

SOFAStack

消息队列 DMS 技术白皮书

产品版本：AntStack Plus 1.13.1


文档版本：20230708

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

 蚂蚁集团
ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.什么是消息队列	05
2.产品优势	06
3.产品架构	07
3.1. 组件角色说明	07
3.2. 消息服务端架构	07
3.2.1. 消息接收与投递	07
3.2.2. 多租户设计	08
4.性能指标	10
5.功能原理	11
6.附录：基础术语	15

1. 什么是消息队列

消息队列，作为一种典型的消息代理组件（Message Broker），是企业级应用系统中常用的消息中间件，主要应用于分布式系统或组件之间的消息通讯，提供具有可靠、异步和事务等特性的消息通信服务。应用消息代理组件可以降低系统间耦合度，提高系统的吞吐量、可扩展性和高可用性。

分布式消息服务主要涉及五个核心角色，消息发布者（Publisher）、可靠消息组件（MsgBroker）、消息订阅者（Subscriber）、消息类型（Message Type）和订阅关系（Subscription），具体描述如下：

- 消息发布者（Publisher）：指发送消息的应用系统，一个应用系统可以发送一种或者多种消息类型，发布者发送消息到可靠消息组件。
- 可靠消息组件（MsgBroker）：负责接收发布者发送的消息，根据消息类型和订阅关系将消息分发投递到一个或多个消息订阅者。整个过程涉及消息类型校验、消息持久化存储、订阅关系匹配、消息投递和消息恢复等核心功能。
- 消息订阅者（Subscriber）：指订阅消息的应用系统，一个应用系统可以订阅一种或者多种消息类型，消息订阅者收到的消息来自可靠消息组件 (MsgBroker)。
- 消息类型（Message Type）：消息类型由 TOPIC 和 EVENT CODE 唯一标识。
- 订阅关系（Subscription）：用来描述一种消息类型被订阅者订阅。

2. 产品优势

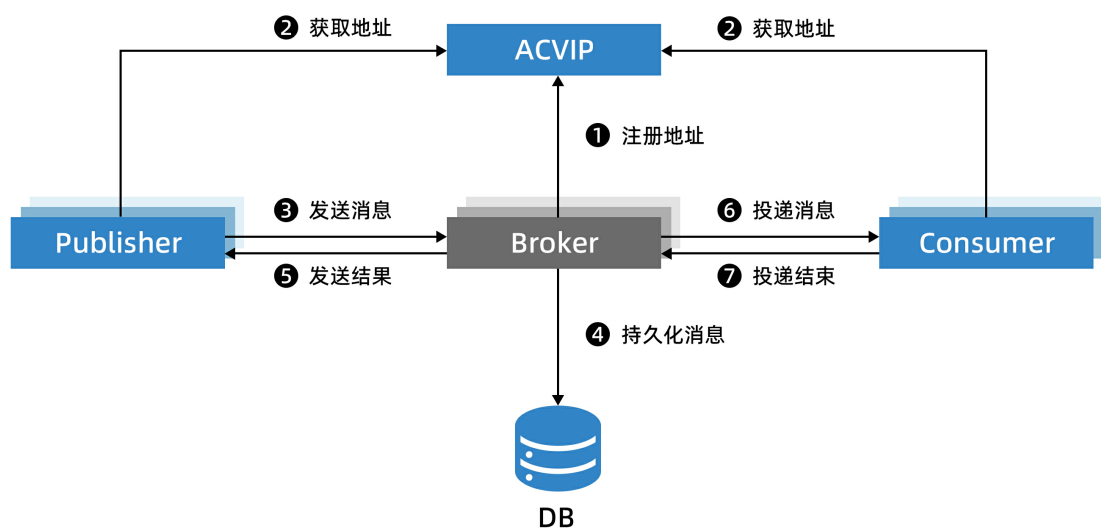
- 可为不同应用系统间提供可靠的消息通信，降低系统间耦合度并提高整体架构的可扩展性和可用性。
- 可为不同应用系统间提供异步消息通信，提高系统吞吐量和性能。
- 发布者系统、消息代理组件以及订阅者系统均支持集群水平扩展，可依据业务消息量动态部署计算节点。
- 支持事务型消息，保证消息与本地数据库事务的一致性。

