



Fescar 在微服务架构下一致性的探索

季敏 Fescar 开源项目发起人

议题介绍

01

常用分布式一致性
模式介绍

02

什么是 Fescar

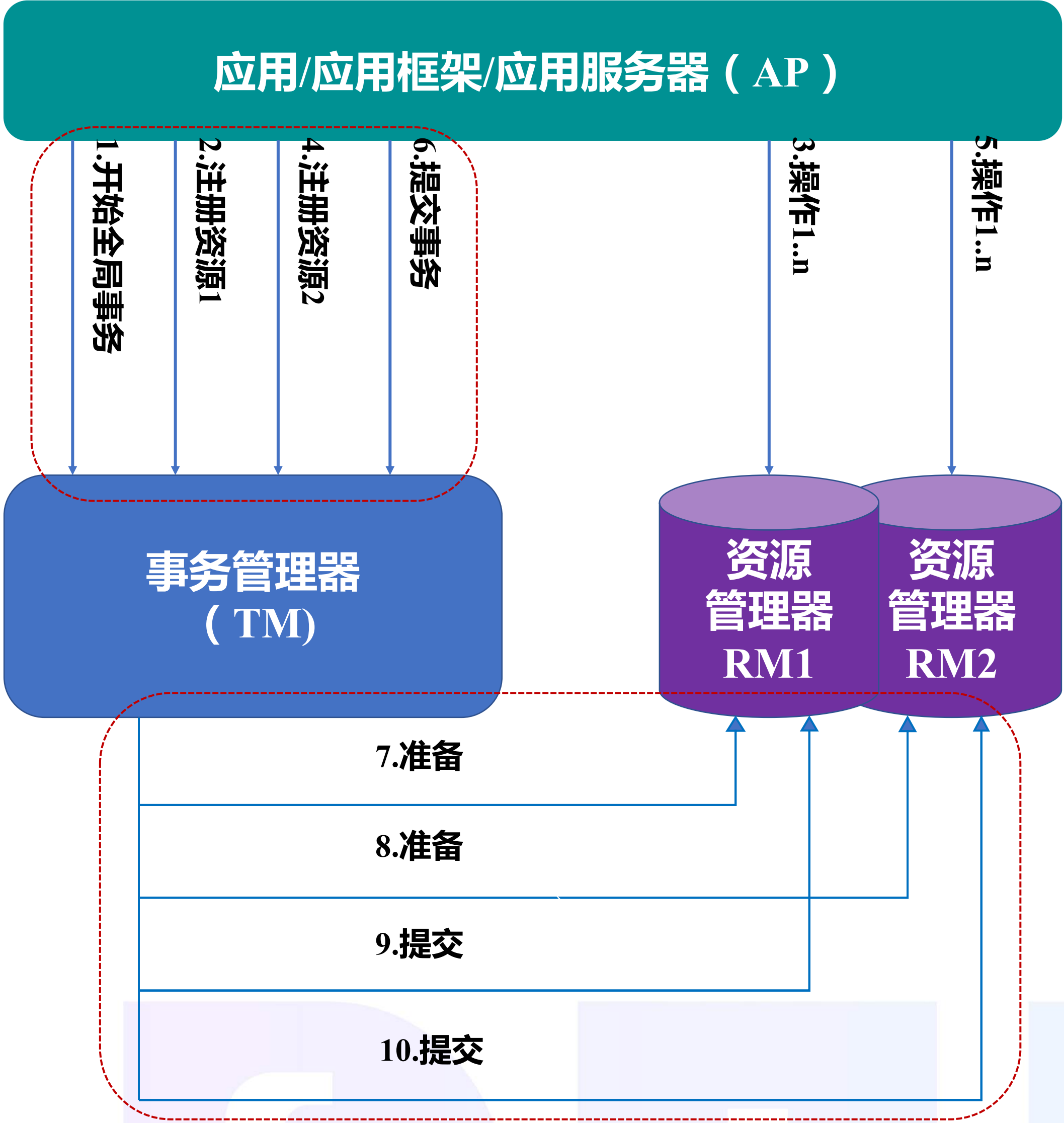
03

Fescar 原理

04

Fescar 过去、现
在和未来

DTP模型--XA



XA是由X/Open组织提出的分布式事务的规范。XA规范主要定义了(全局)事务管理器(TM)和(局部)资源管理器(RM)之间的接口。主流的关系型数据库产品都是实现了XA接口的。

XA接口是双向的系统接口，在事务管理器（TM）以及一个或多个资源管理器（RM）之间形成通信桥梁。

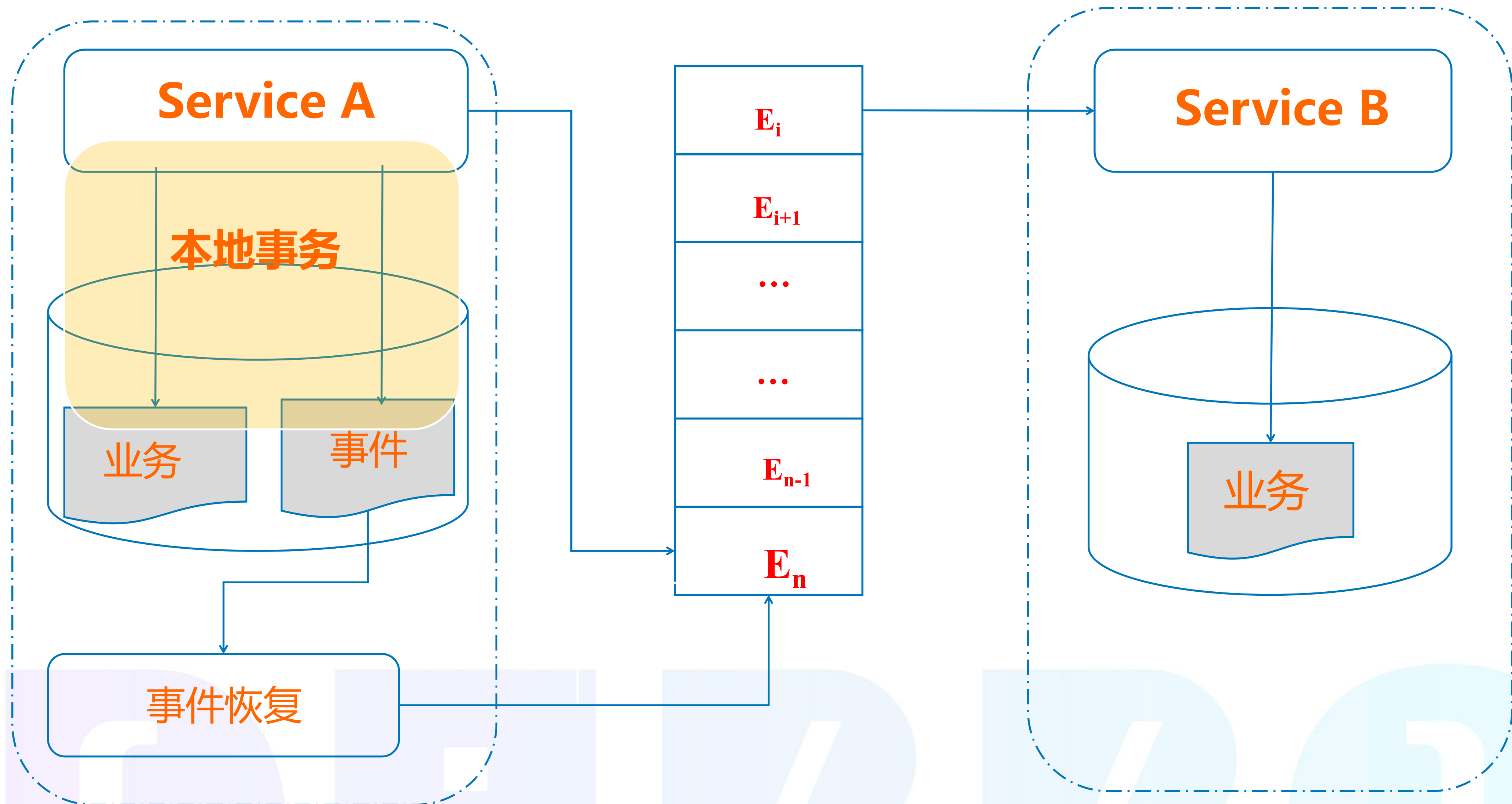
XA之所以需要引入事务管理器是因为，在分布式系统中，从理论上讲两台机器理论上无法达到一致的状态，需要引入一个单点进行协调。

