[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 0 - Getting Started](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Getting Started](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Blockchain 101](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Blockchain History](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Public and Managed Blockchains](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Consensus in Distributed Networks](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Cryptography](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Self-Assessment Quiz](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Go Introduction - First Steps](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Go Basics](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Go Interfaces](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Control Structures in Go](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Arrays and Slices in Go](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Standard Packages in Go](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Concurrency in Go](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Docker Introduction](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Introduction to the Interchain](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[The Interchain Ecosystem](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Getting ATOM and Staking It](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[A Blockchain App Architecture](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Accounts](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Transactions](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Messages](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Modules](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Protobuf](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Multistore and Keepers](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[BaseApp](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Queries](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Events](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Context](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Testing](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Relaying with IBC](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Interchain Security](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Bridges](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Migrations](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 1 Quiz](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 2 - First Steps](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[First Steps](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Setup Your Work Environment](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Run a Node, API, and CLI](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Ignite CLI](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Store Object](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create Custom Messages](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create and Save a Game Properly](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Add a Way to Make a Move](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Emit Game Information](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Record the Game Winner](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 2 Exercise](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[What is IBC?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC Token Transfer](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC Tooling](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[What is CosmJS?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Your First CosmJS Actions](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Compose Complex Transactions](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Learn to Integrate Keplr](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Put Your Games in Order](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Auto-Expiring Games](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Let Players Set a Wager](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Handle wager payments](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Integration tests](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Incentivize Players](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Help Find a Correct Move](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Understand IBC Denoms](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Go Relayer](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Hermes Relayer](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[CosmJS Advanced](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create Custom Objects](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create Custom Messages](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Get an External GUI](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Backend Script for Game Indexing](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC Deep Dive](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[IBC Application Developer Introduction](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Create a Leaderboard Chain](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Run in Production](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Prepare the Software to Run](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Prepare a Validator and Keys](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Prepare Where the Node Starts](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Prepare and Do Migrations](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Simulate Production in Docker](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Tally Player Info After Production](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Simulate a Migration in Docker](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Final Exam](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[What's Next?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

[Continue Your Interchain Journey](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html)

Docs Version Switcher

On this page

[What is a blockchain? And what does it solve?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#what-is-a-blockchain-and-what-does-it-solve)

[Securing peer-to-peer networks](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#securing-peer-to-peer-networks)

[P2P networking & distributed networks](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#p2p-networking-distributed-networks)

[How does blockchain prevent double-spending?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-prevent-double-spending)

[How does blockchain work?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-work)

[Deterministic, atomic operations - all or nothing](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#deterministic-atomic-operations-all-or-nothing)

[Blockchain as a new computing paradigm](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#blockchain-as-a-new-computing-paradigm)

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#blockchain-101) **Blockchain 101**

This first chapter demystifies the foundations of blockchain technology. This background information will be important as you progress through the course.



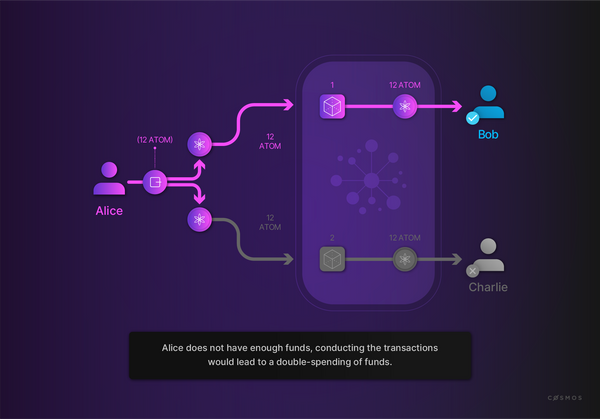
The modules include learning materials for your studies, as well as examples and exercises for you to practice. You will also find freely accessible resources from other sites to help you dive deeper into the subject matter at hand.

To understand Cosmos, one must first understand blockchain technology. So, let's start.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#what-is-a-blockchain-and-what-does-it-solve) What is a blockchain? And what does it solve?

