

超级教程系列

微服务架构的分布式事务解决方案

分布式架构系统中，分布式事务是一个绕不过去的挑战！
微服务架构的流行，让分布式事务问题日益突出！

补偿型

最大努力
通知型

幂等性
操作

异步
确保型

定期
校对

可查询
操作

两阶段型

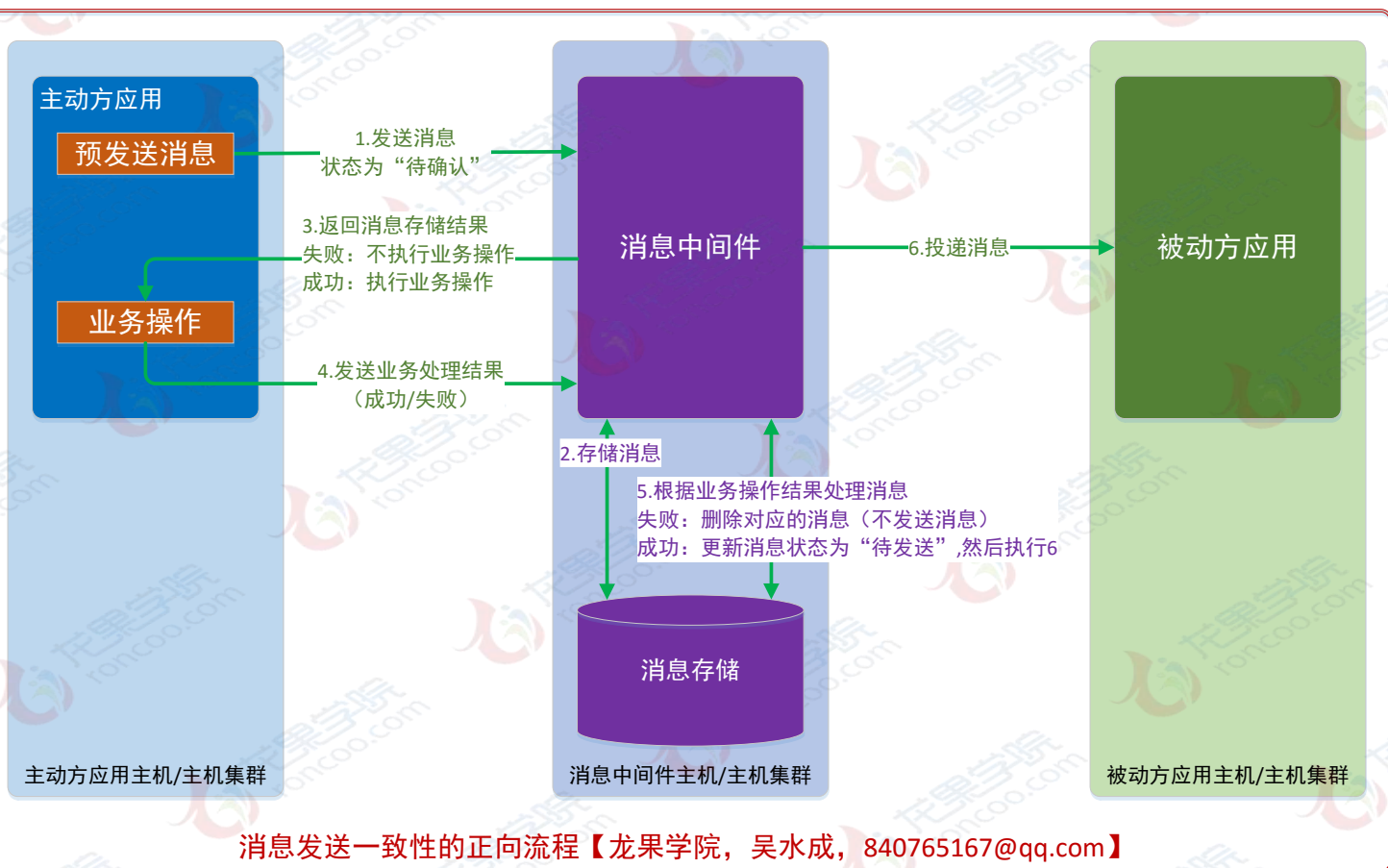


第05节

消息发送一致性的异常流程处理

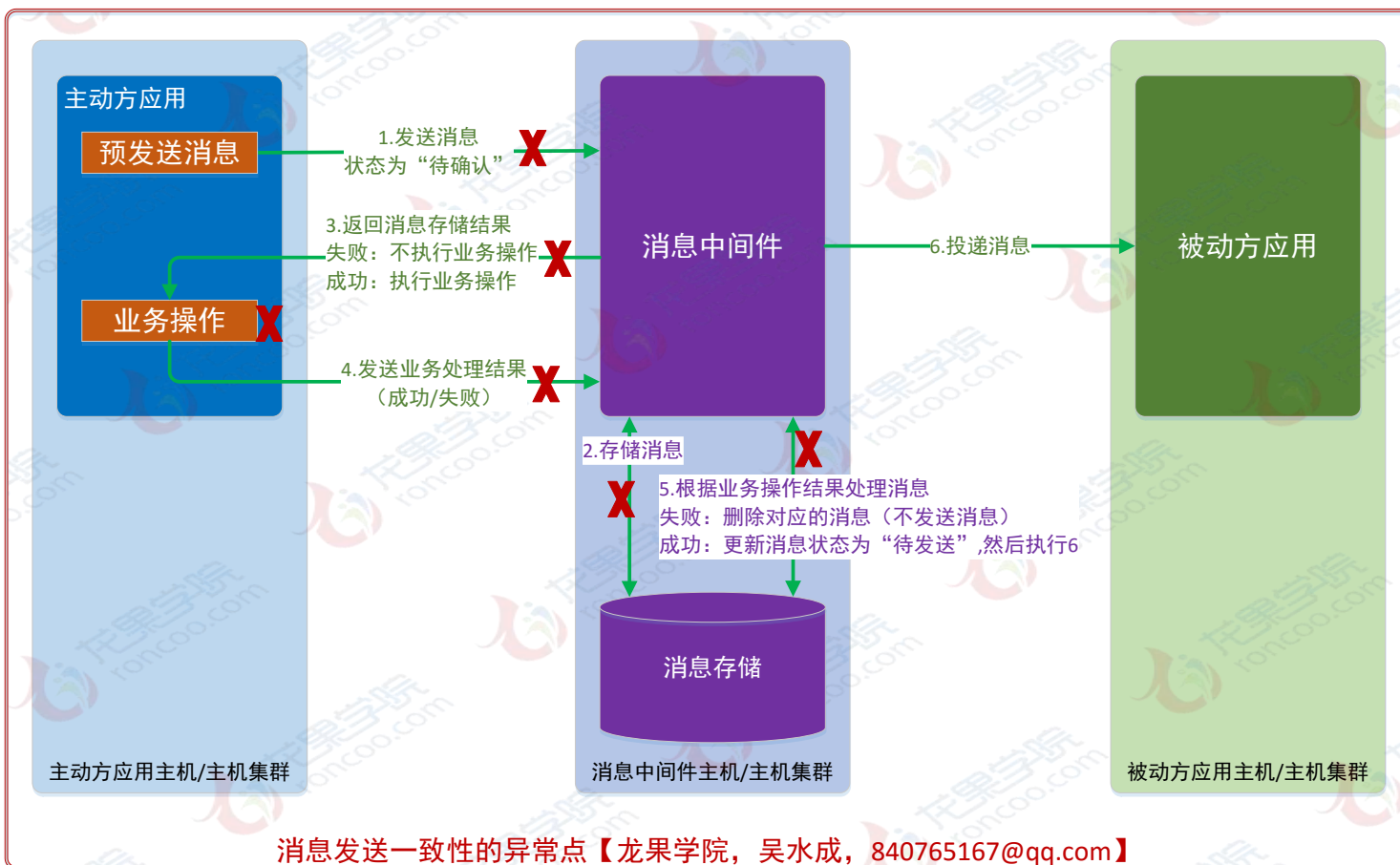
(基于可靠消息的最终一致性方案课程)

消息发送一致性的正向流程



1. 主动方应用先把消息发给消息中间件，消息状态标记为“待确认”；
2. 消息中间件收到消息后，把消息持久化到消息存储中，但并不向被动方应用投递消息；
3. 消息中间件返回消息持久化结果（成功/失败），主动方应用根据返回结果进行判断如何进行业务操作处理：
 - a) 失败：放弃业务操作处理，结束（必要时向上层返回失败结果）；
 - b) 成功：执行业务操作处理；
4. 业务操作完成后，把业务操作结果（成功/失败）发送给消息中间件；
5. 消息中间件收到业务操作结果后，根据业务结果进行处理：
 - a) 失败：删除消息存储中的消息，结束；
 - b) 成功：更新消息存储中的消息状态为“待发送（可发送）”，紧接着执行消息投递；
6. 前面的正向流程都成功后，向被动方应用投递消息；

