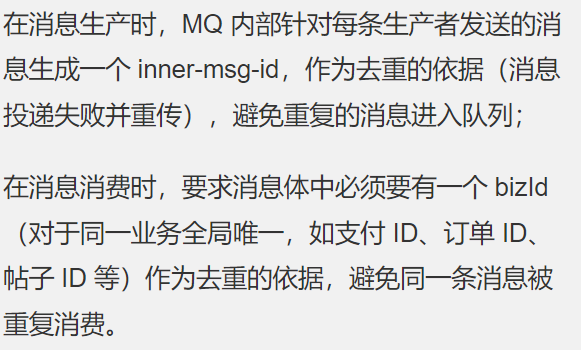
producer -> rabbitmq broker -> exchange -> queue -> consumer

消息从 producer 到 exchange 则会返回一个 confirmCallback

消息从 exchange 到 queue 投递失败则会返回一个 returnCallback

我们将以用这两个 callback 控制消息的可靠性投递

1：避免消息重复



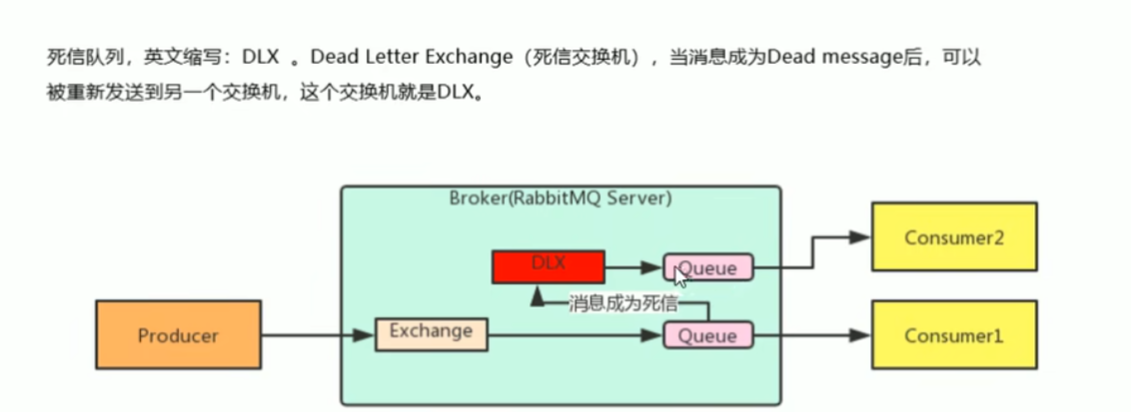
消息自身是有id的 重复的消息会被去重

