

## Chapter 2: Working with components and patterns

### Flow references

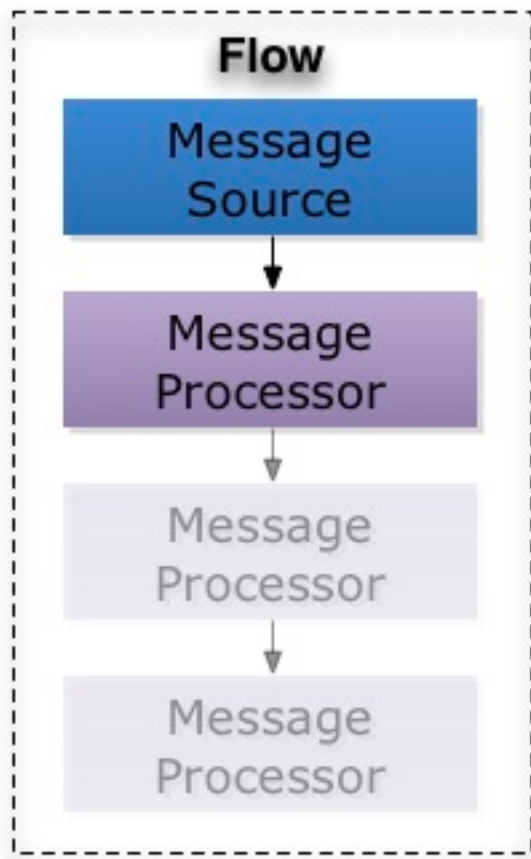
- You can synchronously execute another flow from a first one.
- Mule ESB executes the alternate flow and then return to the original one.

[1]

### Flows in Mule ESB:

- A **Flow** is a mechanism for orchestration supported in the message capabilities of the Mule ESB.
- A **Flow** uses connectors and messages.
- One may use a **Flow** when:
  - Simple integration tasks are needed.
  - Processing data based on scheduled.
  - Connect cloud and on-premise applications.
  - Event processing when multiple services must be composed.

### Anatomy of a Flow:



**Figure:** Anatomy of a flow (<http://goo.gl/ZhPS55>)

- A basic structure of a flow may be as follows:

```
<flow name="">
  - 0..1 MessageSource
  - 1..n MessageProcessor(s)
  - 0..1 ExceptionStrategy
</flow>
```

