Ethereum Basics

Öz, B., Hoops, F., Gebele, J., Gallersdörfer, U., & Matthes, F. (2024). “Blockchain-based Systems Engineering”. Lecture Slides. TU Munich.

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

1. Ecosystem
   - Historical Overview
   - Crowdsale Statistics
   - Technical Papers
   - Foundations
   - Network Metrics

2. System Architecture
   - Concept of a World Computer
   - EVM
   - Accounts
   - Block Header
   - Smart Contracts

3. The Merge
   - Ethereum’s Vision
   - The Beacon Chain
   - New Architecture
   - Summary of Changes
   - What’s Next

Slides in the section “The Merge” are influenced from the slides of Burak Öz - "What does “The Merge” mean for Ethereum?”, Blockchain-Stammtisch Donnersblock, 2022.

History

Vitalik Buterin at a Techcrunch conference

- In **November 2013**, Vitalik Buterin started working on the **first version** of the Ethereum **white paper**
- Buterin made the **first public announcement** of Ethereum on **24th of January 2014** at the Bitcoin conference in Miami
- On the **7th of July 2014**, Buterin **announced the start** of the **public crowd sale**
- The **sale lasted 42 days** until 2nd of September
  - For the first 14 days the price was 1 BTC for 2000 ETH
  - After that period the price went up to 1 BTC for 1337 ETH
- In total, **~60 million Ether** were sold in exchange for **31,591 Bitcoins**
  - Worth around 18.5 million USD at that time
  - Used by the Ethereum foundation

History of Ethereum https://ethereum.org/en/history/

Fun Fact

"On that day I realized what horrors centralized services can bring,..”

Vitalik Buterin wrote on his website's bio that he was driven to create decentralized money because his World of Warcraft character was cruelly nerfed by Blizzard. He said “I saw everything to do with either government regulation or corporate control as just being plain evil. And I assumed that people in those institutions were kind of like Mr. Burns, sitting behind their desks saying, ‘Excellent. How can I screw a thousand people over this time.’”

VITALIK BUTERIN

Zug, Switzerland

Visit my website

#Copy-paste bio for conferences, articles, propaganda materials, etc:

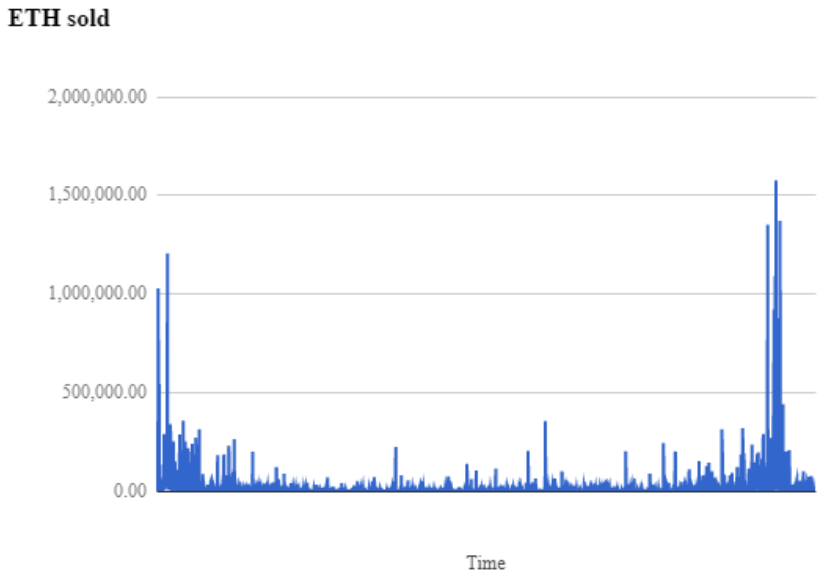
http://vitalik.ca/files/quickbio.txt

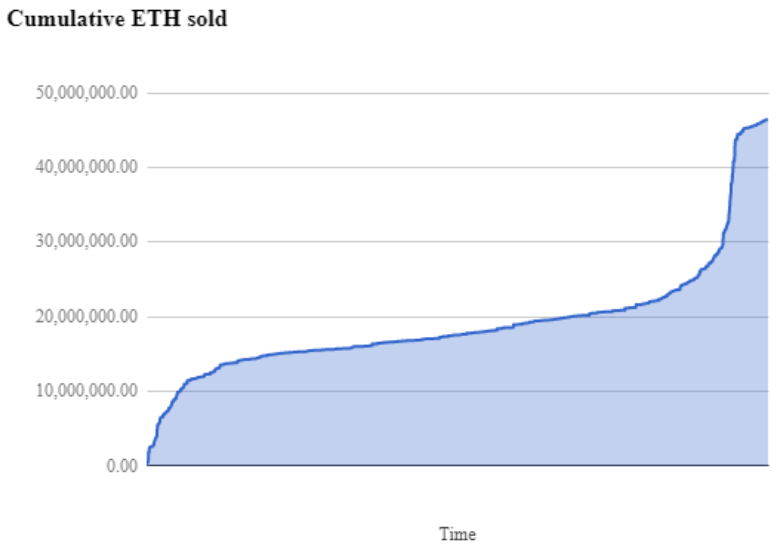
#More interesting bio

I was born in 1994 in Russia and moved to Canada in 2000, where I went to school. I happily played World of Warcraft during 2007-2010, but one day Blizzard removed the damage component from my beloved warlock's Siphon Life spell. I cried myself to sleep, and on that day I realized what horrors centralized services can bring. I soon decided to quit.

Crowdsale Statistics

First 14 Day Period





- Around **48 million ETH** were **sold during** the **first price period** of 14 days
- **Most ETH** were **sold** at the **beginning** and the **end** of the period
- **Biggest single purchase** during the first period was **500 BTC** which equals **1,000,000 ETH**
- **Smallest purchase** was **0.01 BTC**
- **43.6%** bought **2000 or more ETH**
- **0.8%** bought **200,000 or more ETH**
- **11,901,464.23948 ETH** to the **development team** (https://etherscan.io/address/0x5abfec25f74cd88437631a7731906932776356f9)

White Paper

Ethereum Whitepaper

This introductory paper was originally published in 2014 by Vitalik Buterin, the founder of Ethereum, before the project's launch in 2015. It's worth noting that Ethereum, like many community-driven, open-source software projects, has evolved since its initial inception.

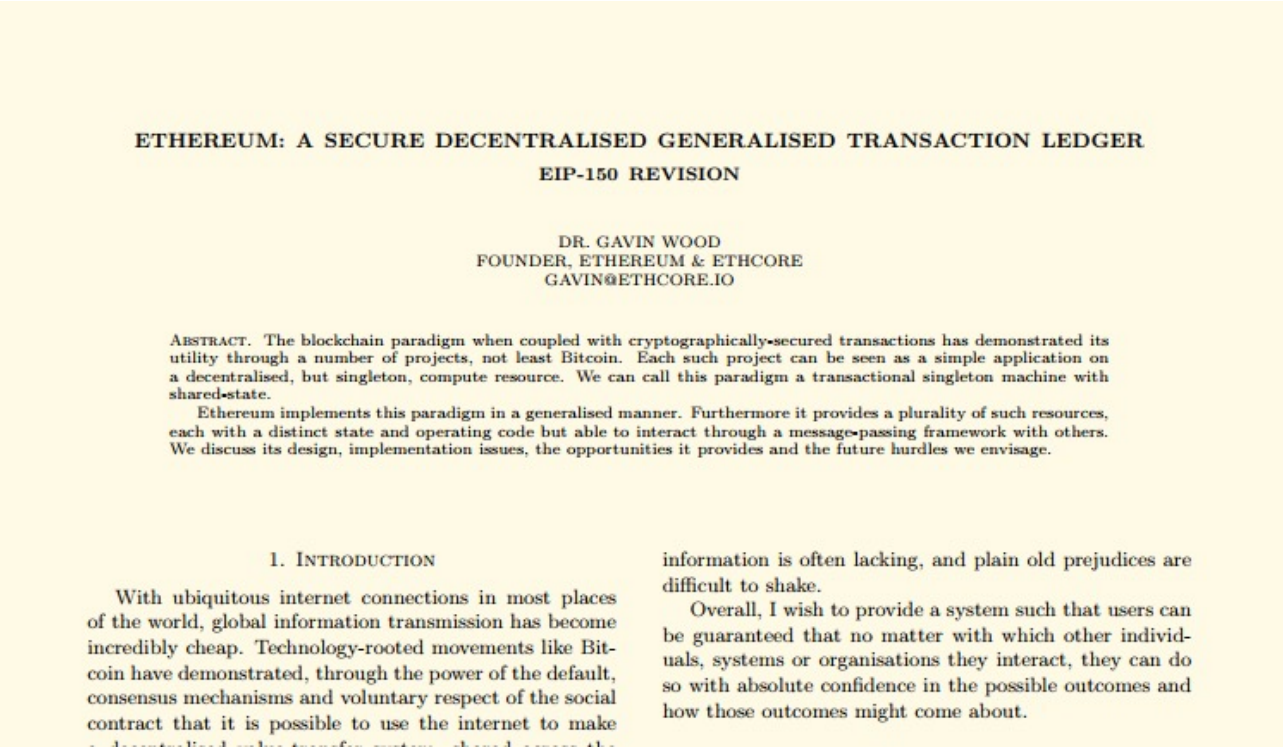
While several years old, we maintain this paper because it continues to serve as a useful reference and an accurate representation of Ethereum and its vision. To learn about the latest developments of Ethereum, and how changes to the protocol are made, we recommend this guide.

Researchers and academics seeking a historical or canonical version of the whitepaper [from December 2014] should use this PDF. ↗

A Next-Generation Smart Contract and Decentralized Application Platform

- **First draft** was written by Vitalik Buterin himself (**2013**)
- Contains **high level descriptions of** Ethereum's **core functionalities**
- **Living document** and **regularly updated** by Ethereum core developers (not only Buterin!)
- **Extensive summary** of the Ethereum platform and technology
- **Current version**: https://ethereum.org/en/whitepaper/

Yellow Paper



ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

EIP-150 REVISION

DR. GAVIN WOOD
FOUNDER, ETHEREUM & ETHCORE
GAVIN@ETHCORE.IO

ABSTRACT. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated its utility through a number of projects, not least Bitcoin. Each such project can be seen as a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactional singleton machine with shared-state.

Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resources, each with a distinct state and operating code but able to interact through a message-passing framework with others. We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

1. INTRODUCTION

With ubiquitous internet connections in most places of the world, global information transmission has become incredibly cheap. Technology-rooted movements like Bitcoin have demonstrated, through the power of the default, consensus mechanisms and voluntary respect of the social contract that it is possible to use the internet to make

information is often lacking, and plain old prejudices are difficult to shake.

Overall, I wish to provide a system such that users can be guaranteed that no matter with which other individuals, systems or organisations they interact, they can do so with absolute confidence in the possible outcomes and how those outcomes might come about.

Ethereum yellow paper: https://ethereum.github.io/yellowpaper/paper.pdf

- **Published** in **April 2014** by **Dr. Gavin Wood**
- **Dr. Gavin Wood** is still listed as **the only author**
- Defines the **technical specification of Ethereum**
- **Very detailed**, contains mathematical function definitions and byte code mappings
- **Required to implement** a full node
- **Only updated** when **errors** are found **or** the **specification changes**

Foundation

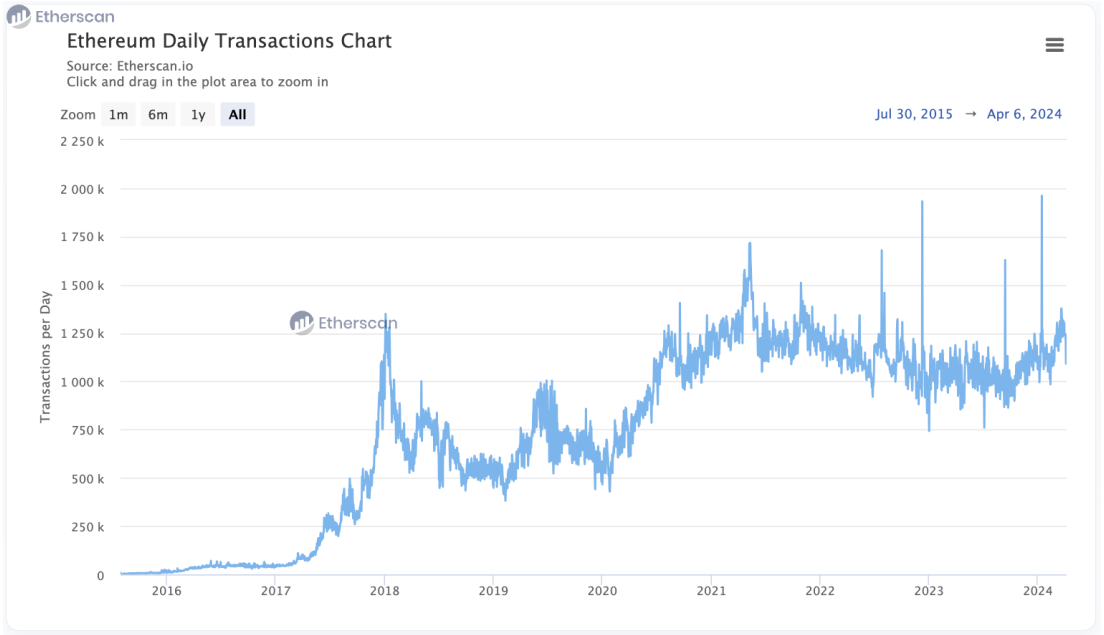
*"The Ethereum Foundation's **mission is** to **promote and support Ethereum platform** and **base layer research**, **development** and **education** to bring decentralized protocols and tools to the world that empower developers **to produce next generation decentralized applications** (dapps), and together build a more globally accessible, more free and more trustworthy Internet."*

- **Founded** in **June 2014 in Zug**, Switzerland
- **Non-profit** organization
- **Foundation council** consists of **Vitalik Buterin** and **Patrick Storchenegger** who is responsible for all legal affairs
- **Owns** (or had owned) at least **31,591 Bitcoins** funding capital due to the crowdsale

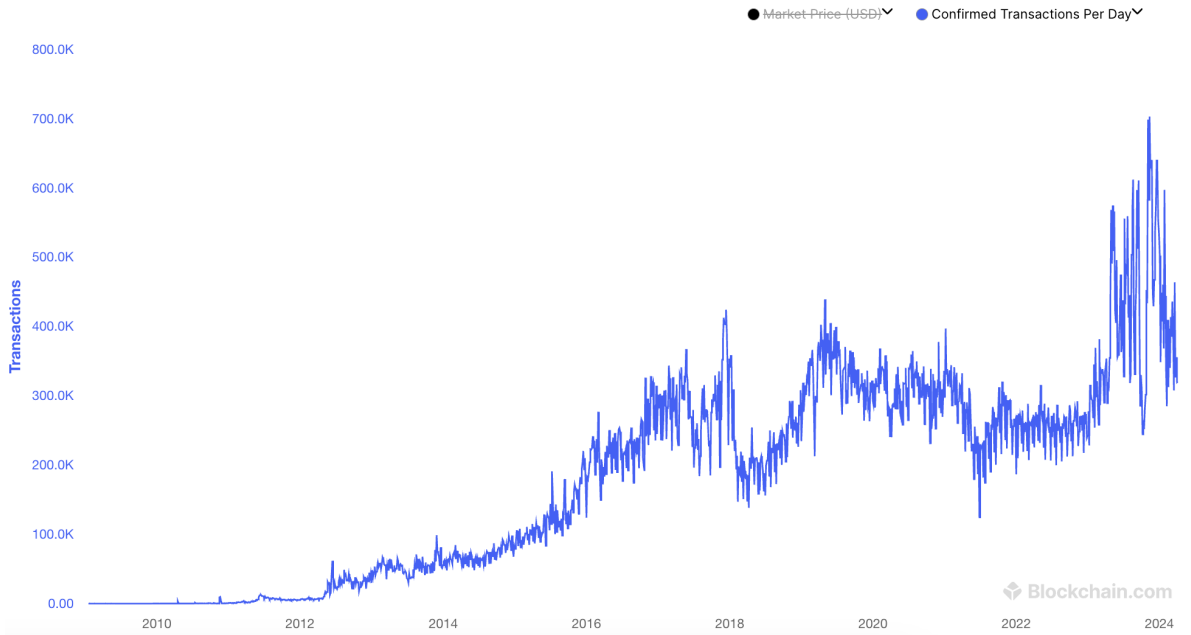
Official website of the Ethereum foundation: https://www.ethereum.org/foundation

