# MQ

## 你们公司生产环境用的是什么消息中间件？

Kafka,虽然未参与过选型，不过大概知道为什么。对比于ActiveMQ，RabbitMQ，RocketMQ，Kafka的优势在于专为超高吞吐量的实时日志采集、实时数据同步、实时数据计算等场景来设计。

## 为什么在你们系统架构中要引入消息中间件？

复杂系统的解耦

复杂链路的异步调用（将调用链路中操作比较耗时的模块抽出来，通过MQ调用）

瞬时高峰的削峰处理

分布式事物

## 如何保证消息中间件全链路数据100%不丢失

第一个会导致数据丢失的地方，就是消费者获取到消息之后，没有来得及处理完毕，自己直接宕机了。这就要调整消费者的ack机制，通过在消费者服务中调整为手动ack机制，来确保消息一定是已经成功处理完了，才会发送ack通知给MQ集群。

第二个是集群接收到消息后整个集群直接宕机了。这个就要确认只要订单服务发送出去的消息确认成功了，此时MQ集群就一定已经将消息持久化到磁盘了。

其次，增加一个额外的系统，发送端成功发送消息给MQ后，同时发送这条消息的id到这个额外系统，服务端处理完这条消息后，将这条消息的状态更新为已处理。

建立死信MQ系统，确保原有MQ系统正常运行。

## 说说系统架构引入消息中间件有什么缺点？

系统可用性降低（MQ挂掉之后的高可用保障方案）

系统稳定性降低（MQ系统故障导致丢数据，多数据等问题）

MQ中间件的这些问题的解决方案，包括：消息高可靠传递（0丢失）；消息幂等性传递（绝对不重复）；百万消息积压的线上故障处理；

分布式一致性问题（C现在处理自己本地数据库成功了，然后发送了一个消息给MQ，系统D也确实是消费到了，但是结果不幸的是，系统D操作自己本地数据库失败了。）