

# ETH2.0 BEHIND THE SCENES

Created: Sept 2023  
Last Edited: Oct 2023

UofT: CSCD71F23  
-David Liu, Founder of dApp Technology Inc.

# ETH ACCOUNTS

Externally Owned Accounts (EOA) and Smart Contracts on Ethereum are ETH Accounts, identified by their ETH address.

Example:

0x397507d0E34756A192dE72787A0309bD3E8C038d

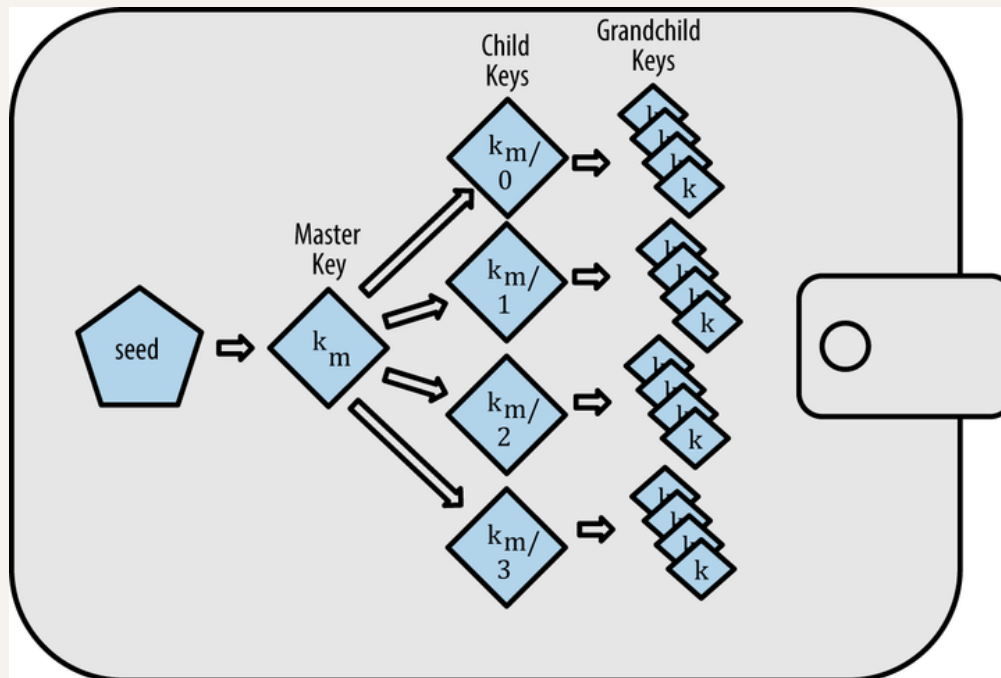
Smart Contracts can hold cryptocurrencies just like EOAs.

Both EOA and Smart Contracts can interact with Smart Contracts.

# HD WALLETS

A Hierarchical Deterministic (HD) Wallet is a type of Deterministic Wallet that utilizes a single root key to derive multiple private keys.

An implementation of HD Wallet is Metamask via its Keyring Module.



# HD WALLETS

The keys can be represented in the form of a Mnemonic Word Sequence (BIP39).

Mnemonic Phrase Example: indoor dish desk flag debris potato excuse depart ticket judge file exit

An implementation of HD Wallet is Metamask via its Keyring Module.

No.	word	No.	word	No.	word	No.	word
1	abandon	513	divorce	1025	length	1537	scale
2	ability	514	dizzy	1026	lens	1538	scan
3	able	515	doctor	1027	leopard	1539	scare
4	about	516	document	1028	lesson	1540	scatter
5	above	517	dog	1029	letter	1541	scene
6	absent	518	doll	1030	level	1542	scheme
7	absorb	519	dolphin	1031	liar	1543	school
8	abstract	520	domain	1032	liberty	1544	science
9	absurd	521	donate	1033	library	1545	scissors
10	abuse	522	donkey	1034	license	1546	scorpion
11	access	523	donor	1035	life	1547	scout
12	accident	524	door	1036	lift	1548	scrap
13	account	525	dose	1037	light	1549	screen
14	accuse	526	double	1038	like	1550	script
15	achieve	527	dove	1039	limb	1551	scrub
16	acid	528	draft	1040	limit	1552	sea
17	acoustic	529	dragon	1041	link	1553	search
18	acquire	530	drama	1042	lion	1554	season
19	across	531	drastic	1043	liquid	1555	seat

# EOA ADDRESS

An EOA private key can be generated using the Elliptic Curve Digital Signature Algorithm (ECDSA). Generation can happen locally on any device.

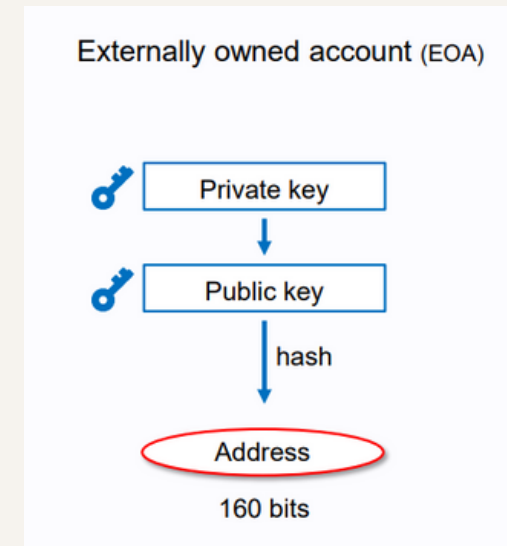
Private Key Example:

fffffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd036415f

A private key can derive a public key which can in turn derive a public ETH address.

ETH Address Example

0xb794f5ea0ba39494ce839613ffba74279579268



# SMART CONTRACT ADDRESS

- Each deployed smart contract has an address calculated via opcodes:
- CREATE (old version) or
- CREATE2

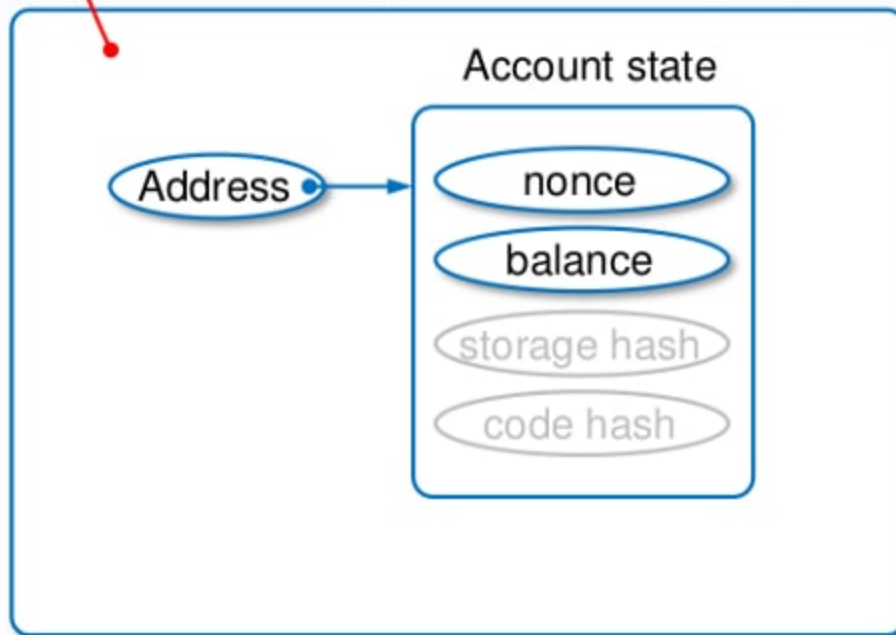
```
// CREATE2 be pre-calculated)  
keccak256(0xff, deployerAddress, salt, keccak256(init_code))[12:]
```

External actor

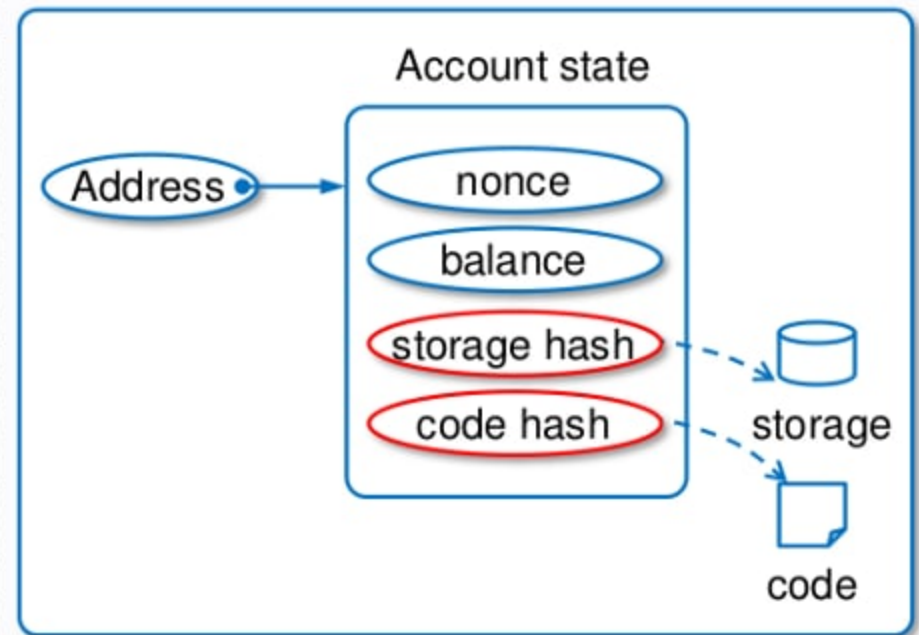


World state

Externally owned account (EOA)



Contract account



EOA is controlled by a private key.  
EOA cannot contain EVM code.

Contract contains EVM code.  
Contract is controlled by EVM code.

# SIGINING

A private key can also sign messages and transactions which output a signature.

ETH Signed Transaction Example:

## Signed Transaction

```
0xf86c0a8502540be400825208944bbeeb066ed09b7aed07bf39e...
```

## Raw Transaction

```
{  
  "value": "0xde0b6b3a7640000",  
  "data": "0x",  
  "to": "0x4bbeeb066ed09b7aed07bf39eee0460dfa261520",  
  "nonce": "0xa",  
  "gasPrice": "0x2540be400",  
  "gasLimit": "0x5208",  
  "chainId": 0  
}
```

```
maxFeePerGas: "300",  
maxPriorityFeePerGas: "10",
```

Send Transaction



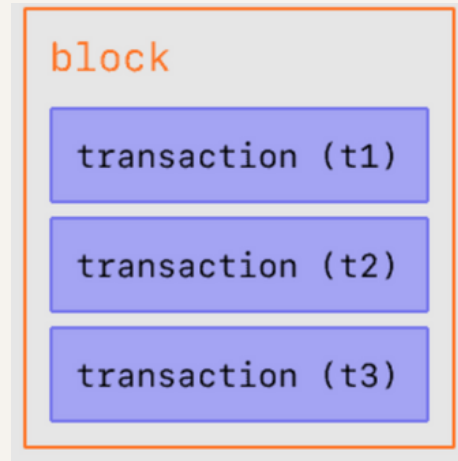
```
// ETH Signed Message Example:
{
  "address": "0x76e01859d6cf4a8637350bdb81e3cef71e29b7c2",
  "msg": "Hello world!",
  "sig":
"0x21fbf0696d5e0aa2ef41a2b4fffb623bcacf070461d61cf7251c74161f82fec3a437085
4bc0a34b3ab487c1bc021cd318c734c51ae29374f2beb0e6f2dd49b4bf41c",
  "version": "2"
}
```

```
const digest =
"0x7c5ea36004851c764c44143b1dcb59679b11c9a68e5f41497f6cf3d480715331";

// Using an expanded Signature
recoverAddress(digest, {
r: "0x528459e4aec8934dc2ee94c4f3265cf6ce00d47cf42bb106afda3642c72e25eb",
s: "0x42544137118256121502784e5a6425e6183ca964421ecd577db6c66ba9bccdcf",
v: 27 });
// '0x0Ac1dF02185025F65202660F8167210A80dD5086'

// Using a flat Signature
const signature =
"0x528459e4aec8934dc2ee94c4f3265cf6ce00d47cf42bb106afda3642c72e25eb42544
137118256121502784e5a6425e6183ca964421ecd577db6c66ba9bccdcf1b";
recoverAddress(digest, signature);
// '0x0Ac1dF02185025F65202660F8167210A80dD5086'
```

# BLOCKS



- Consensus in batches
- Ordered Blocks and Transactions
- Target Size: 15 million gas
- Max Size: 30 million gas

# BLOCKS HEADER

Field	Description
slot	the slot the block belongs to
proposer_index	the ID of the validator proposing the block
parent_root	the hash of the preceding block
state_root	the root hash of the state object
body	an object containing several fields, as defined below

