

What is a blockchain? {#what-is-a-blockchain}

A blockchain is a public database that is updated and shared across many computers in a network.

"Block" refers to data and state being stored in consecutive groups known as "blocks". If you send ETH to someone else, the transaction data needs to be added to a block to be successful.

"Chain" refers to the fact that each block cryptographically references its parent. In other words, blocks get chained together. The data in a block cannot change without changing all subsequent blocks, which would require the consensus of the entire network.

Every computer in the network must agree upon each new block and the chain as a whole. These computers are known as "nodes". Nodes ensure everyone interacting with the blockchain has the same data. To accomplish this distributed agreement, blockchains need a consensus mechanism.

Ethereum uses a [proof-of-stake-based consensus mechanism](#). Anyone who wants to add new blocks to the chain must stake ETH - the native currency in Ethereum - as collateral and run validator software. These "validators" can then be randomly selected to propose blocks that other validators check and add to the blockchain. There is a system of rewards and penalties that strongly incentivize participants to be honest and available online as much as possible.

If you would like to see how blockchain data is hashed and subsequently appended to the history of block references, be sure to check out [this demo](#) by Anders Brownworth and watch the accompanying video below.

Watch Anders explain hashes in blockchains:

What is Ethereum? {#what-is-ethereum}

Ethereum is a blockchain with a computer embedded in it. It is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.

In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on. Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer. Additionally, any participant can broadcast a request for this computer to perform arbitrary computation. Whenever such a request is broadcast, other participants on the network verify, validate, and carry out ("execute") the computation. This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.

Requests for computation are called transaction requests; the record of all transactions and the EVM's present state gets stored on the blockchain, which in turn is stored and agreed upon by all nodes.

Cryptographic mechanisms ensure that once transactions are verified as valid and added to the blockchain, they can't be tampered with later. The same mechanisms also ensure that all transactions are signed and executed with appropriate "permissions" (no one should be able to send digital assets from Alice's account, except for Alice herself).

What is ether? {#what-is-ether}

Ether (ETH) is the native cryptocurrency of Ethereum. The purpose of ETH is to allow for a market for computation. Such a market provides an economic incentive for participants to verify and execute transaction requests and provide computational resources to the network.

Any participant who broadcasts a transaction request must also offer some amount of ETH to the network as a bounty. The network will award this bounty to whoever eventually does the work of verifying the transaction, executing it, committing it to the blockchain, and broadcasting it to the network.

The amount of ETH paid corresponds to the resources required to do the computation. These bounties also prevent malicious participants from intentionally clogging the network by requesting the execution of infinite computation or other resource-intensive scripts, as these participants must pay for computation resources.

ETH is also used to provide crypto-economic security to the network in three main ways: 1) it is used as a means to reward validators who propose blocks or call out dishonest behavior by other validators; 2) It is staked by validators, acting as collateral against dishonest behavior—if validators attempt to misbehave their ETH can be destroyed; 3) it is used to weigh 'votes' for newly proposed blocks, feeding into the fork-choice part of the consensus mechanism.

What are smart contracts? {#what-are-smart-contracts}

In practice, participants don't write new code every time they want to request a computation on the EVM. Rather, application developers upload programs (reusable snippets of code) into EVM state, and users make requests to execute these code snippets with varying parameters. We call the programs uploaded to and executed by the network smart contracts.

At a very basic level, you can think of a smart contract like a sort of vending machine: a script that, when called with certain parameters, performs some actions or computation if certain conditions are satisfied. For example, a simple vendor smart contract could create and assign ownership of a digital asset if the caller sends ETH to a specific recipient.

Any developer can create a smart contract and make it public to the network, using the blockchain as its data layer, for a fee paid to the network. Any user can then call the smart contract to execute its code, again for a fee paid to the network.

Thus, with smart contracts, developers can build and deploy arbitrarily complex user-facing apps and services such as: marketplaces, financial instruments, games, etc.

Terminology {#terminology}

Blockchain {#blockchain}

The sequence of all blocks that have been committed to the Ethereum network in the history of the network. So named because each block contains a reference to the previous block, which helps us maintain an ordering over all blocks (and thus over the precise history).

ETH {#eth}

Ether (ETH) is the native cryptocurrency of Ethereum. Users pay ETH to other users to have their code execution requests fulfilled.

[More on ETH](#)

EVM {#evm}

The Ethereum Virtual Machine is the global virtual computer whose state every participant on the Ethereum network stores and agrees on. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.

[More on the EVM](#)

Nodes {#nodes}

The real-life machines which are storing the EVM state. Nodes communicate with each other to propagate information about the EVM state and new state changes. Any user can also request the execution of code by broadcasting a code execution request from a node. The Ethereum network itself is the aggregate of all Ethereum nodes and their communications.

[More on nodes](#)

Accounts {#accounts}

Where ETH is stored. Users can initialize accounts, deposit ETH into the accounts, and transfer ETH from their accounts to other users. Accounts and account balances are stored in a big table in the EVM; they are a part of the overall EVM state.

[More on accounts](#)

Transactions {#transactions}

A "transaction request" is the formal term for a request for code execution on the EVM, and a "transaction" is a fulfilled transaction request and the associated change in the EVM state. Any user can broadcast a transaction request to the network from a node. For the transaction request to affect the agreed-upon EVM state, it must be validated, executed, and "committed to the network" by another node. Execution of any code causes a state change in the EVM; upon commitment, this state change is broadcast to all nodes in the network. Some examples of transactions:

- Send X ETH from my account to Alice's account.
- Publish some smart contract code into EVM state.
- Execute the code of the smart contract at address X in the EVM, with arguments Y.

[More on transactions](#)

Blocks {#blocks}

The volume of transactions is very high, so transactions are "committed" in batches, or blocks. Blocks generally contain dozens to hundreds of transactions.

[More on blocks](#)

Smart contracts {#smart-contracts}

A reusable snippet of code (a program) which a developer publishes into EVM state. Anyone can request that the smart contract code be executed by making a transaction request. Because developers can write arbitrary executable applications into the EVM (games, marketplaces, financial instruments, etc.) by publishing smart contracts, these are often also called [dapps, or Decentralized Apps](#).

[More on smart contracts](#)

Further reading {#further-reading}

- [Ethereum Whitepaper](#)
- [How does Ethereum work, anyway?](#) - Preethi Kasireddy (**NB** this resource is still valuable but be aware that it predates [The Merge](#) and therefore still refers to Ethereum's proof-of-work mechanism - Ethereum is actually now secured using [proof-of-stake](#))

Know of a community resource that helped you? Edit this page and add it!

Related tutorials {#related-tutorials}

- [A developer's guide to Ethereum, part 1](#) – A very beginner-friendly exploration of Ethereum using Python and web3.py