消息消费的时候添加逻辑， 唯一id 订单id 防止重复消费；

2：防止消息丢失

消息持久化；

3： 死信队列

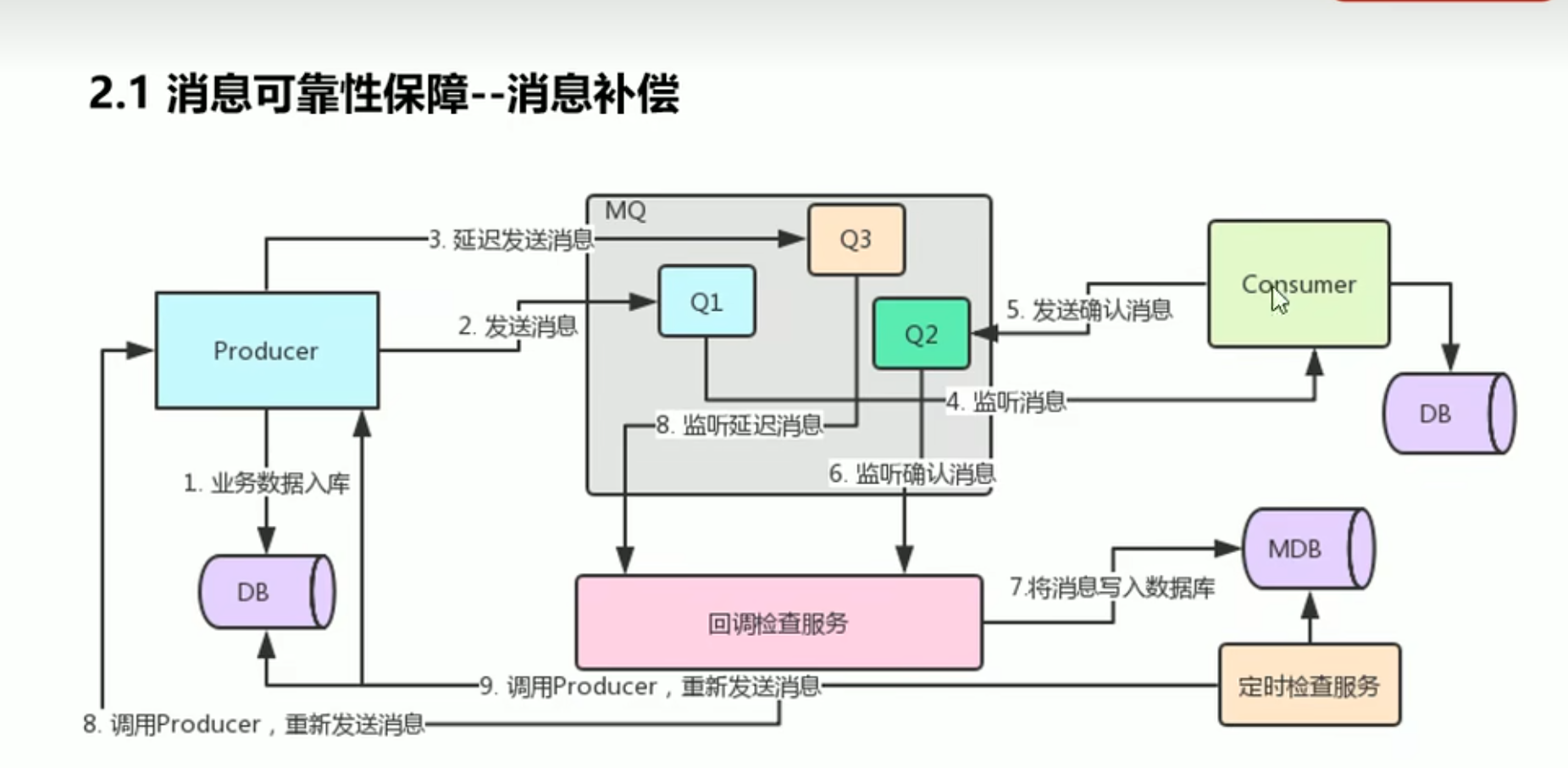


消息ttl到期并且没有消费，就发送给DLX

我们可以用这个特性做成支付订单；30分钟有效， 30分钟后，判定是否付款成功，如果成功

4：消息的可靠

为了保障消息是可靠的，怎么处理？



补偿机制的原理：

使用了三个队列，

pro 创建一个业务order数据 consumer 完成落地安排配送， 实现了业务；

描述 使用3个队列

pro 存order数据到数据库， consumer 存具体的发库之类的业务落地的数据；

