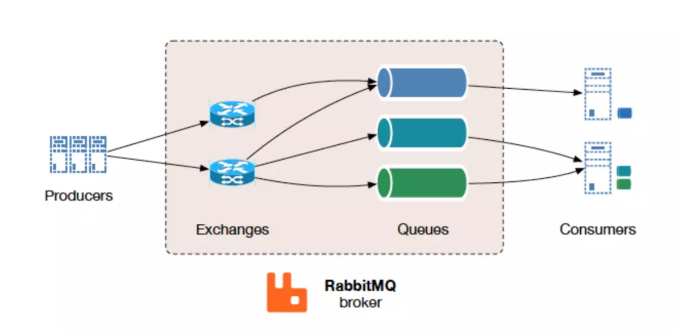
二者概述

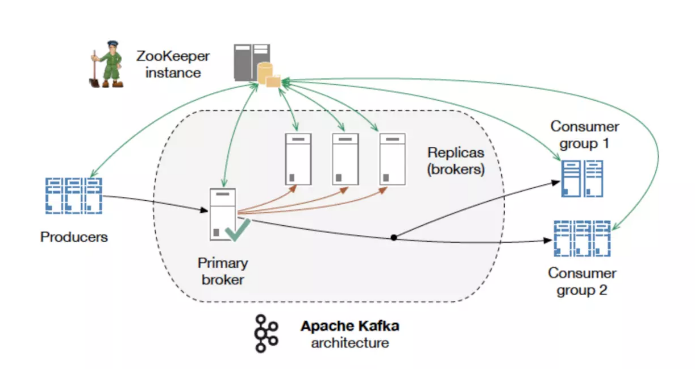
RabbitMQ：关于rabbit的详细介绍这里不说，感兴趣的可以看我之前的文章，一句话rabbit作为传统意义上的消息队列，基于AMQP协议开发，倾向于做按各种规则的消息转发。

Kafka：关于kafka的详细介绍会在以后的文章里写，因为刚开始用，想深入了解后再写出来。kafka更倾向于一个流式管道的概念，消息从一处流向另一处，吞吐量比rabbit更高。

接下来通过俩张图来理解他俩的设计与区别。



首先来看rabbit，他通过broker来进行统一调配消息去向，生产者通过指定的规则将消息发送到broker，broker再按照规则发送给消费者进行消费，消费者方可以选择消费方式为pull或者是broker主动push，支持的消费模式也有多种，点对点，广播，正则匹配等。



Kafka主要为高吞吐量的订阅发布系统而设计，主要追求速度与持久化。kafka中的消息由键、值、时间戳组成，kafka不记录每个消息被谁使用，只通过偏移量记录哪些消息是未读的，kafka中可以指定消费组来实现订阅发布的功能。

适用场景

了解了二者大体的区别以后，我们再来看具体的适用场景。

**Kafka：**

1.从A系统到B系统的消息没有复杂的传递规则，并且具有较高的吞吐量要求。

2.需要访问消息的历史记录的场景，因为kafak是持久化消息的，所以可以通过偏移量访问到那些已经被消费的消息（前提是磁盘空间足够，kafka没有将日志文件删除）

3.流处理的场景。处理源源不断的流式消息，比较典型的是日志的例子，将系统中源源不断生成的日志发送到kafka中。

**rabbit：**

1.需要对消息进行更加细粒度的控制，包括一些可靠性方面的特性，比如死信队列。

2.需要多种消费模式（点对点，广播，订阅发布等）

3.消息需要通过复杂的路由到消费者。

性能

最后是关于性能方面，众所周知，kafka的吞吐量优于rabbit，大约是100k/sec，而rabbit大约是20k/sec，但是这个不应该成为我们选择的主要原因，因为性能方面的瓶颈都是可以通过集群方案来解决的。

最后要说的是，没有最好的队列，只有最合适的队列。