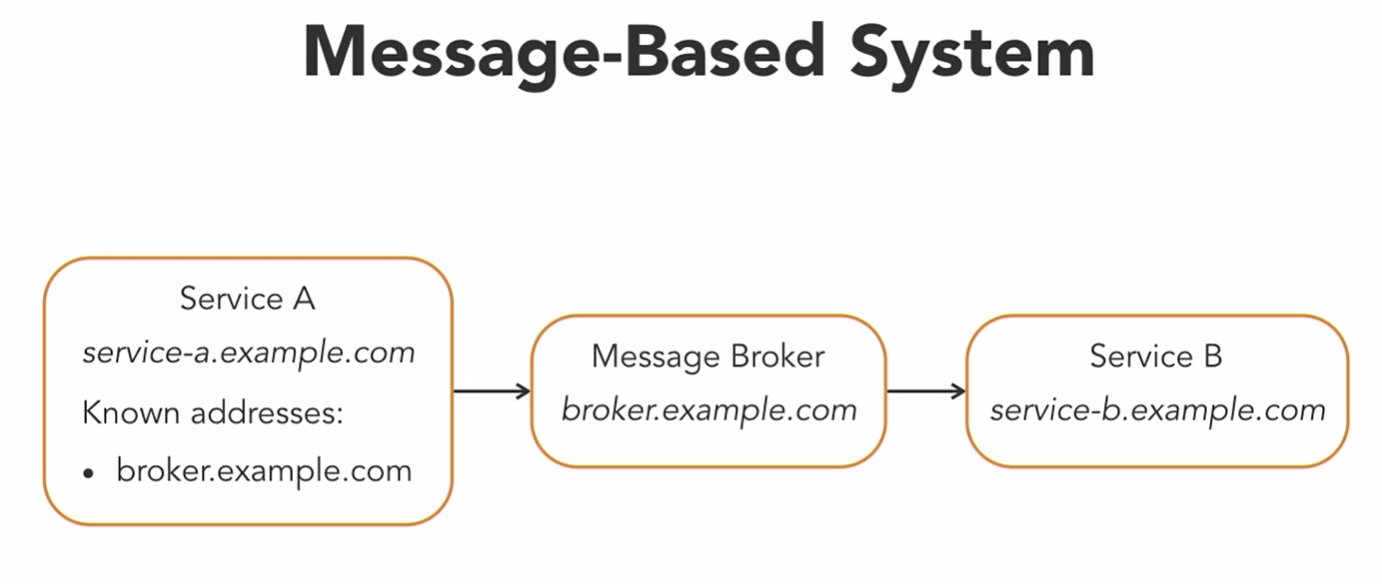
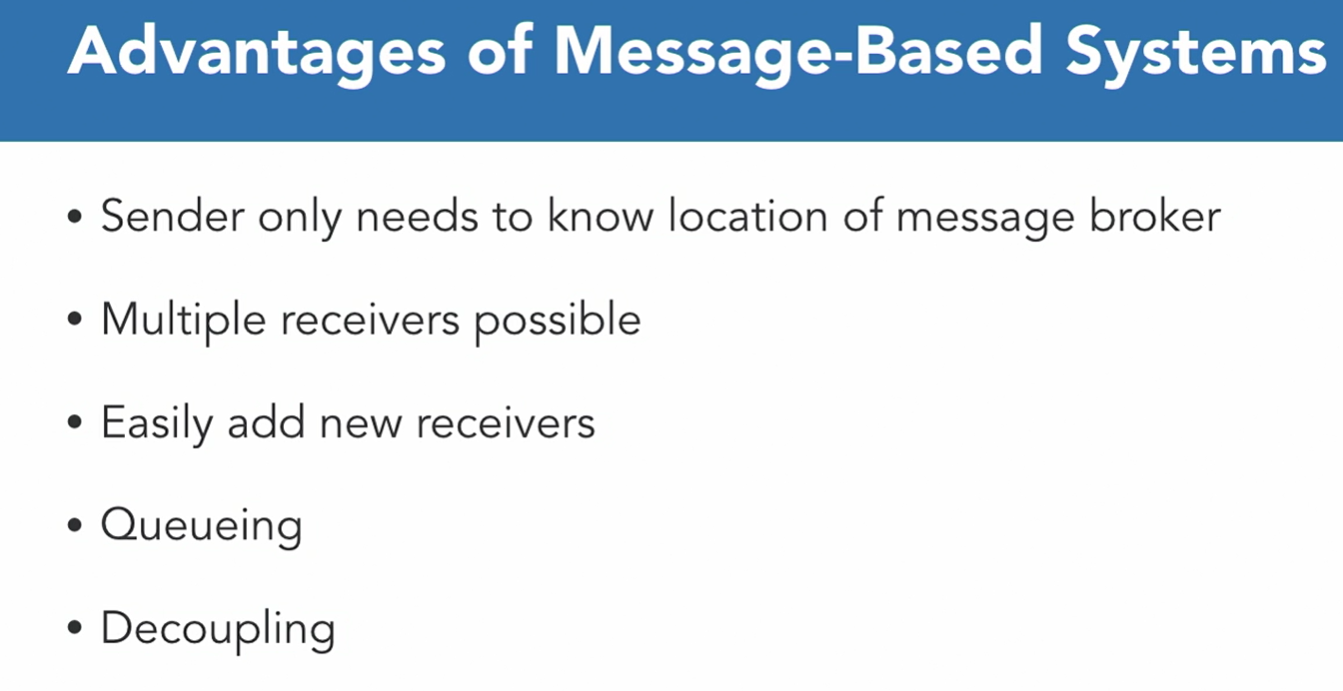
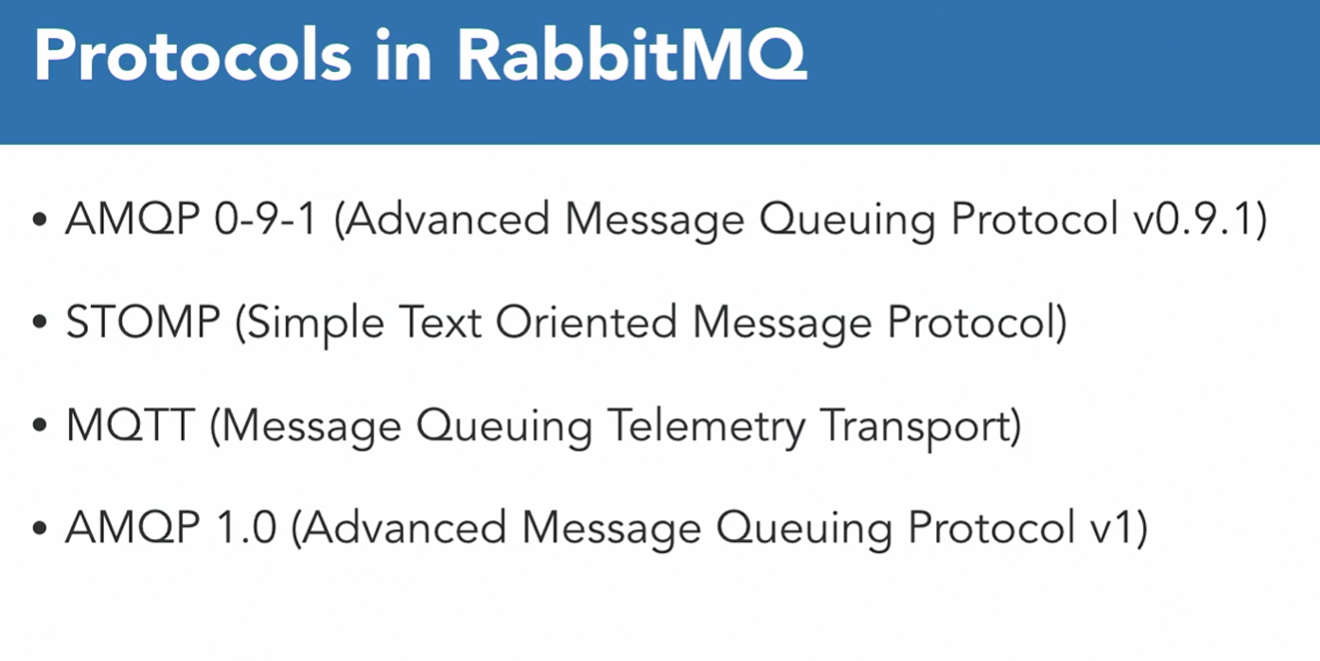