由全局事务管理器管理和协调的事务，可以跨越多个资源（如数据库或JMS队列）和进程。

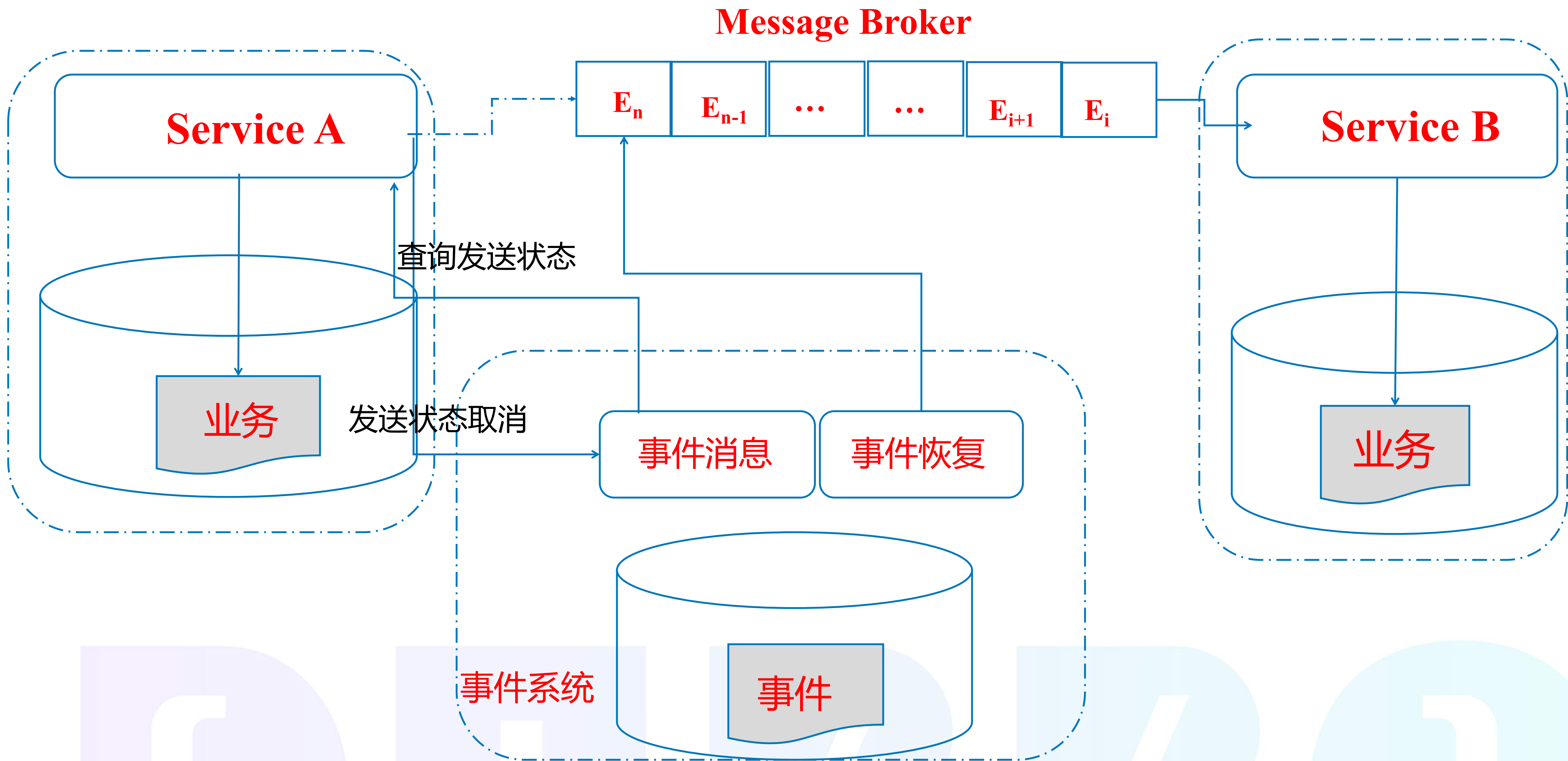
柔性事务-可靠事件模式

```
public void msgTrans() {  
    try {  
        boolean result = dao.update(model); //①  
        //②  
        if (result) {  
            mq.send(model); //③  
        }  
        //④  
    } catch (Exception exx) {  
        rollback();  
    }  
}
```

可靠事件模式—本地事件表



可靠事件模式—外部事件表



What is Fescar

Fescar: Fast & Easy Commit And Rollback

微服务架构下，易用、高效的分布式事务解决方案

技术积累:

