Foundations of Blockchain

Matteo Nardelli

March 23, 2026

Main References

- I. Bashir, “Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more” Packt Publishing Ltd, 2020.
- A. M. Antonopoulos, “Mastering Bitcoin: Programming the open blockchain”. O’Reilly Media, 2017.
- A. M., Antonopoulos and G. Wood, “Mastering Ethereum: building smart contracts and dapps”. O’Reilly Media, 2018.
- N. Kannengießer et al., “Trade-offs between distributed ledger technology characteristics”. ACM Comput Surv 53.2 (2020).
- J. Xu et al., “A Survey of Blockchain Consensus Protocols.” ACM Comput Surv 55.13s (2023).
- A. M. Antonopoulos et al., “Mastering the Lightning Network”. O’Reilly Media, 2021.

Decentralized Ledgers

Database

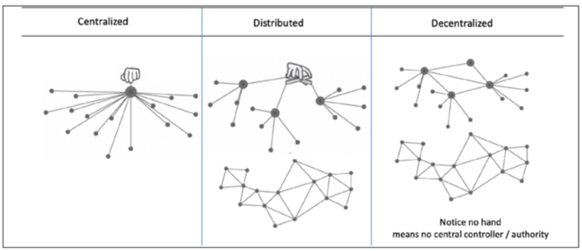
A database is an organized collection of data (also known as a **data store**) stored and accessed electronically through the use of a database management system.

Database

A database is an organized collection of data (also known as a **data store**) stored and accessed electronically through the use of a database management system.

Different possible architectures:

- **Centralized**: a single node stores and processes data;
- **Distributed**: multiple nodes, but (logically) centralized control;
- **Decentralized**: multiple nodes, decentralized control.



Distributed Ledger Technology (DLT)

A Distributed Ledger Technology (DLT) is the consensus of **replicated**, **shared**, and **synchronized** digital data that is geographically distributed across many sites.[1]

- Each node stores a **local replica** of the ledger;
- Nodes share a **protocol** that allows to guarantee safety, integrity, and consistency of data, without the need of a trusted third party;
- The shared ledger is updated through a **consensus algorithm**;
- Nodes may not trust each other (but they trust the protocol).

Unlike centralized databases, DLTs do not require a central administrator (no single point of failure).

[1] https://en.wikipedia.org/wiki/Distributed_ledger

Distributed or Decentralized?

Although the term **distributed** is used in the acronym, DLT usually refers to **decentralized** solutions, with no central authority or trusted intermediary.

Advantages

- Decentralization (no single-point-of-failure);
- High availability;
- Censorship resistance;
- Scalability (# of participants).
- High Transparency (public DLTs);

Disdvantages

- Complexity;
- Costs of transactions (and fees);
- Slow transaction speed;
- Lack of regulation by central authority;
- Energy consumption;
- Privacy concerns (public DLTs);
- Lack of interoperability.

Blockchain

Blockchain

Blockchain:

- A special type of DLT;
- Stores transactions in **blocks**, organized in a chain (i.e., *block-chain*);
- Each block is cryptographically linked to the previous one:
  - A logical order between blocks exists;
  - Changing a block invalidates all subsequent blocks;
  - This guarantees the immutability property of blockchains.
- Nodes read and update the shared ledger in a decentralized manner;

Blockchain

Blockchain:

- A special type of DLT;
- Stores transactions in **blocks**, organized in a chain (i.e., *block-chain*);
- Each block is cryptographically linked to the previous one:
  - A logical order between blocks exists;
  - Changing a block invalidates all subsequent blocks;
  - This guarantees the immutability property of blockchains.
- Nodes read and update the shared ledger in a decentralized manner;
- Two main approaches:
  - Blocks maintained in one **single chain** (guarantees total ordering of transactions);
  - Blocks maintained in a **directed acyclic graph** (to improve performance).

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.
2. **Validation and diffusion**: The transaction is checked for validity and propagated by using data-dissemination protocols (e.g., Gossiping protocol);

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.
2. **Validation and diffusion**: The transaction is checked for validity and propagated by using data-dissemination protocols (e.g., Gossiping protocol);
3. **Find new block**: *Miners* receive and validate the transaction, before adding it to the next block.

