Connector Behavior

This document provides insights on how Conduit communicates with a connector.

Conduit Connector Protocol

Conduit expects all connectors to follow the Conduit Connector Protocol . The connector protocol is a set of protobuf files describing the interface between Conduit and the connector in the form of gRPC services. This approach allows connectors to be written in any language with support for gRPC.

The connector protocol splits the connector interface in 3 gRPC services - one for the source, another for the destination, and a third one for the connector specifications. A connector needs to implement the specifications and at least the source or destination.

Note that you don't need to use the connector protocol directly - we provide a Go connector SDK that hides the complexity of the protocol and simplifies the implementation of a connector.

Standalone vs built-in connectors

While the Conduit Connector Protocol decouples Conduit from its connectors by using gRPC, it also provides a thin Go layer that allows any Go connector to be compiled into the Conduit binary as a built-in connector. The following diagram shows how Conduit communicates with a standalone connector and a built-in connector.

Standalone connectors are run as separate processes, separate from the Conduit process. They need to have an entrypoint (binary or script) which runs the connector and starts the gRPC server responsible for communicating with Conduit. A standalone connector process is started and stopped by Conduit on demand. One connector process will be started for every pipeline connector in Conduit.

Built-in connectors on the other hand are executed in the same process as Conduit and communicate with Conduit through Go channels instead of gRPC. Any connector written in Go can be compiled into the Conduit binary and used as a built-in connector.

Find out more about the Conduit connector plugin architecture .

Protocol gRPC Interface

The protocol interface is hosted on the Buf schema registry . Use it as a starting point when implementing a connector in a language other than Go. Edit this page Previous Referencing Connectors Next Building Connectors