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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 0 - Getting Started](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Getting Started](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Blockchain 101](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Blockchain History](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Public and Managed Blockchains](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Consensus in Distributed Networks](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Cryptography](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Self-Assessment Quiz](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Go Introduction - First Steps](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Go Basics](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Go Interfaces](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Control Structures in Go](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Arrays and Slices in Go](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Standard Packages in Go](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Concurrency in Go](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Docker Introduction](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[The Interchain Ecosystem](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Getting ATOM and Staking It](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[A Blockchain App Architecture](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Accounts](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Transactions](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Messages](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Modules](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Protobuf](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Multistore and Keepers](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[BaseApp](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Queries](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Events](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Context](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Testing](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Relaying with IBC](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Interchain Security](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Bridges](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Migrations](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 1 Quiz](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 2 - First Steps](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[First Steps](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Setup Your Work Environment](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Run a Node, API, and CLI](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Ignite CLI](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Store Object](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create and Save a Game Properly](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Add a Way to Make a Move](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Emit Game Information](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Record the Game Winner](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 2 Exercise](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[What is IBC?](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC Token Transfer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC Tooling](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[What is CosmJS?](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Your First CosmJS Actions](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Compose Complex Transactions](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Learn to Integrate Keplr](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Put Your Games in Order](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Auto-Expiring Games](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Let Players Set a Wager](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Handle wager payments](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Integration tests](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Incentivize Players](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Help Find a Correct Move](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Understand IBC Denoms](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Go Relayer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Hermes Relayer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create Custom Objects](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Get an External GUI](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Backend Script for Game Indexing](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[IBC Application Developer Introduction](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Create a Leaderboard Chain](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Run in Production](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Prepare the Software to Run](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Prepare a Validator and Keys](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Prepare Where the Node Starts](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Prepare and Do Migrations](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Simulate Production in Docker](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Tally Player Info After Production](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Simulate a Migration in Docker](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Final Exam](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[What's Next?](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

[Continue Your Interchain Journey](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html)

Docs Version Switcher

On this page

[Internet of blockchains](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#internet-of-blockchains)

[High-level overview of IBC](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#high-level-overview-of-ibc)

[IBC/TAO - transport layer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-tao-transport-layer)

[IBC/APP - application layer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-app-application-layer)

[Security](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#security)

[IBC light clients](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-light-clients)

[Trust the chains, not the bridge](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#trust-the-chains-not-the-bridge)

[Submit misbehavior](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#submit-misbehavior)

[Dynamic capabilities](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#dynamic-capabilities)

[Development roadmap](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#development-roadmap)

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#what-is-ibc) **What is IBC?**



Inter-Blockchain Communication Protocol solves for communication between blockchains, which is particularly important in the Interchain universe.   
  
In this section, you will learn:

* What IBC is.
* How IBC works.
* More about the IBC security guarantees.

The [Inter-Blockchain Communication Protocol (IBC) (opens new window)↗](https://ibcprotocol.org/) is *a protocol to handle authentication and transport of data between two blockchains*. IBC **requires a minimal set of functions**, specified in the [Interchain Standards (ICS) (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/ics-001-ics-standard). Notice that those specifications do not limit the network topology or consensus algorithm, so IBC can be used with a wide range of blockchains or state machines. The IBC protocol provides a permissionless way for relaying data packets between blockchains, unlike most trusted bridging technologies. The security of IBC reduces to the security of the participating chains.

IBC solves a widespread problem: cross-chain communication. This problem exists on public blockchains when exchanges wish to perform swaps. The problem arises early in the case of application-specific blockchains, where every asset is likely to emerge from its own purpose-built chain. Cross-chain communication is also a challenge in the world of private blockchains, in cases where communication with a public chain or other private chains is desirable. There are already IBC implementations for private blockchains [such as Hyperledger Fabric and Corda (opens new window)↗](https://www.hyperledger.org/blog/2021/06/09/meet-yui-one-the-new-hyperledger-labs-projects-taking-on-cross-chain-and-off-chain-operations).

Cross-chain communication between application-specific blockchains in the Interchain creates the potential for high horizontal scaleability with transaction finality. These design features provide convincing solutions to well-known problems that plague other platforms, such as transaction costs, network capacity, and transaction confirmation finality.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#internet-of-blockchains) Internet of blockchains

IBC is essential for application-specific blockchains like the ones in the Interchain network. It offers a standard communication channel for applications on two different chains that need to communicate with each other.