Like Bitcoin, Ethereum is in Productive Use

Transactions per Day



- **1,961,144** peak of transactions per day
- Currently almost **4x** the number of **BTC transactions** each day

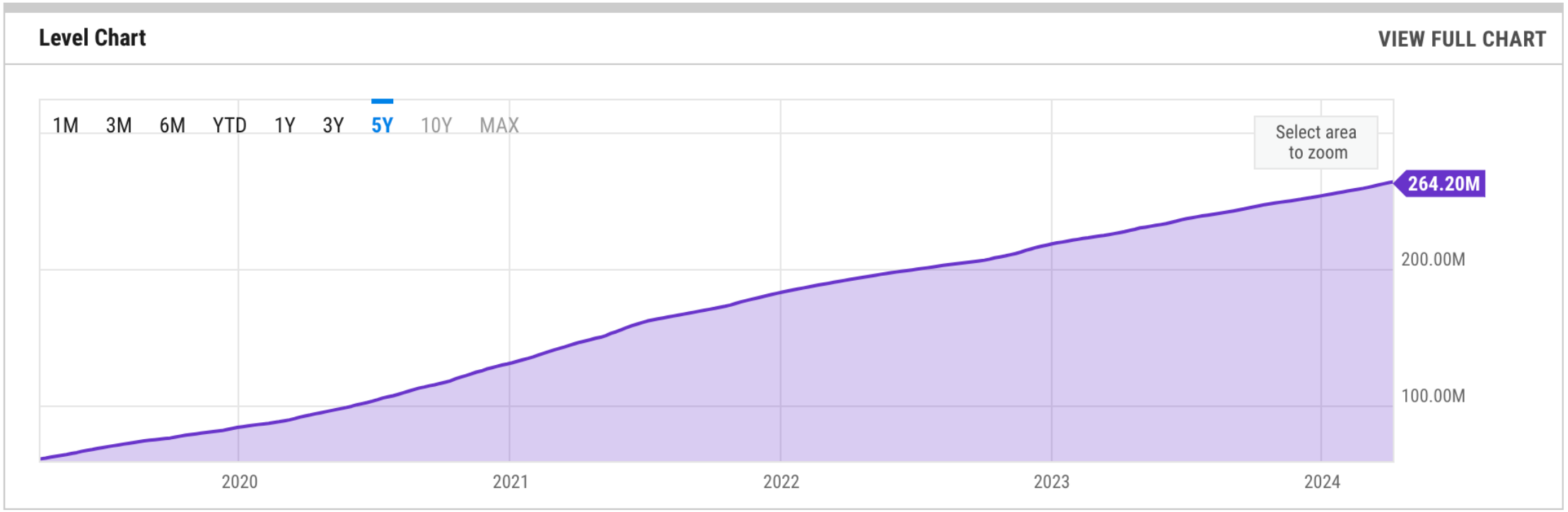


- **710,579** peak of transactions per day

Ethereum transaction chart by Etherscan: https://etherscan.io/chart/tx
Bitcoin transaction chart by Blockchain.com: https://www.blockchain.com/charts/n-transactions

Like Bitcoin, Ethereum is in Productive Use

Unique Addresses

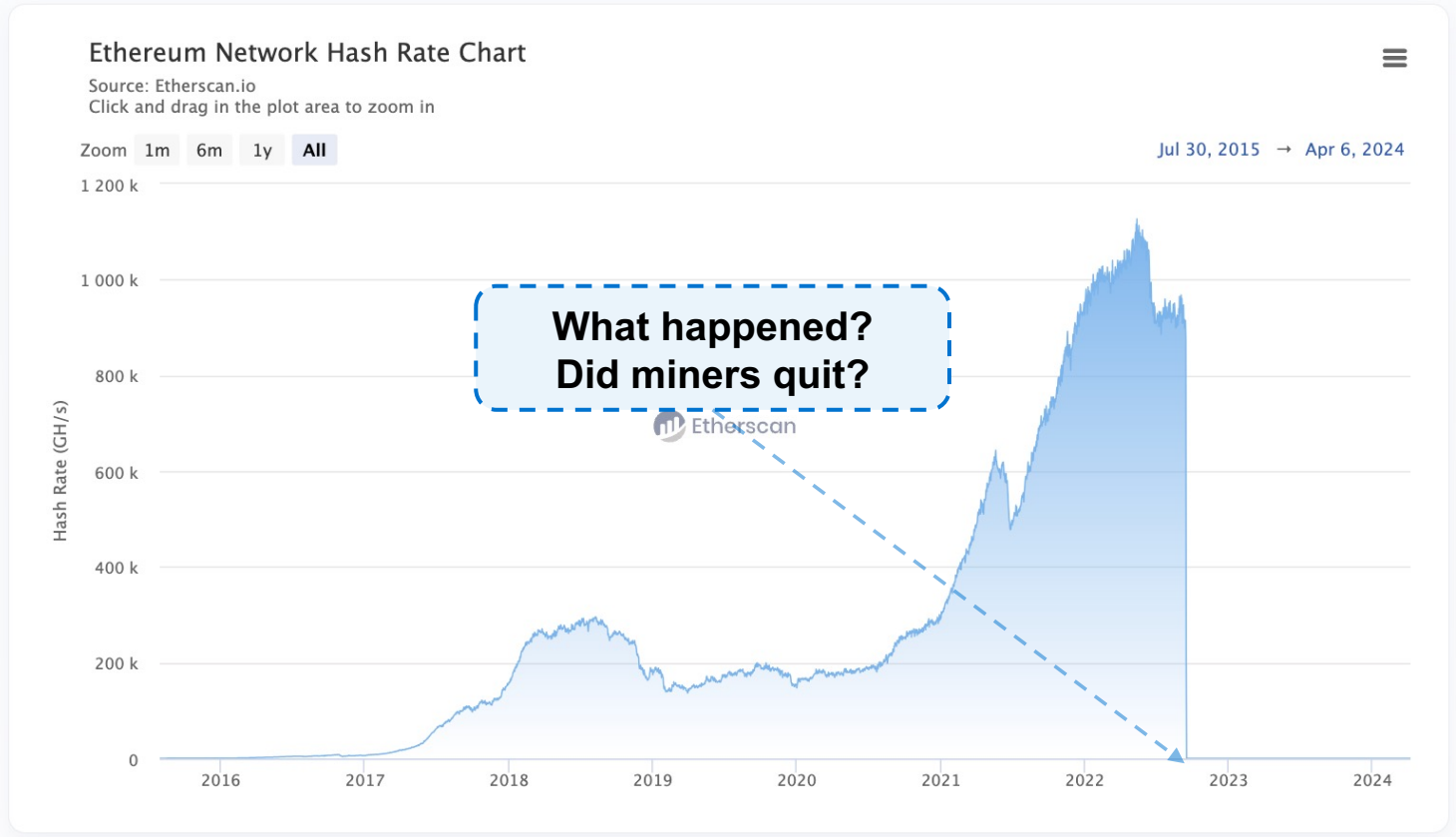


- Currently around **264 million unique addresses**

Ethereum unique address chart by Y Charts: https://ycharts.com/indicators/ethereum_cumulative_unique_addresses

Like Bitcoin, Ethereum is in Productive Use

~~Also Uses Proof of Work~~ Proof-of-Stake Since The Merge



- In May **2022**, the network **hash rate** reached all time high at **~1,126,674 GH/s**
  - Mostly **GPUs** were used **for hashing (ASIC-resistant PoW)**
- In August **2022,** estimated **annual electricity consumption** reached all time high at **~ 93.98 TW/h**[1]
  - As of April 2024, estimated annual electricity consumption is **0.01 TW/h**

The Merge

- In September 2022, Ethereum stopped running **Proof-of-Work**, and switched to a **Proof-of-Stake** based consensus mechanism.
- We will see the details of this transition, known as "**The Merge**", in the upcoming slides.

[1] Bitcoin's annual electicity consumption went up as high as 204.5 TW/h

Energy consumption estimation by Digiconomist: https://digiconomist.net/ethereum-energy-consumption
Ethereum network hash rate chart by Etherscan: https://etherscan.io/chart/hashrate

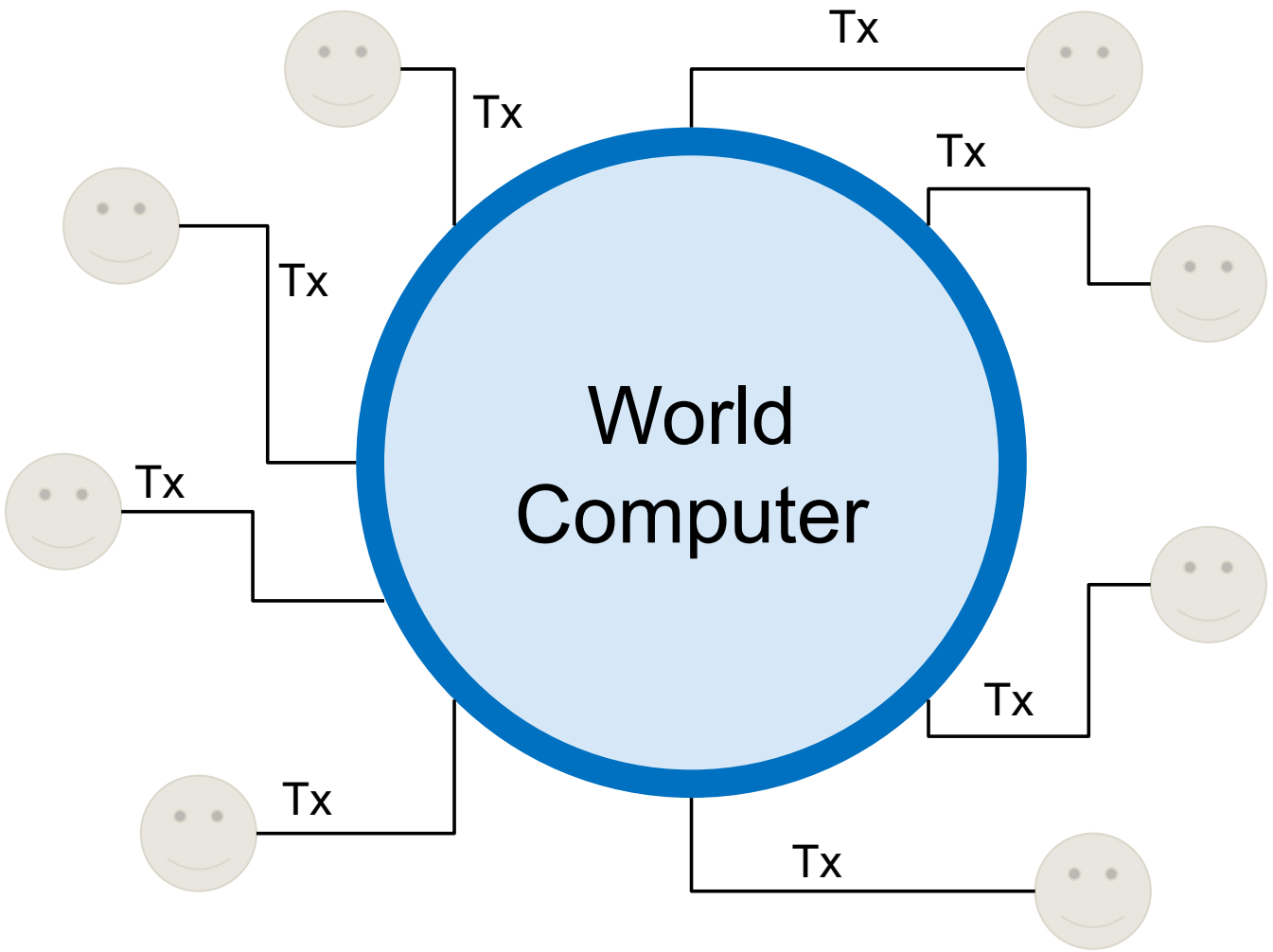
Outline

1. Ecosystem
   - Historical Overview
   - Crowdsale Statistics
   - Technical Papers
   - Foundations
   - Network Metrics
2. System Architecture
   - Concept of a World Computer
   - EVM
   - Accounts
   - Block Header
   - Smart Contracts
3. The Merge
   - Ethereum’s Vision
   - The Beacon Chain
   - New Architecture
   - Summary of Changes
   - What’s Next

Slides in the section “The Merge” are influenced from the slides of Burak Öz - "What does “The Merge” mean for Ethereum?”, Blockchain-Stammtisch Donnersblock, 2022.

