Overview

Current messaging is not high UX due to the time and gas fee costs of multiple transactions occurring. In addition, messaging is designed for one-way communication and is not optimized for two-way communication, such as data acquisition from ChainA on a smart contract on ChainB.

To solve the above problem, we propose a new interoperability solution that allows data from other chains to be retrieved on a smart contract without a transaction.

Background

There are three main issues with current messaging.

One is speed. Because transactions occur, the communication itself takes time.

The second is high gas costs. Since multiple transactions are executed during messaging, the gas cost becomes high.

The third is the limited support of the chain: with cosmos, communication is limited in scope within the IBC, and with Polkadot, communication is limited within parachains

. Therefore, interoperability is only possible within a specific ecosystem.

There is another problem with interoperability. It is designed in such a way that communication is limited to one way.

For example, data may be sent from Ethereum to BSC, but it is not designed to acquire data from Ethereum to BSC and use that data again in Ethereum.

For example, using LayerZero to execute the previous example would look like this

[

1-5

1920×1080 116 KB

](https://ethresear.ch/uploads/default/original/2X/6/6ecbed9373dcbd7a0029bbc00d52e7962fe6767f.png)

Even simply using data in a src chain can result in multiple transactions, which is very time and gas intensive.

Therefore, we believe that the ideal way to improve the UX of interoperability is to reduce the number of transactions and ultimately not to generate any transactions.

Technical Issues

The change in state is important for sending data externally from smart contracts in current messaging and for validating data in messaging.

When information is transmitted to the outside world, it is necessary to issue an event and have it detected by an external bot. Also, to validate data in messaging, it is necessary to use a state root that contains the data to be messaged.

In other words, since a change in state is an essential element of information transfer from the smart contract to the off-chain, it would be necessary to send a transaction in order to perform messaging.

However, we believe that sending transactions would greatly reduce the UX as a communication protocol.

Solution

Therefore, we propose a new interoperability solution specialized in data acquisition.

This solution will enable faster, cheaper, and easier interoperability with other chains.

Specifically, it will allow you to acquire data from other chains on a smart contract without a transaction.

Here we present two solutions that we are currently considering.

1. Original chain with message verification system built on-chain

[

on-chain

1920×1080 121 KB

](https://ethresear.ch/uploads/default/original/2X/2/2b21987afbc535a29681d24f5eabac30a4d97eac.png)

Build a messaging validation system within your own chain.

This allows you to verify messages within your own chain where they would normally be verified off-chain, eliminating the need to bother emitting an event that conveys information off-chain, and thus eliminating the need to issue a transaction to convey information.

This makes it possible to acquire data from other chains asynchronously. A sample code is shown below.

bool data = Contract(1, contractAddress).ownerOf(msg.sender); // true or false if(data){ ... } else { ... }

The problem with this solution is that it is developed in a original chain, so although data can be freely acquired in this chain, the problem of interoperability between other chains has not been solved, and the question is whether the development cost and value are commensurate.

1. On Chain Data Feed

[

data feed

1920×1080 86.8 KB

](https://ethresear.ch/uploads/default/original/2X/e/ee719615973a924ef223f8eaafe9c505e8936c71.png)

The developer defines in advance the data developers want to acquire and our protocol stores it in a specific contract.

This is an on-chain version of Chainlink’s data feed.

Since the data will be acquired in the contract, there will be no cost for DApps users to acquire the data.

There are two problems.

The first is that the data is stored in the contract, so the acquired data may not necessarily be the latest data.

The second is that the data that can be acquired is limited.

Use Cases

We believe that linking data between blockchains can provide a new UX that has never existed before.

There are many ways to use data from other chains, but we would like to introduce two.

Cross-chain NFTFi

.

You can get information about NFTs in another chain and write code for the source chain based on that data. For example, you can easily collaborate and giveaway with NFT projects in completely different chains.

DID

.

Personal information stored in any chain can be accessed by other chains. This allows you to use information from DID applications made by other chains and use it to authenticate your own applications.

Currently we have not yet reached the best solution.Tell me what you think.