



—— 开源与架构技术沙龙-北京站 ——

# 基于支付场景的微服务实战

讲师名称：程超

名字：程超

网名：小程故事多

合著作品：《深入分布式缓存--从原理到实践》

个人简书博客：<https://www.jianshu.com/u/a9b2d43bb94e>

# 目录

---

SOA与微服务

01

03

基于微服务的架构改造

02 老支付架构所遇到的挑战

04

未来计划要做的一些事

# 1

Part 01

SOA与微服务

---



## 麦田理论

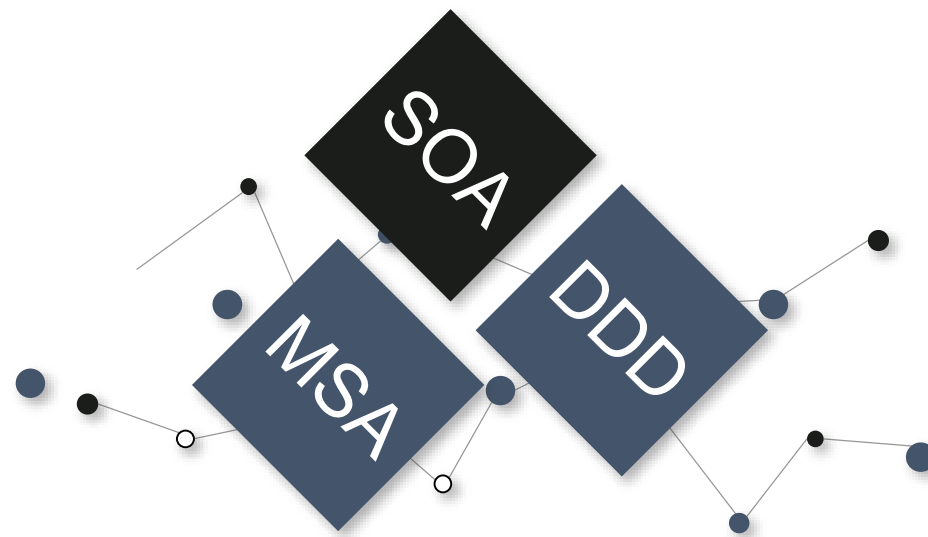


表面看来一块土地得到了充分利用，实际上各农作物得不到充分的光照和适宜的营养，如此一来加大了后期除草、松土、施肥、灌溉及收割的成本。



一整块地根据需要分配为若干大小土地块，每块地之间清晰分界，这样就有了玉米地、土豆地、豆角地，再想种什么划块地再耕作就可以了。专业技术人员施肥；玉米，豆角和土豆分离，避免豆角藤爬上玉米，缠绕玉米不能自由生长。土豆又汲取玉米需要的营养物质等等问题。