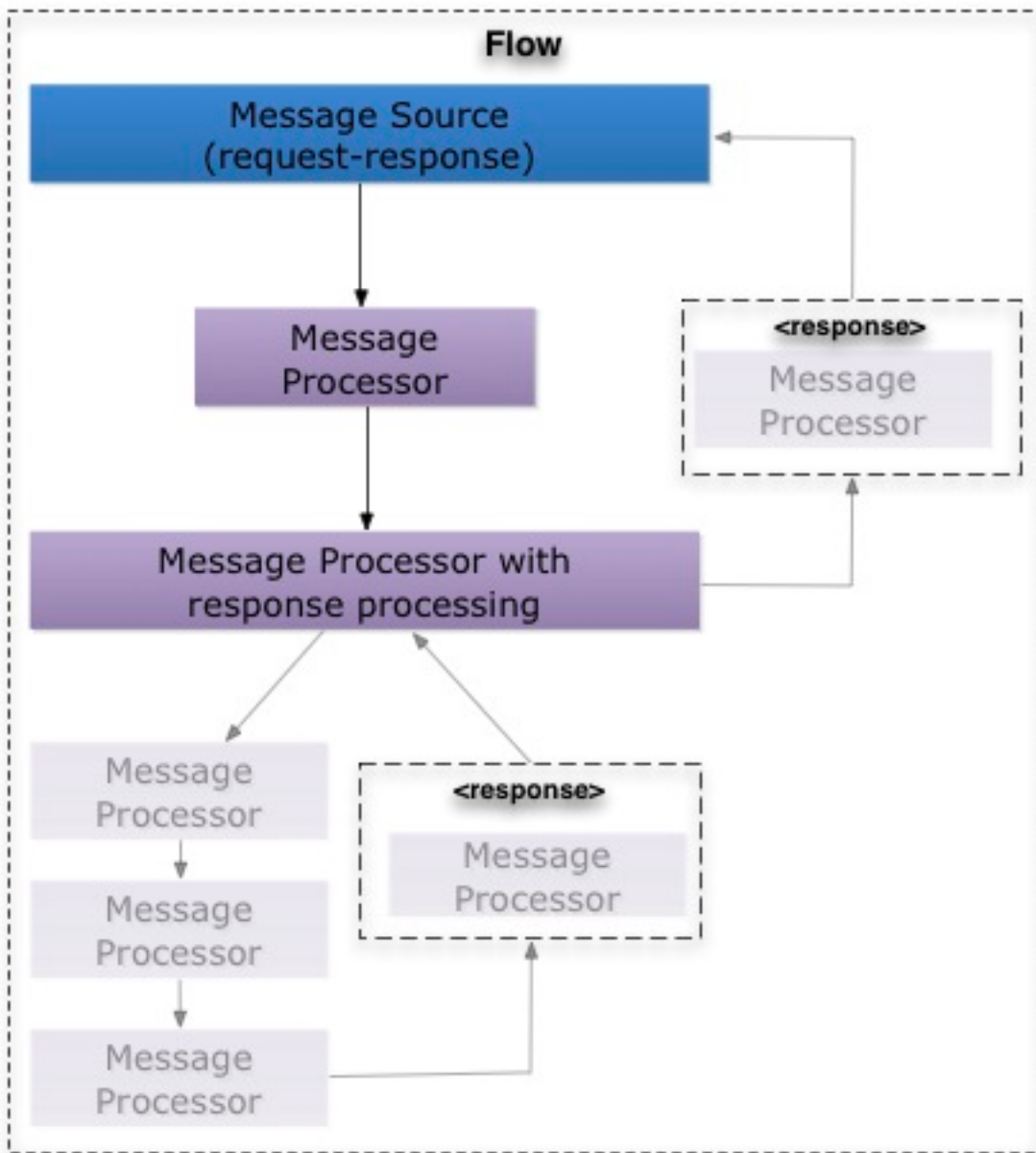
- **Flow example:**

```
<flow>
  <file:inbound-endpoint path="/myDirectory">
    <file:filename-filter name="*.xml"/>
  </file:inbound-endpoint>
  <xml:xslt-transformer xsl-file="bookOrderTransformation.xsl"/>
</flow>
```

```
<splitter expression="xpath://order"/>
  <!-- The following message processors will be invoked for each order in the
xml file -->
  <expression-filter expression="xpath://order[@type='book']"/>
  <component class="org.my.BookOrderProcessor"/>
  <smtp:outbound-endpoint subject="Order Confirmation" address=""/>
  <jdbc:outbound-endpoint />
  <default-exception-strategy>
    <jms:outbound-endpoint queue="failedOrders"/>
  </default-exception-strategy>
</flow>
```

## **Flow Behavior:**

- Normally, a **Flow** starts when a message is generated and from there all the message processors are executed one after another.
- A processor may have sub-processors and in that case the subs are executed before continuing the flow.
- In a request-response flow, the default response is assumed to be the last message generated by a processor. Unless something different is customized inside a message processor by defining a response block.



**Figure:** Flow behavior (<http://goo.gl/RlXYik>).

### Private flows

- Are regular flows but can't be accessed from outside the JVM via Mule Endpoint because has no message source defined.
- Used when referenced from another flow (through the flow-ref connector).
- Differs from a *Processor Chain* in the way that a Flow has its own context and exception strategy.

**[2]**

# Bibliography

2/26/15 18:17

**[1] Mule ESB Cookbook** (Attune infocom) (Dr. Zakir Laliwala / Azaz Desai / Abdul Samad / Uchit Vyas)

**[2] Mule ESB Flows**(MuleSoft)

(<http://www.mulesoft.org/documentation/display/current/Using+Flows+for+Service+Orchestration>)

