

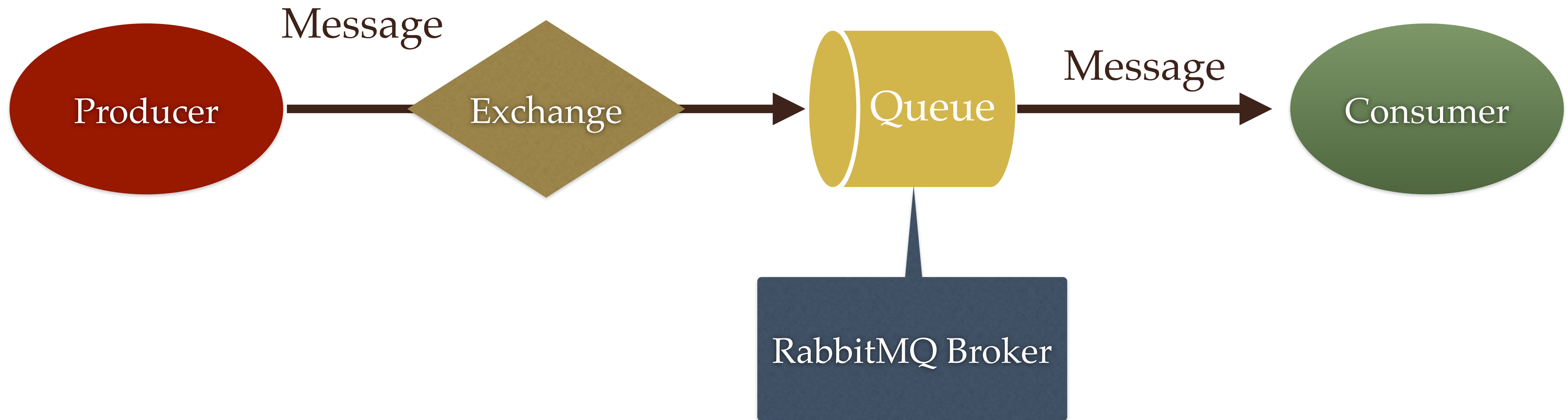


Core Concepts

By Ramesh Fadatare (Java Guides)