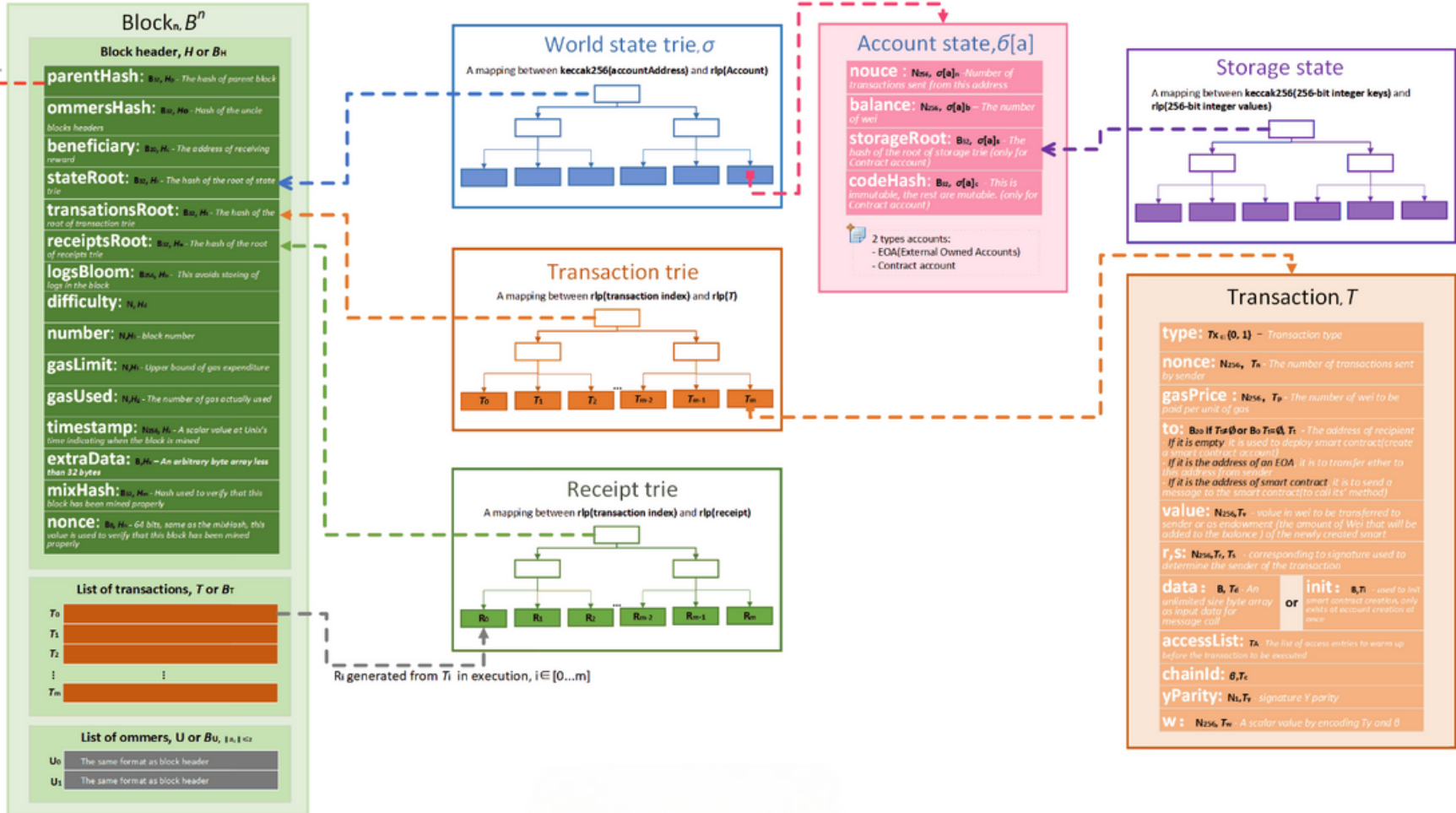
Sept 2023

<https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture>

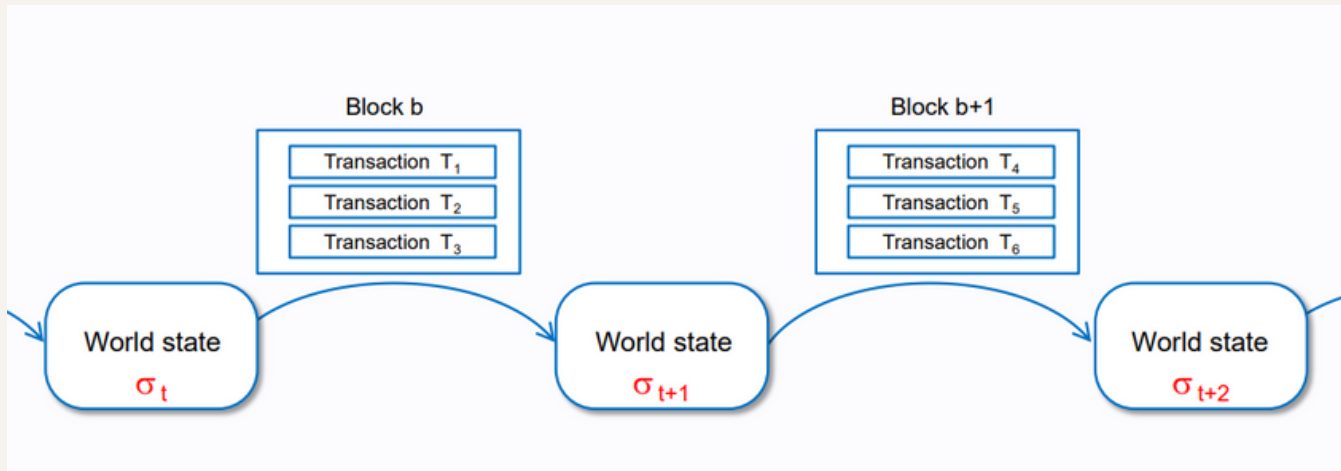
# BLOCKS

The block to be mined (mining case)  
or  
The block to be verified (syn case)

- $B_{n,p} = \text{KEC}(\text{RLP}(B_{n-1}^p))$
- Note:  $B_{n,p}$  means  $H_n$  in the header of  $B^n$

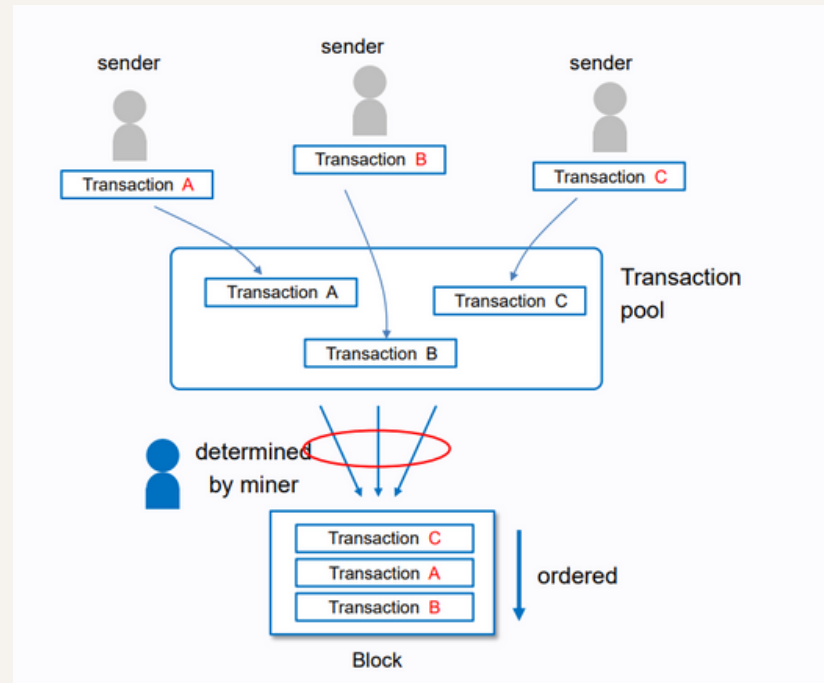


# TRANSACTIONS



Each Transaction contains instructions to move the blockchain from one "World State" to another.

# TRANSACTIONS



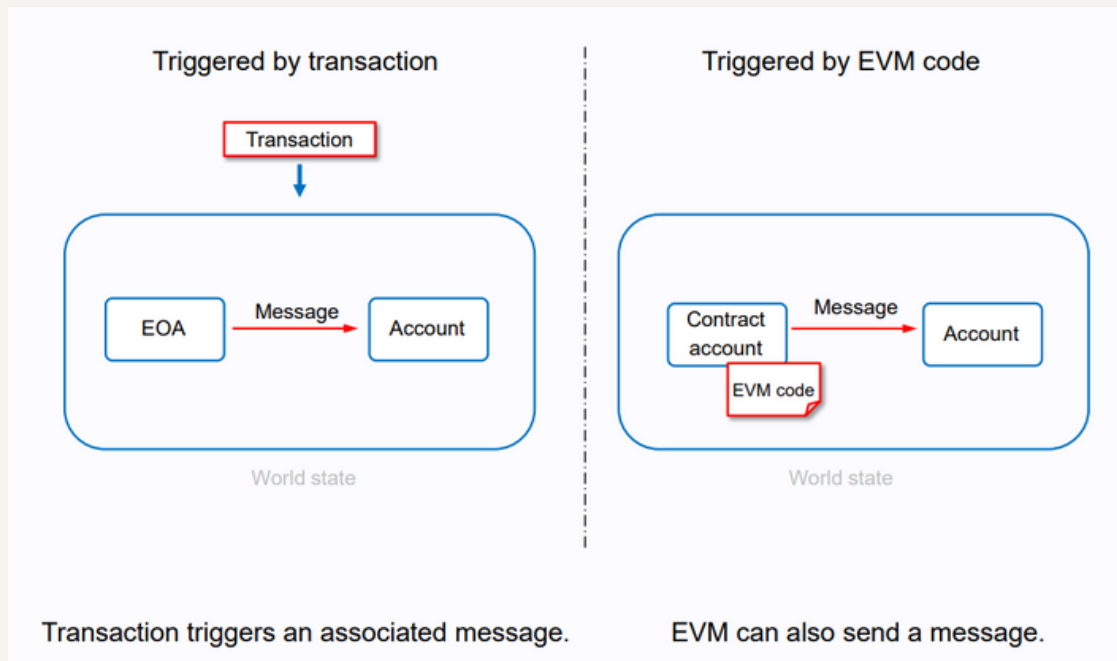
The transactions are selected from the Transaction pool (also called Mempool) to a block, in an order the Validators choose.

All transactions in a block share the same timestamp.

## Definitions from Yellow Paper

**Transaction:** A piece of data, signed by an External Actor. It represents either a Message or a new Autonomous Object. Transactions are recorded into each block of the blockchain.

**Message (Internal Tx):** Data (as a set of bytes) and Value (specified as Ether) that is passed between two Accounts, either through the deterministic operation of an Autonomous Object or the cryptographically secure signature of the Transaction



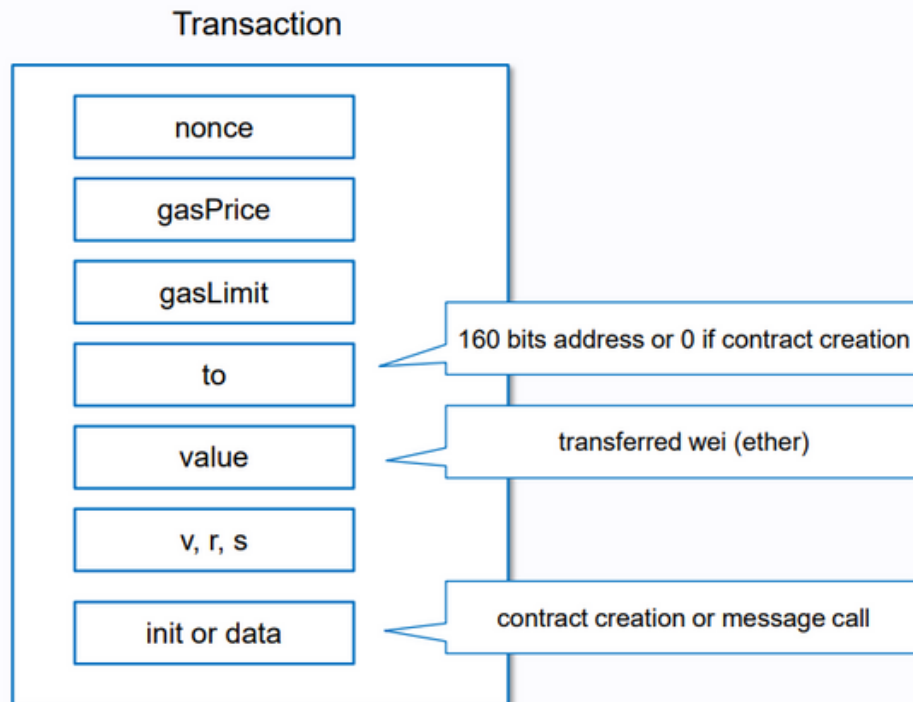
### Transaction Example

```
{
  from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
  to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
  gasLimit: "21000",
  maxFeePerGas: "300",
  maxPriorityFeePerGas: "10",
  nonce: "0",
  value: "10000000000"
}
```



```
{
  from: "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
  to: "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
  gasLimit: "21000",
  maxFeePerGas: "300",
  maxPriorityFeePerGas: "10",
  nonce: "0",
  value: "10000000000"
}
```

