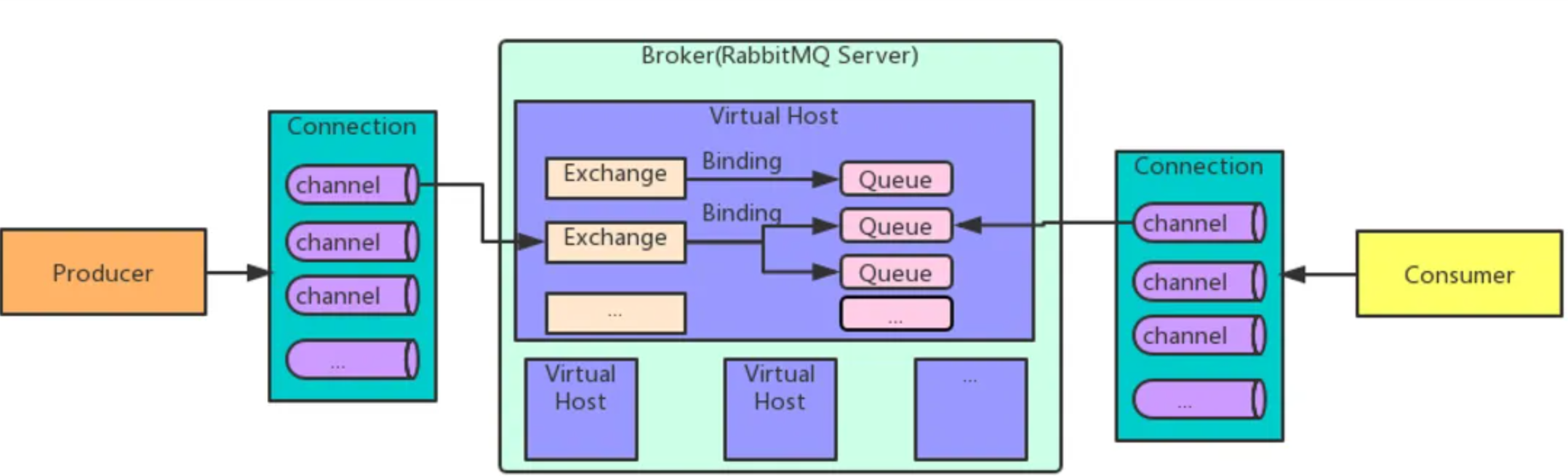
RabbitMQ



相关概念：

Connection：publisher／consumer 和 broker 之间的 TCP 连接。

Channel：Channel 是在 connection 内部建立的逻辑连接，作为轻量级的 Connection 极大减少了操作系统建立 TCP connection 的开销。

Broker：接收和分发消息的应用，RabbitMQ Server就是 Message Broker。

Exchange：message 到达 broker 的第一站，根据分发规则，匹配查询表中的 routing key，分发消息到queue 中去。

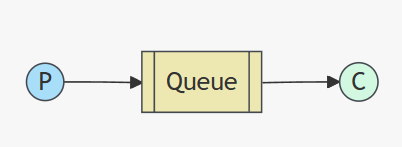
Queue：消息最终被送到队列这里等待 consumer 取走

Binding：exchange 和 queue 之间的虚拟连接，根据工作模式binding 中可以包含 routing key。

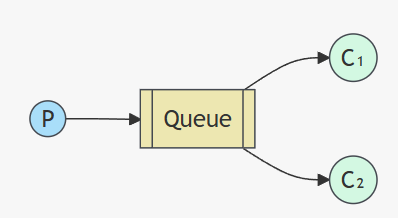
Virtual host：多个不同的用户使用同一个 RabbitMQ server 提供的服务时，可以划分出多个vhost，每个用户在自己的 vhost 创建 exchange／queue

工作模式：

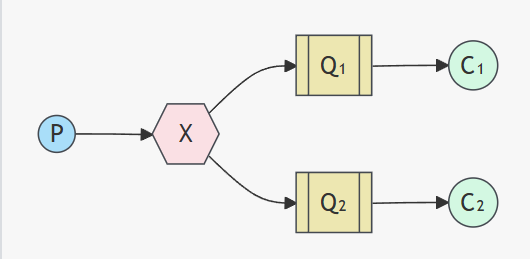
1.生产者消费者队列都只有一个，使用默认交换机



2.工作队列模式：多了一些消费端，使用默认交换机



3.Pub/Sub



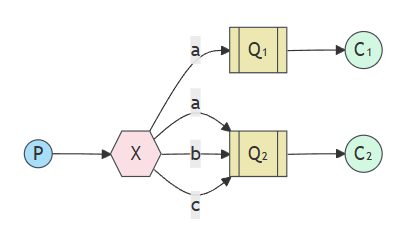
多了一个exchange，要把队列绑定到交换机上。exchange有常见以下3种类型：

Fanout：广播，将消息交给所有绑定到交换机的队列

Direct：定向，把消息交给符合指定routing key 的队列

Topic：通配符，把消息交给符合routing pattern（路由模式） 的队列

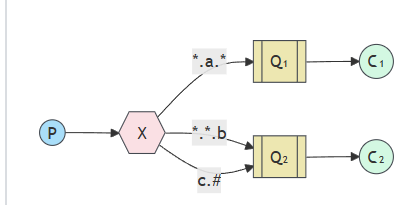
4. Routing路由模式



队列与交换机的绑定时，必须要指定一个RoutingKey。发送方在向 Exchange发送消息时，也必须指定消息的RoutingKey。Exchange根据消息的 Routing Key 进行判断

，队列的Routingkey与消息的Routing key完全一致，才会接收到消息。

5.Topic通配符模式。



在Routing路由模式的基础上，Exchange可以让队列在绑定Routing key的时候使用通配符。

RocketMQ-JMS介绍:

RocketMQ不支持 JMS 接口。rocketmq-jms 是 Apache RocketMQ 提供的 JMS接口实现，提供与 JMS 1.1 规范兼容的 API，但不支持JMS2.0.

rocketmq-jms存在的问题：

项目从8年前到现在只有初始版本，没有更新和维护。

项目没有文档可供参考，一些操作如创建ConnectionFactory时需要new的URI的格式需要翻看源码才知道。

项目对JMS的支持并不完全。如下图，在Session接口的实现中有许多方法仅仅抛出UnsupportedOperationException，并没有实际的实现。