World Computer

State Machine

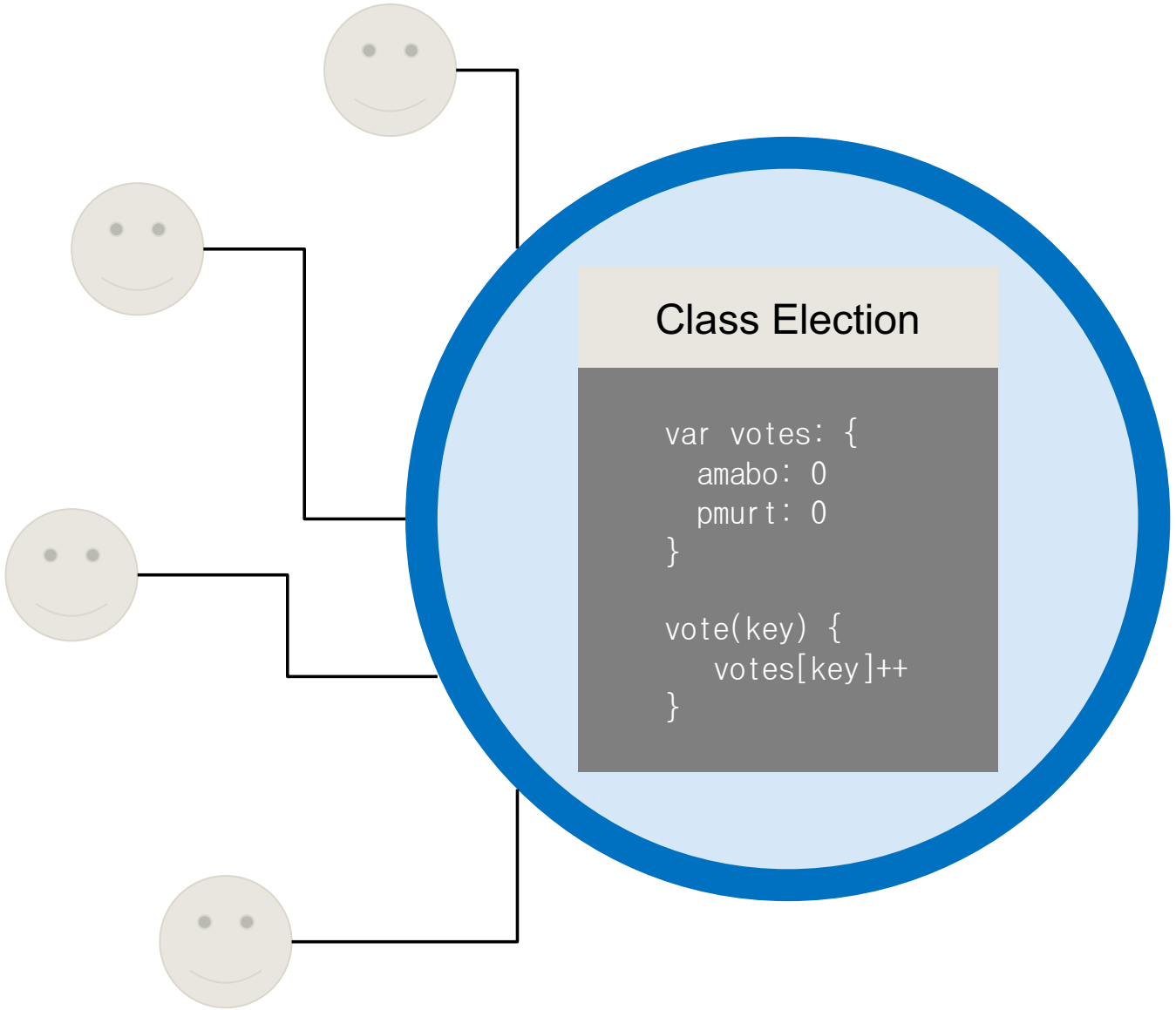


Tx = Transaction

Properties

- **All participants** are **using** the **same computer**
- Users issue **transactions to call programs** on the computer
- **Everyone shares** the same **resources** and **storage**
- The **computer has no explicit**, single **owner**
- **Using** the computer's **resources costs money**

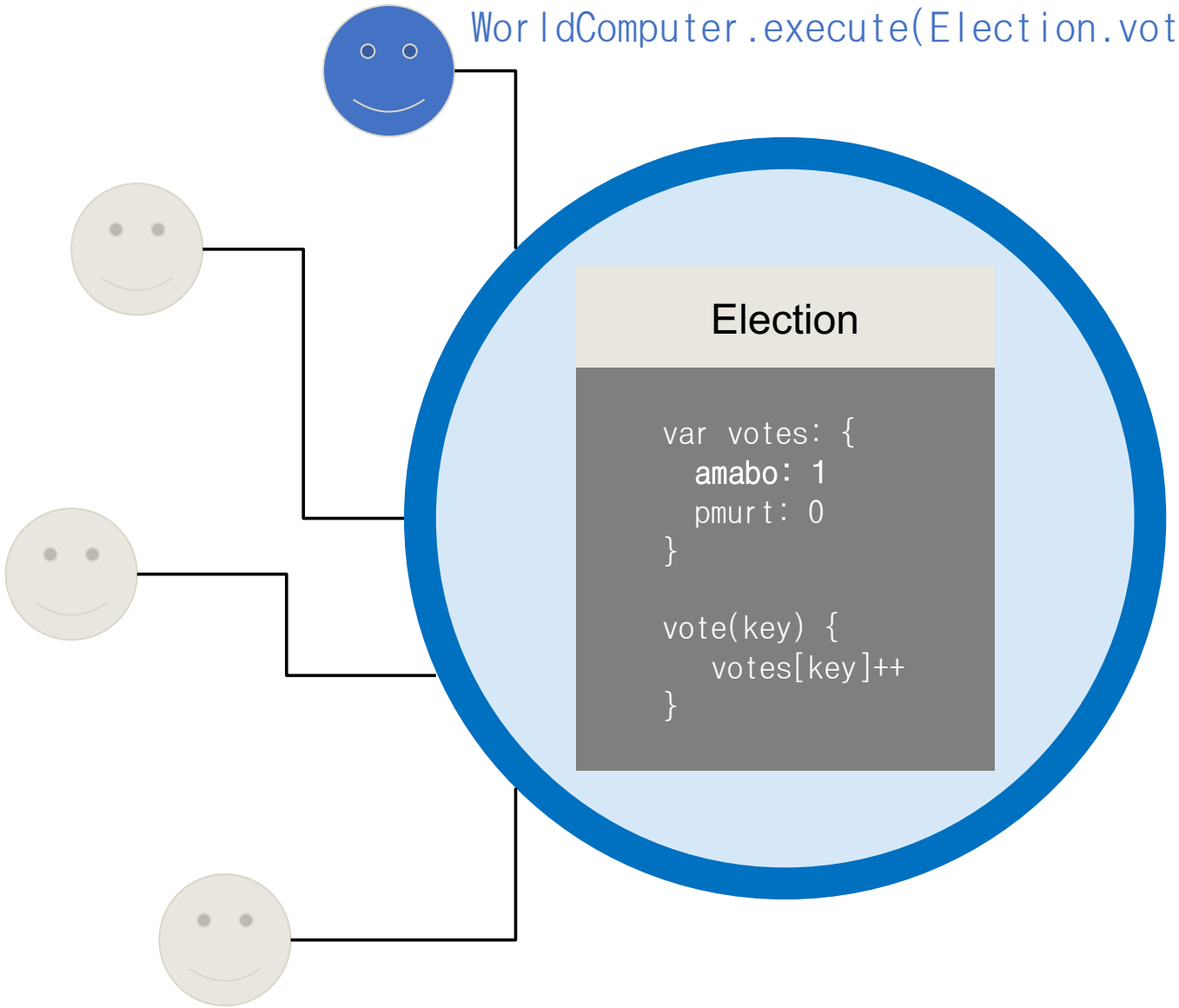
Election Example Using a World Computer



State of the world

State 0 (initial)
(nothing happened yet)

Election Example Using a World Computer (cont.)

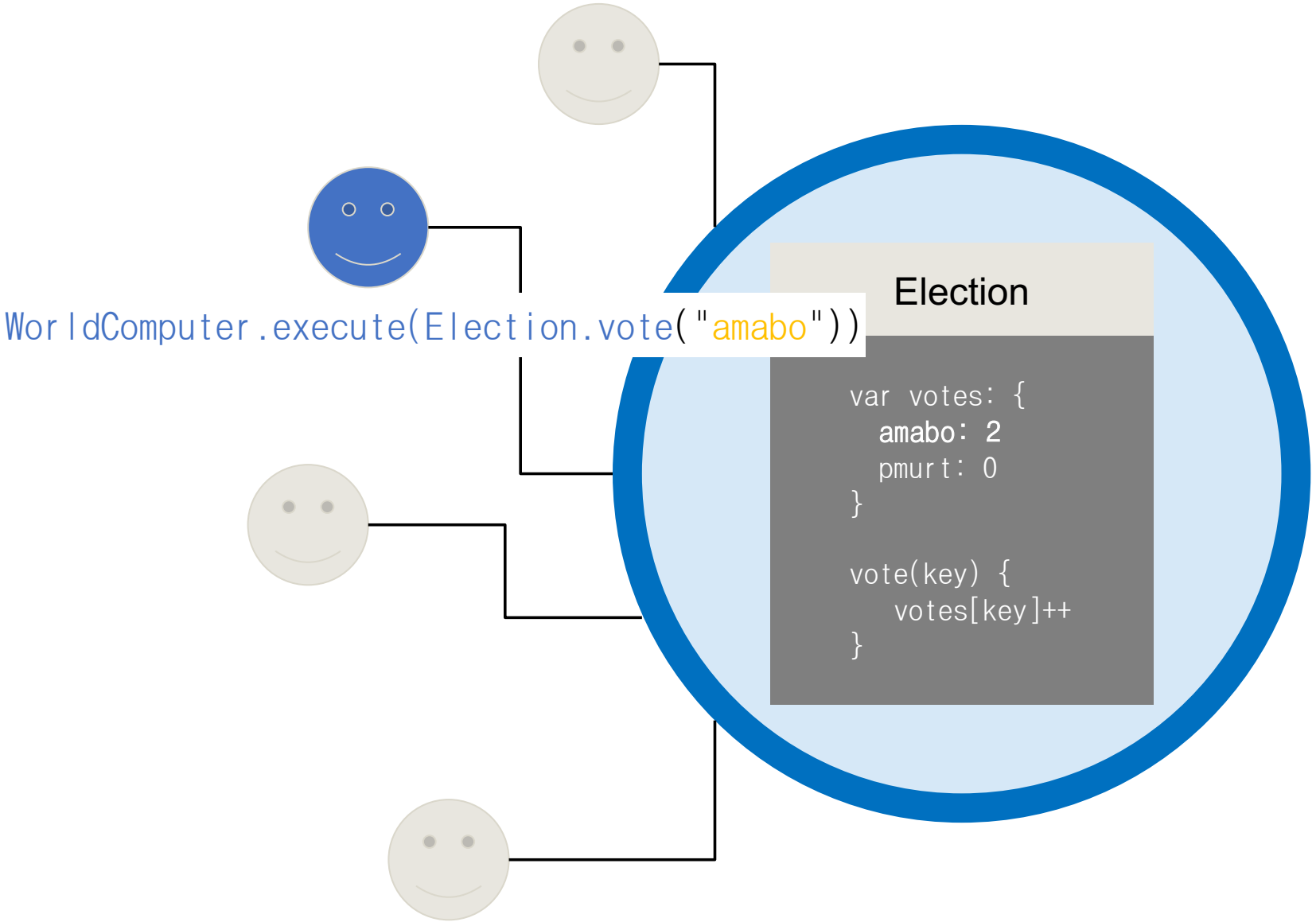


State of the world

State 0 (initial)
(nothing happened yet)

State 1 {amabo:1, pmurt:0}
(vote for amabo)

Election Example Using a World Computer (cont.)



State of the world

State 0 (initial)
(nothing happened yet)

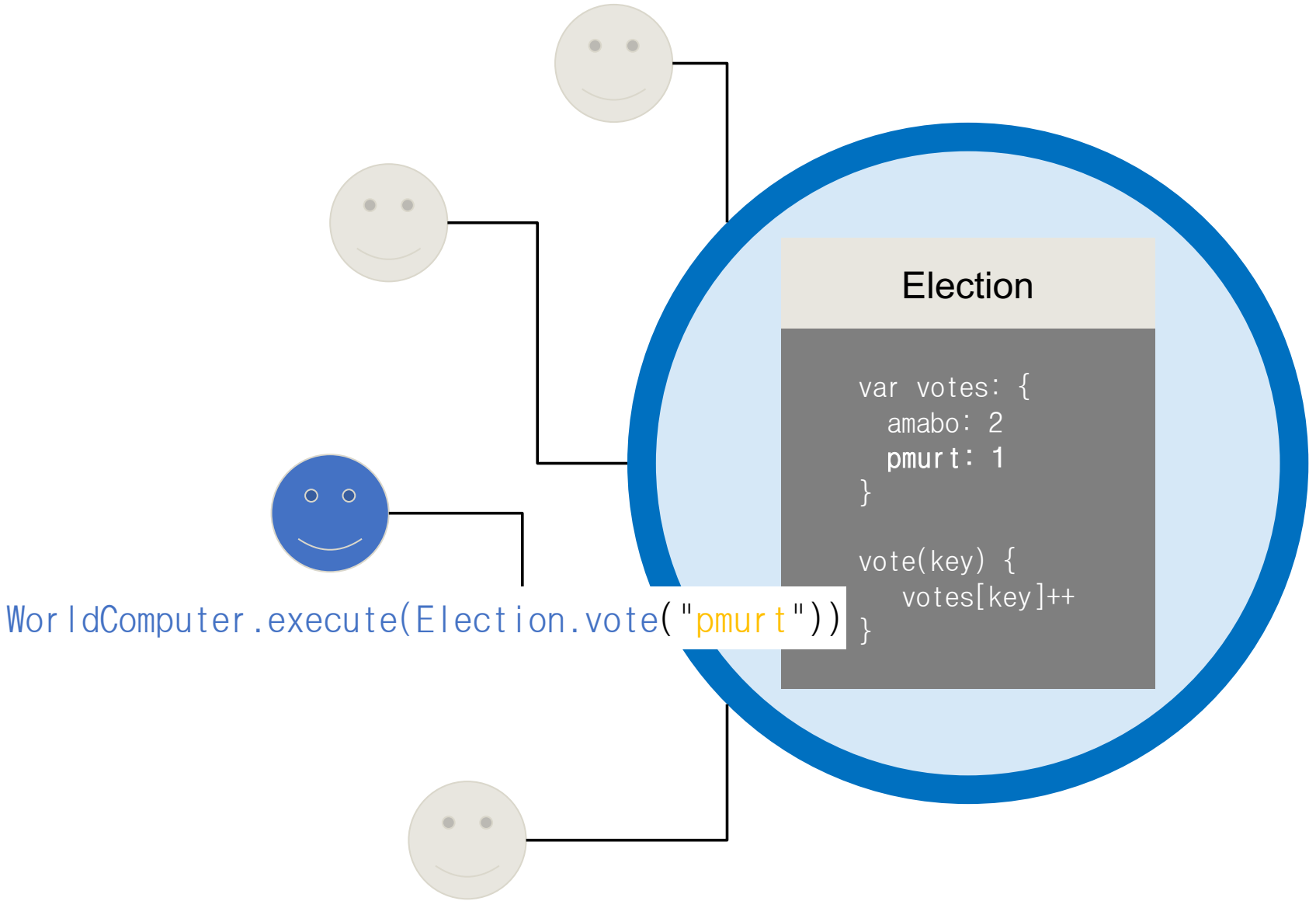
↓

State 1 {amabo:1, pmurt:0}
(vote for amabo)

↓

State 2 {amabo:2, pmurt:0}
(vote for amabo)

Election Example Using a World Computer (cont.)