Blockchain technology is rapidly evolving and is difficult to understand in its entirety. It took many bright minds many years of development, plus a combination of advances in cryptography, distributed computing, and economics to create this nascent technology. Grasping blockchain technology requires both time and effort.

Blockchain is a solution to a particular problem: the **double-spending problem**.



We are all familiar with digital artifacts, like text files or images, and the ease with which they can be copied. This presents obvious problems if digital artifacts are intended to represent assets with value. When it comes to such digital tokens, a more pressing issue is the possibility of spending a token more than once, also known as double-spending. What prevents someone from making copies and spending the same token twice?

The double-spending problem refers to the challenge of designing a digital cash system in which tokens are digital artifacts but cannot be spent more than once.

Satoshi Nakamoto, whose identity remains shrouded in mystery, published his seminal whitepaper in October 2008. The paper presented **a solution to the double-spending problem for digital currencies**. In doing so, he revealed the underlying technology known as blockchain, and an example of blockchain's possible application in the form of a simple implementation called **Bitcoin**.



Want to take a closer look at the first blockchain implementation whitepaper? [Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System (opens new window)↗](https://bitcoin.org/bitcoin.pdf) - it is a fairly straightforward paper.

Bitcoin has gained widespread attention since then. The underlying technology, blockchain, has also gained recognition and has found applications in diverse contexts. Indeed, the world has discovered blockchain's usefulness in many environments and a great variety of possible use cases. Cosmos is one such blockchain implementation.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#securing-peer-to-peer-networks) Securing peer-to-peer networks

Security is challenging in peer-to-peer (P2P) networking for two reasons:

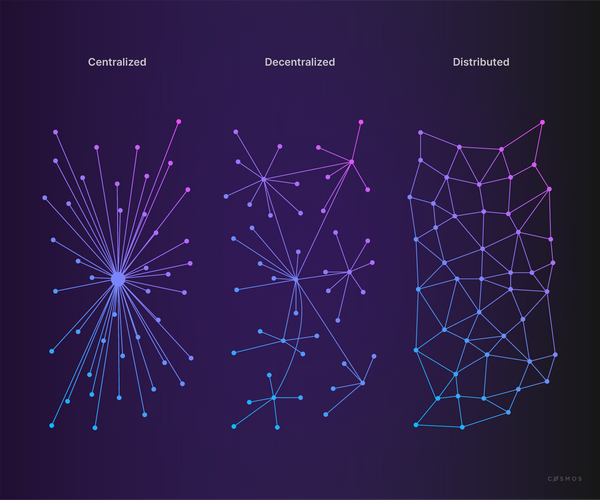
* P2P software has to be downloaded to join a network, making it especially vulnerable to remote exploits.
* Malicious participants can send incorrect requests or responses, as well as malware, and because of interconnectivity corrupted data may propagate throughout the network.

Other security risks include denial-of-service (DoS) attacks, routing attacks, and routing network partitions.

A "secure" P2P network needs to repel malicious and erroneous input.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#p2p-networking-distributed-networks) P2P networking & distributed networks

Networks can be centralized, decentralized, distributed, or decentralized **and** distributed.



A blockchain is a distributed ledger that records all transactions on a network. All nodes in such a distributed ledger need a copy of said ledger. The network and ledger need to run continuously even if nodes join and leave unpredictably. Nodes that join the network must be able to sync up with the latest ledger state. The ledger state must be secure, with strong defenses to prevent malicious nodes from inserting invalid information.