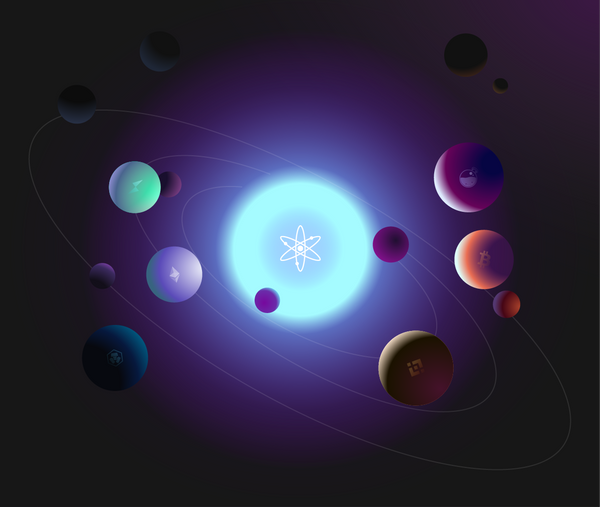
Most Interchain applications execute on their own purpose-built blockchain running their own validator set (at least before the introduction of [Interchain Security (opens new window)↗](https://informal.systems/2022/05/09/building-with-interchain-security)). These are the application-specific blockchains built with the Cosmos SDK. Applications on one chain may need to communicate with applications on another blockchain, for example, an application could accept tokens from another blockchain as a form of payment. Interoperability at this level calls for a method of exchanging data about the state or the transactions on another blockchain.

While such bridges between blockchains can be built and do exist, they are generally constructed ad-hoc. IBC provides chains with a common protocol and framework for implementing standardized inter-blockchain communication. For chains built with the Cosmos SDK, this comes out of the box, but the IBC protocol is not limited to chains built with the Interchain Stack.



More details on the specifications will follow in the next section, but notice that IBC is not limited to Cosmos blockchains. Solutions can even be found for cases where some requirements are not initially met. For example, IBC was already providing connectivity between the Interchain and the Ethereum blockchain before "the Merge", which saw Ethereum migrate from a Proof-of-Work (PoW) model to Proof-of-Stake (PoS).   
  
As a PoW consensus algorithm does not ensure finality, one of the main requirements to use IBC is not met. Therefore, compatibility with Ethereum was enabled by creating a peg-zone where probabilistic finality is considered deterministic (irreversible) after a given threshold of block confirmations. This solution can serve any IBC connection to a PoW blockchain.

Although application-specific blockchains offer superior (horizontal) scalability compared to general-purpose blockchain platforms, smart contract development for general-purpose chains and generic virtual machines (VMs) like in Ethereum offer their own benefits. IBC provides a method of incorporating the strengths of general-purpose and application-specific blockchains into unified overall designs. For example, it allows a Cosmos chain tailored towards performance and scalability to use funds that originate on Ethereum and possibly record events in a Corda distributed ledger; or, in the reverse, a Corda ledger initiating the transfer of underlying assets defined in the Interchain or Ethereum.



With cross-chain communication via IBC, a decentralized network of independent and interoperable chains exchanging information and assets is possible. This "internet of blockchains" brings the promise of increased and seamless scalability. In the Interchain, the vision being implemented is to have a universe of independent chains that are all connected using peg-zones as bridges between the Interchain network and chains outside of it, and connecting all chains via hubs. All of these make up the internet of blockchains.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#high-level-overview-of-ibc) High-level overview of IBC

The transport layer (TAO) provides the necessary infrastructure to establish secure connections and authenticate data packets between chains. The application layer builds on top of the transport layer and defines exactly how data packets should be packaged and interpreted by the sending and receiving chains.

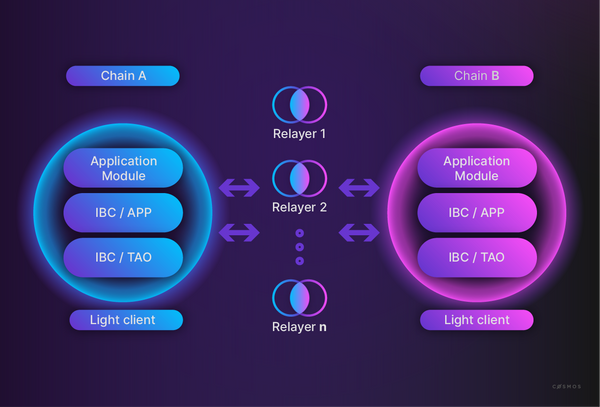


The great promise of IBC is providing a reliable, permissionless, and generic base layer (allowing for the secure relaying of data packets), while allowing for composability and modularity with separation of concerns by moving application designs (interpreting and acting upon the packet data) to a higher-level layer.