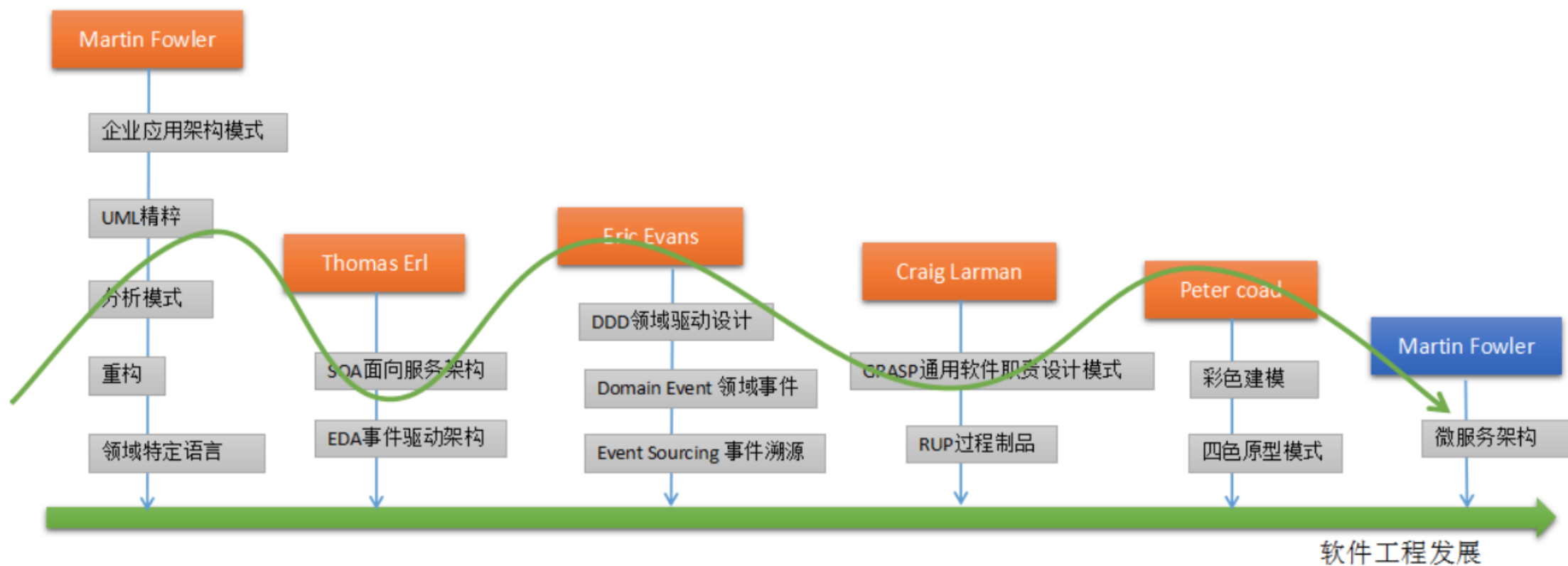
SOA的提出是在企业计算领域，就是要将紧耦合的系统，划分为面向业务的，粗粒度，松耦合，无状态的服务



微服务的特点：

- 1、组件化与多服务
- 2、围绕业务组建团队
- 3、做产品而不是做项目
- 4、智能端点与弱管道
- 5、去中心化管理数据
- 6、基础设施自动化

贫血模型  
充血模型  
限界上下文  
实体、值对象  
领域建模





# 2

Part 02

老支付架构所遇到的挑战

---





## 判断项目好坏的二个角度



## 从代码和设计角度来看

- 1、由不合理的代码所引起的项目无扩展性
- 2、数据库经常发生死锁
- 3、数据库事务乱用，导致事务占用时间过长
- 4、代码容错能力很差，经常因为考虑不足引起事故
- 5、程序中打印的大量的无用日志，并且引起性能问题
- 6、常用配置信息依然从数据库中读取
- 8、滥用线程池，造成栈和堆溢出
- 9、从库中查询数据，每次全部查出
- 10、业务代码研发不考虑幂等操作
- 11、使用缓存不合理，存在惊群效应、缓存穿透等情况
- 12、代码上下流流程定义混乱
- 13、异常处理机制混乱

## 从整体架构角度来看

- 1、整体依然使用单体集群架构
- 2、采用单机房服务器布署方式
- 3、采用Nginx+hessian的方式实现服务化。
- 4、业务架构划分不彻底，边界模糊
- 5、项目拆分不彻底，一个tomcat共用多个应用
- 5、无故障降级策略
- 6、监控系统不合理（网络、系统）
- 7、支付运营报表，大数据量查询
- 8、运维手动打包，手动上线
- 9、系统扩容手动布署