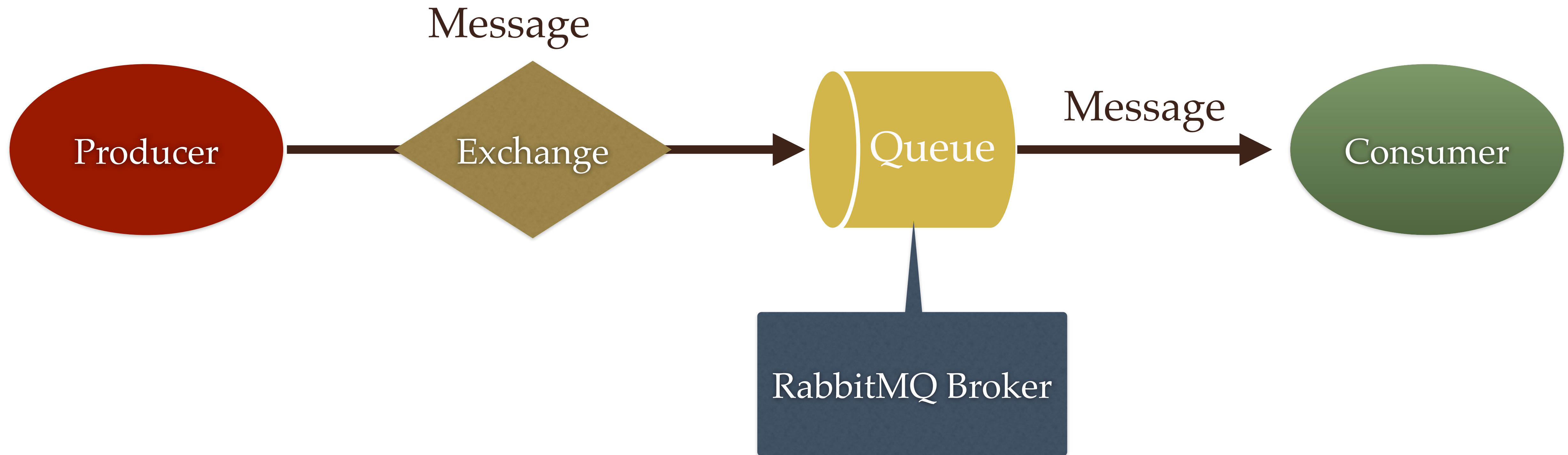
Producer

Producer is an application that sends messages. It does not send messages directly to the consumer. It send messages only to the RabbitMQ broker.



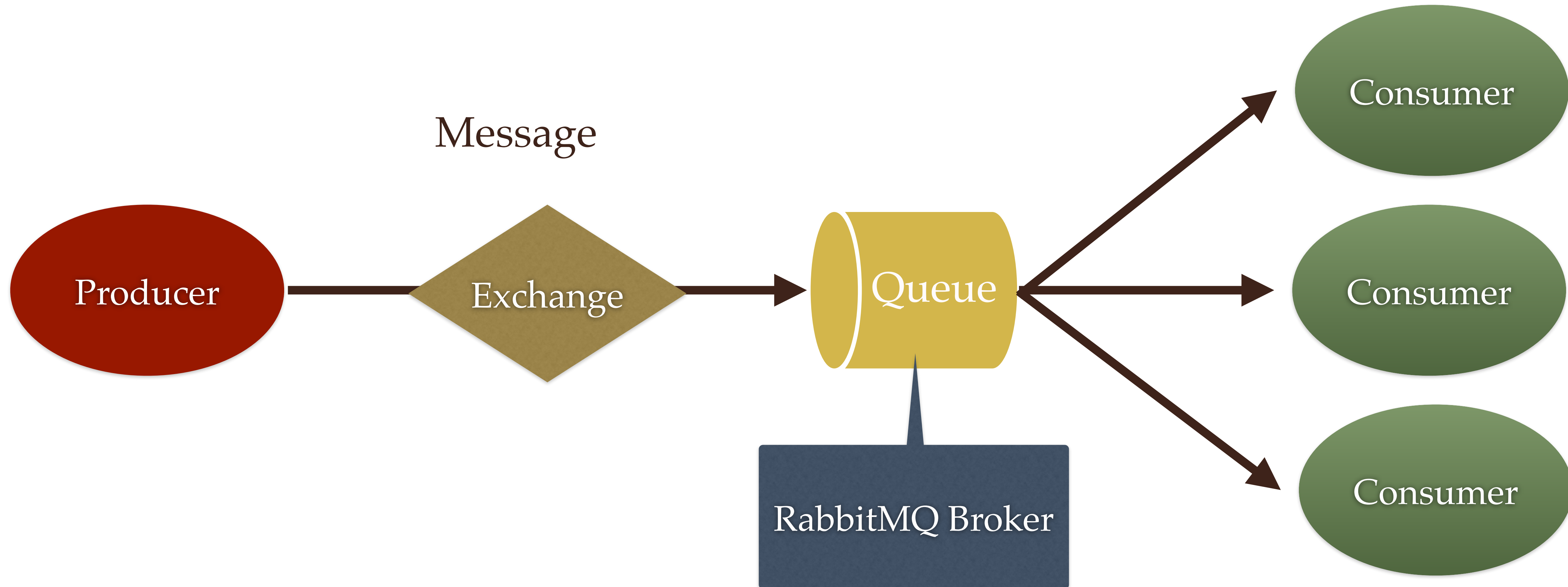
Consumer

Consumer is an application that reads messages from the RabbitMQ broker.



