size is surpassed.

Gas Fees

2. Priority fee:

- Apart from the base fee obtained, the advancement presented a priority fee i.e. tip to miners to contain a trade in the alliance.
- Without tips, miners can see it financially possible to drill cleared blocks because they can obtain the exact alliance prize.
- Under normal circumstances, a little tip provides miners with the slightest encouragement to maintain a transaction.
- For dealings that require getting preferentially conducted ahead of different trades in the same block, a more elevated tip will be essential to endeavor to outbid contending trades.

3. Max fee:

- To conduct trade on the P2P network, users can select the greatest limit they are inclined to spend for their dealing to be completed.
- This additional circumference is called the max fee/Gas.
- For a trade to be completed, the max fee must surpass the aggregate of the base and priority fees.
- The trade sender is repaid the contrast between the max fee and the total of the ground(base) fee and tip.

Gas Fees

4. Calculating fees:

- The main benefit of the upgrade is enhancing the user's knowledge when charging trade fees.
- In the wallets that sustain the upgrade rather than explicitly commenting to pay the transaction from a wallet, providers will automatically set a suggested transaction fee (base fee + suggested priority fee) to decrease the quantity of complexness loaded on their users.

Why gas fees can go high?

- Increased gas prices exist because of the fame of Ethereum.
- Conducting any function on Ethereum needs depleting gas, and the gas area is determined per alliance.
- Prices contain measures, holding or exploiting data, and swallowing various parts of "gas" units.
- As app functionality produces additional complexity, the digit of functions a wise contract achieves also develops, representing each trade brings up more additional area of a fixed measure block.
- If the demand is quite high, users must present a more elevated tip payment to try and outbid other users' dealings.
- A more increased tip can construct it additional possible that your trade will convey into the subsequent block.

Why do gas fees exist?

- Gas prices assist maintain the Ethereum grid safe. By demanding a price for every analysis performed on the grid, it controls harmful attackers from spamming the grid.
- To bypass unexpected code, every trade needs to set a limitation to the multiple computational efforts of regulation performance.
- The absolute unit of analysis is "gas".
- Moreover, a trade contains a limitation, any gas not operated in a transaction is produced to the user (i.e. max fee - (base fee + tip) is produced).

TOKENS

- In **Ethereum**, tokens are **digital assets** built on top of the Ethereum blockchain, using smart contracts. Unlike **Ether (ETH)** (which is Ethereum's native cryptocurrency), tokens are created and managed by developers for specific purposes.

Tokens can represent:

- **Cryptocurrencies / Digital Money** → e.g., stablecoins like USDT, DAI
- **Utility** → access to a service or platform feature (e.g., Basic Attention Token (BAT) for Brave browser)
- **Governance Rights** → voting power in decentralized protocols (e.g., UNI for Uniswap)
- **Assets** → tokenized stocks, real estate, gold, or NFTs
- **Certificates / Identity** → proof of membership, credentials, or reputation

Tokens Work in Ethereum

Tokens are **smart contracts** written mainly in **Solidity**.

The most common standards are:

- **ERC-20** → Fungible tokens (each token is identical, like money).
- **ERC-721** → Non-Fungible Tokens (NFTs, unique digital items).
- **ERC-1155** → Hybrid standard supporting both fungible and non-fungible tokens.

Example

- If a company wants to raise funds, they can issue ERC-20 tokens on Ethereum.
  - Investors buy these tokens using ETH.
  - These tokens can later be traded, used for governance or represent shares in the project.

Tokens in Ethereum are **programmable digital assets** created via smart contracts, and they can represent money, property, voting rights or even unique digital collectibles.

- ETH = fuel + money of Ethereum itself
- Tokens = apps, currencies, and assets built using Ethereum

ETH vs Tokens

Feature	**Ether (ETH)**	**Tokens**
Definition	Native crypto currency of the Ethereum blockchain	Digital assets built on top of Ethereum via smart contracts
Standard	No external standard (it's built-in to Ethereum)	Usually follow standards like **ERC-20**, **ERC-721**, **ERC-1155**
Purpose	Used to pay **gas fees** (transaction fees) and secure the network	Used for **payments, governance, stablecoins, NFTs, utility, etc.**
Control	Controlled directly by the Ethereum protocol	Created and managed by **developers / projects** via smart contracts
Supply	Issued by Ethereum protocol, limited by consensus rules (not hard-capped)	Supply depends on each project (e.g., fixed, inflationary, or burnable)
Example	ETH (always the same coin)	USDT (stablecoin), UNI (governance), AAVE (lending), NFTs (unique tokens)
Necessity	Always required to use Ethereum (for gas fees, deploying contracts, transactions)	Optional, depends on which DApps or projects you use

