

Seata-微服务架构开发的必备利器

季敏

Seata 开源社区负责人

蔡铭霞

阿里巴巴分布式事务团队负责人

Seata 开源项目发起人/负责人

开源布道师，致力于打造下一代的分布式事务标准

GitHub : <https://github.com/slievrly>

Email : slievrly@163.com (招聘)

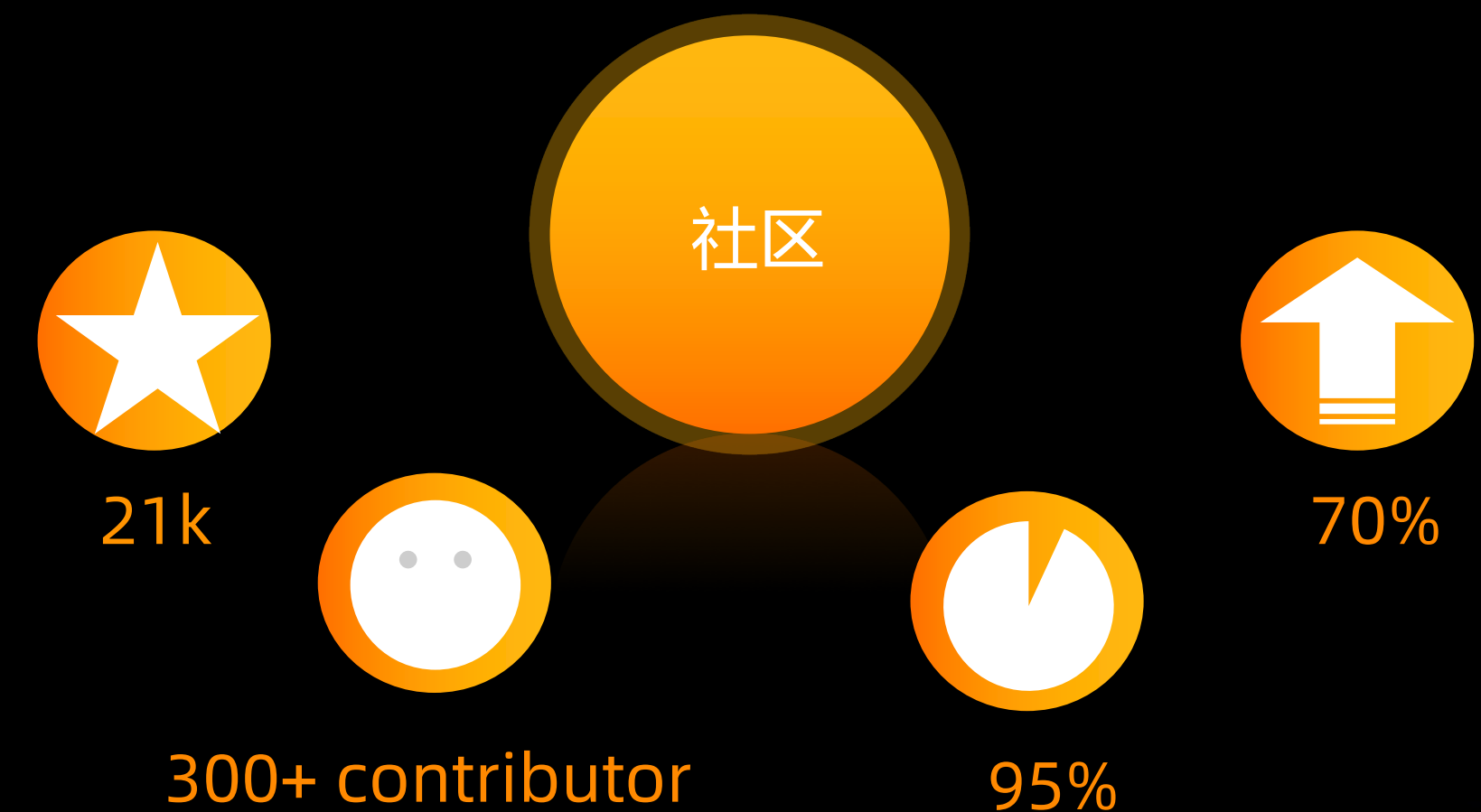


