

付钱拉

API架构演进

复杂业务背景下微服务化的是与非



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲视频信息



全球软件开发大会 [北京站]

2017年4月16-18日 北京·国家会议中心

咨询热线: 010-64738142



全球架构师峰会 2016 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682

主讲人



闫金明

- **从业务模型说起** 简化的业务介绍
业务痛点、难点分析
- **系统模型演进** 简单的集中式与分布式
过程分析 高性能的阶段性事件驱动 (SEDA)
热议的微服务
- **带刺的玫瑰** 看上去很美
背后的困扰
- **微服务跳坑经**

从业务模型说起



🌐 互联网化

💰 聚合支付

📈 鉴权服务

🏦 银行流水

💰 余额增值

🏠 金融云

📱 线下扫码

💡 资金管理

👤 征信报告

🛒 理财超市

业务功能

网银支付

实时收款

聚合支付

快捷支付

批量收款

线下扫码

实时付款

批量付款

退 款

充 值

提 现

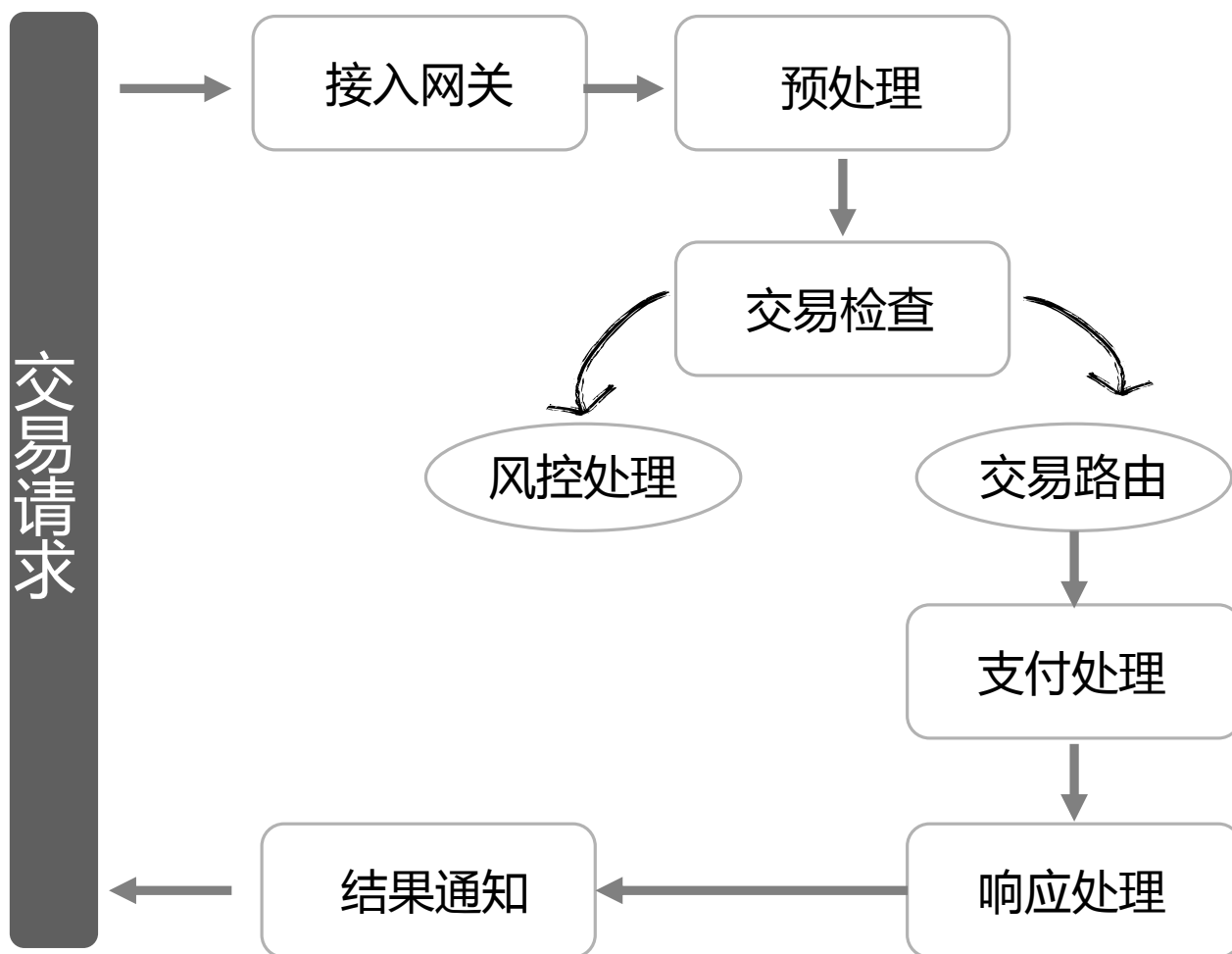
转 账

收银台

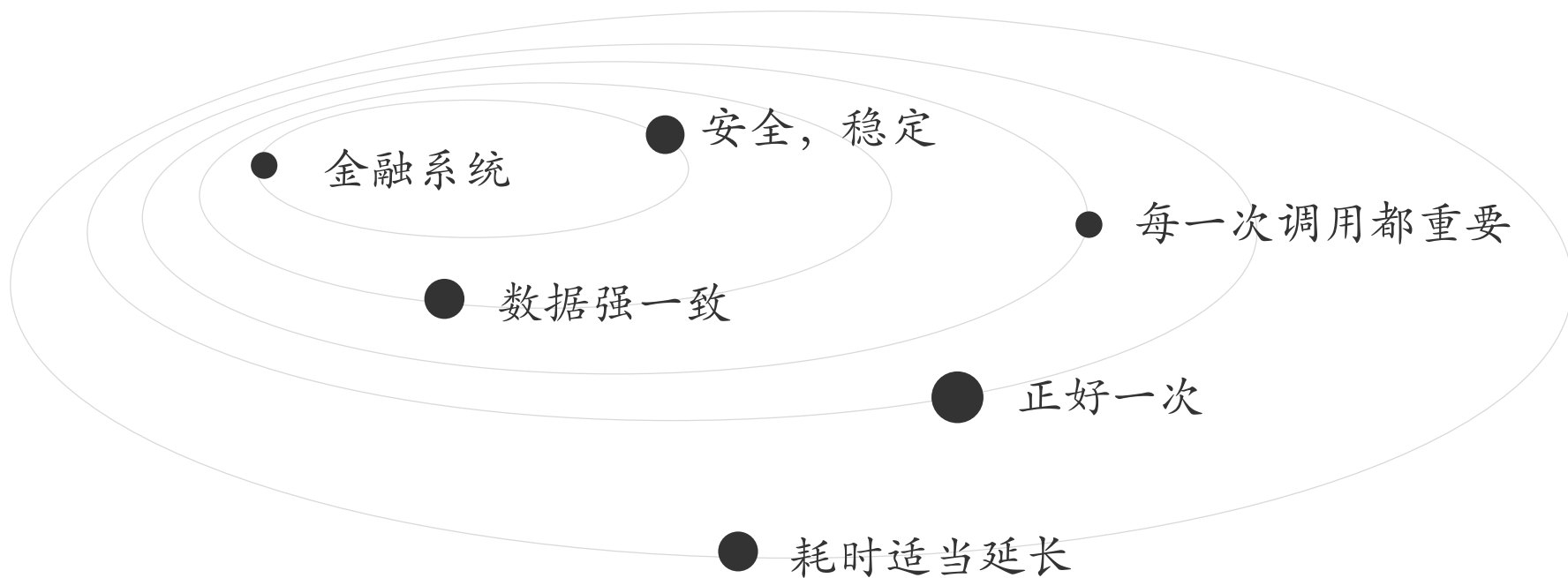
卡bin

鉴 权

系统流程

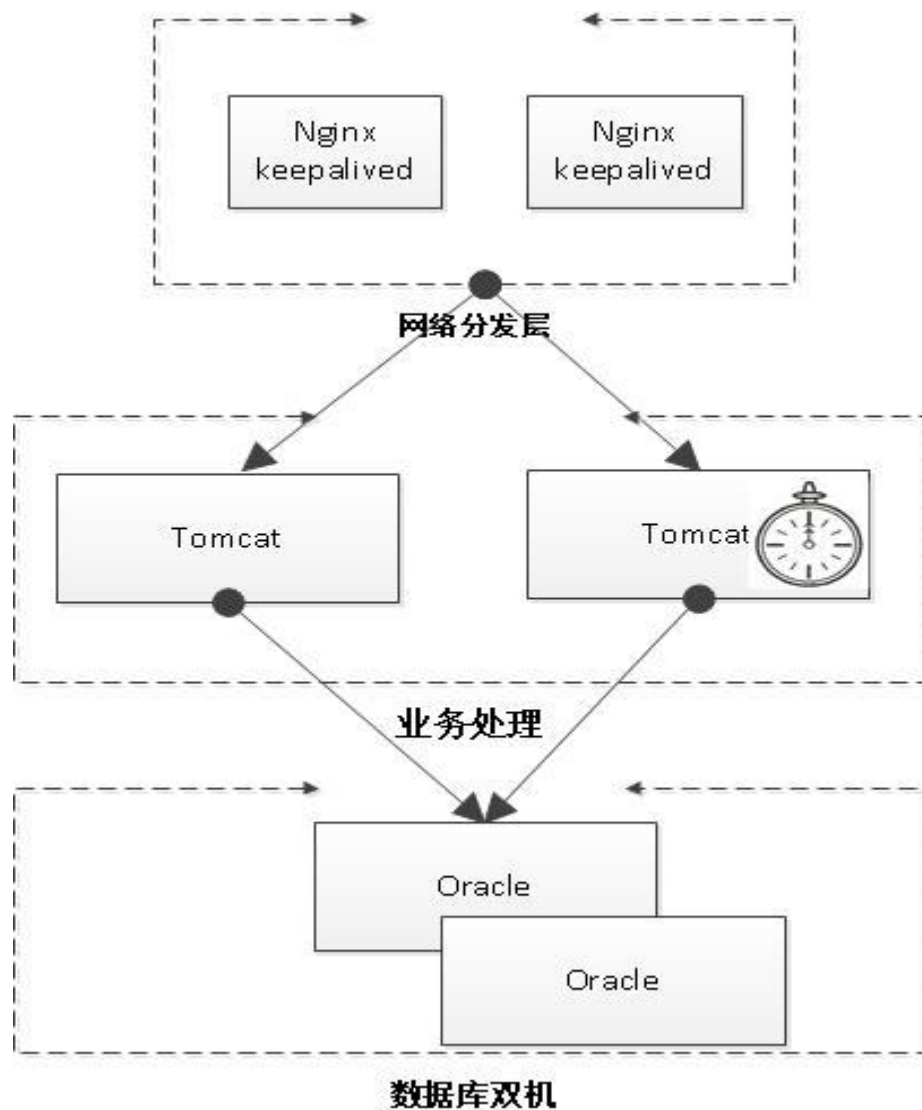