Jab address change hoga tw update karna hoga har jaga.

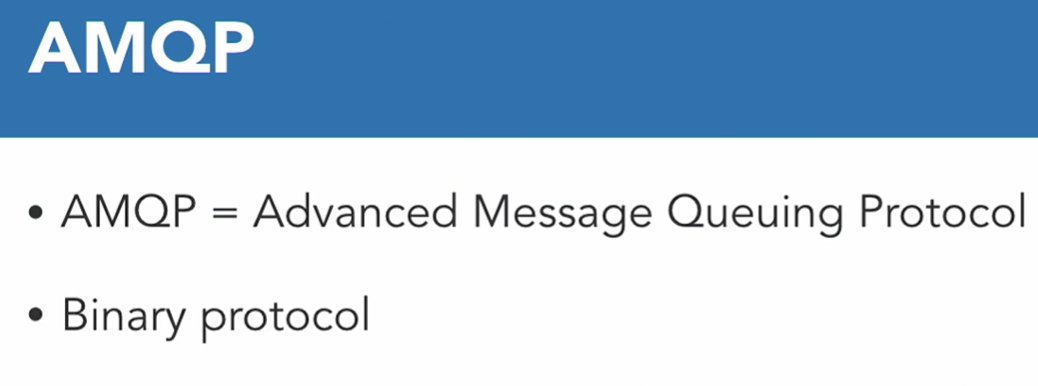
Message base Systems work differently in message base system we put message a message broker in between two services.