3. 产品架构

3.1. 组件角色说明

消息服务的整体架构主要包括以下组件：

- 消息中心（Broker）：消息存储和转发。
- 发布者（Publisher）：发送消息到消息中心。
- 订阅者（Consumer）：接收消息中心的消息。
- 持久化数据库（DB）：持久化消息存储。
- 服务发现路由（ACVIP）：发布者和订阅者从 ACVIP 获取消息中心的地址，并和消息中心建连。



下面以一条消息的收发流程介绍消息中心的整体流程：

1. 消息中心的地址按照租户的维度注册到 ACVIP。
2. 发布者或订阅者启动时，根据租户信息，从 ACVIP 获取消息中心的地址。
3. 发布者会选择消息中心列表，轮询发送消息到消息中心。
4. 消息中心会先将消息持久化到数据库中，保证消息不丢失。
5. 消息中心返回发送结果给发送者。
6. 消息中心会立即将消息投递（推送）给订阅者。
7. 订阅者处理业务逻辑，告诉消息中心投递结果。
8. 消息中心根据投递结果决定是否删除消息，如果投递成功，则删除消息。

3.2. 消息服务端架构

3.2.1. 消息接收与投递

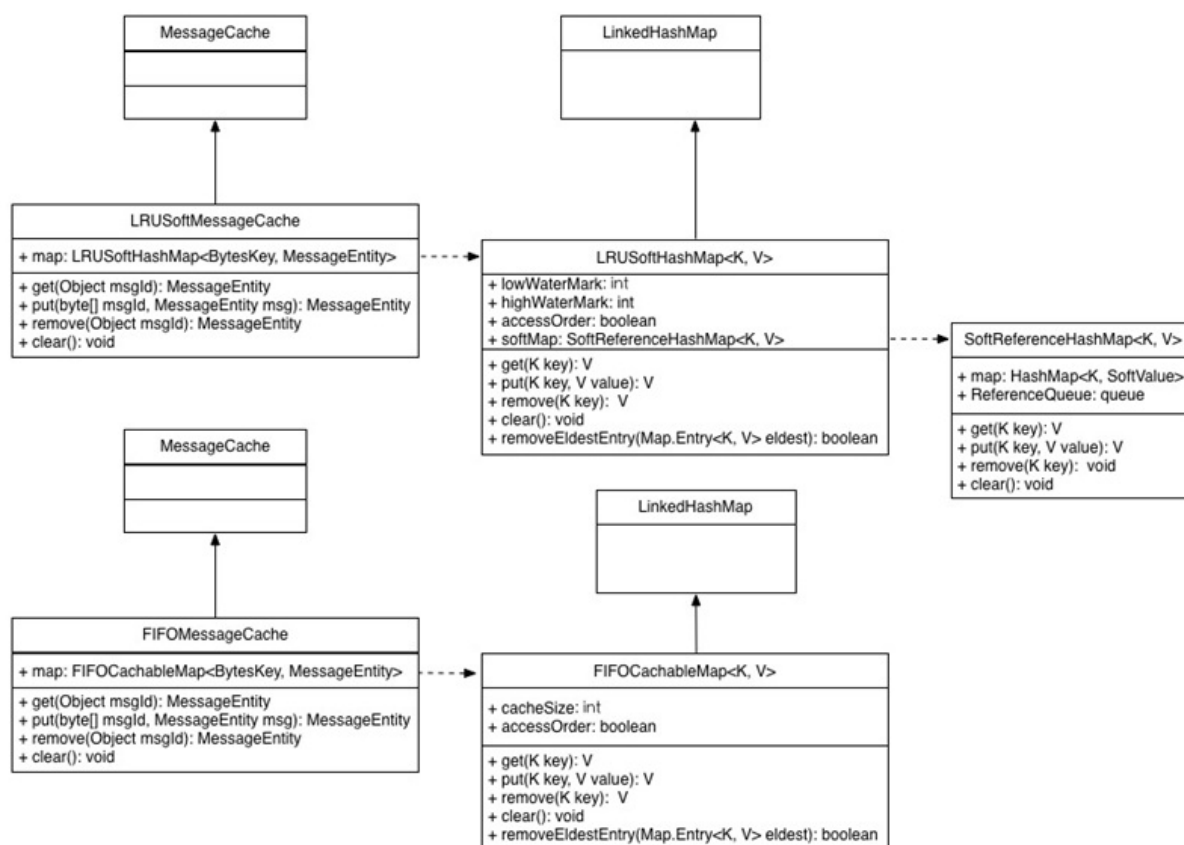
发送端发送的一条消息会先被消息中心持久化存储在数据库中，消息中心在投递消息给消费端之后，会将消息删除掉。消息中心支持多种消息存储方案，可以满足不同的可靠性需求：