消息发送一致性流程中的异常点

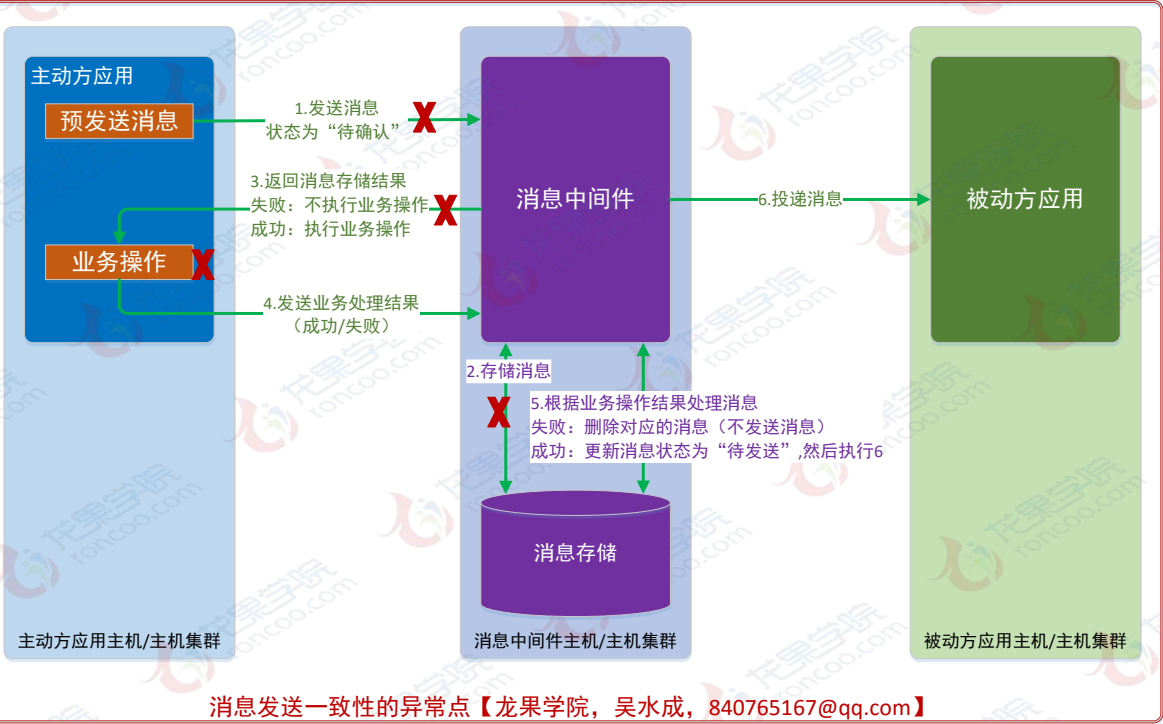


任何一个环节都可能会出问题!

消息发送一致性的异常情况分析

1、从主动方应用的角度来分析：

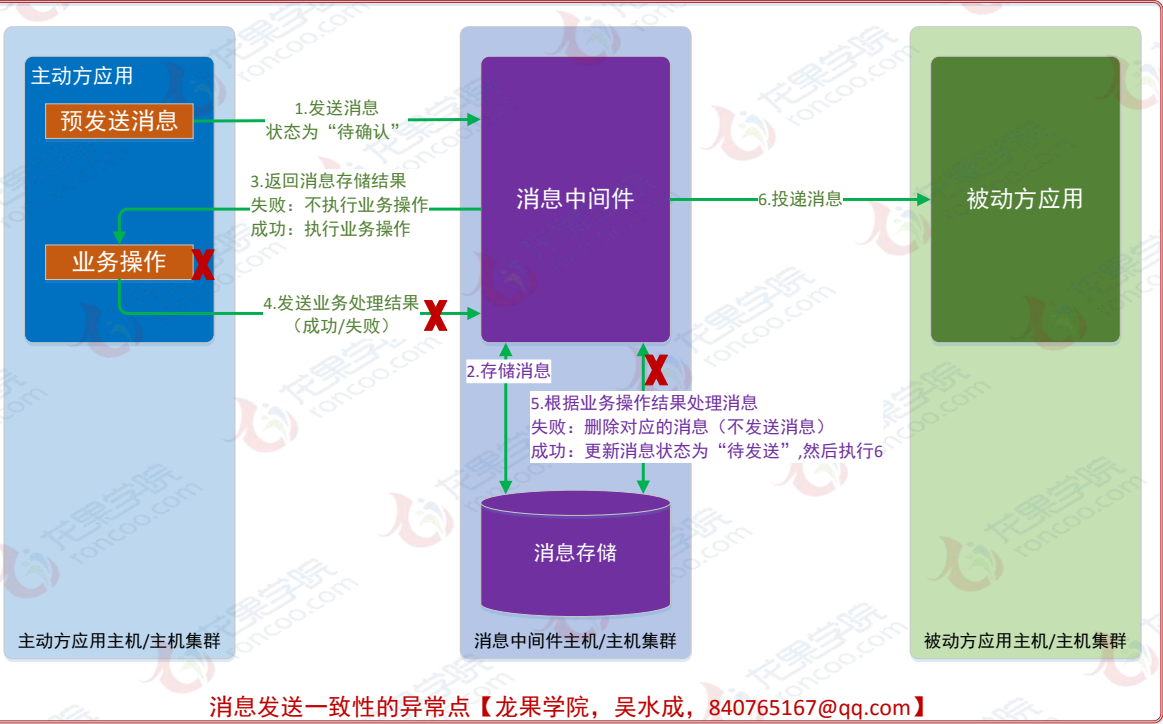
异常情况	可能的状态	一致性
预发送消息失败	消息未进存储，业务操作未执行（可能的原因：主动方应用、网络、消息中间件、消息存储）	一致
预发送消息后，主动方应用没有收到返回消息存储结果	(1)消息未进存储，业务操作未执行	一致
	(2)消息已进存储（待确认），业务操作未执行	不一致
收到消息存储成功的返回结果，但未执行业务操作就失败	消息已进存储（待确认），业务操作未执行	不一致