01 Seata 1.5.0 有哪些不一样？

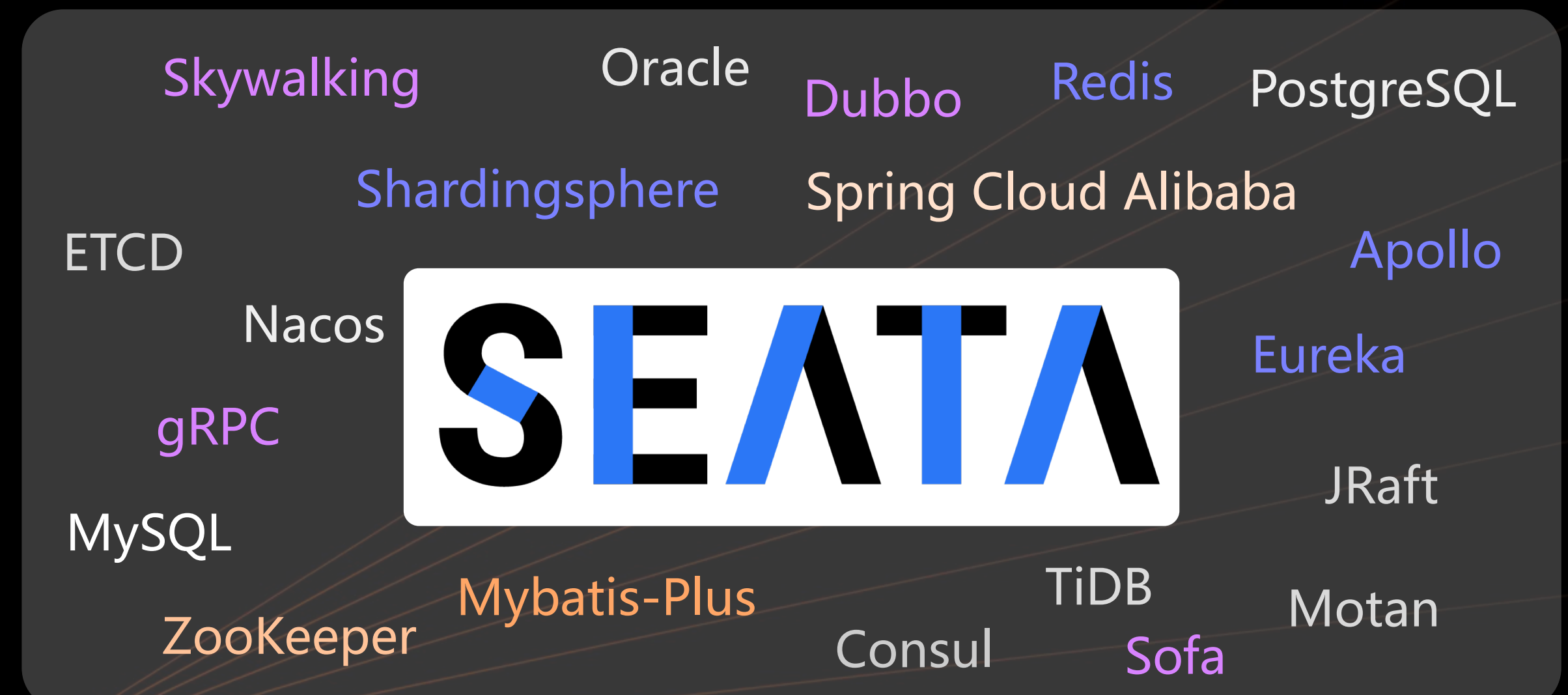
Seata 开源社区

SEATA : Simple **E**xtensible **A**utonomous **T**ransaction **A**rchitecture

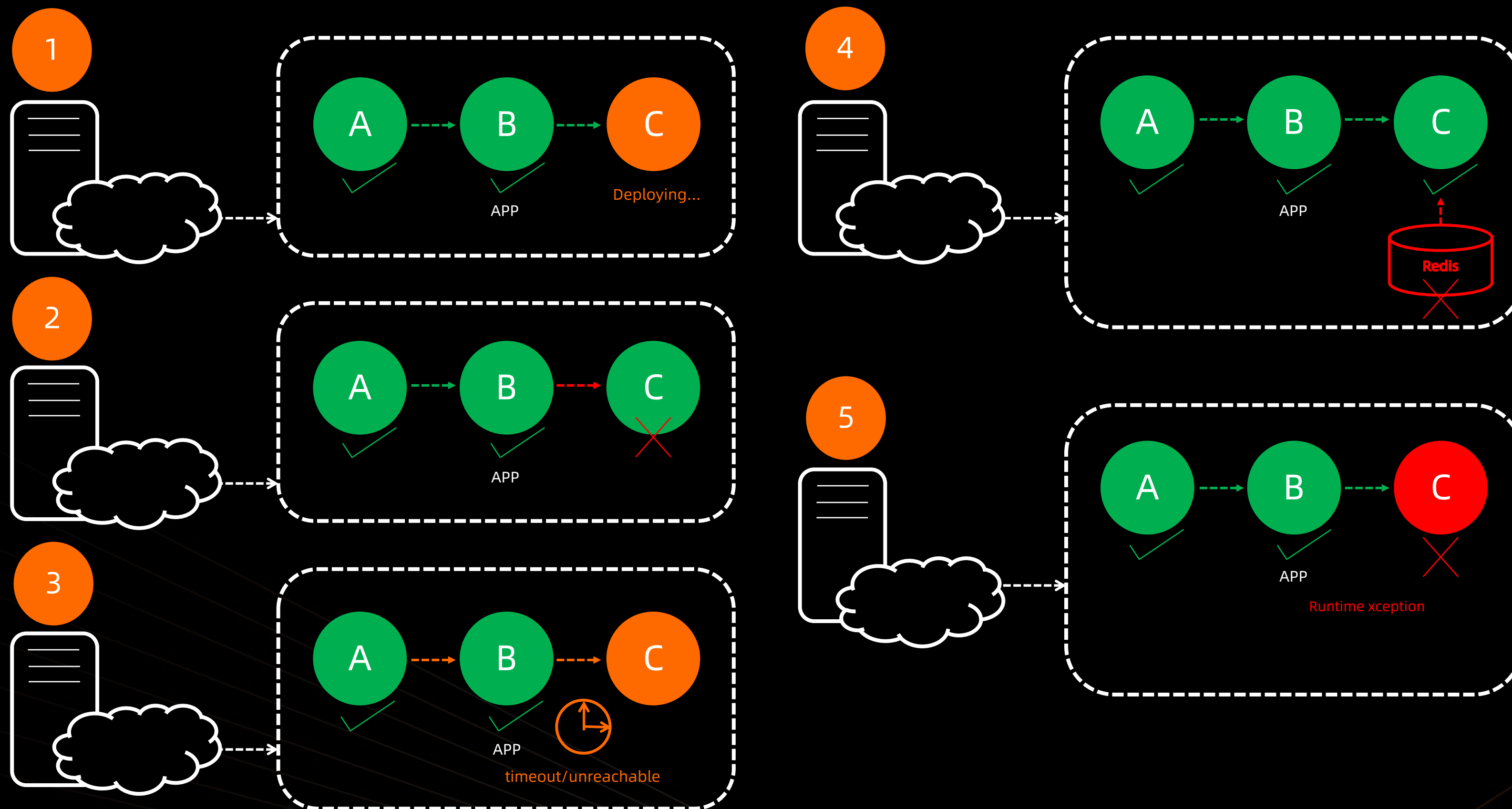
一款开源的分布式事务解决方案，致力于在微服务架构下提供高性能和简单易用的分布式事务服务。



- Seata 对于市面上主流的关系数据库，RPC框架做了广泛的支持，同时被许多第三方社区做了主动和被动集成。
- 支持 Java、Golang、PHP、Python 多语言
- 目前 Seata 开源产品已被上千家企业在业务系统中应用，金融企业纷纷试点。
- 阿里经济体增长最快的开源项目。