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.
2. **Validation and diffusion**: The transaction is checked for validity and propagated by using data-dissemination protocols (e.g., Gossiping protocol);
3. **Find new block**: *Miners* receive and validate the transaction, before adding it to the next block.
    - A *miner* (or *validator*) is a node participating to consensus (miner is used when consensus requires to mine, e.g., with proof-of-work);

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.
2. **Validation and diffusion**: The transaction is checked for validity and propagated by using data-dissemination protocols (e.g., Gossiping protocol);
3. **Find new block**: *Miners* receive and validate the transaction, before adding it to the next block.
    - A *miner* (or *validator*) is a node participating to consensus (miner is used when consensus requires to mine, e.g., with proof-of-work);
    - The transaction is temporary stored in the so-called *transaction pool*;

How a Blockchain Works

1. **Transaction initialization**: Alice creates a transaction to send money to Bob; Alice digitally signs the transaction (using her wallet); Alice submits the transaction to the blockchain network.
2. **Validation and diffusion**: The transaction is checked for validity and propagated by using data-dissemination protocols (e.g., Gossiping protocol);
3. **Find new block**: *Miners* receive and validate the transaction, before adding it to the next block.
    - A *miner* (or *validator*) is a node participating to consensus (miner is used when consensus requires to mine, e.g., with proof-of-work);
    - The transaction is temporary stored in the so-called *transaction pool*;
    - The consensus protocol starts.

How a Blockchain Works

4. **New block**: when consensus is reached, the next block is considered finalized.

How a Blockchain Works

4. **New block**: when consensus is reached, the next block is considered finalized.
    - The transaction is considered confirmed;

How a Blockchain Works

4. **New block**: when consensus is reached, the next block is considered finalized.
    - The transaction is considered confirmed;
    - In Bitcoin, the consensus algorithm is proof-of-work; the miner is rewarded with a certain amount of new coin as incentive (plus the fees of all transaction included in the block).

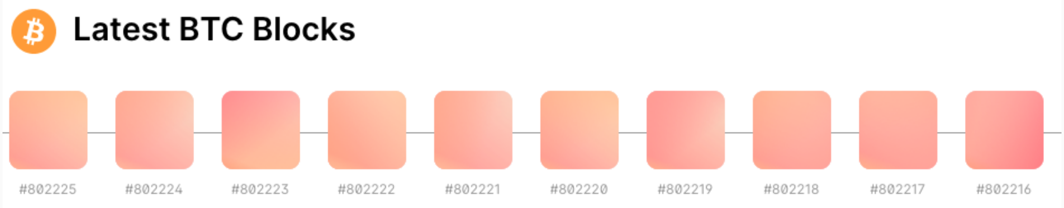
How a Blockchain Works

4. **New block**: when consensus is reached, the next block is considered finalized.
    - The transaction is considered confirmed;
    - In Bitcoin, the consensus algorithm is proof-of-work; the miner is rewarded with a certain amount of new coin as incentive (plus the fees of all transaction included in the block).
5. **Add new block to the blockchain**: The block is propagated to other nodes, who execute the transactions in it and further propagate the block.

Blockchain: Notable Examples

- Bitcoin
  - 2009 by Satoshi Nakamoto (pseudonym).
  - Designed to exchange the *bitcoin* crypto-currency.
  - Introduced the Proof-of-Work consensus algorithm.
  - Approximately, 1 new block every 10 minutes.
- Ethereum
  - 2015 by Vitalik Buterin and Gavin Wood.
  - Designed to exchange the *ether* crypto-currenty and run smart-contract.
  - Enabled Decentralized Finance and NFTs.
  - Approximately, 1 block every 12 s.
- Algorand
  - 2019 by Silvio Micali.
  - 1 block every 3.9 s.