- Memory 存储，关闭即丢失消息，最低可靠级别。
- Store4j 存储，store4j 是 Notify 开发的一个磁盘存储引擎，可以将消息存储在本地磁盘，可靠性一般，不支持事务。
- MySQL 存储，将消息存储在 MySQL 上，投递完成即删除。
- Oracle 存储，将消息存储在 Oracle 数据上，投递完成不删除，而是不断切换表，即 truncate 方案。Oracle 不删除数据的主要原因是为了避免删除产生的大量日志。
- HA MySQL 存储，即双写方案，先写 slave，再写 master，投递完成即删除。

为了避免对 DB 进行频繁查询，保持性能稳定，消息中心引入了消息缓存机制，每次取消息时先尝试从缓存中取，没有获取到结果再查询db。

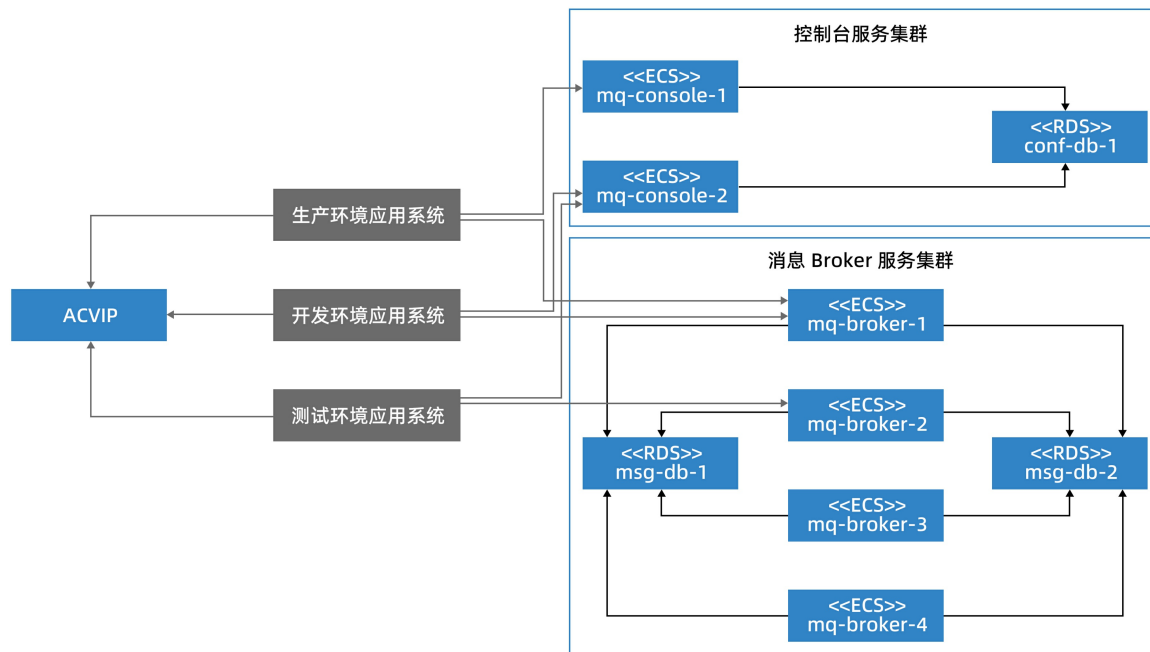
在消息持久化到 DB 之前，会首先将消息放至缓存中，因此消息放至缓存的 TPS 等同于消息发送的 TPS；当消息发送失败，消息中心会使用定时投递或者消息恢复机制来保证消息的重新投递，重新投递时会首先根据消息 ID 尝试从缓存中取出消息，只有当缓存没命中时，才会执行 DB 查询。因此，从缓存中获取消息的 TPS 与消息堆积正相关。

