

# Overview

## What is Neutron and why is it needed?

Neutron is a blockchain network that brings Smart Contracts into the Cosmos-family blockchains using [CosmWasm](#) . Neutron works with networks using the IBC protocol. Neutron security (block validation) is provided by the [Cosmos Hub network](#) using [Interchain Security](#) .

## Cosmos SDK

Neutron is built using the Cosmos SDK framework — a generalized framework that simplifies the process of building secure blockchain applications on top of Tendermint BFT. It is based on two major principles: Modularity & capabilities-based security.

## Interchain Security

[Interchain Security](#) is a technology which allows Neutron to be secured by the validator set of the Cosmos Hub and do not have another one separate validator set particularly for the Neutron blockchain.

## Interchain Queries

Neutron is bringing customizable Interchain Queries (ICQs) to CosmWasm smart-contracts. ICQs are an essential building block enabling devs to securely retrieve data from remote zones.

## Interchain Transactions

Neutron is bringing interchain accounts (ICAs) to CosmWasm smart-contracts. ICAs allow modules and smart contracts to execute and track transactions on foreign zones, without deploying additional code. With ICQs, they're the Interchain's secret DeFi sauce.

## CosmWasm

CosmWasm is a smart contracting platform built for the Cosmos ecosystem.

## Become an interchain builder

The concepts described above constitute the list of core technologies used in the majority of projects on Neutron. In other words, these technologies are the main attraction for projects to opt for Neutron. Though it might take some time to grasp and apply them consistently, doing so is vital for evolving into a proficient interchain builder. If you're looking to enhance your understanding, below are a few resources that can assist you in that endeavor.

- [Neutron tutorials section](#)
- . This section contains information covering CosmWasm smart contracts, handy tools to help you develop and test them, and examples of using Neutron's ICQs and ICAs in smart contracts.
- [Tour of Rust](#)
- . Usually CosmWasm smart contracts are developed using the [cosmwasm](#)
- framework written in Rust. It only requires superficial knowledge of Rust, and a good way to learn it could be this short guide.
- [Rust book](#)
- . In case you want to acquire more in-depth Rust skills, this book might help you. It contains comprehensive articles about dozens of different Rust concepts.
- [CosmWasm Academy](#)
- . This educational project focuses on smart contracts, covering areas such as developing, testing, deploying, and how smart contracts interact with each other and a UI. It's designed for both smart contracts developers and frontend developers to find valuable insights.
- [Area-52](#)
- . A casual learning initiative exploring CosmWasm dApps and NFTs, enriched with numerous external links that explain various auxiliary odds and ends.
- [Interchain Builders Program](#)
- . If you're a team building or planning to develop your own dApp in the Cosmos ecosystem, or if you already have a dApp outside the ecosystem and are seeking guidance, acceleration, and integration into the Cosmos environment — this is likely the ideal place for you to consider applying for assistance.
- [IBC-Go docs](#)
- . An extensive resource explaining the Golang implementation of the Inter-Blockchain Communication Protocol.

## F.A.Q.

Please refer to the [F.A.Q.](#) page! [Next F.A.Q.](#)