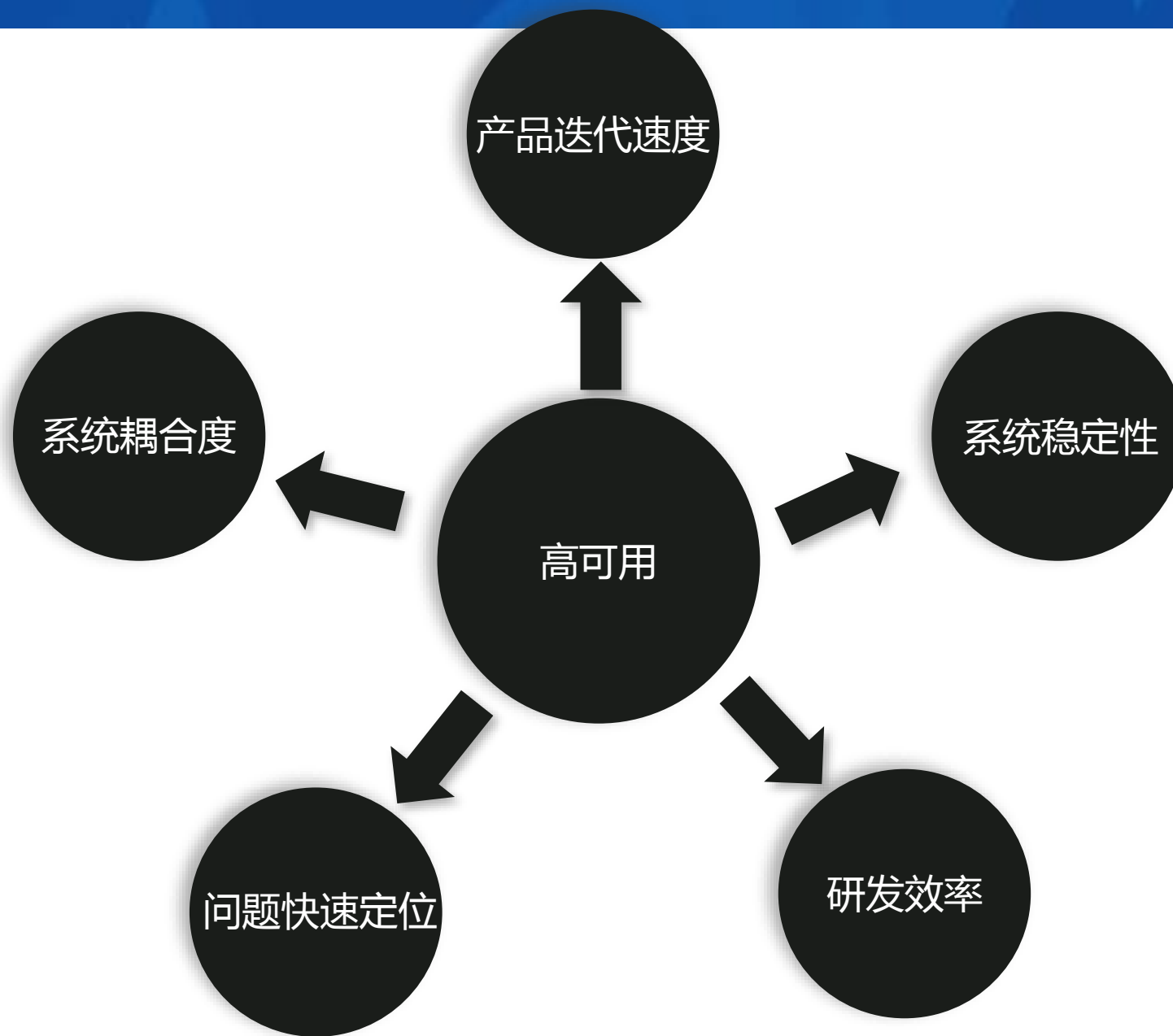
# 3

Part 03

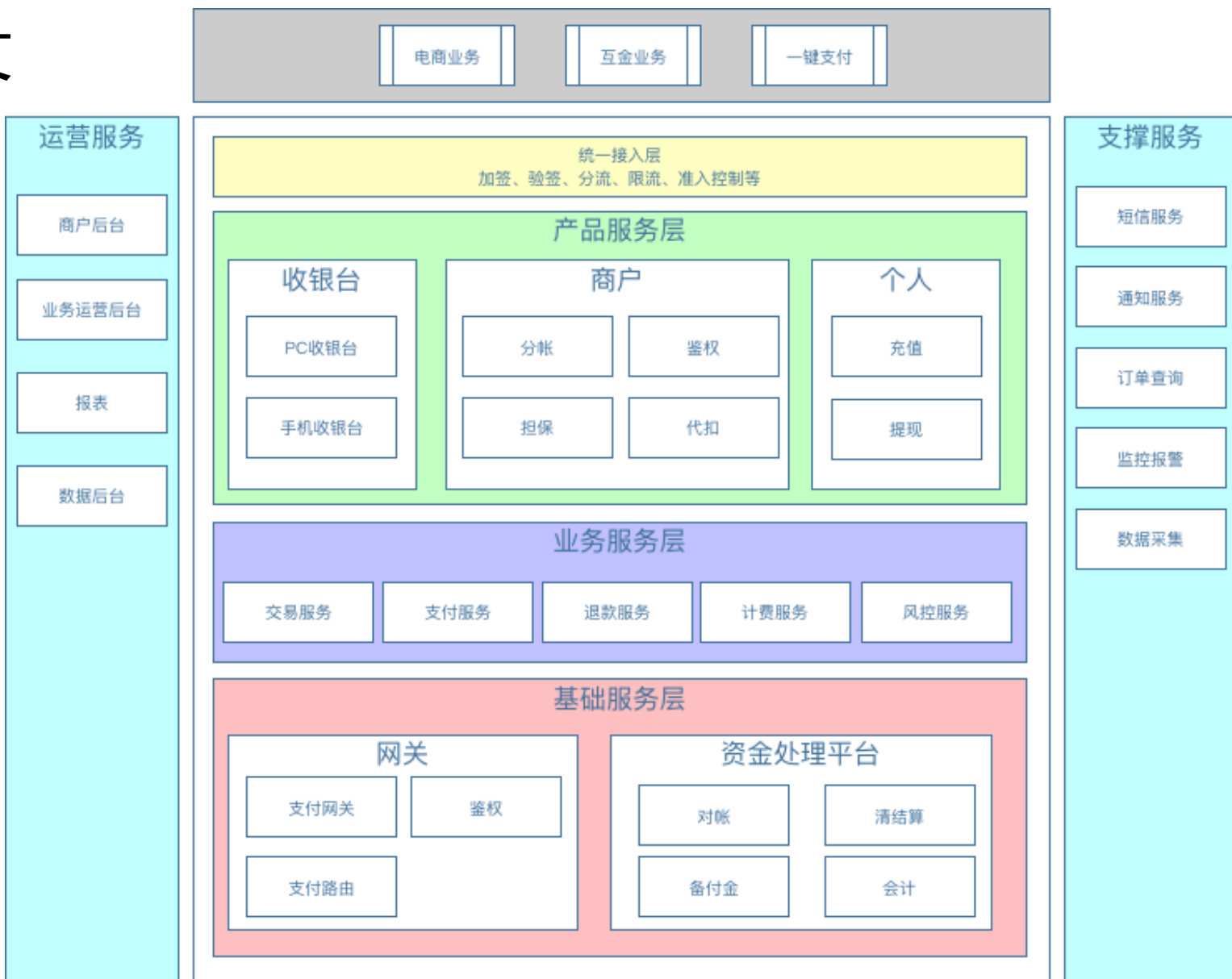