微服务架构下数据一致性的困扰



常见一致性场景：

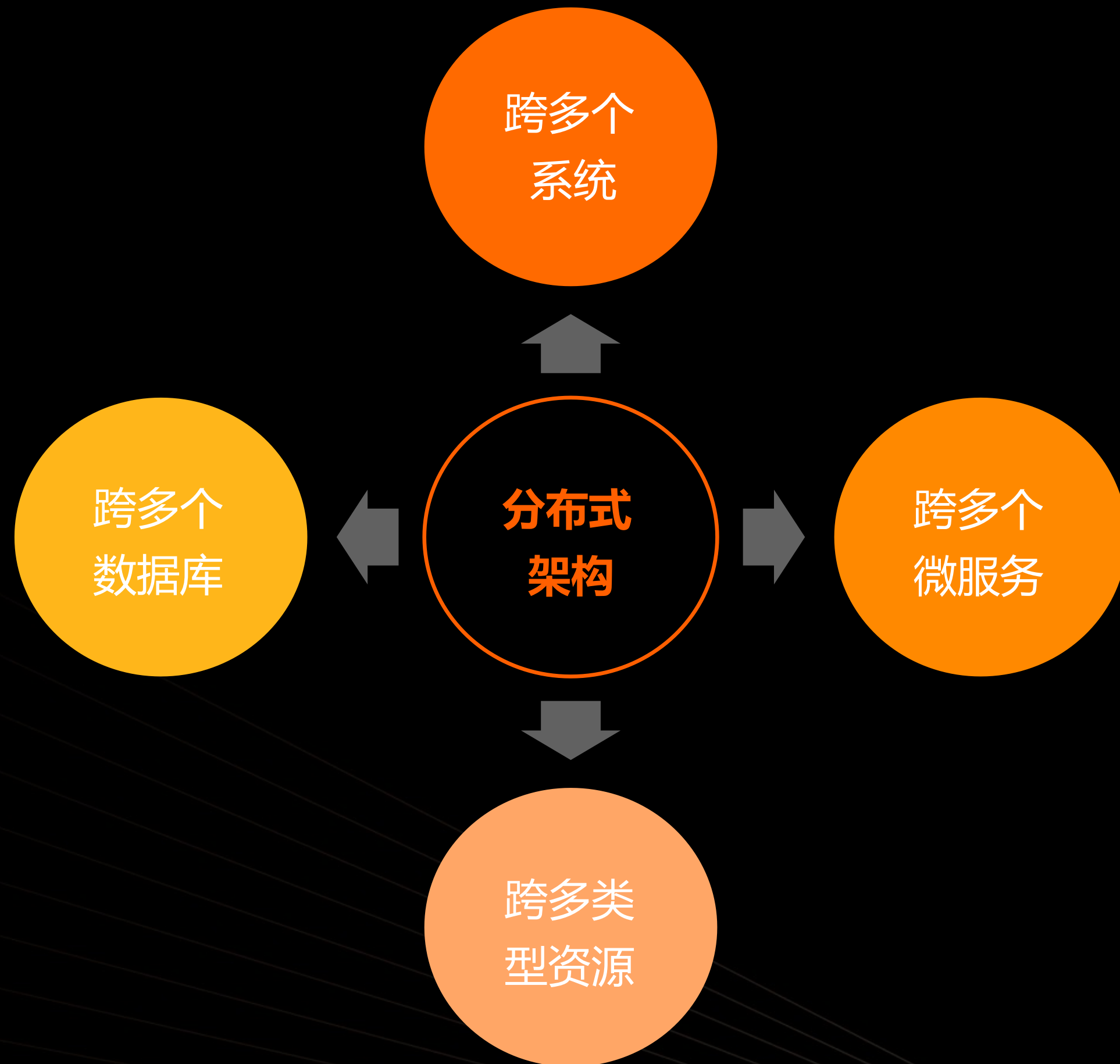
系统异常：

1. 服务不可达：网络、宕机、服务上下线，熔断限流
2. 服务超时

业务异常：

1. 业务校验非法
2. 运行时异常

分布式事务产生原因总结



1. 跨数据库操作

随着业务规模的增大，单库难以满足业务量和连接数需求，会出现分库的情况，这时原有的单机事务会变成分布式事务。

2. 跨系统的分布式事务

