

Hello dYdX community, and congratulations on launching the dYdX v4 public testnet! As mainnet approaches and dYdX gets closer to joining the Cosmos ecosystem, I thought it might be helpful to write up a high-level overview of Cosmos, some important terms, tools, and concepts, and some links to Cosmos tooling and infrastructure. My hope is that by reading this guide, the dYdX community will have a sufficient understanding of the ecosystem to make informed governance decisions surrounding things like Cosmos-related integrations, grants, and infrastructure.

This guide is in no way comprehensive, as there's simply too much to distill into one forum post, but I will try to link to more detailed sources (docs, github repos, articles) wherever possible for those looking to do a deeper dive. You can also feel free to reach out to me here on the forum or on [twitter](#) with any additional questions! Cosmos has a welcoming and helpful community, and you'll find many people eager to answer your questions.

Table of Contents

1. Helpful Terms and Definitions
2. Differences Between Cosmos and Ethereum
3. IBC
4. Interchain Security
5. Native USDC
6. Bridges to Cosmos
7. Cosmos Wallets
8. Cosmos Tooling and Infrastructure
9. Helpful Links

Helpful Terms and Definitions

Cosmos - This term is used to refer to the entire ecosystem of interconnected sovereign blockchains, currently numbering close to 100 chains.

Cosmos SDK - This refers to the Software Development Kit that underpins most Cosmos blockchains (including dYdX v4). Using the Cosmos SDK, developers can build their own chains with a number of pre-existing modules that obviate the need to spin up an entire chain from scratch. Custom modules can also be built on top of this stack. When someone says they are "building on Cosmos," they are referring to using the Cosmos SDK.

Cosmos Hub - This is the native blockchain of the ATOM token.

IBC - This refers to the Inter-Blockchain Communication Protocol, the Cosmos ecosystem's native "bridge" / interoperability protocol that enables the transfer of assets and arbitrary messages from one chain to any other IBC-enabled chain.

Interchain Accounts (ICA) - A feature of IBC which allows for one chain to interact with another module on another chain.

Interchain Queries (ICQ) - A feature of IBC which allows for one chain to query a module or smart contract on another chain.

Differences Between Cosmos and Ethereum

Note: The intention of this section is not to comment on whether one ecosystem or another is "better" than the other. That's not an argument worth having. I ascribe to [Antonio's view](#) that blockchain products will become ecosystem agnostic, and all cross-ecosystem UX will eventually be abstracted away. In this end state, all ecosystems with good products will thrive.

Much of the confusion surrounding Cosmos lies in the sovereign nature of each of the chains that make up the ecosystem. If you're mostly familiar with the Ethereum ecosystem or other primarily monolithic systems, it takes a bit of a perspective shift to understand how Cosmos works.

Many people look at the Cosmos Hub and its ATOM token as the Ethereum of Cosmos, with every Cosmos chain and application being built on top of or secured by the Cosmos Hub. This is not correct. Unlike Ethereum's ecosystem, in which all applications, L2s, or rollups are either built directly on or somehow settle to the Ethereum L1, blockchains in the Cosmos ecosystem are fully sovereign and have their own native validator set and settlement layer. Most Cosmos chains run a delegated proof of stake network in which token holders delegate their tokens to one or more validators, who form the consensus layer. In exchange for delegating to validators, delegators are rewarded with tokens (usually emissions or fees in the chain's native token collected by the network)

If the Cosmos Hub were to disappear today, the other chains in the Cosmos ecosystem would continue to operate as normal (the exception being consumer chains which we'll get into later).

The benefits of this model are numerous. By building on Cosmos, appchain developers can do things like customize and

control block production and mempools to stave off MEV, enable transaction privacy, and even build core functionality directly into the consensus layer in unique and interesting ways (subject to validator approval). A few examples:

- [Osmosis](#), an appchain decentralized exchange in the Cosmos ecosystem, supports an [on-chain module](#) that back runs swap transactions occurring in its liquidity pools, capturing MEV opportunities before value-extractive MEV bots can do so. The MEV revenue is then returned to the protocol (either as revenue to holders of the chain's native OSMO token, burning, or distribution to the protocol's treasury). This would not be possible without the ability to allow for the chain to append the back-running logic to the end of every swap transaction, giving that logic absolute priority over MEV bots that would seek to frontrun the module.
- [Stargaze](#), an NFT chain, is working on a feature that will allow native staking of NFTs to secure the chain. It does this by allowing for a staked NFT to represent a specific number of tokens tied to the NFT's floor price, and then minting the chain's native STARS token and delegating it to a validator. This can't be done on other chains without the ability to access that chain's native asset minting and staking functionality.
- [Kujira](#), a defi focused layer-1, requires that its validators run an oracle price feeder on their nodes to help power its natively on-chain order book, Fin.

