---

title: Bridges description: An overview of bridging for developers lang: en

---

With the proliferation of L1 blockchains and L2scaling solutions, alongside an ever-growing number of decentralized applications going cross-chain, the need for communication and asset movement across chains has become an essential part of network infrastructure. Different types of bridges exist to help make this possible.

Need for bridges {#need-for-bridges}

Bridges exist to connect blockchain networks. They enable connectivity and interoperability between blockchains.

Blockchains exist in siloed environments, meaning there is no way for blockchains to trade and communicate with other blockchains naturally. As a result, while there could be significant activity and innovation within an ecosystem, it is limited by the lack of connectivity and interoperability with other ecosystems.

Bridges offer a way for isolated blockchain environments to connect with each other. They establish a transportation route between blockchains where tokens, messages, arbitrary data, and even smart contract calls can be transferred from one chain to another.

Benefits of bridges {#benefits-of-bridges}

Put simply, bridges unlock numerous use cases by allowing blockchain networks to exchange data and move assets between them.

Blockchains have unique strengths, weaknesses, and approaches to building applications (such as speed, throughput, costliness, etc.). Bridges help the development of the overall crypto ecosystem by enabling blockchains to leverage the innovations of each other.

For developers, bridges enable the following:

- the transfer of any data, information, and assets across chains.
- unlocking new features and use cases for protocols as bridges expand the design space for what protocols can offer. For example, a protocol for yield farming originally deployed on Ethereum Mainnet can offer liquidity pools across all EVM-compatible chains.
- the opportunity to leverage the strengths of different blockchains. For example, developers can benefit from the lower fees offered by the different L2 solutions by deploying their dapps across rollups, and sidechains and users can bridge across them.
- collaboration among developers from various blockchain ecosystems to build new products.
- attracting users and communities from various ecosystems to their dapps.

How do bridges work? {#how-do-bridges-work}

While there are many types of bridge designs, three ways to facilitate the cross-chain transfer of assets stand out:

- **Lock and mint –** Lock assets on the source chain and mint assets on the destination chain.
- **Burn and mint –** Burn assets on the source chain and mint assets on the destination chain.
- **Atomic swaps –** Swap assets on the source chain for assets on the destination chain with another party.

Bridge types {#bridge-types}

Bridges can usually be classified into one of the following buckets:

- **Native bridges –** These bridges are typically built to bootstrap liquidity on a particular blockchain, making it easier for users to move funds to the ecosystem. For example, the Arbitrum Bridge is built to make it convenient for users to bridge from Ethereum Mainnet to Arbitrum. Other such bridges include Polygon PoS Bridge, Optimism Gateway, etc.
- **Validator or oracle based bridges –** These bridges rely on an external validator set or oracles to validate cross-chain

transfers. Examples: Multichain and Across.
- **Generalized message passing bridges –** These bridges can transfer assets, along with messages and arbitrary data across chains. Examples: Nomad and LayerZero.
- **Liquidity networks –** These bridges primarily focus on transferring assets from one chain to another via atomic swaps. Generally, they don't support cross-chain message passing. Examples: Connext and Hop.

Trade-offs to consider {#trade-offs}

With bridges, there are no perfect solutions. Rather, there are only trade-offs made to fulfill a purpose. Developers and users can evaluate bridges based on the following factors:

- **Security –** Who verifies the system? Bridges secured by external validators are typically less secure than bridges that are locally or natively secured by the blockchain's validators.
- **Convenience –** How long does it take to complete a transaction, and how many transactions did a user need to sign? For a developer, how long does it take to integrate a bridge, and how complex is the process?
- **Connectivity –** What are the different destination chains a bridge can connect (i.e., rollups, sidechains, other layer 1 blockchains, etc.), and how hard is it to integrate a new blockchain?
- **Ability to pass more complex data –** Can a bridge enable the transfer of messages and more complex arbitrary data across chains, or does it only support cross-chain asset transfers?
- **Cost-effectiveness –** How much does it cost to transfer assets across chains via a bridge? Typically, bridges charge a fixed or variable fee depending on gas costs and the liquidity of specific routes. It is also critical to evaluate the cost-effectiveness of a bridge based on the capital required to ensure its security.

At a high level, bridges can be categorized as trusted and trustless.

- **Trusted –** Trusted bridges are externally verified. They use an external set of verifiers (Federations with multi-sig, multi-party computation systems, oracle network) to send data across chains. As a result, they can offer great connectivity and enable fully generalized message passing across chains. They also tend to perform well with speed and cost-effectiveness. This comes at the cost of security, as users have to rely on the security of the bridge.
- **Trustless –** These bridges rely on the blockchains they are connecting and their validators to transfer messages and tokens. They are 'trustless' because they do not add new trust assumptions (in addition to the blockchains). As a result, trustless bridges are considered to be more secure than trusted bridges.

To evaluate trustless bridges based on other factors, we must break them down into generalized message passing bridges and liquidity networks.

- **Generalized message passing bridges –** These bridges excel with security and the ability to transfer more complex data across chains. Typically, they are also good with cost-effectiveness. However, these strengths generally come at the cost of connectivity for light client bridges (ex: IBC) and speed drawbacks for optimistic bridges (ex: Nomad) that use fraud proofs.
- **Liquidity networks –** These bridges use atomic swaps for transferring assets and are locally verified systems (i.e., they use the underlying blockchains' validators to verify transactions). As a result, they excel with security and speed. Moreover, they are considered comparatively cost-effective and offer good connectivity. However, the major tradeoff is their inability to pass more complex data – as they don't support cross-chain message passing.