业务特点



系统模型演进

集中式与分布式

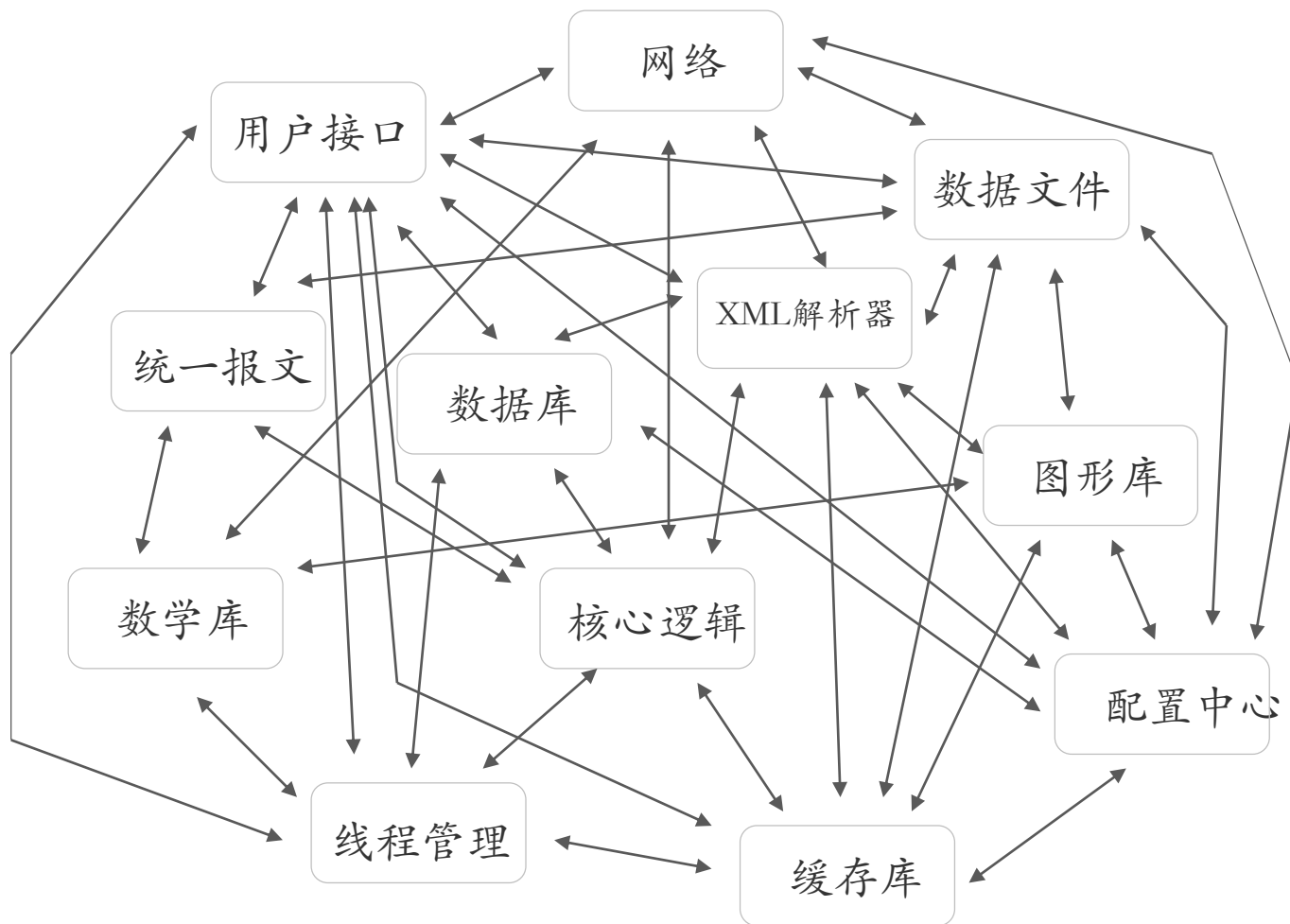


应用分离

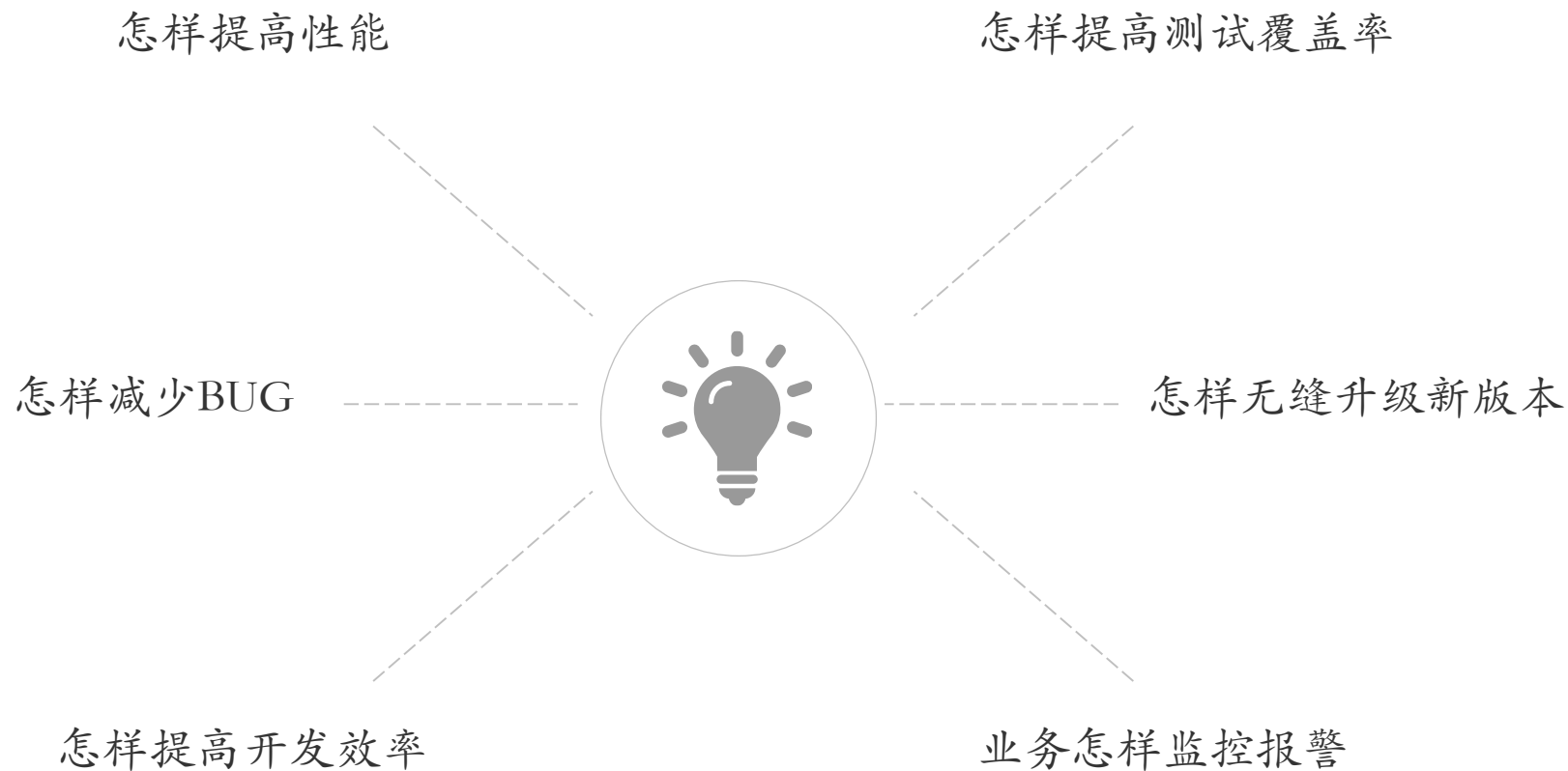
定时分离

池化线程

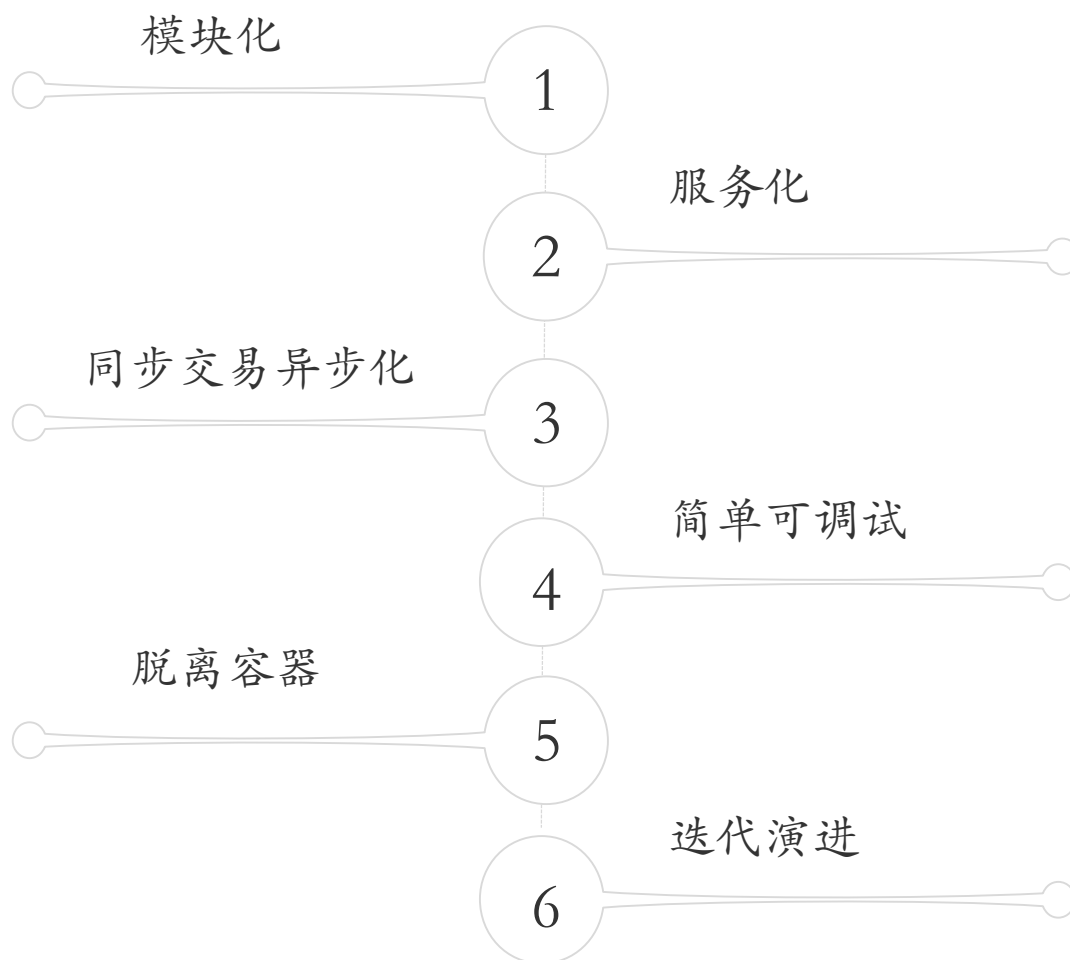
多级缓存



问题在哪里

