Java消息服务（java Message Server）应用程序接口是一个java平台中关于面向消息中间件的api，用于两个程序之间，或分布式系统中发送消息，进行异步通信。

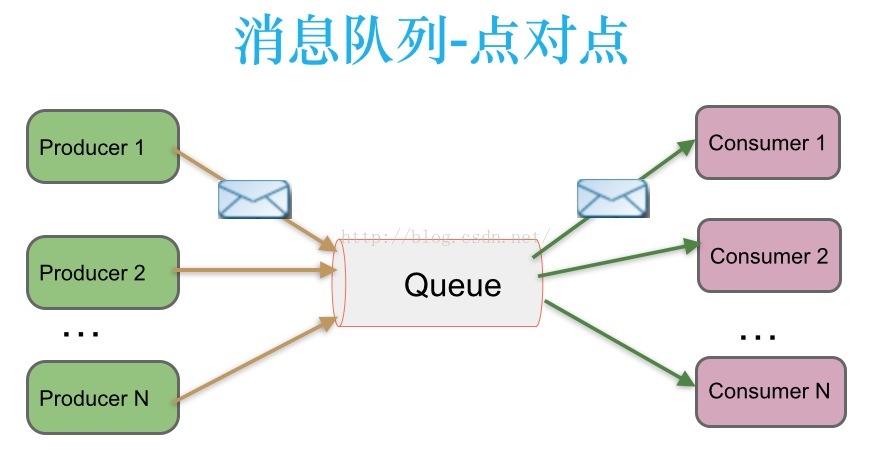
点对点和发布订阅最初是由jms定义的。这两种模式的主要区别就是发送到队列的消息是否可以重复消费。

定义：

Jms支持两种消息模型：点对点（point to point ,queue），发布订阅(publish/subscribe ,topic)

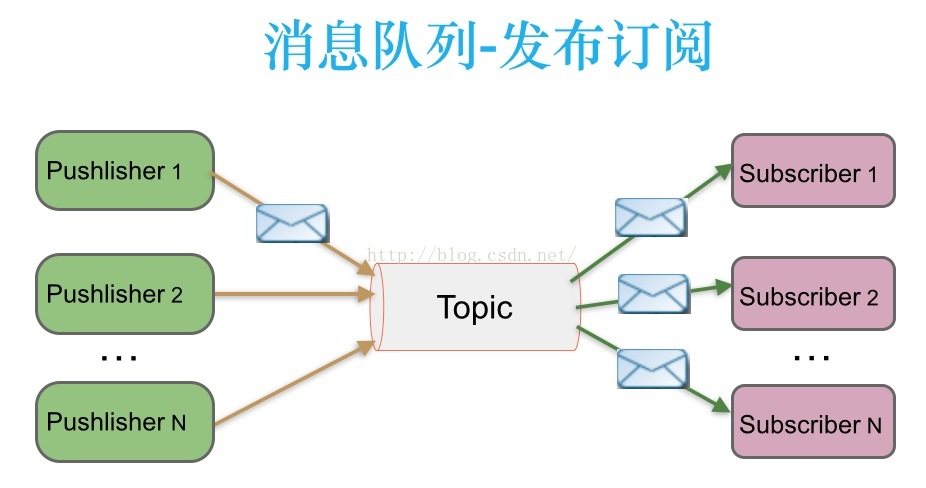
点对点 QUEUE 不可重复消费

消息生产者生产消息发送到queue之中，然后消息消费者从queue中取出消息，然后消费消息。消息消费之后，queue中不再有存储，所以消息消费者不可能消费到已经消费过的消息。**QUEUE支持有多个消费者，但是一条消息只能被一个消费者消费。**