目前消息中心主要实现了两种缓存算法，分别为 LRU 和 FIFO，都是基于 LinkedHashMap 实现，类图如下所示：



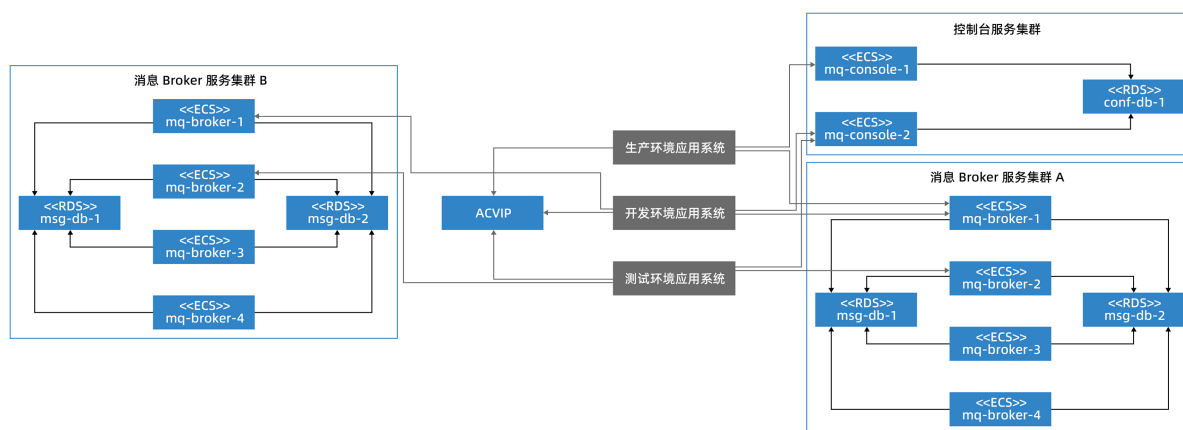
3.2.2. 多租户设计

消息队列产品默认提供共享服务集群，包括共享计算资源和存储资源，用户不同环境的元数据和消息流量数据逻辑隔离，但共享底层资源。共享服务集群的示意图如下：



- 消息 Broker 集群：负责接收用户不同环境的消息发布和消息订阅流量，当测试环境发布大量消息时，可能会影响生产环境的服务质量，因为计算单元和存储单元都是共享的。
- 控制台服务集群：负责维护不同环境的元数据信息，属于静态信息维护，一般不会出现服务质量相互影响的风险。
- ACVIP：代表寻址组件，被用户的不同环境共用，按照逻辑实例维度进行寻址，逻辑实例与 Workspace 一一对应。

如果用户预期不同环境的消息队列服务质量不受影响，可以采取按照逻辑实例维度划分消息 Broker 服务集群的解决方案，例如：生产环境与非生产环境依赖不同的消息服务集群。示意图如下：