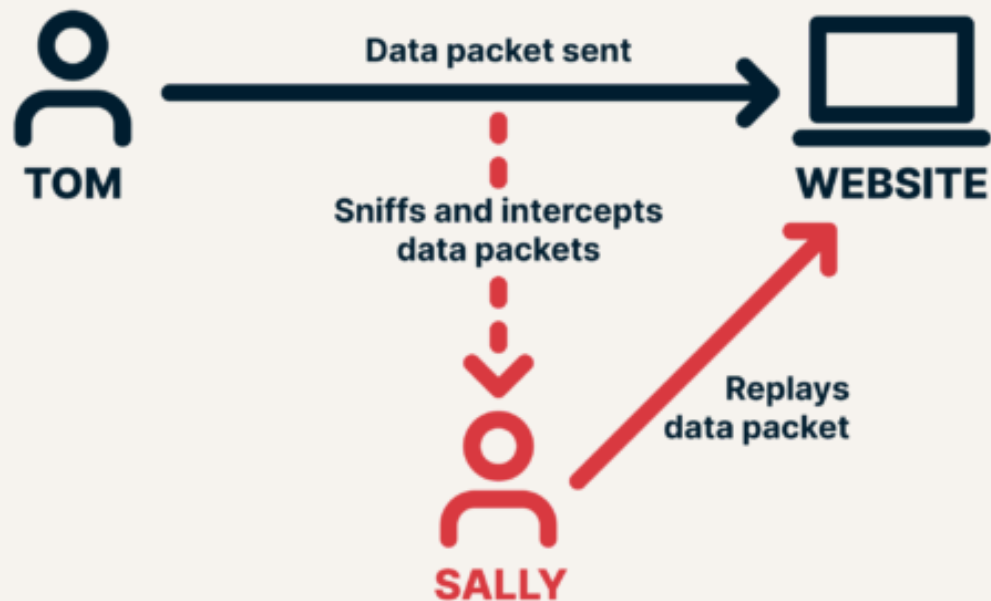
## Field of a transaction



v, r, s are the values for the transaction's signature. The signature is actually of the Keccak-256 hash of the RLP-serialized transaction data, not of the data itself.

# TRANSACTION NONCE

## Replay Attack Prevention



# TRANSACTION NONCE

- Incremental count
- Nodes will only execute transactions with the next nonce from the mempool
- If there's a gap between the last submitted transaction's nonce and a new transaction's nonce, then the new transaction will not be processed
- If 2 transactions in the mempool have the same nonce, validators can choose which one to process (mostly random in signer's perspective)

# TRANSACTION BOTTLENECK

- Transactions takes ~2 seconds to be added to blockchain
- EOAs can only submit transactions with incremental nonce

**How to process 1000s of transactions quickly?**

# TRANSACTION BOTTLENECK

- Transactions takes ~2 seconds to be added to blockchain
- EOAs can only submit transactions with incremental nonce

**How to process 1000s of transactions quickly?**

Use multiple EOAs!

# SPECIAL TRANSACTION

- Transaction sent to the zero address 0x0...0
- “data” field of the transaction will contain the smart contract compiled bytecode
- “value” field is optional, for an address starting balance

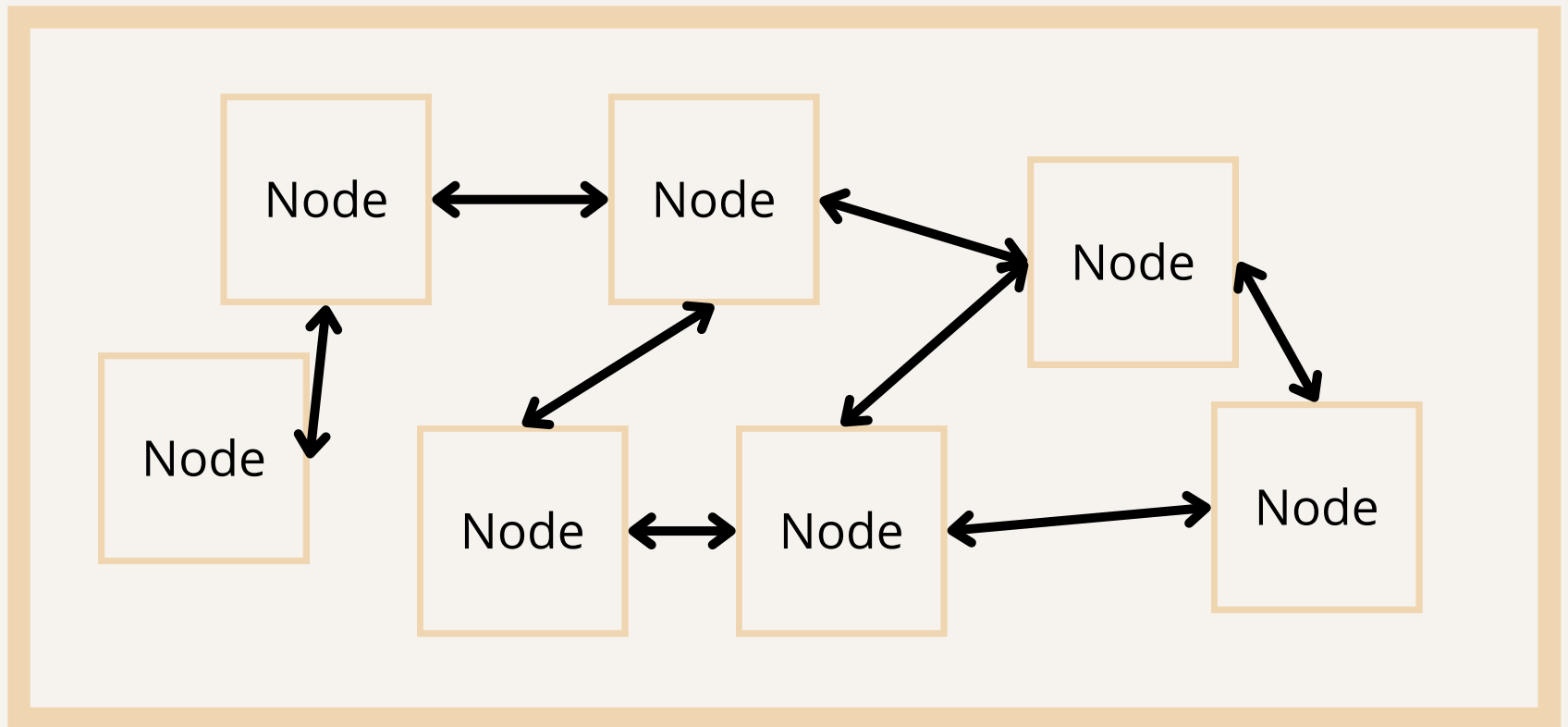
Note: Burn tokens by sending to 0x0...0 or 0x0...dEaD

# EIP155

- EIP155 “Simple Replay Attack Protection”
- Signed transactions can take the form of a hash “0x...” or in the form of 3 integers stored in  $r$ ,  $s$  and recovery value  $v$
- A signed transaction submitted on a chain (ex. Sepolia), can also be submitted on another (ex. Mainnet)
- EIP155 added chain id to the “ $v$ ” to protect against crosschain replay attacks

Chain Name	Chain ID
Mainnet	1
ETH Classic	61
Polygon	137
Sepolia	11155111
Goerli	5

# ETHEREUM NETWORK



- Each Node stores a portion of the blockchain and runs the EVM to execute code from Smart Contracts
- Ethereum Validators receives data from the Nodes and adds new blocks to the blockchain



# ETHEREUM NETWORK (SEPT 2023)

## Ethereum today

The latest network statistics

### TOTAL ETH STAKED

The total amount of ETH currently being staked and securing the network.

**25.19M** ⓘ

30d 90d

### TRANSACTIONS TODAY

The number of transactions successfully processed on the network in the last 24 hours.

**1.022M** ⓘ

30d 90d

### VALUE LOCKED IN DEFI (USD)

The amount of money in decentralized finance (DeFi) applications, the Ethereum digital economy.

**\$48.34B** ⓘ

30d 90d

### NODES

Ethereum is run by thousands of volunteers around the globe, known as nodes.

**7,875** ⓘ

30d 90d

# ETHEREUM STATS (SEPT 2023)

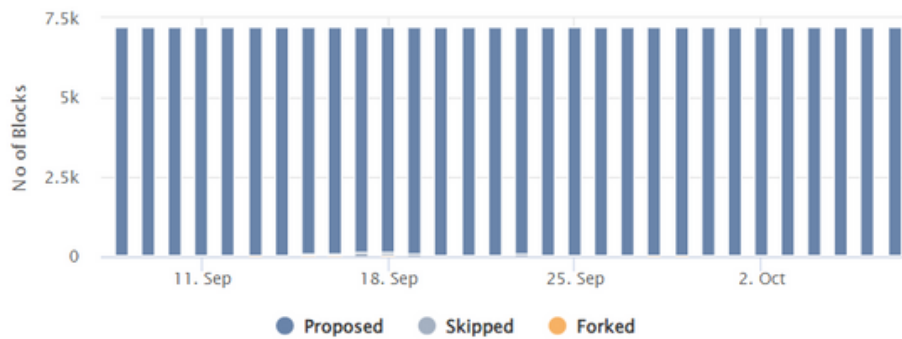
Showing the last 30 days

07 Sep 2023 - 06 Oct 2023

## BLOCKS

[View Details](#)

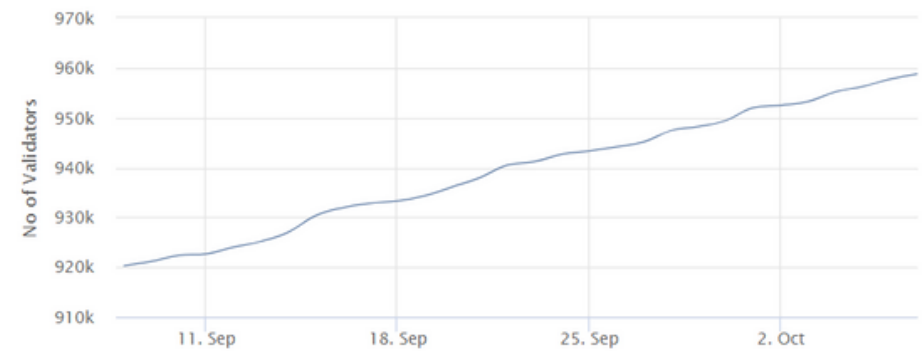
Blocks produced on the Beacon Chain.



## VALIDATORS

[View Details](#)

The number of validators being run on the Beacon Chain.



## ATTESTATIONS

[View Details](#)

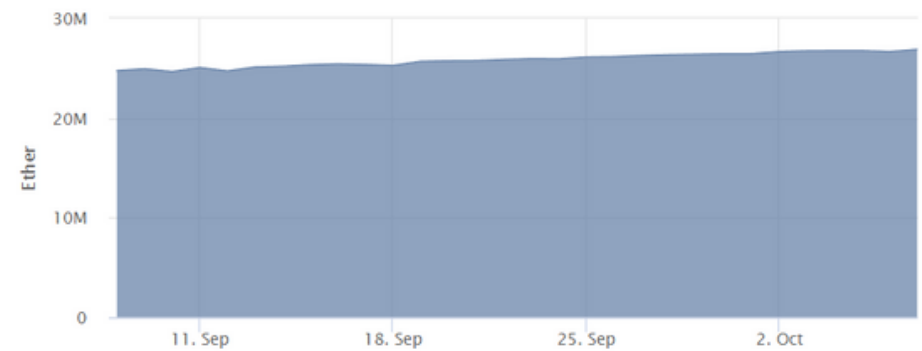
The votes on the validity of newly created blocks on the Beacon Chain.



## ETHER VOTED

[View Details](#)

The number of ether staked in the participation on the Beacon Chain.



# ETHEREUM TIMELINE

## Roadmap to Ethereum 2.0



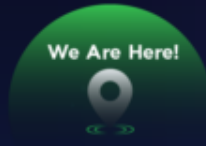
**The Merge**  
PoS replace ► PoW



**Beacon Chain**  
Launch

Altair Upgrade

We Are Here!