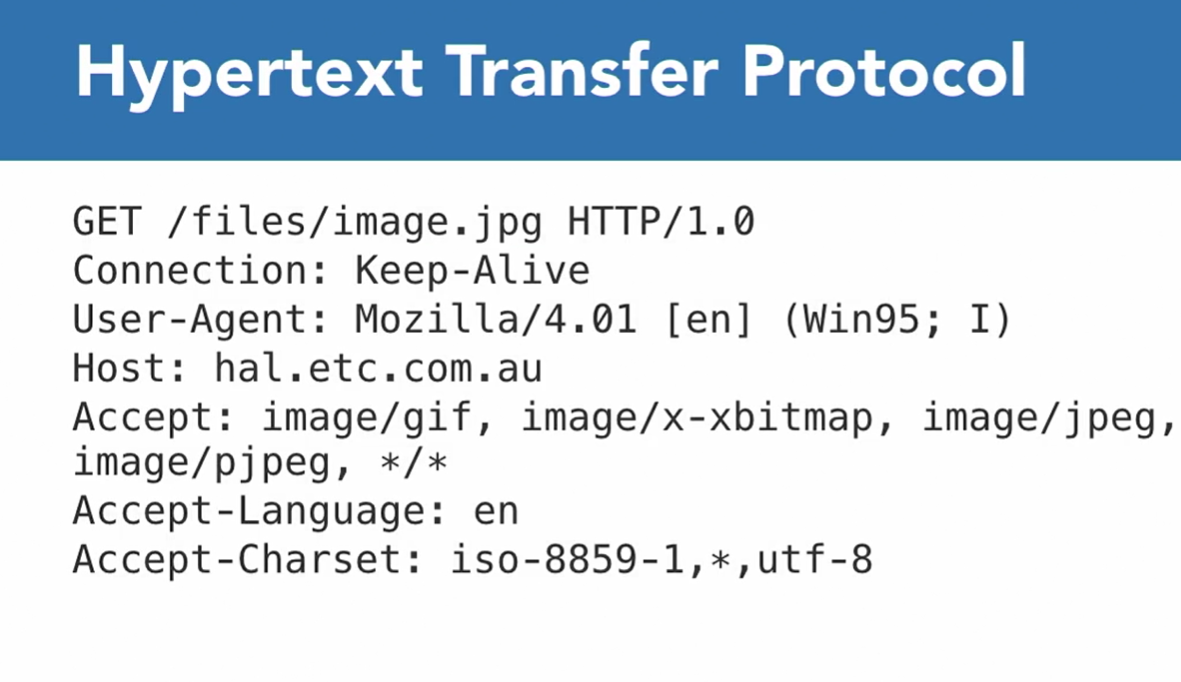


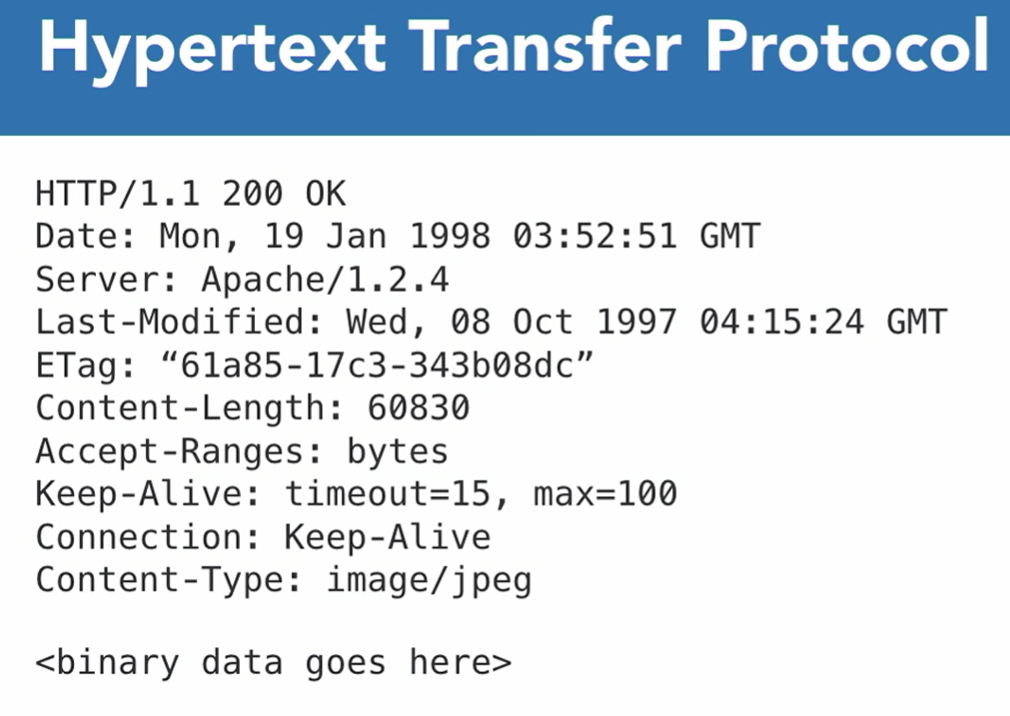


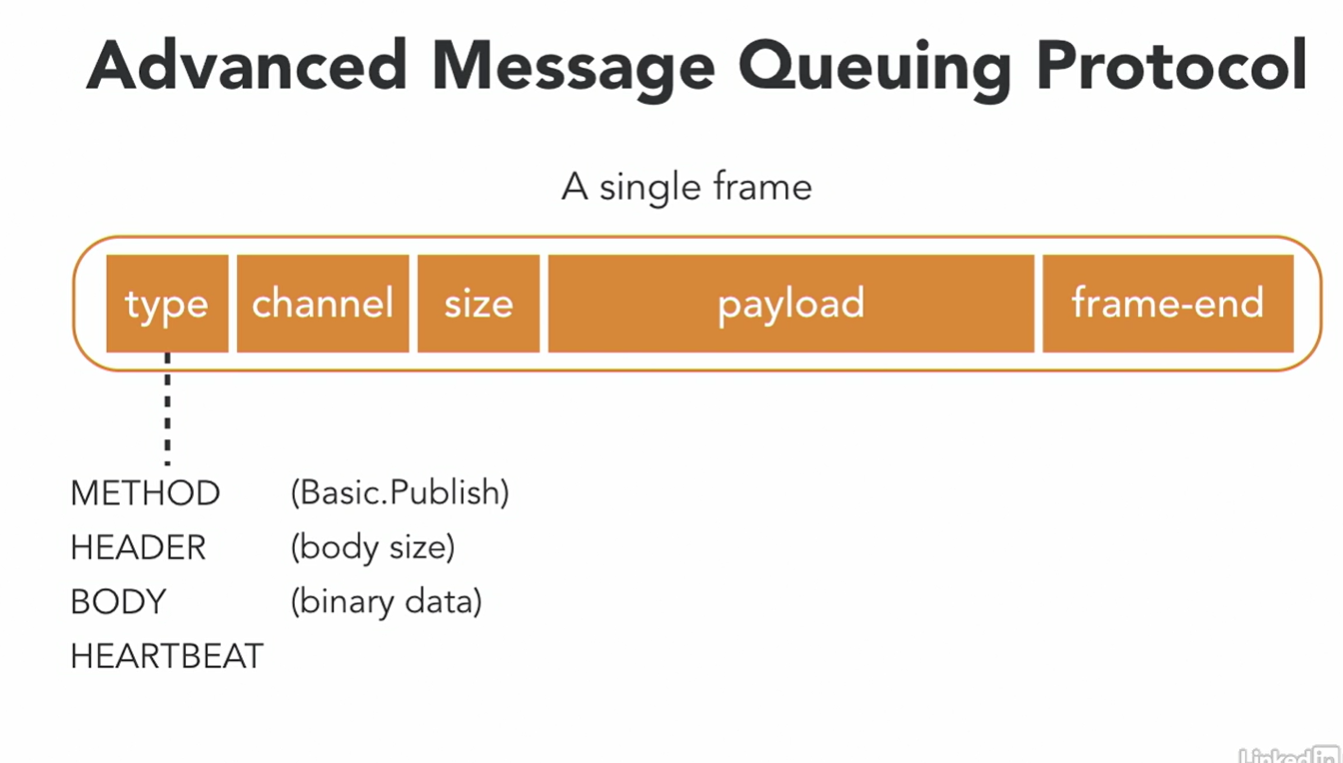
AMPQ 0-9-1 is probably the most popular and widely supportable.