Risk with bridges {#risk-with-bridges}

Bridges account for the top threebiggest hacks in DeFi and are still in the early stages of development. Using any bridge carries the following risks:

- **Smart contract risk –** While many bridges have successfully passed audits, all it takes is one flaw in a smart contract for assets to be exposed to hacks (ex: Solana's Wormhole Bridge).
- **Systemic financial risks** – Many bridges use wrapped assets to mint canonical versions of the original asset on a new chain. This exposes the ecosystem to systemic risk, as we have seen wrapped versions of tokens exploited.
- **Counterparty risk –** Some bridges utilize a trusted design that requires users to rely on the assumption that validators will not collude to steal user funds. The need for users to trust these third-party actors exposes them to risks such as

rug pulls, censorship, and other malicious activities.

- **Open issues –** Given that bridges are in the nascent stages of development, there are many unanswered questions related to how bridges will perform in different market conditions, like times of network congestion and during unforeseen events such as network-level attacks or state rollbacks. This uncertainty poses certain risks, the degree of which is still unknown.

How can dapps use bridges? {#how-can-dapps-use-bridges}

Here are some practical applications that developers can consider about bridges and taking their dapp cross-chain:

Integrating bridges {#integrating-bridges}

For developers, there are many ways to add support for bridges:

1. **Building your own bridge –** Building a secure and reliable bridge is not easy, especially if you take a more trust-minimized route. Moreover, it requires years of experience and technical expertise related to scalability and interoperability studies. Additionally, it would require a hands-on team to maintain a bridge and attract sufficient liquidity to make it feasible.

2. **Showing users multiple bridge options –** Many dapps require users to have their native token to interact with them. To enable users to access their tokens, they offer different bridge options on their website. However, this method is a quick fix to the problem as it takes the user away from the dapp interface and still requires them to interact with other dapps and bridges. This is a cumbersome onboarding experience with the increased scope of making mistakes.

3. **Integrating a bridge –** This solution doesn't require the dapp to send users to the external bridge and DEX interfaces. It allows dapps to improve the user onboarding experience. However, this approach has its limitations:

4. Assessment and maintenance of bridges are hard and time-consuming.

5. Selecting one bridge creates a single point of failure and dependency.
6. The dapp is limited by the bridge's capabilities.

7. Bridges alone might not be enough. Dapps might need DEXs to offer more functionality such as cross-chain swaps.

8. **Integrating multiple bridges –** This solution solves many problems associated with integrating a single bridge. However, it also has limitations, as integrating multiple bridges is resource-consuming and creates technical and communication overheads for developers—the scarcest resource in crypto.

9. **Integrating a bridge aggregator –** Another option for dapps is integrating a bridge aggregation solution that gives them access to multiple bridges. Bridge aggregators inherit the strengths of all the bridges and thus are not limited by any single bridge's capabilities. Notably, the bridge aggregators typically maintain the bridge integrations, which saves the dapp from the hassle of staying on top of the technical and operational aspects of a bridge integration.

That being said, bridge aggregators also have their limitations. For instance, while they can offer more bridge options, many more bridges are typically available in the market other than those offered on the aggregator's platform. Moreover, just like bridges, bridge aggregators are also exposed to smart contract and technology risks (more smart contracts = more risks).

If a dapp goes down the route of integrating a bridge or an aggregator, there are different options based on how deep the integration is meant to be. For instance, if it's only a front-end integration to improve the user onboarding experience, a dapp would integrate the widget. However, if the integration is to explore deeper cross-chain strategies like staking, yield farming, etc., the dapp integrates the SDK or API.

Deploying a dapp on multiple chains {#deploying-a-dapp-on-multiple-chains}

To deploy a dapp on multiple chains, developers can use development platforms like Alchemy, Hardhat, Truffle, Moralis, etc. Typically, these platforms come with composable plugins that can enable dapps to go cross-chain. For instance, developers can use a deterministic deployment proxy offered by the hardhat-deploy plugin.

Examples:

- How to build cross-chain dapps
- Building a Cross-Chain NFT Marketplace
- Moralis: Building cross-chain NFT dapps

Monitoring contract activity across chains {#monitoring-contract-activity-across-chains}

To monitor contract activity across chains, developers can use subgraphs and developer platforms like Tenderly to observe smart contracts in real-time. Such platforms also have tools that offer greater data monitoring functionality for cross-chain activities, such as checking for events emitted by contracts, etc.

Tools

- The Graph
- Tenderly

Further reading {#further-reading}

- Blockchain Bridges – ethereum.org
- Blockchain Bridges: Building Networks of Cryptonetworks Sep 8, 2021 – Dmitriy Berenzon
- The Interoperability Trilemma Oct 1, 2021 – Arjun Bhuptani
- Clusters: How Trusted & Trust-Minimized Bridges Shape the Multi-Chain LandscapeOct 4, 2021 – Mustafa Al-Bassam
- LI.FI: With Bridges, Trust is a Spectrum Apr 28, 2022 – Arjun Chand

Additionally, here are some insightful presentations byJames Prestwich that can help develop a deeper understanding of bridges:

- Building Bridges, Not Walled Gardens
- Breaking Down Bridges
- Why are the Bridges Burning