This separation is reflected in the categories of the ICS in the [general ICS definition (opens new window)↗](https://github.com/cosmos/ibc/blob/master/spec/ics-001-ics-standard/README.md):

* **IBC/TAO:** Standards defining the Transport, Authentication, and Ordering of packets, i.e. the infrastructure layer. In the ICS, this is comprised by the categories *Core*, *Client*, and *Relayer*.
* **IBC/APP:** Standards defining the application handlers for the data packets being passed over the transport layer. These include but are not limited to fungible token transfers (ICS-20), NFT transfers (ICS-721), and interchain accounts (ICS-27), and can be found in the ICS in the *App* category.

The following diagram shows how IBC works at a high level:



Note three crucial elements in the diagram:

* The chains depend on relayers to communicate. Relayer algorithms ([ICS-18 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/relayer/ics-018-relayer-algorithms)) are the "physical" connection layer of IBC: off-chain processes responsible for relaying data between two chains running the IBC protocol by scanning the state of each chain, constructing appropriate datagrams, and executing them on the opposite chain as is allowed by the protocol.
* Many relayers can serve one or more channels to send messages between the chains.
* Each side of the relay uses the light client of the other chain to quickly verify incoming messages.



If you're interested in another overview of the IBC protocol, in the following video Callum Waters, Engineering Manager for the Tendermind Core, gives a talk on the methodology allowing interoperability between countless sovereign blockchains and how to build an IBC-compatible app.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-tao-transport-layer) IBC/TAO - transport layer

In the diagram, the relationship between the ICS definitions in the category TAO are illustrated - the arrows illustrating the requirements.

Simply put, the transport layer includes:

* **Light Clients** - [ICS-2 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/core/ics-002-client-semantics), [6 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/client/ics-006-solo-machine-client), [7 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/client/ics-007-tendermint-client), [8 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/client/ics-008-wasm-client), [9 (opens new window)↗](https://github.com/cosmos/ibc/tree/main/spec/client/ics-009-loopback-cilent): IBC clients are light clients that are identified by a unique client ID. IBC clients track the consensus state of other blockchains and the proof specs of those blockchains required to properly verify proofs against the client's consensus state. A client can be associated with any number of connections to the counterparty chain.
* **Connections** - [ICS-3 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/core/ics-003-connection-semantics): connections, once established, are responsible for facilitating all cross-chain verifications of an IBC state. A connection can be associated with any number of channels. Connections encapsulate two ConnectionEnd objects on two separate blockchains. Each ConnectionEnd is associated with a light client of the other blockchain - for example, the counterparty blockchain. The connection handshake is responsible for verifying that the light clients on each chain are the correct ones for their respective counterparties.
* **Channels** - [ICS-4 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/core/ics-004-channel-and-packet-semantics): a module on one blockchain can communicate with other modules on other blockchains by sending, receiving, and acknowledging packets through channels that are uniquely identified by the (channelID, portID) tuple. Channels encapsulate two ChannelEnds that are associated with a connection. Channels provide a way to have different types of information relayed between chains, but do not increase the total capacity. Just like connections, channels are established with a handshake.

A channel can be ORDERED, where packets from a sending module must be processed by the receiving module in the order they were sent, or UNORDERED, where packets from a sending module are processed in the order they arrive (which might be different from the order they were sent).

* **Ports** - [ICS-5 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/core/ics-005-port-allocation): an IBC module can bind to any number of ports. Each port must be identified by a unique portID. The portID denotes the type of application, for example in fungible token transfers the portID is transfer.



While this background information is useful, IBC has been designed in such a way that exposure to the transport layer is kept to a minimum for IBC application developers.