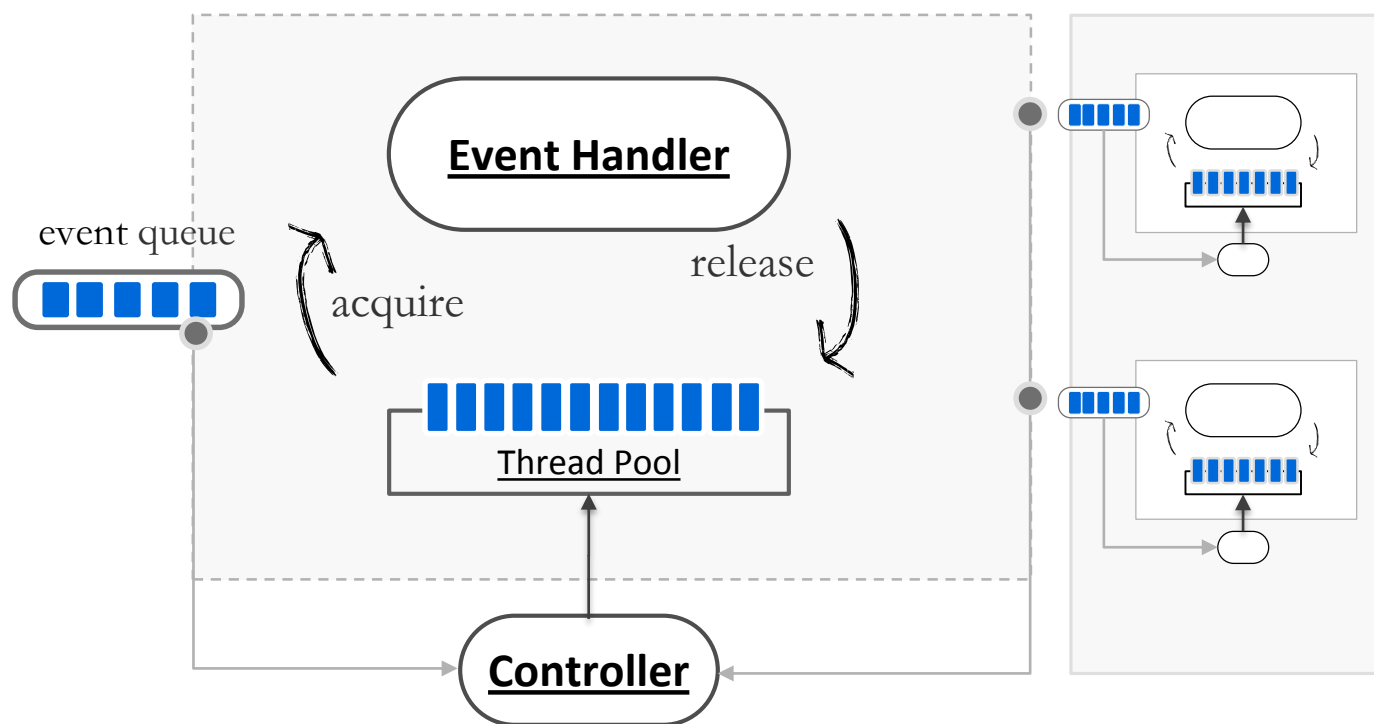


微服务化



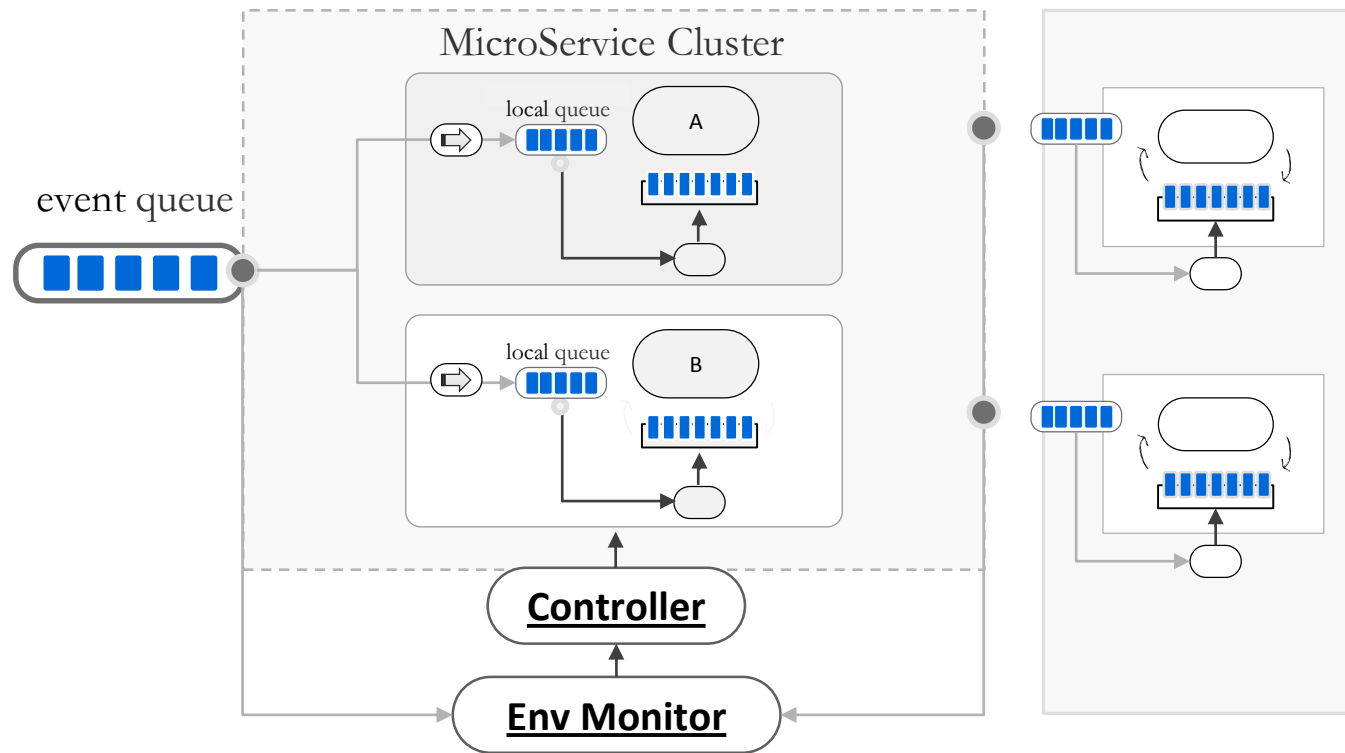
高效的SEDA

SEDA(Staged Event-Driven Architecture)



扩展后的的SEDA

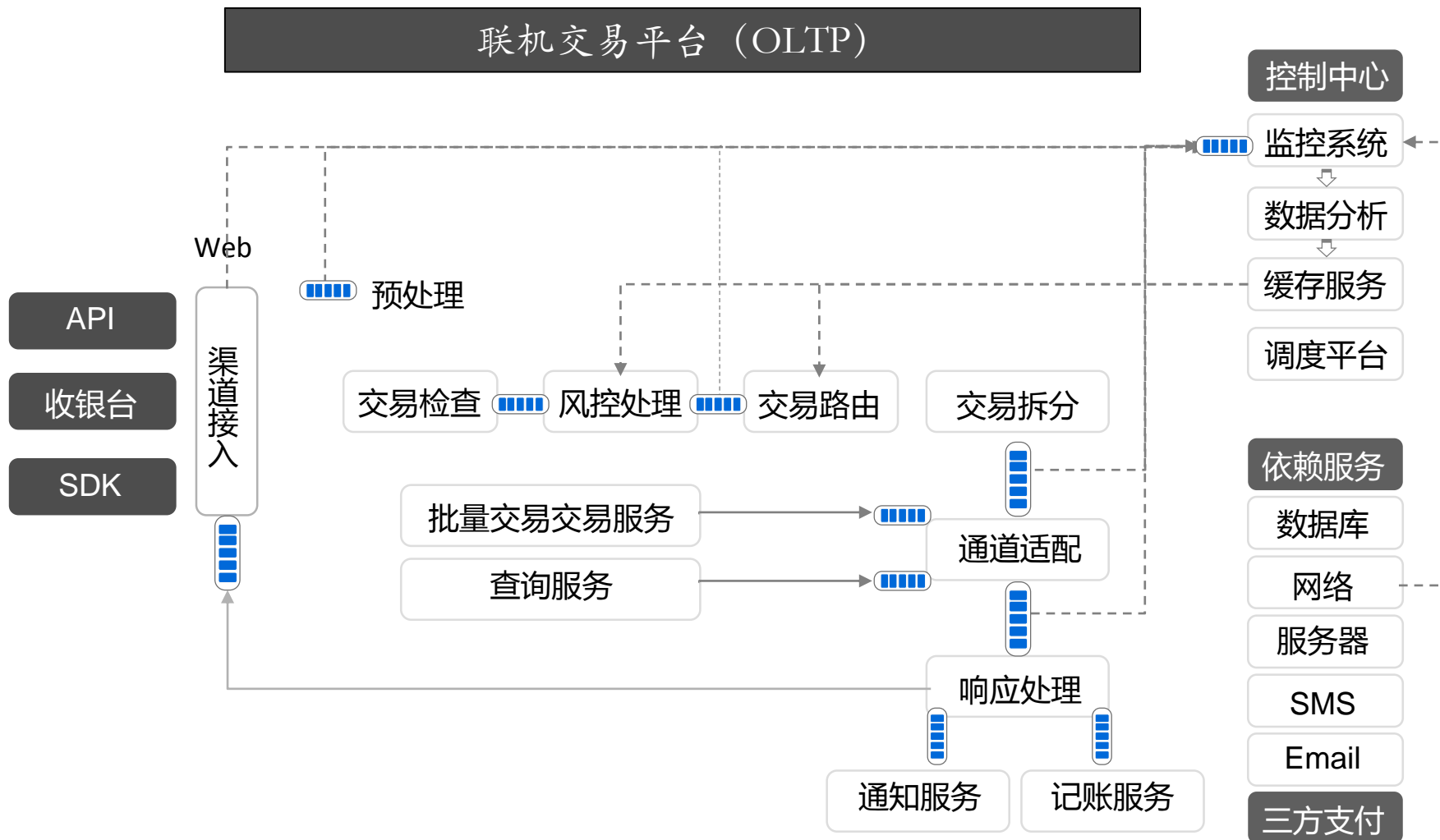
SEDA(Staged Event-Driven Architecture)