State of the world

State 0 (initial)
(nothing happened yet)

↓

State 1 {amabo:1, pmurt:0}
(vote for amabo)

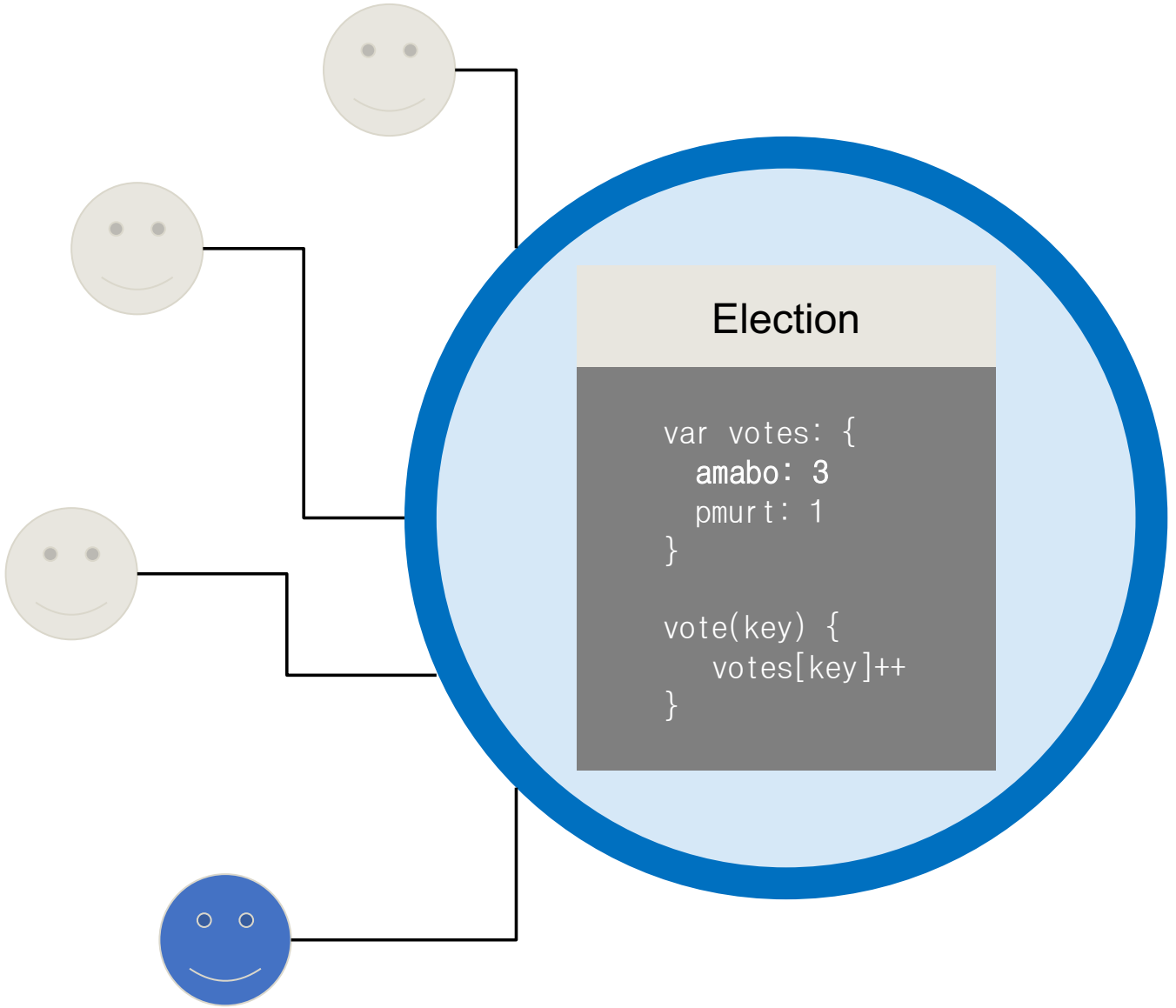
↓

State 2 {amabo:2, pmurt:0}
(vote for amabo)

↓

State 3 {amabo:2, pmurt:1}
(vote for pmurt)

Election Example Using a World Computer (cont.)



State of the world

State 0 (initial)
(nothing happened yet)

↓

State 1 {amabo:1, pmurt:0}
(vote for amabo)

↓

State 2 {amabo:2, pmurt:0}
(vote for amabo)

↓

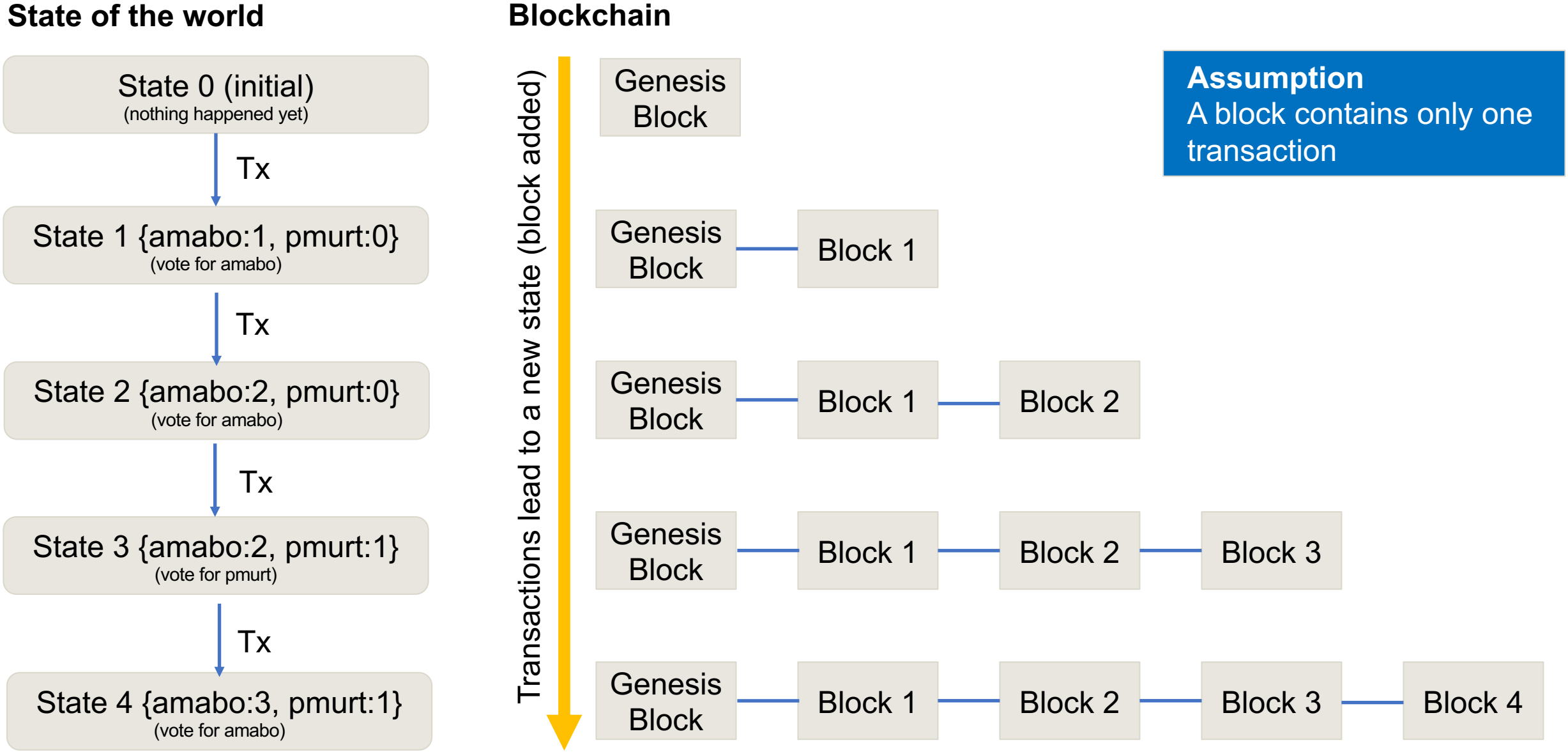
State 3 {amabo:2, pmurt:1}
(vote for pmurt)

↓

State 4 {amabo:3, pmurt:1}
(vote for amabo)

WorldComputer.execute(Election.vote("amabo"))

Blockchain as a State Machine



Assumption
A block contains only one transaction

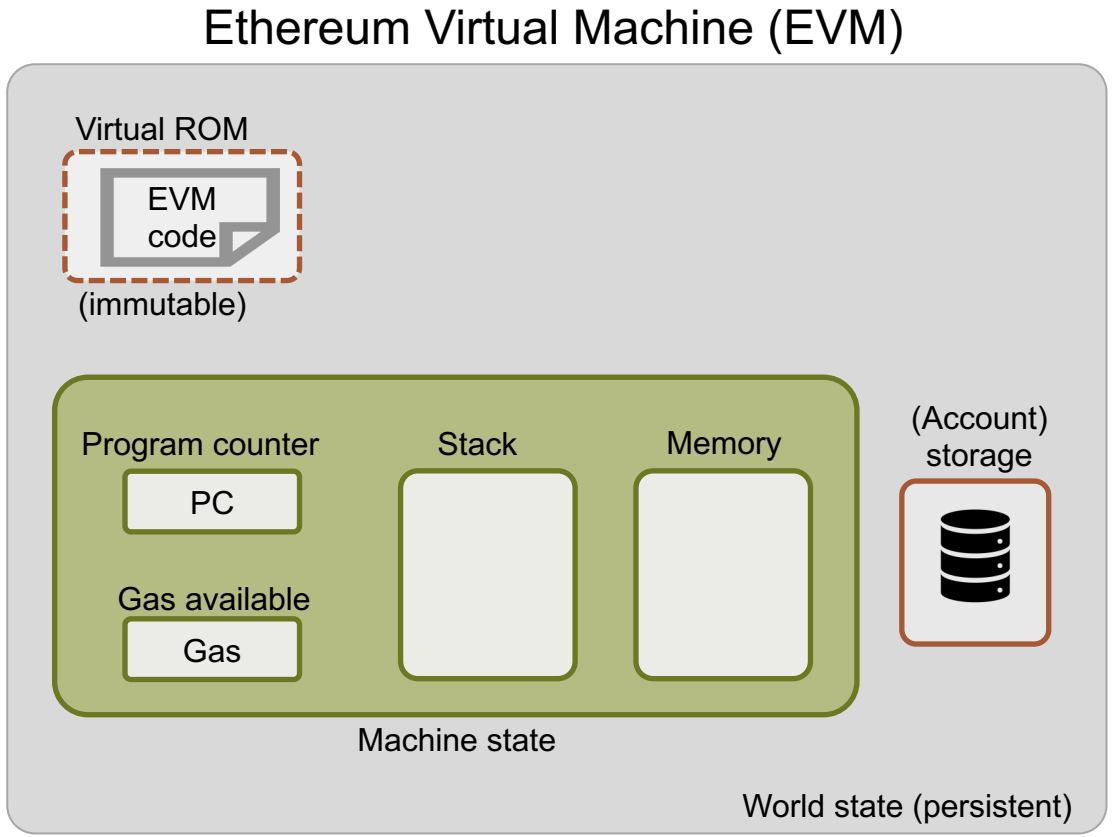
Ethereum Virtual Machine

What is a virtual machine in a blockchain?

- Virtual machines (VM) are mechanisms for creating instances of software that imitate real machines.

*What is the **Ethereum Virtual Machine** (EVM)?*

- EVM was the first virtual machine to be placed on a blockchain network, allowing to conduct calculations on the blockchain in **runtime**.
- It has a **stack-based architecture** for executing programs on the Ethereum blockchain.
- Running operations on EVM have associated costs denominated in **gas**.[1]
- **World state** refers to all accounts and smart contracts with their respective storage.



[1] We will see more about the concept of gas and EVM operations in the upcoming slides and exercise sessions. This representation of EVM is taken from: https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

Ethereum Virtual Machine

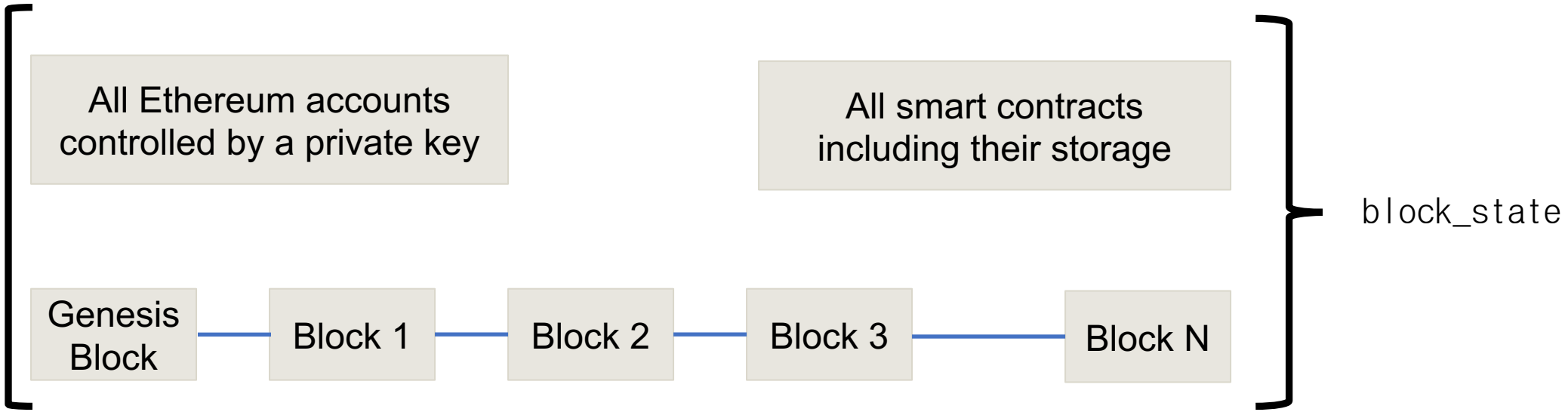
State Machine

Unlike Bitcoin where state refers to UTXOs owned by addresses, in Ethereum, state represents accounts and smart contracts with their storage. Hence, Ethereum is more than a distributed ledger**,** it is a **distributed state machine**.[1]

- **EVM** specifies an **execution model for state changes** on Ethereum.

Formally, **EVM** can be specified by the **following tuple**:
(block_state, transaction, message, code, memory, stack, pc, gas)

- The *block_state* represents the **world state** of the whole blockchain including **all accounts, contracts and storage**.



[1] See Ethereum documents for more about this: https://ethereum.org/en/developers/docs/evm/

Accounts on Ethereum

Ethereum has an **account-based ledger** where each **distinct address** represents a separate, unique account.

→ Remember that Bitcoin uses a *transaction-based ledger* model!

Ethereum supports two types of accounts:

Externally Owned Accounts (EOA): *controlled by private keys*

- Accounts that are controlled by a private key do not have any code stored on the blockchain
- **Default wallet of a user**[1].
- They can sign transactions, issue smart contract functions calls, and send Ether.
- The **origin of any transaction** is always an account controlled by a private key.

Smart Contract Accounts: *controlled by their code*

- Smart contracts[2] are treated as **account** entities with their **own**, unique **address**.
- Contracts[3] **can send messages** to other accounts, both externally controlled and smart contracts.
- They **can't issue a transaction** themselves**.**
- They have a **persistent internal storage** for reading and writing.

[1] In this context, a user is an human being. Thus, EOAs are controlled by humans.
[2] Deeper overview on smart contracts can be found in the following slides.
[3] "Contract" is used as a short form of the term "smart contract".

Accounts on Ethereum (cont.)

On an abstract level, an Ethereum account is a **4-tuple** containing the following data:
(nonce, balance, contract_code, storage)

nonce
An increasing number that is attached to any transaction to prevent **double spending attacks**.

balance
The current balance of the account in Ether.

contract_code
The bytecode representation of the account. If no contract code is present, then the account is externally controlled.

storage
The data storage used by the account (empty by default). Only contract accounts can have their own storage.

Accounts on Ethereum (cont.)