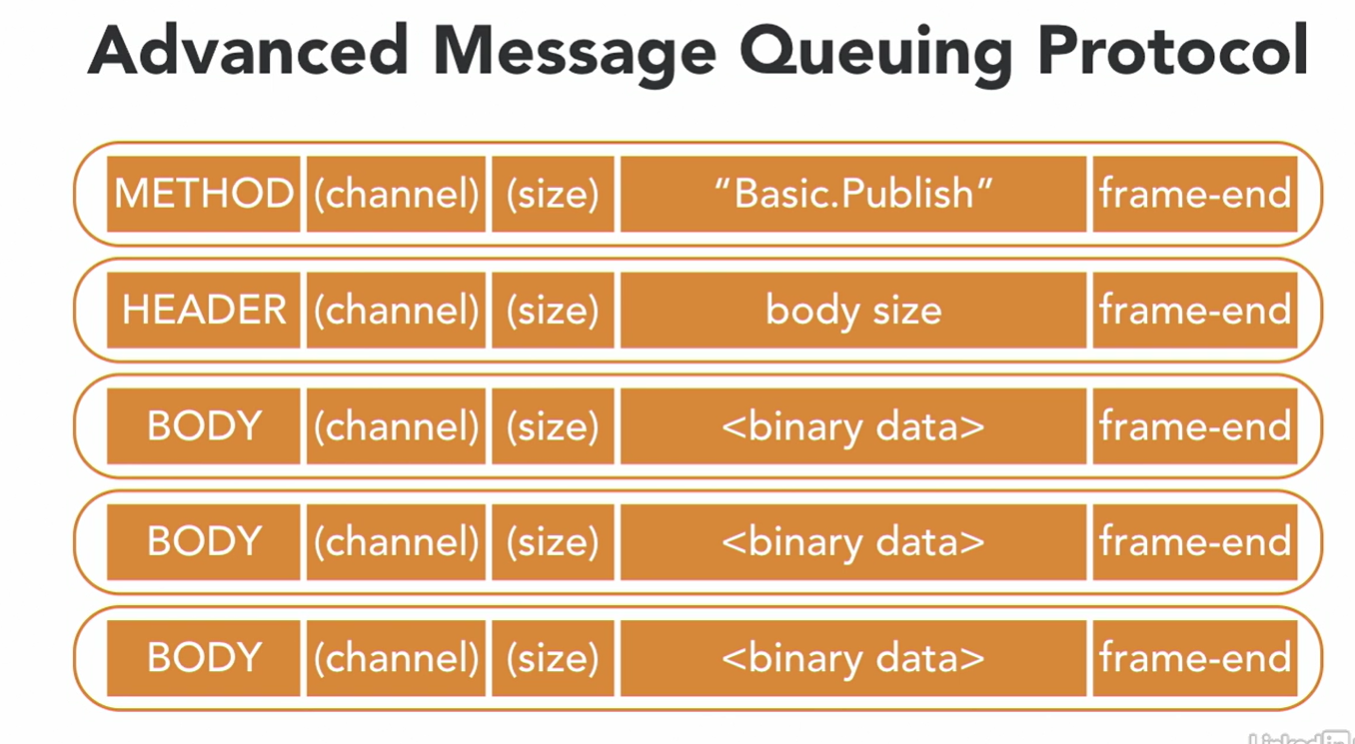


It’s a binary protocol which means your message will be send on binary format.

