



---

# Service Mesh的延伸 之论道Database Mesh

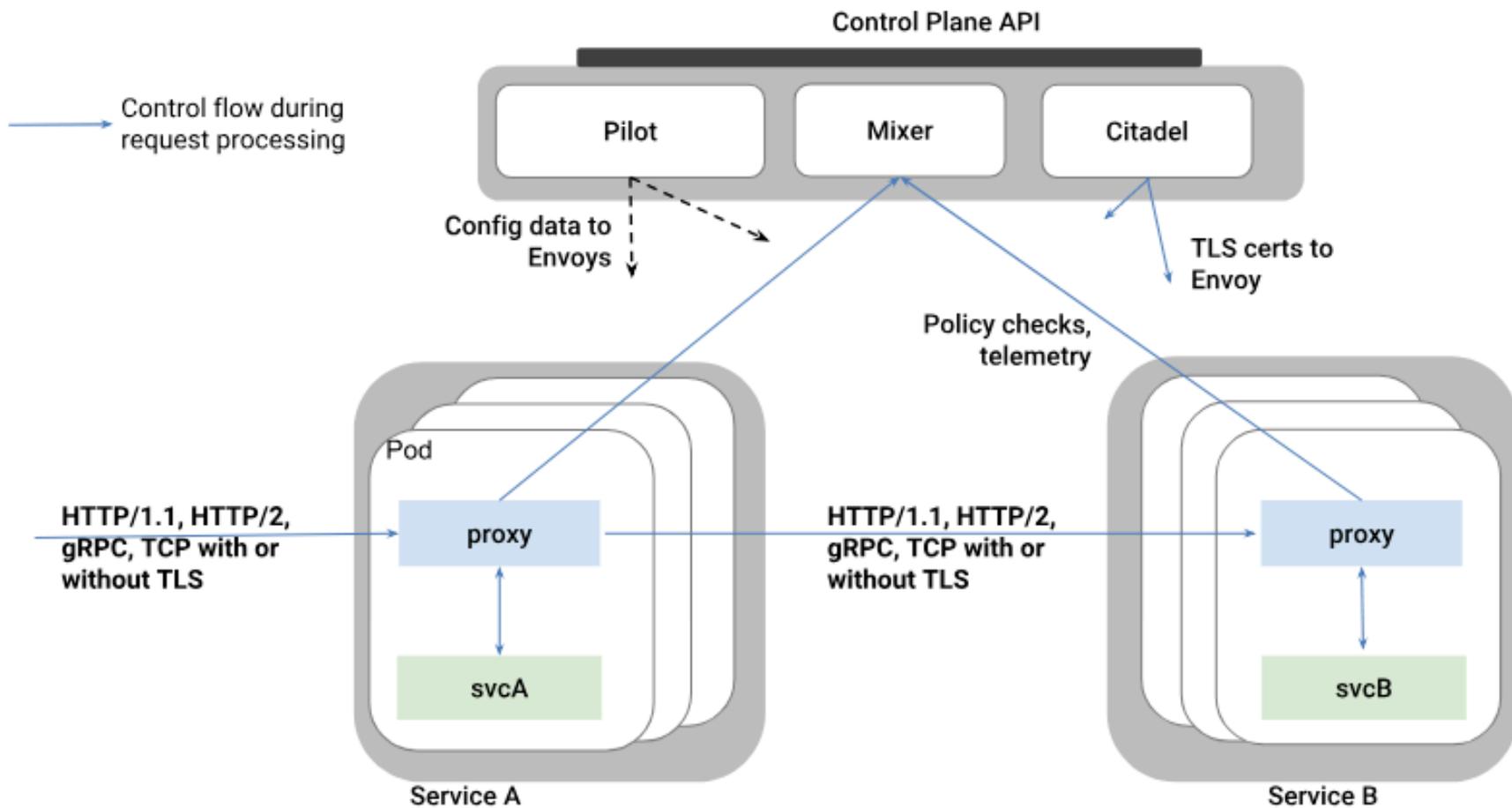
---

分享人：张亮

日期：2018年07月25日



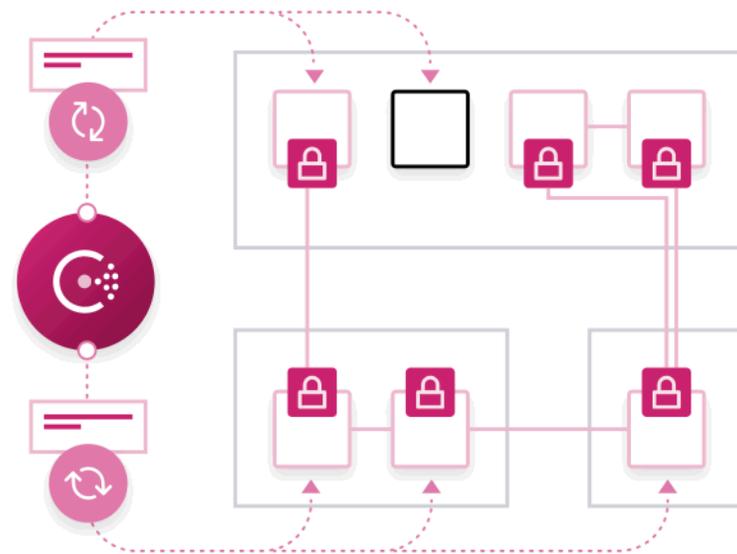
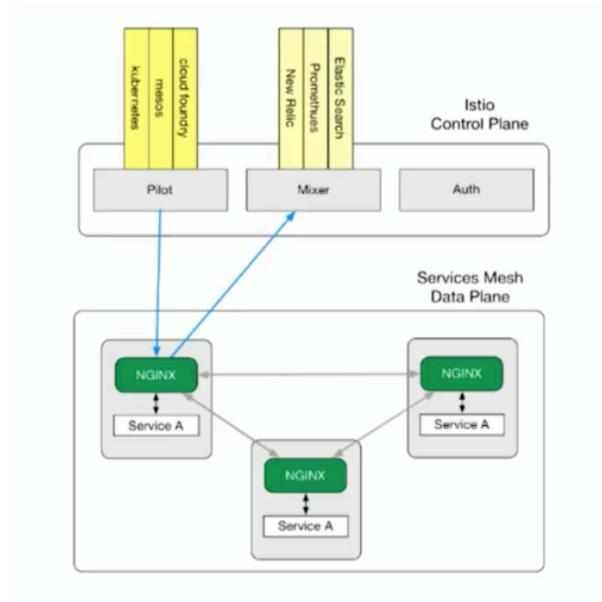
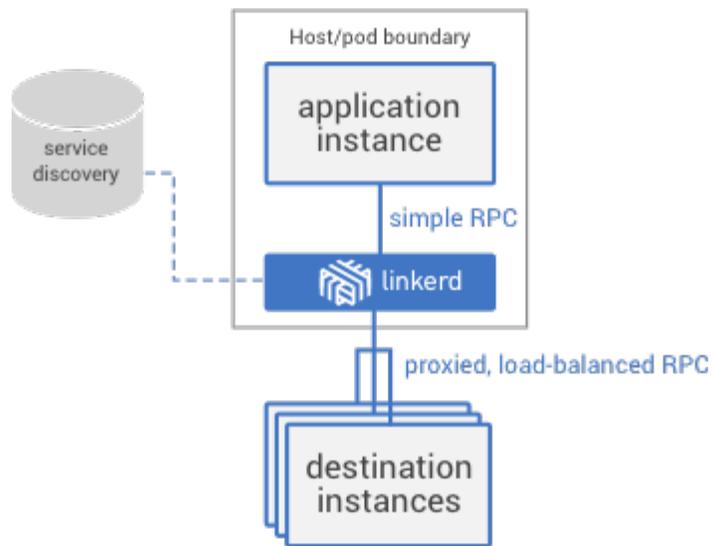
# Service Mesh风头正劲