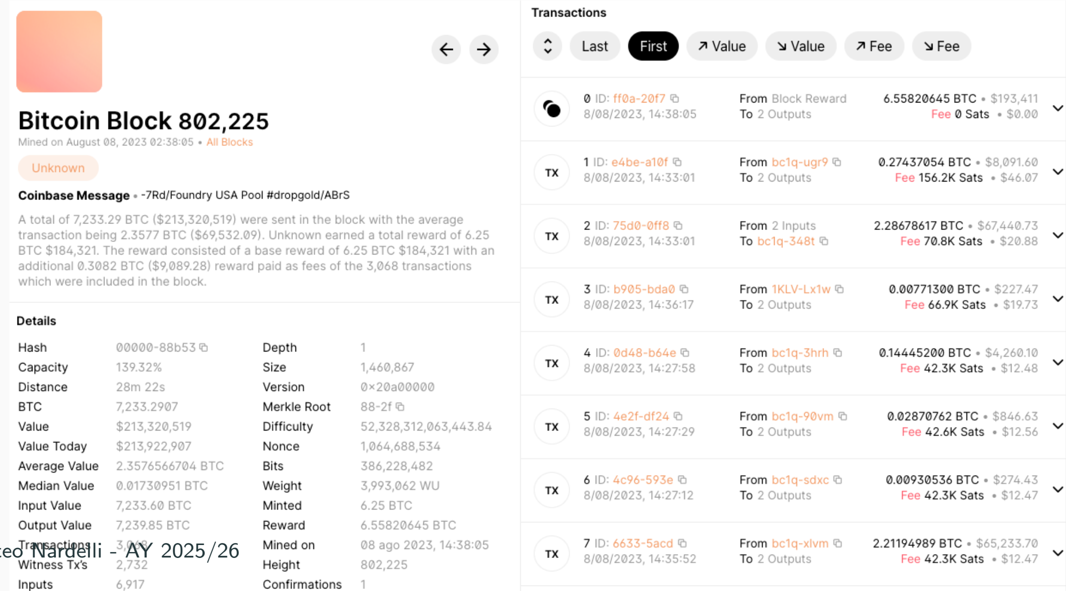
Blockchain: An Example from BitCoin (BTC)



- Block #802225 includes a *hash pointer* to block #802224, which hash points to #802223, and so on.
- A hash pointer[2] is a tuple that contains a traditional pointer along with the hash of the data element that is being pointed to. It allows us to validate that the information being pointed to has not been modified.

[2] https://people.cs.rutgers.edu/~pxk/419/notes/bitcoin.html

Blockchain: An Example from BitCoin



Components of a Blockchain

- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);

- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);
- Messages representing state transitions;
- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);
- Messages representing state transitions;
- A **consensus protocol** that decentralizes control to determine valid state transitions;
- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);
- Messages representing state transitions;
- A **consensus protocol** that decentralizes control to determine valid state transitions;
- A **state machine** that processes transactions w.r.t. the consensus rules;
- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);
- Messages representing state transitions;
- A **consensus protocol** that decentralizes control to determine valid state transitions;
- A **state machine** that processes transactions w.r.t. the consensus rules;
- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);
- An **incentivization** scheme (e.g., proof-of-work costs plus block rewards);

Components of a Blockchain

- A peer-to-peer (P2P) **network** connecting participants and propagating transactions and blocks of verified transactions (through gossiping protocol);
- Messages representing state transitions;
- A **consensus protocol** that decentralizes control to determine valid state transitions;
- A **state machine** that processes transactions w.r.t. the consensus rules;
- A chain of cryptographically secured blocks (acting as a journal of all the accepted state transitions);
- An **incentivization** scheme (e.g., proof-of-work costs plus block rewards);
- One or more open source software implementations.

Blockchain: Nodes, Transactions, and Blocks

Node:

- Participant of the peer-to-peer network implementing the blockchain;
- Each node has the same role;
- Each node stores a copy of the (possibly entire) ledger;
- Can propose and validate transactions;
- Can participate to the consensus protocol.

Blockchain: Nodes, Transactions, and Blocks

Node:

- Participant of the peer-to-peer network implementing the blockchain;
- Each node has the same role;
- Each node stores a copy of the (possibly entire) ledger;
- Can propose and validate transactions;
- Can participate to the consensus protocol.