在和第三方系统（含企业内外）集成时，本地的操作强依赖于第三方系统的响应结果，会产生分布式事务的场景。

3. 跨服务的分布式事务

业务服务化后，资源与客户端调用解耦，同时又要保证多个服务调用间资源的变化保持强一致，否则会造成业务数据的不完整。

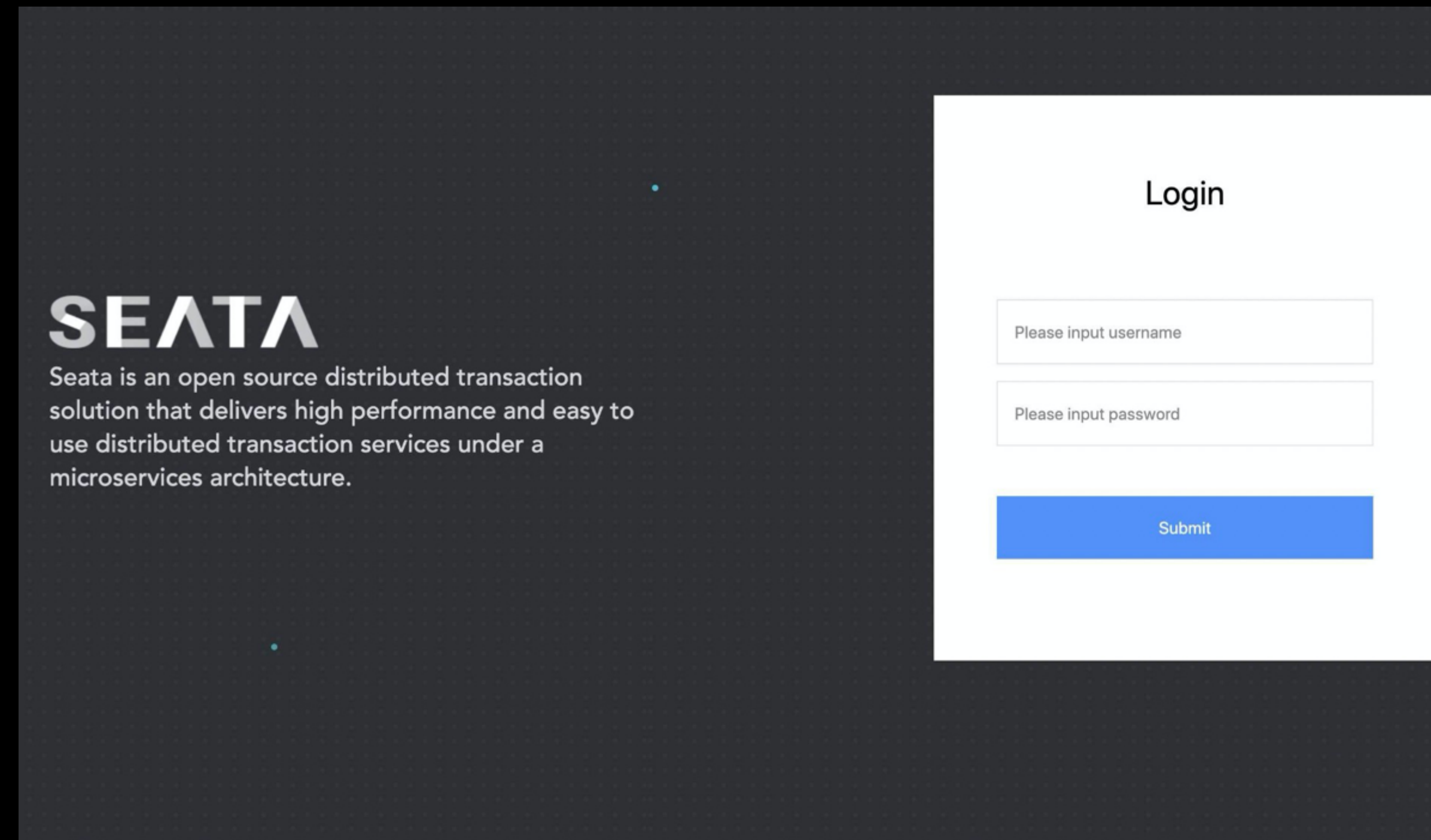
4. 跨数据库与消息的分布式事务

在某些业务场景中，需要进行多个 DB 操作的同时，还会调用消息系统，需要保证整个链路的一致性。

Seata 1.5.0

SEATA 1.5.0:

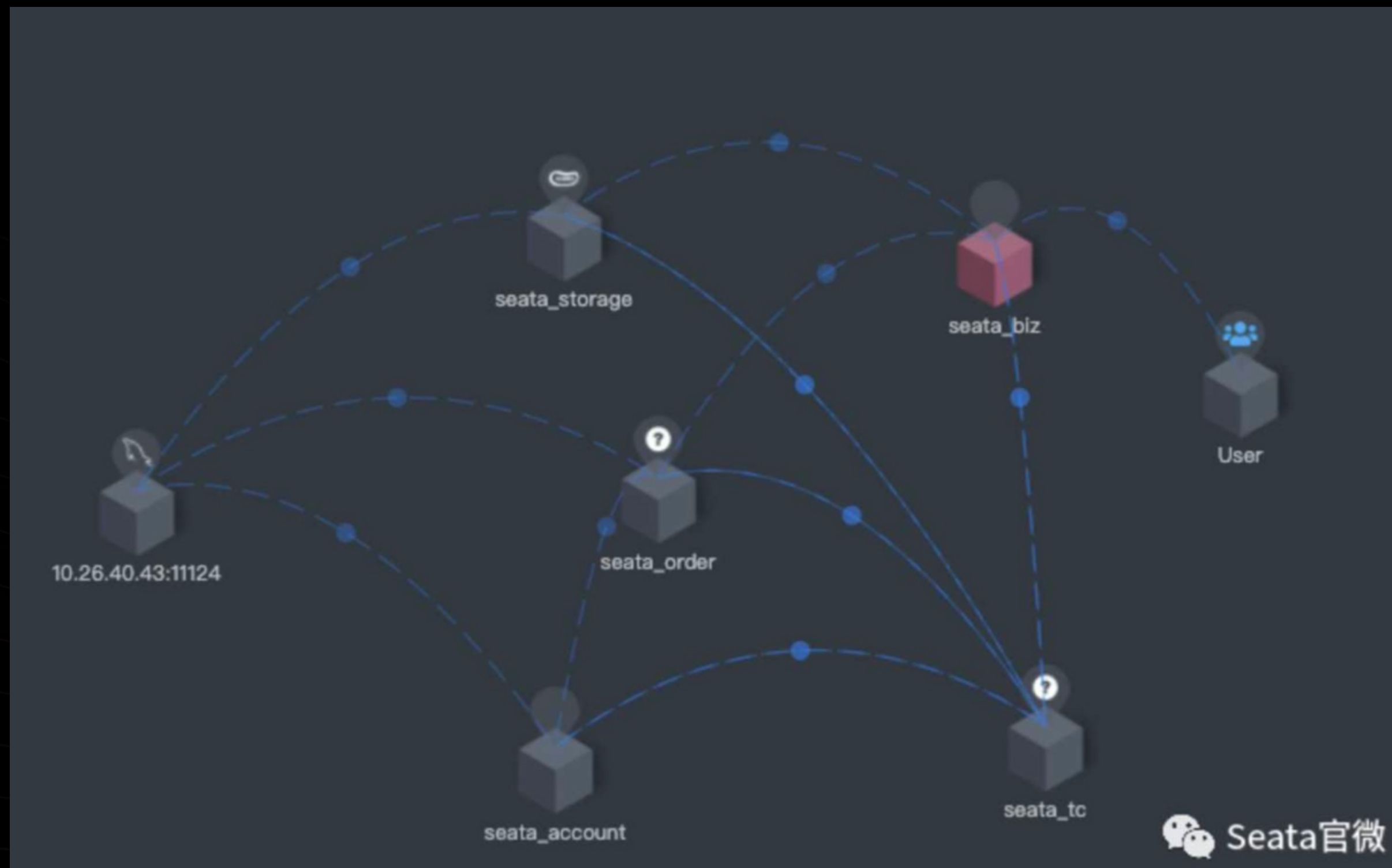
1. 共60+ contributor 参与了代码提交, +60,778 -9,434 .
2. 累计修改文件872, 合并pr 230, 其中新特性pr 30+, 修复问题pr 60+, 优化类pr 100+。
3. 主要新增特性: 支持用户控制台, 支持Skywalking tracing集成, 支持bRPC、EDAS hsf, 支持 kotlin 协程, 支持 TCC幂等/空回滚/防悬挂, 支持分布式任务调度, 支持redis存储lua模式, 支持ON DUPLICATE KEY UPDATE 等语法解析。



Seata 1.5.0

The screenshot shows the Seata web console's TransactionInfo management page. It includes a navigation bar with 'HOME', 'DOCS', 'BLOG', and 'COMMUNITY'. The main content area has a search and filter section with fields for 'CreateTime', 'xid', 'applicationId', and 'status'. Below this is a table with columns: 'xid', 'transactionId', 'applicationId', 'transactionServiceGroup', 'transactionName', 'status', 'timeout', 'beginTime', 'applicationData', and 'operate'. The table currently displays 'No Data'. There are also controls for 'Whether to include branch sessions' and pagination options.

The screenshot displays the Seata transaction tracing interface for a specific transaction. The top shows the transaction ID '1022b80b1f3f4c228af8574115e36365.114.16173424735180001' and its start time '2021-04-02 13:47:53'. The interface includes a tree view on the left and a detailed call stack on the right. The call stack shows the sequence of operations: 'Http - SpringMVC', 'Seata/TM/GlobalBeginRequest', 'Seata/TC/GlobalBeginRequest', 'Mysql/JDBI/PreparedStatement/executeUpda...', 'Seata/TC/GlobalBeginResponse', 'Seata/TM/GlobalBeginResponse', 'io.seata.samples.integration.common.dubb...', 'Mysql/JDBI/PreparedStatement/executeQuer...', 'Mysql/JDBI/PreparedStatement/execute', 'Mysql/JDBI/PreparedStatement/executeQuer...', 'Seata/RM/BranchRegisterRequest', 'Seata/TC/BranchRegisterRequest', 'Mysql/JDBI/PreparedStatement/executeQuer...', 'Mysql/JDBI/PreparedStatement/executeQuer...', 'Mysql/JDBI/PreparedStatement/executeUpda...', 'Mysql/JDBI/Connection/commit', 'Mysql/JDBI/PreparedStatement/executeUpda...', and 'Seata/TC/BranchRegisterResponse'. A timeline at the top right shows the duration of each step.

