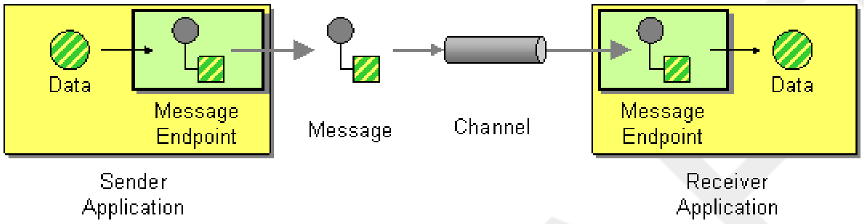
Message Endpoint

Overview

A *message endpoint* is the interface between an application and a messaging system. You can have a sender endpoint (sometimes called a proxy or a service consumer), which is responsible for sending *In* messages, and a receiver endpoint (sometimes called an endpoint or a service), which is responsible for receiving *In* messages.

Figure 3. Message Endpoint Pattern



Types of endpoint

Mediation Router defines two basic types of endpoint:

- *Source endpoint*—appears at the start of a Mediation Router route and is the source of all the *In* messages coming into the route (equivalent to a *receiver* endpoint).
- *Target endpoint*—appears at the end of a Mediation Router route and is the destination for the *In* messages leaving the route (equivalent to a *sender* endpoint). As a matter of fact, it is also possible to define a route with more than one target endpoint.

Endpoint URIs

In Mediation Router, an endpoint is represented by an *endpoint URI*, which typically encapsulates the following kinds of data:

- *Endpoint URI for a source endpoint*—can advertise a specific location (for example, to expose a service to which senders can connect). Alternatively, the URI could specify a message source, such as a message queue. The endpoint URI can include settings to configure the endpoint.
- *Endpoint URI for a target endpoint*—contains details of where to send messages and includes settings to configure the endpoint. In some cases,

the URI would specify the location of a remote receiver endpoint; in other cases, the destination could have an abstract form, such as a queue name.

An endpoint URI in Mediation Router has the following general form:

```
ComponentPrefix:ComponentSpecificURI
```

Where `ComponentPrefix` is a URI prefix that identifies a particular Mediation Router component (see *Components* in the *Component Reference* for details of all the supported components). The remaining part of the URI, `ComponentSpecificURI`, has a syntax defined by the particular component. For example, to connect to the JMS queue, `Foo.Bar`, you could define an endpoint URI like the following:

```
jms:Foo.Bar
```

To define a route that connects the source endpoint, `file://local/router/messages/foo`, directly to the target endpoint, `jms:Foo.Bar`, you could use the following Java DSL fragment:

```
from("file://local/router/messages/foo").to("jms:Foo.Bar");
```

Alternatively, you could define the same route in XML, as follows:

```
<camelContext id="CamelContextID" xmlns="http://act
ivemq.apache.org/camel/schema/spring">
  <route>
    <from uri="file://local/router/messages/foo"/>
    <to uri="jms:Foo.Bar"/>
  </route>
</camelContext>
```