功能应用

- 系统解耦
- 同步转异步
- 负载均衡
- 流量缓冲
- 数据总线
- 任务队列

系统架构图



带来的改变

提升性能

快速交付

自动化

弹性伸缩

适用范围

简单易用小中规模

处在技术债的积累期

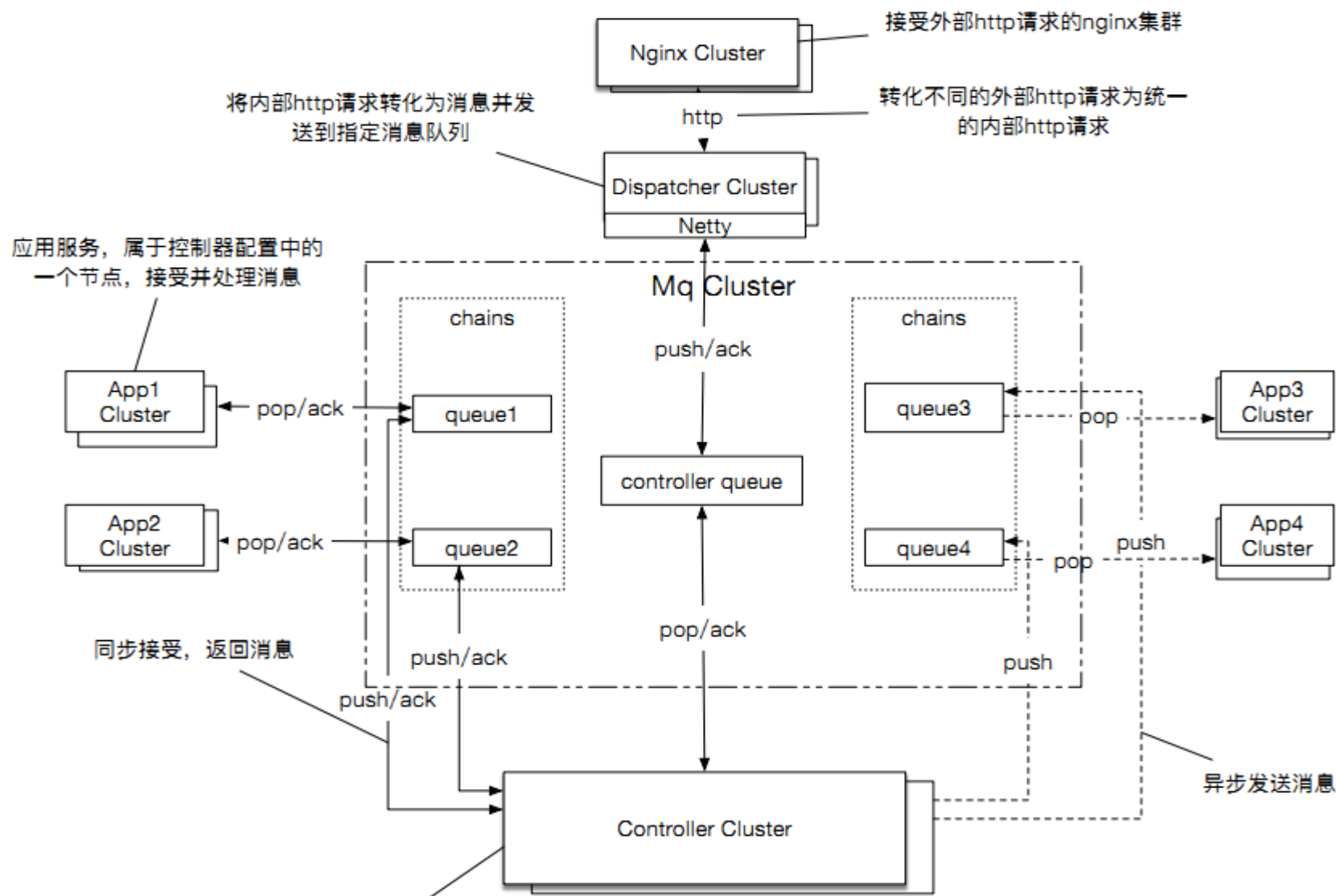
开发团队不过百

硬件规模不过百

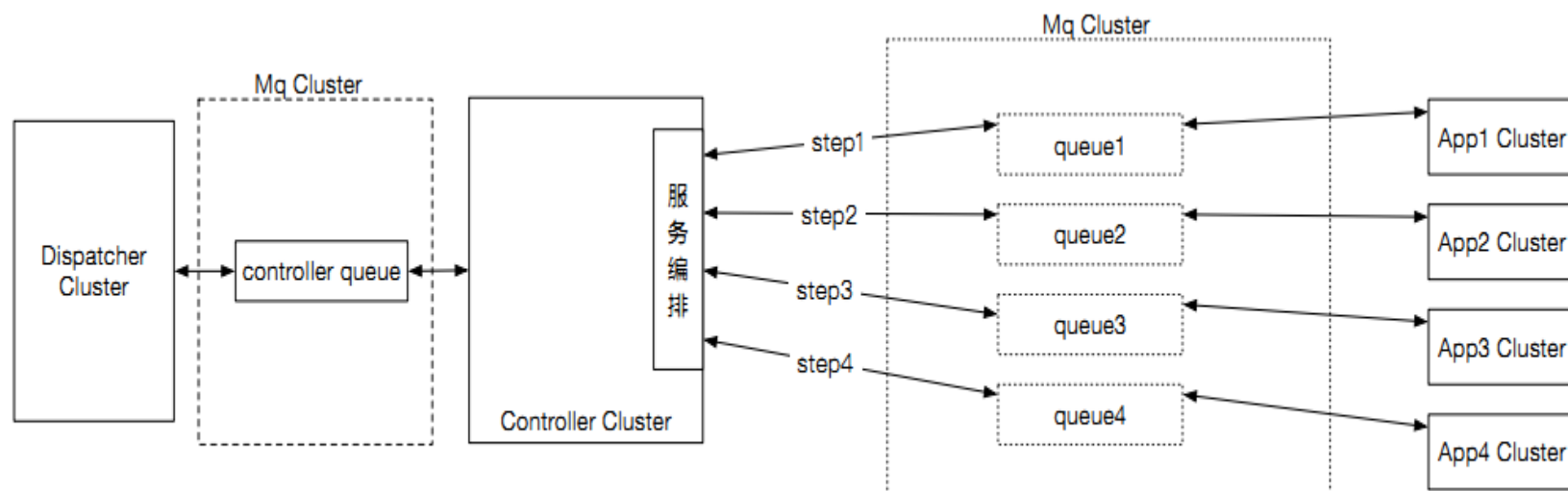
服务调用不过亿

热议的微服务

热议的微服务



服务编排



```
<catalog id="testtrycatch" desc="测试try">
  <chain id="testTryChain">
    <tryBlock desc="测试try">
      <publish queueName="parametersCheck" desc="参数检查" />
      <catchBlock desc="捕获所有异常">
        <publish queueName="excephandle" desc="异常处理" />
      </catchBlock>
      <finallyBlock desc="测试最终处理">
        <publish queueName="finallyhandle" desc="最终处理" />
      </finallyBlock>
    </tryBlock>
    <condition cond="DATASIZE >'0'" class="la.fuqian.controller.chain.command.WhileCondParser" desc="循环拆分数据">
      <publish queueName="fetchdata" desc="分拆数据"/>
      <remotecall queueName="batch_controller" serviceName="batchTest" desc="将消息发送到被另一个controller监控" isSync="false" />
    </condition>
    <publish queueName="fetchdata" desc="分拆数据"/>
  </chain>
</catalog>
```