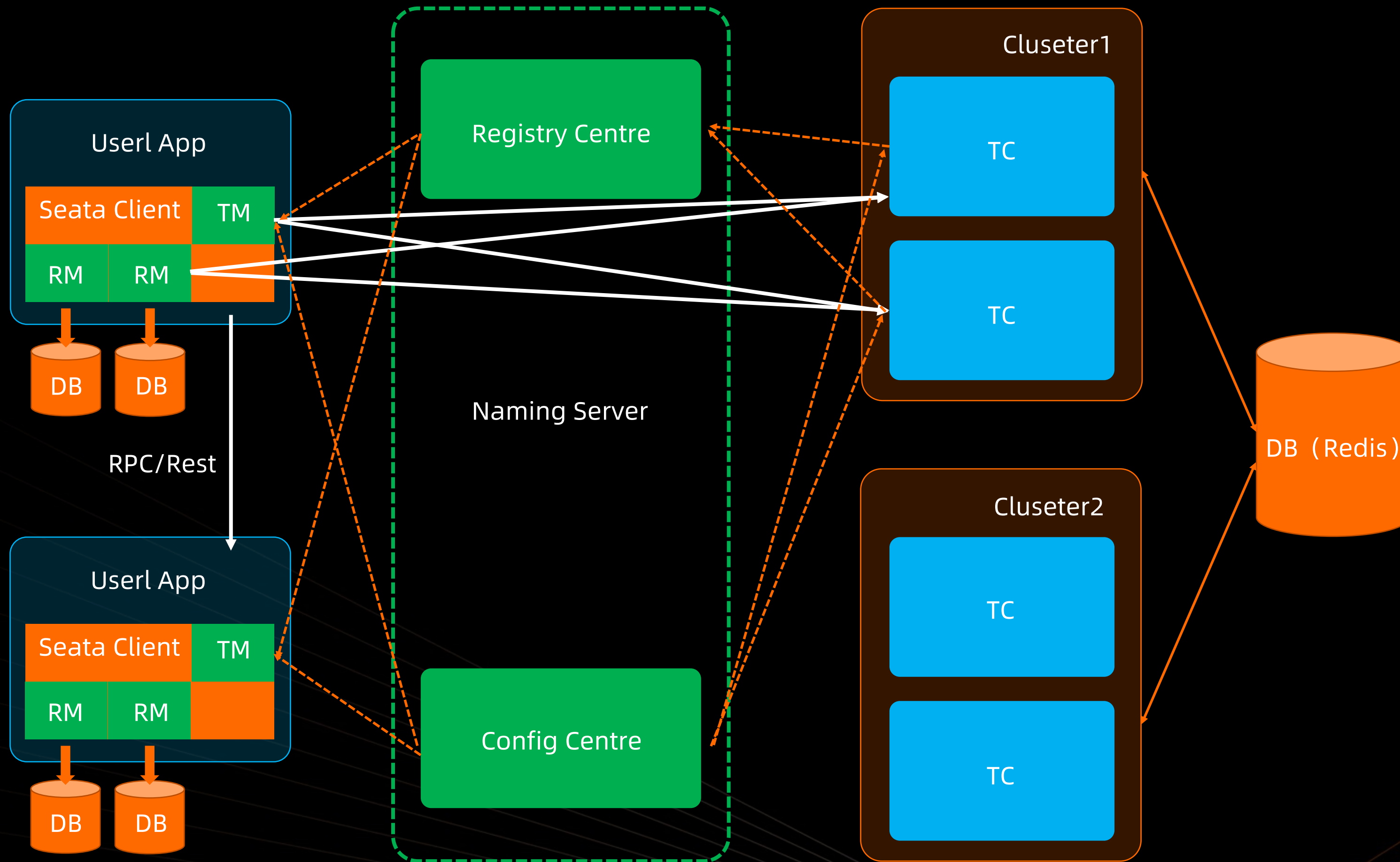


02 使用Seata企业版平滑迁移上云

Seata 企业版迁移上云

Seata 企业版提供**免运维**、**高性能**的企业级分布式事务解决方案，**100%兼容开源**。

目前已在阿里云北京区域上线公测（微服务引擎MSE控制台-分布式事务），**免费开通**。



对比项	MSE Seata 企业版	自建 Seata
效率和性价比	资源全托管、免运维，内置依赖的注册中心和配置中心，支持故障节点自动下线	不支持，需自行购买各种资源搭建系统，运维升级需投入精力，人工成本高
	资源全托管、免运维，内置依赖的注册中心和配置中心，支持故障节点自动下线	不支持可视化运维操作，需要根据不同的注册中心和配置中心类型，手动添加事务规则和事务路由配置，手动修改多节点的元数据配置文件，易出错
性能提升	深度调优，比开源 Seata 1.5.x 内核性能提高 30%	不支持，需自行调优
可观测性	提供监控大盘，可对事务量、事务处理 RT、正常/异常事务、事务 TPS 等指标进行监控	不支持，需自行搭建监控和告警体系
高可用和安全	多可用区部署，故障自动检测及恢复	不支持，需要人工处理，自行设计
	支持 RAM 鉴权，RAM 用户（即子账号）可管理实例访问权限	不支持