非生产环境应用系统依赖单独的消息 Broker 服务集群 B，不再与生产环境应用系统共用资源，达到消息发布和订阅流量隔离的目标。

对于配置元数据，不同环境可以共享一套控制台服务集群，静态数据因为资源共享而导致的相互影响风险可控。

4. 性能指标

不同的机器配置下的性能不同，即对于不同的 ECS 和 RDS 规格，系统吞吐量（TPS）会有所不同。详细的性能数据如下表所示：

ECS 规格	RDS 规格	TPS
4C 8G	4C 16G	4000
4C 8G	8C 16G	10000

② 说明

一组机器要求至少包括 4 台 ECS 及 2 RDS，如有需要，可以在此基础上以机器组的维度进行扩容。

5. 功能原理

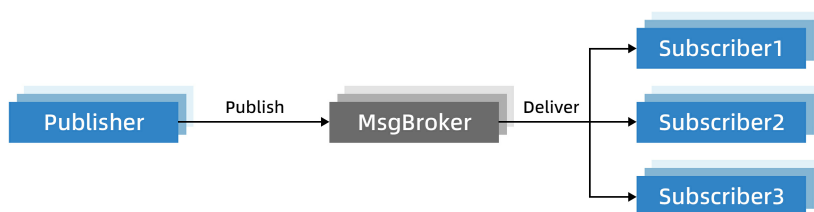
消息队列的主要功能如下：

- 支持 Pub-Sub 消息模型：消息发布者允许发送多个消息类型，消息类型可被多个订阅者订阅，分布式消息服务负责按照消息类型维度做消息分发投递。
- 消息异步投递：为不同应用系统间提供异步消息通信，分布式消息队列服务负责将消息异步投递到订阅者。
- 支持事务消息：消息发送客户端支持消息状态与本地数据库事务的一致性，对应用系统透明。
- 水平伸缩：发布者系统、消息代理组件、订阅者系统都支持集群水平扩展，可依据业务消息量动态部署计算节点。
- 管理工具：提供 Web 控制台，支持消息类型管理、订阅关系管理、消息查询、消息轨迹查询等。

消息客户端介绍

基本功能

消息发布者（Publisher）与消息订阅者（Subscriber）是一对多的关系，即一个消息类型由一个消息发布者发出，但可以被一个或者多个订阅者接收。

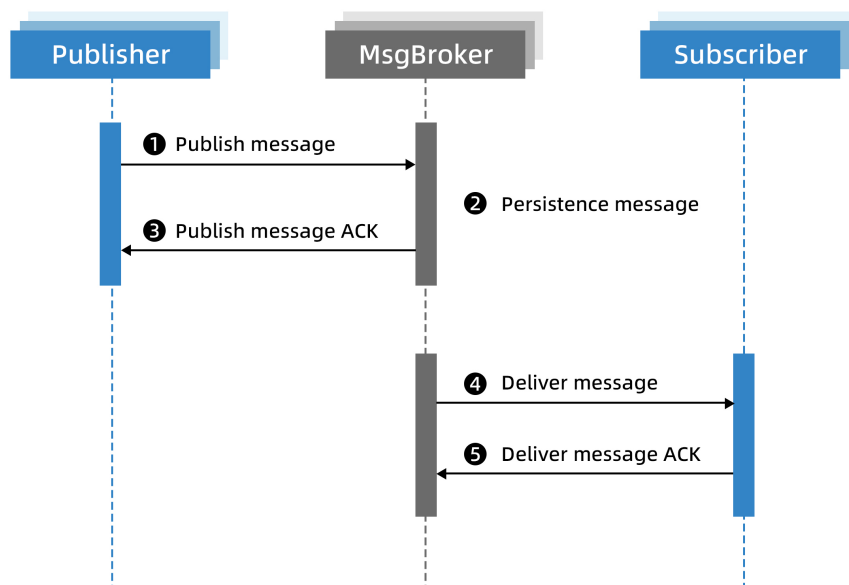


基本特性

- 消息订阅者收到的消息不保证有序，即收到消息的顺序与发布者发送消息的顺序可能会不一致。
- 消息投递策略为至少一次，即对于同一条消息消息订阅者可能收到多次，要求订阅者保证幂等特性。