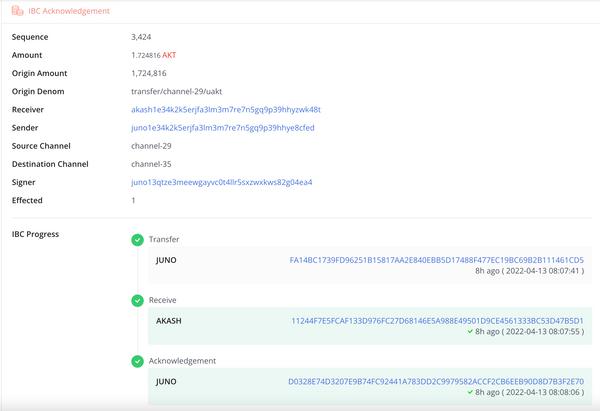
In the following video Colin Axnér of Interchain, a core contributor to ibc-go in the Cosmos SDK, explains how different blockchains can be connected with the Inter-Blockchain Communication (IBC) protocol, with a particular focus on light clients, connections, channels, and packet commitments.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-app-application-layer) IBC/APP - application layer

The ICS also offer definitions for IBC applications:

* **Fungible token transfer** - [ICS-20 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/app/ics-020-fungible-token-transfer): The first and most apparent application for IBC is the transfer of fungible tokens across chains. With the standards set out by ICS-20, a user can send tokens across IBC-enabled chains. This is achieved by escrowing tokens on the source chain: the proof along with the token metadata is relayed to the destination chain, upon which the proof is verified by the light client of the source chain, stored on the destination chain. If the verification passes, vouchers for the tokens on the destination chains are minted and an acknowledgement is sent back to the source chain.

Packet flow is explored in more detail in a later section, but you can look at the steps when following the progress of the IBC token transfer on [Mintscan (opens new window)↗](https://www.mintscan.io/cosmos). The following example shows the transactions submitted for the original Transfer on the source, the Receive message on the destination, and the Acknowledgement again on the source:



* **Interchain accounts** - [ICS-27 (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/app/ics-027-interchain-accounts): interchain accounts outlines a cross-chain account management protocol built on IBC. Chains having enabled ICS-27 can programmatically create accounts on other ICS-27-enabled chains and control these accounts via IBC transactions, instead of having to sign with a private key. Interchain accounts contain all of the capabilities of a normal account (i.e. stake, send, vote) but instead are managed by a separate chain via IBC in a way such that the owner account on the controller chain retains full control over any interchain accounts it registers on host chains.



This list can be and will be extended with time. New concepts such as interchain accounts will continue to increase adoption and provide application diversity in the Interchain ecosystem.   
  
Find a list of ecosystem efforts on IBC applications and light clients in the ibc-go repo's [readme (opens new window)↗](https://github.com/cosmos/ibc-go#ecosystem) or the [ibc-apps repo (opens new window)↗](https://github.com/cosmos/ibc-apps).

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#security) Security

Along with protocol extensibility, reliability, and security without the need for trusted third parties, the permissionless nature of IBC as a generalized interoperability standard is one of the most valuable discerning features of IBC in comparison to standard bridge protocols. However, as it is permissionless to create IBC clients, connections, and channels, or to relay packets between chains, you may have wondered: *What about the security implications?*



The design of IBC security is centered around **two main principles**:

* Trust in (the consensus of) the chains you connect with.
* The implementation of fault isolation mechanisms, in order to limit any damage done should these chains be subject to malicious behavior.

The security considerations which IBC implements are worth exploring:

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-light-clients) IBC light clients

Unlike many trusted bridge solutions, IBC does not depend on an intermediary to verify the validity of cross-chain transactions. As explained previously, the verification of packet commitment proofs are provided by the light client. The light client is able to track and efficiently verify the relevant state of the counterparty blockchain, to check commitment proofs for the sending and receiving of packets on the source and destination chains respectively.



In IBC, blockchains do not directly pass messages to each other over the network. To communicate, blockchains commit the state to a precisely defined path reserved for a specific message type and a specific counterparty. Relayers monitor for updates on these paths and relay messages by submitting the data stored under the path along with proof of that data to the counterparty chain.



The paths that all IBC implementations must support for committing IBC messages are defined in the [ICS-24 host state machine requirements (opens new window)↗](https://github.com/cosmos/ibc/tree/master/spec/core/ics-024-host-requirements).