Consumer

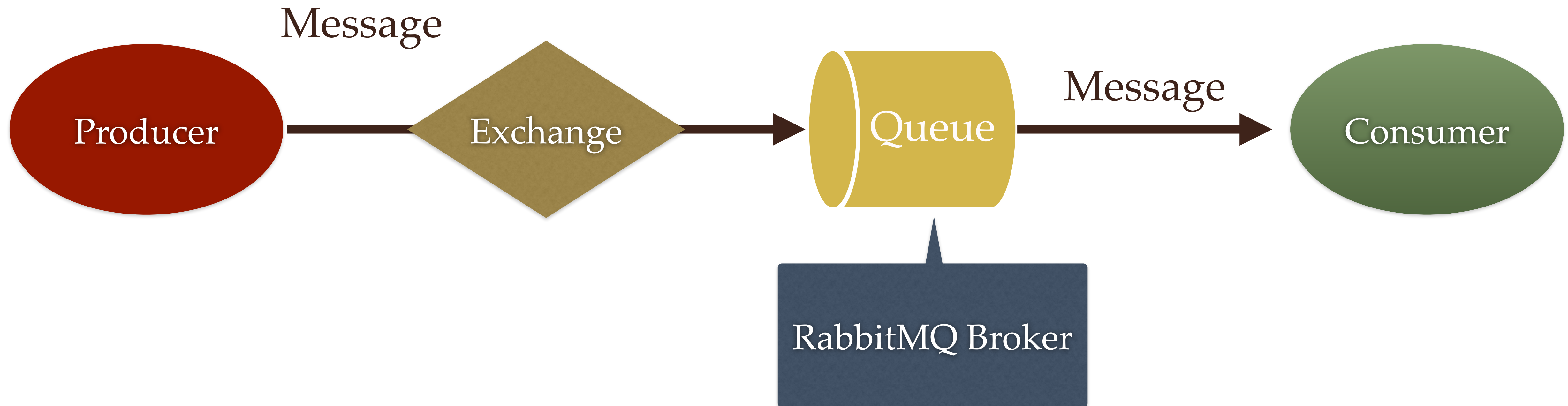
Consumer is an application that reads messages from the RabbitMQ broker.



Queue

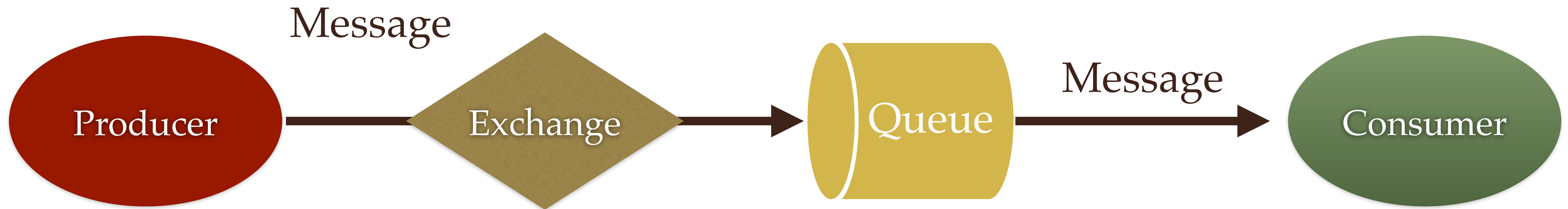
Queue is a buffer or storage in a RabbitMQ broker to store the messages.

The messages are put into a queue by a producer and read from it by a consumer. Once a message is read, it is consumed and removed from the queue. A message can thus only be processed exactly once.