基于微服务架构的改造

---





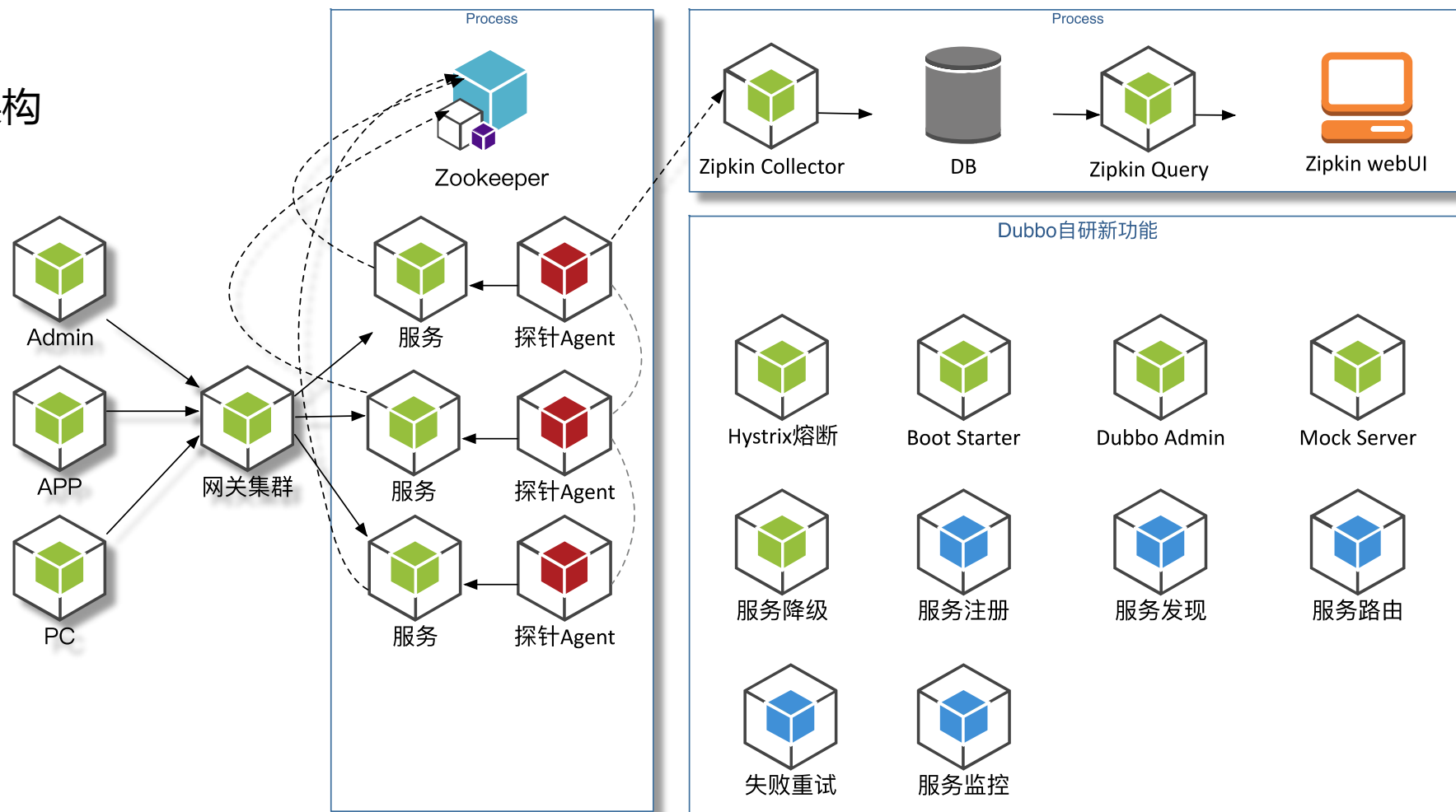
