

## Problem

There are multiple cross-chain messaging protocols in production at this point. Some notable ones are layerzero, wormhole, and Hyperlane (previously abacus).

These cross-chain messaging protocols' operations vary in many aspects, but their primary objective is to transfer states from one chain to another.

Irrespective of their operation, these protocols have implemented their interface for sending and receiving messages.

Standardization of such interfaces would help achieve greater scalability and interoperability between multiple chains.

For eg., in Hyperlane for sending a message to a different chain, the `dispatch()` function is used.

```
function dispatch( uint32 _destinationDomain, bytes32 _recipientAddress, bytes calldata _messageBody ) external;
```

But in the case of Layerzero, for similar functionality, a different function name is used.

```
function send( uint16 _dstChainId, bytes calldata _remoteAndLocalAddresses, bytes calldata _payload, address payable _refundAddress, address _zroPaymentAddress, bytes calldata _adapterParams ) external payable;
```

The same goes with wormhole.

However both these functions intend to do the same process, the smart contract interface level differences make it difficult for integrating multiple cross-chain messaging protocols into one single application.

Standardization of such processes would help in better scalability and operations. Would love to discuss and hear more about this hypothesis.