# LayerZero Messages

Normally with blockchains, each network operates in isolation.

But what if you could have an interaction on one blockchain (say,Ethereum ) automatically trigger a reaction on another (likeBinance Smart Chain ), all without relying on a central authority to relay that trigger?

This idea is at the core of LayerZero's omnichain messaging, reshaping how blockchains interact.

## But First, A Primer on Smart Contracts

Before we dive deeper, let’s take a step back and look at how smart contracts work.

At its core, a token contract on Ethereum (e.g.,ERC20 ) is a smart contract that keeps track of balances.

It uses the blockchain as a digital ledger, recording who owns what. When you“transfer” tokens, you’re not moving physical objects.

Instead, the contract updates the ledger to reflect the change in ownership:

```
// @dev the _transfer function from the base ERC20 token standard function

_transfer ( address

from ,

address to ,

uint256 amount )

internal virtual { require ( from

!=

address ( 0 ) ,

"ERC20: transfer from the zero address" ) ; require ( to !=

address ( 0 ) ,

"ERC20: transfer to the zero address" ) ;

_beforeTokenTransfer ( from , to , amount ) ;

uint256 fromBalance = _balances [ from ] ; require ( fromBalance

      = amount ,

"ERC20: transfer amount exceeds balance" ) ; unchecked { // @dev the sender's account is debited the amount _balances [
from ]

= fromBalance - amount ; // @dev the receiver's account is credited the amount _balances [ to ]

+= amount ; }

emit

Transfer ( from , to , amount ) ;

_afterTokenTransfer ( from , to , amount ) ; }
```

This ledger is the heart of every smart contract, a fundamental aspect of blockchain technology.

## A Ledger That Lives Everywhere

LayerZero enables your on-chain application to use any or every blockchain as that ledger.

For example, theOmnichain Fungible Token (OFT) Standard allows developers to extend the normal ERC20 token to record balances on any supported blockchain ledger.

This works by deploying an OFT contract on every blockchain you want to interact with, enabling you todebit a token from an address on one chain...

// @dev the _debit function from the base OFT token standard // @notice called on the source chain internally by the msg.sender function

_debit ( uint256 _amountLD , uint256 _minAmountLD , uint32 _dstEid )

internal virtual override returns

( uint256 amountSentLD ,

uint256 amountReceivedLD )

{ ( amountSentLD , amountReceivedLD )

=

_debitView ( _amountLD , _minAmountLD , _dstEid ) ;

// @dev In NON-default OFT, amountSentLD could be 100, with a 10% fee, the amountReceivedLD amount is 90, // therefore amountSentLD CAN differ from amountReceivedLD.

// @dev Default OFT burns on src. _burn ( msg . sender , amountSentLD ) ; } ...andcredit it to an address on another chain.

// @dev the _credit function from the base OFT token standard // @notice called on the destination chain by the Executor function

_credit ( address _to , uint256 _amountLD , uint32

/*_srcEid*/ )

internal virtual override returns

( uint256 amountReceivedLD )

{ // @dev Default OFT mints on dst. _mint ( _to , _amountLD ) ; // @dev In the case of NON-default OFT, the _amountLD MIGHT not be == amountReceivedLD. return _amountLD ; } This cross-chain interaction opens up a universe of possibilities for decentralized applications.

## Omnichain Contract Standards

Because every blockchain is just a digital ledger, and there are no rules about what smart contracts on that ledger have to do, developers can create variousomnichain standards for determining how contracts on different ledgers can interoperate with one another.

LayerZero offers two specialized contract standards: theOApp andOFT .

- OApp
- : a generic message passing interface for moving arbitrary data across blockchains for custom usage.
- OFT
- : an omnichain ERC20 built for sending and receiving tokens across different blockchains.

Read theGetting Started guide in theDevelopers section to deploy your first OApp smart contract.Edit this page

Previous Why LayerZero Next Deployed Endpoints