**Sharding**



**Ethereum**  
**2.0**

2015

2020

2021

2022

2023 - 2024

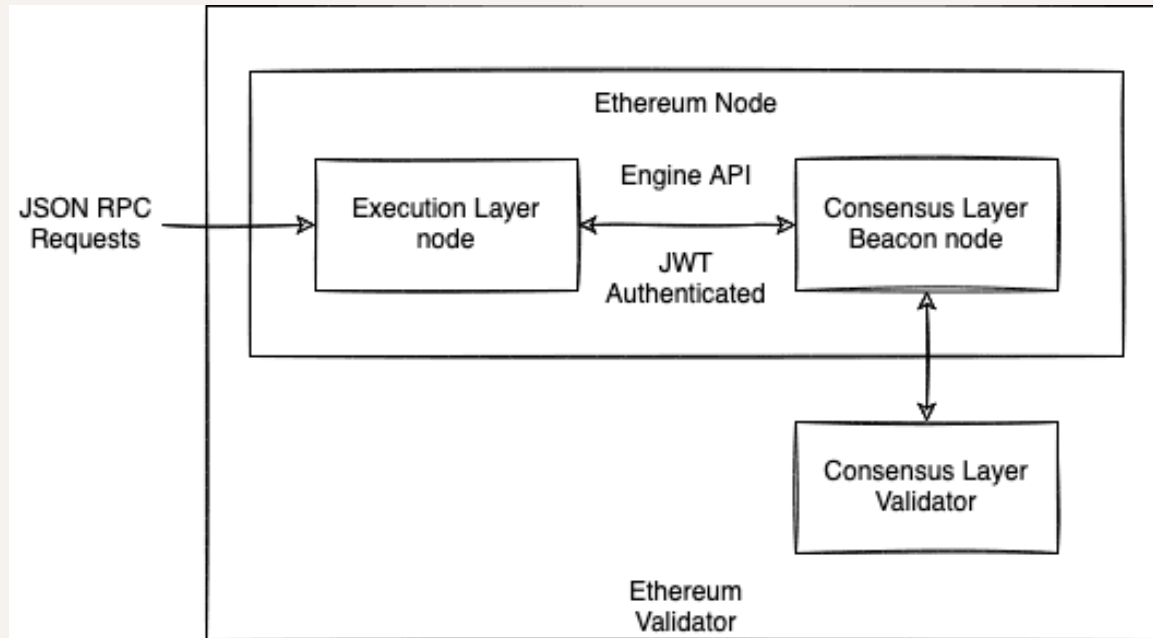
**Proof of Stake (PoS)**

**Proof of Work (PoW)**

**Ethereum**  
**1.0**

Berlin Upgrade  
London Upgrade  
Shanghai Upgrade

# ETHEREUM NODE



- Execution Node: transaction gossip , executes transactions and holds the Ethereum state.
- Beacon Node: implements PoS Consensus, block gossip
- Validator: Stakes 32ETH, ~4% APY, attests data correctness, votes for next block proposer, submits new blocks

# ETHEREUM NODE TYPES

<b>Full</b>	<b>Archive</b>	<b>Light</b>
Stores recent blockchain data (~128 blocks). Other data can be regenerated from 'snapshots' by a full node.	Stores all blockchain data	Stores block headers
Participates in block validation, verifies all blocks and states	Participates in block validation, verifies all blocks and states	Can be used to verify data against state roots in block headers
Serves the network and provides data on request	Serves the network and provides data on request	Can be run with average bandwidth phones or embedded devices

Special: Bootnodes

# ETHEREUM NODE HARDWARE EXAMPLE

## Minimum requirements

- CPU with 2+ cores
- 8 GB RAM
- 2TB SSD
- 10+ MBit/s bandwidth

## Recommended specifications

- Fast CPU with 4+ cores
- 16 GB+ RAM
- Fast SSD with 2+TB
- 25+ MBit/s bandwidth

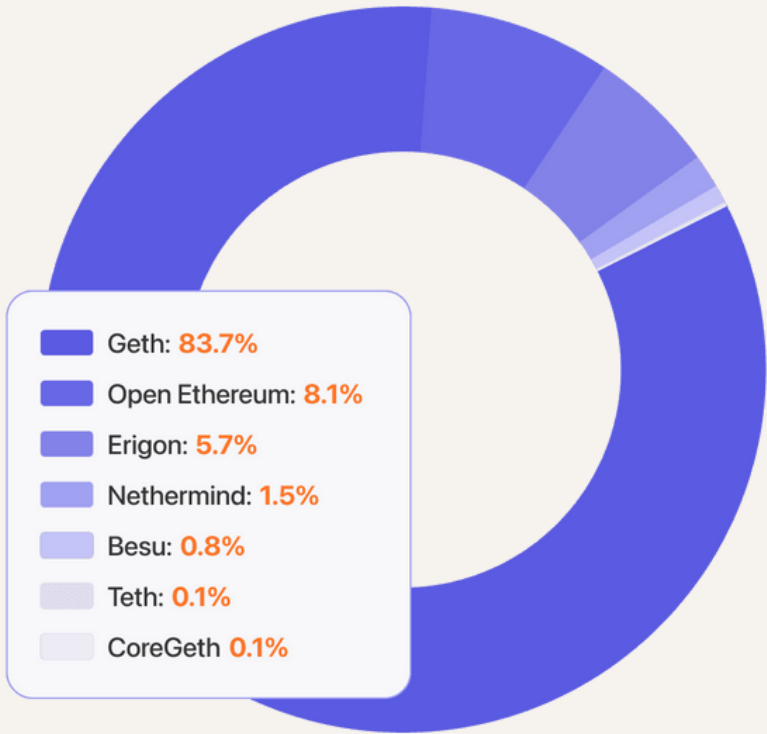
## GreyWizard's NUC10i7FNH



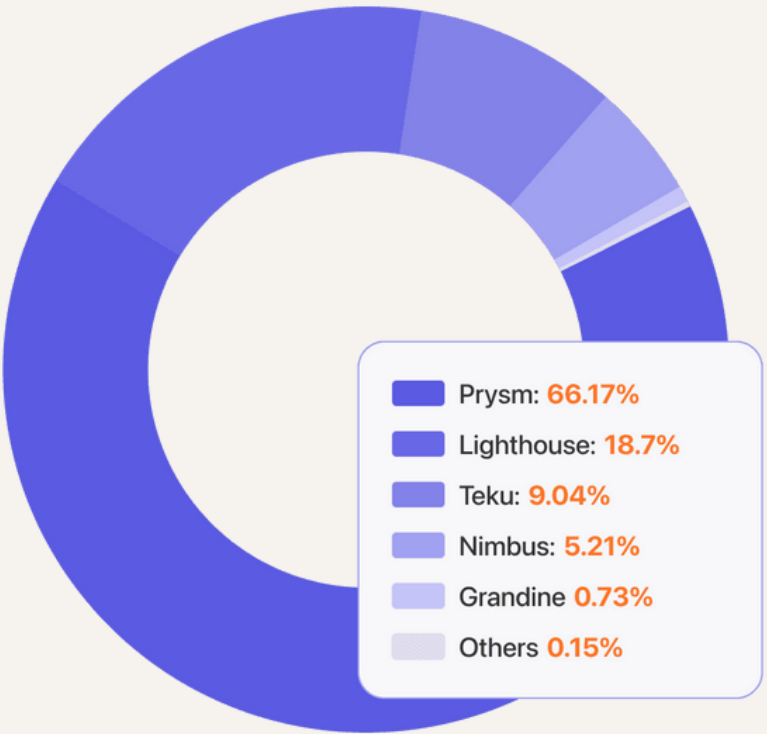
Sept 2023  
rocketpool.net

- GreyWizard's Setup:
- Base: Intel BXNUC10I7FNH1 (\$445)
- RAM: 2x Samsung M471A4G43MB1 32GB DDR4 SODIMM 2666 MHz (\$154 ea.)
- SSD: Samsung 970 EVO Plus 2TB M.2 2280 NVMe SSD (\$315)
- Total: \$1068

# ETHEREUM CLIENTS



EXECUTION CLIENTS



CONSENSUS CLIENTS

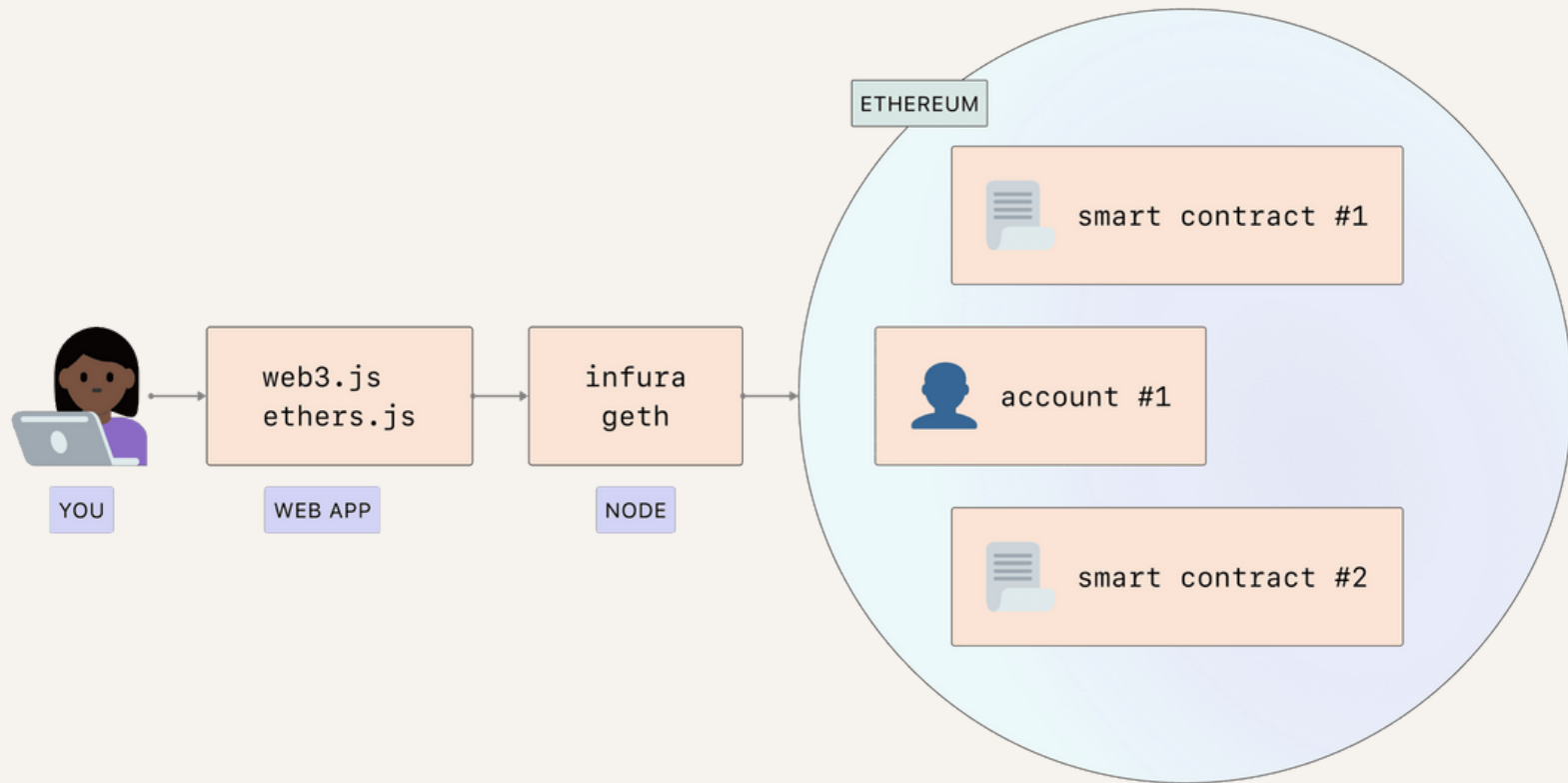
# ETHEREUM NODE AS A SERVICE



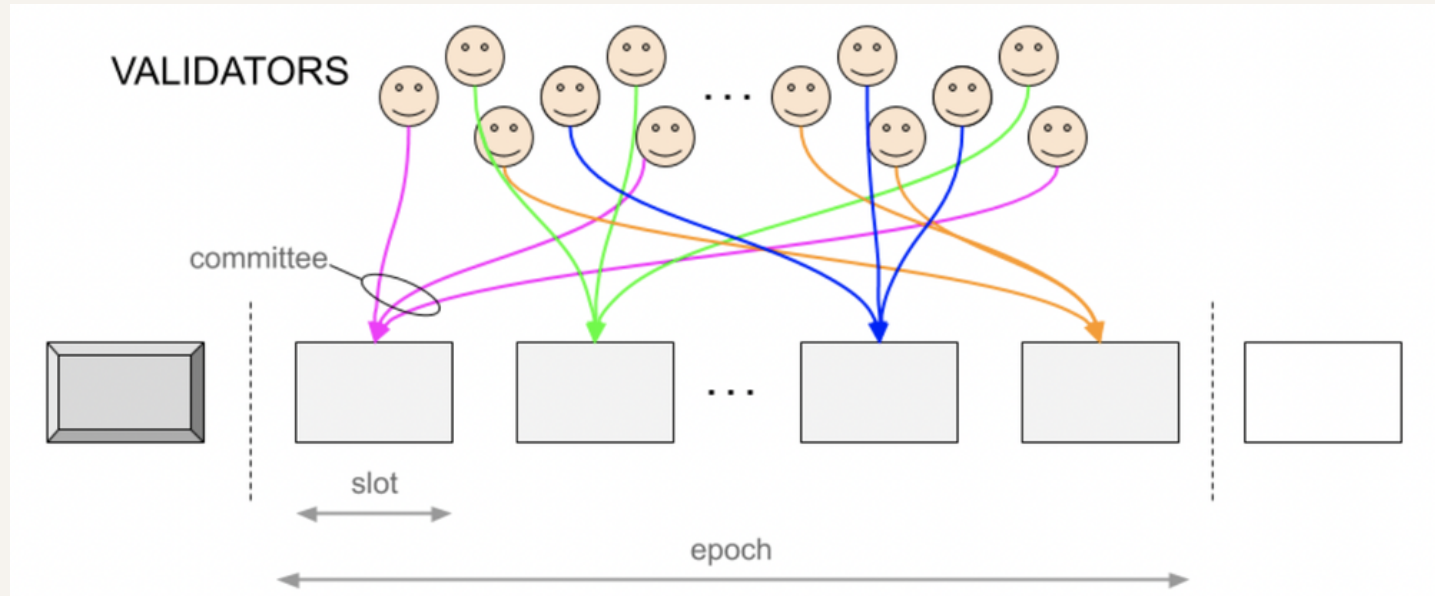
- Distributed node client management is handled by node service providers, relieving you of the operational burden
- These providers typically give you an API key and JSON RPC URL for blockchain read and write operations