# 利用DDD来划分限界上下文



# 微服务治理架构图

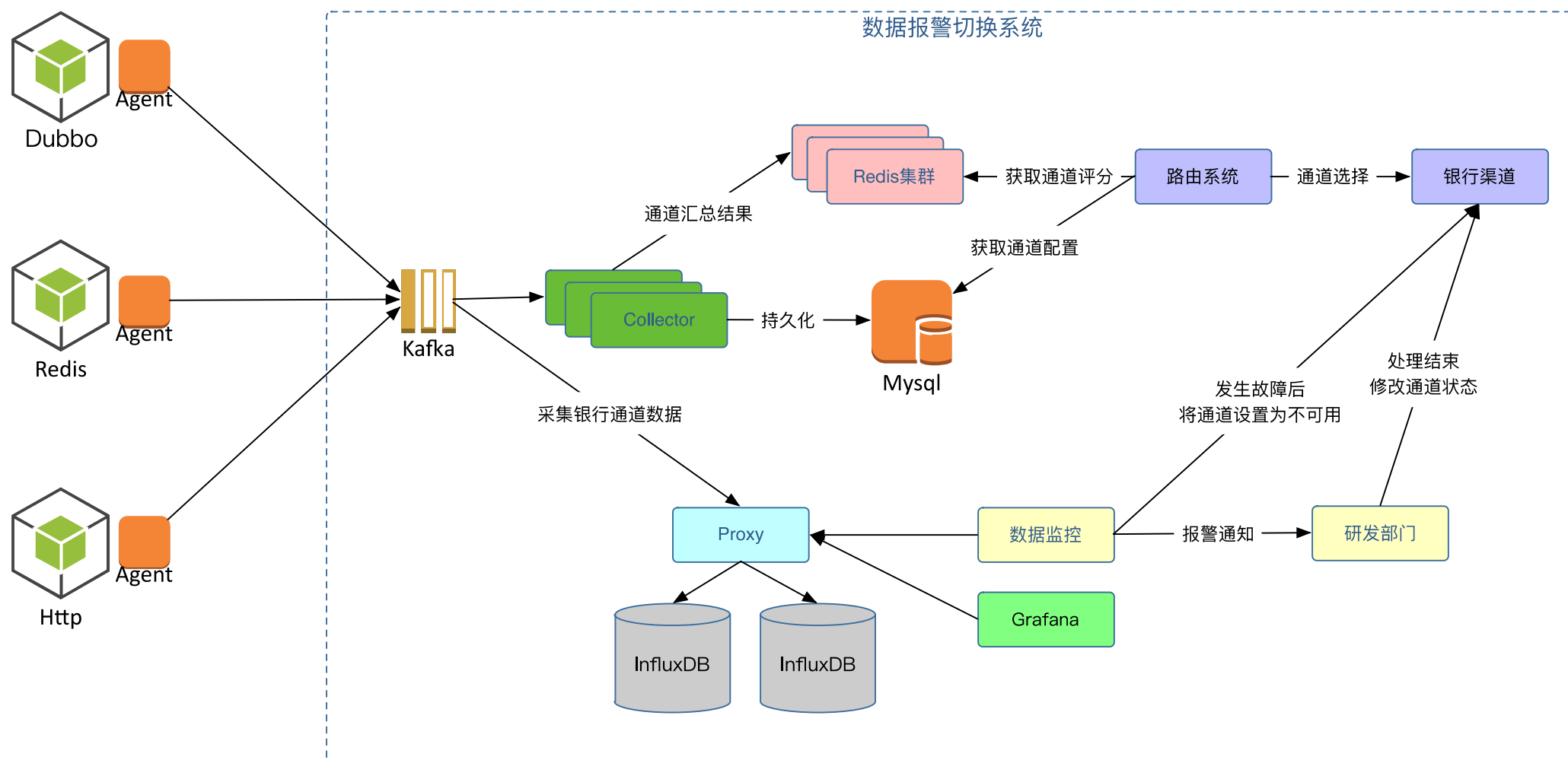
采用Spring boot+Dubbo架构