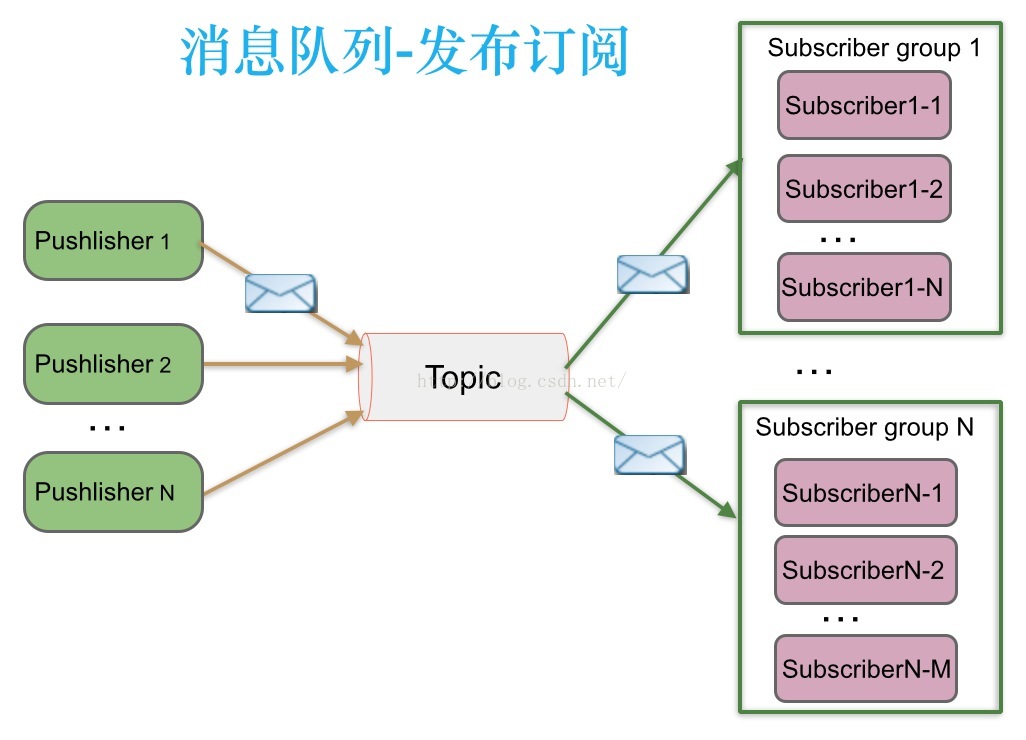
发布订阅 topic 可以重复消费

消息生产者将消息发布到topic之中，同时又多个消息消费者订阅消息。和点对点不同的是，发布到topic之中的消息会被所有订阅者消费。



支持订阅组的发布订阅模式

发布订阅模式下，当发布者消息量很大时，显然订阅者的处理能力是不足的。实际上现实场景中是多个多个订阅者节点组成一个订阅组负载均衡消费topic消息即分组订阅，这样订阅者很容易实现消费能力线性扩展。订阅组模式也可以理解成**点对点和发布订阅的组合。**



流行的模型比较

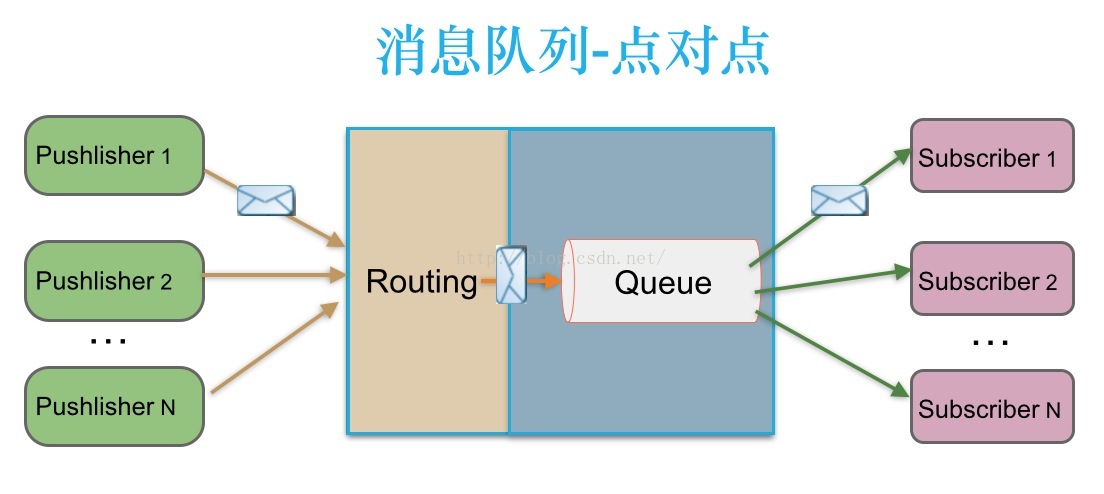
传统企业消息队列ActiveMq 遵循了jms规范，实现了点对点和发布订阅模型，但是其他流行的消息队列并没有遵循jms规范，如RabbitMq 和Kafka