# CONNECTING APPS TO ETHEREUM



# COMMITTEES



- All validators are divided into committees with at least 128 validators in each
- One or more Committees attest to each slots, 12 secs to attest
- Each Slot may or may not have a block
- 32 Slots = 1 Epoch

Problems?

# COMMITTEES

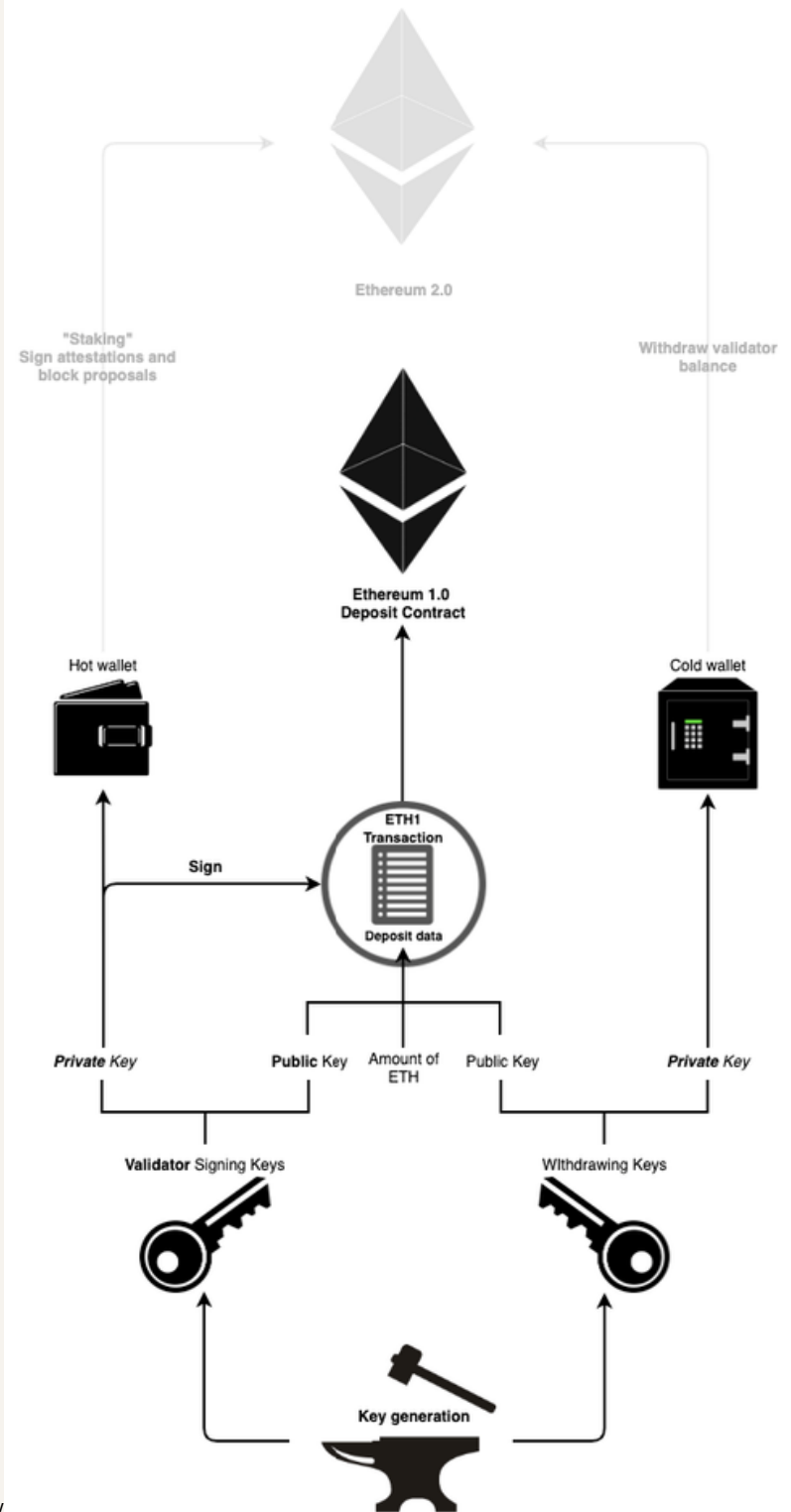
- Fork Choice Algorithm, LMD Ghost
- Reward and Slashing, Gasper
- Inclusion Delay
- 16 Validators from amongst the committees are randomly chosen to be aggregators per Epoch

# ETHEREUM NODE KEYS

Validators have BLS Withdrawing Key and Validating key

BLS is a separate cryptographic scheme than ECDSA

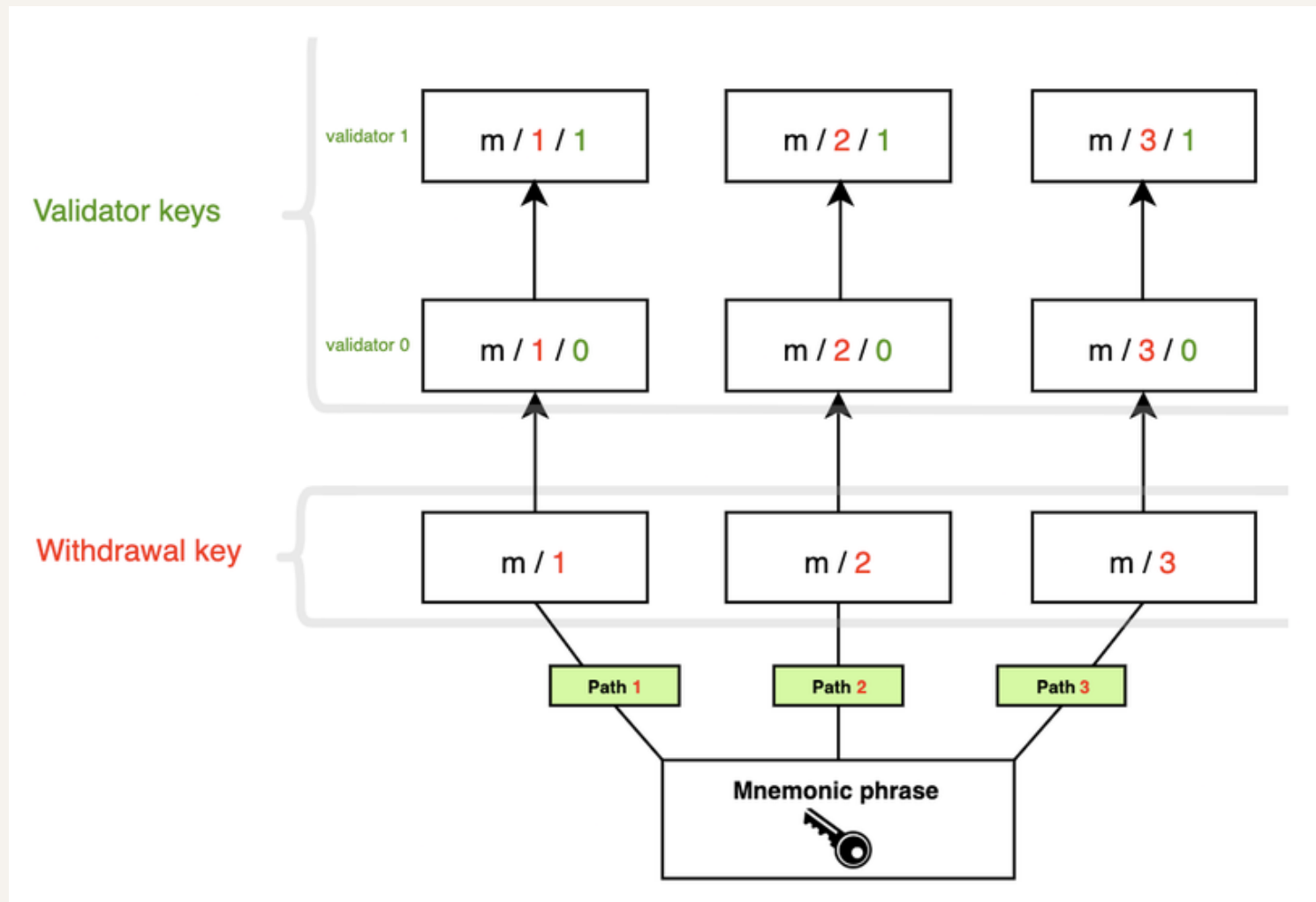
BLS keys allows combining multiple signatures into one signature. Collective signature validation is now done in constant time



# ETH 2.0 KEY GENERATION

master\_key/purpose/coin\_type/account/change/address\_index

Eth 2.0 Ex: m/12381/3600/withdrawal\_path\_i/0/validator\_i



# FINALITY

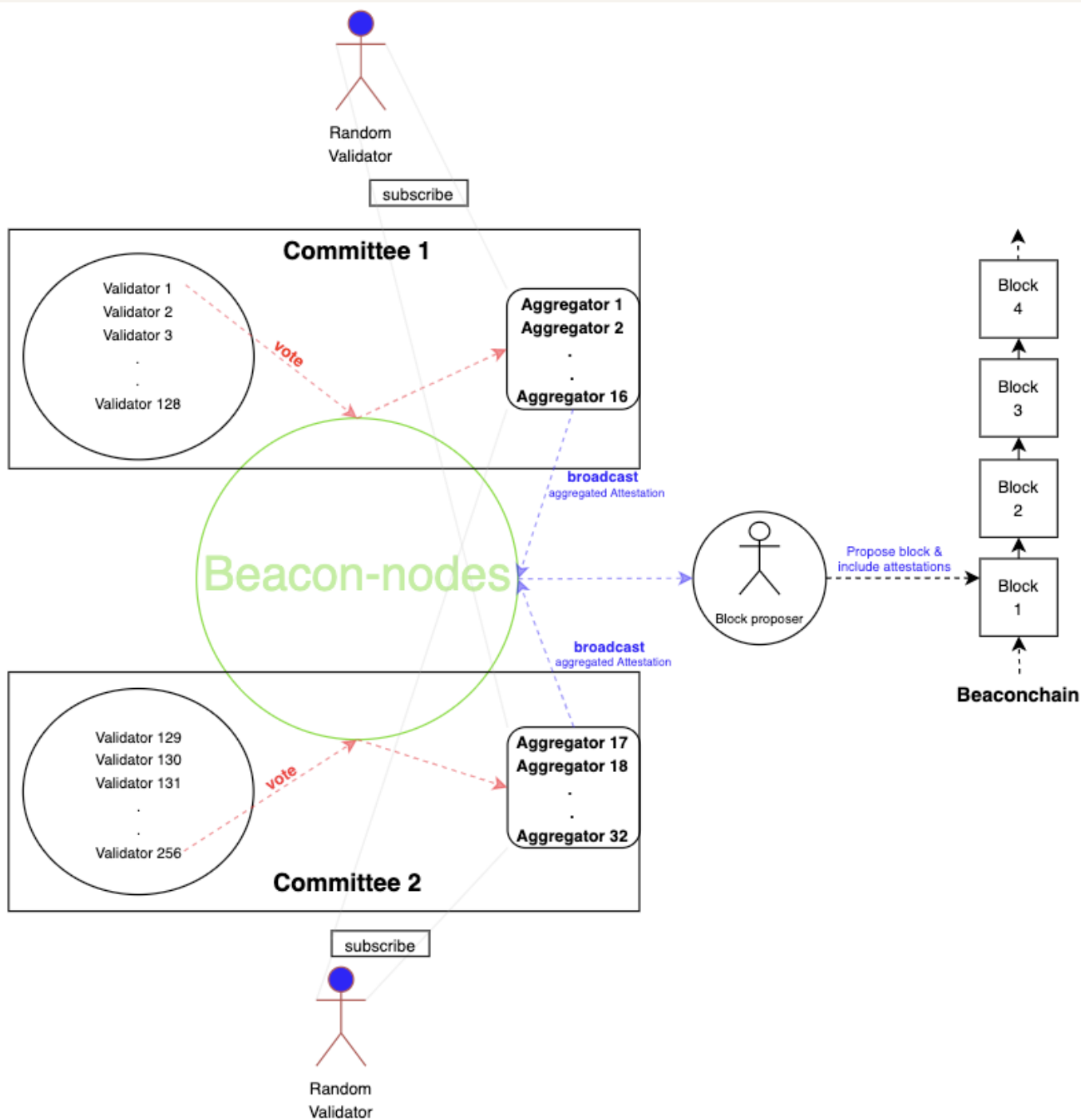
Epochs

Show 10 entries

Epoch	Time	Blocks	Attestations	Deposits	Slashings P / A	Finalized
4615	2 minutes ago	32	70	0	0 / 0	No
4614	8 minutes ago	32	598	0	0 / 0	No
4613	15 minutes ago	32	408	0	0 / 0	Yes
4612	21 minutes ago	32	749	0	0 / 0	Yes
4611	27 minutes ago	32	485	0	0 / 0	Yes
4610	34 minutes ago	32	366	0	0 / 0	Yes