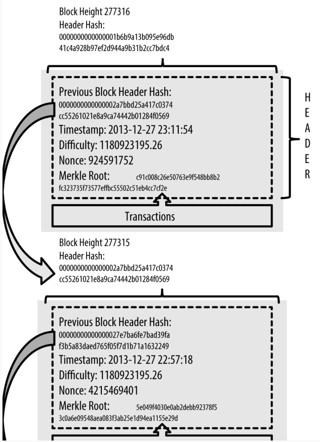
Transaction:

- Represents any change of the ledger (or data store) state;
- Includes input and output data (e.g., money to spent), a timestamp, and a digital signature;
- Bitcoin, Ethereum, Algorand, etc, have their own specific definition.

Blockchain: Nodes, Transactions, and Blocks

Block:

- Includes a list of valid transactions (body) and an header;
  - The header includes: block size, previous block hash (hash pointer), timestamp, difficult, nonce, and Merkle Root.
- Its structure depends on the specific blockchain;
- To add a new block to the chain, nodes run a consensus algorithm:
  - All information needed to validate a transaction must be available on the chain.



Merkle Tree

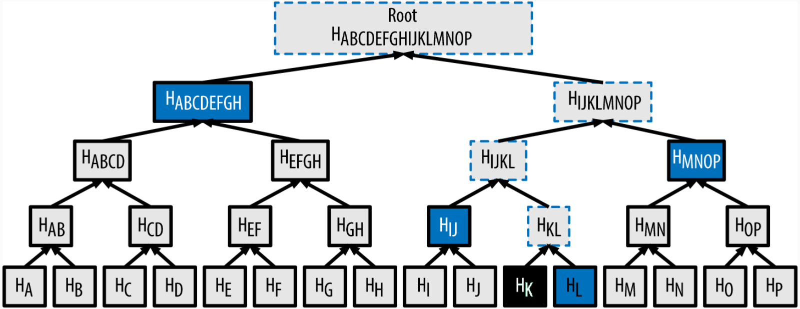
- A Merkle tree is used as a summary of all the transactions in the block;
- Merkle trees are binary trees containing cryptographic hashes;
- Constructed by recursively hashing pairs of nodes until there is only one hash, the Merkle root;
  - If there is an odd number of transactions, the last transaction hash will be duplicated.
- As hash function, Bitcoin uses SHA256 applied twice;
  - SHA256 is applied twice to prevent the length extension attack.
- Example:
  - $H_A = \text{SHA256}(\text{SHA256}(\text{Transaction A}))$
  - $H_{AB} = \text{SHA256}(\text{SHA256}(H_A + H_B))$
  - $H_{ABCD} = \text{SHA256}(\text{SHA256}(H_{AB} + H_{CD}))$
  - ...

Length Extension Attack

- A type of attack where:
  - an attacker can use Hash(message1) and the length of message1
  - to calculate Hash(message1 —— message2) for an attacker-controlled message2;
- An attacker can include extra information at the end of the message and produce a **valid hash**;
- Algorithms based on Merkle–Damgård construction (like MD5, SHA-1, SHA-2) are susceptible to this attack;
  - SHA-256 and SHA-512 are in the familiy of SHA-2;
  - Truncated versions of SHA-256 and SHA-512 are resistant to this attack;
  - SHA-3 is resistant as well.

Merkle Tree

How to quickly verify all transactions in a block? A Merkle tree is a binary tree used to efficiently summarize and verify the integrity of a large number of transactions.



A node can prove that transaction K is included in the block by producing a Merkle path, which consists of 4 hashes: H_L, H_{IJ}, H_{MNOP} e $H_{ABCDEFGH}$.

Only $\log_2(N)$ 32-byte hashes needed.

Types of Blockchains

- **Visibility**: public or private, based on *read* permissions;
- **Permission**: permissionless or permissioned, whether all nodes or a subset of them is authorized to participato to the consensus protocol (hence, updating the blockchain state—*write* permissions).