Fungible Vs Non- Fungible Token

Feature	FT (Fungible Token)	NFT (Non-Fungible Token)
Interchangeable	Yes, tokens are identical.	No, each token is unique.
Divisible	Yes (e.g., 0.001 BTC).	No (can’t split an artwork NFT).
Value	Same value for same type.	Each has different value.
Use Cases	Currency, utility tokens.	Digital art, collectibles, property.
Standards	ERC-20 (Ethereum).	ERC-721, ERC-1155 (Ethereum).

Bitcoin vs Ethereum

Ethereum is not just a digital currency

Bitcoin

- Bitcoin is the first decentralised cryptocurrency created in 2009 by an unknown person named Satoshi Nakamoto
- A cryptocurrency and worldwide payment system
- Bitcoin enables peer-to-peer transactions, purchase of goods/services, long-term storage of value

Ethereum

- Ethereum was released in 2015 by a cryptocurrency research and programmer named Vitalik Buterin
- A decentralised programmable platform which supports DApps (distributed applications) running smart contracts and using digital tokens
- Ethereum enables peer-to-peer transactions and can handle accounts, transactions as well store, execute newly coded programming logic

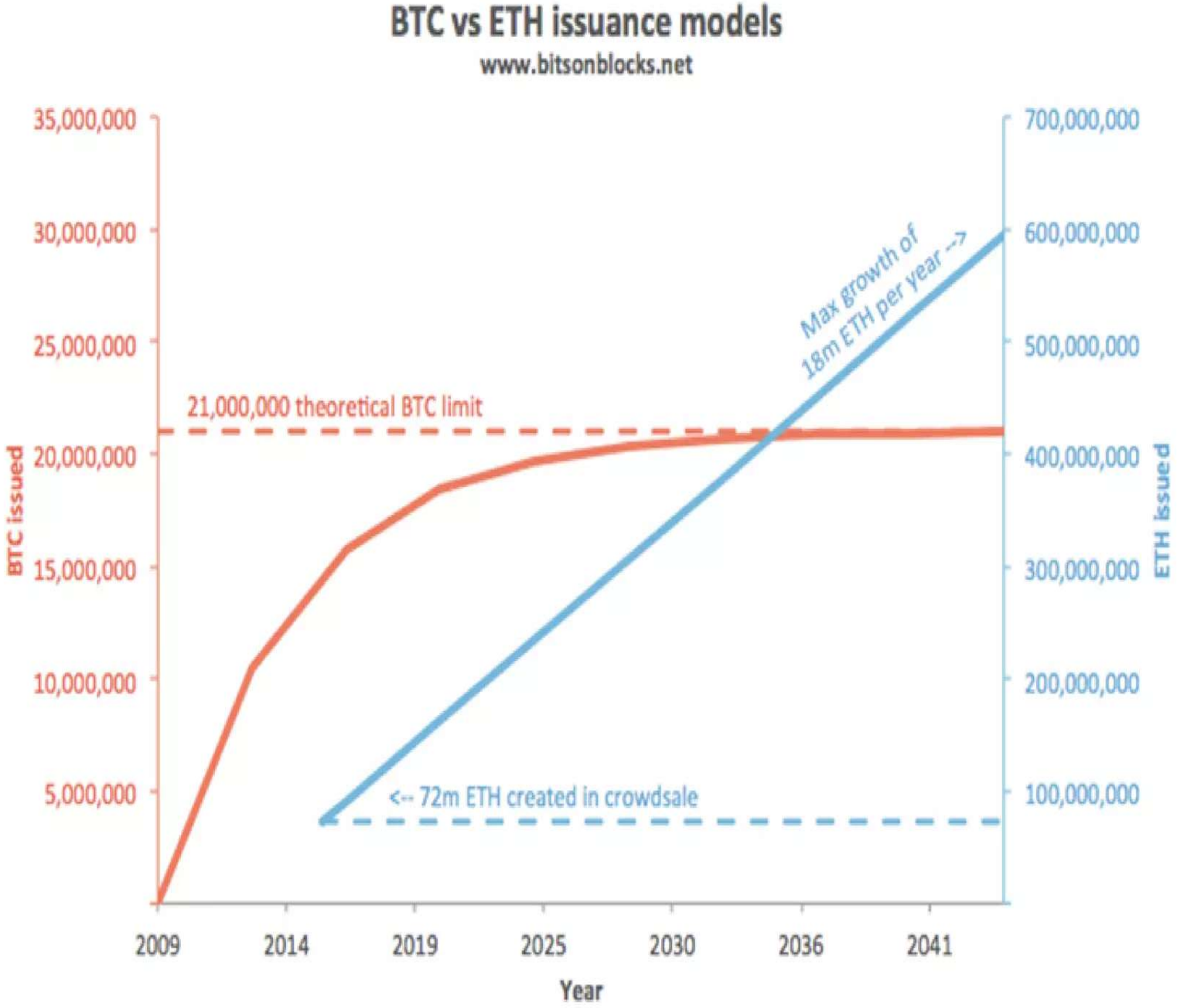
Bitcoin is deflationary by design due to its limited supply