疑问：  
为何不用Spring Cloud？

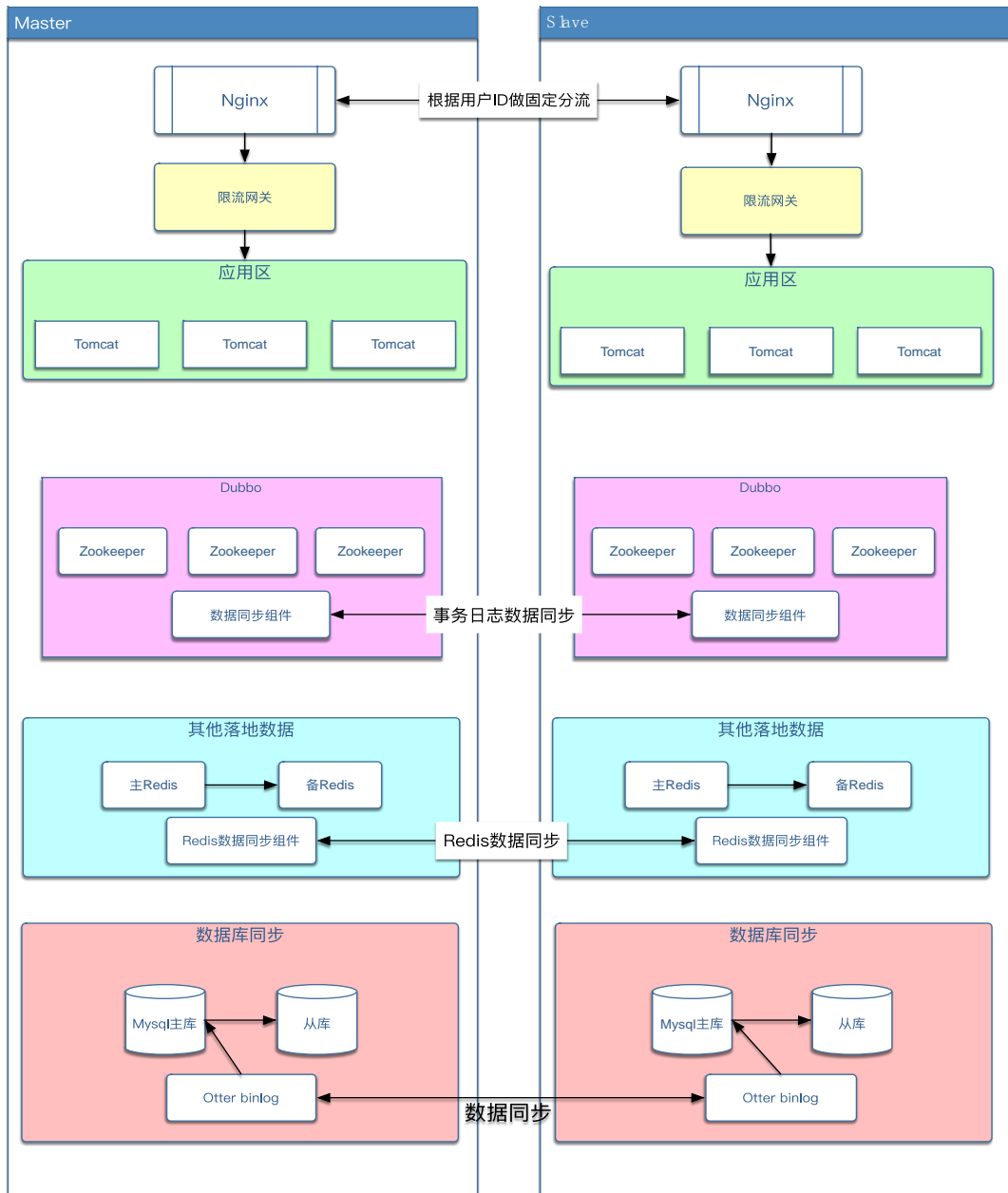