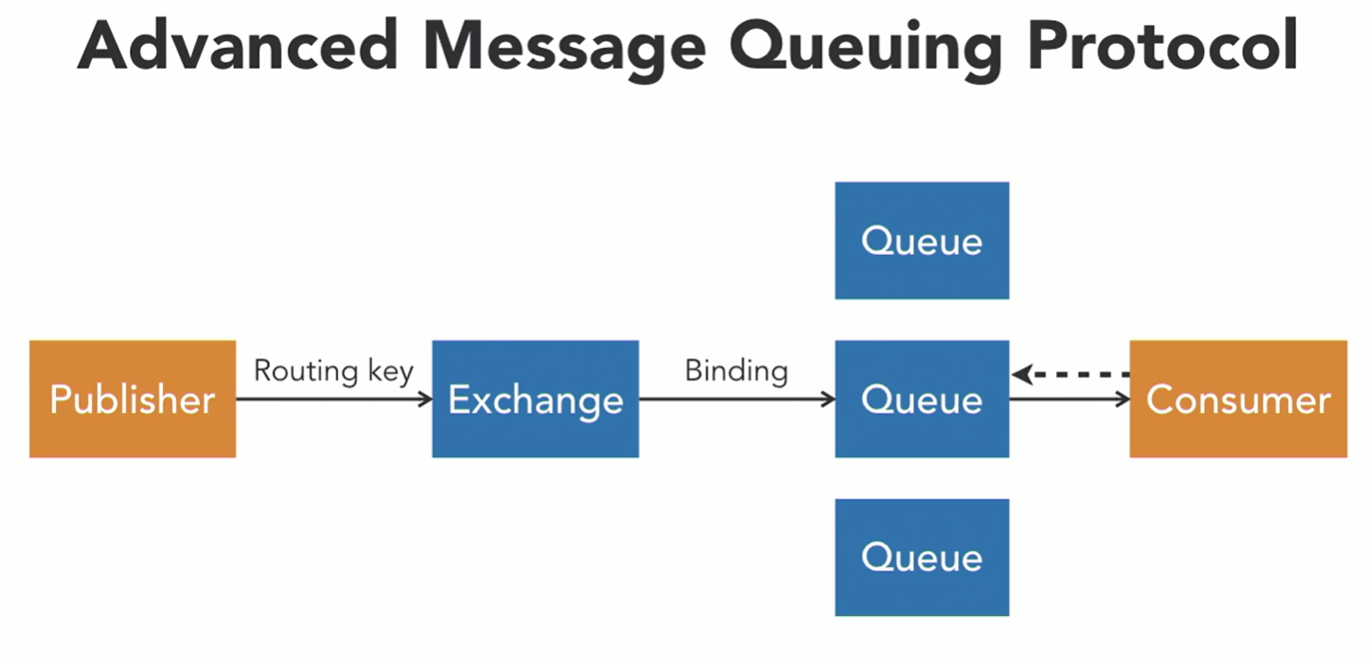


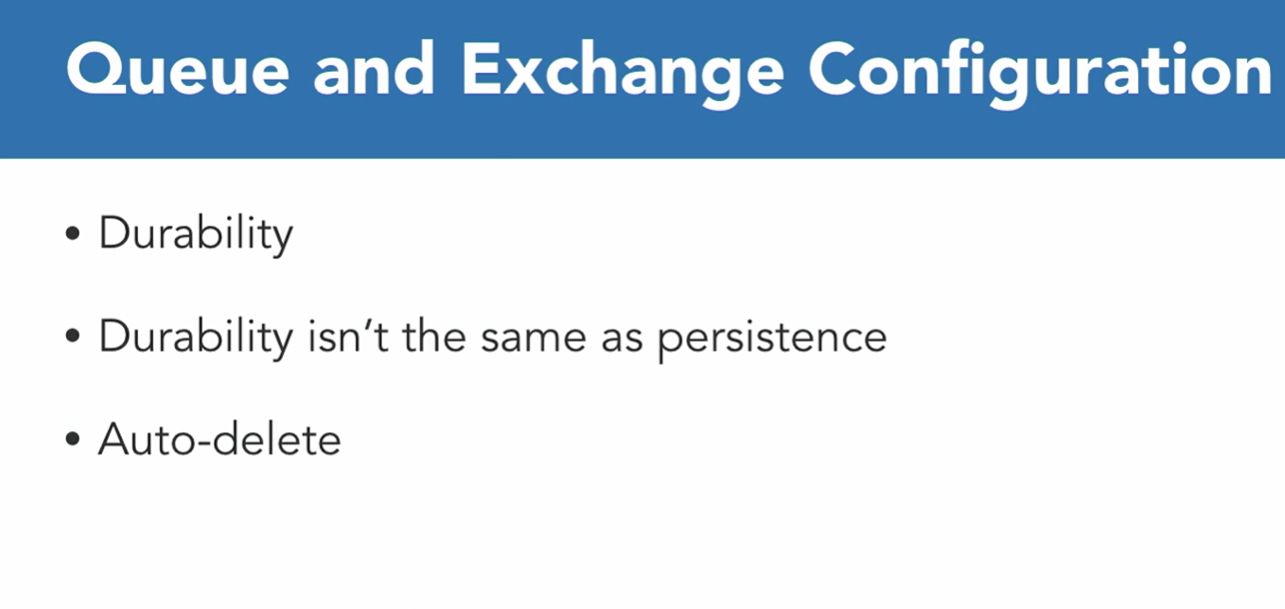




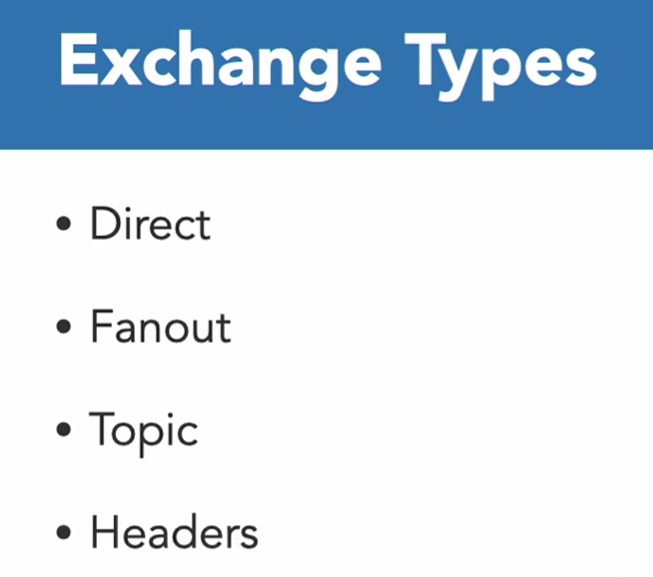


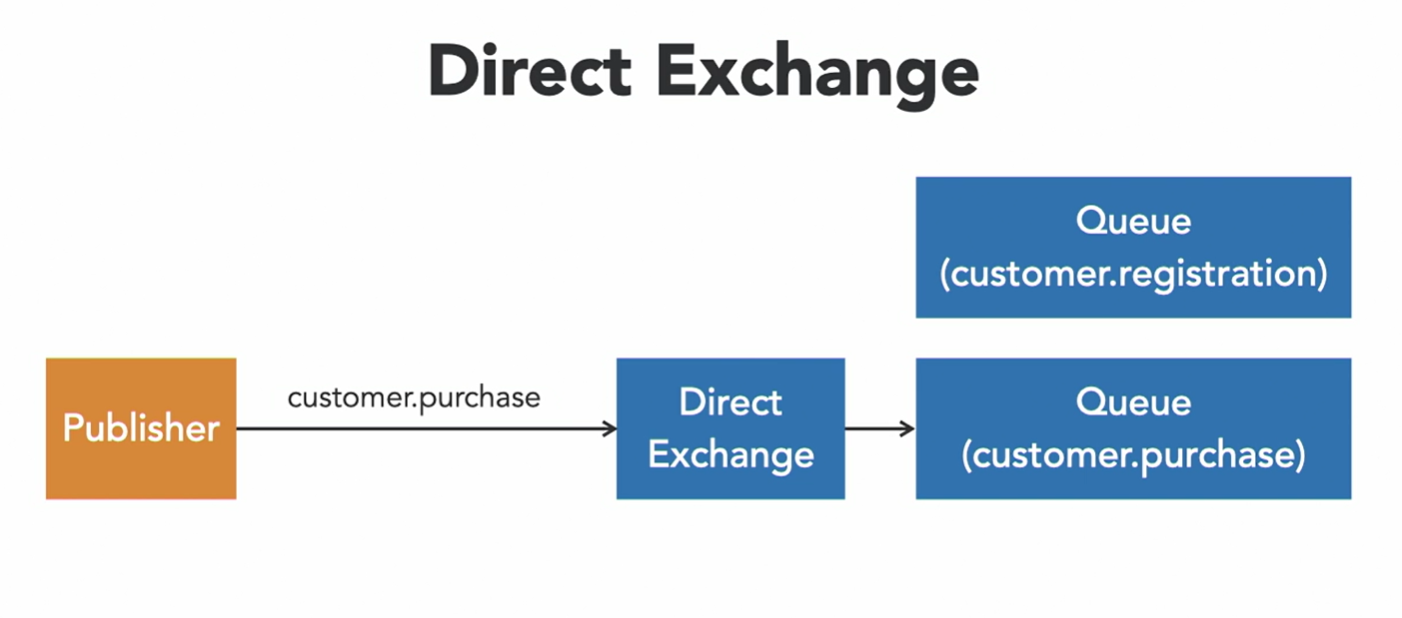
# RABBITMQ and the AMQ Model:

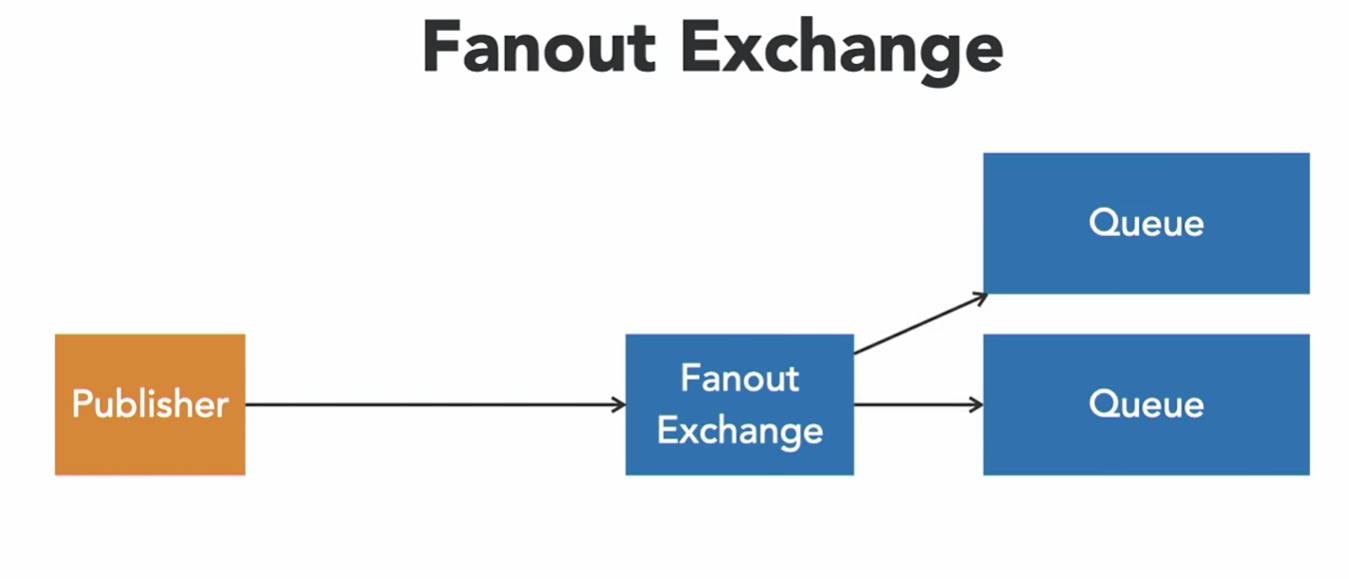


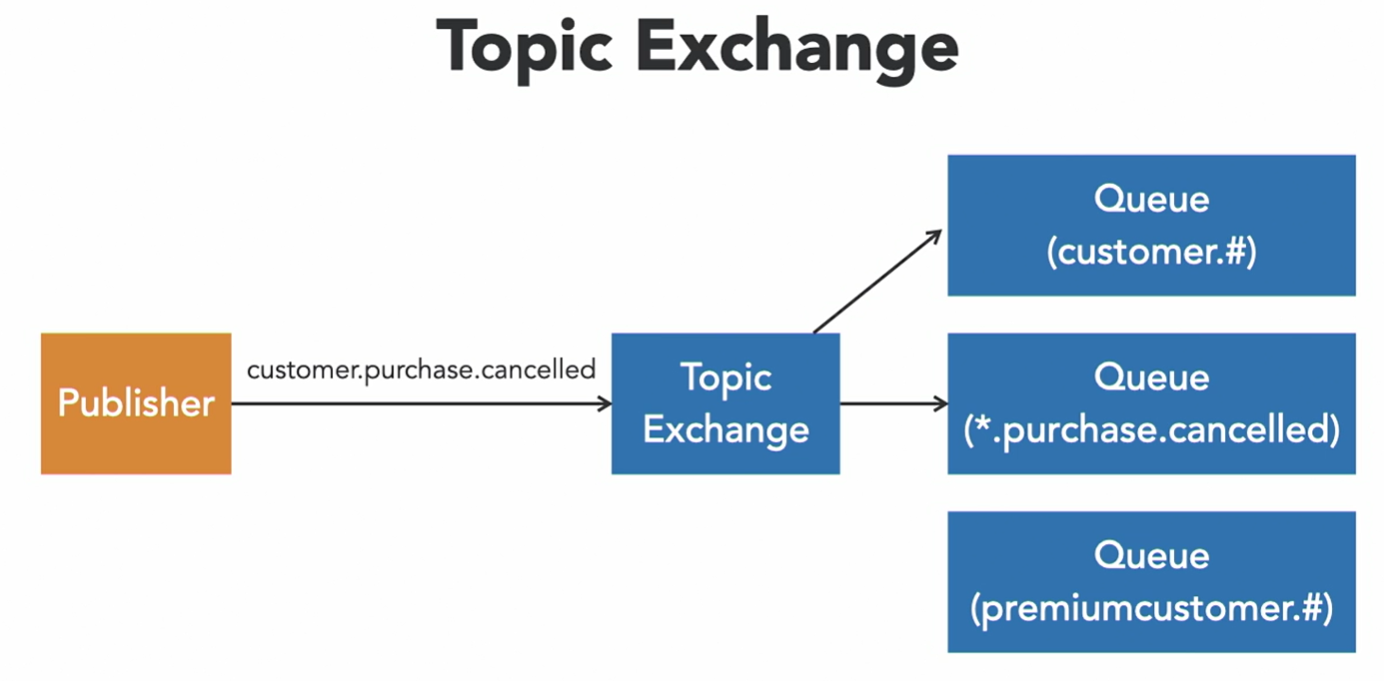


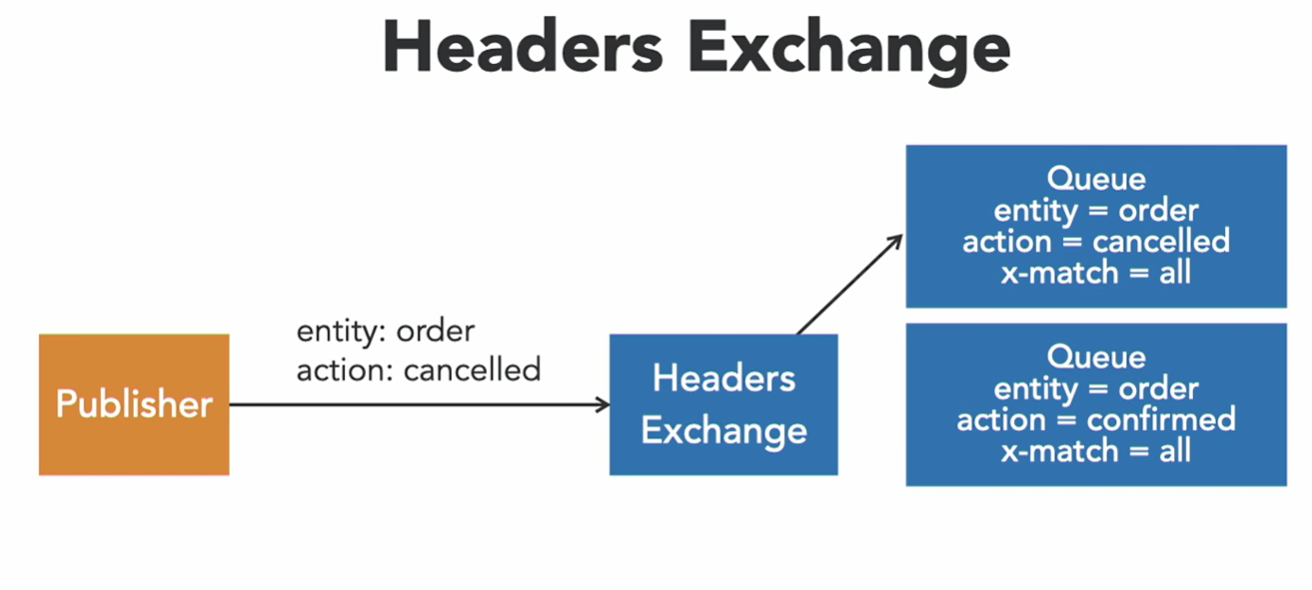
# Exchange Types











# Exchange type use cases:

# When to use what

The fanout exchange will route published messages to any queues bound to it without any conditions and no balancing between the queues. You shouldn’t even specify a binding key, and, if you do, the exchange will just ignore it. Furthermore, there is no limit to how many queues you can bind to the exchange.

As an example, say you have a website that sells shoes . When there is a sale, you want to notify all your customers. The problem is that some customers wish to have the information over email while others prefer to receive messages over SMS, and others still like it sent over a social media network.

You have three different workers with different skills; one knows how to send an email, one can send SMS, and one sends messages over the social network. They are all bound to a fanout exchange in RabbitMQ. All you need to do is to write your text about the sale and send that message to the fanout exchange. Queues bound to that exchange will get the message and send it out to your customers in their preferred way.

## What is RabbitMQ Direct Exchange?

The direct exchange will route the message to a queue whose binding key matches the routing key of the message exactly. So if you bind a queue to a direct exchange with the binding key “blue”, all messages published to that exchange with the routing key “blue” will end up in that queue. One queue can have multiple bindings to the same exchange with different binding keys.