pro利用死信队列延迟发送一条消息， 如果consumer成功 回调写入mdb（消息数据库）；

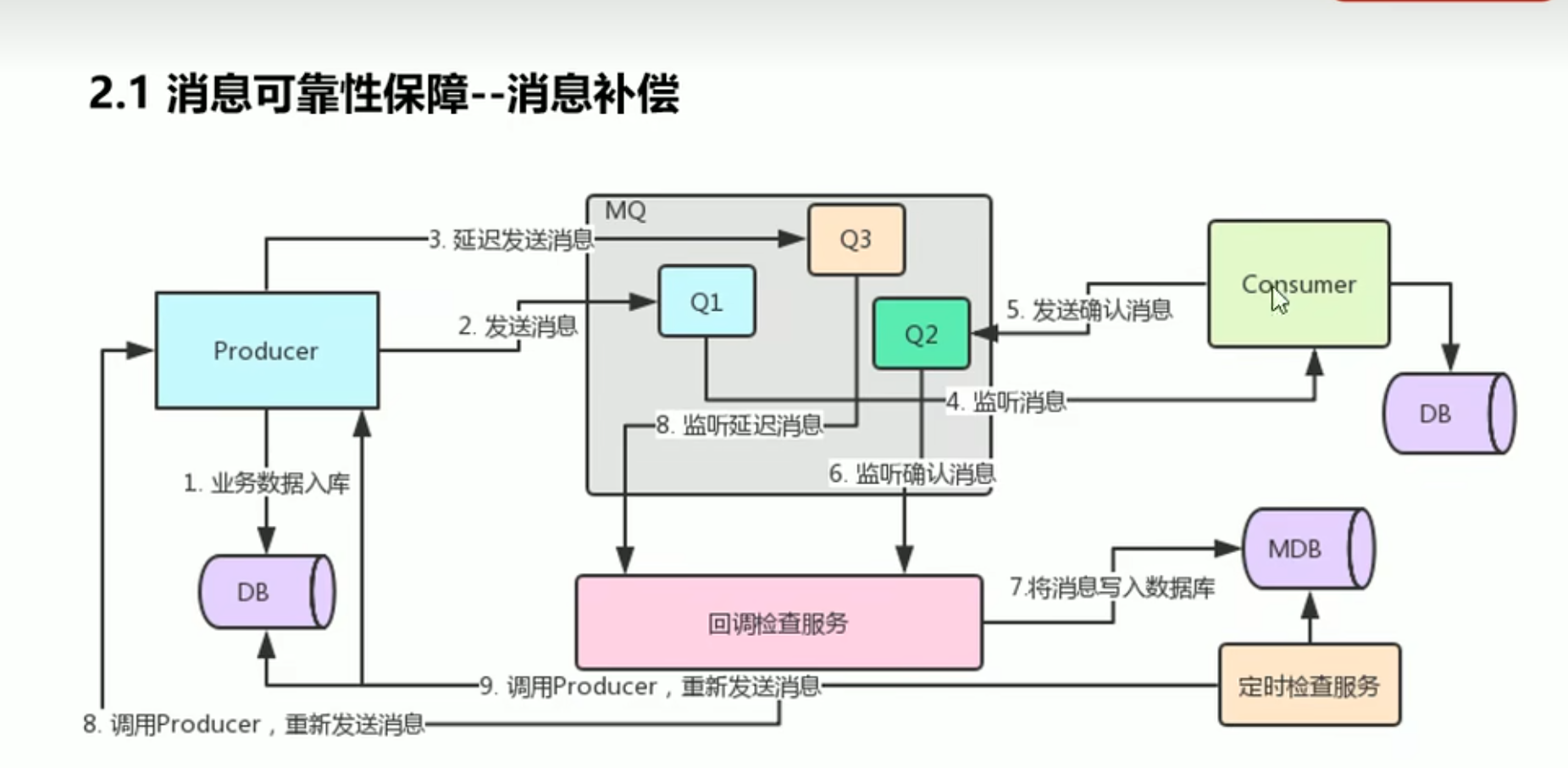
如果consumer失败， 则不写入mdb（）

延迟消息到了以后， 比对是否在mdb之中，不在就重新回调producer 重新执行逻辑，直到成功；

5：幂等性

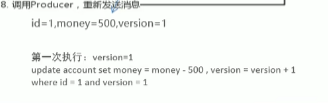
保障接口的幂等性：

乐观锁：



多次发送消息，对于相同的订单，之后的由于已经完成了， 所以不会重复调用；

乐观锁机制：

第一次 为1的 执行成功；

第二次： version已经是2了 不满足条件 所以失败；