	Public	**Private**
Permissionless	Bitcoin, Ethereum	Ark Ecosystem
Permissioned	GoChain	Hyperledger Fabric, Quorum, R3 Corda

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;
- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back;

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;
- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back;
- **Transparency**: As blockchains are shared, every node can see what the information on the blockchain;

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;
- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back;
- **Transparency**: As blockchains are shared, every node can see what the information on the blockchain;
- **Security**: Only valid transactions are selected for inclusion; all transactions on a blockchain are cryptographically secured;

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;
- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back;
- **Transparency**: As blockchains are shared, every node can see what the information on the blockchain;
- **Security**: Only valid transactions are selected for inclusion; all transactions on a blockchain are cryptographically secured;
- **Consensus**: A block is added only after solving the consensus problem among network participants;

Key Principles

- **Decentralization**: Each peer validates the transaction independently and (potentially) participates in the consensus algorithm;
- **Immutability**: Once the data has been written to the blockchain, it is extremely difficult to change it back;
- **Transparency**: As blockchains are shared, every node can see what the information on the blockchain;
- **Security**: Only valid transactions are selected for inclusion; all transactions on a blockchain are cryptographically secured;
- **Consensus**: A block is added only after solving the consensus problem among network participants;
- **Programmability**: Most blockchains offer programmability features, thus actions can be triggered when specific conditions occur.

Confidentiality vs Privacy

Confidentiality

Assures that private or confidential information is not made available or disclosed to unauthorized individuals. Data-centric property.

Privacy

Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed. User-centric property.

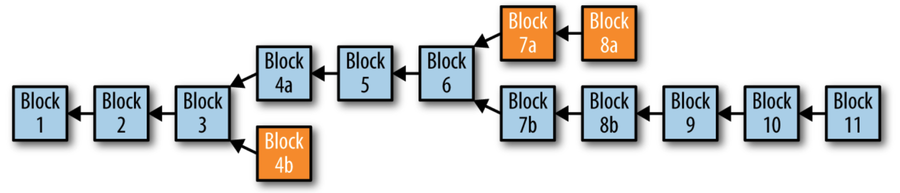
Consensus Algorithm

- Nodes agree on the next block to add to the chain;
- Choosing the consensus algorithm depends on the type of blockchain;
- Some examples:
  - **Proof-of-Work**: is based on the creation of a "proof" computationally hard to obtain but easy to verify; the proof testifies the work done to the entire blockchain network.
  - **Proof-of-Stake**: is based on the idea of "stake" (e.g., money owned). The probability to be selected for proposing the next block increases with the amount of stake committed. Several variants exist.
  - **Proof-of-Authority**: validator nodes are known; they have the authority to propose a new block. Usually used in permissioned networks.

Xu et al. "A Survey of Blockchain Consensus Protocols", ACM Comput. Surv. 55, 13s, Art. 278. 2023.

Fork

- The blockchain is a decentralized data structure, different copies of it are not always consistent.
- A fork is what happens when a blockchain diverges into two potential paths forward;



Fork

- **Intentional fork**: due to changes of the blockchain rules.
  - **Hard fork**: not backward-compatible change; all users are required to upgrade their software;
    - E.g., Ethereum/Ethereum Classic, Bitcoin/Bitcoin Cash.
  - **Soft fork**: backward-compatible change; the rest of the network can continue to follow the old version but will be unable to validate blocks that follow the updated rules.
    - E.g., Bitcoin's SegWit (Read more).

Fork

- **Intentional fork**: due to changes of the blockchain rules.
  - **Hard fork**: not backward-compatible change; all users are required to upgrade their software;
    - E.g., Ethereum/Ethereum Classic, Bitcoin/Bitcoin Cash.
  - **Soft fork**: backward-compatible change; the rest of the network can continue to follow the old version but will be unable to validate blocks that follow the updated rules.
    - E.g., Bitcoin's SegWit (Read more).
- **Accidental fork** (or, temporary fork) are temporary inconsistencies of the blockchain:
  - They are resolved as more blocks are added to one of the forks;
  - The chain representing the most Proof-of-Work should be selected:
    - In Bitcoin, it is the *longest chain*;
    - In Ethereum, it is the *heaviest chain*.
  - What happens if the same transaction is in more than a chain?

Double Spending

Double Spending

The same single digital token can be spent more than once.