Private key

controls

Externally Owned Account (EOA)

Address[1]

Account State	
Nonce	**132**
Balance	**0.79 ETH**
Contract Code	-
Storage	-

EVM code

controls

Smart Contract Account

Address[2]

Account State	
Nonce	**1**[3]
Balance	**2.42 ETH**
Contract Code	**0eb388e…**
Storage	**87c1d2d…**

[1] EOA addresses are public key hashes.
[2] Smart contract addresses are **not** public key hashes. They are made up from the creator's address and its nonce.
[3] A smart contract account also has a nonce, fixed to 1. Messages do not increment the nonce.
This slide is influenced from: https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

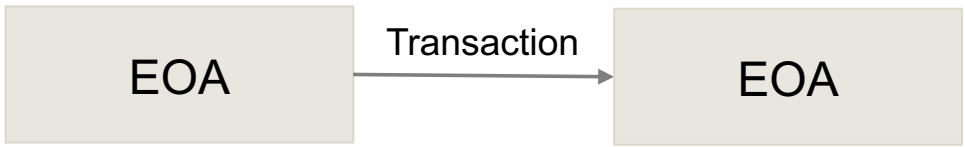
Ethereum Virtual Machine

Transaction

A **transaction** is a **signed data package** that is **always sent by an EOA**. Ethereum state transitions are triggered by transactions.

A transaction contains the following data:

- The recipient of the transaction
- A signature identifying the sender
- The amount of ether to transfer from the sender to the recipient
- An optional data field – data field is used for function call arguments (e.g. function inputs)
- A *GASLIMIT*[1] value, representing the maximum amount of gas you are willing to consume on a transaction
- A *GASPRICE*[2] value, representing the unit fee the sender pays per computational step
  - *Transaction Fee = Gas Used x GASPRICE*
- Transactions can be addressed to a smart contract or another EOA



[1] Same terminology is also used for defining the maximum amount of gas a block can use.

[2] Since August 2021, Ethereum users can issue EIP-1559 transactions where main part of the GASPRICE is calculated automatically based on the network conditions. We will cover the details of the EIP-1559 transactions in the exercise sessions.

Ethereum Virtual Machine

Message

A **message** is very similar to a transaction. Messages are only sent by contracts and **exist only virtually**.

→ Messages are not mined into a block like transactions!

A message contains:

- The sender of the message (implicit)
- The recipient of the message
- The amount of ether to transfer alongside the message
- An optional data field
- A *GASLIMIT* value

Whenever a **contract calls** a method on **another contract**, a virtual **message** is sent.
Whenever an **EOA calls** a method on a contract, a **transaction** is sent.

Ethereum Virtual Machine

Code & Memory

code

The code basically represents a smart contract as bytecode. For the EVM, a smart contract is a sequence of instructions (called *opcodes*) similar to assembly code.

Example:

PUSH1 0x60
PUSH1 0x40
MSTORE
PUSH1 0x04
CALLDATASIZE
LT
PUSH2 0x00b6
JUMPI
PUSH4 0xffffffff

memory

An infinitely expandable byte array that is non-persistent and used as temporal storage during execution.

Ethereum Virtual Machine

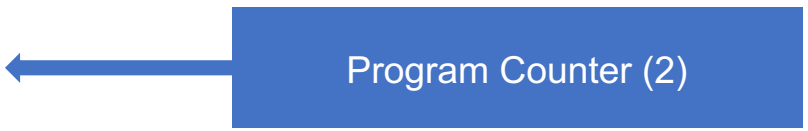
Stack & Program Counter

stack
The stack is also used as a fast, non-persistent buffer to which 32 byte values can be pushed and popped during execution.

pc
PC stands for “program counter”. The program counter is always initialized with 0 and points to the position of the current opcode instruction.

Simple Opcode Execution Example:

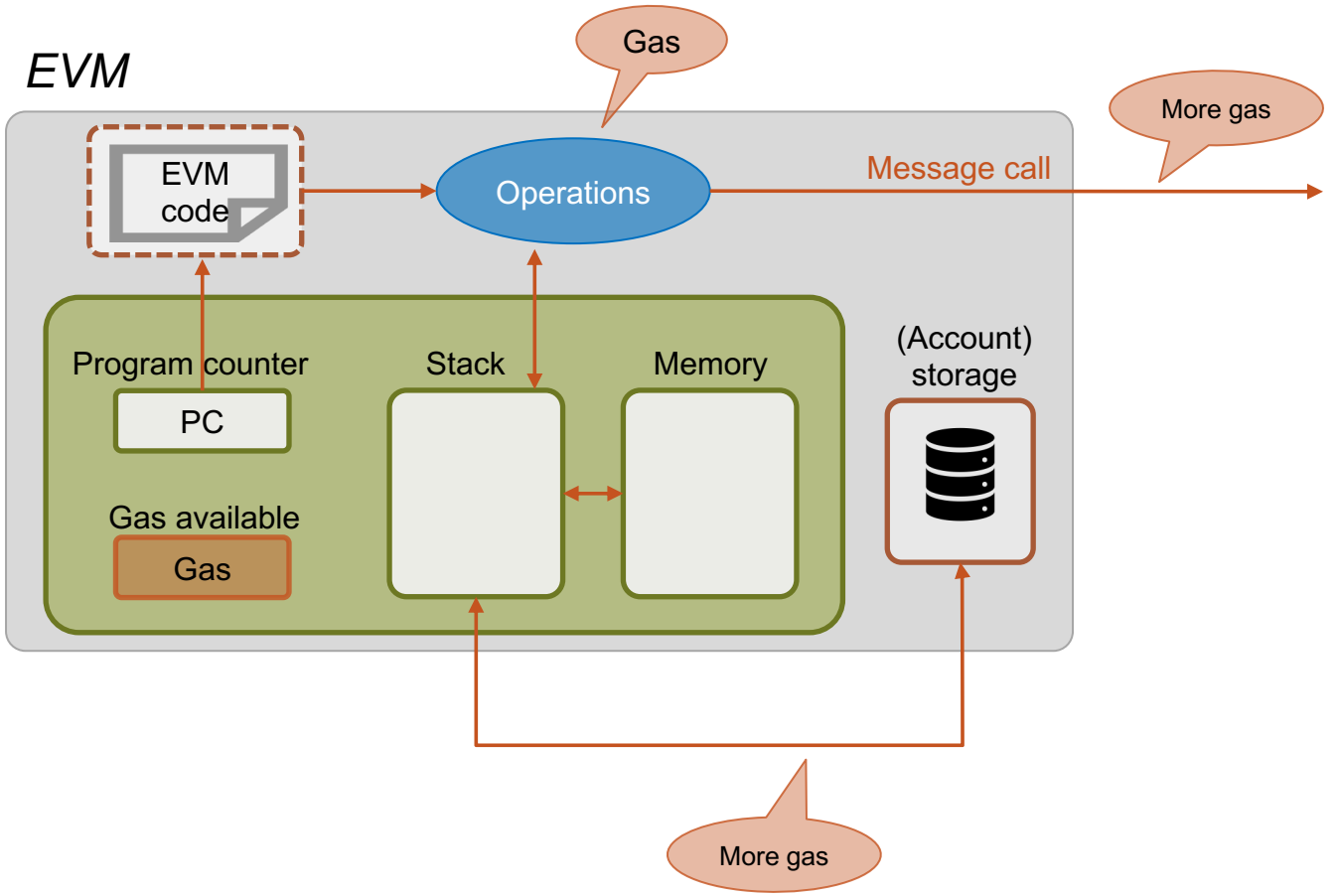
```
0       PUSH1 0x60
1       PUSH1 0x40
2       MSTORE
3       PUSH1 0x04
4       CALLDATASIZE
5       LT
6       PUSH2 0x00b6
7       JUMPI
8       PUSH4 0xffffffff
```



Ethereum Virtual Machine

Gas

- Executing EVM *opcodes* use computational resources and therefore **has an associated fee** called **gas**.
- Each opcode costs a certain amount of gas which may depend on the arguments of the operation, e.g., the number of bytes to be allocated.
- Opcodes such as SSTORE, which places data into the storage, are more expensive than simple operations like ADD or PUSH1.[1]
- The opcode for **SELFDESTRUCT (address)** uses **negative gas** as it frees up space on the blockchain.[2]



[1] EVM opcodes reference: https://www.evm.codes/?fork=cancun

[2] SELFDESTRUCT opcode is deprecated since the Shangai upgrade: https://eips.ethereum.org/EIPS/eip-4758

Brief Insight into Ethereum Blocks

Block Header

- **More complex than Bitcoin block header** due to the advanced state structure.
- Some fields like *nonce*, *mix hash*, and *uncles hash* are **not used anymore** since the merge.
- Additional fields are introduced to realize the new consensus mechanism.

State Root

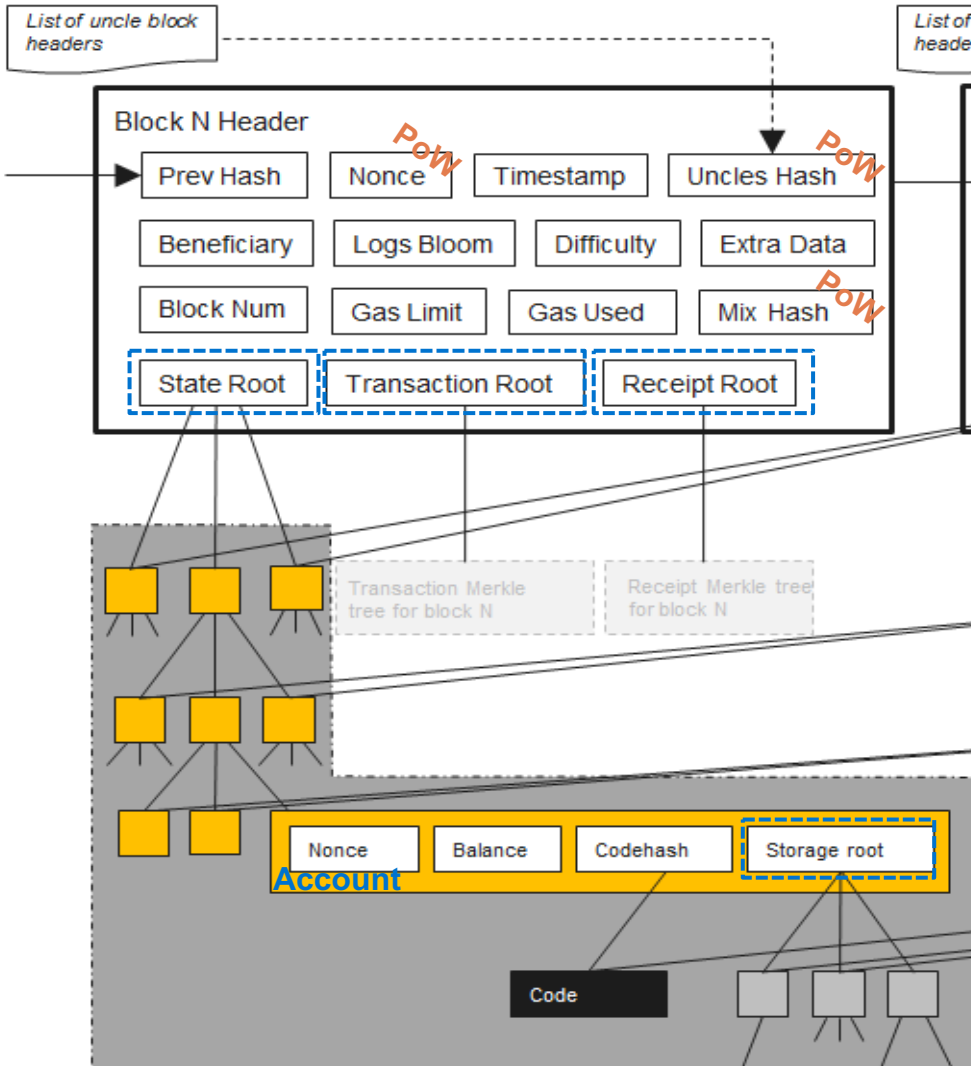
- Keccak hash of the root of the *Merkle Patricia Trie (MPT)*[1], **which stores the world state** after applying the transactions introduced in the block.
- The MPT contains a **key-value pair** for each Ethereum account.
- The **key** is the **address** of the account like a public key hash, and the **value** is an **account object** (*nonce*, *balance*, *codehash*, *storage root*).
  - **Storage root** represents the MPT of a smart contract's storage data.

Transaction Root

- Keccak hash of the root of the MPT of transactions included in the block.

Receipt Root

- Keccak hash of the root of the MPT of receipts of transactions included in the block.
- Useful for checking if a specific *event*[2] occurred during the execution of a transaction.



[1] More on Merkle Patricia Tries here: https://medium.com/shyft-network/understanding-trie-databases-in-ethereum-9f03d2c3325d
[2] Ethereum events are covered in detail in "Ethereum Smart Contracts" lecture.

To learn more about the Ethereum architecture: https://noxx.substack.com/p/evm-deep-dives-the-path-to-shadowy-5a5?s=r/
Image source: https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture

Brief Insight into Smart Contracts

- A **smart contract** is a **set of functions** that can be called by other users or contracts.
- They can be used to **execute functions, send ether or store data**.
- Each smart contract is an account holding object, i.e., **has its own address**.
- Smart contracts have some peculiarities compared to traditional software.

Security

The **development process** of smart contracts **requires special attention on security**.
Once **deployed**, a **contract** is **publicly accessible** by anyone on the network with the following information:

- Address of the smart contract
- OPCODE
- Number of public functions and their hash signature

- Furthermore, the whole transaction history is accessible (function calls + actual arguments).
- **Smart contracts** – once **deployed** – **cannot** be **changed** or **patched** anymore.

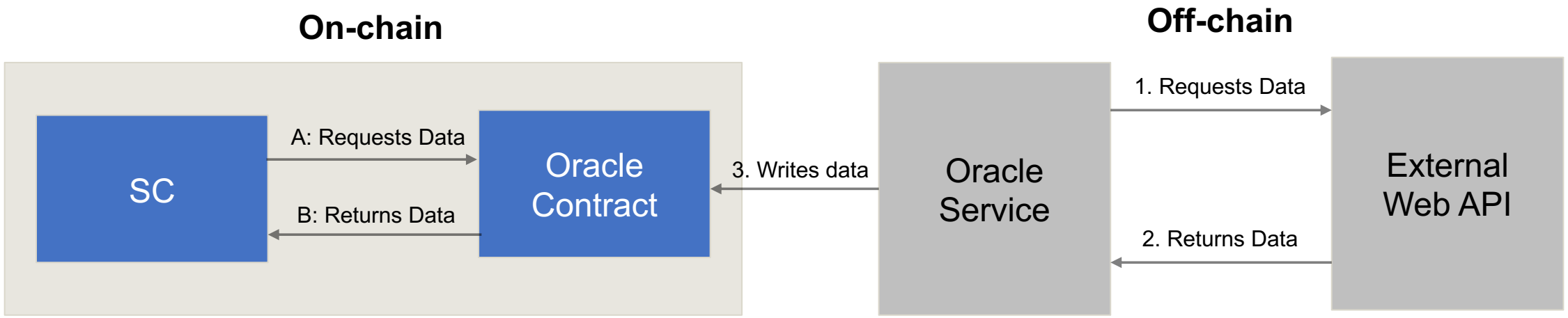
➔ All contracts deployed on the Ethereum blockchain are publicly accessible and cannot be patched.

Brief Insight into Smart Contracts as a Closed System

Smart contracts can't access any **data** from **outside** the **blockchain** on their own. There are no HTTP or similar network methods implemented to call external services. This is on purpose to **prevent non-deterministic behavior** once a function is called.

→ No function to generate random values!

Currently, the **only way** to write **external data** (e.g., weather data, traffic data etc.) to smart contracts is to **use oracles**. Oracles are basically third-party services that verify data from web services and write the data via a specially designed smart contract to the blockchain. Other smart contracts can call the oracle contract to get the data.