- 66% votes on an Epoch is considered Justified
- 3 consecutive justified latest Epoch is considered finalized (~6 mins per epoch)

# ATTESTATION



1. Generation
2. Propagation
3. Aggregation
4. Propagation
5. Inclusion

# TIMELINE

The attestation contains the following components:

- `aggregation_bits`: a bitlist of validators where the position maps to the validator index in their committee; the value (0/1) indicates whether the validator signed the data (i.e. whether they are active and agree with the block proposer)
- `data`: details relating to the attestation, as defined below
- `signature`: a BLS signature that aggregates the signatures of individual validators



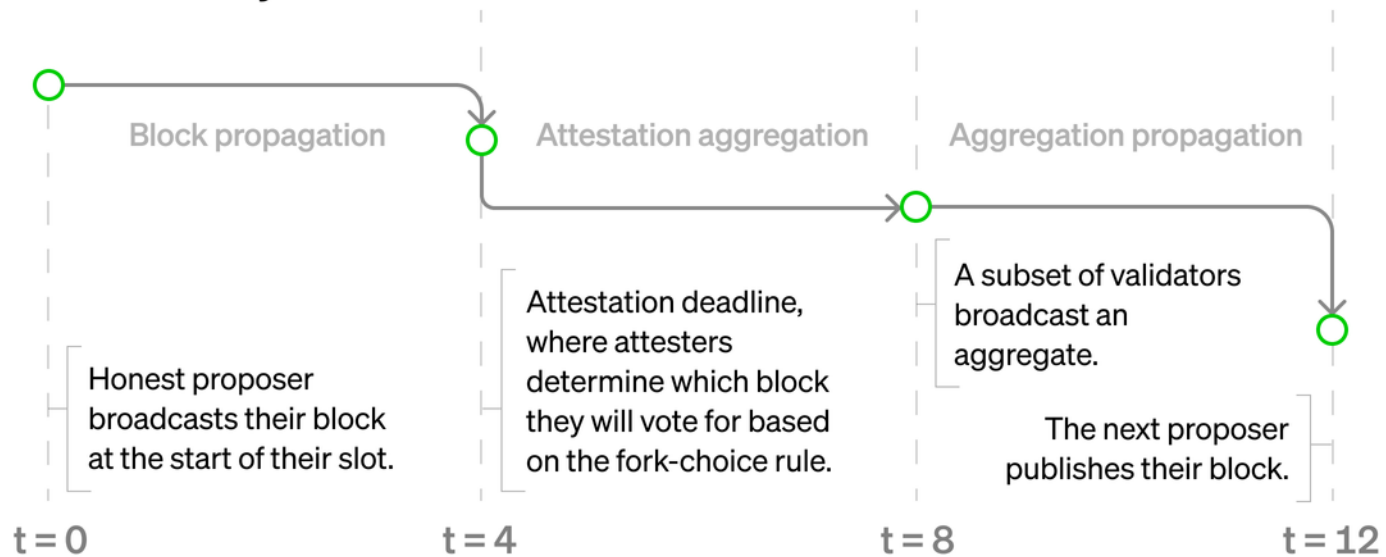
# TIMELINE

The first task for an attesting validator is to build the data. The data contains the following information:

- `slot`: The slot number that the attestation refers to
- `index`: A number that identifies which committee the validator belongs to in a given slot
- `beacon_block_root`: Root hash of the block the validator sees at the head of the chain (the result of applying the fork-choice algorithm)
- `source`: Part of the finality vote indicating what the validators see as the most recent justified block
- `target`: Part of the finality vote indicating what the validators see as the first block in the current epoch

# TIMELINE

## Slot anatomy



# ETHEREUM IS A DARK FOREST



# MEV

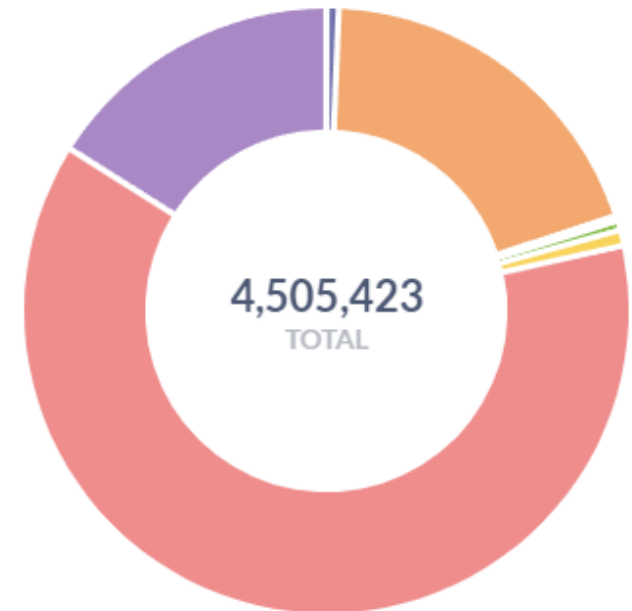
- Maximal Extractable Value
- Block Proposing Validator's goal is to earn ETH from gas fees.
- Will choose and rearrange transactions in a block to maximize profit
- Revenue = Base Issuance + Tx Fees + MEV

Cumulative Extracted MEV - Gross Profit ⓘ



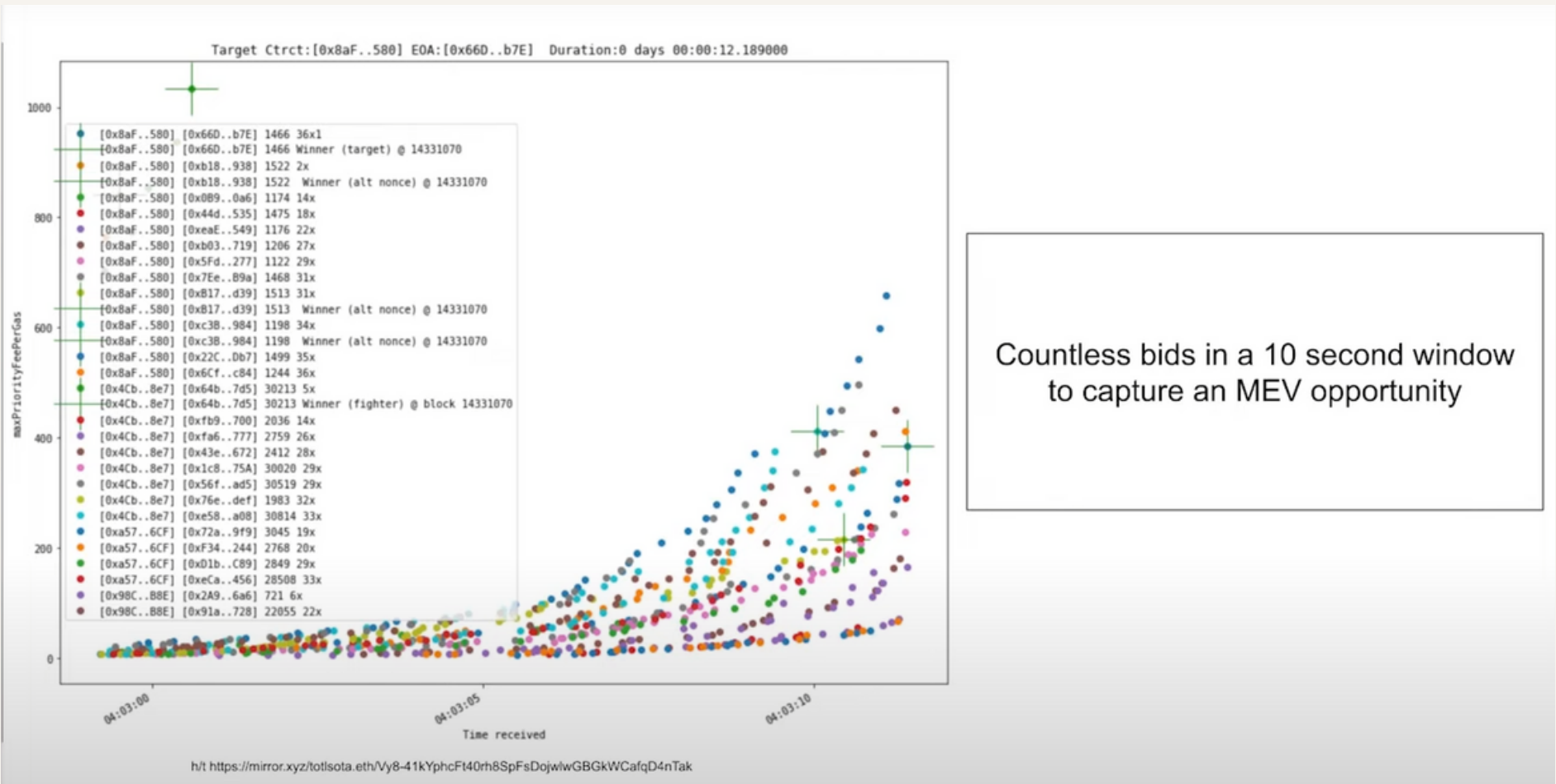
Extracted MEV Split by Protocol ⓘ

- Aave
- Balancer V1
- Bancor
- Compound V2
- Curve
- Uniswap V2
- Uniswap V3



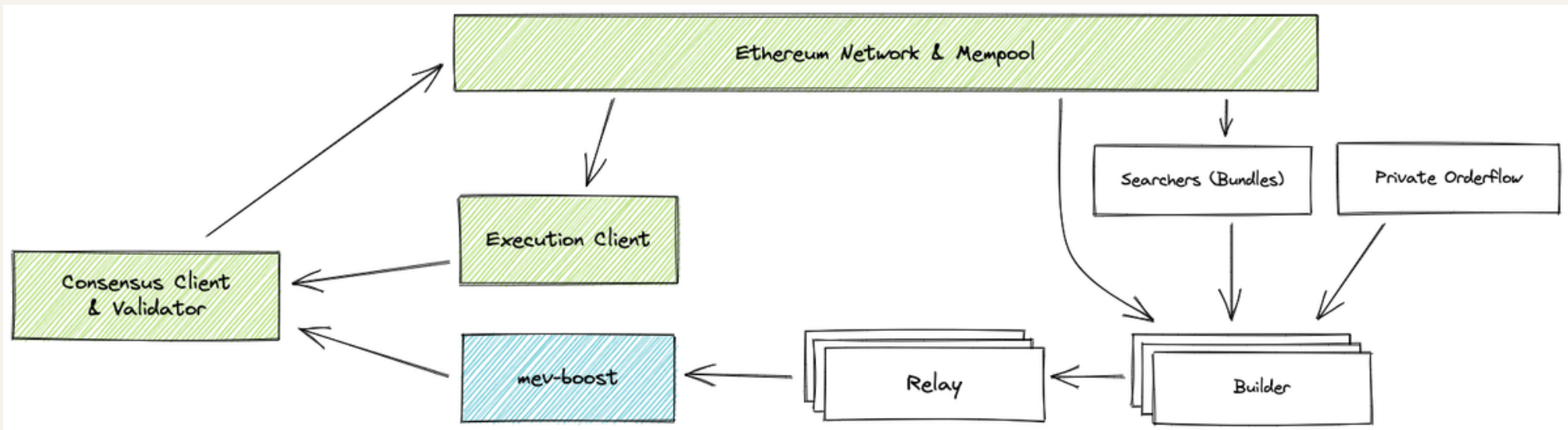
# MEV

- Time, resources and expertise needed to keep mempool quickly updated and find optimal block reorganization