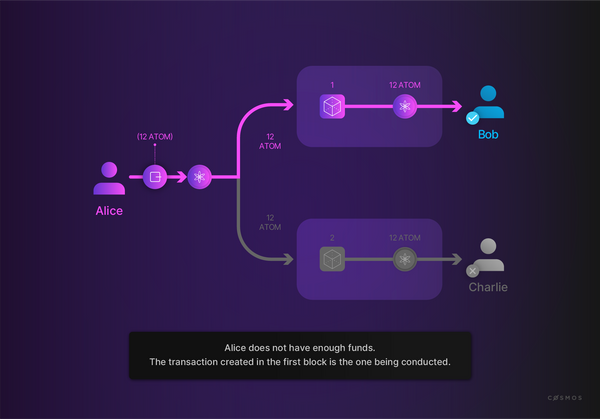
Therefore, a blockchain is a highly-available network that ensures a stateful protocol, always operating as designed and never permitting a departure from well-defined rules.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-prevent-double-spending) How does blockchain prevent double-spending?

In the current financial system, double-spending is avoided by involving legacy actors and institutions of the financial sector, i.e. centralized third parties that manage and control financial transactions. A third party such as a bank, credit card company, or payment service is used as a trusted ledger keeper. They maintain the digital ledgers and do not allow funds to be spent twice.

Consequently, it is generally not possible for two parties to exchange value online without involving a trusted third party to handle the settlement process and update their ledgers, as well as the account balances.

At a high level, Bitcoin solves the double-spending problem by replacing the trusted, central ledger-keepers with a decentralized and distributed ledger that is maintained by a large network of ledger-keepers. Each member of the network has an exact replica of the ledger, and no one can update the ledger without establishing consensus with the other ledger-keepers.



It is as though each transaction is observed by a *large crowd of witnesses* who reach consensus about proposed changes. The crowd prohibits events that should not occur, such as spending the same funds twice (i.e. double-spending).

Bitcoin and its underlying technology convincingly demonstrate that a network of participants that do not necessarily trust each other can achieve consensus about the validity of a transaction, its history, and the resulting state of the ledger. This is interesting, because simple account balance ledgers and protocols to move funds are far from the only use cases for distributed consensus.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-work) How does blockchain work?

Imagine you want to **retain and monitor changes** to a file, for example a log file. Now, imagine you also want to verify an unbroken history of all changes ever made to the file. How can you proceed?

A well-understood solution uses **cryptographic** [hash functions (opens new window)↗](https://en.wikipedia.org/wiki/Cryptographic_hash_function). Let us briefly introduce this concept in case you are unfamiliar with them.

The ideal cryptographic hash function has **five main properties**:

* **Deterministic:** the same message always results in the same hash.
* **Fast:** the hash value for any given message is computed quickly.
* **Resistant:** it is not feasible to generate a message from its hash value except by trying all possible messages.
* **Uncorrelated:** a small change to a message alters the hash value so extensively that the new value shows no relation to the old.
* **Collision-resistant:** it is infeasible to find two different messages with the same hash value.



You can see hashing in action to get the feel for it here: [https://www.browserling.com/tools/all-hashes (opens new window)↗](https://www.browserling.com/tools/all-hashes). As you type into the text box, the hash updates automatically. Even a minuscule change to the input creates completely different hashes. You can also see that different hashing algorithms produce different output. Hash algorithms have evolved over time, often for security reasons. Try it out!

Notice that there are many different hashing algorithms that aim for similar results and fulfill the properties described above. Each algorithm consistently produces hashes of the same size regardless of the input's size.

A hash can be used to prove an input exactly matches the original, but the original cannot be reconstructed from a hash. So, a hash function can demonstrate that a copy of the file is an authentic replica of the original in every detail.

What about subsequent changes to the file? Suppose you want to demonstrate that a series of changes is authentic, complete, and correctly ordered?

Accountants have known the advantages of "append-only" ledgers for centuries, so suppose that you will only add new entries (i.e. *make changes*) to the end of the file. You start with an empty file and then append a series of changes that, when replayed in order, will produce the current state of the file - Git users are familiar with this concept.

How can a hash function help you be certain that a series of entries is the unbroken chain of inputs? You make a rule that states that, in addition to the new content, the previous hash will also be an input of the *next* hash.