消息发送一致性的异常情况分析

2、从消息中间件的角度来分析：

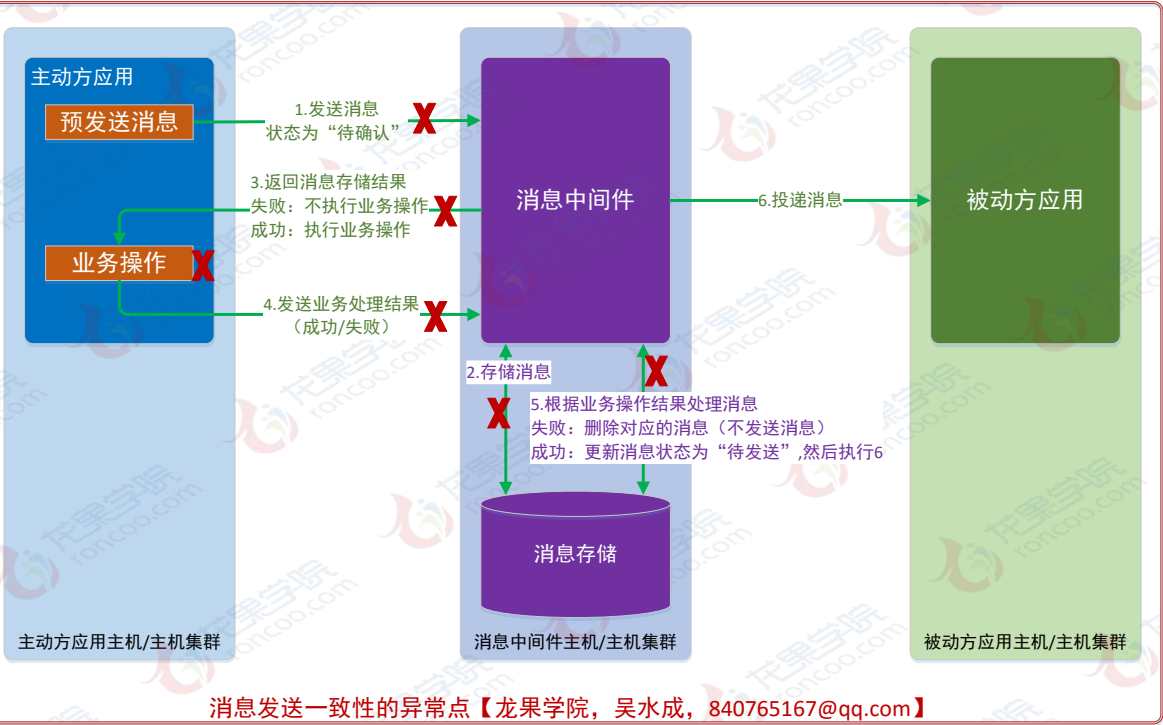
异常情况	可能的状态	一致性
消息中间件没有收到主动方应用的业务操作处理结果	(1)消息已进存储（待确认），业务操作未执行（或业务操作出错回滚了）	不一致
	(2)消息已进存储（待确认），业务操作成功	不一致
消息中间件收到业务操作结果（成功/失败），但处理消息存储中的消息状态失败	(1)消息已进存储（待确认），业务操作未执行（或业务操作出错回滚了）	不一致
	(2)消息已进存储（待确认），业务操作成功	不一致