# Service Mesh产品多样化



京东金融  
JD Finance





# Service Mesh的优势

---



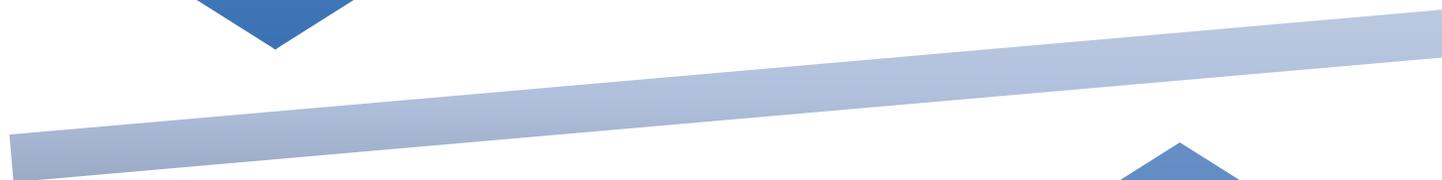


# 服务化之后，数据库怎么办？



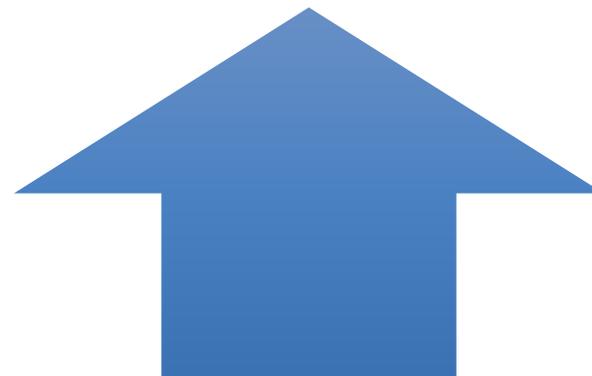
## 服务

- 无状态
- 根据规则路由
- 业务方处理事务



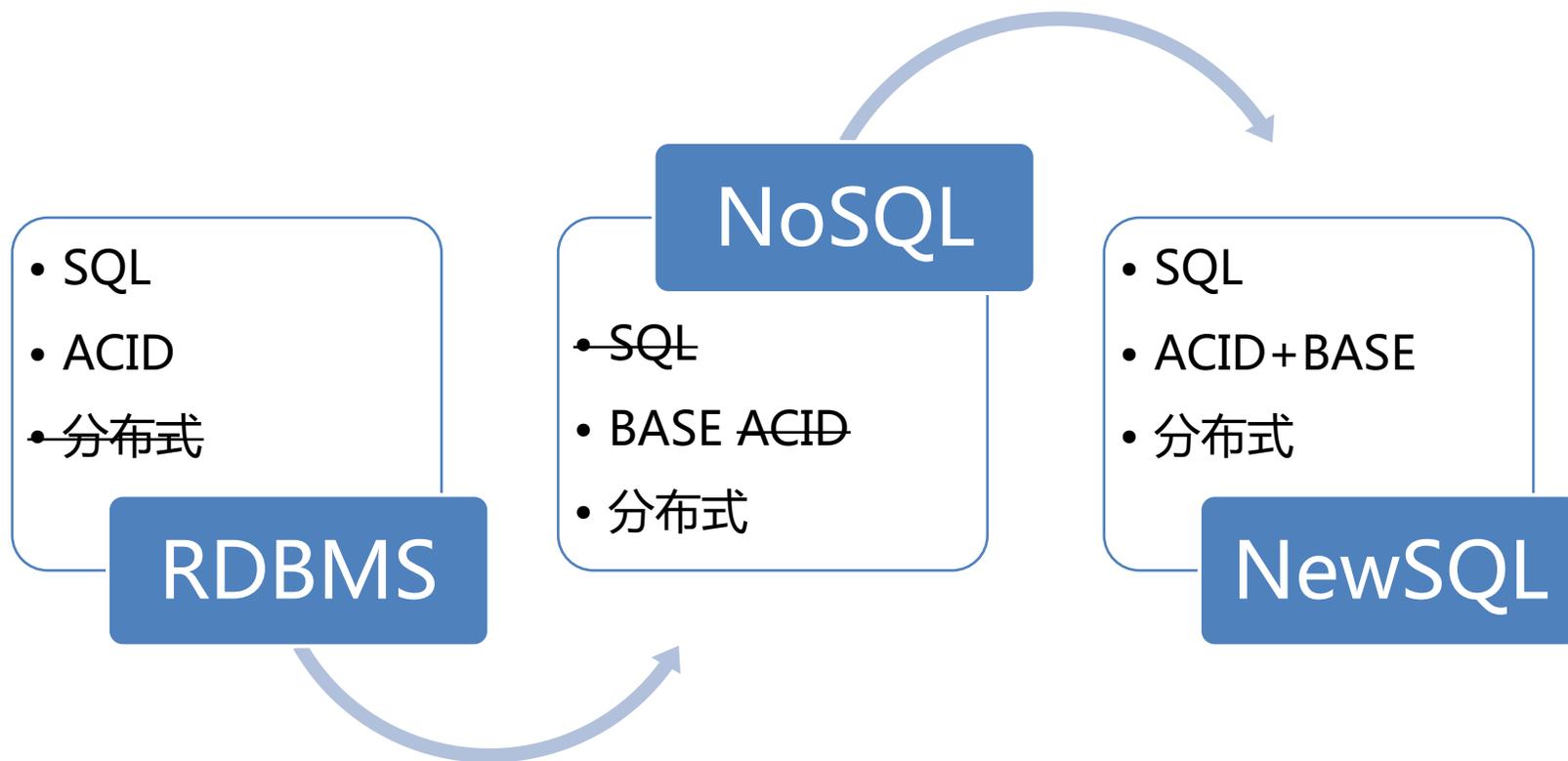
## 数据库

- 有状态
- 根据SQL路由
- 数据库自动处理事务





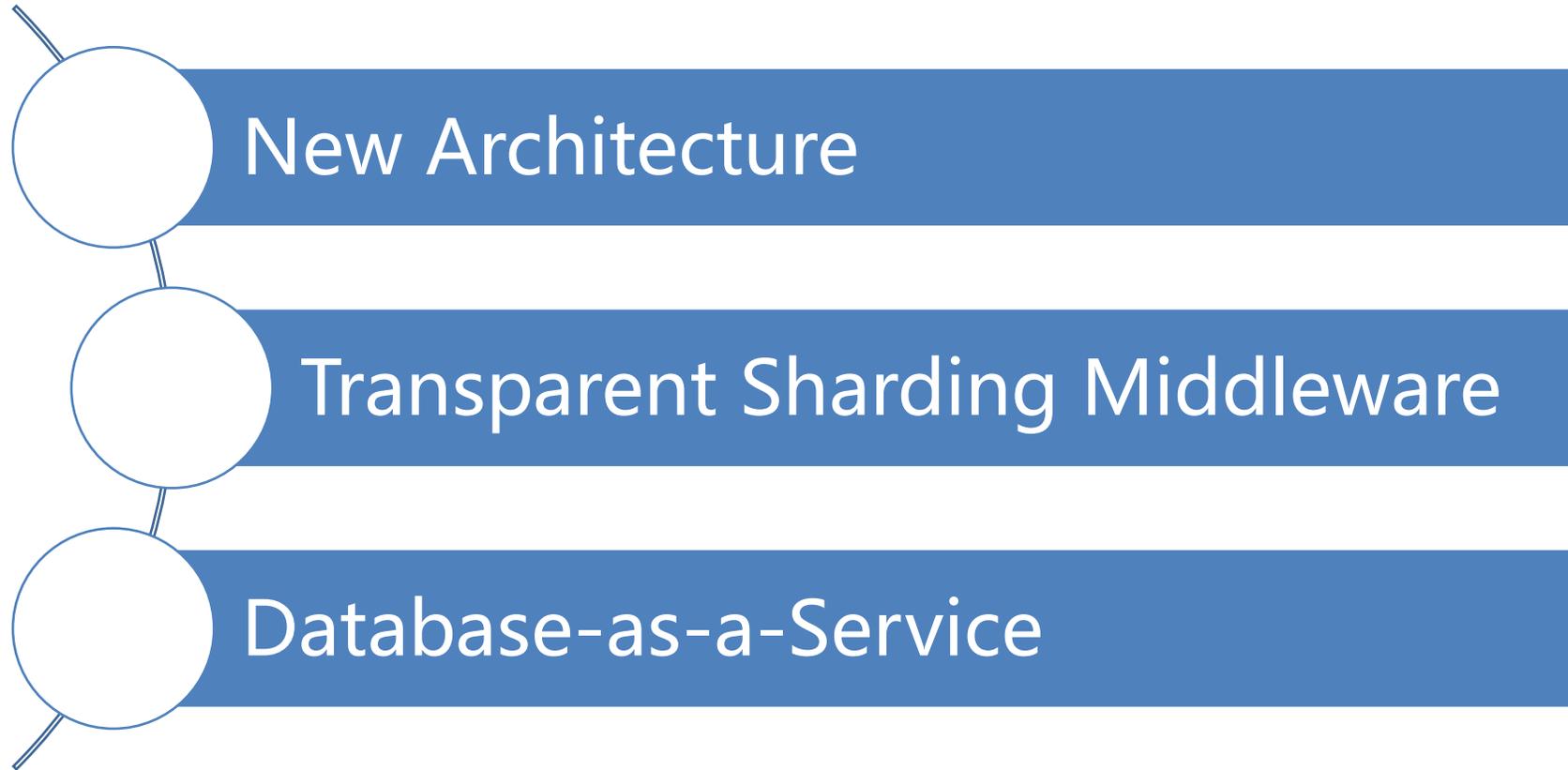
# 数据库的进化趋势





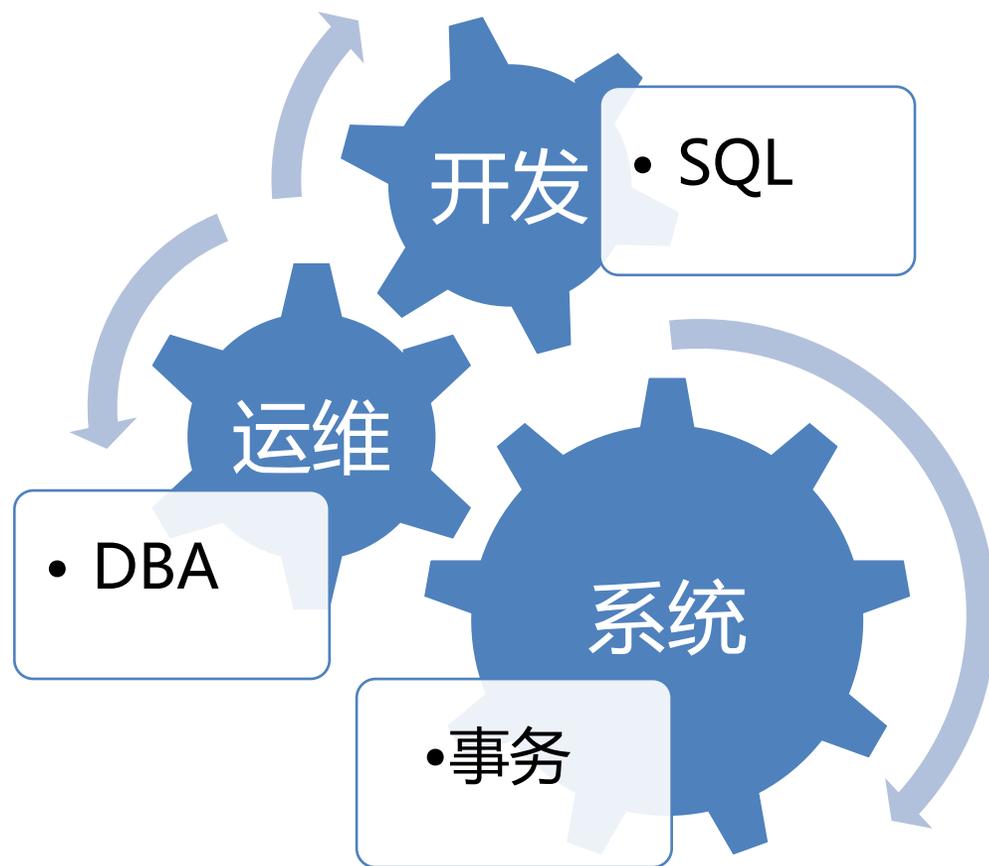
# NewSQL的分类

What's Really New with NewSQL?



