# 什么是消息中间件

消息：

消息即为数据，数据就会有规则，有长度，有大小。

标准JMS（JSR914）：<https://www.jcp.org/en/jsr/detail?id=914>

Java Message Service简称JMS，为Java 程序提供一种通用的方式，来创建、发送、接收以及读取企业消息系统的消息。

JMS 元素

Provider提供方：服务提供者

Producer生产者：创建和发送JMS消息的客户端

Consumer消费者：接收JMS消息的客户端

Client客户端：生产或消费消息的应用&进程

Message消息：服务端与客户端之间的传输数据对象

Queue队列 ：包含待读取消息的准备区域（点对点）

Topic主题：发布消息的分布机制（发布&订阅）

JMS提供五种消息主体：

StreamMessage（流）、MapMessage（属性集合）、TextMessage（文本、字符串）、ObjectMessage、BytesMessage（字节）

## 中间件

为我们提供发送消息的程序或者服务.

主流有： RocketMq、 Kafka、ActiveMq（严格jms jsr914）、RabbitMq

# 为什么要用消息中间件



带来了**异步**、系统进行了**解耦**



# 什么是RocketMq

国内阿里开源 java语言写的。贡献给apache 4.2版本

简称Rmq并没有严格遵循jms规范（参考）。本身电商背景有关。

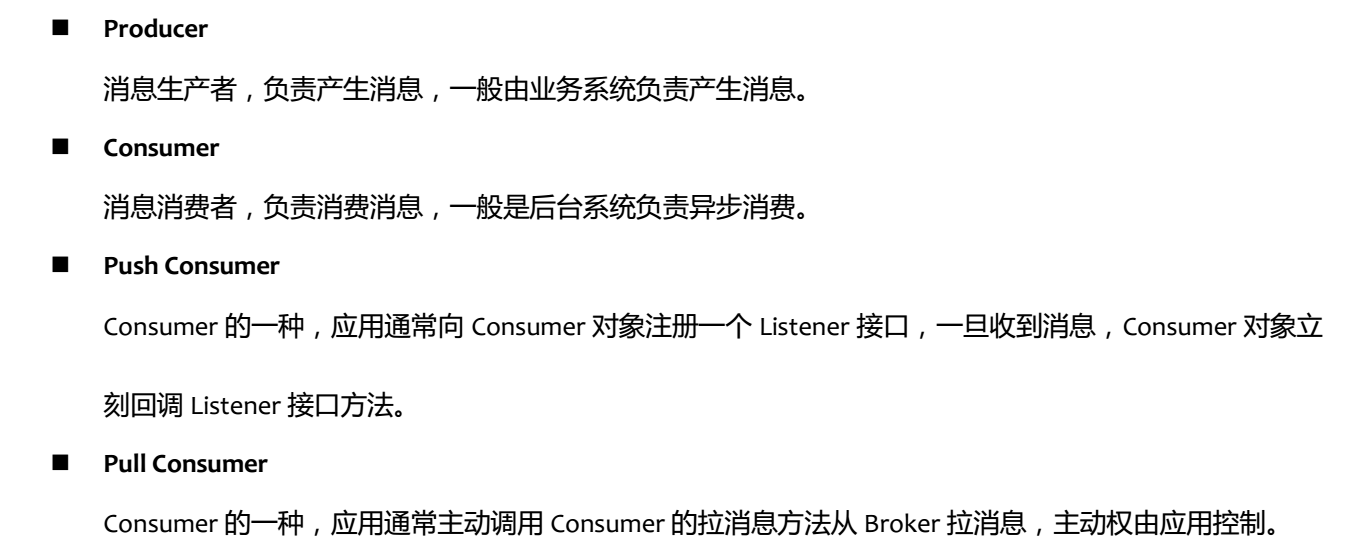
Java版本的kafka

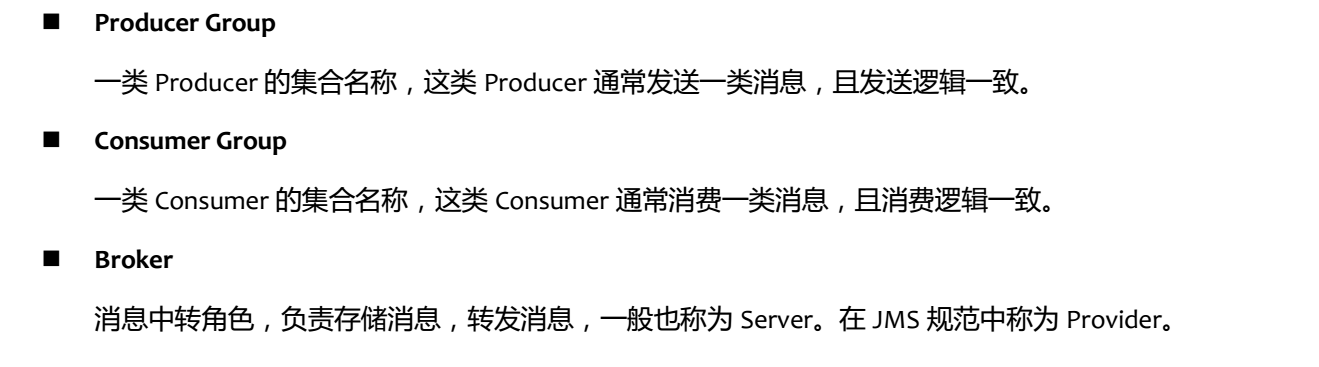
Storm（实时计算框架） jstrom（java写的）

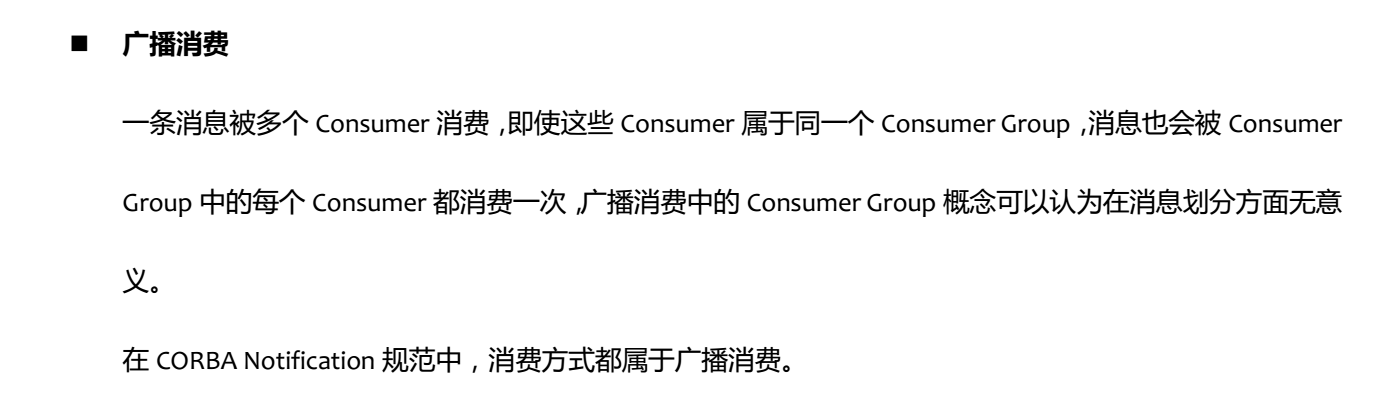
# 为什么选择RocketMq

1. 稳定无单点故障 不会因为某台机器或者某个节点宕机而影响系统正常使用
2. 集群功能完善
3. 经历过双十一
4. Java语言实现
5. 架构轻、源码可读性好
6. 生态圈完善，配套好
7. 开源社区活跃