The (pseudo-)code would look like this:



Copy

version2Hash = hash({

changes: version2Changes,

previousVersion: version1Hash

})

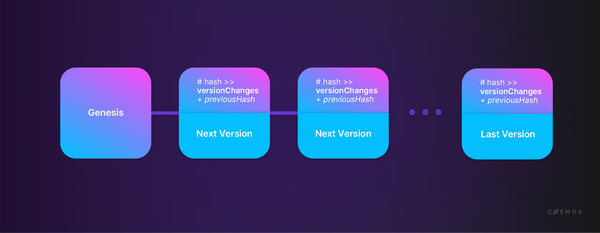
This way, you can examine *candidate* changes, and confirm that the proposed changes belong to the known, authentic, previous version of the file. This method ensures that changes to the file are accurately disclosed.

This process repeats for all subsequent versions. Any version of the file contents can be shown to be part of an **unbroken chain of changes** all the way back to the file's inception. This is pure mathematics.

Any departure from the system (e.g. a hash does not compute as expected) proves a break in the history and is therefore invalid.

Since the hash of the latest valid version is an input to the next version's hash function, it is not possible to generate a new valid version without knowledge of the valid version that precedes it. This process forces changes to be **appended** to a previous valid version.

Blockchains function similarly: blocks of transactions are appended, using hashes of previous blocks as inputs into hashes of subsequent blocks. Any participant can quickly verify an unbroken chain of blocks (i.e. the correct historical order).



**Transaction blocks** are logical units that wrap a set of transactions in a specific order. While the implementation details are somewhat more subtle, for now think of this as a set of transactions that occurred during a specific time interval and in a specific order.

**Transaction ordering** is surprisingly challenging in a distributed system due to design goals and constraints. The example of Bitcoin and its novel solution is valuable to understanding how this challenge can be addressed.

There are some important constraints to keep in mind when talking about transaction ordering in distributed networks like Bitcoin:

* As it is a distributed network, **everyone** has a little **authority**. For example, all nodes can propose transactions and then announce the information to other nodes. Theoretically, anyone can listen to transaction proposals and organize a valid block containing an *opinion* about the correct order of events.
* In a truly distributed network, no one's clock is considered more authoritative than anyone else's clock: **a blockchain is a distributed timestamp server without a central network time**.
* Because of **physics and network latency**, even if all members of a network mean well and participate honestly, everyone in the network will learn about transaction proposals in a slightly different order and each node will arrive at a slightly different opinion about the ordering of transactions.

So, how is the correct **order of transactions** determined?

Although there is no obvious way to settle it, **transaction order must be resolved** because processing transactions out of order would produce non-trivial differences in outcomes. Such a non-trivial difference could be an instance of double-spending. Without agreement about the transaction order, there can be no agreement about the balance of accounts.

Bitcoin uses a **consensus algorithm**, a process called **Proof-of-Work (PoW)**, which (simplistically) can be thought of as a lottery. In this process, the lucky winner gets the privilege of being the authority for one block of transactions. The winning lottery ticket, called the **nonce**, is used as an input for a hash function, the result of which needs to have certain properties that are arbitrarily difficult to achieve - more on this later. However, the "winning" result can be very easily verified by other participants. The verified lottery winner's opinion about the order of transactions within the block becomes the network's de facto official result, provided that all proposed transactions are valid and include the necessary signatures from the transaction originators.

In case the foregoing is not clear:

* A valid block is a well-ordered **set of transactions**.
* Each block contains the **hash** of the previous block.
* Each block also includes a "winning lottery ticket", the **nonce**.
* The hash of the block has properties that are especially difficult to generate but very easy to verify.



In Bitcoin, the block hash must start with a certain number of zeroes (e.g. 00001B3d). The only way to achieve that is to find a number, the nonce, that can be inserted into the hash function's input so the hash created meets that criteria. There is no more efficient method of finding a suitable nonce than a brute force search of random numbers.

Discovering a nonce which, when calculated with the previous hash value, generates a new hash value *that satisfies the necessary criteria*, is by definition evidence of considerable computational work because brute force is the only known method of finding it. Creating a new valid block is therefore not a trivial event, and other participants recognize this unlikely combination.