微服务，带刺的玫瑰

看上去很美

较小的颗粒度

更多的选择性

分散的数据存储

背后的困扰

资源分配难题，如何准确按需分配

数据库连接管理难题

监控复杂度提升，监控要求更高

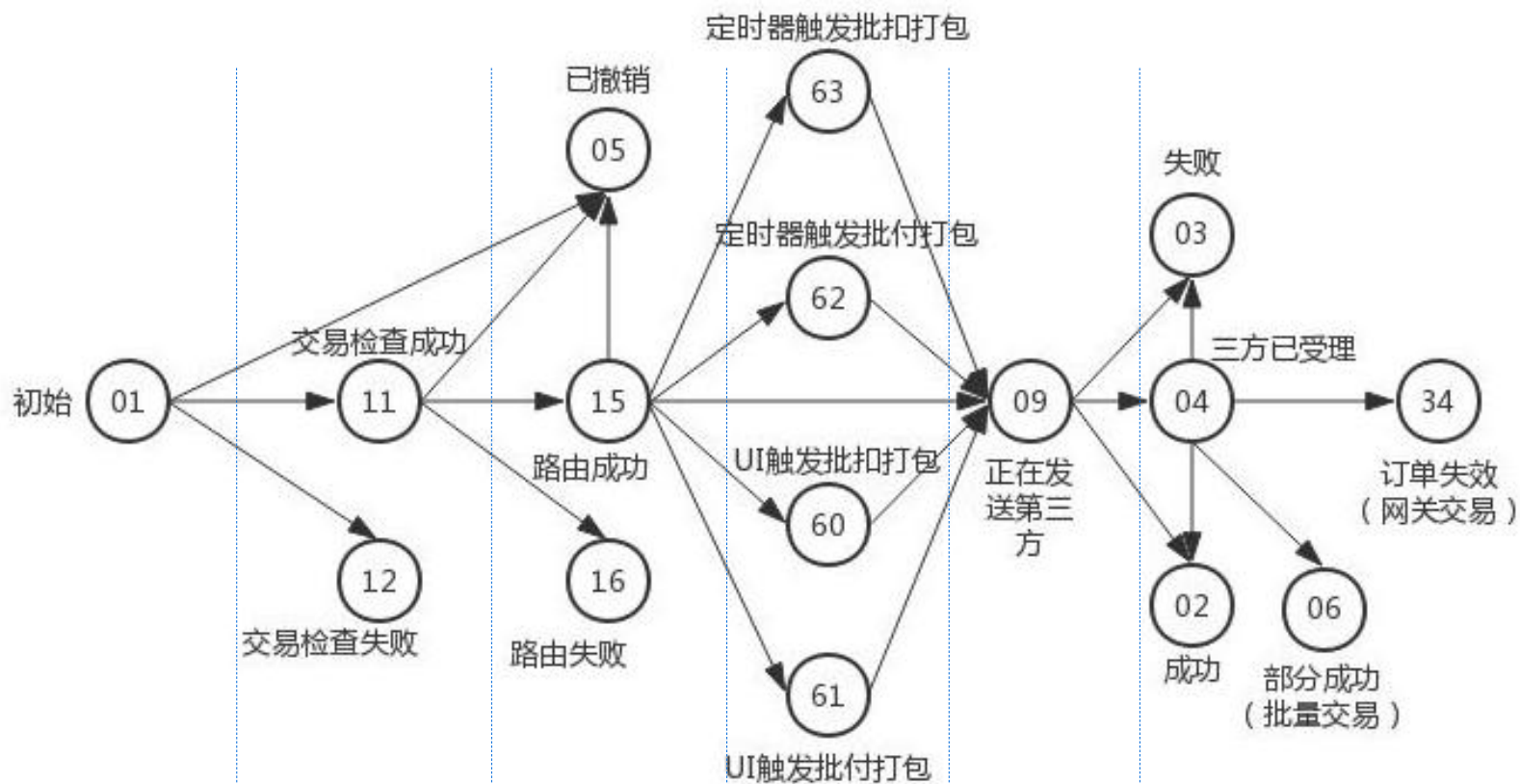
部署流程复杂

自动化程度要求高

微服务，跳坑经

微服务划分

服务划分



接入网关

交易检查

服务路由

批量任务

通道适配