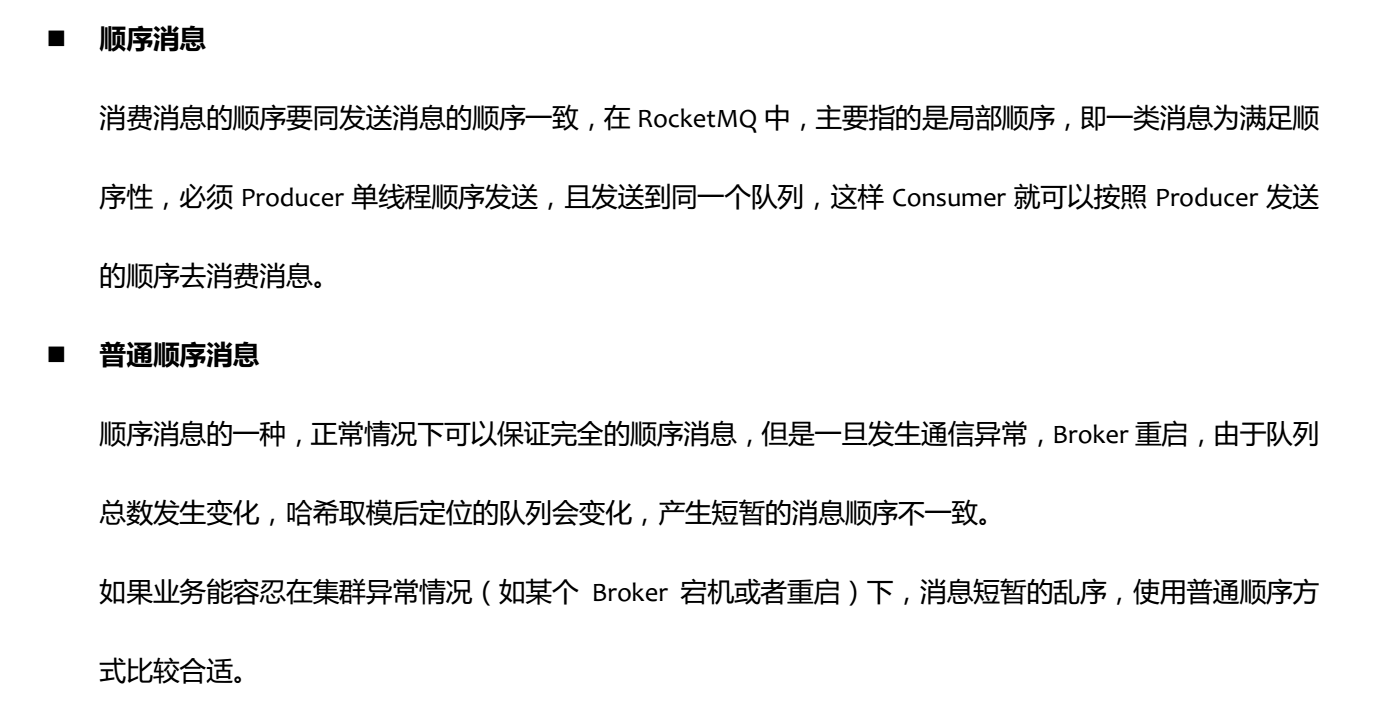
# Rocketmq术语：

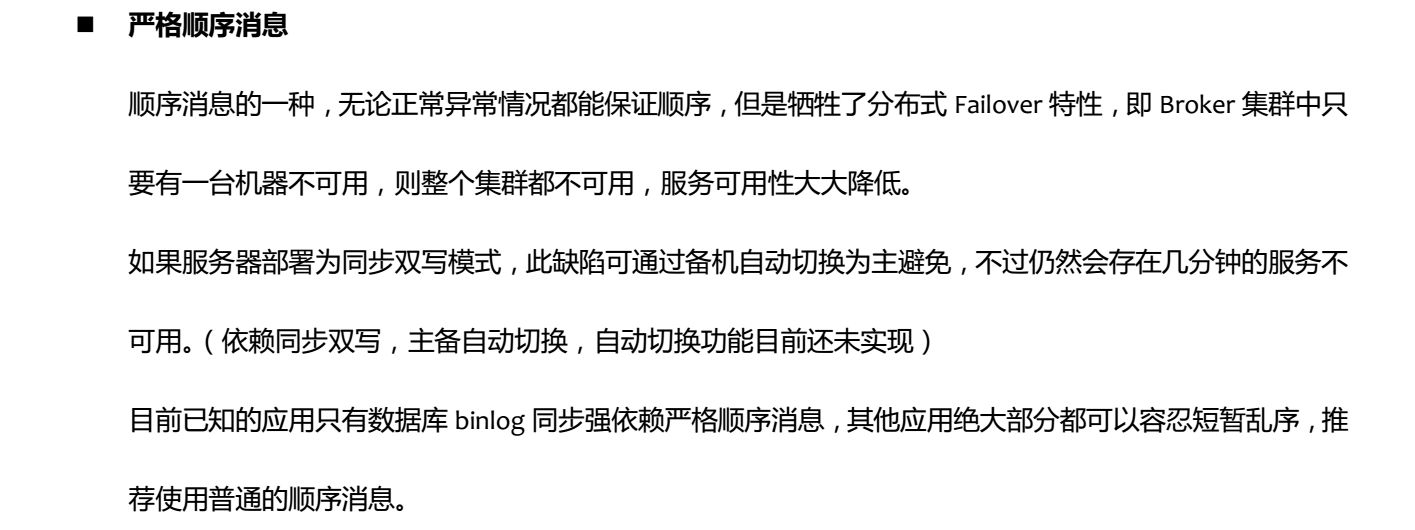


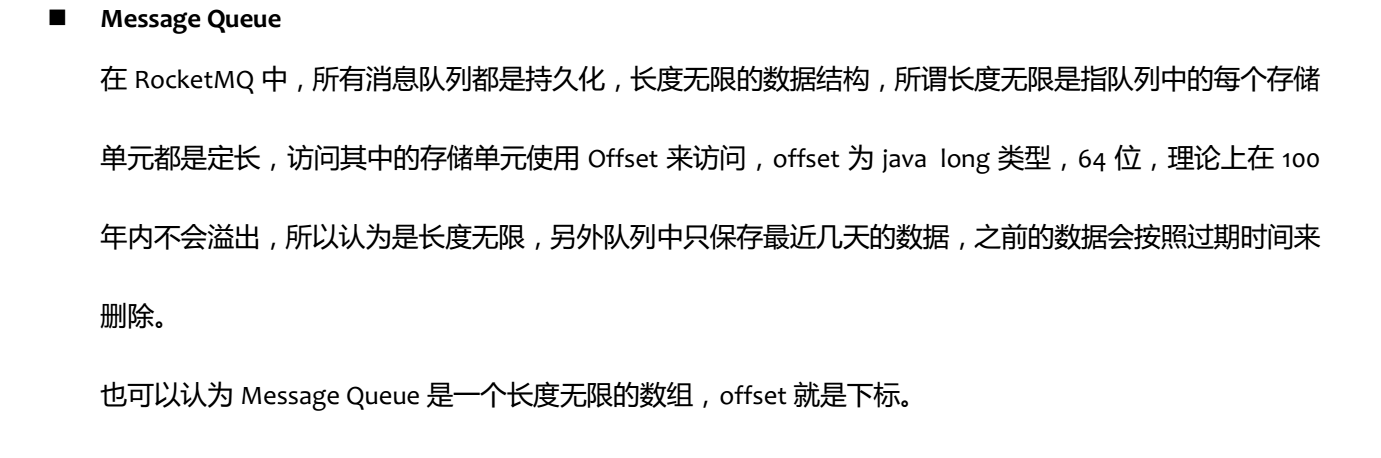






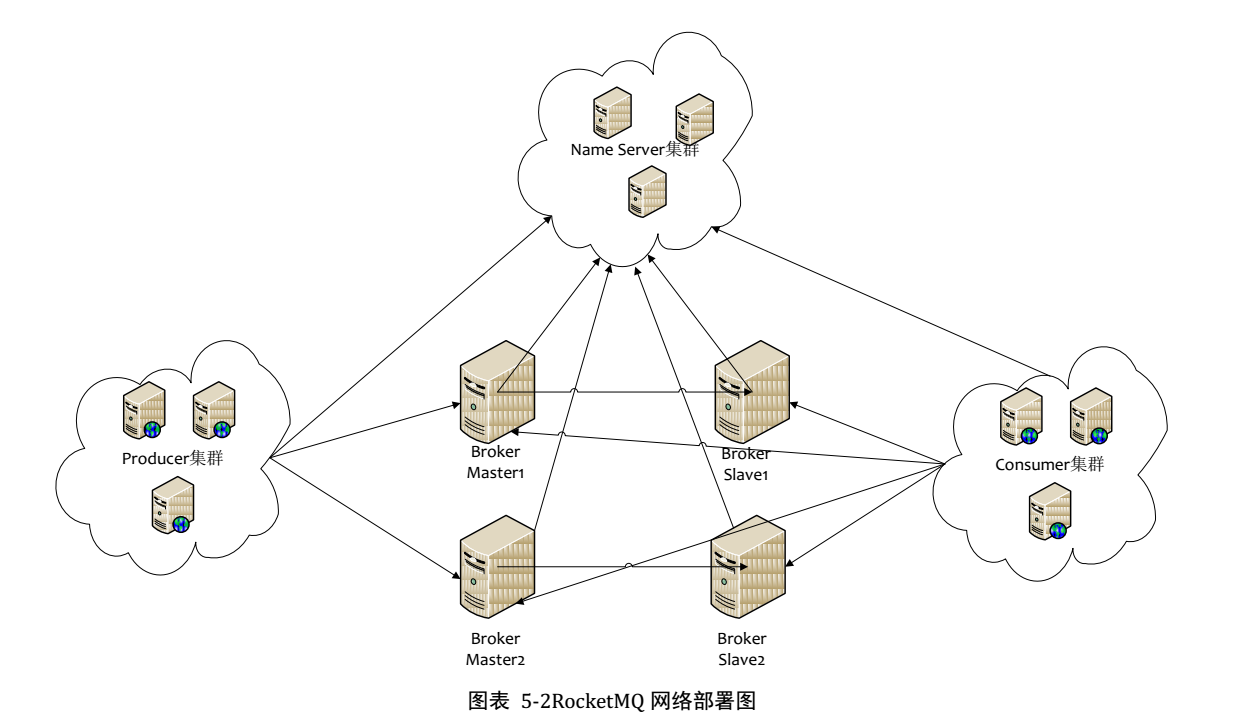






Group

# RocketMq部署图



Name Server：注册中心（zookeeper）频繁更新offset

Producer：消息生产者 生产消息 寄件人

Consumer：消息消费者、复制消息消费、收件人

Broker：中介（邮政） 提供消息中转服务

Group ：分组好处

Tag：多个标签 where

Key：区分业务系统

Msgid： broker在这个系统中它是独一无二的。

预习：rocketmq 4.x版本