- Easily solvable in centralized systems; how solved in decentralized systems?
- Consensus protocol
  - Still, if consensus finality is not deterministic, the double spending is still an issue:
  - Two blocks with **conflicting transactions** mined at the same approximate time.
  - As new blocks arrive, they must commit to one history or the other, and eventually a single chain will continue on, while the other(s) will not.
  - Since the heaviest chain is considered to be valid, miners are incentivized to only build blocks on that chain.
  - As blocks are built on top of a transaction, it becomes increasingly costly and thus unlikely for another chain to overtake it.

Double Spending and 51% Attack

- Double spending can be exploited for attacks, such as the popular 51% attack of proof-of-work and proof-of-stake blockchains.
  - More on 51% attack in the Consunsus protocols section.
- Bitcoin requires waiting a certain number of confirmations (6) before considering the transaction spent.
  - However, the transaction can still be reverted!
  - But its probability decreases as new blocks are attached to the chain containing the transaction.

Programmability

Blockchains enable the execution of *smart contracts* to execute payments or to carry out actions upon the occurrence of specific conditions:

- **Immutable**: Once deployed, the code of a smart contract cannot change;
- **Deterministic**: The execution outcome is the same for everyone who runs it;
- All information needed is **contained** within the script, transaction, or the blockchain (no external dependencies);

Programmability

Blockchains enable the execution of *smart contracts* to execute payments or to carry out actions upon the occurrence of specific conditions:

- **Immutable**: Once deployed, the code of a smart contract cannot change;
- **Deterministic**: The execution outcome is the same for everyone who runs it;
- All information needed is **contained** within the script, transaction, or the blockchain (no external dependencies);
- Different blockchains adopt languages with different expressivity:
  - Bitcoin's Script enables the specification of spending conditions on payments. Limited expressivity, but enough to cover most of business use cases (e.g., verify a digital signature).
  - Ethereum's Smart Contracts exploit a Turing-complete languages, enabling the creation of novel tokens (e.g., programmable money, NFT).

Fungible and Non-fungible Tokens

Crypto Coin

A form of digital currency that are often native to a blockchain, with the main purpose of storing value and working as a medium of exchange.

- Fungible Tokens
  - **fungibility**: is the property of a good (or a commodity) whose individual units are essentially interchangeable, and each of whose parts are indistinguishable from any other part;
  - Differently from coins, tokens are digital assets built on top of an existing blockchain (using smart contracts);
  - Wide variety of functions: from representing a physical object to granting access to platform-specific services and features.
  - Standards: Ethereum ERC-20.
  - Example: Stablecoins, e.g., Tether USD, USD Coin, DAI.

Fungible and Non-fungible Tokens

- Non-Fungible Token (NFT)
  - A unique digital identifier that is recorded on a blockchain;
  - It is used to certify ownership and authenticity: cannot be copied, substituted, or subdivided;
  - Its ownership is recorded in the blockchain and can be transferred by the owner;
  - Standard: Ethereum ERC-721.
  - Examples: "Everydays: The First 5000 Days" ($69.3 million), CryptoPunks ($7–23 million), Bored Ape ($50–60k)

The Bored Ape Yacht Club is a collection of 10k unique NFTs living on Ethereum.

Popular Blockchains

Bitcoin (BTC)

- A **protocol** that supports decentralized anonymous peer-to-peer digital currency;
- A publicly disclosed **ledger** of transactions;
- A **reward**-driven system for achieving **consensus** (mining) based on:
    - Proof-of-Work (PoW) for helping to secure the network;
    - *Longest-chain* policy;
- A **scarce token** economy with an eventual cap of about 21M bitcoins.

Bitcoin: Market Value

Ethereum (ETH)

- Not only focused on digital currency, but aimed to realize the so-called *World Computer*;
- A decentralized platform that runs smart contracts;
- Defined using a Turning complete language (e.g., *Solidity*, Vyper);
- A virtual machine for cryptocurrency (Ethereum Virtual Machine—EVM);
  - Executes (deterministic) smart contracts;
  - Allows creating and transferring currencies;
  - Allows creating and transferring fungible tokens and non-fungible tokens (NFTs).