消息发送一致性的异常情况总结

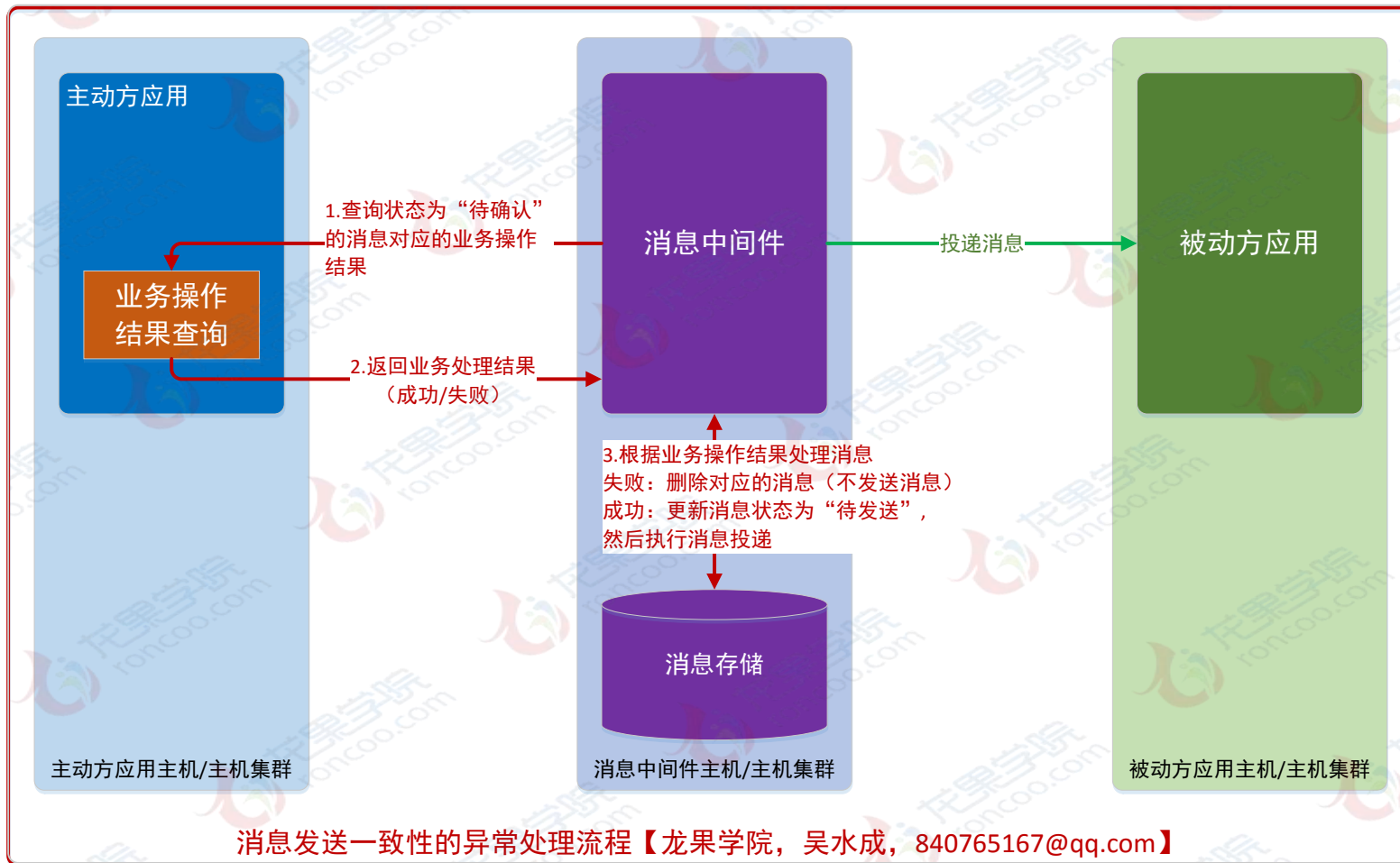
异常情况总结及处理

异常情况	一致性	异常处理方法
消息未进存储，业务操作未执行	一致	
消息已进存储（状态待确认），业务操作未执行	不一致	确认业务操作结果，处理消息（删除消息）
消息已进存储（状态待确认），业务操作成功	不一致	确认业务操作结果，处理消息（更新消息状态，执行消息投递）



消息发送一致性的异常处理

➤ 异常处理流程也有可能发生异常，又该怎么处理呢？



谢谢

THANK YOU

技术支持：Along、Hugo、Peter



龙果学院官方微信公众号