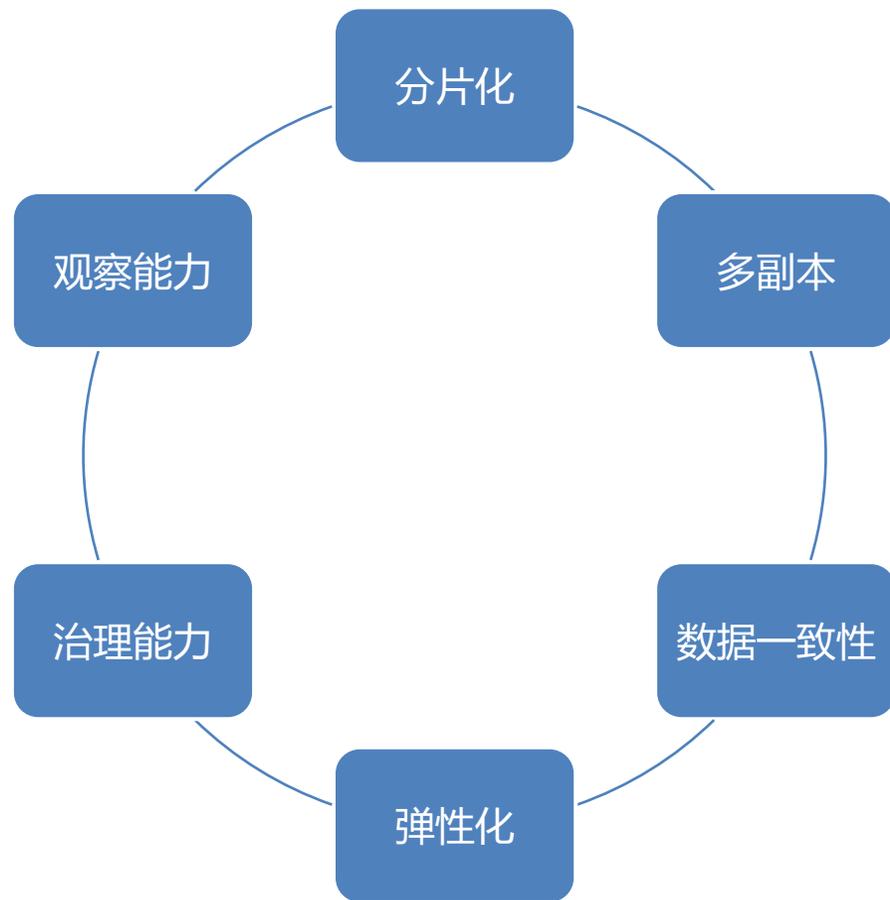


# 数据库中间层的优势



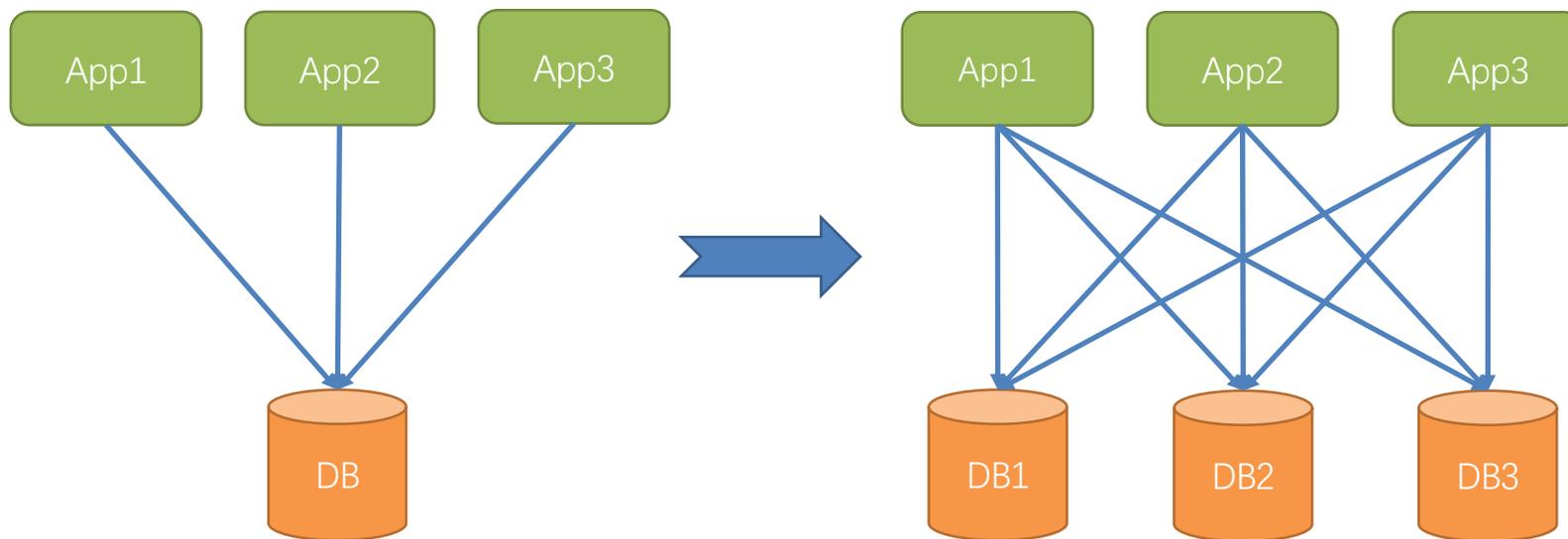


# 数据库中间层应具备的能力



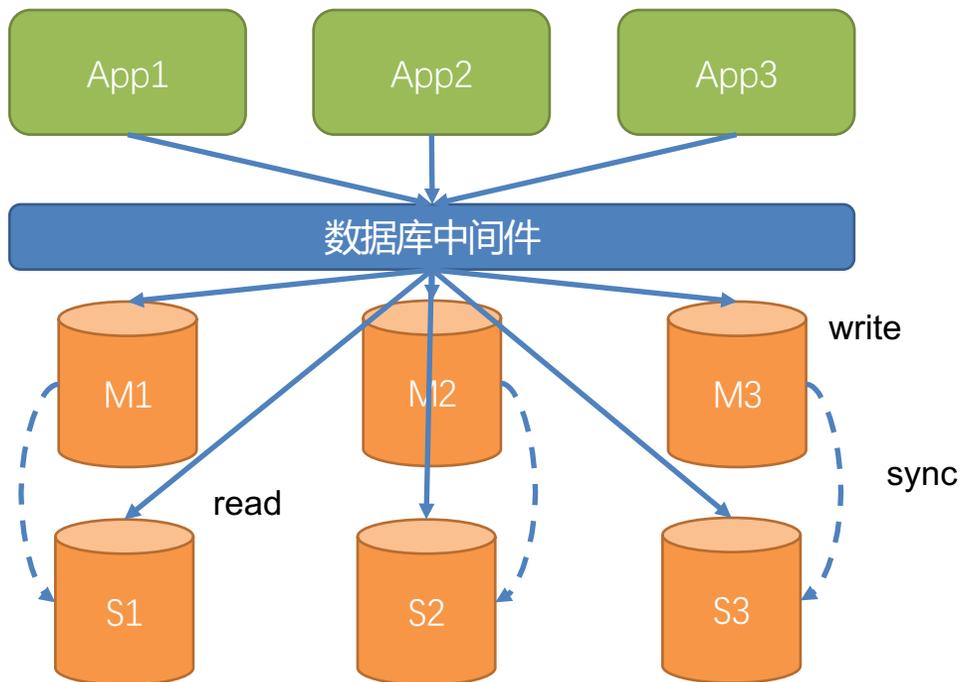


# 数据分片



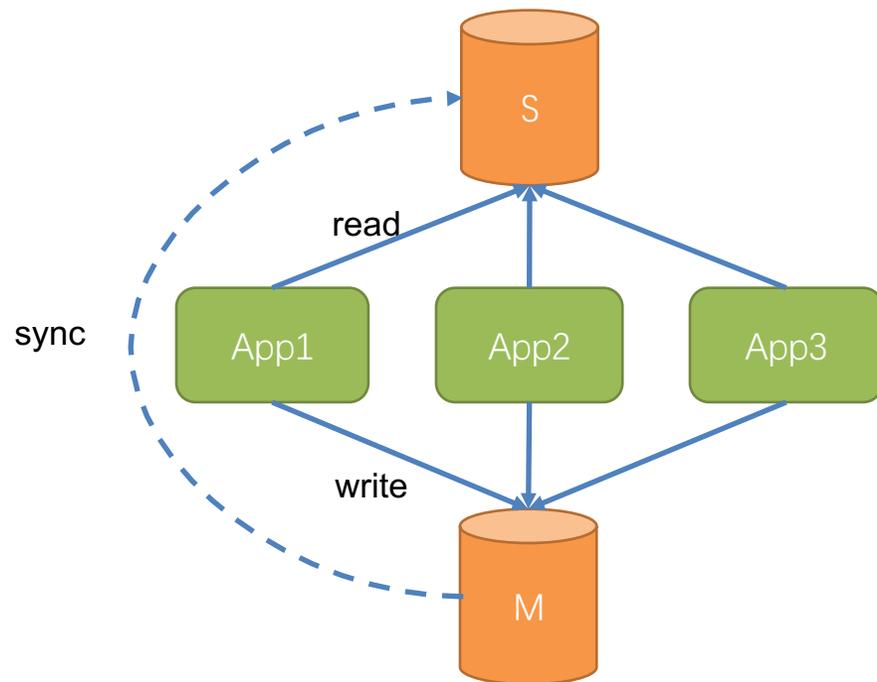


# 数据分片：引入中间件



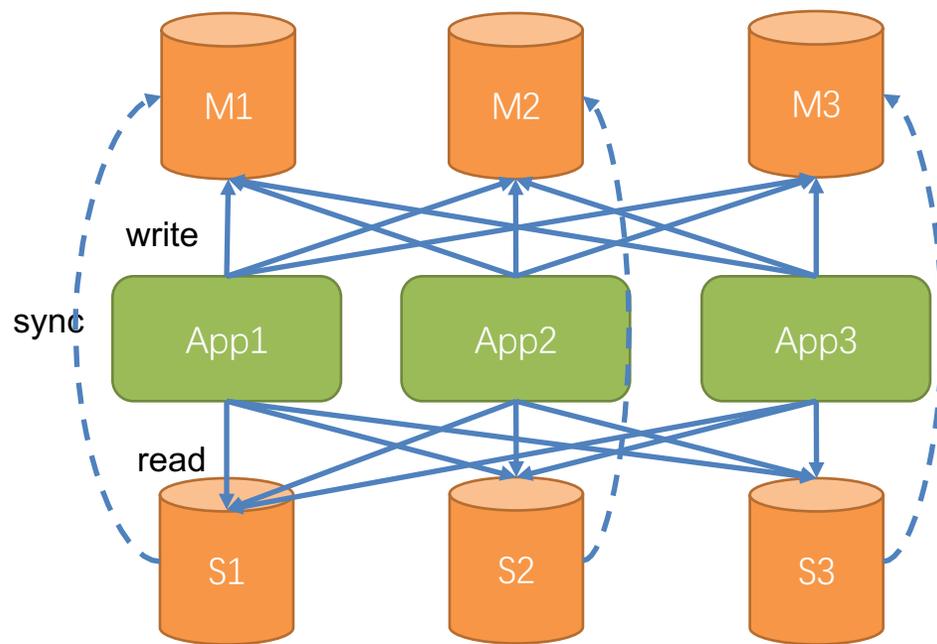


# 读写分离



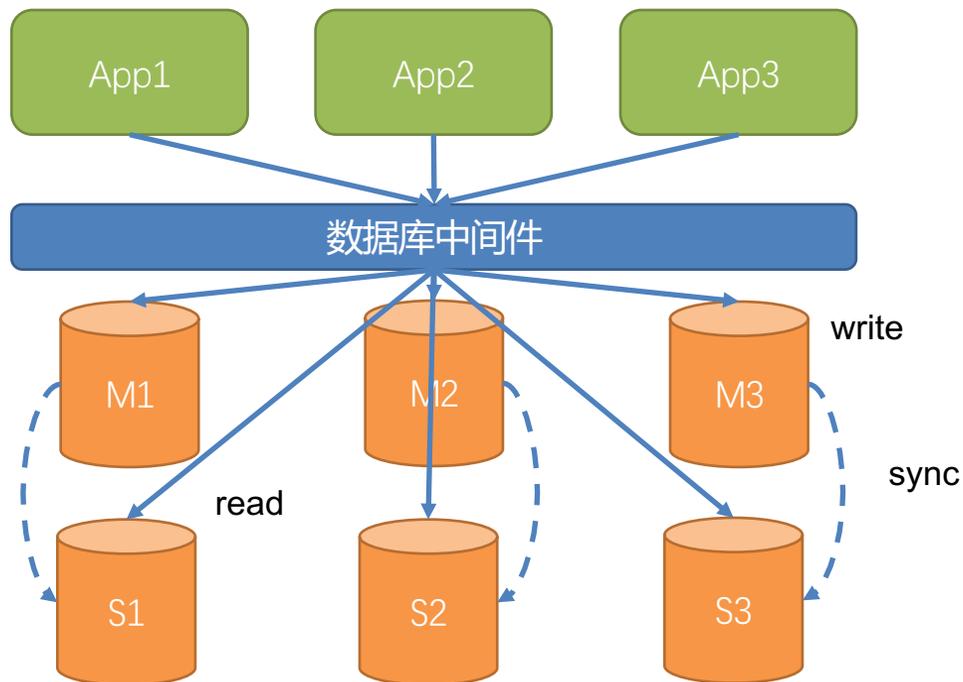


# 数据分片 + 读写分离





# 数据分片 + 读写分离：引入中间件





# 分布式事务：定义

## 传统事务：ACID

Atomicity - 原子性