普通消息接收与投递流程

普通消息是相对于事务型消息而言的，消息接收与投递流程如下图所示：

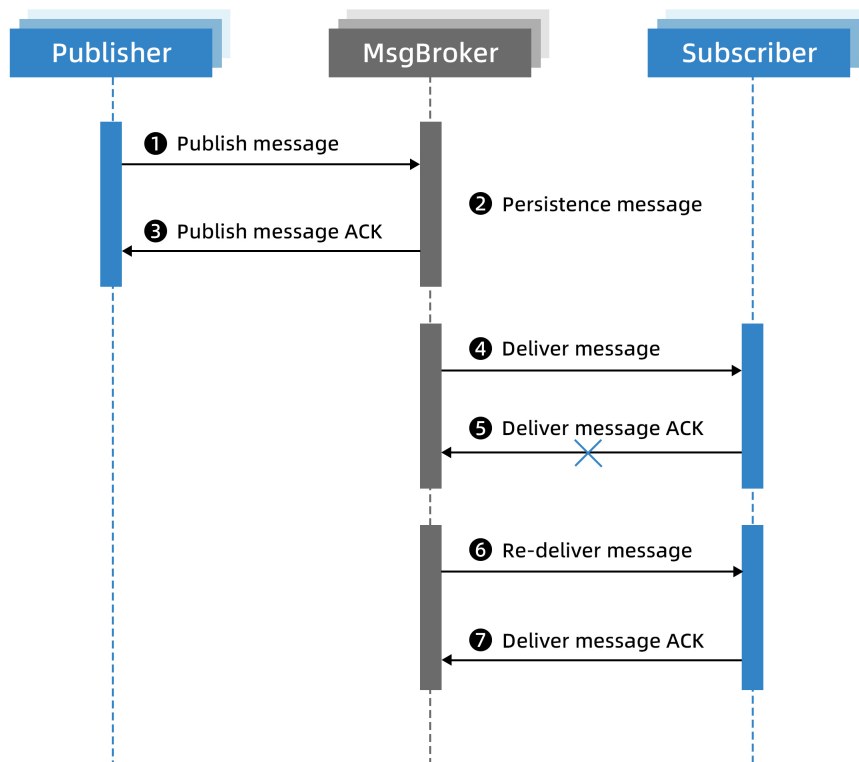


消息重试投递流程

消息投递异常的常见原因如下：

- 订阅端系统未连接到可靠消息组件（MsgBroker）。
- 订阅端系统收到消息后处理超时，默认消息投递超时时间是 10 秒。
- 订阅端系统收到消息后处理异常。
- 订阅端系统收到消息后主动回滚。
- 当消息投递异常时，可靠消息组件会按照预定时间间隔策略重新投递异常消息，直至订阅端系统消费成功。