Similarly, dYdX v4 intends to leverage these unique benefits of the appchain model to implement its own MEV features and validator order-matching engine, allowing it to handle additional TPS while maintaining the ease of UX that its users have come to expect.

Given that most of the applications and chains in the Cosmos ecosystem operate independently of each other, they need a way to communicate seamlessly to enable the free-flow of assets and messages back and forth to each other. This is especially true of DeFi appchains. While this concept may seem simple to many, I'm going to explain why just in case it's helpful.

Let's take Osmosis, a decentralized exchange appchain in the Cosmos ecosystem, as an example. Since Osmosis isn't built on top of an L1, but is rather its own blockchain, the only asset that is native to Osmosis is the OSMO token (which is responsible for payment of fees and governance). Standing alone, Osmosis wouldn't have any tradeable assets to pair with OSMO in its liquidity pools. This isn't an issue with d/apps built on Ethereum or an L2, where any asset built on the L1/L2 can be added to, and tradeable on, DeFi platforms built on the same L1/L2.

From this, it's clear that some kind of bridge / messaging layer is needed to bring assets from other chains to Osmosis to allow for spot trading. With most bridge solutions, you need to enable an independent integration for each chain that connects to Osmosis, which is a highly time-consuming and resource-intensive process, especially in an ecosystem like Cosmos in which each chain is often connected with anywhere from 5 to 80 different chains. Additionally, a lack of a uniform bridging standard can lead to fragmentation of liquidity and lack of asset fungibility between Cosmos chains.

IBC

IBC, or the Inter-Blockchain Communication protocol, solves for both of the above problems. IBC is an arbitrary message passing protocol that acts as the canonical bridging standard for all Cosmos ecosystem chains. IBC uses light-clients to read the state of a chain's connected counterparty chains. This means that the only way that a fraudulent message will be accepted from a counterparty chain is if the counterparty chain's own validator set approves and signs the transaction as valid. Effectively, this introduces a mechanism for cross-chain message security that comes as close as possible to the level of security afforded by blockchain consensus itself.

Rather than working to build a custom, bespoke connection between a chain and each counterparty chain that it seeks to connect with, any Cosmos chain with an activated IBC module can connect permissionlessly to any other chain with an IBC module by creating an IBC client between them as well as a channel to relay message packets between the two chains.

Using this uniform standard, Cosmos SDK chains can permissionlessly connect with each other and facilitate the free flow of assets between blockchains in a matter of minutes in a maximally secure environment.

IBC contains a number of features that make connected chains highly composable with each other in ways not currently offered by many existing bridge solutions. Interchain Accounts allow one connected chain to send a message that interacts with an account or smart contract on another chain.

This unlocks a whole host of interesting usecases. For example, this feature could allow you to send assets from chain A to chain B, interact with a swap module or contract on chain B, send those assets from chain B to chain C, and stake them on chain C. These transactions are all batched into an interchain account transaction that you only have to execute once. This allows for the creation of a cross-chain user experience that comes close to being as simple and intuitive as a monolithic chain experience.

Similarly, Interchain Queries (ICQ), allows a user, module, or smart contract to query contracts or modules on another chain. This is extremely helpful in determining things like downtime on another chain, spot or oracle pricing of DeFi assets for things like cross-chain liquidations, and more!

However, this tight composability comes at a significant cost. While IBC light clients are easily deployed between chains using the Cosmos SDK, they are notoriously difficult to deploy outside of Cosmos. Light client state verification for IBC is

particularly difficult to deploy on Ethereum due to the cost involved (estimated at millions per year).