Consistency - 一致性

Isolation - 隔离性

Durability - 持久性

## 柔性事务：BASE

Basically Available - 基本可用

Soft state - 软状态

Eventual consistency - 最终一致性

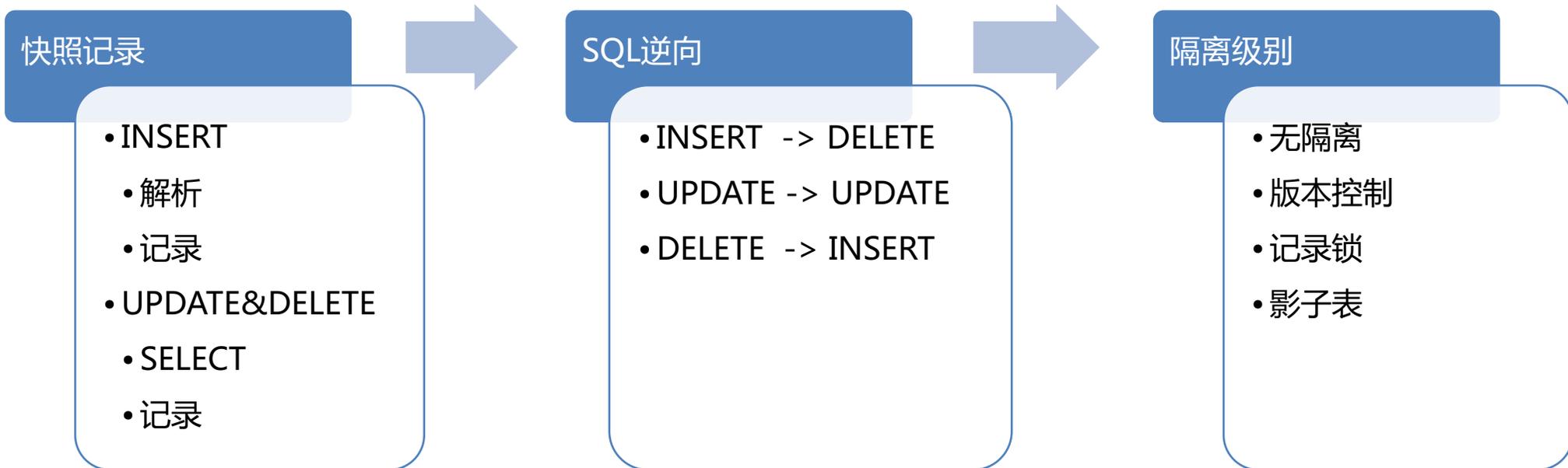


# 分布式事务：分类

	XA	最大努力送达	Saga	TCC
业务改造	无	无	实现补偿接口	实现TCC接口
回滚	支持	不支持	支持	支持
一致性	强一致	最终一致	最终一致	最终一致
隔离性	原生支持	不支持	不支持	Try接口支持
并发性能	严重衰退	无影响	无影响	略微衰退
适合场景	短事务 并发较低	事务最终成功 高并发	长事务 应用方控制并发 高并发	长事务 高并发