Two different queues can also bind to the exchange with the same binding key, in that case when a message is published with that routing key the exchange will behave like the

## When to use RabbitMQ Direct Exchange

For example, you own a delivery company that delivers using a bicycle, motorcycle, or car. To trigger a delivery you post a message to one of the queues with an address. If you want a delivery using a bicycle you post a message to the direct exchange using the routing key “bicycle” and that message will get routed to the bicycle delivery queue. To generate an event log of all deliveries happening in the system, you add a log queue, and bind that to the same exchange but with three bindings of car, motorcycle, and bicycle. That way, you get all messages to that queue and can save that to the log.

## What is RabbitMQ Topic Exchange?

It is similar to the [direct exchange](https://www.cloudamqp.com/blog/rabbitmq-direct-exchange-explained.html)in that it will route messages where the routing key matches the binding key from the queue binding. However, with a topic exchange, you can also use wildcards in the binding key. When using a topic exchange the routing key of the message must be a list of words separated by dots, like “metrics.server.cpu”. The reason is that topic exchange allows you to match on parts of the routing key and uses dots as separators.

There are two types of wildcard you can use:

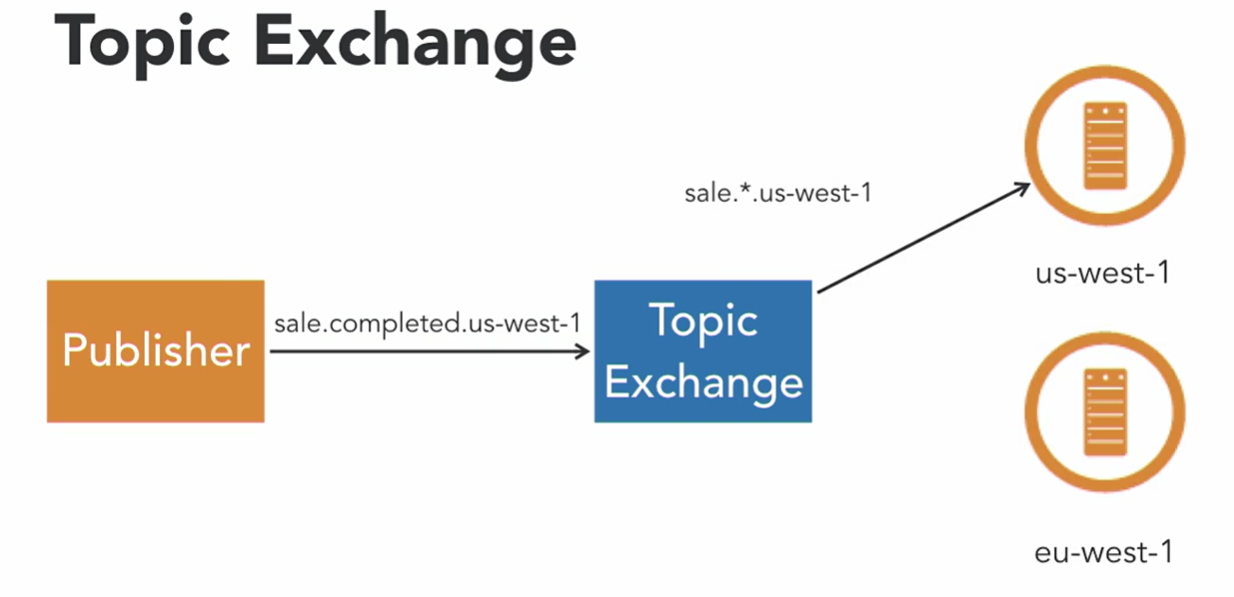
* # (hash) will match zero or more words, for example, “metrics.#” will match all routing keys that start with “metrics.”
* \* (star) will match one word, for example, “metrics.\*.cpu” will match all routing keys that start with “metrics.” and end in “.cpu”.