Bitcoin

- Supply cap: 21 million
- Total existing coins: ~18 million BTC
- Market capitalisation: ~65 billion USD
- Smallest unit: 1 Satoshi = 0.00000001 BTC (10^{-8})
- Supply style: deflationary (a finite number of bitcoin will be made)
- Price: US$3,628 as of 2019-1-13

Ethereum

- Supply cap: 18 million every year
- Total existing coins: ~104 million ETH
- Market capitalisation: ~13 billion USD
- Smallest unit: 1 Wei = 0.000000000000000001 ETH (10^{-18})
- Inflationary (much like fiat currency, where more tokens can be made over time)
- Price: US$125 as of 2019-1-13


BTC vs ETH issuance models
www.bitsonblocks.net
35,000,000
30,000,000
25,000,000
20,000,000
15,000,000
10,000,000
5,000,000
BTC issued
700,000,000
600,000,000
500,000,000
400,000,000
300,000,000
200,000,000
100,000,000
ETH issued
21,000,000 theoretical BTC limit
Max growth of
18m ETH per year -->
<-- 72m ETH created in crowdsale
2009
2014
2019
2025
2030
2036
2041
Year

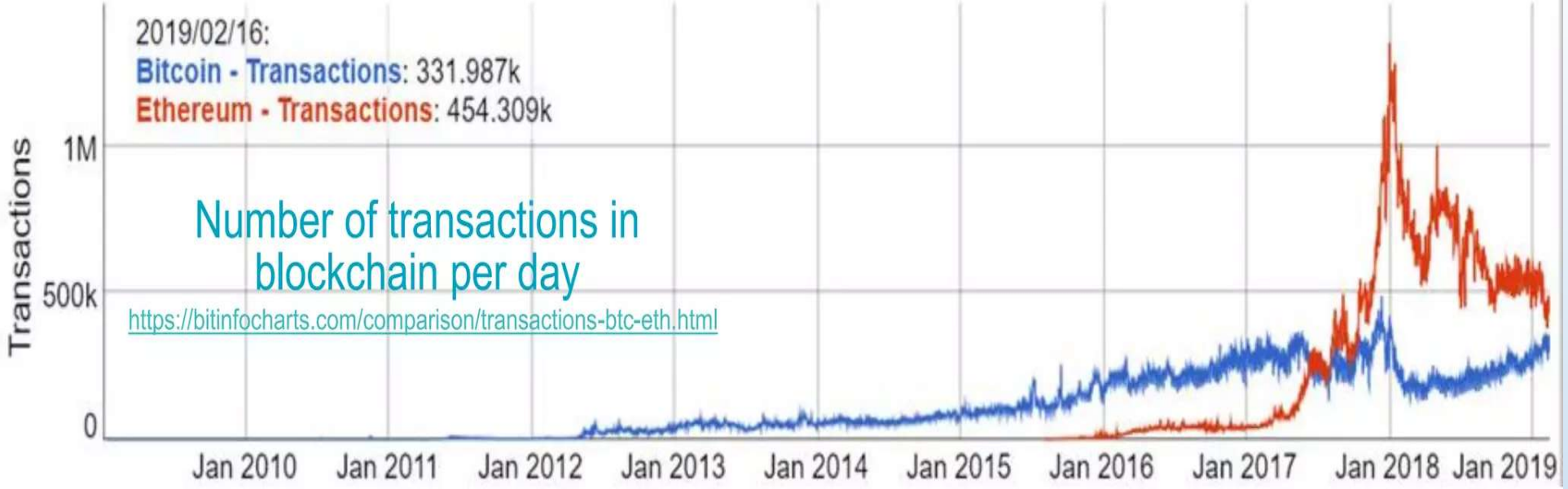
Ethereum block sizes have been continually adjusted to facilitate a healthy network through a voting process

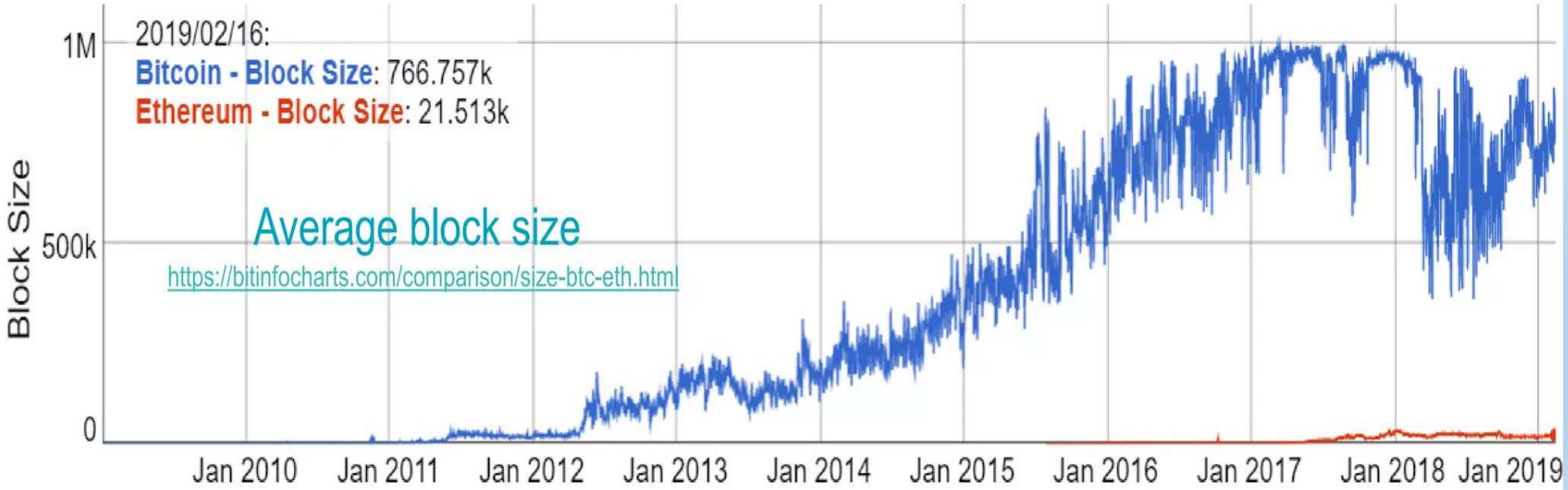
Bitcoin

- Number of transactions in blockchain per day: 332M
- Blocks in blockchain: 564K
- Average block size: 767 Kbytes
- Block size limit: 1MB or 8MB

Ethereum

- Number of transactions in blockchain per day: 455
- Blocks in blockchain: 7.23M
- Average block size: 26 Kbytes
- Block size limit: limited by gas-limit, which is the total overhead for all operations within the block


2019/02/16:
Bitcoin - Transactions: 331.987k
Ethereum - Transactions: 454.309k
Transactions
1M
500k
0
Number of transactions in blockchain per day
https://bitinfocharts.com/comparison/transactions-btc-eth.html
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019


2019/02/16:
Bitcoin - Block Size: 766.757k
Ethereum - Block Size: 21.513k
Block Size
1M
500k
0
Average block size
https://bitinfocharts.com/comparison/size-btc-eth.html
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019

Ethereum uses the Ethash cryptographic algorithm, over which special-hardware will not provide any advantage

Bitcoin

- Average block time: 10 minutes
- Cryptographic algorithm: SHA-256, for which special hardware can be used

Ethereum

- Average block: 10-20 seconds using the GHOST protocol
- Cryptographic algorithm: Ethash (the most commonly used hashing function is KECCAK-256, also called SHA-3), which is more complicated and memory intensive, for which no special-purpose chip provides any advantage
- Runs Ethereum Virtual Machine (EVM) and the programming language Solidity