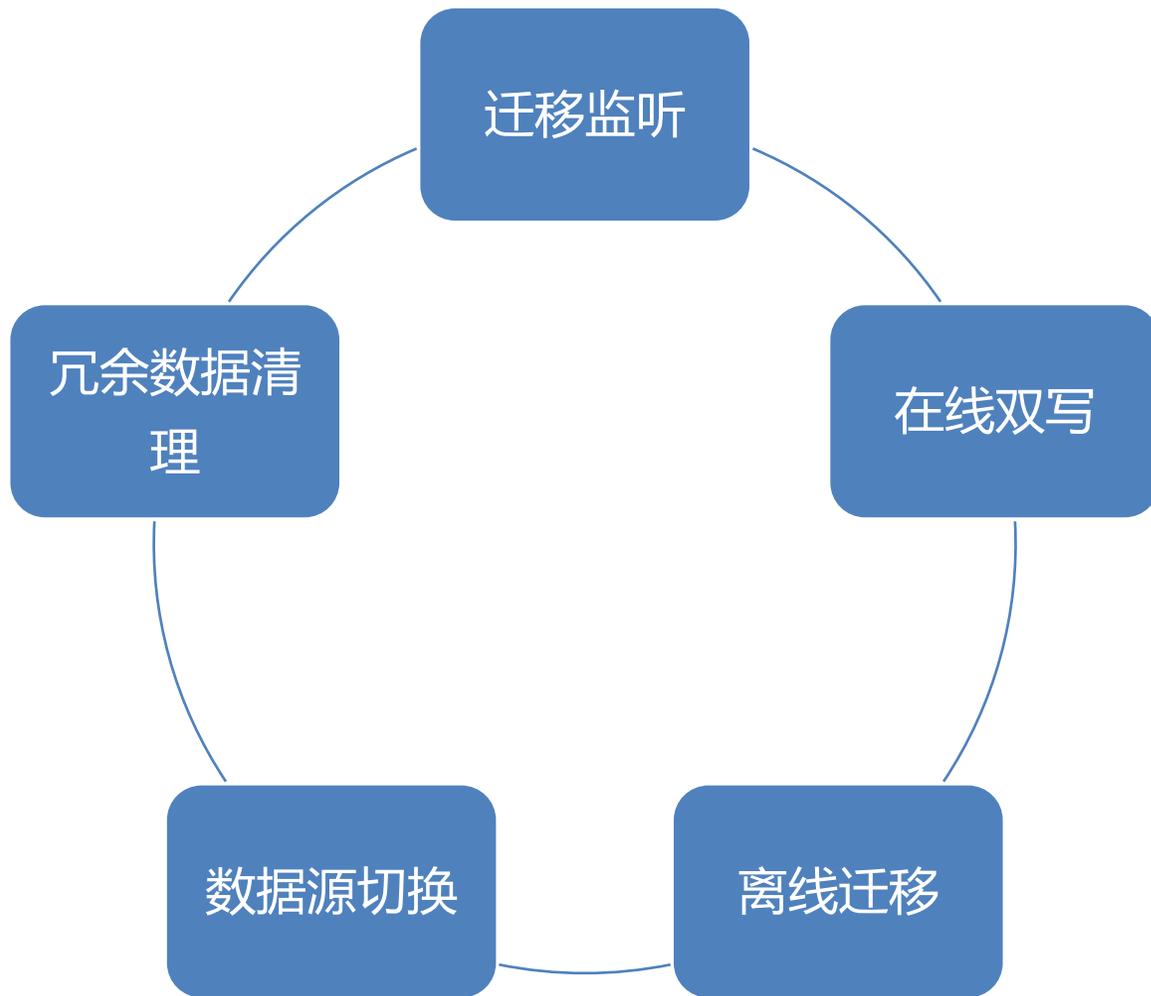


# 分布式事务：柔性事务自动化



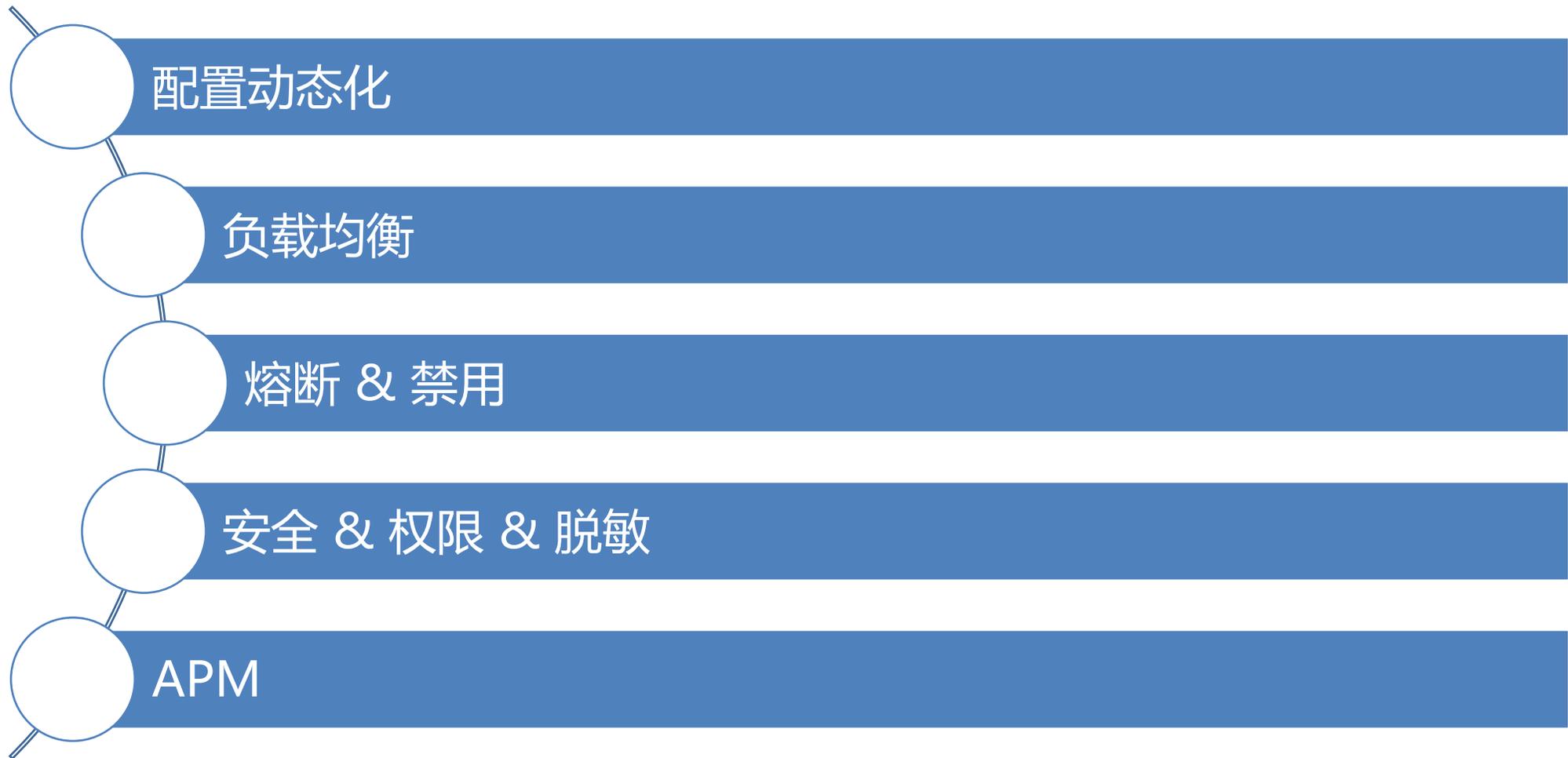


# 弹性伸缩：数据迁移





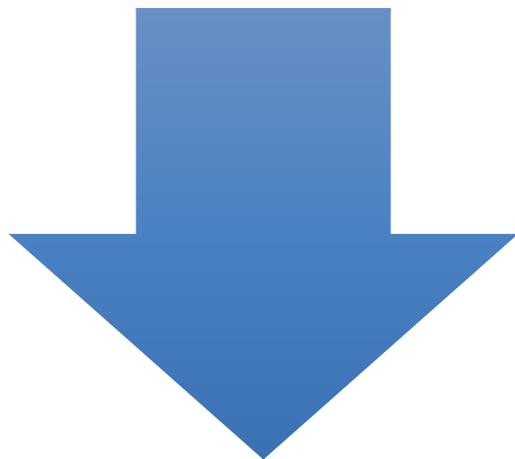
# 治理监控





# 现有实现方案对比

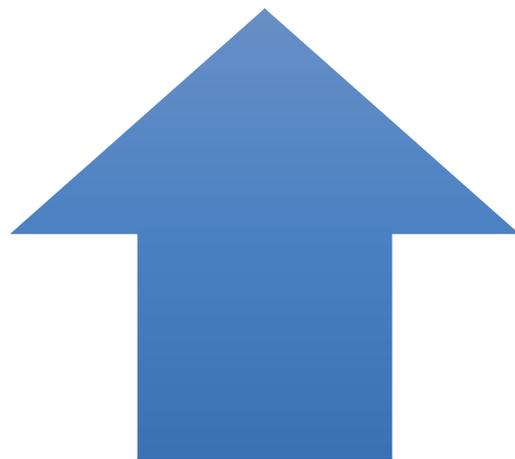
## 客户端



- 支持任意数据库
- 数据库连接数占用高
- 仅支持单一开发语言
- 性能损耗低
- 无中心化

## 代理端

- 仅支持单一数据库
- 数据库连接数占用低
- 支持任意开发语言
- 性能损耗高



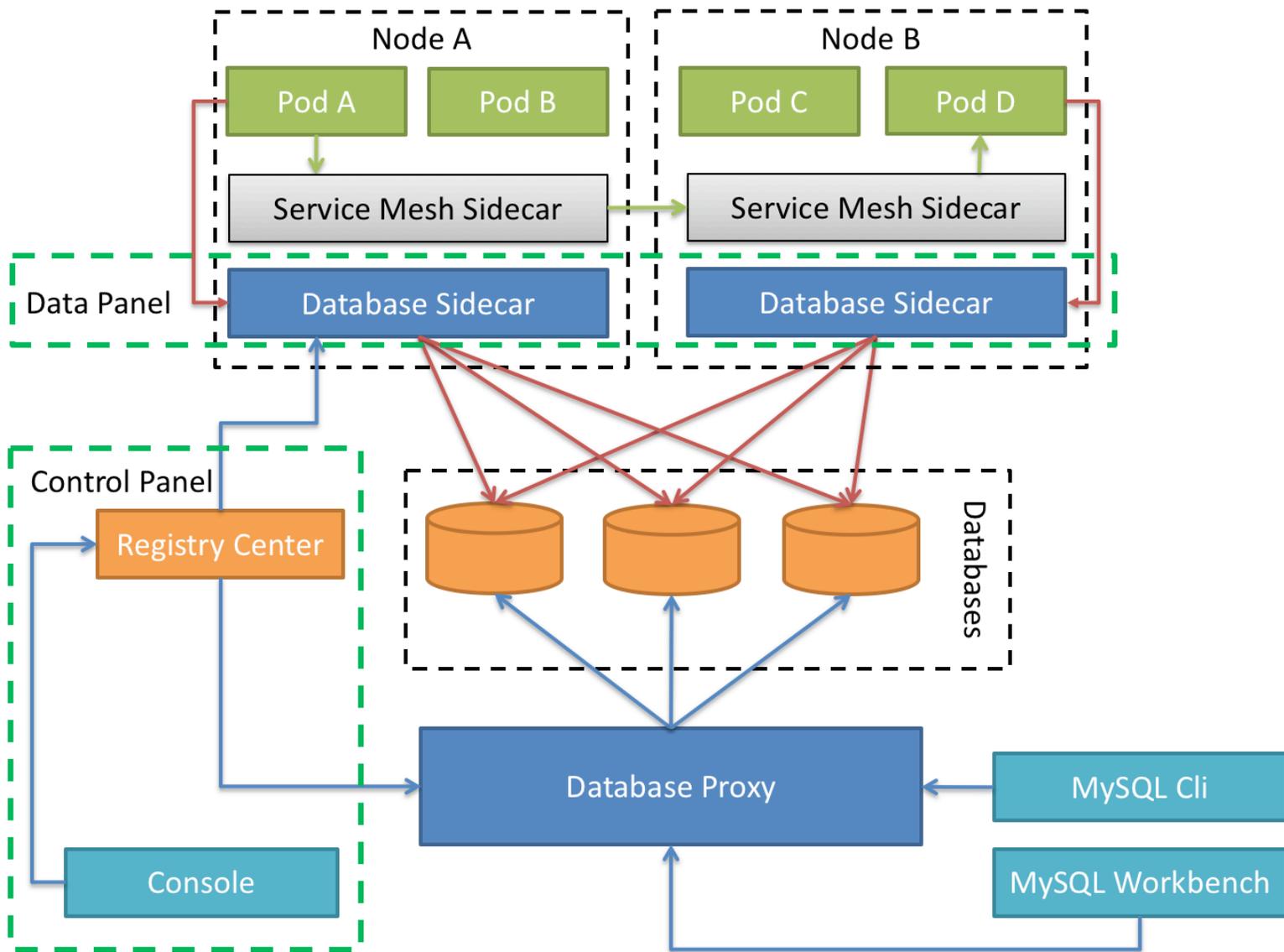