TXC

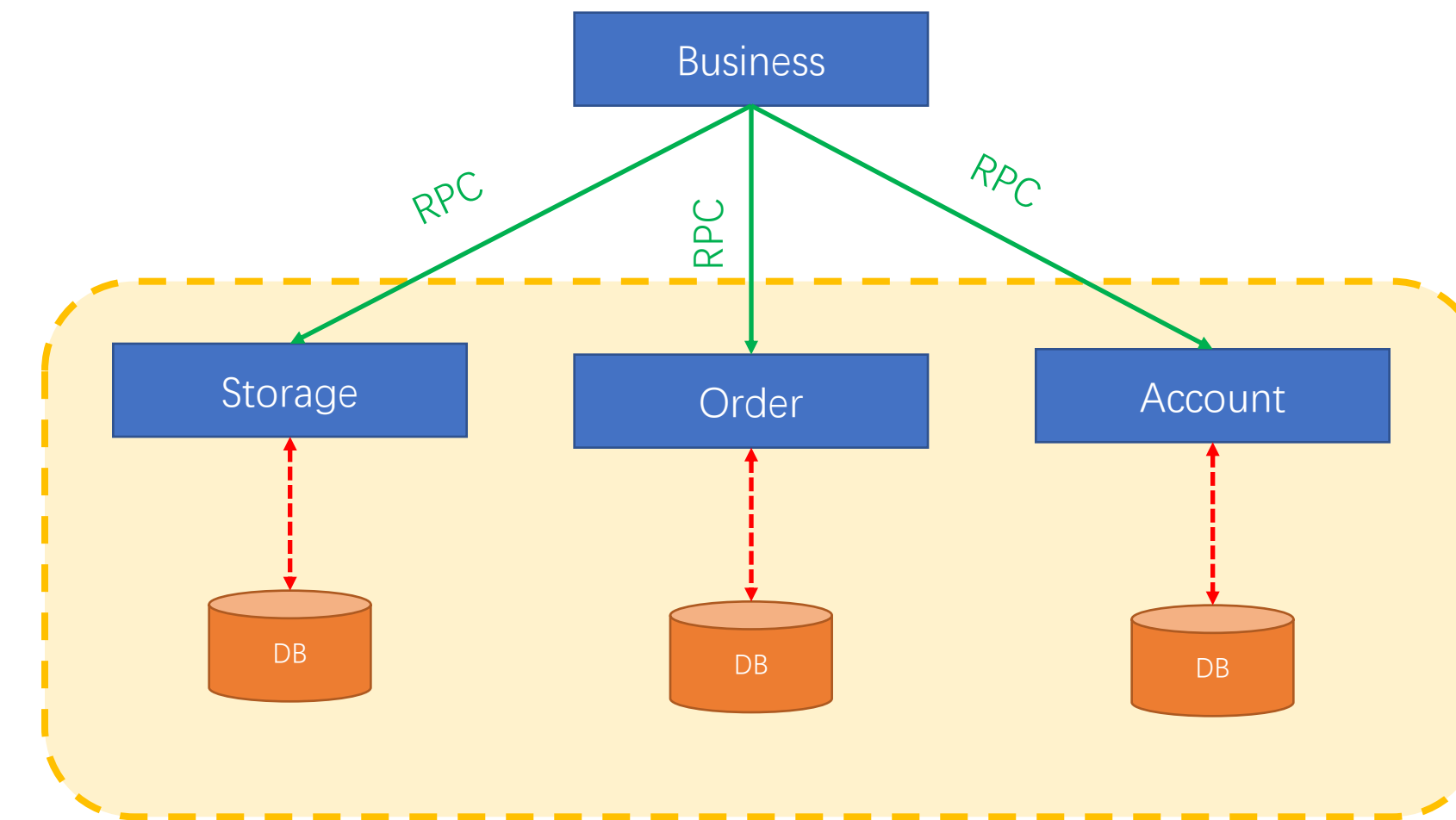
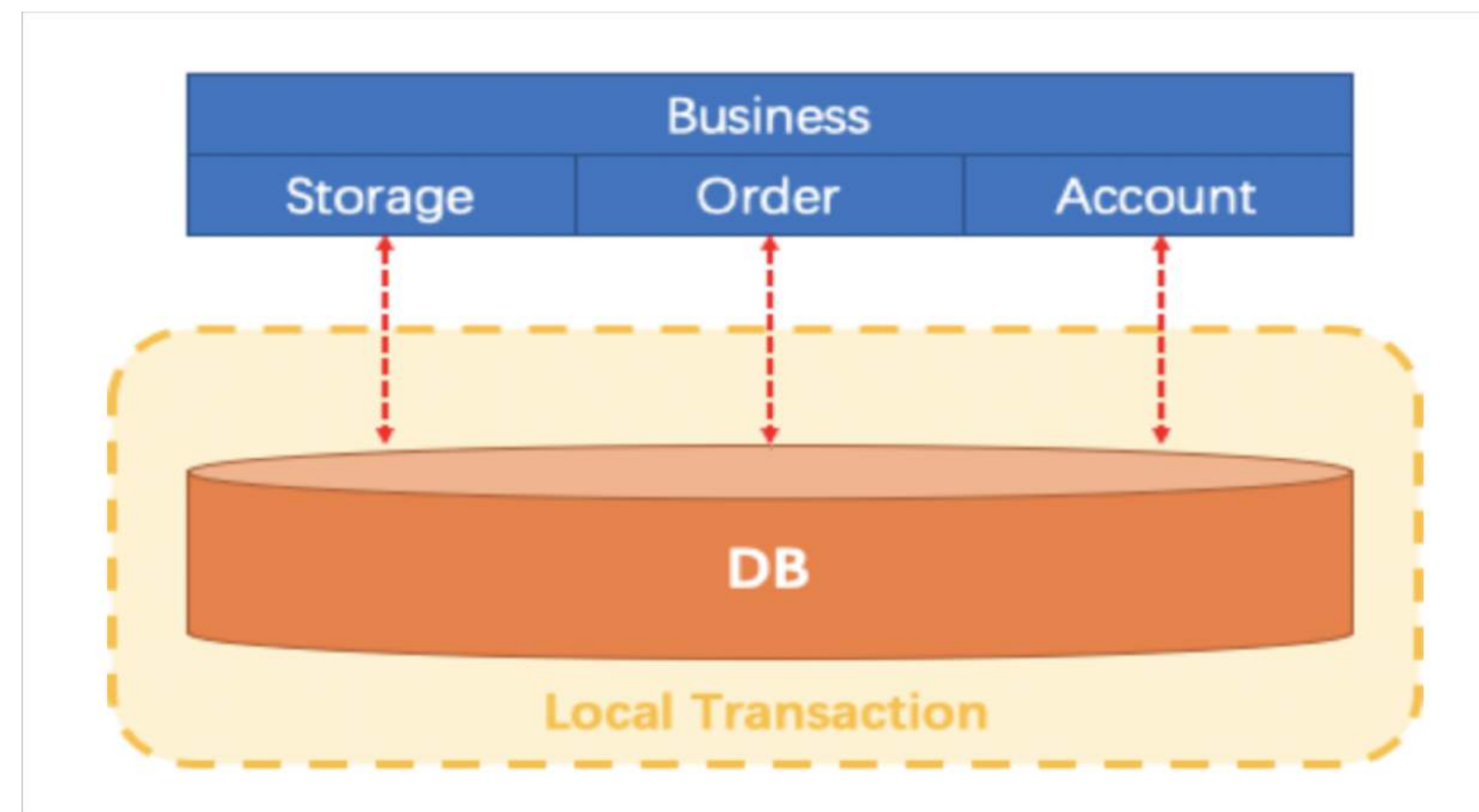
GTS