消息重试投递流程如下图所示：

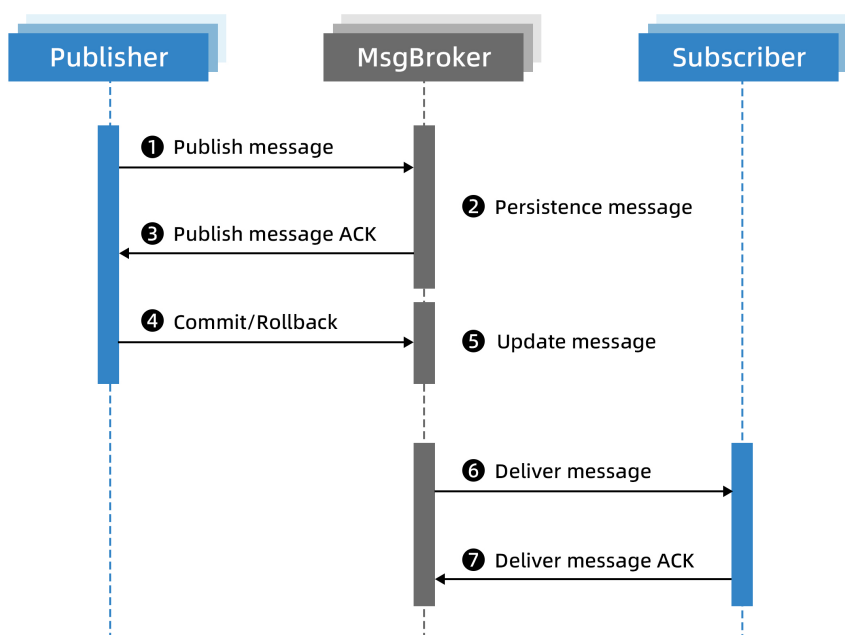


事务型消息接收与投递流程

事务型消息涉及的角色是消息发送端系统和可靠消息组件，与消息订阅端系统无关。

事务型消息是否被投递与发送端系统本地数据库事务保持一致，如果本地数据库事务提交则消息会被投递给订阅端；如果本地数据库事务回滚，则直接丢弃消息不投递给订阅端系统。

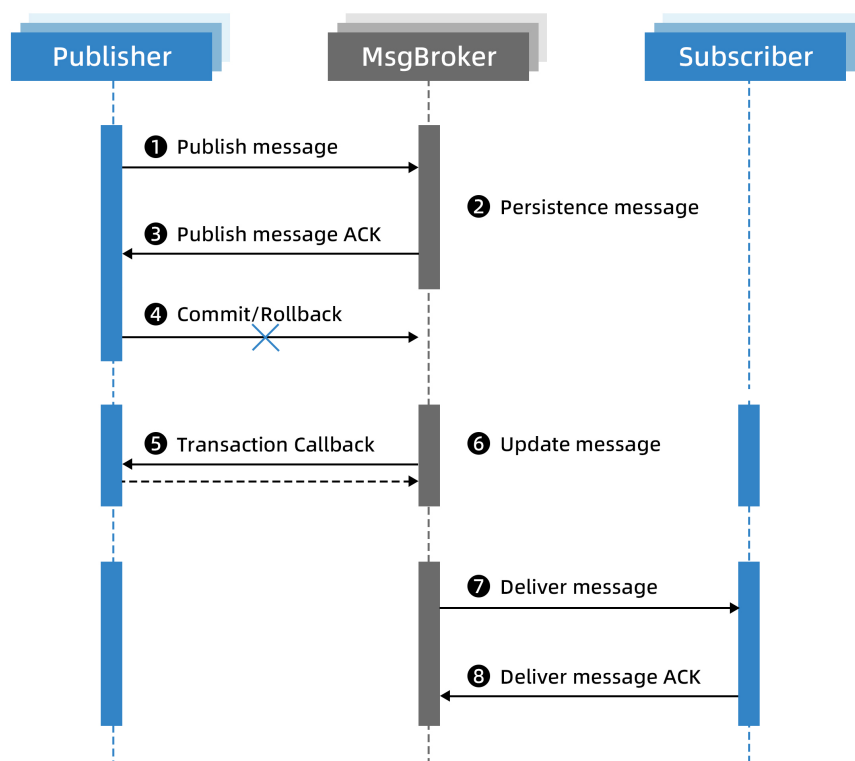
事务型消息接收与投递流程如下图所示：



事务型消息回查流程

事务型消息涉及典型的两阶段消息流程，第一阶段是消息发布端发送消息到可靠消息组件，第二阶段是消息发布端发送提交或者回滚指令到可靠消息组件，可靠消息组件根据此指令决定是否投递消息到订阅端系统。当第二阶段指令出现异常时，可靠消息组件在一定时间后主动回查消息发送端系统，确认对应的事务型消息是否投递。

事务型消息回查流程如下图所示：



6.附录：基础术语

消息中间件

在分布式系统环境中，支持从发布者系统接收消息并投递到订阅者系统的软件组件。

可靠消息组件

支持消息高可靠特性的消息中间件。

消息发布者（Publisher）

发送消息的应用系统。

消息订阅者（Subscriber）

接收一种或者多种消息的应用系统，也可叫做消息消费者（Consumer）。

Topic

代表一个消息大类，表示一类具体的业务（并不是表示一个系统）。

Eventcode

代表一个消息大类（Topic）下的一个消息子类，用来表述一个大的业务下具体的功能。

pub groupId

一类的标识，这类 publisher 通常生产并发送一类消息，且发送逻辑一致。

sub groupId

一类的标识，这类 subscriber 通常接收并消费一类消息，且消费逻辑一致。

事务消息

消息中心提供类似 X/Open XA 的分布事务功能，并保证消息与本地数据库事务一致性。

订阅关系

订阅关系表示订阅方的 GroupId 和消息类型的映射关系，是消息中心投递消息的唯一依据。

租户

多租户场景下，租户的名字，用来做配置隔离。

环境

多租户场景下，环境的名字，用来做配置隔离。