# Sidecar的优势



	JDBC	Proxy	Sidecar
数据库	任意	单一	单一
连接数	高	低	高
异构语言	仅Java	任意	任意
性能	损耗低	损耗略高	损耗低
无中心化	是	否	是
静态入口	无	有	无



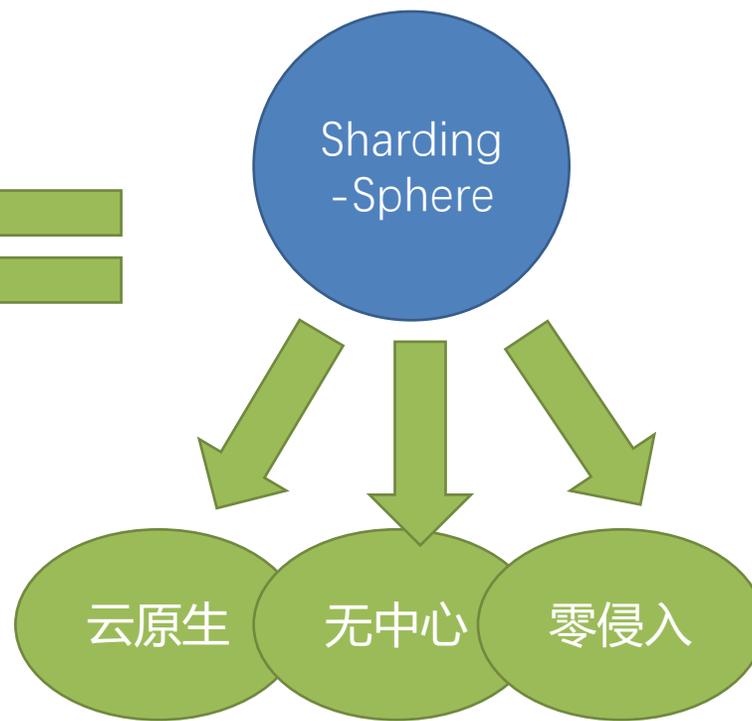
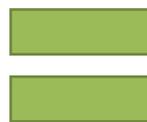
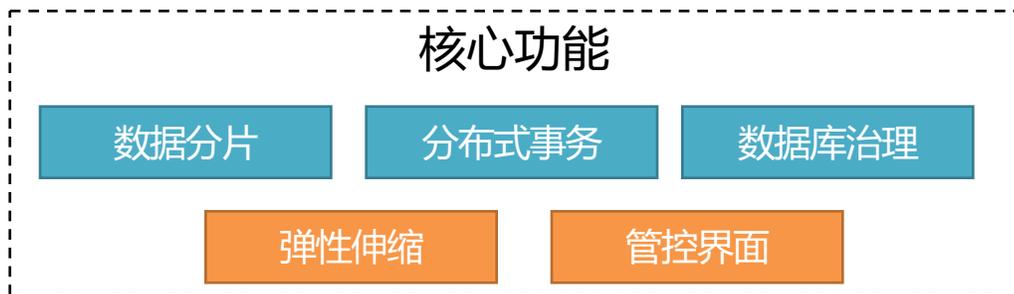
# Database Mesh架构图



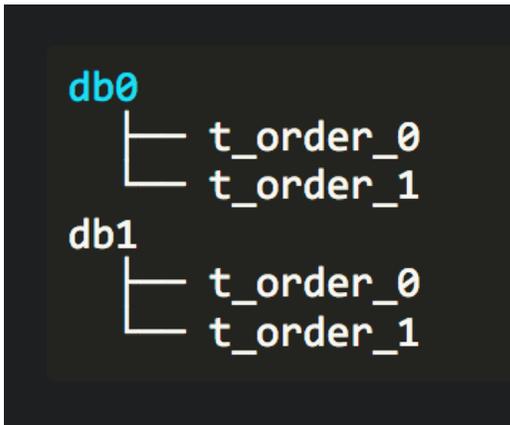
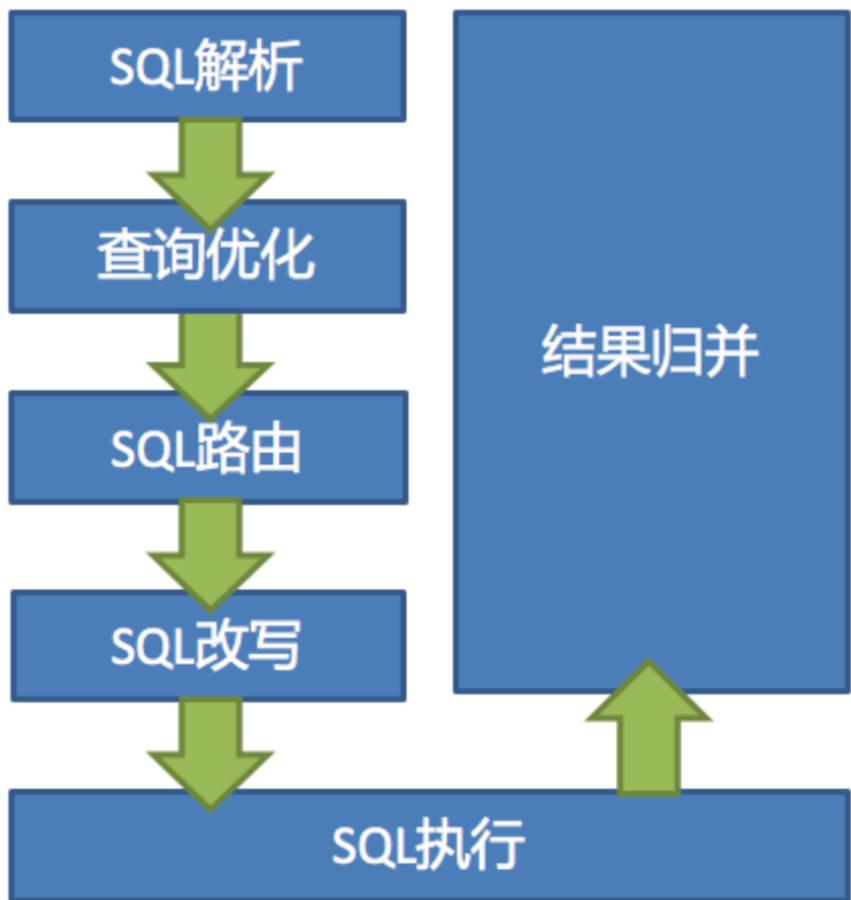
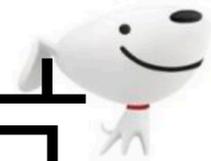
# Sharding-Sphere



