消息系统：

概念：消息系统负责将数据从一个应用传递到另一个应用，应用只需要关注数据，无需 关注数据在两个应用间是如何传递的，分布式的消息传递基于可靠地消息队列， 在客户端应用和消息系统之间异步传递消息。

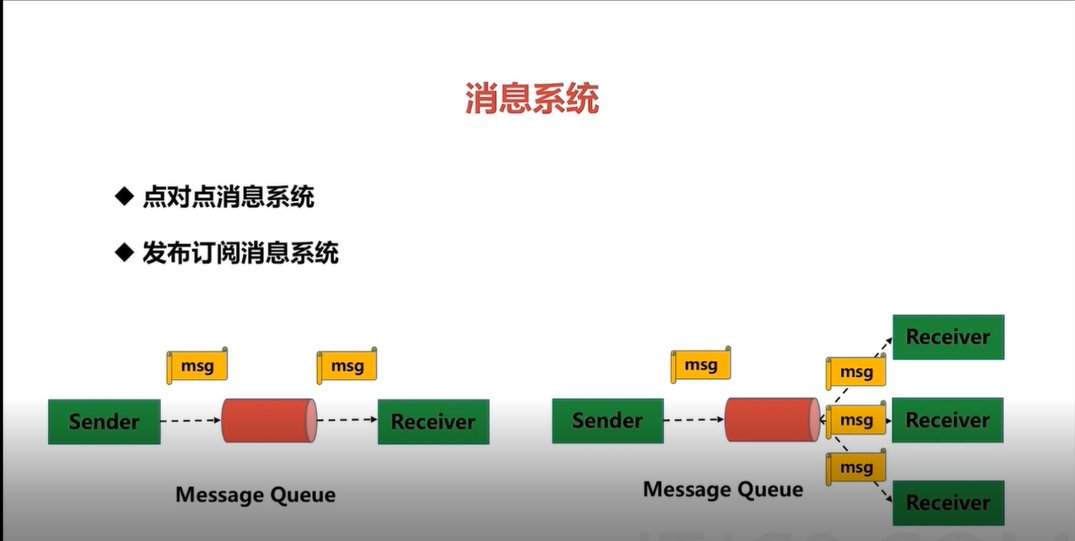
分类：

1. 点对点的消息系统

消息持久化到一个队列中，此时将有一个过多个消费者消费队列中的数据，但是一条消息只能被消费一次，当一个消费者消费了队列中的某条数据后，该数据将从消费队列中删除，此模式下，即使有多个消费者同时消费数据，也能够保证数据处理的顺序

1. 发布订阅消息系统

将数据持久化到一个topic中，与点对点不同的是，消费者可以订阅一个或多 个topic，消费者可以消费topic中所有的数据，同一条数据可以被多个消费者 消费，数据被消费后不会被立马删除，在发布订阅消息系统中，消息的生产者 称为发布者，消费者称为订阅者。



Kafka：

概念：apache的kafka是一个分布式的发布订阅消息系统，能够支撑海量数据的数据 传递，在离线和实时的消息处理业务系统中kafka都有广泛的应用，kafka会将 消息持久化到磁盘中，并对消息创建了备份，保证了数据的安全；kafka在保证 了较高的数据处理速度的同时，又能够保证数据处理的低延迟和数据的零丢失。

术语：producer、consumer、Topics、partition、kafka brokers

Topics：kafka中使用一个类别属性Topic来划分数据的所属类，如果把kafka看做 一个数据库，topic类似于数据库中的表，topic的名字也就是表的名字。

Partition:topic中的数据被划分为一个或多个partition，每个topic至少有一个 partition，每个partition中的数据使用offset？文件去存储，partition 中的数据是有序的，partition间的数据丢失了数据的顺序，即多个partition 间的数据可能是没有顺序的，如果topic有多个partition，那消费数据时 就不能保证数据的顺序，如果想要保证消息的严格消费顺序的场景，那么 就需要将partition的数字设为1

Offset：（图中0123），每一个producer产生的msg都会有一个当前partition下的 唯一的64字节的offset指明了该条消息的起始位置，也就是说0标识了第 一条消息的起始位置，1标识了第二条消息的起始位置。

Partition副本：即partition的备份，副本不会被消费者消费，只用于防止数据丢失， 消费者不会去kafka Brokers的副本中的partition中去消费数据

Kafka brokers：kafka的集群包括一个或多个服务器，服务器节点就被称为broker （集群中的每一台机器），broker存储了topic的数据，如果某个topic 有n个partition，集群中有n个broker，那么每个broker存储topic 的一个partition，主要是为了维护整个系统的负载均衡。

Producer：生产数据的发布者，将消息发布到kafka的topic中，broker接收到发布 者发布的消息之后，将该消息追加到该消息对应的segment？文件中，生 产者发布的消息自动存储到一个partition中（kafka默认分区策略），生产 者也可以指定消息存储的partition。

Consumer：消费生产者生产的数据，consumer可以从broker中读取消息数据，可 以消费多个topic中的数据，多个消费者组成了一个group，