This process disambiguates the order of transactions even though well-meaning nodes independently arrive at slightly different opinions about the matter. The process does so **without relying on an authoritative time source**.

**A well-ordered set of blocks that each contain well-ordered transactions is a well-ordered set of all transactions that have ever occurred.**

Cryptographic hash functions are instrumental, in that they empower all participants to ensure that they possess an undistorted history of everything. Since all nodes can verify the chain independently, they can proceed on the assumption that all other nodes will eventually come into agreement about the history of everything. This is known as **eventual consensus**.

What are the implications of blockchain's way of ordering transactions and blocks on its overall state?

A blockchain starts with a known state. This is a simple matter of an initialized universe in which nothing has happened. It is often referred to as the **genesis block**.

It proceeds by constructing a verifiable and widely agreed history of everything that has ever happened on an append-only basis. Nodes independently construct a present state of the universe by reviewing the ordered history of every change (i.e. the transactions) that has ever occurred. This comprehensive history moves forward in time as "lottery winners" announce new transaction blocks and these are accepted as valid by a consensus of network participants.

Thus, transactions being included in blocks and those becoming part of the chain leads to the state changing and being amended.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#deterministic-atomic-operations-all-or-nothing) Deterministic, atomic operations - all or nothing

In computer science, an event is said to be "atomic" if it cannot be split into smaller parts. For example, the statement x = y is atomic if the language guarantees that y cannot be partially copied to x.

In the world of databases, atomicity is often specified by the developer by grouping multiple operations in a wrapper such as a *commit* or *rollback* block, to ensure that all of the steps are complete or none of the steps execute at all. This method is often used to ensure database integrity.

In the context of a blockchain, a transaction is a single instruction allowed by the protocol, signed as required by the protocol, and sent to the blockchain through a local node connected to the network. Transactions are either completely successful or they fail. Generally, the actual result cannot be known with certainty until the transaction is included in a block to establish an execution order. For example, a transaction to send funds from Alice to Bob depends on Alice's balance at execution time.

All nodes must arrive at the same conclusion; given a transaction in a certain sequence, all nodes must agree on the result. This means the protocols must be deterministic. Either the transaction was successful or it failed, and the effect must be indisputable.

Therefore, **blockchain transactions are both deterministic and atomic**.

[#Copy link](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#blockchain-as-a-new-computing-paradigm) Blockchain as a new computing paradigm

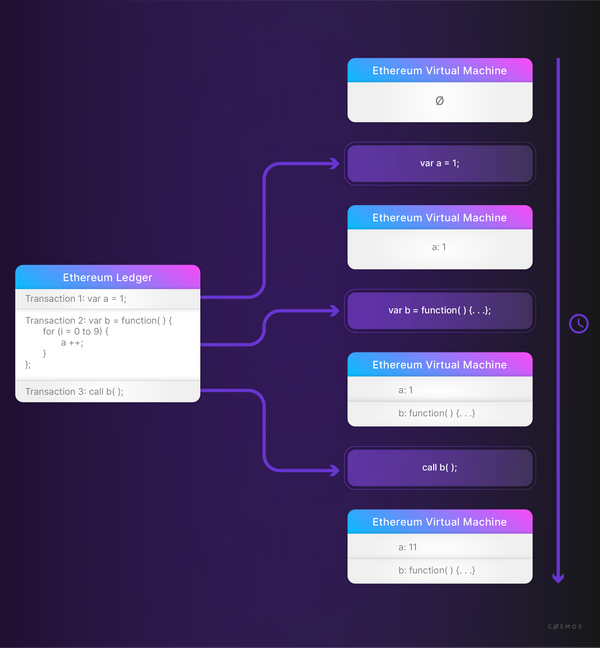
Imagine a computer. How do you describe its process?

It takes commands and then executes them in an ordered and deterministic fashion. The sequence of execution determines the state the computer is in.

Now apply this thought to blockchain technology.