京东金融  
JD Finance



# Sharding-Sphere : 数据分片

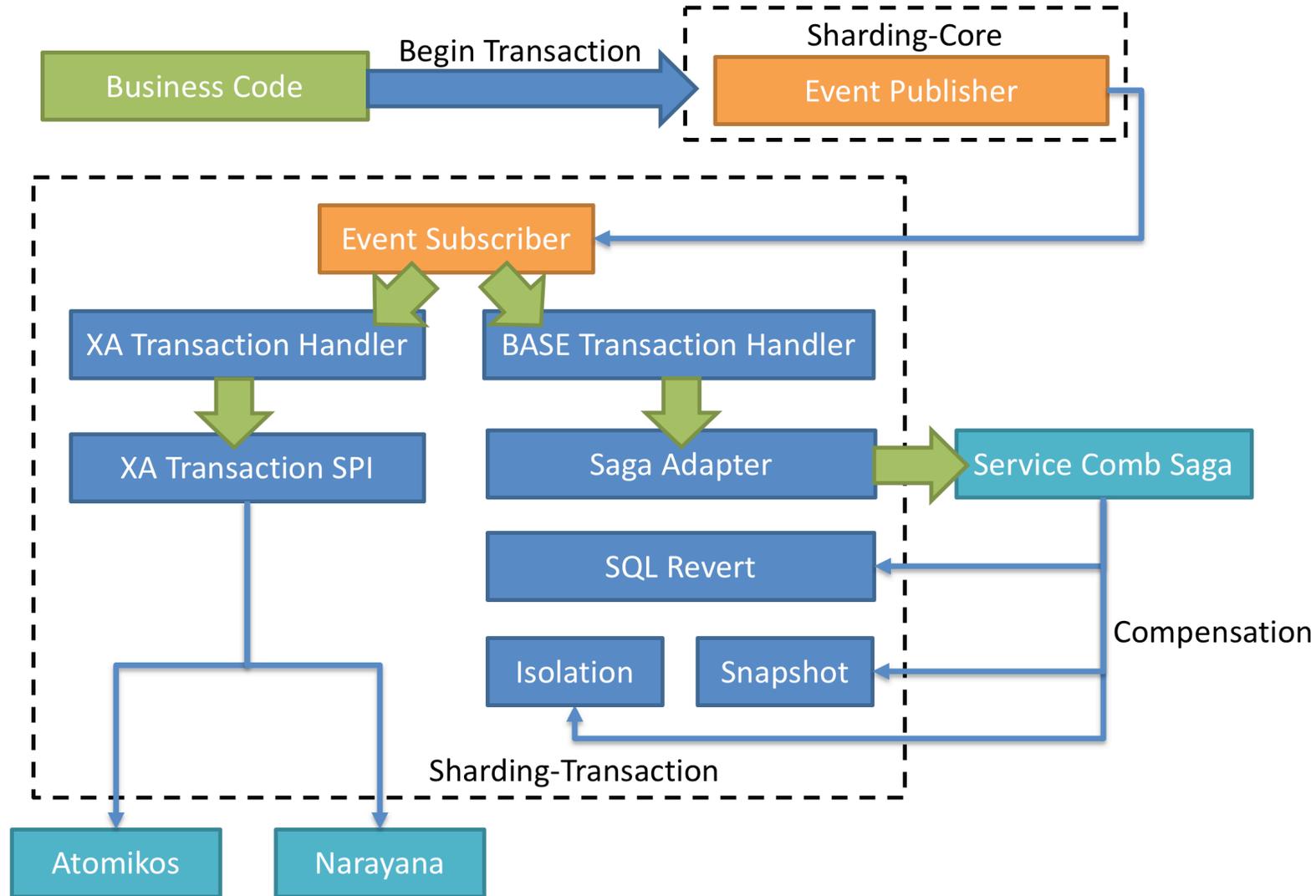


```
select * from t_order where user_id = ? and order_id = ?;
```



```
select * from db0.t_order_0 where user_id = ? and order_id = ?;
```

# Sharding-Sphere : 分布式事务



# Sharding-Sphere : 治理中心



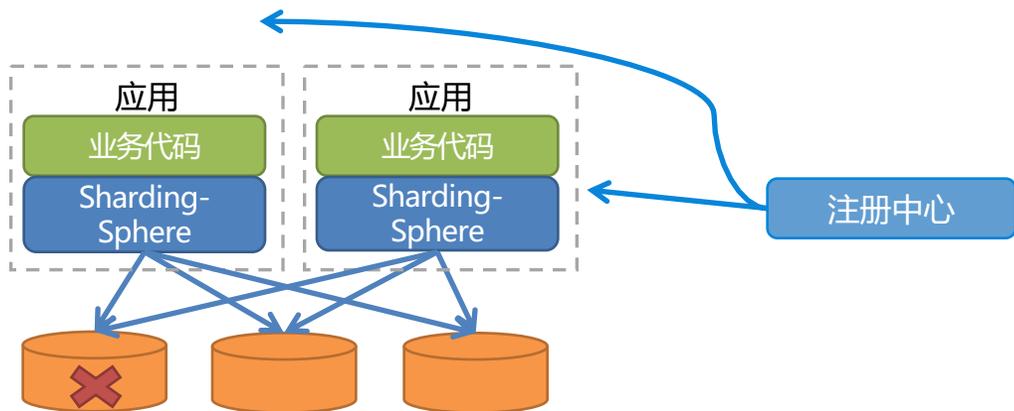
京东金融  
JD Finance

## 核心功能

- 配置集中化 & 动态化
- 数据库熔断 & 禁用

## 支持的注册中心

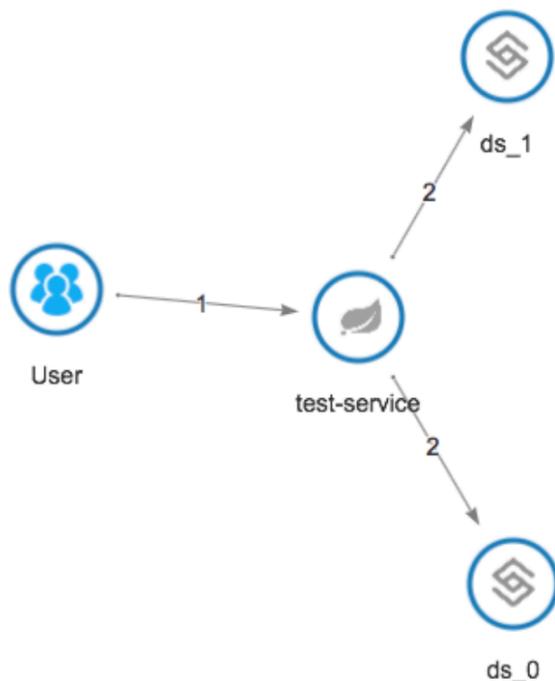
- ZooKeeper
- Etc



# Sharding-Sphere : APM



## 拓扑图展示



## Span Info

OperationName: /SJDBC/BRANCH/QUERY  
2017/10/20 09:37:18 605 - 2017/10/20 09:37:18 612

### Tags

#### span type

Exit

#### component

ShardingJDBC

#### peer

ds\_1

#### db.type

sql

#### db.instance

ds\_1

#### db.statement

```
SELECT i.* FROM t_order_0 o JOIN t_order_item_0 i ON o.order_id=i.order_id WHERE o.user_id in (?,?) AND o.order_id in (?,?)
```

