Documentation Index

Fetch the complete documentation index at: /llms.txt

Use this file to discover all available pages before exploring further.

Welcome

- What is deBridge?
- Quickstart
- deBridge Bundles

Use Cases

- Swaps
- User Onboarding
- Chain Abstraction
- Hooks & Workflows
- Cross-Chain Messaging

Products

- deBridge Liquidity Network

CRYP2NOVA

Archived copy · cryp2nova.com

- Infrastructure-as-a-Service
- dePort
- deBridge Messaging Protocol

Architecture

- Execution Model
- Fees & Pricing
- Supported Chains

Security

- Security Overview

Guides

- Every Use Case Quickstart

Resources

- FAQ
- Supported Tokens
- Status & Limits

Monetization

- Affiliate Fees
- Custom Linking
- deBridge Points
- Referral Codes

Legal

- SDK & API License Agreement

close

What is deBridge?

Swap, bridge, and program actions across chains with non-custodial, 0-TVL execution

deBridge is a **non-custodial execution layer** for **cross-chain** and **same-chain** actions. It uses a 0-TVL architecture where competitive solvers provide liquidity on-demand, with no shared liquidity pools. deBridge handles liquidity, cross-chain messaging, and execution. For same-chain swaps, it compares

Archived copy · cryp2nova.com

outcomes across aggregators to deliver the best rate. For cross-chain trades, it coordinates settlement through the deBridge Liquidity Network (DLN). deBridge has processed **$20B+ in cross-chain volume** across 25+ blockchains and is non-custodial by design. Bundles, a new execution primitive for intent-based, chain-agnostic interactions, is currently in private rollout.

What Can You Build?

deBridge powers applications that need **cross-chain execution**. Typical use cases include:

- Same-Chain and Cross-Chain Swaps inside your app
  Let users trade any token across chains without leaving your product.
- One-click onboarding and transactions
  Onboard users from any chain into your product.
- Automated cross-chain workflows
  Trigger DeFi actions across chains with hooks.
- Chain abstraction (coming soon)
  Build chain-abstracted apps with Bundles.

If your product spans multiple chains — or hides them entirely — **deBridge is the execution layer**.

How It Works

deBridge coordinates cross-chain and same-chain execution end-to-end. For same-chain swaps, deBridge compares outcomes across aggregators to deliver the best rate. For cross-chain trades, orders are fulfilled through the deBridge Liquidity Network.

1

Define the outcome

A user or application specifies what they want to happen.
Example: “Swap SOL on Solana to USDC on Base.”

2

deBridge coordinates execution

deBridge routes and executes the trade using:

- **Unified liquidity** from a competitive network
- **Secure cross-chain messaging** via the deBridge Messaging Protocol
- **MEV-protected settlement** to ensure predictable outcomes
- Optional hooks or contract calls run automatically

3

Receive the result

The outcome is delivered to the destination chain wallet. Quoted rates are guaranteed.

Execution & Fund Safety

When a user creates an order, the input tokens are deposited into the `DlnSource` smart contract on the source chain. The order is then fulfilled on the destination chain, delivering the requested tokens to the recipient. Smart contracts act as pipes, not pools. Funds pass through briefly on a per-order basis.

- **No shared liquidity pools**
  Each order is independent. There is no pooled TVL for attackers to target.
- **Guaranteed recovery**
  If an order is not fulfilled, users can always cancel and reclaim their tokens in full, including all fees. Auto-cancellations can handle this automatically.
- **No stuck funds**
  User funds are never held on a third-party address. The `DlnSource` contract ensures funds can always be either claimed after fulfillment or returned to

Archived copy · cryp2nova.com

the user via cancellation.

This is enforced by deBridge's 0-TVL architecture.

Integration Options

↗

Widget

Low-code universal swaps
↗

deBridge Liquidity Network API

Programmatic routing and outcomes
↗

Hooks & External Calls

Advanced conditional execution

Getting Started

↗

Concepts

New to deBridge? Start here to understand the architecture.
↗

CRYP2NOVA

Archived copy · cryp2nova.com

Quickstarts

Want to ship fast? Jump into integration guides.
↗

Use Cases

Have a specific goal? Find your use case.
↗

Hooks & Execution

Building advanced flows? Learn about hooks.

FAQ

▶

Is deBridge just a bridge?

▶

Is deBridge non-custodial?

▶

CRYP2NOVA

Archived copy · cryp2nova.com

Does deBridge support same-chain execution?

▶

What happens if a deBridge transaction fails?

▶

How does the execution flow work?

▶

Can I deploy synthetic assets across chains?

Ask a question...

⌘I



discord x github telegram

linkedin medium

Archived copy · cryp2nova.com

Powered by



This documentation is built and hosted on Mintlify, a developer documentation platform

Archived copy · cryp2nova.com