Ethereum

"THE WORLD COMPUTER"

From a computer science perspective

Ethereum is a **deterministic** but practically unbounded **state machine**, consisting of:

a globally accessible **singleton state** and

a **virtual machin**e that applies changes to that state.

From a more practical perspective

Ethereum is an **open source**, globally **decentralized computing infrastructure** that executes programs called smart contracts.

It uses a blockchain to synchronize and store the system's state changes, along with a cryptocurrency called ether to meter and constrain execution resource costs.

Compared to Bitcoin

Common elements:

a peer-to-peer network connecting participants,

a Byzantine fault–tolerant consensus algorithm for synchronization of state updates (a proof-of-work blockchain),

the use of cryptographic primitives such as digital signatures and hashes, and

a digital currency (ether).

Compared to Bitcoin

Differences:

purpose. Ethereum is not primarily a digital currency payment network.

construction. Ethereum is designed to be a general-purpose computer. Turing complete.

Components of a blockchain

P2P network
- Connecting nodes
- “Gossip” protocol

Transactions

Consensus rules
- Valid transactions
- Valid state transitions

State machine

Chain of blocks

- Consensus algorithm
  - Decentralizes control over Blockchain
  - Enforces consensus rules
- Incentivization scheme
  - Block rewards
- Software implementations of the above

The Birth of Ethereum

Moving beyond cryptocurrency applications.

Build on top of Bitcoin or start a new blockchain?

- transaction types, data types, sizes of data storage
- avoid off-chain layers

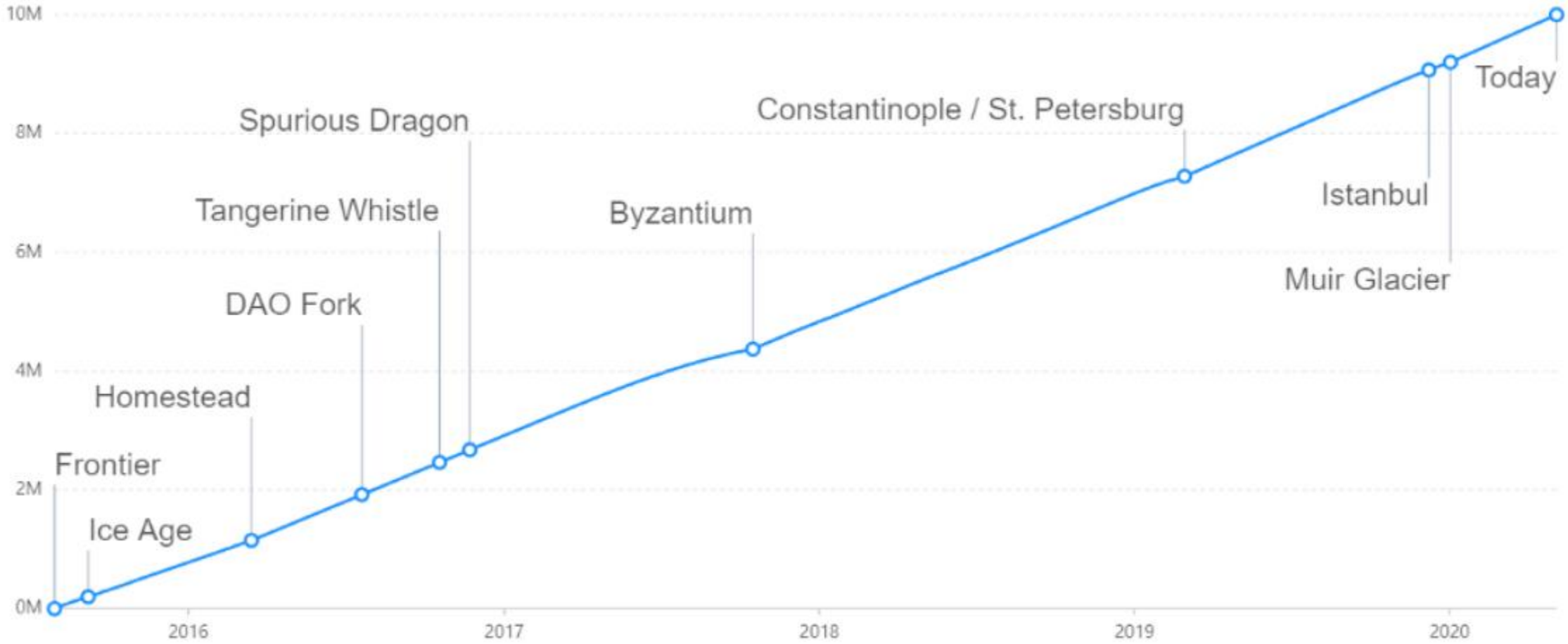
In 2013 Vitalik Buterin started writing the Ethereum whitepaper: a Turing-complete, general-purpose blockchain.

The Birth of Ethereum

Web 3

- Ethereum (contracts – decentralized logic)
- Whisper (decentralized messaging)
- Swarm (decentralized storage)

The main development stages



A General-Purpose Blockchain

Bitcoin’s blockchain tracks the state of units of bitcoin and their ownership.

Ethereum tracks the state transitions of a general-purpose data store.

Data storage model of Random Access Memory (RAM)

A General-Purpose Blockchain

Ethereum stores both code and data.

Like a general-purpose stored program computer

- It uses the blockchain to track how memory changes over time.
- It can load code into its state machine and run that code, storing the resulting state changes in its blockchain.