Ethereum: Market Value

Ethereum ETH

Ether (ETH) is the native cryptocurrency that powers Ethereum. It's primarily used to pay transaction fees and the creation of blockchain smart contracts.

Price History

€1,778.95 • 09:30
Vol 12,445,591,251 ETH

1D 1W 1M 1Y **MAX** EUR

Key Concepts from Financial Market

Crypto asset

- A digital representation of value that you can transfer, store, or trade electronically;
- Broad definition, which includes:
  - **Cryptocurrencies**: digital currencies, e.g., Bitcoin, Ether;
  - **Utility tokens**: represent token to access specific services;
  - **Security tokens** (or equity tokens): cryptographic tokens representing a share of a company that emitted the token (e.g., give voting rights)
  - Both fungible and non-fungible tokens (NFT).
- **NFT**: special type of *token* representing a **unique** (digital or physical) good or object; hence, NFTs are not inter-changeable.
  - E.g., the Mona Lisa painting.

Stablecoin

Stablecoins are cryptocurrencies whose value is pegged, or tied, to that of another currency, commodity, or financial instrument.

- Aimed to provide an alternative to high volatility of most popular cryptocurrencies;
- Different types:
  - Fiat-Collateralized Stablecoins: maintain a reserve of a fiat currency (e.g., USD) as collateral assuring the stablecoin's value. Such reserves are maintained by independent custodians and are regularly audited.

Stablecoin

Stablecoins are cryptocurrencies whose value is pegged, or tied, to that of another currency, commodity, or financial instrument.

- Aimed to provide an alternative to high volatility of most popular cryptocurrencies;
- Different types:
  - Fiat-Collateralized Stablecoins: maintain a reserve of a fiat currency (e.g., USD) as collateral assuring the stablecoin's value. Such reserves are maintained by independent custodians and are regularly audited.
  - Crypto-collateralized stablecoins: backed by other cryptocurrencies. Being such reserve prone to high volatility, these stablecoins are overcollateralized

Stablecoin

Stablecoins are cryptocurrencies whose value is pegged, or tied, to that of another currency, commodity, or financial instrument.

- Aimed to provide an alternative to high volatility of most popular cryptocurrencies;
- Different types:
  - Fiat-Collateralized Stablecoins: maintain a reserve of a fiat currency (e.g., USD) as collateral assuring the stablecoin's value. Such reserves are maintained by independent custodians and are regularly audited.
  - Crypto-collateralized stablecoins: backed by other cryptocurrencies. Being such reserve prone to high volatility, these stablecoins are overcollateralized
  - Algorithmic stablecoins: may or may not hold reserve assets. They keep the stablecoin's value stable by controlling its supply through an algorithm.

Stablecoin

- Many stablecoins: *(crypto.com/stablecoins)*
  - Fiat-Collateralized: e.g., Tether, USDCoin, Binance USD, MakerDAO (DAI);
  - Crypto-collateralized: e.g., MakerDAO (DAI);
  - Algorithmic: e.g., DefiDollar (DUSD), Ampleforth (AMPL);
- Most of stablecoins are fiat-collateralized and backed in $ (USD).
- Negatively affected by, e.g., the failure of Silicon Valley Bank (read)

Currency Exchange

A (cryptocurrency or digital currency) exchange:

- Is a platform for trading cryptocurrency for other assets and traditional currencies (e.g., EUR, USD);
- Provides a level of anomicity for users and transparency of both trading parties;
- May accept credit card payments or other forms of payment;
- Requires the payment of a commission:
  - the bid–ask spreads as a transaction commission;
  - or, simply charges fees.