#### db.bind\_vars

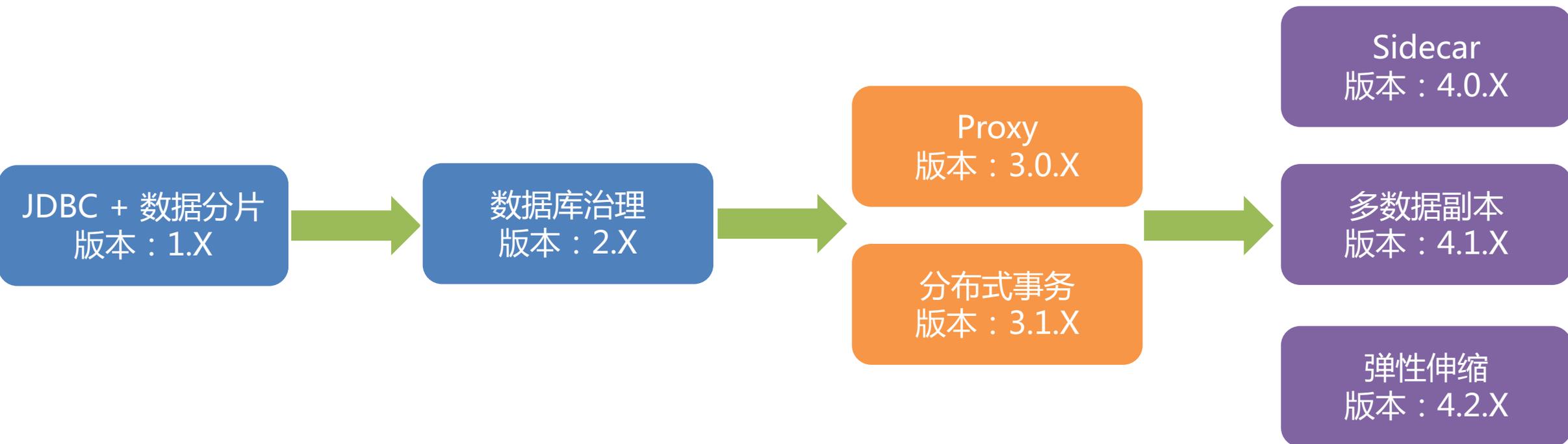
10,11,1001,1100

#### is error

false



# 演进线路图



图例:

已完成

进行中

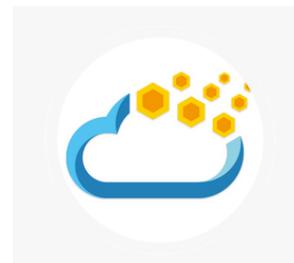
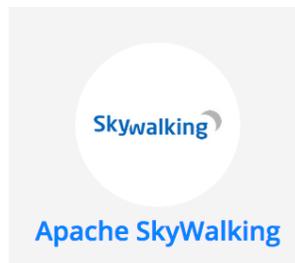
规划中

# 联合社区打造开源解决方案

---



京东金融  
JD Finance



# Sharding-Sphere : 官网



京东金融  
JD Finance

<http://shardingsphere.io/>

The screenshot shows the Sharding-Sphere website homepage. At the top left is the Sharding-Sphere logo and name. To its right are navigation links: '文档' (Documentation), '源码' (Source Code), and '社区' (Community). The '源码' link is active, showing a dropdown menu with 'Github' and '码云' (Gitee). On the far right are language options: '中文' (Chinese) and 'EN' (English). The main content area features the Sharding-Sphere logo and the text 'Sharding-Sphere' and '开源 快速 共享' (Open Source, Fast, Shared). Below this is a paragraph describing the project as an ecosystem of open-source distributed database middleware solutions, including Sharding-JDBC, Sharding-Proxy, and Sharding-Sidecar. A '了解更多' (Learn More) button is located at the bottom left.

# Sharding-Sphere : 官档



京东金融  
JD Finance

<http://shardingsphere.io/document/current/>



1. 概览

2. 快速入门

3. 概念 & 功能

4. 使用手册

5. FAQ

Sharding-Sphere > 概览

Release v2.0.3 Star 4k Fork 1750 Watch 651

Sharding-Sphere是一套开源的分布式数据库中间件解决方案组成的生态圈，它由 Sharding-JDBC、Sharding-Proxy和Sharding-Sidecar这3款相互独立的产品组成。他们均提供标准化的数据分片、读写分离、柔性事务和数据治理功能，可适用于如Java同构、异构语言、容器、云原生等各种多样化的应用场景。

Sharding-Sphere定位为关系型数据库中间件，旨在充分合理地在分布式的场景下利用关系型数据库的计算和存储能力，而非实现一个全新的关系型数据库。它与NoSQL和NewSQL是并存而非互斥的关系。NoSQL和NewSQL作为新技术探索的前沿，放眼未来，拥抱变化，是非常值得推荐的。反之，也可以用另一种思路看待问题，放眼未来，关注不变的东西，进而抓住事物本质。关系型数据库当今依然占有巨大市场，是各个公司核心业务的基石，未来也难于撼动，我们目前阶段更加关注在原有基础上的增量，而非颠覆。

license Apache 2 chat on gitter

maven central 2.0.3 release v2.0.3

build passing codecov 79% OpenTracing-1.0 enabled Skywalking Tracing enable

核心功能

# Sharding-Sphere : Github



京东金融  
JD Finance

https://github.com/sharding-sphere/sharding-sphere



## Sharding-Sphere

Distributed database middleware

<http://shardingsphere.io/> <https://github.com/sharding-sphere>

Repositories 3 | People 9 | Teams 0 | Projects 0

sharding-sphere / sharding-sphere

Watch 692 | Unstar 4,688 | Fork 1,820

Code | Issues 32 | Pull requests 1 | Projects 0 | Wiki | Insights

Distributed database middleware

transaction | shard-databases | jdbc | microservice | cloud-native | distributed-database | orchestration

4,260 commits | 5 branches | 29 releases | 30 contributors | Apache-2.0

Branch: dev | New pull request

Create new file | Upload files | Find file | Clone or download

Commit	Message	Time
terrymanu	Merge pull request #951 from ma-xiao-guang-64/dev	Latest commit b8332a8 a day ago
.github	upgrade issue template	a month ago
sharding-core	Merge pull request #951 from ma-xiao-guang-64/dev	a day ago
sharding-jdbc-orchestration-spring	move getRawDataSourceMap and getMasterSlaveConfiguration static metho...	a month ago
sharding-jdbc-orchestration	remove json files.	a day ago

## sharding-sphere-example

Sharding-Sphere examples

Java 162 | 146 | Apache-2.0 | Updated 42 minutes ago

## sharding-sphere

Distributed database middleware

microservice | jdbc | transaction | orchestration

Java 4,373 | 1,750 | Apache-2.0 | Updated 9 hours ago

## sharding-sphere-doc

bsite & documents

53 | Updated 3 days

# Sharding-Sphere : 社区建设



京东金融  
JD Finance

<http://shardingsphere.io/community/>

The screenshot shows the Sharding-Sphere community website. The left sidebar contains a navigation menu with the following items: 1. 组织, 2. 参与和贡献 (with sub-items: 贡献者指南, 提交者指南, 2FA, 开发规范, 外部依赖, 测试引擎), and 3. 采用公司. The main content area is titled '开发规范' (Development Guidelines) and includes sections for '开发理念' (Development Philosophy) and '代码提交行为规范' (Code Submission Behavior Guidelines). The '开发理念' section contains one bullet point: '用心写代码, 追求整洁和优雅到极致的极简代码, 认同《重构》和《代码整洁之道》的理念。'. The '代码提交行为规范' section contains five bullet points: '确保通过全部测试用例。', '确保覆盖率不低于dev分支。', '确保使用Checkstyle检查代码, 违反验证规则的需要有特殊理由。模板位置在 `sharding-jdbc/src/resources/sharding_checks.xml`, 请使用checkstyle 8.8运行规则。', '确保执行 `mvn clean install` 可以编译和测试通过。', and '及时删除无用代码。'. Below this is the '编码规范' (Coding Guidelines) section, which contains six bullet points: '使用linux换行符。', '缩进 (包含空行) 和上一行保持一致。', '不应有无意义的空行。', '日志与注释一律使用英文。', '注释只能包含javadoc, todo和fixme。', and '变量命名要有意义。返回值变量使用result命名; 循环中使用each命名循环变量; map中使用entry代替'.

Sharding-Sphere > 参与和贡献 > 开发规范

## 开发理念

- 用心写代码, 追求整洁和优雅到极致的极简代码, 认同《重构》和《代码整洁之道》的理念。

## 代码提交行为规范

- 确保通过全部测试用例。
- 确保覆盖率不低于dev分支。
- 确保使用Checkstyle检查代码, 违反验证规则的需要有特殊理由。模板位置在 `sharding-jdbc/src/resources/sharding_checks.xml`, 请使用checkstyle 8.8运行规则。
- 确保执行 `mvn clean install` 可以编译和测试通过。
- 及时删除无用代码。

## 编码规范

- 使用linux换行符。
- 缩进 (包含空行) 和上一行保持一致。
- 不应有无意义的空行。
- 日志与注释一律使用英文。
- 注释只能包含javadoc, todo和fixme。
- 变量命名要有意义。返回值变量使用result命名; 循环中使用each命名循环变量; map中使用entry代替

# Sharding-Sphere : 使用案例



京东金融  
JD Finance

<http://shardingsphere.io/community/cn/company/>

Sharding-Sphere

1. 组织
2. 参与和贡献
3. 采用公司

Chinese

Sharding-Sphere > 采用公司

## 电子商务

- 当当
- 搜好货
- 唯品会
- TCL
- 男衣库
- 义乌购
- 京东商城俄文站
- 京东商城海外站

## 互联网金融

- 小牛在线
- 华夏信财
- 你我金融
- 海尔消费金融
- 我爱卡
- 随手记
- 美利金融
- 团贷网
- 京东金融

## 信息资讯

- 搜狐资讯
- 凤凰汽车
- 天涯社区
- 易车网
- 百姓网

# 欢迎加入我们



感谢您的支持与关注  
我们将不忘初心，不负众望！

Sharding-Sphere



长按 请扫码识别关注



长按 请指纹识别关注