This is important because it ensures the IBC protocol remains secure even in Byzantine environments where relayers could act in a malicious or faulty manner. You do not need to trust the relayers; instead, you trust the proof verification provided by the light client. **In the worst case situation where all relayers are acting in a Byzantine fashion, the packets sent would get rejected because they do not have the correct proof. This would affect only the liveness, not the security, of the particular part of the Interchain network where the relayers are malicious.**

Note that this effect would only affect the network if all relayers were Byzantine. As relaying is permissionless, a simple fix would be to spin up a non-malicious relayer to relay packets with the correct proof. This fits the *security over liveness* philosophy that IBC and the wider Interchain ecosystem adopts.



In the following video Colin Axnér of Interchain, a core contributor to ibc-go in the Cosmos SDK, looks at the IBC packet lifecycle and the security properties of a light client.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#trust-the-chains-not-the-bridge) Trust the chains, not the bridge

IBC clients and transactions assume the trust model of the chains they are connected to. In order to represent this accurately in assets which have been passed through the Interchain, the information of the path that an asset has traveled (the security guarantee of the asset) is stored in the denomination of the asset itself. In the case that the end user or an application does not trust a specific origin chain, they would be able to verify that their asset has not come from the untrusted chain simply by looking at the denomination of the asset, rather than referring to the validator set of a bridge or some other trusted third party verifier.



All tokens transferred over a particular channel will be assigned the same denomination as other tokens flowing over the channel, but a different one than the same assets between the same chains would have if they were sent across a different channel. The IBC denom looks like ibc/<hash of the channel-id & port-id>.   
  
You can find more detailed information in the tutorial on [IBC denoms](https://ida.interchain.io/tutorials/6-ibc-dev/).

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#submit-misbehavior) Submit misbehavior

One type of Byzantine behavior that can happen on an IBC-enabled chain is when validators double-sign a block - meaning they sign two different blocks at the same height. This scenario is called a fork. Unlike in Proof-of-Work blockchains (like Bitcoin or Ethereum) where forks are to be occasionally expected, in CometBFT the fast finality of chains is desired (and is a prerequisite for IBC) so forks should not occur.

Through the principle of [fork accountability (opens new window)↗](https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md#fork-accountability) the processes that caused the consensus to fail can be identified and punished according to the rules of the protocol. However, if this were to happen on a foreign chain, it would start a race for the light client of this compromised chain on counterparty chains to become aware of the fork.

The IBC protocol provides the functionality to submit a proof of misbehavior, which could be provided by the relayers, upon which the light client is frozen to avoid consequences as a result of the fork. The funds could later be recovered by unfreezing the light client via a governance proposal when the attack has been neutralized. The *submit misbehavior* functionality thus enables relayers to enhance the security of IBC, even though the relayers themselves are intrinsically untrusted.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#dynamic-capabilities) Dynamic capabilities

IBC is intended to work in execution environments where modules do not necessarily trust each other. Thus, IBC must authenticate module actions on ports and channels so that only modules with the appropriate permissions can use them. This module authentication is accomplished using a dynamic capability store. Upon binding to a port or creating a channel for a module, IBC returns a dynamic capability that the module must claim in order to use that port or channel. The dynamic capability module prevents other modules from using that port or channel since they do not own the appropriate capability.

It is worth mentioning that on top of the particular security considerations IBC takes, the security considerations of the Cosmos SDK and the application-specific chain model of the Cosmos white paper still hold. With reference to previous sections, remember that while iteration on the modules may be slower than iteration on contracts, application-specific chains are exposed to significantly fewer attack vectors than smart contract setups deployed on-chain. The chains would have to purposely adopt a malicious module by governance.

[#Copy link](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#development-roadmap) Development roadmap

As previously mentioned, even though IBC originated from the Interchain Stack it allows for chains not built with the Cosmos SDK to adopt IBC, or even those with a different consensus than CometBFT altogether. However, depending on which chain you want to implement IBC for or build IBC applications on top of, it may require prior development to ensure that all the different components needed for IBC to work are available for the consensus type and blockchain framework of your choice.



Generally speaking, you will need the following:

1. An implementation of the IBC transport layer.
2. A light client implementation on your chain, to track the counterparty chain you want to connect to.
3. A light client implementation for your consensus type, to be encorporated on the counterparty chain you want to connect to.

Click the expansion panel below to have a more detailed look at the different options.