Seata 企业版 VS 开源版

自建版1.5.0:

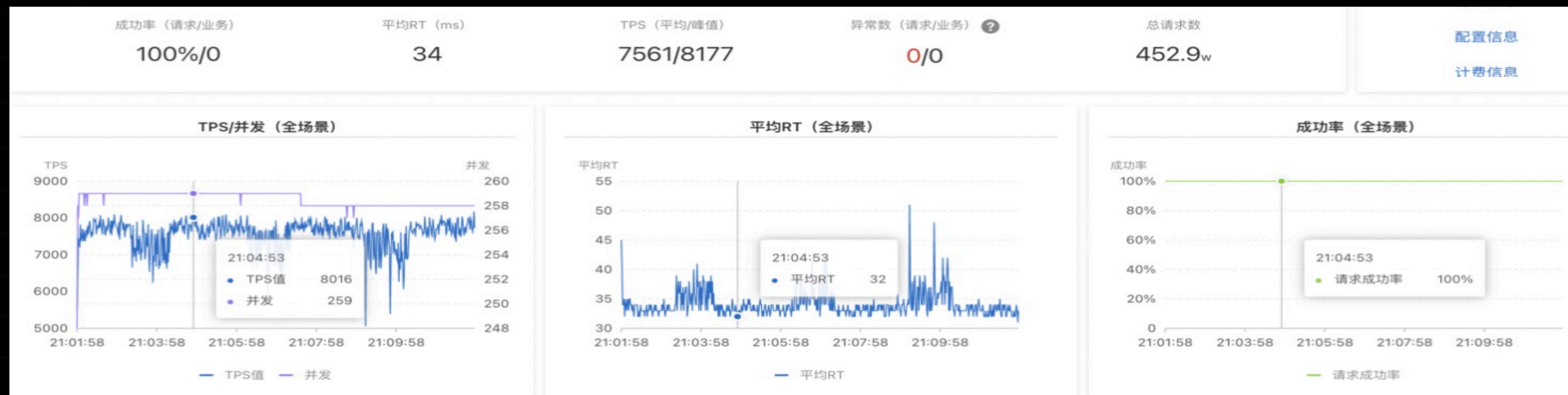


企业版相比开源版内核:

RT 降低

20%+

企业版1.5.0:



TPS 提升

30%+

Seata 企业版开通

步骤一：创建Seata实例

微服务引擎MSE / 实例列表

实例列表

[创建Seata实例](#) ①

实例ID/名称	运行状态	版本	NamingServer地址	规格	创建时间	到期时间	操作
s-xxxxxx	运行中	1.5.0-20220509-1	xxxxxx	1C2G	2022-05-17 13:05:42	2022-08-20 23:59:59	管理 释放

每页显示 10 共1条 < 上一页 1 下一页 >

步骤二：创建事务分组

步骤三：复制配置到应用

微服务引擎MSE / 事务分组列表

事务分组列表

[创建事务分组](#) ②

事务分组ID	对应实例	创建时间	操作
xxxxxx	xxxxxx	2022-05-17 13:09:52	显示配置 ③ 监控 删除

每页显示 10 共1条 < 上一页 1 下一页 >

MSE-Seata 企业用户支持群
146人

扫一扫群二维码，立刻加入该群。

阿里云