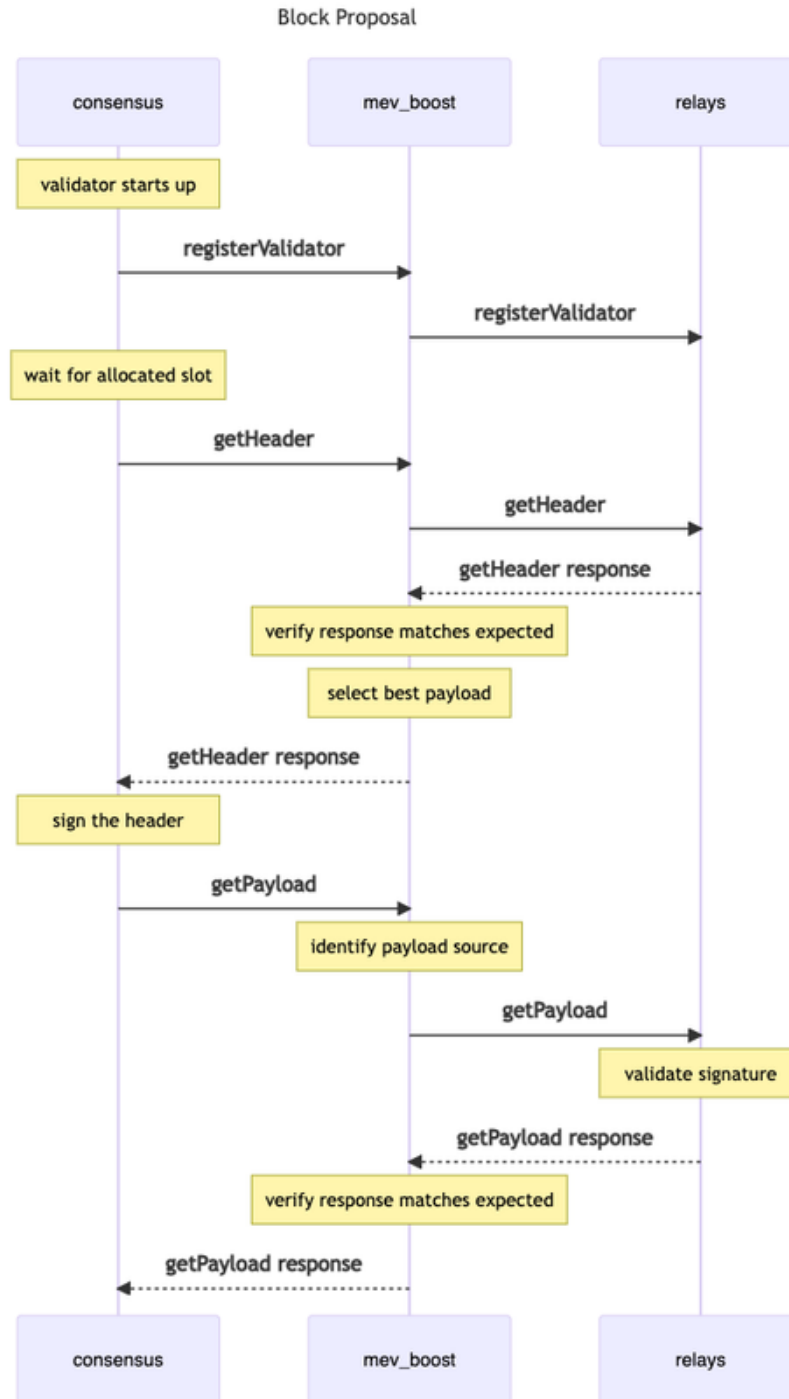


# MEV

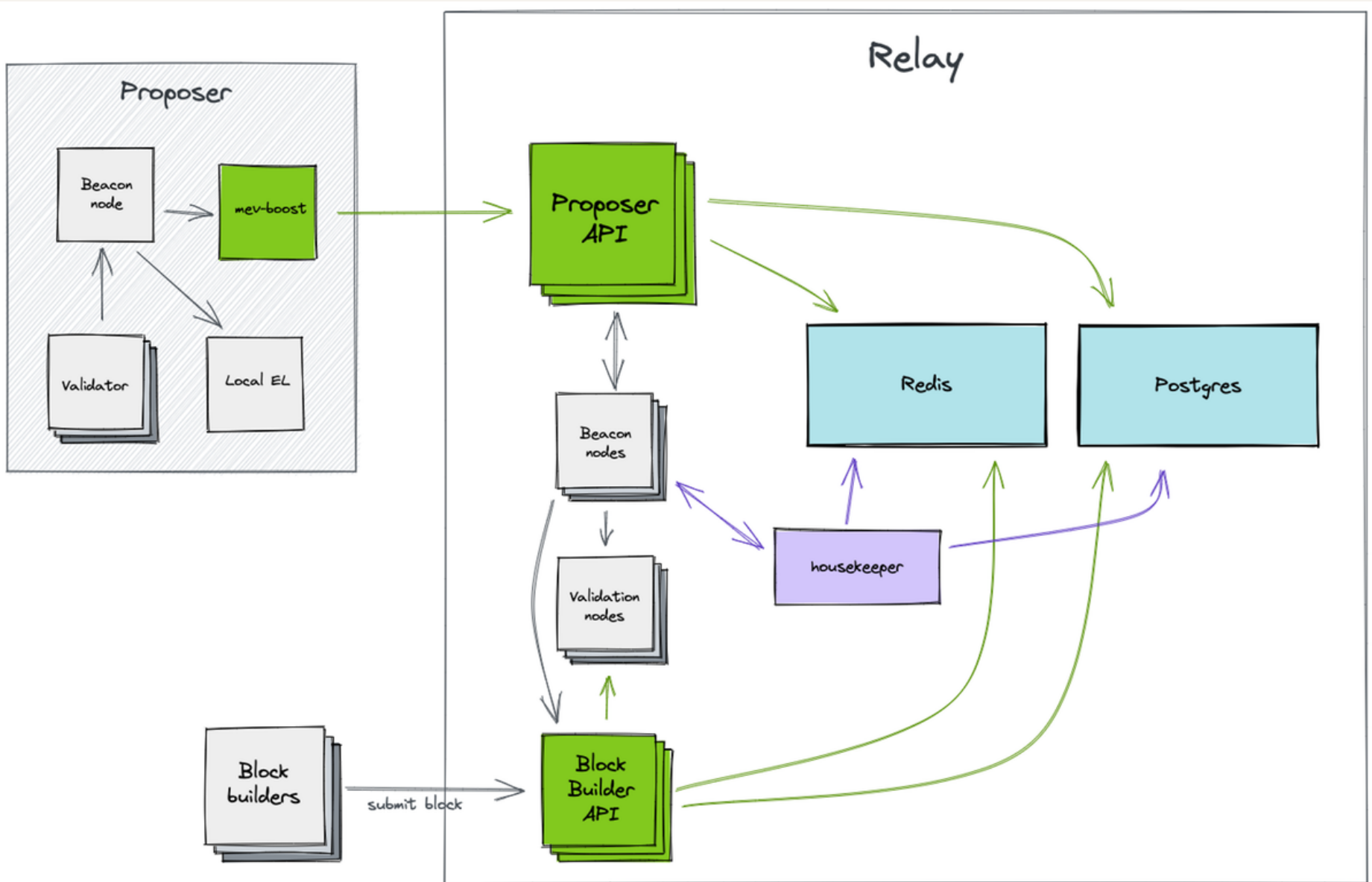
- MEV block production is usually outsourced to Searchers + Builders + Relayers, by the Block Producer (MEV)
- Searchers scans mempool for transaction opportunities
- Builders Construct the optimized blocks of transactions
- Relayers delivers the blocks to the Block Producer



# MEV



# MEV RELAY

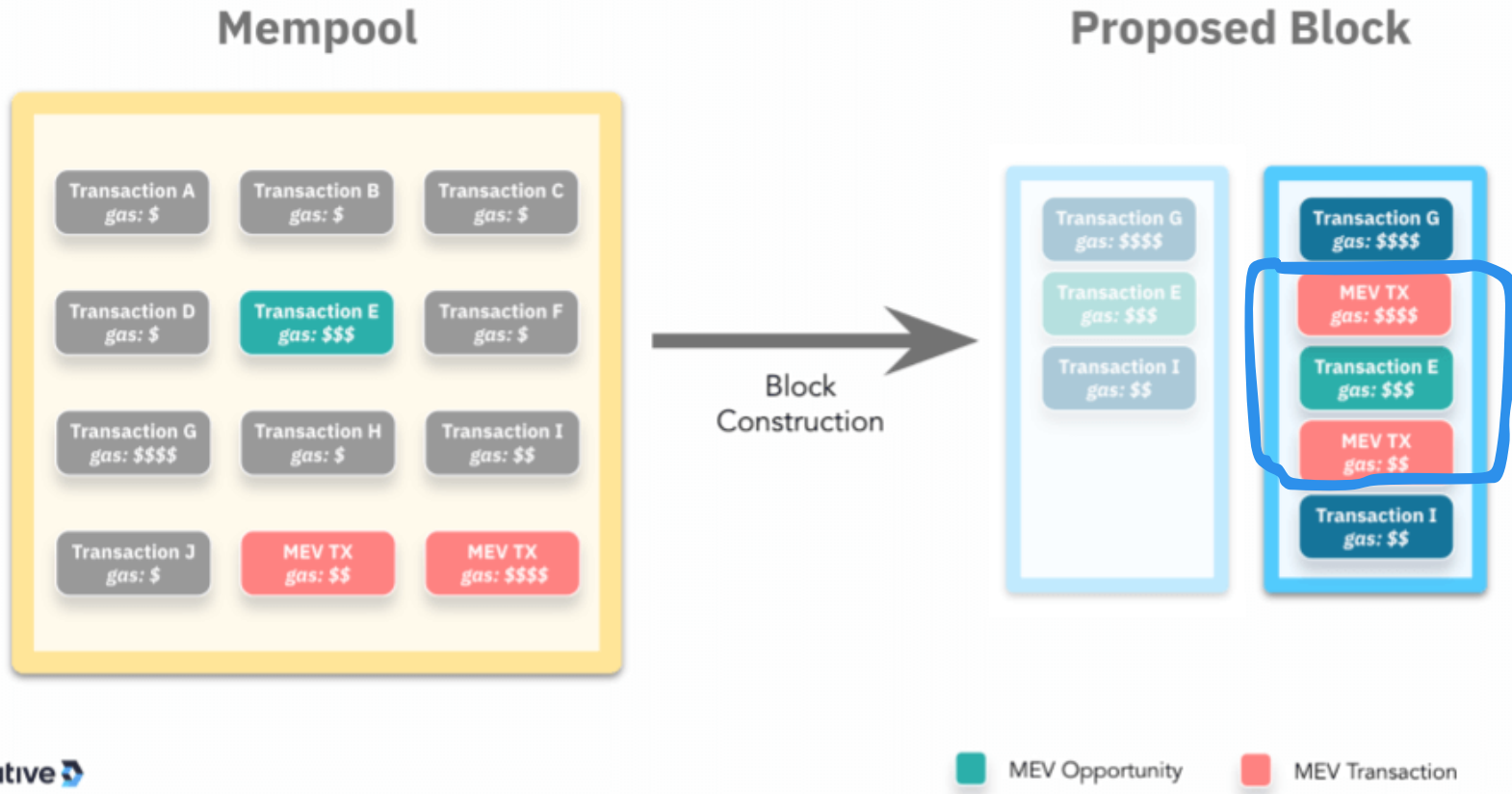


Arrows indicate request direction  
(from originator to target)