There are multiple teams working on ways to bring IBC to other chains. [Composable Finance](#) was recently the first project to succeed in doing so, bringing IBC natively to [Picasso](#), a Polkadot Parachain. [Landslide Network](#), which will be launching their mainnet soon, will do the same with Avalanche subnets. [Polymer DAO](#) is also working to bring IBC to Ethereum and other chains. Until these teams can successfully scale IBC, however, Cosmos uses a number of bridges and messaging / interoperability protocols to enable asset bridging and cross-chain messaging to external chains.

Interchain Security

A common complaint about Cosmos is that because each chain has its own sovereign validator set, these individual chains lack the economic security assumptions that Ethereum and its network of L2s and rollups have. Cosmos SDK chains run on a delegated proof of stake network, which requires two-thirds of validators to come to consensus in order to validate a block. This means that if a malicious attacker were to buy up and stake 34% of a chain's staked assets, that attacker could halt the chain. If the attacker were to buy up 67% of a chain's staked assets, the attacker could subvert consensus and produce invalid transactions.

While this has never happened in practice, it is a concern to many as Cosmos TVL grows. More TVL makes this type of attack more attractive, so having a mechanism that makes this type of economic attack unviable is important.

Interchain security is one such mechanism. A form of shared security, Interchain security refers to utilizing part or all of a Provider Chain's validator set to secure one or more smaller chains, referred to as Consumer Chains. In exchange for the Provider Chain's economic security, the consumer chain usually pays a percentage of its emissions or fees to the provider chain and its delegators.

Interchain security is live on one Provider Chain currently; the Cosmos Hub. There are two consumer chains currently consuming security from the Cosmos Hub's validator set, allowing them access to the Hub's \$2.2 billion in staked tokens ATOM. By doing so, these smaller chains cannot be subject to an economic attack like the one described above without the attacker first buying up two-thirds of all staked ATOM. More consumer chains are in the process of being onboarded to consume security from the Cosmos Hub.

Interchain security is in its infancy, and planned improvements are already in the works. V1 of interchain security is called Replicated Security, which involves using the entirety of the Cosmos Hub validator set to secure a consumer chain. This model isn't highly scalable, as it requires even lower-ranked validators to opt-in, a problem because interchain security increases costs of operation for consumer chain validators (additional node costs, extra slashing risk, etc).

The likely end state of Interchain security is called Mesh Security, a form of sharded shared security in which multiple chains provide and consume security from each other. This will create a security network within the Cosmos Ecosystem that is comparable to monolithic or rollup-centric security models. In fact, using mesh security, chains can even opt in to consuming security from Ethereum or other monolithic systems using solutions like Eigenlayer.

For a more in-depth discussion of Interchain security and how it will evolve over the next couple of years, I highly recommend the following two resources:

- Intro to Mesh Security by Osmosis Cofounder Sunny Aggarwal: [Mesh Security by Sunny Aggarwal at the Shared Security Summit 2023 - YouTube](#)
- Replicated vs Mesh Security by Informal Systems: [Informal Blog - Replicated vs. Mesh Security](#)

Native USDC

I'm adding a brief section to this post about Cosmos native USDC, as this is somewhat relevant to the dYdX v4 launch.

One major issue that has hindered the inflow of liquidity to the Cosmos ecosystem is the lack of a native stablecoin. Ethereum and many L2s have natively deployed smart contracts for the USDC and USDT tokens. Cosmos, on the other hand, has up until recently only had bridged versions of these assets, which adds an additional layer of risk (bridge risk) to holding these key assets in the Cosmos Ecosystem.

With the launch of [Noble](#), a generic asset issuance chain built using the Cosmos SDK, this problem is solved. Circle has announced its intention to [deploy native USDC on Noble](#), which will then be distributed throughout the Cosmos ecosystem via IBC. dYdX v4 traders will use this Cosmos-native USDC as collateral on the dYdX perpetuals markets. It is still unclear what this integration will look like, but more details are said to be forthcoming.

Bridges to Cosmos

Cosmos currently has three major bridges that connect various Cosmos chains to external ecosystems. There are other, smaller bridges in the ecosystem currently like Celer, Hyperlane, and Router Protocol, but in the interest of word count I'm going to stick with the major bridges. This section may be helpful in determining a useful bridge or bridges for facilitating asset transfers to dYdX chain, as is being discussed here: [Bridging to v4: Assessing Token Bridges](#)

Axelar

: [Axelar](#) is a Cosmos SDK chain that facilitates bridging from Ethereum and several other chains to the Cosmos ecosystem using Generic Message Passing (outside of Cosmos) and IBC (within Cosmos). Axelar currently has the most bridged liquidity to the Cosmos ecosystem, and can facilitate cross-chain swaps using its liquidity layer, [Squid Router](#).

