

# What is a Bridge?

A bridge (AKA crosschain messaging protocol or interoperability network) is a system that relays information between blockchains.

While there are many bridges out there today, every bridge has the same core structure and components.

Layer Function Transport Read root or hash of data on origin chain and post to destination. Verification Ensure that the above data is correct. Execution Generate merkle roots or hashes on origin. Prove against the root and execute the target function on destination. Application Handle specific usecases such as token transfers, NFTs, governance, and more!

## Messaging Layers

### Transport

Transport is how a payload of data gets read from one domain and posted to another, making no assumptions about the correctness of the data.

This is typically done by one or many offchain actors that watch anoutbox of data on the origin chain, and then post the corresponding data to aninbox on a connected destination domain. To keep this process scalable, protocols will typically relay merkle roots rather than data in raw form.

### Verification

Verification is how crosschain communication is secured. After a payload is transported across chains, a bridge will verify the data prior to it becoming usable on the destination domain.

There are many different ways to verify messages across domains, each with their own tradeoffs on trust, cost, and latency. See Message Verification for a longer discussion about this step.

### Execution

Once a verified payload is available on the destination, some offchain infrastructure is needed to "push" that payload into a target function.

Bridge execution layers are the interface that developers will typically interact with when integrating with a bridge. Execution layers collect gas fees (in Connex, this is collected as additional gas paid on the origin chain) and use those fees to execute a transaction on the destination chain against the target function the developer intends to interact with.

Execution layers may additionally be responsible for generating merkle roots on each origin domain, and then generating a merkle proof against that root on a destination domain.

## Application Layers

In addition to the above messaging layers, bridges may implement one or many application layers that define how specific usecases are enabled across domains.

## Additional Resources

- [The Messaging Bridge Stack](#)
- 

[Previous Background](#) [Next Modular Bridges](#) Last updated 5 months ago On this page \* [Messaging Layers](#) \* [Transport](#) \* [Verification](#) \* [Execution](#) \* [Application Layers](#) \* [Additional Resources](#)

[Edit on GitHub](#)