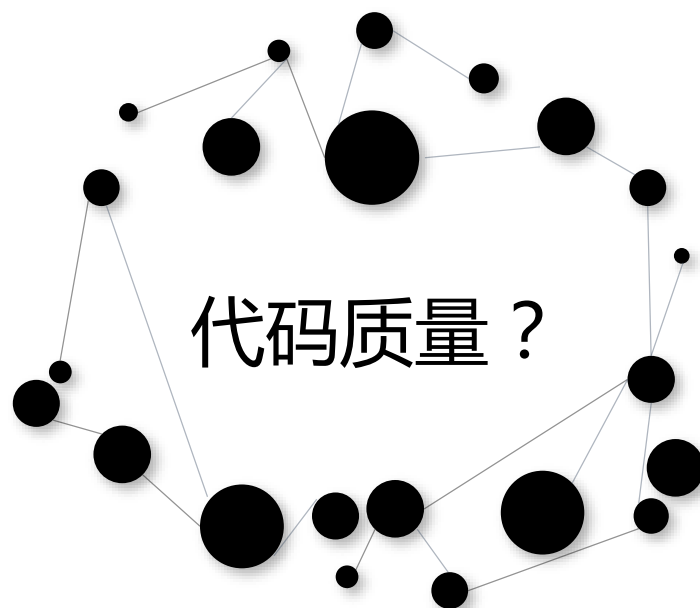


# 通道报警切换系统的演进



# 双活体系架构的演进





&gt;&gt;&gt;&gt;