Currency Exchange

Examples:

- Binance, Gate.io, OKY, Coinbase Exchange, PrimeXBT, Zengo Wallet, Kraken, Crypto.com
- More than 220 exchanges; an extensive list: CoinMarketCap

#	Exchange	Trading volume(24h)	Avg. Liquidity	Weekly Visits	# Markets	# Coins	Fiat Supported	Volume Graph (7d)
1	Binance	$6,183,458,223	893	9,150,960	2096	633	ARS, AUD, BRL and +86 more	
2	Coinbase Exchange	$1,065,544,846	764	74,849	504	387	USD, AED, ARS and +61 more	
3	Upbit	$894,224,224	537	1,751,171	699	311	KRW	
4	OKX	$1,185,552,820	754	5,044,570	1343	376	EUR, BRL, AUD and +3 more	
5	Bybit	$1,536,085,636	721	2,763,868	1211	691	BRL, CHF, CZK and +72 more	
6	Bitget	$635,080,553	656	3,786,065	1261	731	EUR, MXN, ZAR and +12 more	
7	Gate	$1,363,704,443	785	2,910,277	2614	1850	EUR, BRL, KES and +61 more	

Decentralized Finance (DeFi)

A key feature of blockchains is disintermediation, i.e., the ability to move tokens (coins) without relying on trusted third parties.

Decentralized finance (DeFi):

- Offers **financial instruments** without relying on intermediaries (e.g., brokers, exchanges, or banks) by using **smart contracts** on a blockchain.
- Platforms allow people to **lend** or **borrow** funds from others, trade cryptocurrencies, exchange/swap assets, insure against risks, and earn interest in savings-like accounts.

Decentralized Finance (DeFi)

Decentralized finance (DeFi):

- Users can directly exchange transactions among them; safety is guaranteed by the blockchain:
  - The blockchain stores the history of transactions and state of balances;
  - Crypto-currencies are used as assets;
  - Smart contracts are used to implement DeFi applications.
- Popular blockchain used for DeFi applications: Ethereum, Cardano, Binance, and Solana.

Decentralized Finance (DeFi)

- The core characteristics of DeFi may appear utopian, but the development and adoption have already begun to accelerate.
- DeFi applications like **Uniswap** and **SushiSwap** allow users to swap and exchange fungible tokens (ERC20) in the Ethereum ecosystem.

Why a Blockchain?

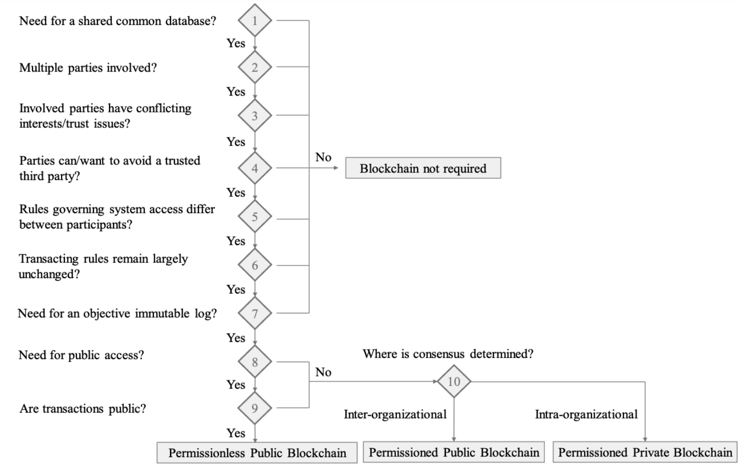
Blockchain vs Database

	Blockchain	Database
Authority	Decentralized (Permissioned are more centralized)	Administrator
Architecture	Peer-to-peer	Client-server
Data Handling	Read/Write	CRUD
Integrity	Cryptographically enforced	Malicious actors can alter data
Performance	Slowed down by verification and consensus	Fast and better scalability

Distributed Database vs DLT vs Blockchain

- A distributed database assumes a logically centralized control;
  - Example: Apache Cassandra, DHT, Google Spanner.
- Differently from a distributed database, a DLT assumes an adversarial model;
  - (Usually,) presence of malicious nodes assumed;
  - Example: R3 Corda.
- Different from a DLT, a blockchain structures transactions in a chain of cryptographically linked blocks and uses a global data broadcast.
  - Example: Bitcoin, Ethereum, Algorand.

When to Use a Blockchain?



Pedersen et al. "A Ten-Step Decision Path to DetermineWhen to Use Blockchain Technologies", MIS Quarterly Executive: Vol. 18: Iss. 2, Article 3. 2019.

Matteo Nardelli