Brief Insight into Solidity

Usually, smart contracts are not written as a sequence of opcodes instructions directly. **Solidity** is a high-level language with a JavaScript-like syntax and the de facto **standard** for writing **Ethereum smart contracts**. However, unlike JavaScript, Solidity is **statically typed**.

Language properties

- Statically typed
- Object-oriented
- Supports inheritance
- Complex, user-defined types
- Public & private methods
- Dynamic binding
- Compiled to EVM opcode instructions

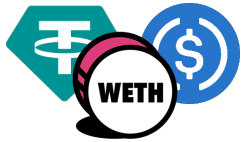
```
pragma solidity ^0.8.10;

contract HelloWorld {
  string public hello = "Hello World";
}
```

Solidity Docs: https://docs.soliditylang.org/en/v0.8.25/

Use Cases for Smart Contracts

Fungible and Non-Fungible Tokens (NFT)

- Fungible token systems are currently the largest use case for smart contracts, functioning a sub-currency of Ethereum and represent a certain asset such as a stock.
  - **Examples**: USDT, USDC,WETH
- NFTs are unique cryptographic tokens that are stored on a blockchain, representing digital assets such as drawings, music, media, virtual fashion items, etc.
  - **Examples**: Bored Ape, CryptoKitties



Decentralized Finance (DeFi)

- Public, permissionless, and interoperable finance ecosystem built on smart contract-enabled blockchains.
  - **Examples**: Decentralized Exchanges (DEX), Lending and Borrowing Platforms

Play-to-Earn Games

- Games where users earn game-native tokens which they can use to buy or trade in-game assets.
  - **Examples**: Axie Infinity, Decentraland

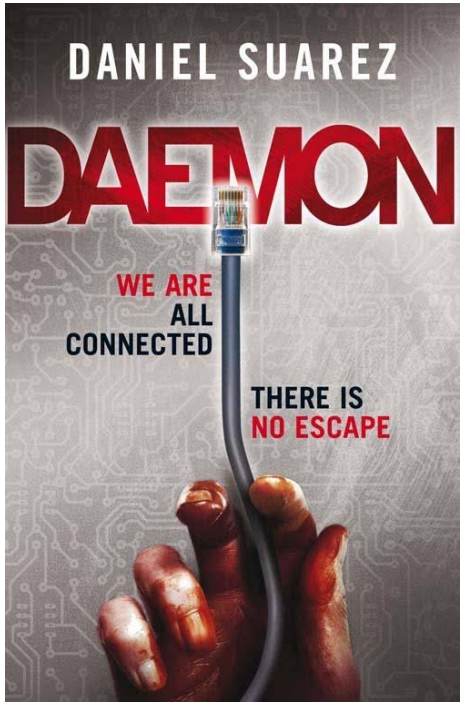
Decentralized Autonomous Organizations (DAO) and Governance

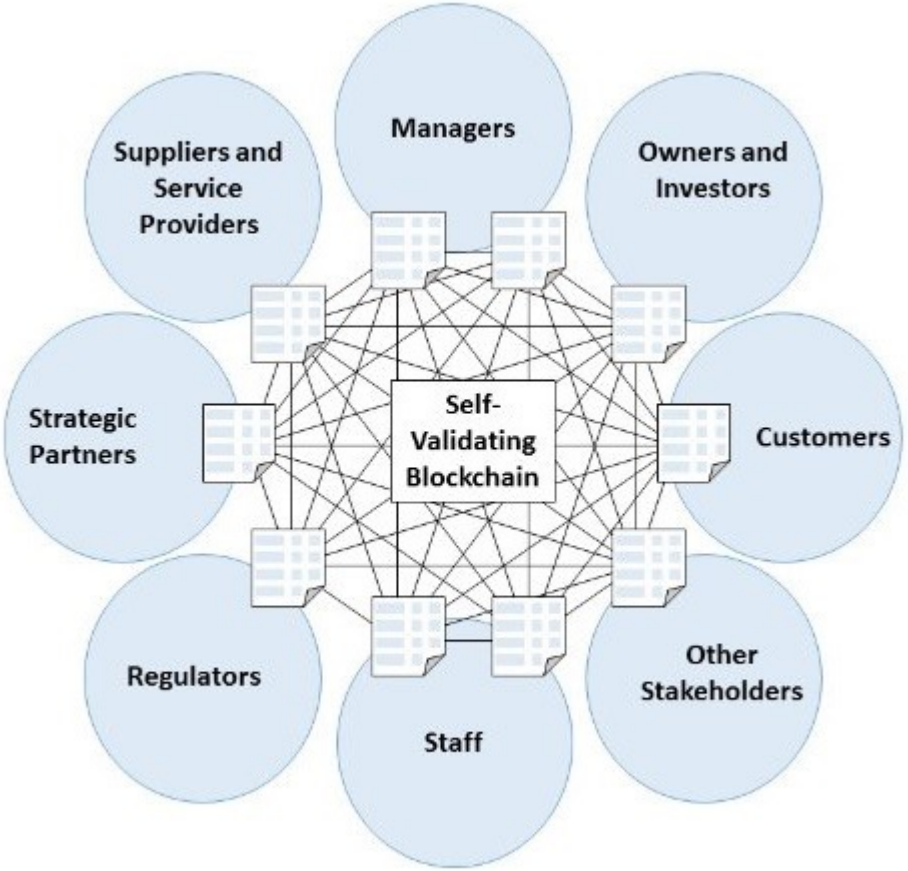
- DAOs are non-hierarchical business models that are collectively owned by a group of members.
- Think of it as a generalization of multi-signature wallets where all decisions are taken based on members' votes. Thus, no single member can unliterally take a decision (e.g., send a transaction).
- DAOs are useful for starting an organization with people you have limited trust (e.g., only communicating over the internet) since you only need to trust the system design and no other single member.
  - **Examples**: MakerDAO, Friends With Benefits (FWB)

Brief Insight into Decentralized Autonomous Organizations (DAOs)

Vision

- Create a fully digital (virtual) organisation.
- The organisation exclusively uses smart contracts to interact with its shareholders, employees, customers, suppliers, partners and public authorities.
- These stakeholders can be humans or organizations in the “real world” or other DAOs.





Outline

1. Ecosystem
   - Historical Overview
   - Crowdsale Statistics
   - Technical Papers
   - Foundations
   - Network Metrics

2. System Architecture
   - Concept of a World Computer
   - EVM
   - Accounts
   - Block Header
   - Smart Contracts

3. The Merge
   - Ethereum’s Vision
   - The Beacon Chain
   - New Architecture
   - Summary of Changes
   - What’s Next

Slides in the section “The Merge” are influenced from the slides of Burak Öz - "What does “The Merge” mean for Ethereum?”, Blockchain-Stammtisch Donnersblock, 2022.

Ethereum's Vision

Although Ethereum transitioned to Proof-of-Stake (PoS) only in September 2022, **this was the plan since the early days**.

Vitalik wanted to deploy Ethereum as a PoS chain back in 2014 but did not do it due to the **complexity of bootstrapping**.

- PoS requires a certain set of native cryptocurrencies to be minted and deployed to the participants.
- Participants have to stake their coins to become involved in block production and consensus.

Hence, in 2015, **Ethereum launched as a PoW chain**, following *Nakamoto Consensus* like Bitcoin.

Proof-of-Stake

"Proof of Stake (PoS) is a category of consensus algorithms for public blockchains that **depend on a validator's economic stake in the network**."

"In PoS-based public blockchains (e.g. Ethereum's upcoming Casper implementation), a set of **validators take turns proposing and voting on the next block**, and the weight of each validator's vote depends on the size of its deposit (i.e. stake). Significant advantages of PoS include **security**, **reduced risk of centralization**, and **energy efficiency**."

"Proof of Stake FAQ", Vitalik Buterin

"A Proof of Stake Design Philosophy" by Vitalik Buterin: https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51

The Beacon Chain

Six years after the launch of the Ethereum main network, in December 2020, the Ethereum community launched the **Beacon Chain**.

The Beacon Chain uses **PoS** as its Sybil-control mechanism and **Gasper**[1] for consensus. As there is no more mining, block proposers and consensus participants who vote on blocks are called **validators**.

- A network participant is **required to stake 32 ETH** to become a validator.
- Consensus (finding the head of the blockchain and reaching finality) is achieved through **validators' votes**.

Epochs and Slots

- The Beacon Chain divides time into **epochs** and **slots**, where each epoch contains **32 slots**, and every slot lasts **12 seconds**.
- In each epoch, every validator has a role in a particular slot, either as a **block proposer** or an **attestor** (voter).
- If everything goes as planned, a block will be added to the blockchain at every slot, and by the end of the epoch, the chain length will increase by 32 blocks.
- Through this block production mechanism, Beacon chain reaches a **deterministic block time** of 12 seconds.

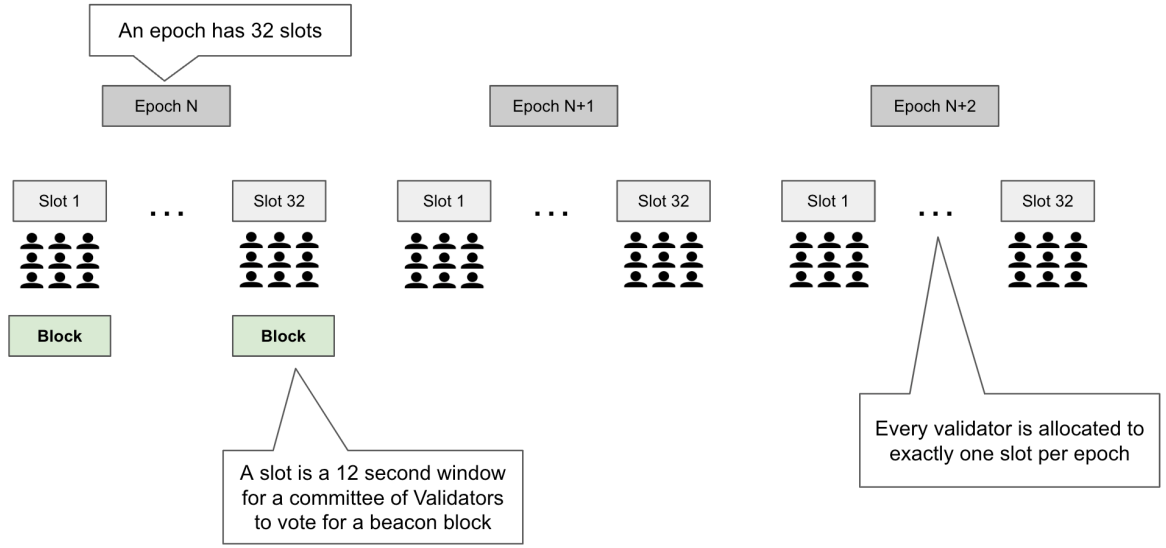


Image taken from: https://stonecoldpat.substack.com/p/epochs-slots-and-beacon-blocks

[1] *Gasper* is a mix of longest-chain and Byzantine-Fault tolerant consensus mechanisms. The details of it can be found here: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/gasper/

The Merge

On September 15th 2022, the Ethereum main network **stopped running PoW** and **merged with the Beacon Chain**, carrying over all the historical transaction and state data.

→ The merge was **triggered by reaching a particular total terminal difficulty** instead of a certain block height to prevent malicious entities from mining empty blocks to pull the merge date sooner than planned.

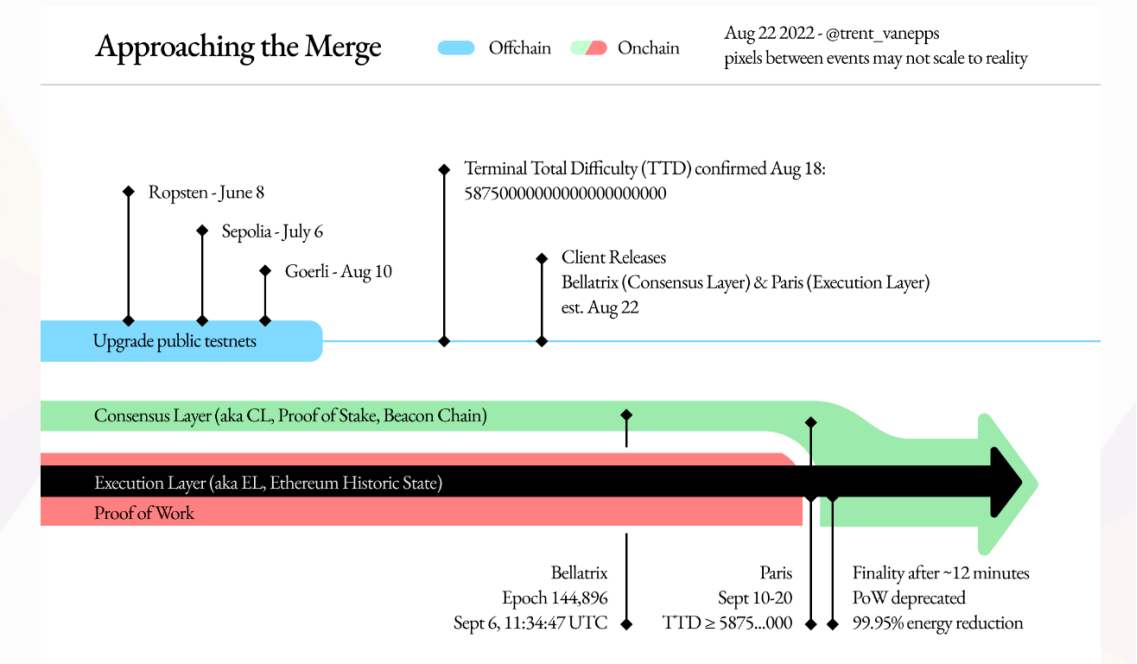
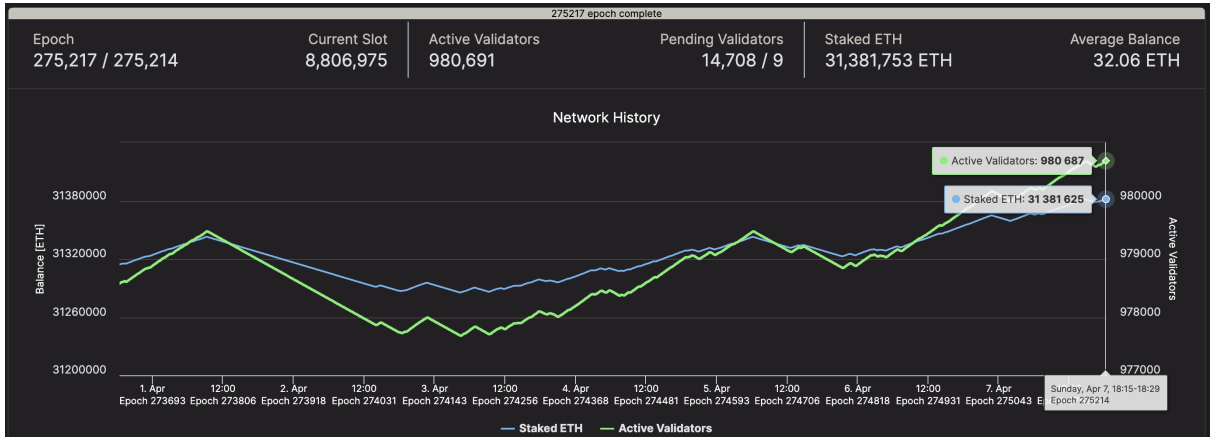


Image taken from: https://www.weforum.org/agenda/2022/09/ethereum-merge-crypto-currency-sustainability/

- As of April 2024, there are more than **980,000 active validators** on Ethereum, with a total **staked** amount exceeding **31.3M ETH**.