Unlike a general-purpose stored program computer

- Ethereum state changes are governed by the rules of consensus
- The state is distributed globally.

A General-Purpose Blockchain

“What if we could track any arbitrary state and program the state machine to create a world-wide computer operating under consensus?”

Ethereum's Components

P2P network
- Ethereum main network,
- addressable on TCP port 30303
- runs ĐΞVp2p protocol

Consensus rules
- Valid chain
- Rewards
- Etc.

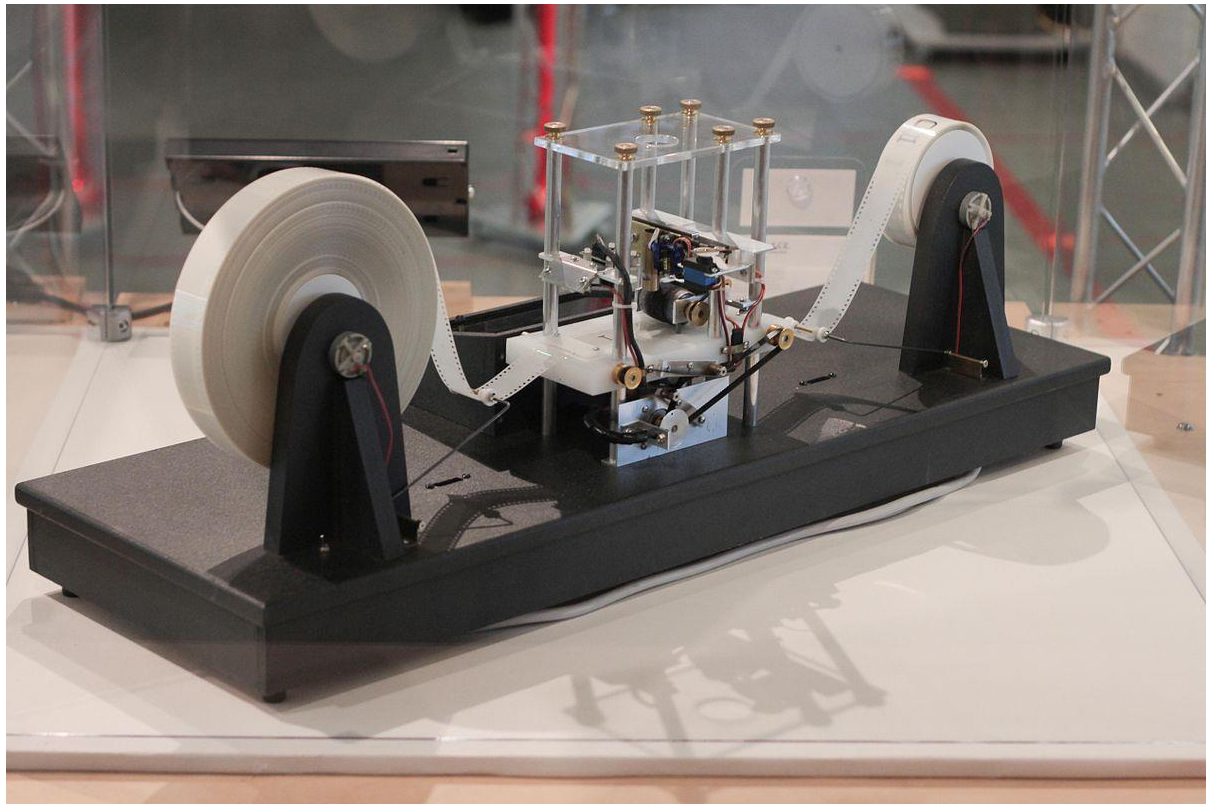
Transactions

State machine
- Ethereum Virtual Machine (EVM)
- Smart contracts

- Data structures
  - Transactions and system state
  - Merkle Patricia Tree
- Consensus algorithm
  - Currently PoW (Ethash)
  - PoS (Casper) in the future
- Economic security
  - Through consensus
- Clients
  - Geth
  - Parity

Turing Completeness

Universal computability. Are all problems solvable?



Turing Completeness

Universal computability. Are all problems solvable?

Alan Turing proved that the *halting problem* (whether it is possible, given an arbitrary program and its input, to determine whether the program will eventually stop running) is not solvable.

Ethereum programs run “everywhere,” yet produce a common state (singleton) that is secured by the rules of consensus.

The halting problem

We cannot predict whether a program will terminate without running it.

Turing-complete systems can run in “infinite loops”

In Ethereum every participating node must validate every transaction, running any smart contracts it calls.

DoS attacks!

The halting problem

How does Ethereum constrain the resources used by a smart contract if it cannot predict resource use in advance?

The answer: a metering mechanism called ***gas***.

Gas

EVM accounts for every instruction of a smart contract.

Each instruction has a predetermined cost in units of gas.

Each transaction includes an amount of gas that sets the upper limit of what can be consumed running the smart contract.

EVM will terminate execution if the amount of gas consumed by computation exceeds the gas available in the transaction.

Gas

How does one get gas to pay for computation on the Ethereum world computer?

gas price * gas limit = total cost

ETH Gas Station

Gas Price

DApps

A DApp is composed of at least:

- Smart contracts on a blockchain
- A web frontend user interface

May also include:

- A decentralized (P2P) storage protocol and platform
- A decentralized (P2P) messaging protocol and platform