RabbitMq

RabbitMq实现了aomp协议，aomp协议定义了消息路由规则和方式。生产端通过路由规则发送消息到不同的queue之中，消费端根据queue名称消费消息。

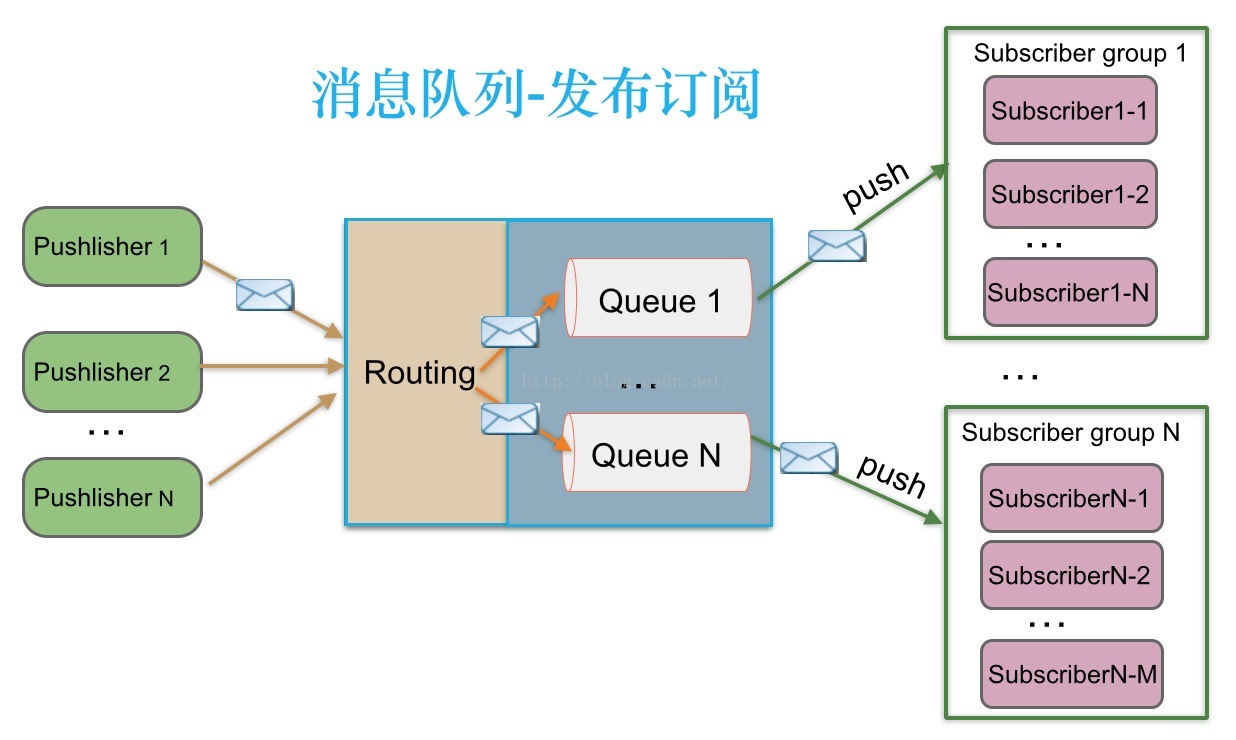
点对点

生产端发送一条消息通过路由投递到Queue，只有一个消费者能消费到。



多订阅

当rabbitmq需要多订阅的时候，发布者发布的消息通过路由规则同时写到多个queue之中，完成消息的拷贝，以支持多个订阅。



Kafka

Kafka只支持消息持久化，消费端为拉模型，消费状态和订阅关系由客户端负责维护，消息消费后不会立即删除，会保留历史消息，因此支持多订阅时，消息只用存储一份就可以了。但是可能产生重复消费的情况。