Screenshot taken from: https://beaconcha.in/

New Architecture

After the merge, Ethereum has a new architecture that separates the **consensus layer** (CL) and the **execution layer** (EL) into their own p2p networks. The main motivation behind this separation is to reach **better modularity**.

A user running a single PoW-Ethereum node now has to run two nodes, each with its own Ethereum client[1].

Consensus Client

- Operates on the PoS-Beacon Chain.
- Responsible for building and broadcasting blocks and **reaching a consensus** about the current world state.

Execution Client

- Operates on the old PoW-Ethereum network.
- Responsible for validating transactions, running EVM, and **preparing execution payload**[2] for consensus clients.
- Offers an interface for end-users to interact with Ethereum through RPC methods.

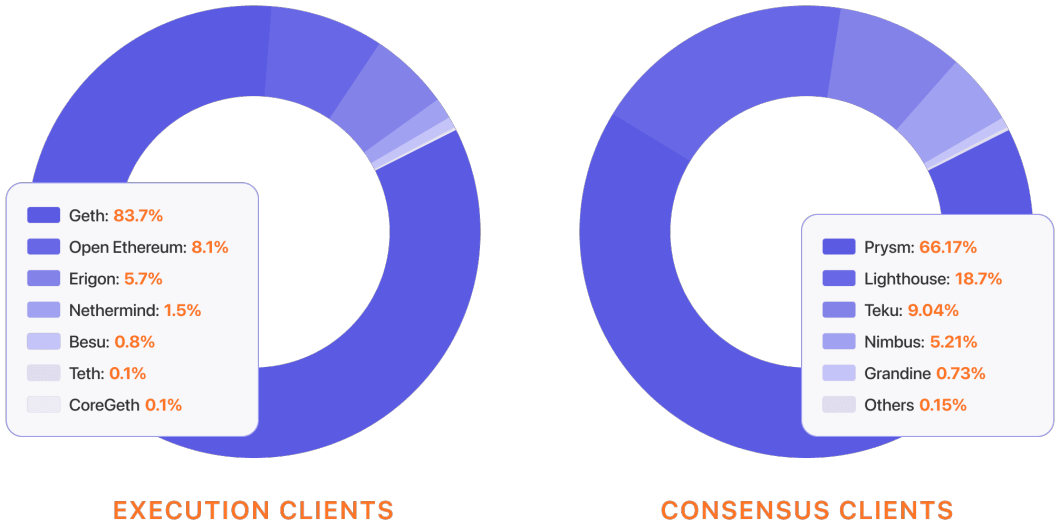


Figure taken from: https://ethereum.org/en/developers/docs/nodes-and-clients/client-diversity/

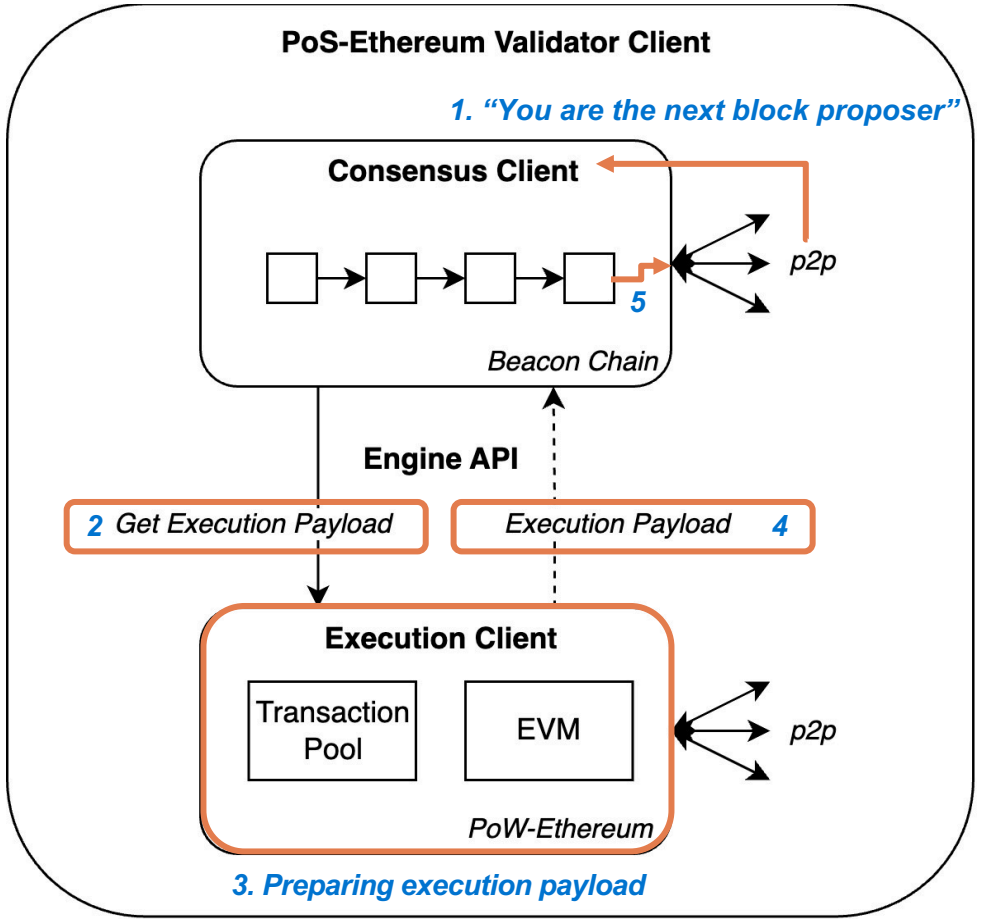
[1] Clients are software implementations of Ethereum that follow particular specifications.
[2] Execution payload refers to transactions to be included in the block, updated state trie, and other execution related data.
You can read more about Ethereum's node types and consensus and execution clients here: https://ethereum.org/en/developers/docs/nodes-and-clients/

New Architecture (cont.)

A **PoS-Ethereum validator runs** both a **consensus client** and an **execution client** in **parallel**. These clients are used during block production and attestation periods.

Block Production Workflow

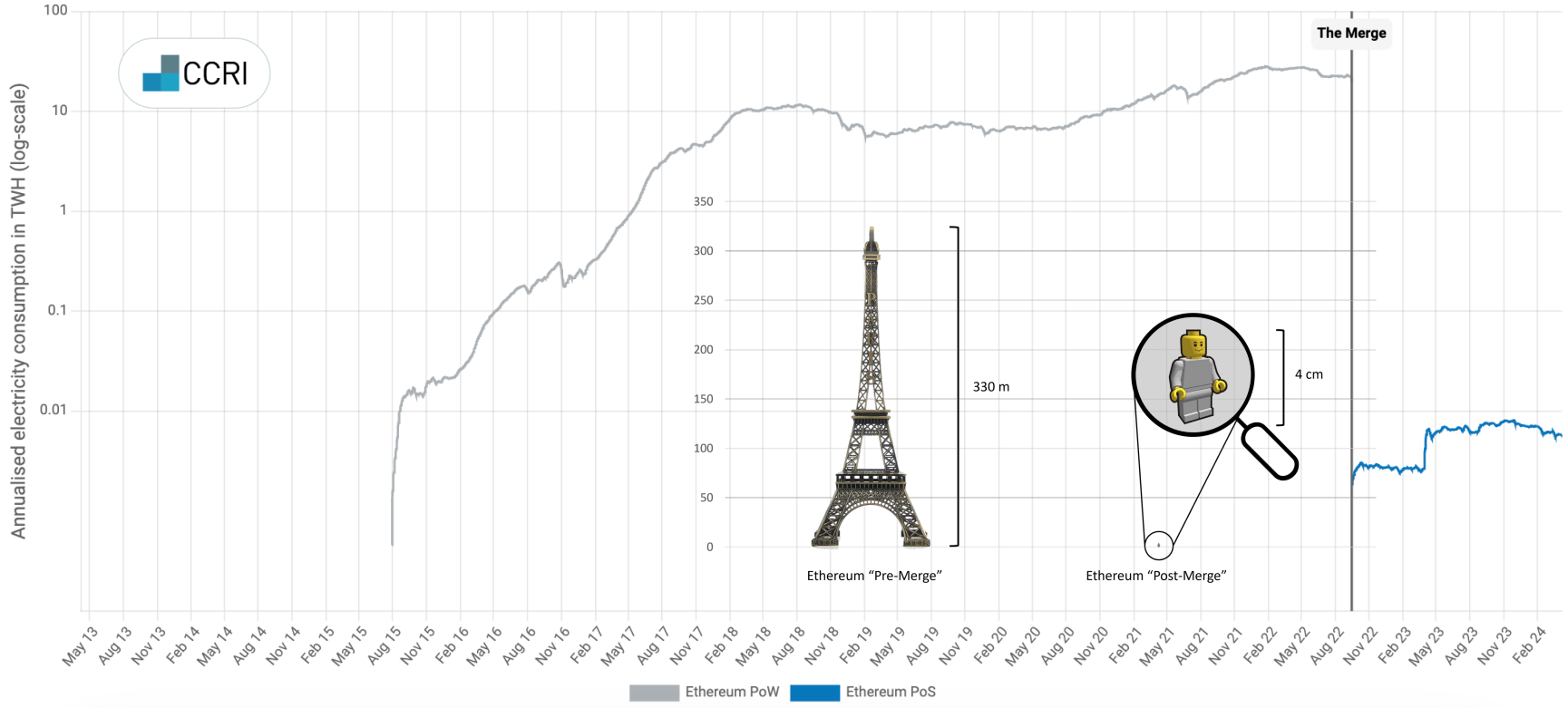
1. Consensus layer p2p network **notifies** the consensus client (CC) of the **next block proposer validator**
2. CC **requests a block** from the **execution client** (EC) through Engine API
3. EC **collects transactions** from its mempool, **executes** them, and **generates a block hash** (i.e., an execution payload)
4. EC responds to CC with the **execution payload**, and CC places the payload into a **beacon block**
5. CC **broadcasts the block** to the consensus layer p2p network
6. Other CCs in the network **retrieve** the block, **validate** it through their own execution client, and **attest** to it



Öz, B., & Hoops, F. (2023). The Advent of a New Incentive - Maximal Extractable Value. In Cronqvist, Pely (2023), *A Companion to Decentralized Finance, Digital Assets, and Blockchain Technologies*.

Summary of Changes

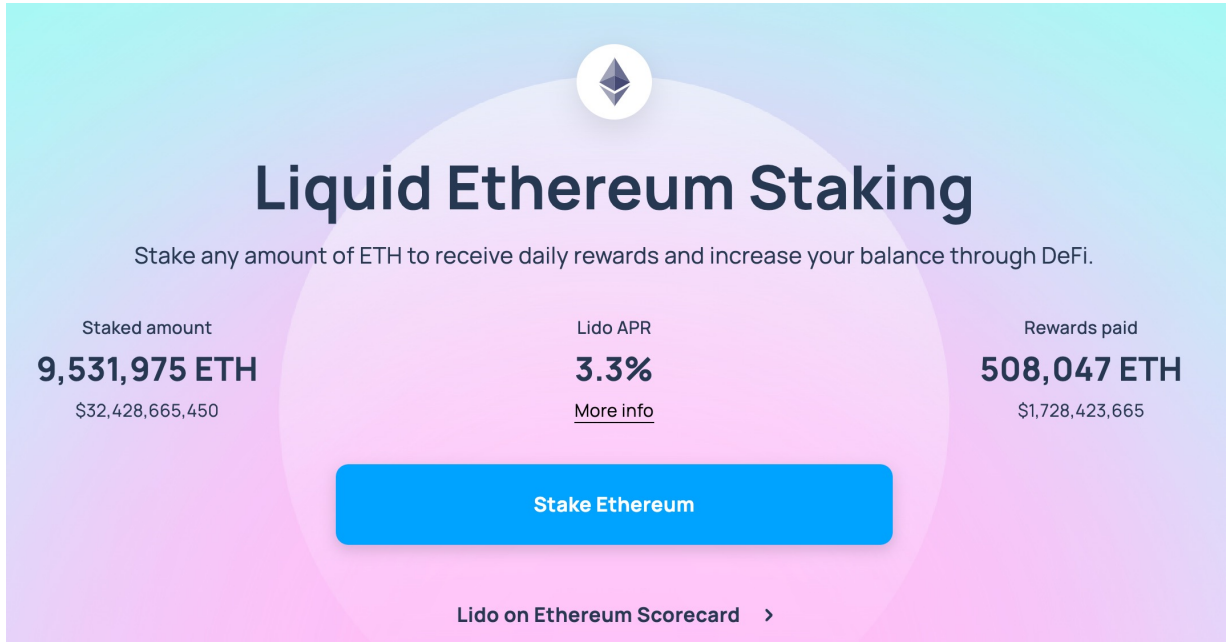
- **Lower energy consumption** as Sybil-resistance does not depend on computational power anymore (99.5% decrease).



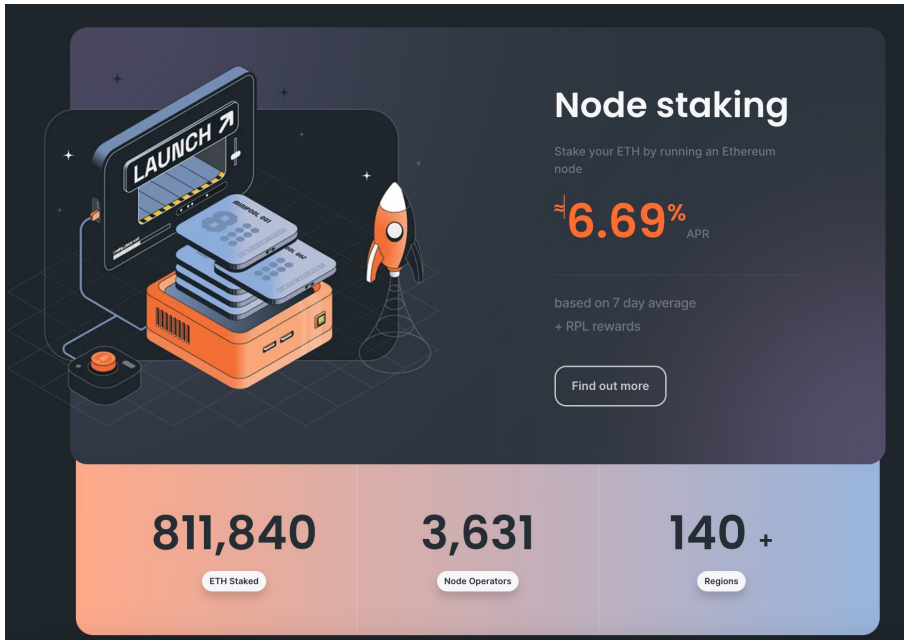
Screenshots taken from Crypto Carbon Ratings Institute: https://indices.carbon-ratings.com/ethereum-merge

Summary of Changes

- Any address staking **32 ETH** can become a **validator** and fulfil consensus duties such as proposing blocks and attesting to **earn rewards**.
  - No need to run specialized hardware as in PoW systems.
- One can also join a **staking pool** with **less than 32 ETH stake** and earn staking rewards without directly participating in consensus.
  - Joining a staking pool requires less technical knowledge than running a validator node.