****

**A roadmap towards an IBC-enabled chain**

The following decision tree helps visualize the roadmap towards an IBC-enabled chain. **For simplicity, it assumes the intent to connect to a Cosmos SDK chain.** Do you have access to an existing chain?

* **No.** You will have to build a chain:
  + Cosmos SDK chain: see the [previous chapters](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/).
  + Another chain.
    - Is there a CometBFT light client implementation available for your chain?
      * Yes. Continue.
      * No. Build a custom CometBFT light client implementation.
    - Is there a light client implementation for your chain’s consensus available in the SDK's IBC module?
      * Yes. Continue.
      * No. Build a custom light client for your consensus to be used on a Cosmos SDK chain (with IBC module).
    - Implement IBC core:
      * Source code.
      * Smart contract based.
* **Yes.** Is it a Cosmos SDK chain?
  + Yes. Move on to application development.
  + No.
    - Does your chain support a CometBFT light client?
      * Yes. Continue.
      * No. Source a CometBFT light client implementation for your chain.
    - Is there a Cosmos SDK light client implementation for your chain’s consensus?
      * Yes. Continue.
      * No. Build a custom light client implementation of your chain's consensus + governance proposal to implement it on the IBC clients of the Cosmos SDK chains you want to connect to.
    - Does your chain have a core IBC module available (connection, channels, ports)? Source code or smart contract based?
      * Yes. Move on to application development.
      * No. Build IBC Core implementation (source code or smart contract).

The most straightforward way to use IBC is to build a chain with the Cosmos SDK, which already includes the IBC module - as you can see when examining the [ibc-go repository (opens new window)↗](https://github.com/cosmos/ibc-go). The IBC module supports an out-of-the-box CometBFT light client. Other implementations are possible but may require further development of the necessary components; go to the [IBC website (opens new window)↗](https://ibcprotocol.org/implementations) to see which implementations are available in production or are being developed.



If you're interested in a detailed introduction to Inter-Blockchain Communication, check out the following video Thomas Dekeyser, Developer Relations Engineer for IBC.

synopsis

To summarize, this section has explored:

* How the Inter-Blockchain Communication Protocol (IBC) solves the problem of cross-chain communication by handling the authentication and transport of data between two blockchains through a minimal set of functions specified in the Interchain Standards (ICS).
* How the IBC functions permissionlessly and can be used with a wide range of blockchains or state machines regardless of their network topologies or consensus algorithms, with IBC security reduced to that of the participating chains.
* How IBC is the foundation of interoperability in the Interchain Ecosystem, with relayers such as light clients verifying the validity of cross-chain transactions, while also offering solutions to the issue of communicating with non-Cosmos blockchains including those which do not meet the criteria of Proof-of-Stake (PoS)finality.
* How a Cosmos blockchain's transport layer (TAO) provides the infrastructure for establishing secure connections to other chains, while the application layer built on top defines how authenticated data packets should be packaged and interpreted.
* How relayer algorithms provide the essential off-chain processes that share data between chains running the IBC protocol by scanning the state of each chain, constructing appropriate datagrams, and executing them on the opposite chain as permitted by the protocol.
* How the Byzantine behavior that leads to forking is prevented within the fast finality Interchain Ecosystem, and how the IBC protocol can achieve fork prevention outside the ecosystem by submitting proof of validator misbehavior and freezing affected light clients until an issue or attack has been neutralized.

previous

[](https://ida.interchain.io/ida-course/LPs/week-3/)

**[Introduction to IBC and CosmJS](https://ida.interchain.io/ida-course/LPs/week-3/)**

up next

**[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/2-connections.html)**

[[](https://ida.interchain.io/academy/3-ibc/2-connections.html)](https://ida.interchain.io/academy/3-ibc/2-connections.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[Internet of blockchains](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#internet-of-blockchains)

[High-level overview of IBC](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#high-level-overview-of-ibc)

[IBC/TAO - transport layer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-tao-transport-layer)

[IBC/APP - application layer](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-app-application-layer)

[Security](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#security)

[IBC light clients](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#ibc-light-clients)

[Trust the chains, not the bridge](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#trust-the-chains-not-the-bridge)

[Submit misbehavior](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#submit-misbehavior)

[Dynamic capabilities](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#dynamic-capabilities)

[Development roadmap](https://ida.interchain.io/academy/3-ibc/1-what-is-ibc.html#development-roadmap)

#### **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)