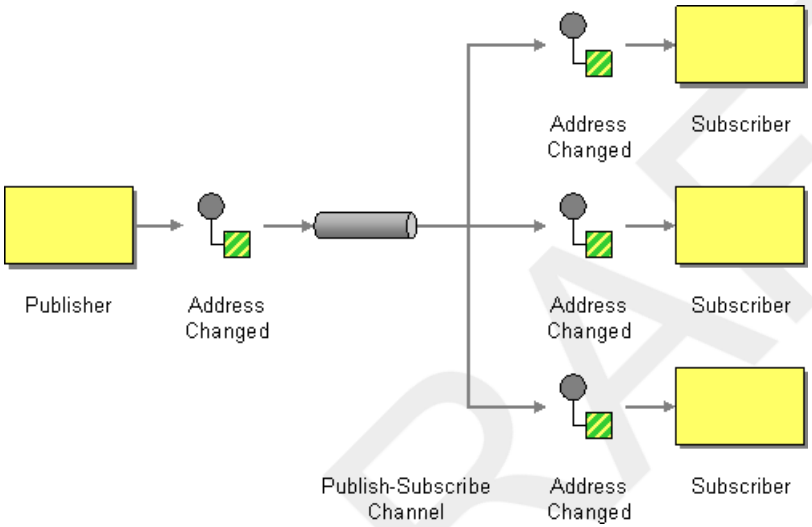
Publish Subscribe Channel

Overview

A *publish-subscribe channel* is a message channel on page 22 that enables multiple subscribers to consume any given message (contrast this with a point-to-point channel on page 36). Publish-subscribe channels are frequently used as a means of broadcasting events or notifications to multiple subscribers.

Figure 10. Publish Subscribe Channel Pattern



Components that support publish-subscribe channel

The following Mediation Router components support the publish-subscribe channel pattern:

- JMS on page 38
- xActiveMQ on page 39
- XMPP on page 39

JMS

In JMS, a publish-subscribe channel is represented by a *topic*. For example, you could specify the endpoint URI for a JMS topic called `StockQuotes` as follows:

```
jms:topic:StockQuotes
```

See JMS in the *Component Reference* for more details.

ActiveMQ

In ActiveMQ, a publish-subscribe channel is represented by a topic. For example, you could specify the endpoint URI for an ActiveMQ topic called `StockQuotes` as follows:

```
activemq:topic:StockQuotes
```

See ActiveMQ in the *Component Reference* for more details.

XMPP

The XMPP (Jabber) component supports the publish-subscribe channel pattern when it is used in the group communication mode. See XMPP in the *Component Reference* for more details.

Static subscription lists

If you prefer, you can also implement publish-subscribe logic within the Mediation Router application itself. A simple approach is to define a *static subscription list*, where the target endpoints are all explicitly listed at the end of the route (this approach is not as flexible as a JMS or ActiveMQ topic, however).

Java DSL example

The following Java DSL example shows how to simulate a publish-subscribe channel with a single publisher, `seda:a`, and three subscribers, `seda:b`, `seda:c`, and `seda:d` (works only for the *InOnly* message exchange pattern):

```
from("seda:a").to("seda:b", "seda:c", "seda:d");
```

XML configuration example

The following example shows how to configure the same route in XML:

```
<camelContext id="buildStaticRecipientList" xmlns="http://act
ivemq.apache.org/camel/schema/spring">
  <route>
    <from uri="seda:a"/>
    <to uri="seda:b"/>
    <to uri="seda:c"/>
    <to uri="seda:d"/>
  </route>
</camelContext>
```