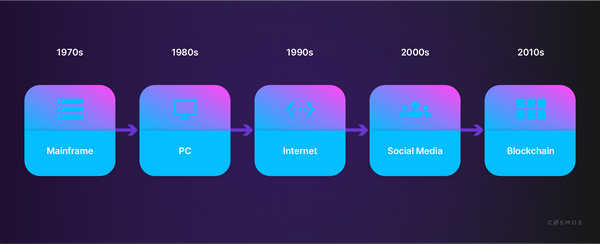
Ethereum was the first blockchain to implement [Turing-complete (opens new window)↗](https://csrc.nist.gov/glossary/term/turing_complete) smart contracts. Each transaction is in effect a command, and the "execution" of those commands results in a new state, the Ethereum computer's state. An Ethereum blockchain defines a distributed state machine with a state all nodes agree on.

A **smart contract** is a program on "the world computer", as Ethereum has been called. A distributed application (dApp) is a program too, as is a coordinated collection of smart contracts.



Major computing paradigms in the past were:

* The **mainframe** in the 1970s.
* The **personal computer** in the 1980s.
* The **internet** in the 1990s.
* **Mobile and social networking**, often referred to as "social media", in the 2000s.



It is argued that the development of blockchain technology is initiating a new computational paradigm.

In some opinions, mobile and social networking and the internet have been exploited regarding their commercial use. For some, the ideal of an interconnected community - in which information and knowledge are available and shared and where egalitarian values like equality, transparency, and freedom are encouraged - has led to disappointment over what could have been but never was.

Blockchain technology is often seen as presenting the possibility of more private, transparent, and equal forms of interaction and community with the possibility of preventing imbalances of power that often exist when one group is in charge of a widely used platform and can exercise privileges that are not enjoyed by the rest of the participants.



**Further readings**

* [Wikipedia: Double-spending (opens new window)↗](https://en.wikipedia.org/wiki/Double-spending)
* [Satoshi Nakamoto (2008): Bitcoin: A Peer-to-Peer Electronic Cash System (opens new window)↗](https://bitcoin.org/bitcoin.pdf)

synopsis

To summarize, this section has explored:

* How blockchain is a form of append-only ledger, a database in which immutably ordered blocks of immutably ordered transactions provide an easily verifiable historical record that can be held and updated across multiple nodes of a network.
* How blockchain is a solution to the **double-spending problem**, which provides a way to ensure that digital tokens (such as of an online currency) can only be spent on one occasion, despite the relative ease with which digital files and artifacts can be copied.
* How networks can generally be categorized as **centralized**, **decentralized**, or **distributed**, with blockchain using the advantages of distributed, decentralized networking to minimize or completely remove the need for any form of centralized authority.
* How blockchain uses the *deterministic*, *fast*, *resistant*, *uncorrelated*, and *collision-resistant* properties of **hash functions** to allow for easy and rapid verification of an unbroken history of all changes ever made to its ledger of transactions.

previous

[](https://ida.interchain.io/ida-course/LPs/week-0/)

**[Getting Started](https://ida.interchain.io/ida-course/LPs/week-0/)**

up next

**[Blockchain History](https://ida.interchain.io/ida-course/0-blockchain-basics/2-public.html)**

[[](https://ida.interchain.io/ida-course/0-blockchain-basics/2-public.html)](https://ida.interchain.io/ida-course/0-blockchain-basics/2-public.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

On this page

[What is a blockchain? And what does it solve?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#what-is-a-blockchain-and-what-does-it-solve)

[Securing peer-to-peer networks](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#securing-peer-to-peer-networks)

[P2P networking & distributed networks](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#p2p-networking-distributed-networks)

[How does blockchain prevent double-spending?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-prevent-double-spending)

[How does blockchain work?](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#how-does-blockchain-work)

[Deterministic, atomic operations - all or nothing](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#deterministic-atomic-operations-all-or-nothing)

[Blockchain as a new computing paradigm](https://ida.interchain.io/ida-course/0-blockchain-basics/1-blockchain.html#blockchain-as-a-new-computing-paradigm)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)