Relayer

Relayer Module

The WormholeRelayer module allows developers to deliver their VAAs via an untrustedDeliveryProvider , rather than needing to develop and host their own relay infrastructure.

Interacting with the Module

There are three relevant interfaces to discuss when utilizing the WormholeRelayer module:

- IWormholeRelayer
  - the primary interface by which you interact with the module. It allows you to request the sending of messages and VAAs.
- IWormholeReceiver
  - this is the interface you are responsible for implementing. It allows the selected Delivery Provider to deliver messages/VAAs to your contract.
- IDeliveryProvider
  - this interface represents the delivery pricing information for a given relayer network. Each delivery provider implements this on every blockchain they support delivering from.

Check theEVM page for contract addresses on each supported blockchain.

Check theHello Wormhole page for a quick example on using these interfaces!

Delivery Guarantees & Delivery Failures

The WormholeRelayer protocol is intended to create a service interface whereby mutually distrustful integrators and DeliveryProviders can work together to provide a seamless Dapp experience. You don't trust the delivery providers with your data, and the delivery providers don't trust your smart contract. The primary agreement which is made between integrators and delivery providers is that:

When a delivery is requested, the delivery provider will attempt to deliver the VAA within the provider's stated delivery timeframe.

This creates a marketplace whereby providers can set different price levels and service guarantees. Delivery providers effectively accept the slippage risk premium of delivering your VAAs in exchange for a set fee rate. Thus, the providers agree to deliver your messageseven if they have to do so at a loss .

Delivery providers should set their prices such that they turn a profit on average, but not necessarily on every single transfer. Thus, some providers may choose to set higher rates for tighter guarantees, or lower rates for less stringent guarantees.

Delivery Statuses

All deliveries should result in one of following four outcomes prior to the delivery timeframe of the delivery provider. These outcomes are emitted as EVM events from the WormholeRelayer contract when they occur.

- (0) Delivery Success
- (1) Receiver Failure
- (2) Forward Request Success
- (3) Forward Request Failure

A receiver failure is a well-defined term which means that the selected providerperformed the delivery, but the delivery was not able to be completed. There are only three causes for a delivery failure:

- the target contract does not implement theIWormholeReceiver
- interface
- the target contract threw an exception or reverted during execution ofreceiveWormholeMessages
- the target contract exceeded the specifiedgasLimit
- while executingreceiveWormholeMessages

All three of these scenarios should generally be avoidable by the integrator, and thus it is up to integrator to resolve them.

Any other senario which causes a delivery to not be performed should be considered anoutage by some component of the

system, including potentially the blockchains themselves.

Forward Request Success and Forward Failure represent when the delivery succeeded, and a 'forward' was requested by the user during the delivery. If the user had enough funds left over as a refund to complete the forward, the forward will be executed and the status will be 'Forward Request Success' - otherwise, it will be 'Forward Request Failure'.

Last updated1 month ago

On this page * Relayer Module * Interacting with the Module * Delivery Guarantees & Delivery Failures

Was this helpful? Edit on GitHub