普通业务四象限

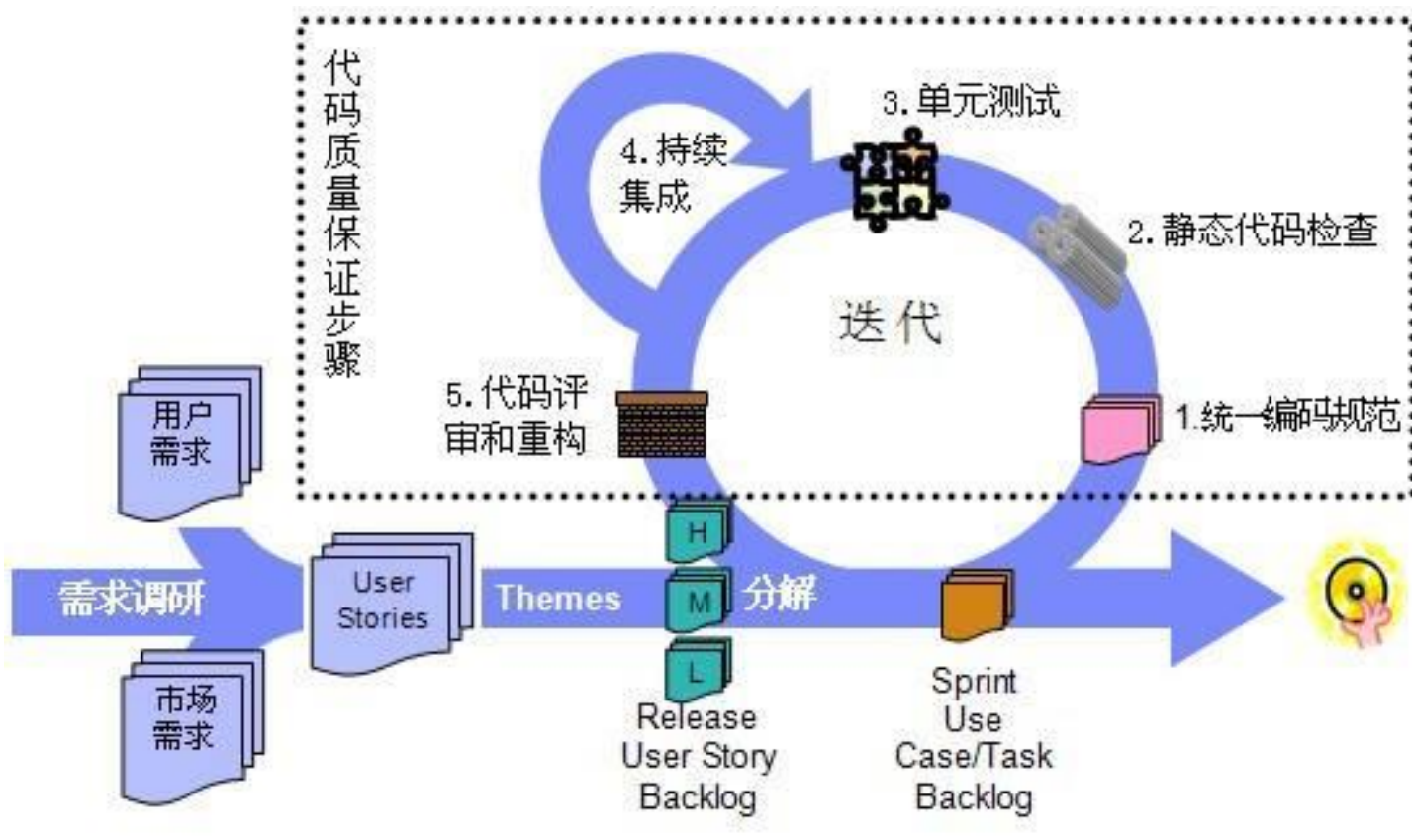
单元测试、验收测试、探索测试、压力测试

持续集成测试

# 持续集成测试

持续集成和代码质量检测

sonarQube  
Gerrit/Gitlab



# 4

Part 04

未来计划要做的事





DBAplus

[www.dbaplus.cn](http://www.dbaplus.cn)

THANK YOU!