愿景:

像使用本地事务一样使用分布式事务



Problem VS Solution

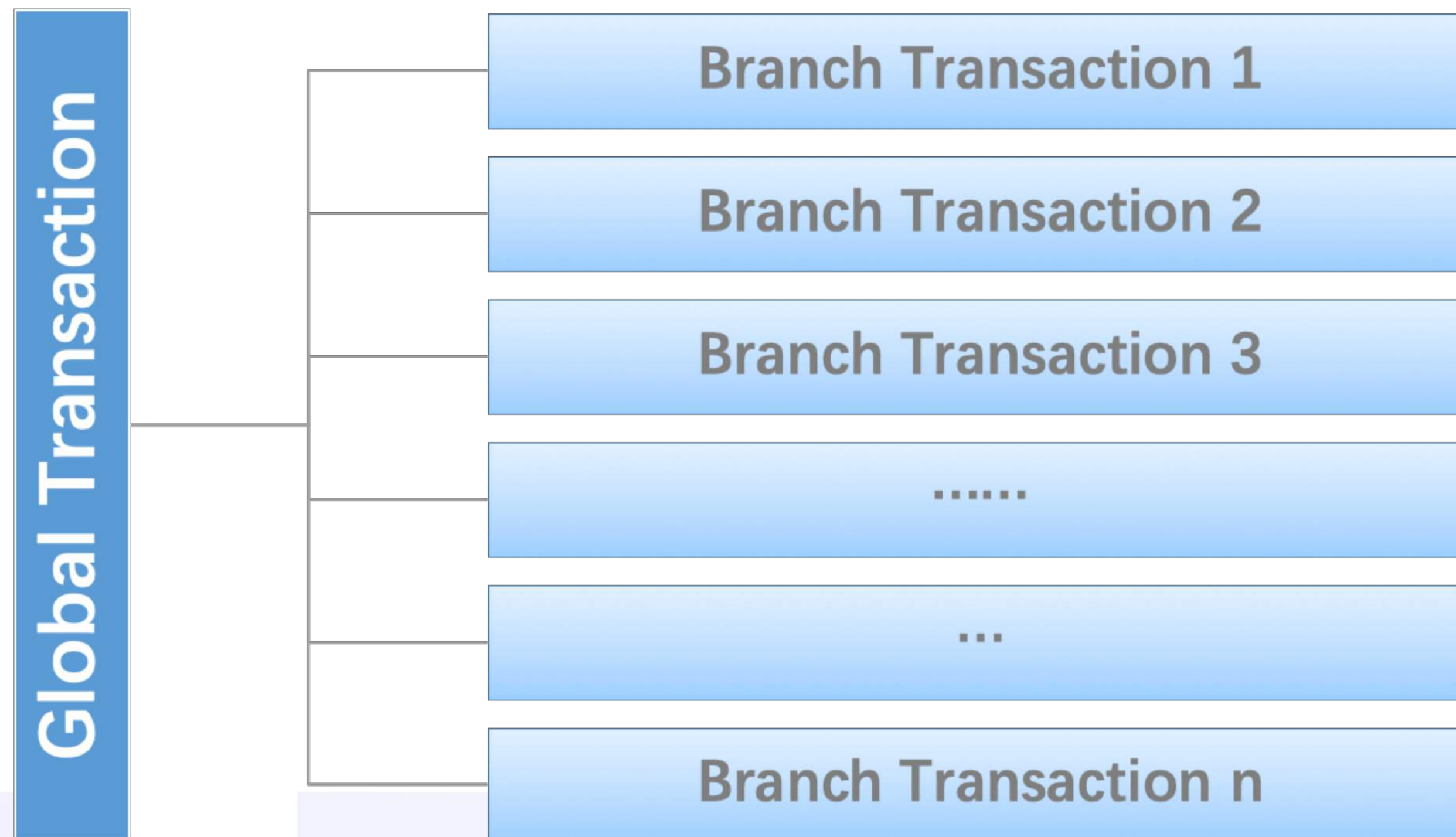


```
@Transactional(timeout = 300000)
public void purchaseLocal(String userId,
                          String commodityCode,
                          int count) {
    storageLocalService.deduct(
        commodityCode, count);
    int needPay = orderLocalService.create(
        userId, commodityCode, count);
    accountLocalService.debit(userId, needPay);
}
```

```
@GlobalTransactional(timeout = 300000)
public void purchaseRemote(String userId,
                           String commodityCode,
                           int count) {
    storageRemoteService.deduct(
        commodityCode, count);
    int needPay = orderRemoteService.create(
        userId, commodityCode, count);
    accountRemoteService.debit(userId, needPay);
}
```