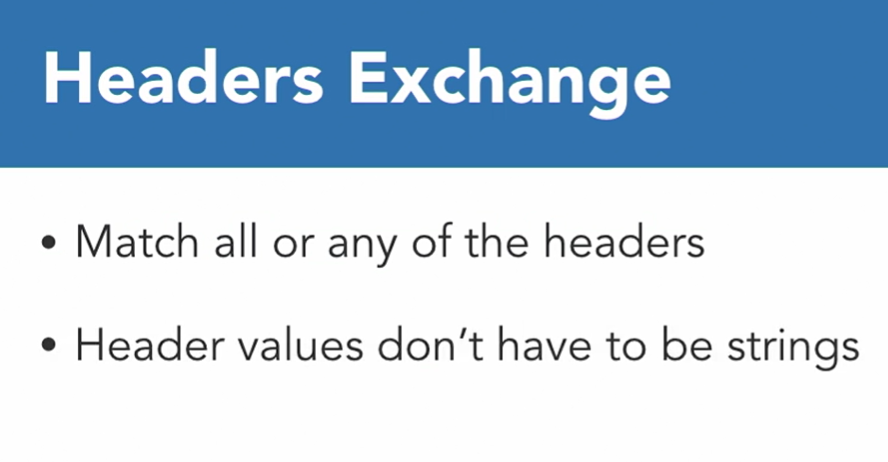
As you can see this exchange is very powerful and allows for a lot of use cases.

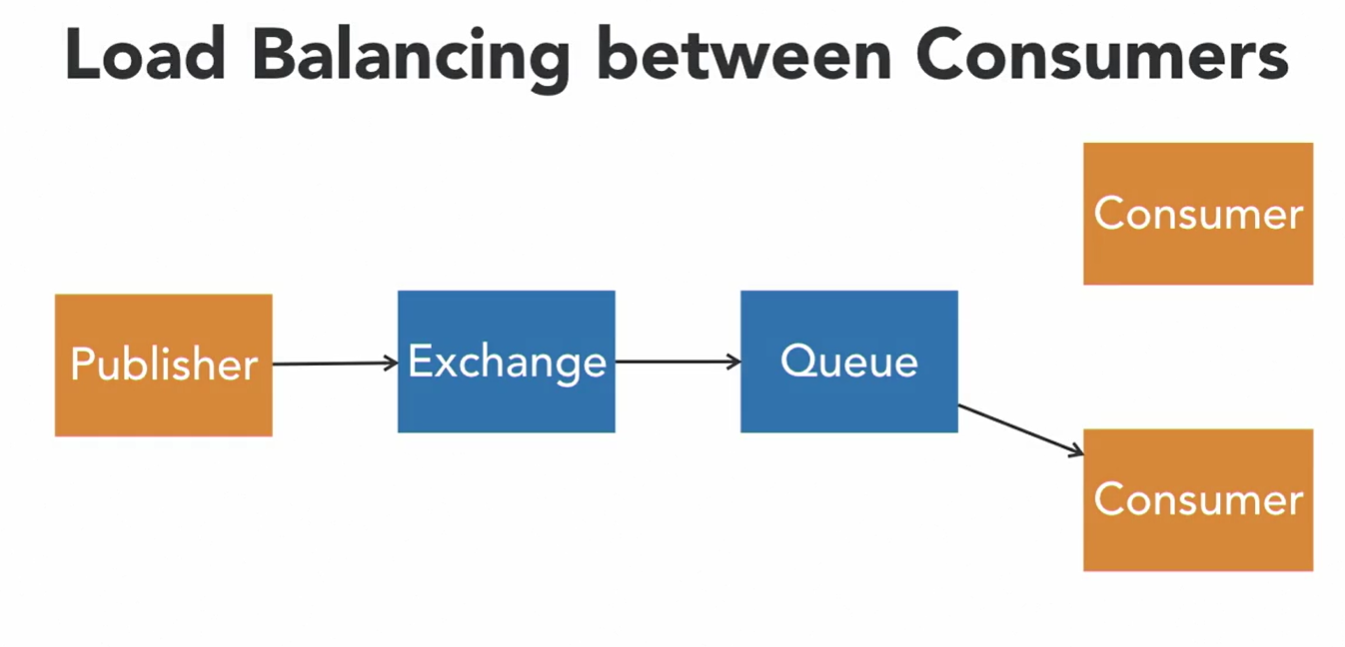
Things to think about: If you publish a message that doesn’t match any binding key, the message will be lost.

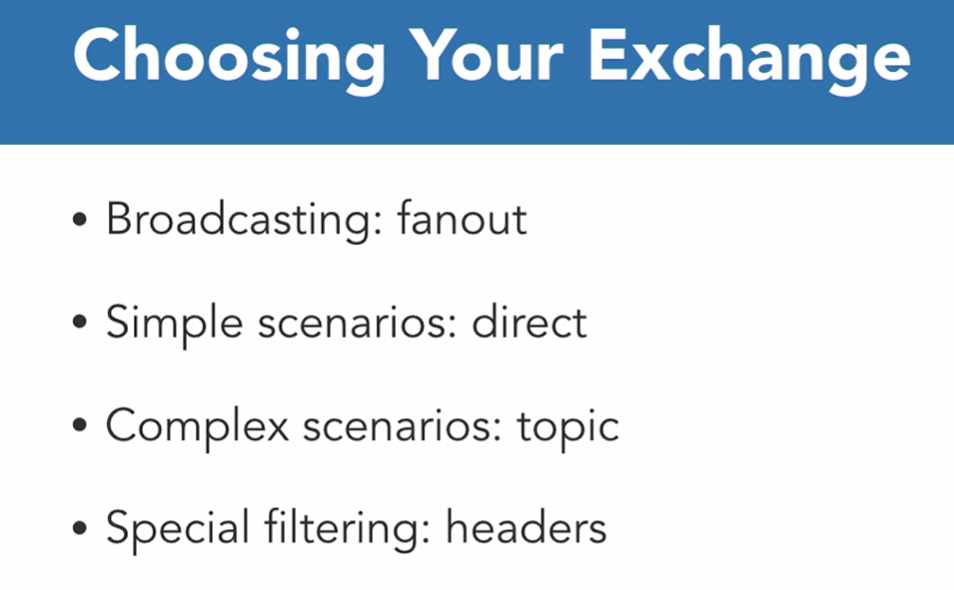
## When to not use the topic exchange

* You can use the binding key “#” to catch all messages published to a topic exchange, but that is not recommended and you should probably look at [fanout exchange](https://www.cloudamqp.com/blog/rabbitmq-fanout-exchange-explained.html)instead.
* If none of the binding keys uses any wildcards, the exchange behaves just like the [direct exchange](https://www.cloudamqp.com/blog/rabbitmq-direct-exchange-explained.html)so that might be a better choice.









**PUBLISH TO RABBITMQ EXCHANGE**

****