Wormhole

: [Wormhole](#) is a generic message passing protocol that facilitates the transfer of assets and cross-chain messages from over 23 connected chains to the Cosmos ecosystem. It is currently connected to several Cosmos chains, with plans to connect more very soon. Wormhole just announced the launch of its own Cosmos SDK appchain called [Wormhole Gateway](#) to facilitate bridging of assets from any of its connected chains to Cosmos using IBC. Cross-chain swaps and other generic message passing usecases are also possible with Wormhole. Wormhole has the most liquidity and bridge volume out of any of the bridges that service the Cosmos ecosystem. Wormhole is the only bridge to have received an unconditional [approval from the uniswap bridge committee](#) for use in all cross-chain deployments of the uniswap protocol, which is a significant testament to the bridge's security (Axelar was also approved for use in all deployments, but subject to conditions).

Gravity Bridge

: [Gravity bridge](#) is another Cosmos SDK appchain that enables the free-flow of assets between Ethereum and Cosmos. It doesn't have the same number of supported chains as the other two bridges, but it does operate in a more decentralized and trustless manner, using no multisigs for core bridge functionality.

Cosmos Wallets

As with Bridges, Cosmos has many wallets that can be used to securely store your assets, but this post will only focus on the major wallets for the sake of word count. Notably, Mystic Labs is in the process of [building snaps](#) that will allow for the use of Metamask in Cosmos, but because this feature is still in its infancy it will not be discussed here either.

Keplr Wallet

: [Keplr](#) is the largest wallet in the Cosmos ecosystem, with nearly 1 million users. The wallet is a browser extension wallet, and offers a simple way to facilitate transfers, staking of assets, voting on governance, and more. It also has a simple to use [dashboard](#) where you can manage your assets across multiple chains and a mobile app where you can manage assets, NFTs, and LP positions on certain chains like Stargaze or Osmosis.

Cosmostation

: [Cosmostation](#) wallet offers a comprehensive staking and governance dashboard for Cosmos chains and assets. Unlike Keplr, it also offers limited in-wallet bridge and swap functionality, making it preferable for users who prefer to handle as many transactions as possible from within the wallet interface.

Leap Wallet

: [Leap](#) is the newest and smallest of the three wallets listed here, but have already proven themselves to be one of the more innovative. Leap offers a comprehensive in-wallet swap and bridge aggregator, information about airdrops and whether you qualify, and more, all in addition to the main features offered by Keplr and Cosmostation. Like those wallets, leap also has a mobile app

Cosmos Tooling and Infrastructure

This section briefly lists several key tools that newcomers to the Cosmos Ecosystem may find useful. I won't do a deep dive on these, but will include links in case you'd like to explore them further!

Block Explorers

- Mintscan: <https://www.mintscan.io/>
- BigDipper: [Big Dipper | Cosmos Hub Block Explorer](#)
- PingPub: <https://ping.pub/>

Aggregators

- TFM: [TFM](#)
- Rango: [Rango Exchange - Swap everything everywhere. Eth, Bsc, Terra, Poly](#)
- IBC.fun: <https://ibc.fun/>

Governance

- Commonwealth: <https://commonwealth.im/>

- SubtractFi: <https://proposals.subtract.fi/>

Fiat Onramps

- Kado: <https://www.kado.money/>
- Transak: <https://transak.com/>

Helpful Links

- Cosmos SDK Documentation: [Cosmos SDK Documentation | Cosmos SDK](#)
- IBC Docs: <https://ibc.cosmos.network/>
- IBC Github Repository: [GitHub - cosmos/ibc: Interchain Standards \(ICS\) for the Cosmos network & interchain ecosystem.](#)
- Interchain Security Docs: [Interchain Security Docs | Interchain Security](#)
- IBC ecosystem MAU and volume tracker: <https://mapofzones.com/>
- Cosmos Ecosystem Twitter: <https://twitter.com/cosmos>
- Cosmos Hub Twitter: <https://twitter.com/cosmoshub>
- Cosmos Website: <https://cosmos.network/>
- Cosmos Discord: [Cosmos Network](#)