Fescar 原理介绍

一个分布式（全局）事务是由若干本地（分支）事务组成。



Fescar 原理介绍

◆Transaction Coordinator(TC)：

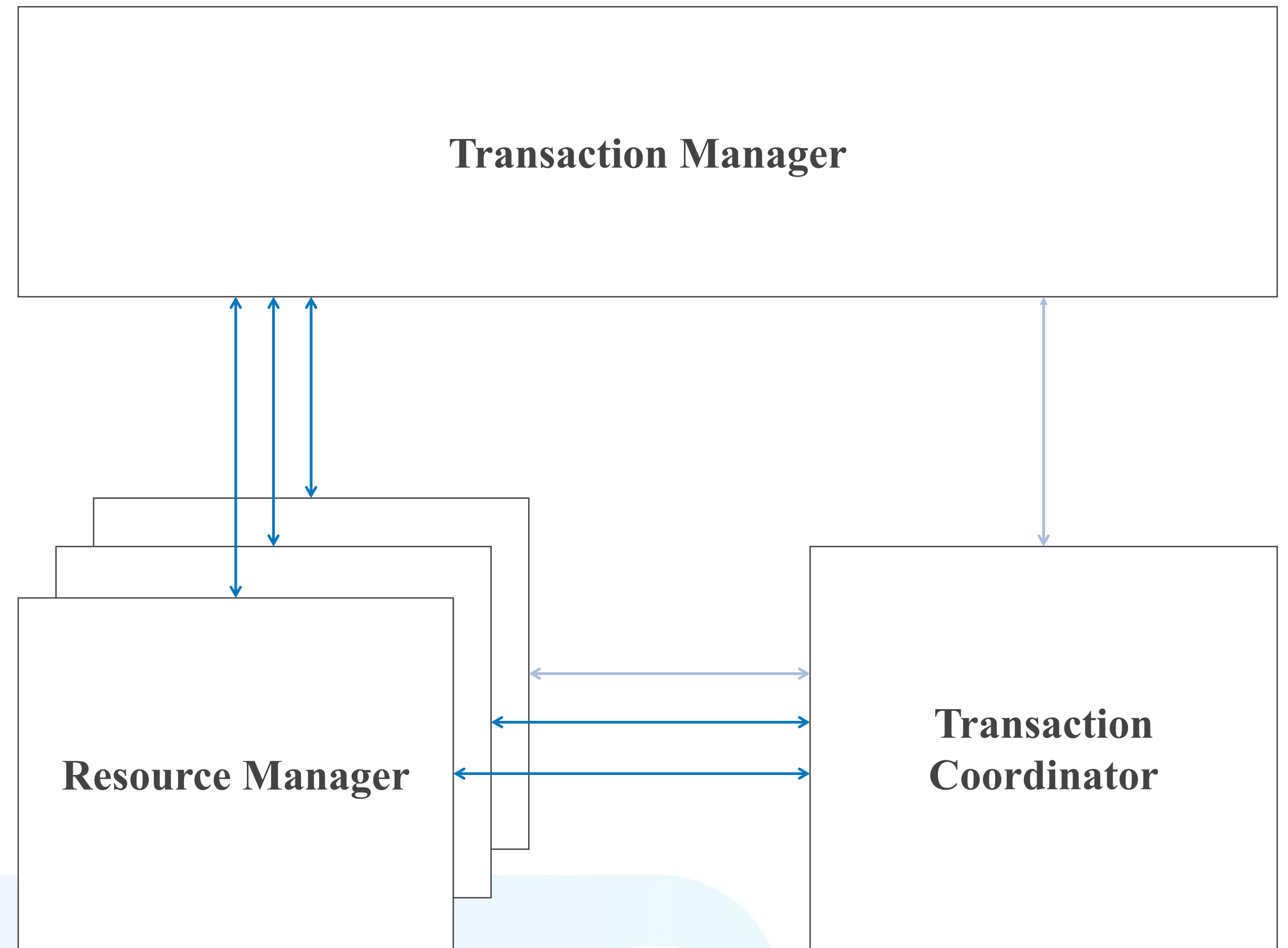
事务协调器，维护全局事务的运行状态，驱动全局事务的提交或回滚。

◆Transaction Manager(TM)：

