

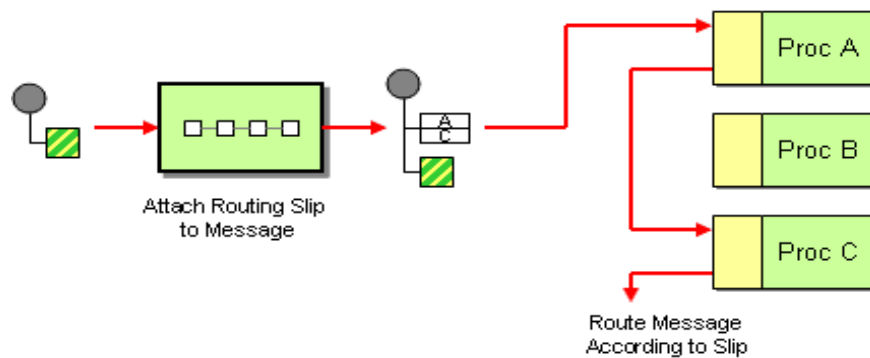
# ROUTING SLIP

Most of the routing patterns presented in this section route incoming messages to one or more destinations based on a set of rules. Sometimes, though, we need to route a message not just to a single component, but through a whole series of components. Let's assume, for example, that we use a *Pipes and Filters* architecture to process incoming messages that have to undergo a sequence of processing steps and business rule validations. Since the nature of the validations varies widely and may depend on external systems (e.g., credit card validations), we implement each type of step as a separate filter. Each filter inspects the incoming message, and applies the business rule(s) to the message. If the message does not fulfill the conditions specified by the rules it is routed to an exception channel. The channels between the filters determine the sequence of validations that the message needs to undergo.

Now let's assume, though, that the set of validations to perform against each message depends on the message type (for example, purchase order request do not need credit card validation or customers who send orders over a VPN may not require decryption and authentication).

To accommodate this requirement we need to find a configuration that can route the message through a different sequence of filters depending on the type of the message.

**How do we route a message consecutively through a series of processing steps when the sequence of steps is not known at design-time and may vary for each message?**



**Attach a *Routing Slip* to each message, specifying the sequence of processing steps. Wrap each component with a special message router that reads the *Routing Slip* and routes the message to the next component in the list.**

We insert a special component into the beginning of the process that computes the list of required steps for each message. It then attaches the list as a *Routing Slip* to the message and starts the process by routing the message to the first processing step.

After successful processing, each processing step looks at the *Routing Slip* and passes the message to the next processing step specified in the routing table.

Source:

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