# MEV TRANSACTION BUNDLE

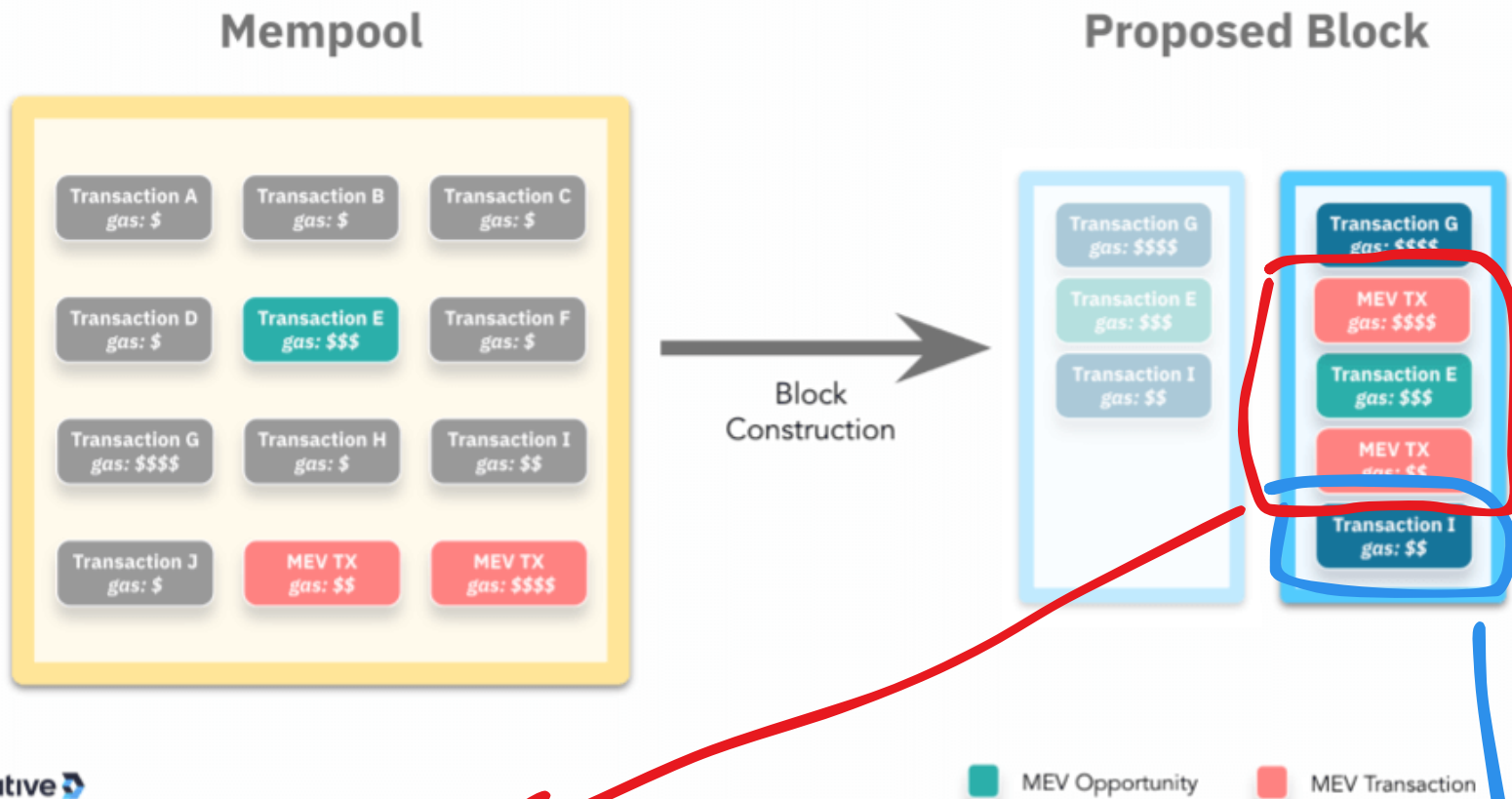
## Transaction Ordering Sandwich Attacks



# MEV EXAMPLES

## SPONSORED TRANSACTIONS

### Transaction Ordering **Sandwich Attacks**



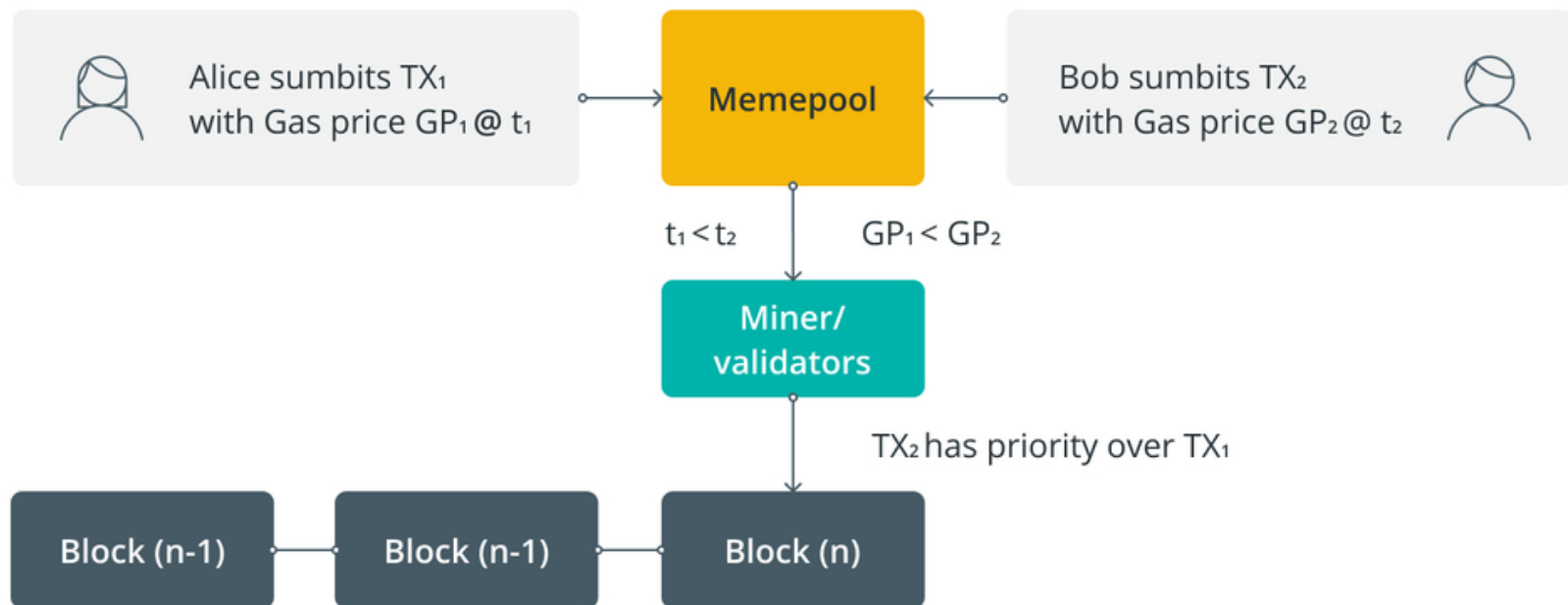
Account A: Perform some transactions

Account B: Pay ETH to Block Proposer to compensate for gas

# MEV EXAMPLES

## GENERALIZED FRONT RUNNING

### Technical representation of frontrunning



# MEV EXAMPLES

## DEX ARBITRAGE

### DEX Arbitrage

**DEX 1**



1 BTC = 14 ETH

Sell BTC

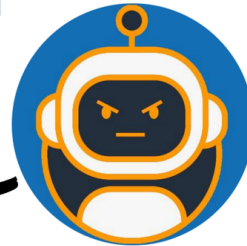
**DEX 2**



1 BTC = 13 ETH

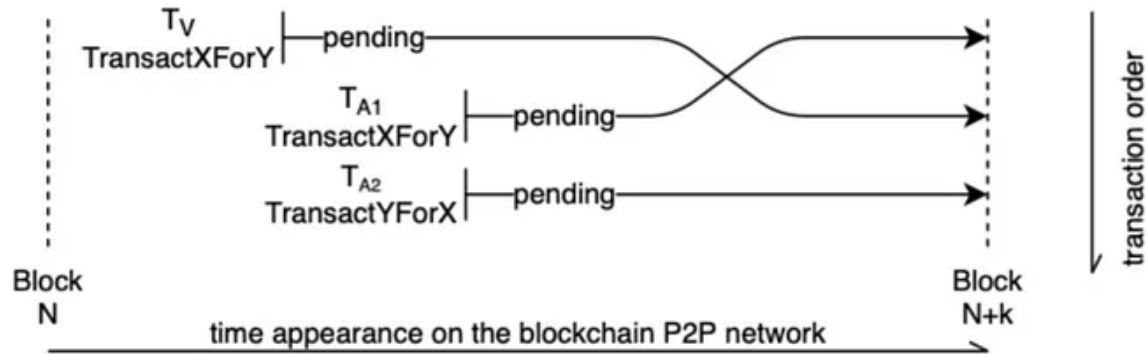
Buy BTC

BOT

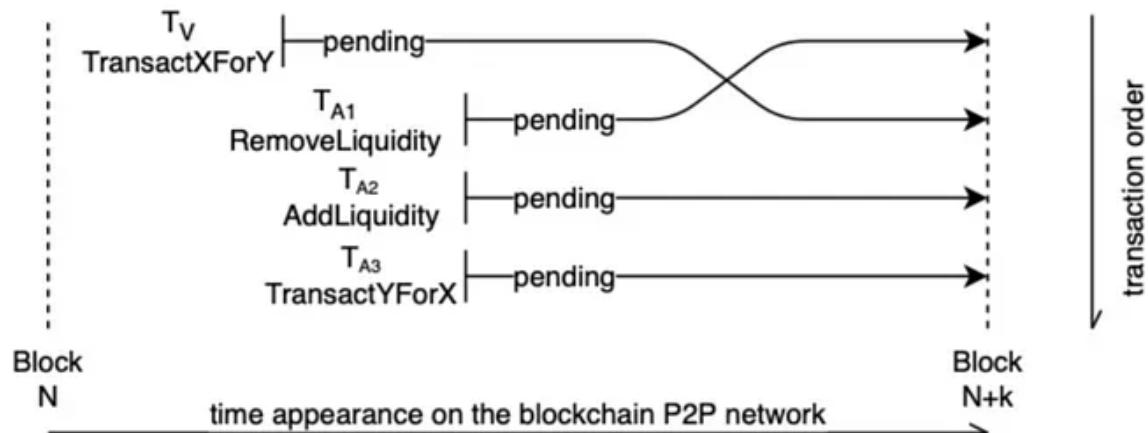


# MEV EXAMPLES

## SANDWICH ATTACK



Sandwich attack strategy 1, when liquidity taker attacks taker



Sandwich attack strategy 2, when liquidity provider attacks taker

# MEV EFFECTS

- Speeds up Validator centralization due to large validator staking pools will have more resources to invest into MEV extraction
- Creation of Permissioned Mempools (called Dark Pools if large enough)
- Priced Gas Auctions
- Users will use blockchains with cheaper gas
- Proposer-Builder Separation (MEV Boost), with Commit Reveal Scheme




## Key Takeaways

- The rise of Flashbots and other MEV-Boost relays, which reorder transactions within Ethereum blocks to squeeze out profits, has come with unintended consequences.
- Flashbots, the largest MEV-Boost relay, refuses to process any transaction related to mixing protocol Tornado Cash.
- This places Ethereum under the threat of censorship, as more than 51% of the network's blocks are being produced by MEV-Boost relays that refuse to process certain transactions.

Sept 2023

<https://cryptobriefing.com/51-of-ethereum-blocks-can-now-be-censored-its-time-for-flashbots-to-shut-down/>

# FLASHBOT HACK




Welcome to the Flashbots forum ⚡🤖

Please start by reading our [code of conduct](#) & [introduce yourself](#) before joining the conversation!

## Post mortem: April 3rd, 2023 mev-boost relay incident and related timing issue

The Flashbots Ship mev-boost

 bert mate Apr 4

### Summary

On April 3rd, 2023, a malicious proposer exploited the [ultra sound relay](#) <sup>116</sup> through a vulnerability in the [open sourced mev-boost-relay implementation](#) <sup>72</sup> maintained by Flashbots to steal ~\$20M from multiple sandwich bots. The attack was possible because of a vulnerability in the majority of mev-boost relays ([mev-boost-relay](#) <sup>72</sup>, [Dreamboat](#) <sup>43</sup>) detailed below which has [since been patched](#) <sup>42</sup>. In following up to this event, a related timing attack was identified and mitigated, although more research is needed to understand if this mitigation is worth its negative externalities.

mev-boost is an open-source [proposer-builder separation \(PBS\)](#) <sup>54</sup> protocol that allows proposers to sell their blockspace to an open market of specialized actors called builders. Builders compete to create the most valuable block possible, mostly by aggregating many individual MEV searchers' bundles of transactions.

mev-boost works through a commit and reveal scheme where proposers commit to blocks created by builders without seeing their contents, by signing block headers. Only after a block header is signed are the block body and corresponding transactions revealed. A trusted third party called a relay facilitates this process. mev-boost is designed to allow block builders to

- Summary
- Timeline
- Mitigations
- Looking forward
- Thank you

Sept 2023

<https://collective.flashbots.net/t/post-mortem-april-3rd-2023-mev-boost-relay-incident-and-related-timing-issue/1540>

## Resources Used:

<https://coinsbench.com/about-evm-opcode-gas-ethereum-accounts-9f0896f09d04>

<https://ethereum.org/>

<https://hardhat.org/>

<https://docs.ethers.io/v5/>

<https://www.openzeppelin.com/>

[https://takenobu-hs.github.io/downloads/ethereum\\_evm\\_illustrated.pdf](https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf)

<https://www.skillsoft.com/>