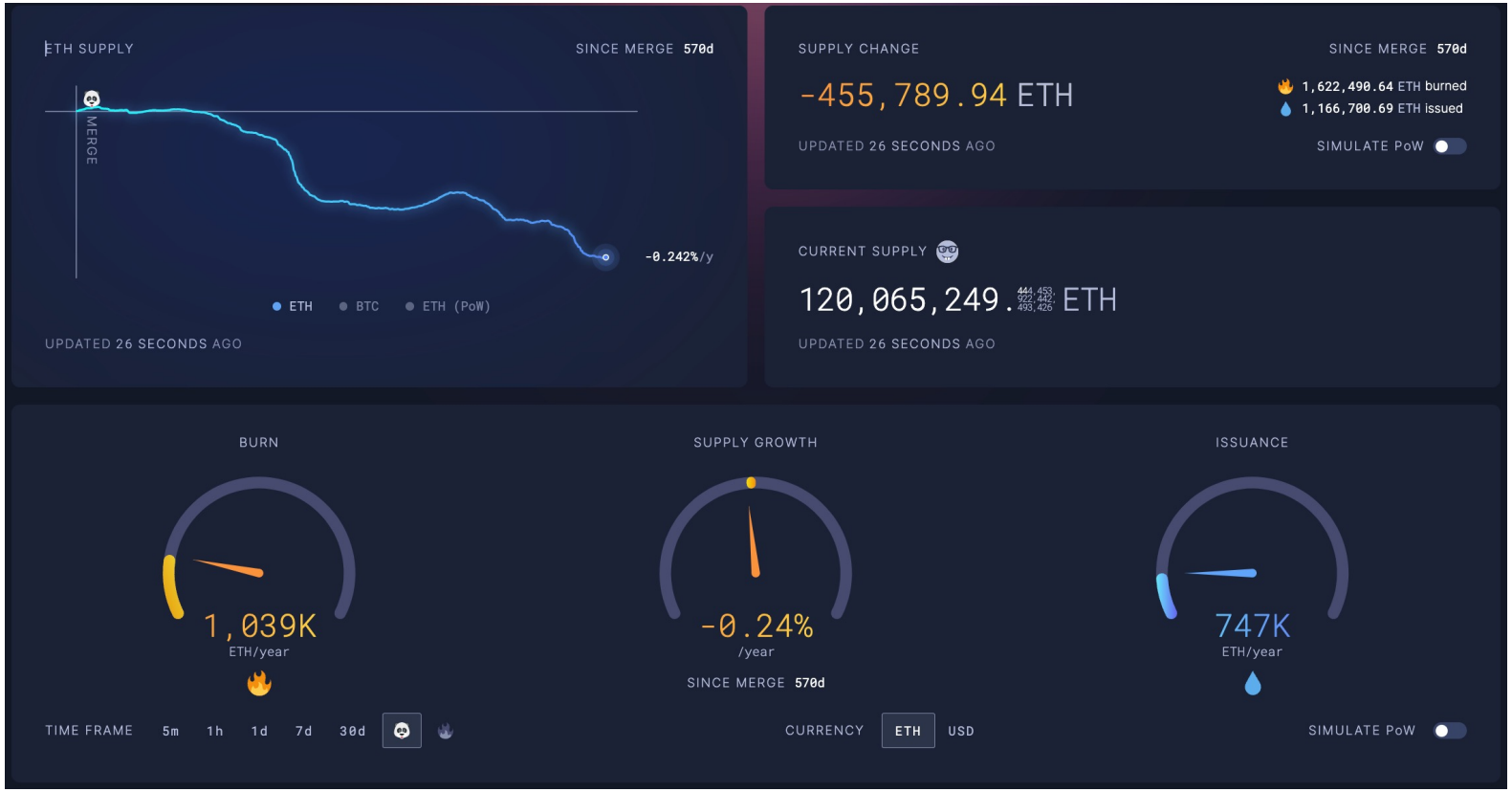
Lido



Rocket Pool

Summary of Changes

- **Block rewards are reduced** as block proposers do not require an incentive as high as they did when running PoW.
  - **PoS imposes a much lower energy requirement** on block proposers.
- Combined with EIP-1559 transactions[1], which **burn** a significant part of the transaction fee, Ethereum heads towards becoming **deflationary** (i.e., supply diminishes).

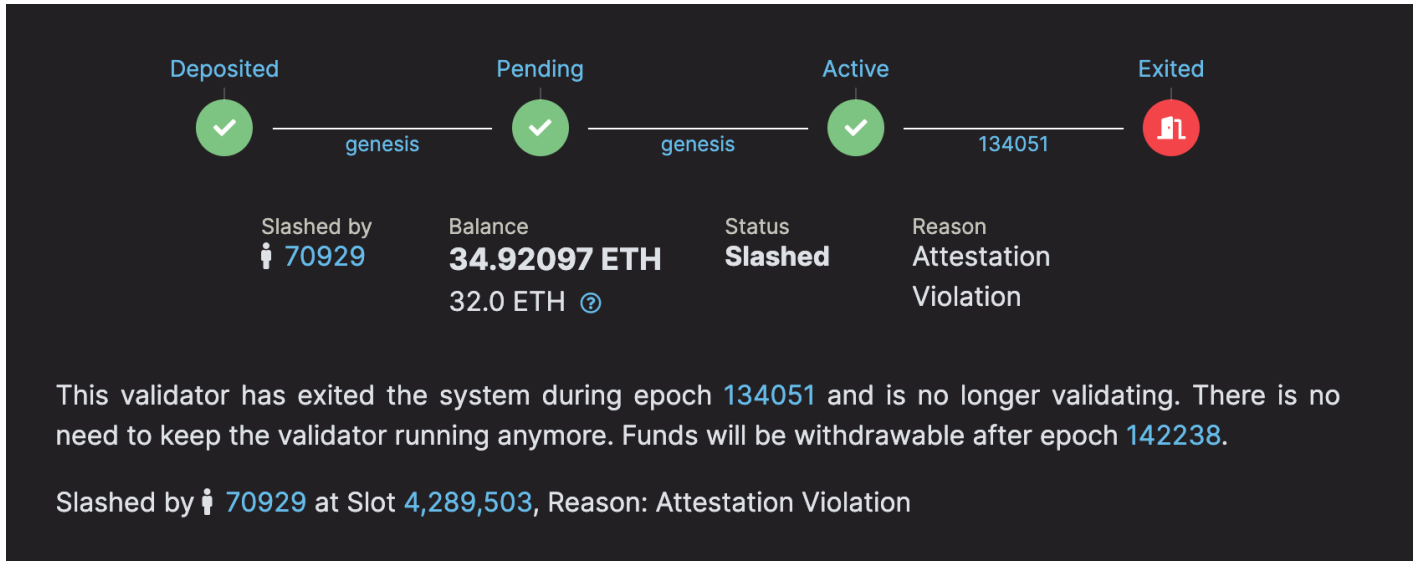


[1] EIP-1559 transactions will be covered in the central exercise session.

"Ultra sound money" is a meme referring to Ethereum becoming deflationary, making it even more solid than gold. You can read more about it here: https://ultrasound.money/

Summary of Changes

- **PoS enabled in-protocol penalties/slashing** to punish misbehaving validators.
- **Social recovery from a 51% attack** using minority-activated forks became possible.
  - Honest minority keeps building on a minority chain, ignoring the attacker's fork. Eventually, the attacker is removed from the network, and his staked ETH gets destroyed.
  - *How can you remove the hardware of a PoW-Ethereum miner from the network?*

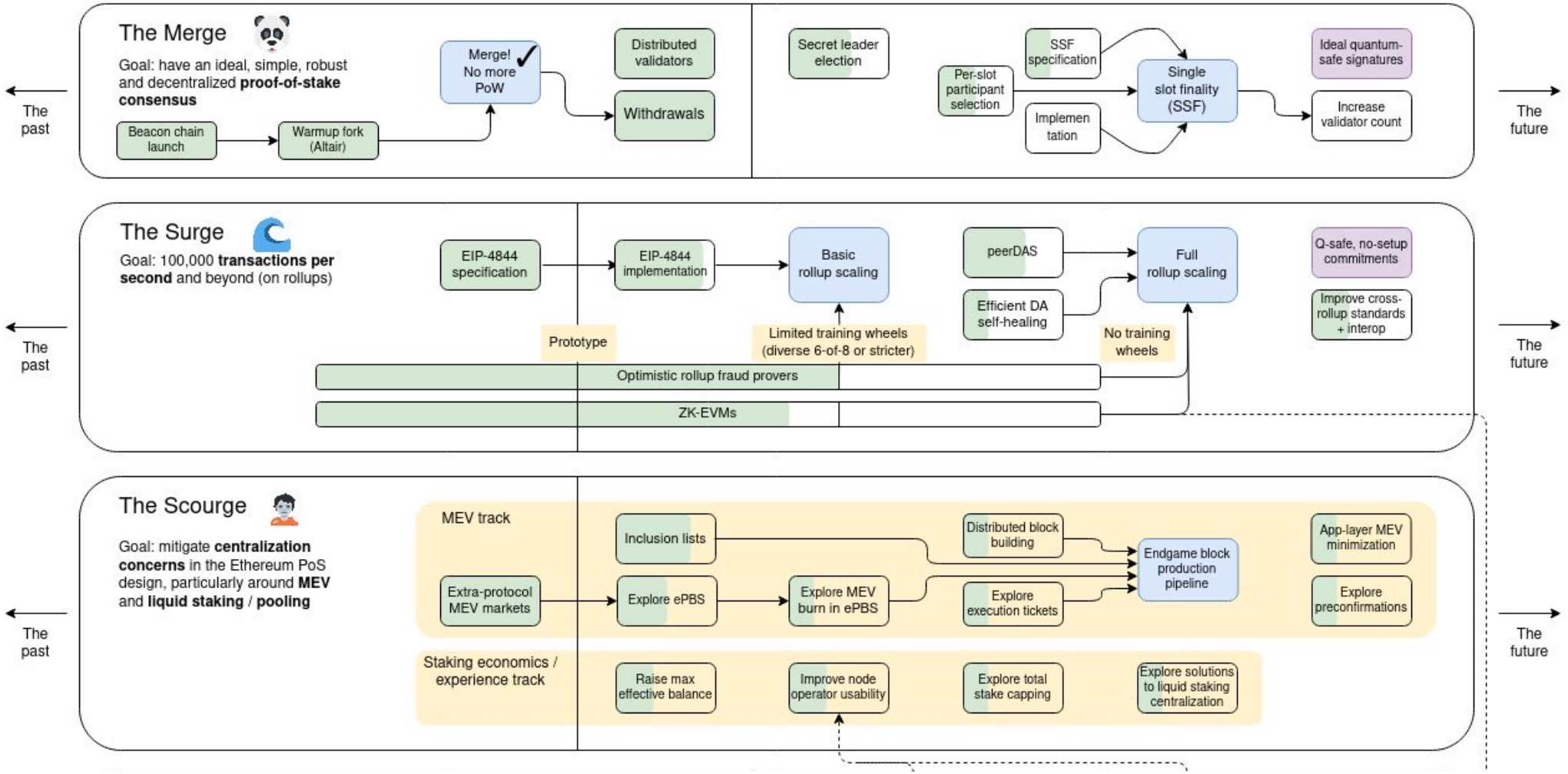


A validator which got **slashed** and **exited** the Beacon Chain.

Read more on PoS and security here: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/#pos-and-security

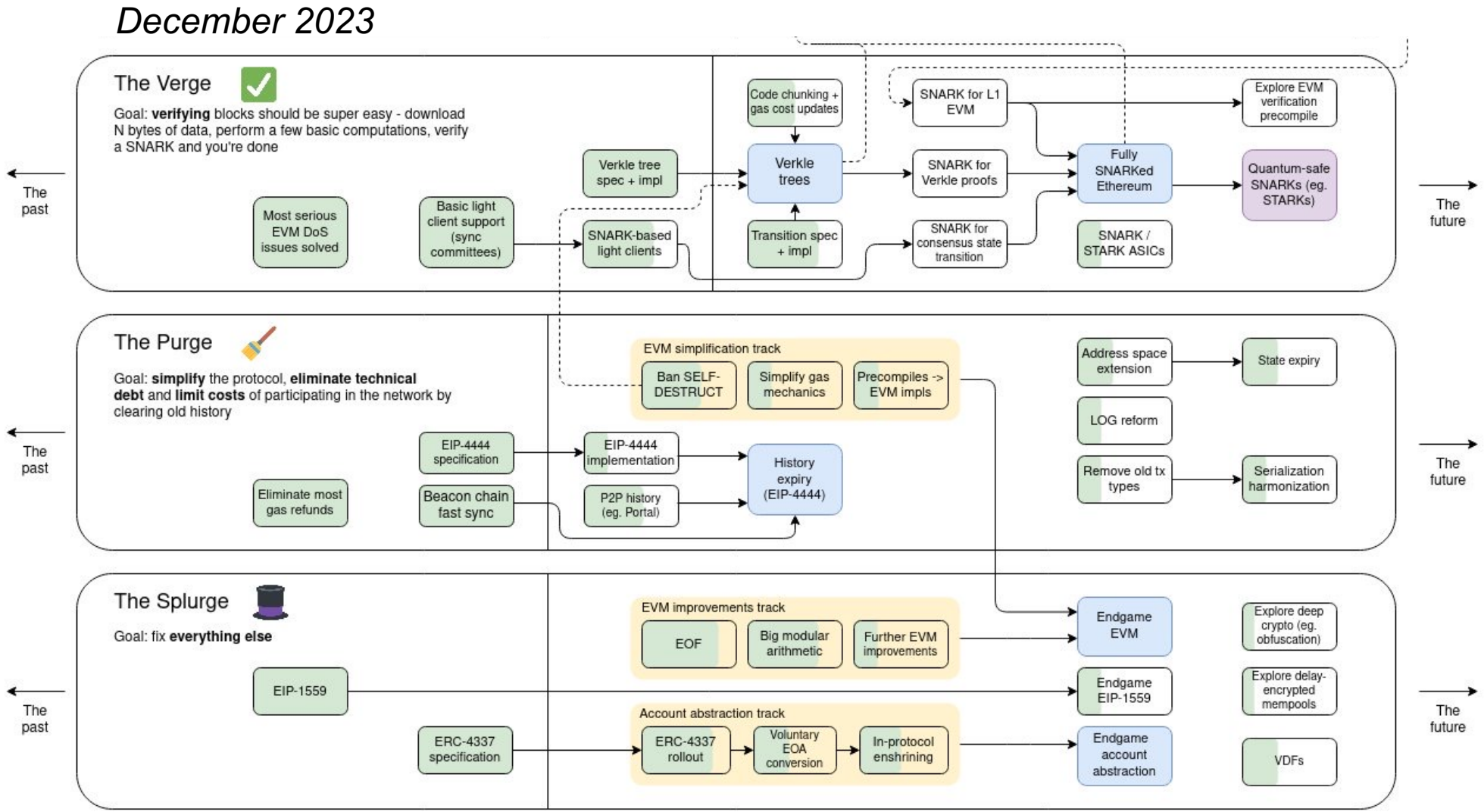
What's Next

December 2023



Original version taken from Vitalik's Twitter: https://twitter.com/VitalikButerin/status/1741190491578810445

What's Next



Original version taken from Vitalik's Twitter: https://twitter.com/VitalikButerin/status/1741190491578810445