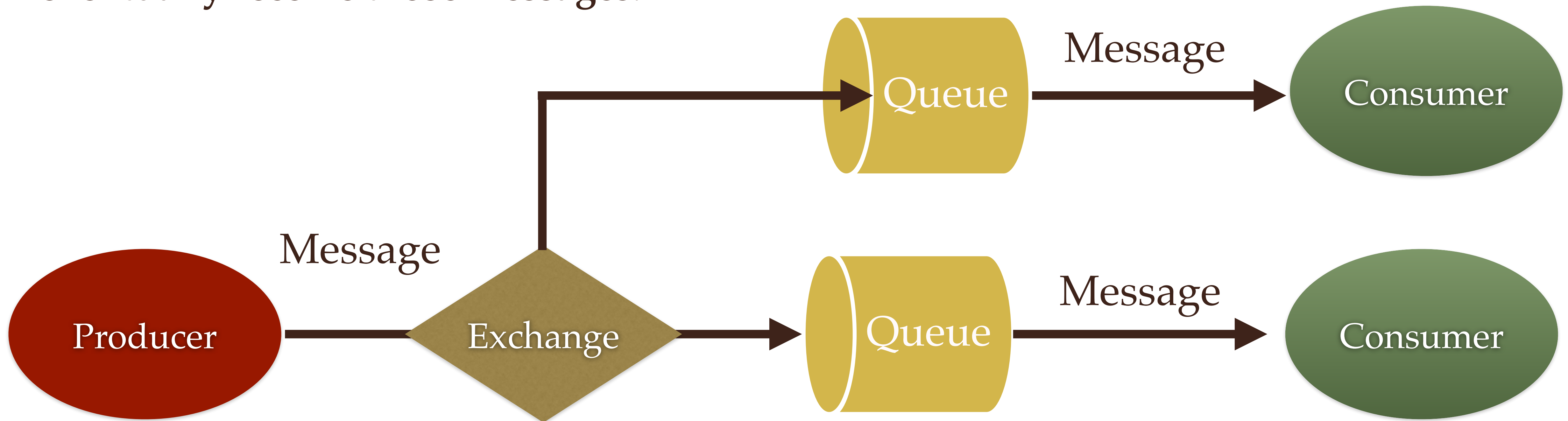
Message

Information that is sent from the producer to a consumer through RabbitMQ.



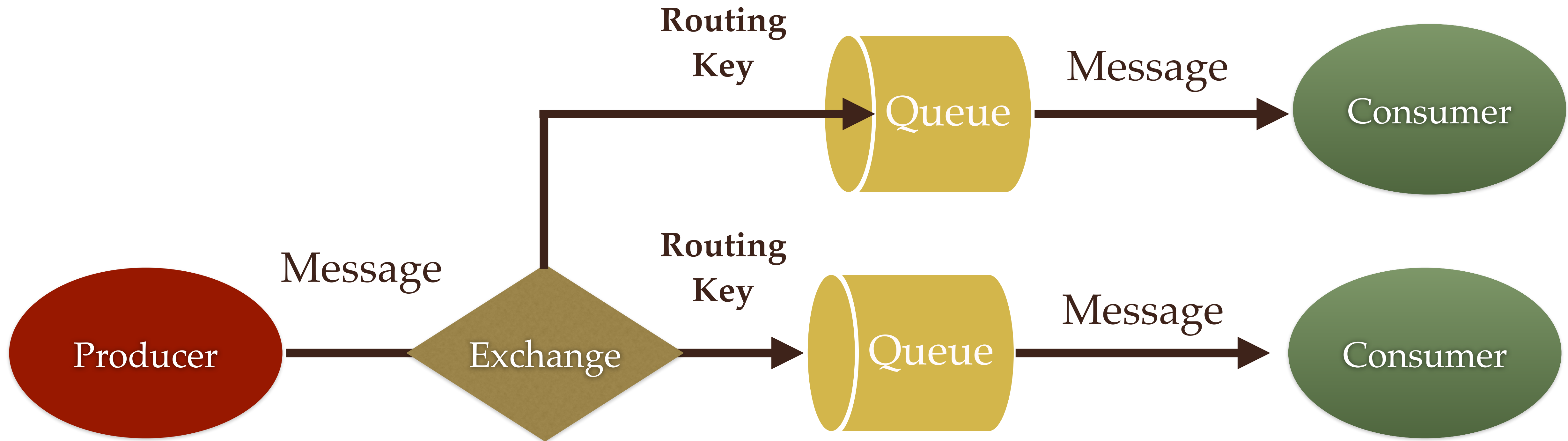
Exchange

Basically, it acts as an intermediary between the producer and a queue. Instead of sending messages directly to a queue, a producer can send them to an exchange instead. The exchange then sends those messages to one or more queues following a specified set of rules. Thus, the producer does not need to know the queues that eventually receive those messages.



Routing Key

The routing key is a key that the exchange looks at to decide how to route the message to queues. The routing key is like an address for the message.



Binding

A binding is a link between a queue and an exchange.