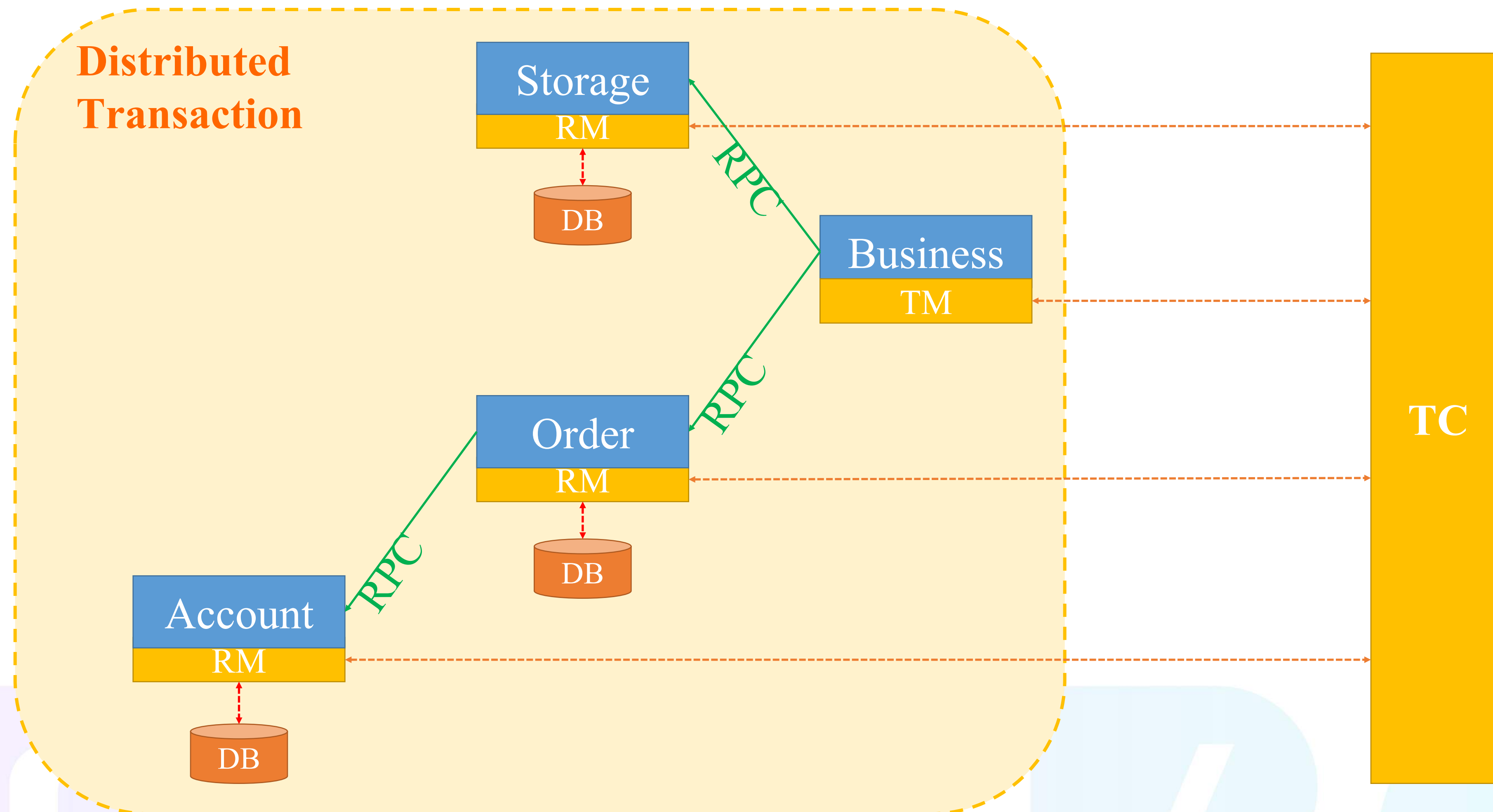
控制全局事务的边界，负责开启一个全局事务，并最终负责发起全局提交或全局回滚。

◆Resource Manager(RM)：

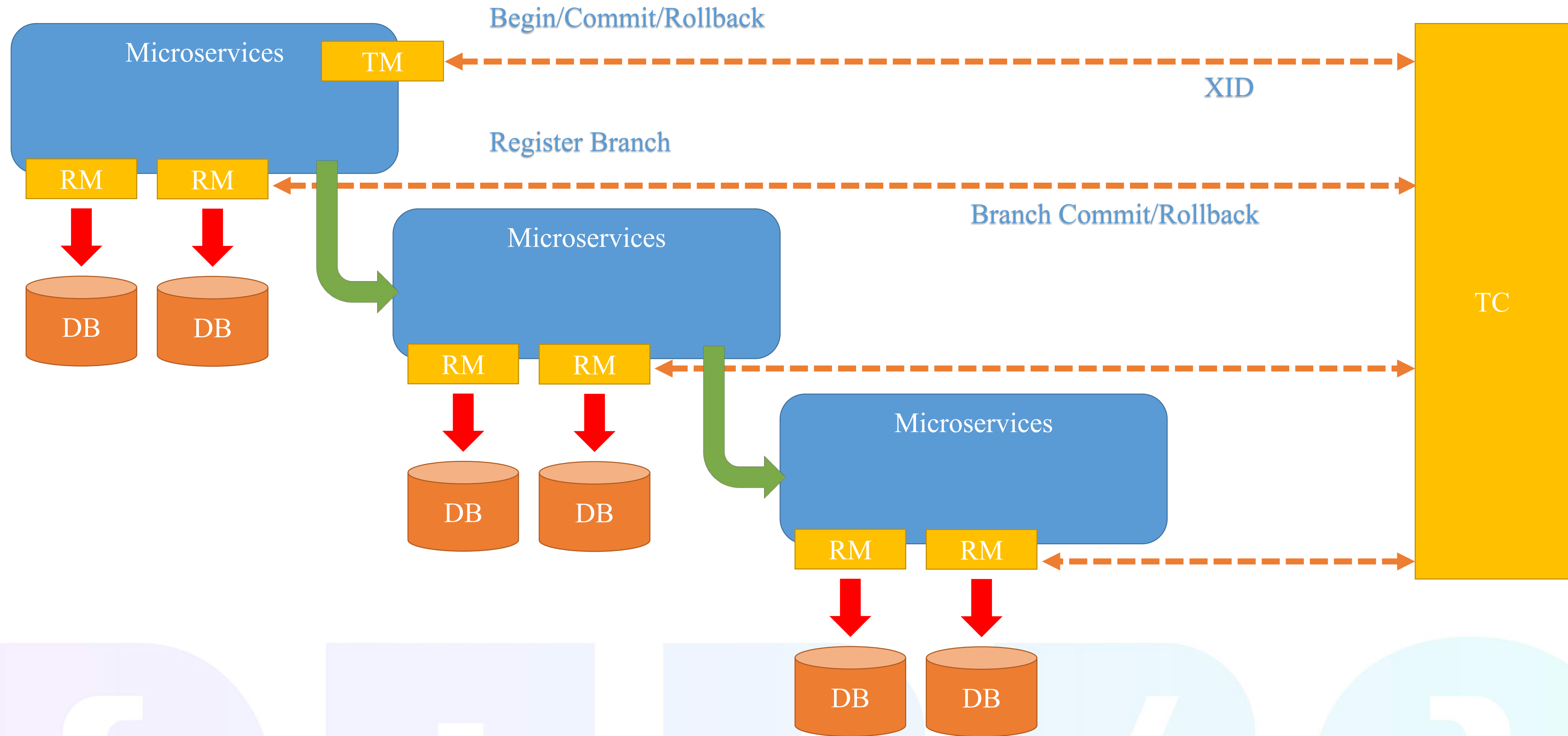
控制分支事务，负责分支事务的注册、状态汇报，并驱动分支（本地）事务的提交和回滚。



