## AMQP (Advanced Message Queuing Protocol) in brief :

Messaging brokers receive messages from publishers (applications that publish/send them, also known as producers) and route them to consumers (applications that process them).

## Publish path from publisher to consumer via exchange and queue

Some key features:

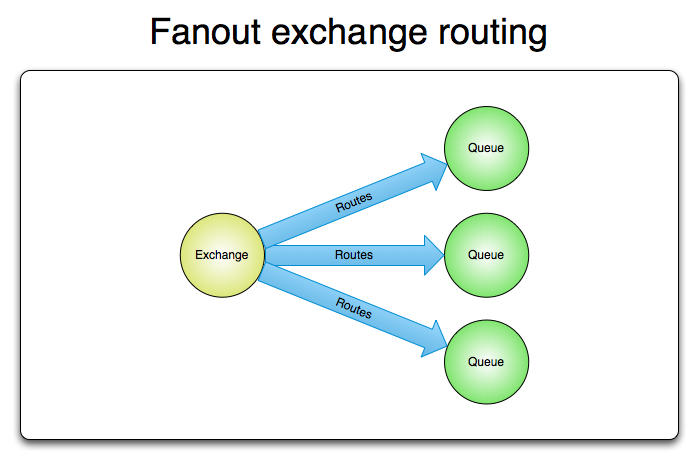
* Push message to consumers.
* Persist for Offline consumers.
* Reply when message received.

## Exchanges

In previous demo we sent and received messages to and from a queue. Now it's time to introduce the full messaging model in RabbitMQ: Exchange.

*Exchanges* are AMQP entities where messages are sent. Exchanges take a message and route it into zero or more queues. Those created during subscriptions, subscriptions could be instant or persisted. RabbitMQ supports many kinds of exchanges.

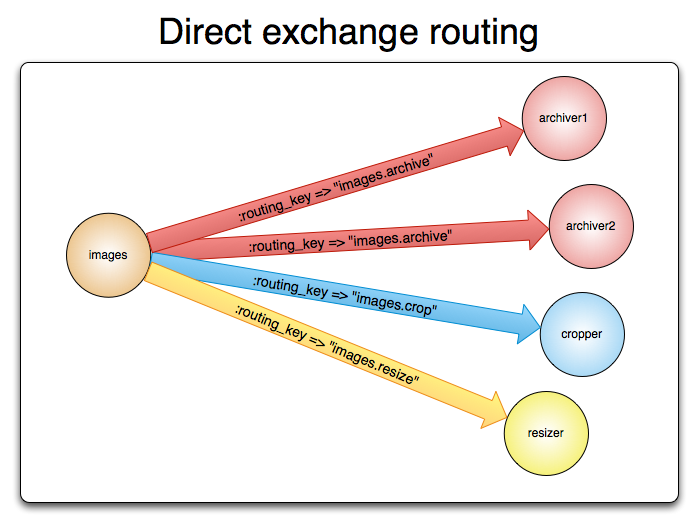
Fanout: The fanout exchange is very simple. It just broadcasts all the messages it receives to all the queues it knows (subscribed).



Direct exchange

From the previous demo, exchange broadcasts all messages to all consumers. What if we to allow filtering messages based on subscriber’s interests.

We will use a direct exchange instead. The routing algorithm behind a direct exchange is simple - a message goes to the queues whose binding key exactly matches the routing key of the message.



Topic exchange

Although using the direct exchange improved our system, it still has limitations - it can't do routing based on multiple criteria.

Messages sent to a topic exchange can't have an arbitrary routing\_key - it must be a list of words, delimited by dots. There can be as many words in the routing key as you like, up to the limit of 255 bytes.

* (star) can substitute for exactly one word.
* # (hash) can substitute for zero or more words.

It's easiest to explain this in an example:

