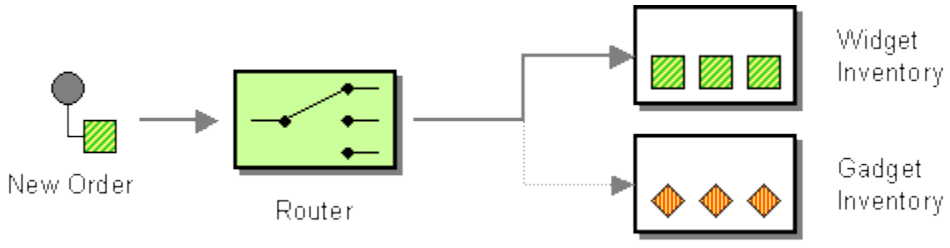
CONTENT-BASED ROUTER AND MESSAGE FILTER

Content-Based Router

Assume that we are building an order processing system. When an incoming order is received, we first validate the order and then verify that the ordered item is available in the warehouse. This function is performed by the inventory system. This sequence of processing steps is a perfect candidate for the *Pipes and Filters* style. We create two filters, one for the validation step and one for the inventory system, and route the incoming messages through both filters. However, in many enterprise integration scenarios more than one inventory system exists with each system being able to handle only specific items.

How do we handle a situation where the implementation of a single logical function (e.g., inventory check) is spread across multiple physical systems?



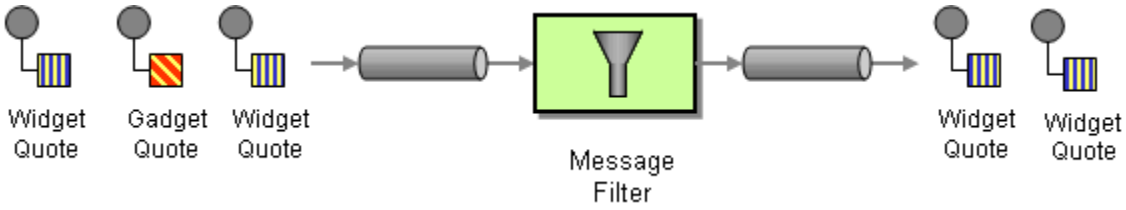
Use a *Content-Based Router* to route each message to the correct recipient based on message content.

The *Content-Based Router* examines the message content and routes the message onto a different channel based on data contained in the message. The routing can be based on a number of criteria such as existence of fields, specific field values etc. When implementing a *Content-Based Router*, special caution should be taken to make the routing function easy to maintain as the router can become a point of frequent maintenance. In more sophisticated integration scenarios, the *Content-Based Router* can take on the form of a configurable rules engine that computes the destination channel based on a set of configurable rules

Message Filter

Continuing with the order processing example, let's assume that company management publishes price changes and promotions to large customers. Whenever a price for an item changes, we send a message notifying the customer. We do the same if we are running a special promotion, e.g. all widgets are 10% off in the month of November. Some customers may be interested in receiving price updates or promotions only related to specific items. If I purchase primarily gadgets, I may not be interested in knowing whether widgets are on sale or not.

How can a component avoid receiving uninteresting messages?



Use a special kind of Message Router, a *Message Filter*, to eliminate undesired messages from a channel based on a set of criteria.

The *Message Filter* has only a single output channel. If the message content matches the criteria specified by the *Message Filter*, the message is routed to the output channel. If the message content does not match the criteria, the message is discarded.

Source:
http://www.enterpriseintegrationpatterns.com/patterns/messaging/Filter.html