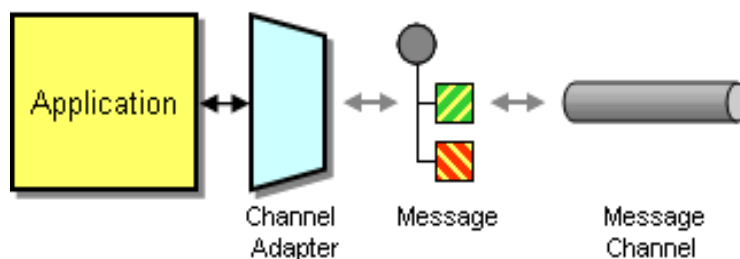


CHANNEL ADAPTER AND MESSAGING BRIDGE

Channel Adapter

Many enterprises use *Messaging* to integrate multiple, disparate applications.

How can you connect an application to the messaging system so that it can send and receive messages?



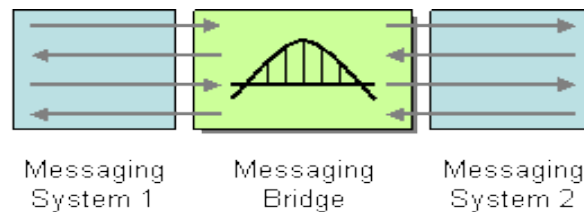
Use a *Channel Adapter* that can access the application's API or data and publish messages on a channel based on this data, and that likewise can receive messages and invoke functionality inside the application.

The adapter acts as a messaging client to the messaging system and invokes applications functions via an application-supplied interface. This way, any application can connect to the messaging system and be integrated with other applications as long as it has a proper *Channel Adapter*.

Messaging Bridge

An enterprise is using *Messaging* to enable applications to communicate. However, the enterprise uses more than one messaging system, which confuses the issue of which messaging system an application should connect to.

How can multiple messaging systems be connected so that messages available on one are also available on the others?



Use a *Messaging Bridge*, a connection between messaging systems, to replicate messages between systems.

Typically, there is no practical way to connect two complete messaging systems, so instead we connect individual, corresponding channels between the messaging systems. The *Messaging Bridge* is a set of *Channel Adapters*, where the non-messaging client is actually another messaging system, and where each pair of adapters connects a pair of corresponding channels. The bridge acts as map from one set of channels to the other, and also transforms the message format of one system to the other.

The connected channels may be used to transmit messages between traditional clients of the messaging system, or strictly for messages intended for other messaging systems.

Source:

<http://www.enterpriseintegrationpatterns.com/patterns/messaging/MessagingBridge.html>