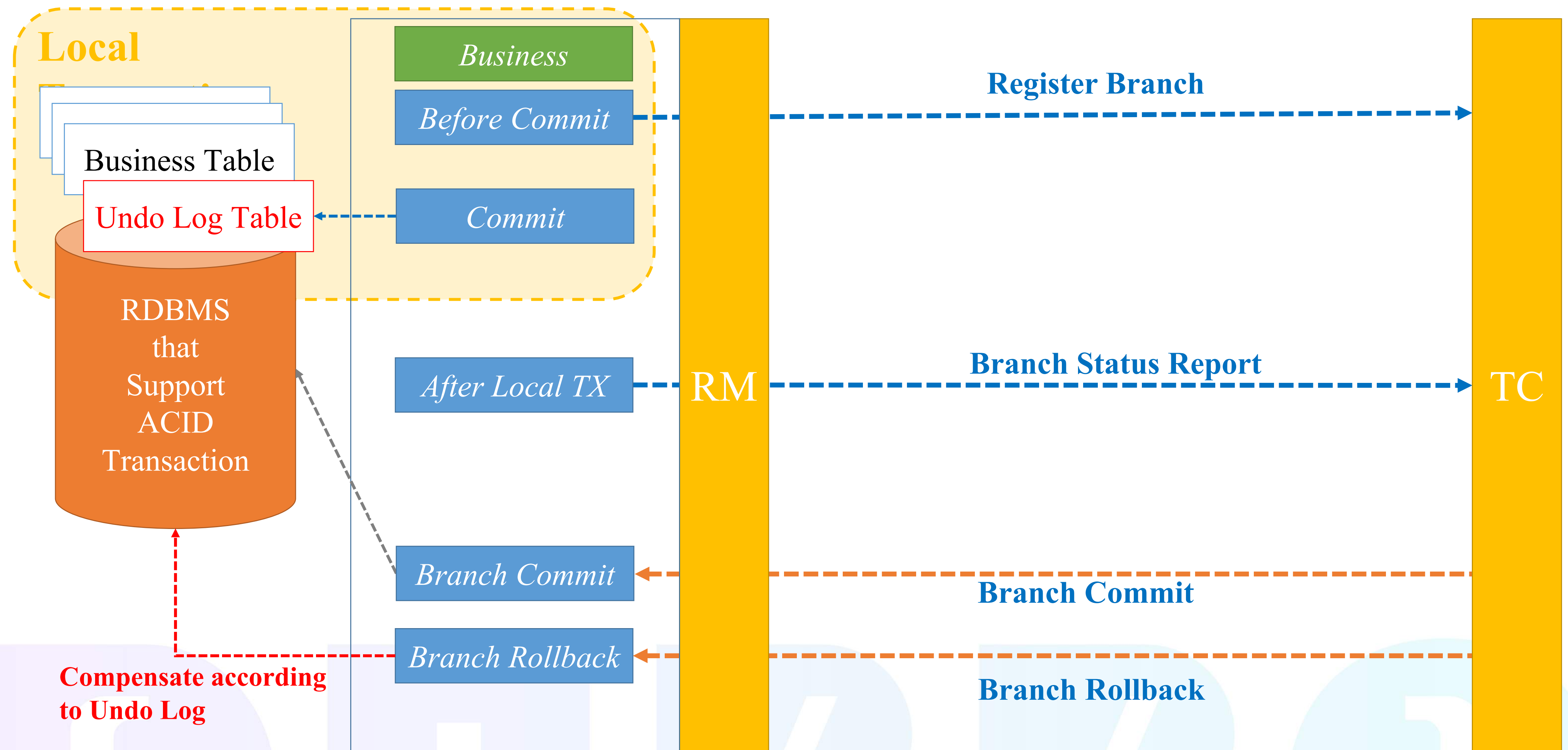
Fescar 原理介绍



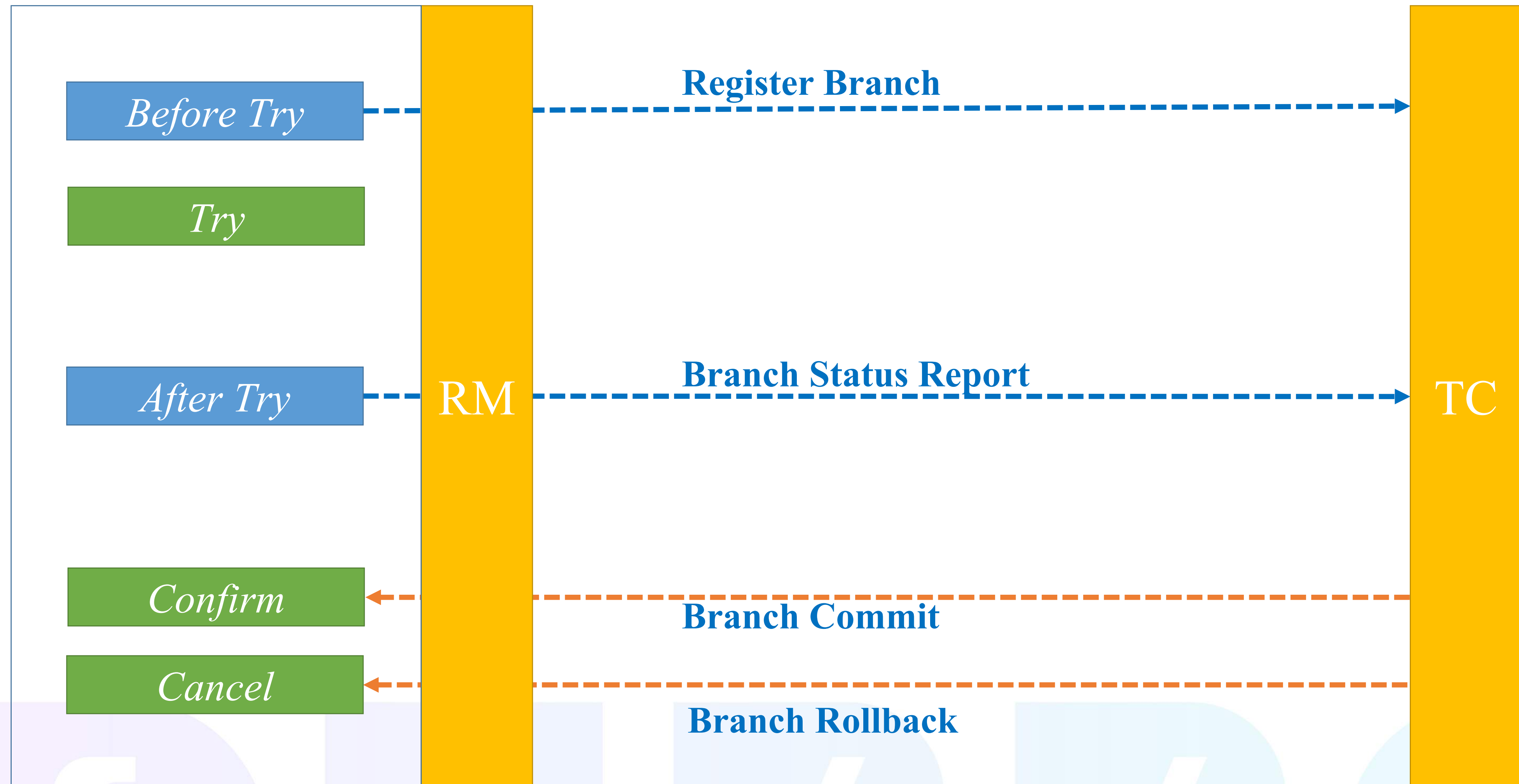
Fescar 原理介绍



Fescar-AT



Fescar-TCC



阿里巴巴分布式事务



蚂蚁金服分布式事务

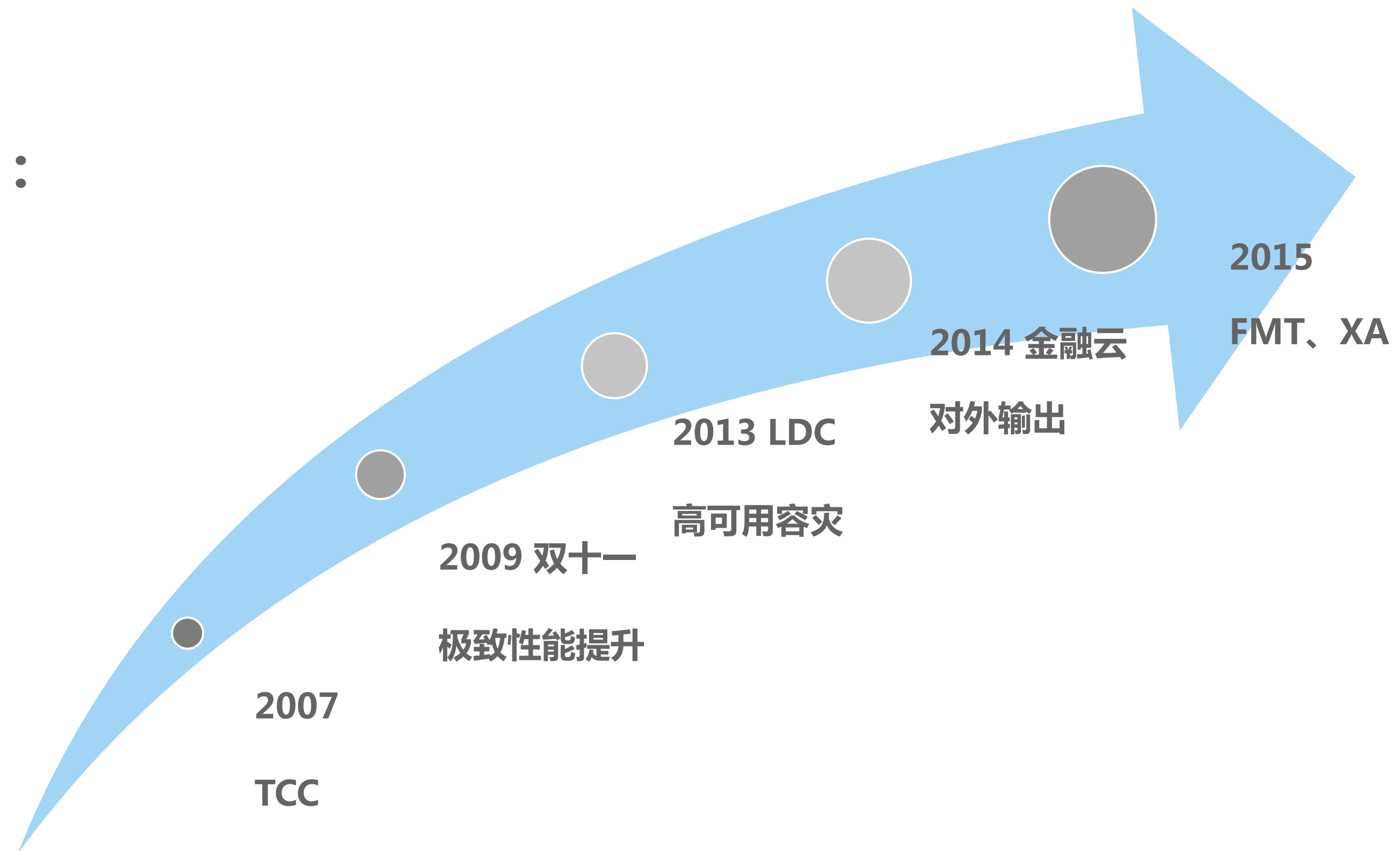
❖ 三种模式，丰富的应用场景：

- TCC模式
- FMT模式
- XA模式

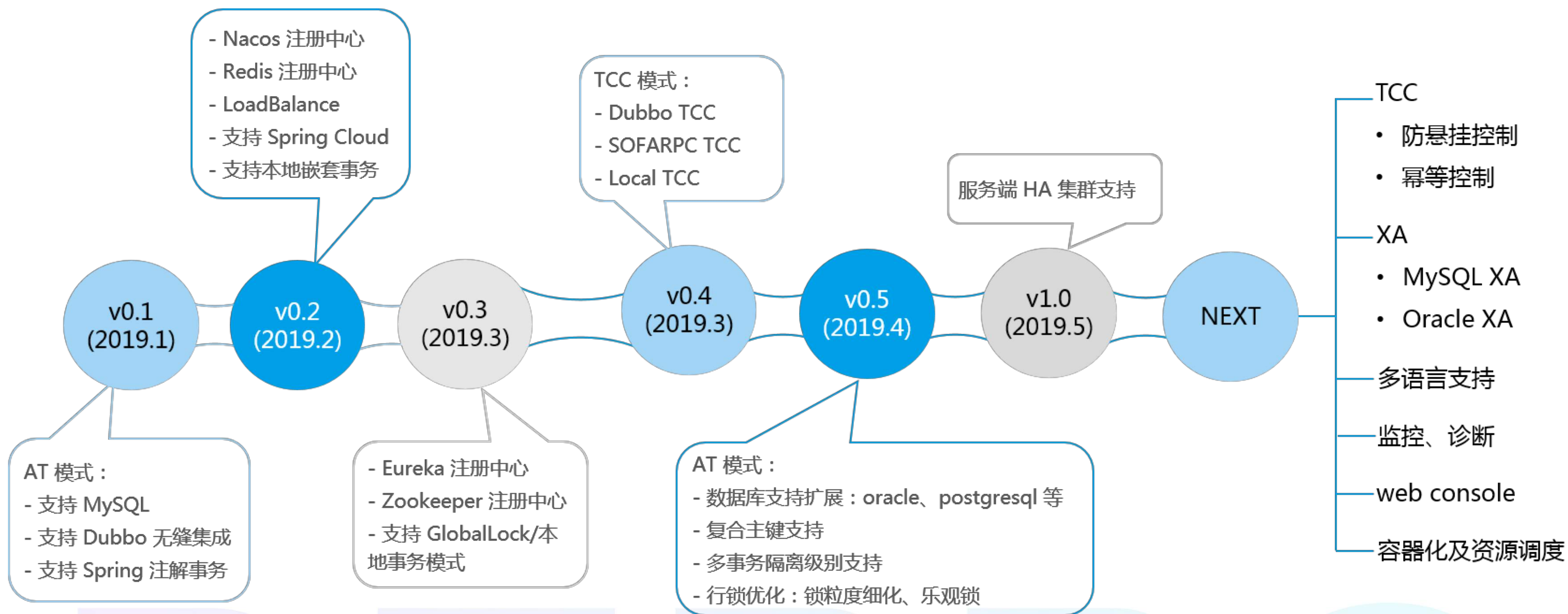
❖ 极致性能提升

❖ 高可用容灾

❖ 无侵入解决方案



阿里巴巴 + 蚂蚁金服 社区共建



实例演示



环境要求

JDK 1.8+

MySQL 5.6+

Nacos 0.8+

演示内容

模拟创建订单、库存和账户

Dubbo微服务间的一致性

<https://github.com/fescar-group/fescar-workshop>

Join Us



欢迎钉钉扫描二维码进群

GitHub: <https://github.com/alibaba/fescar>





Thank you !