响应处理

监控，被忽视的价值

监控的目的是什么？

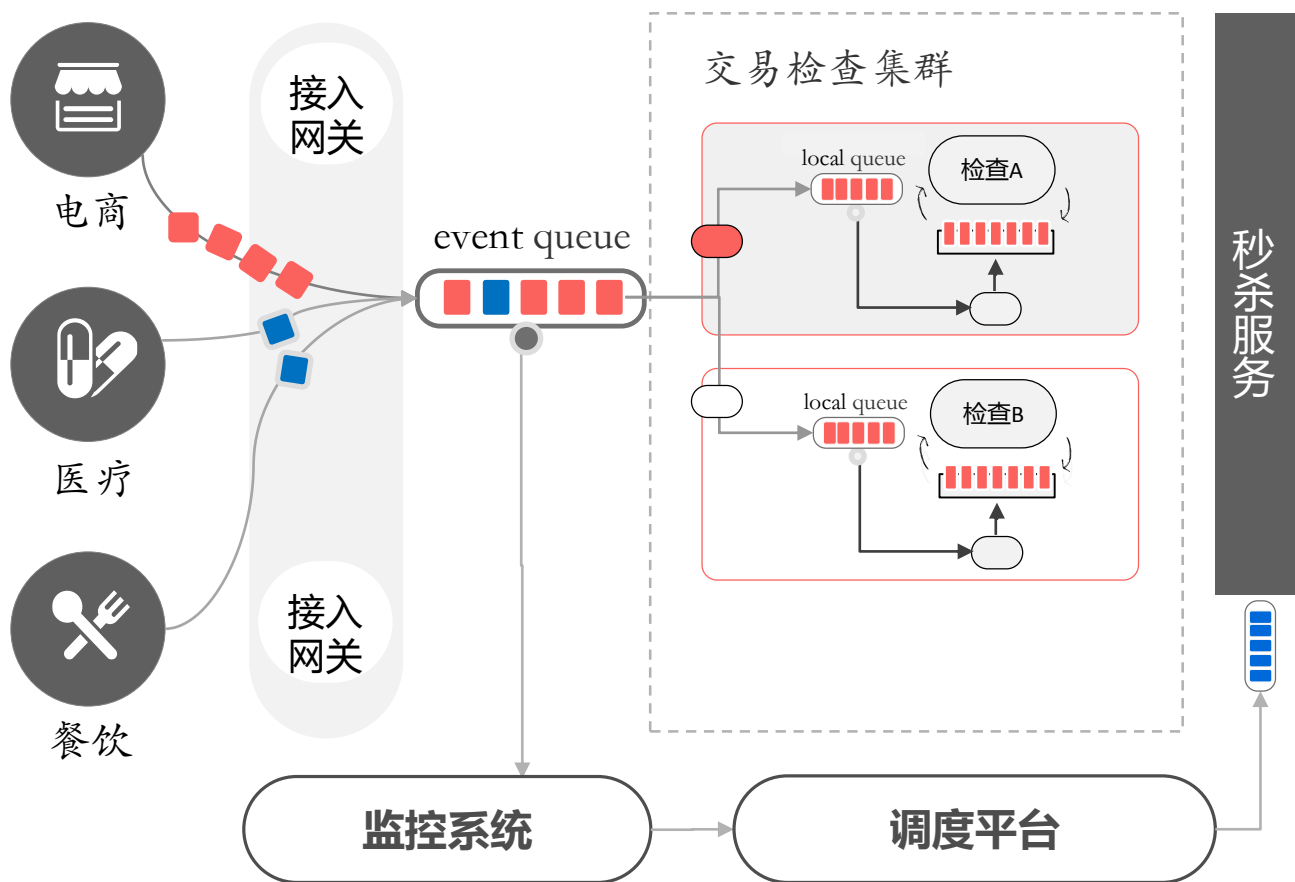
- 关注服务状况
- 故障提前预知
- 系统瓶颈诊断
- 服务性能优化
- 排查安全隐患

雪崩效应 | 服务降级 | 熔断

系统依赖与风险



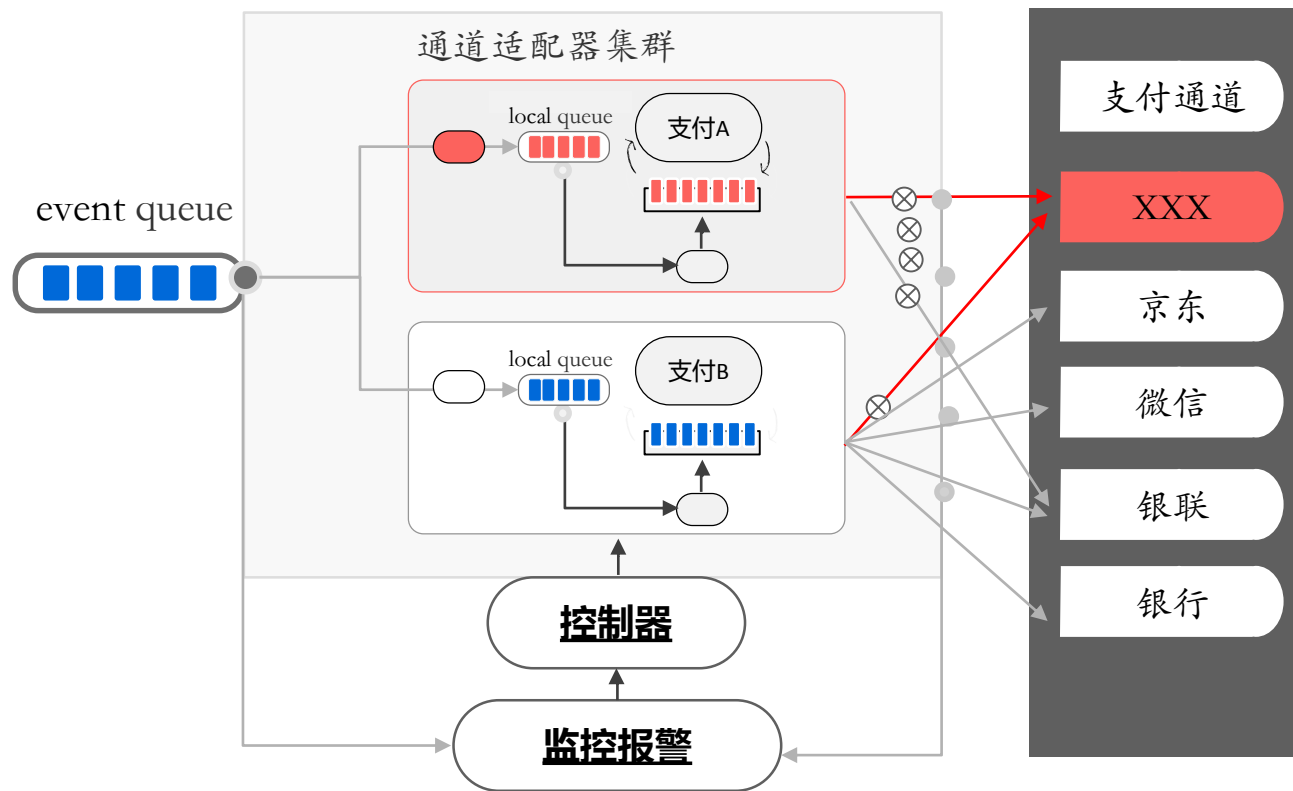
雪崩效应-流量激增、资源抢占-服务降级



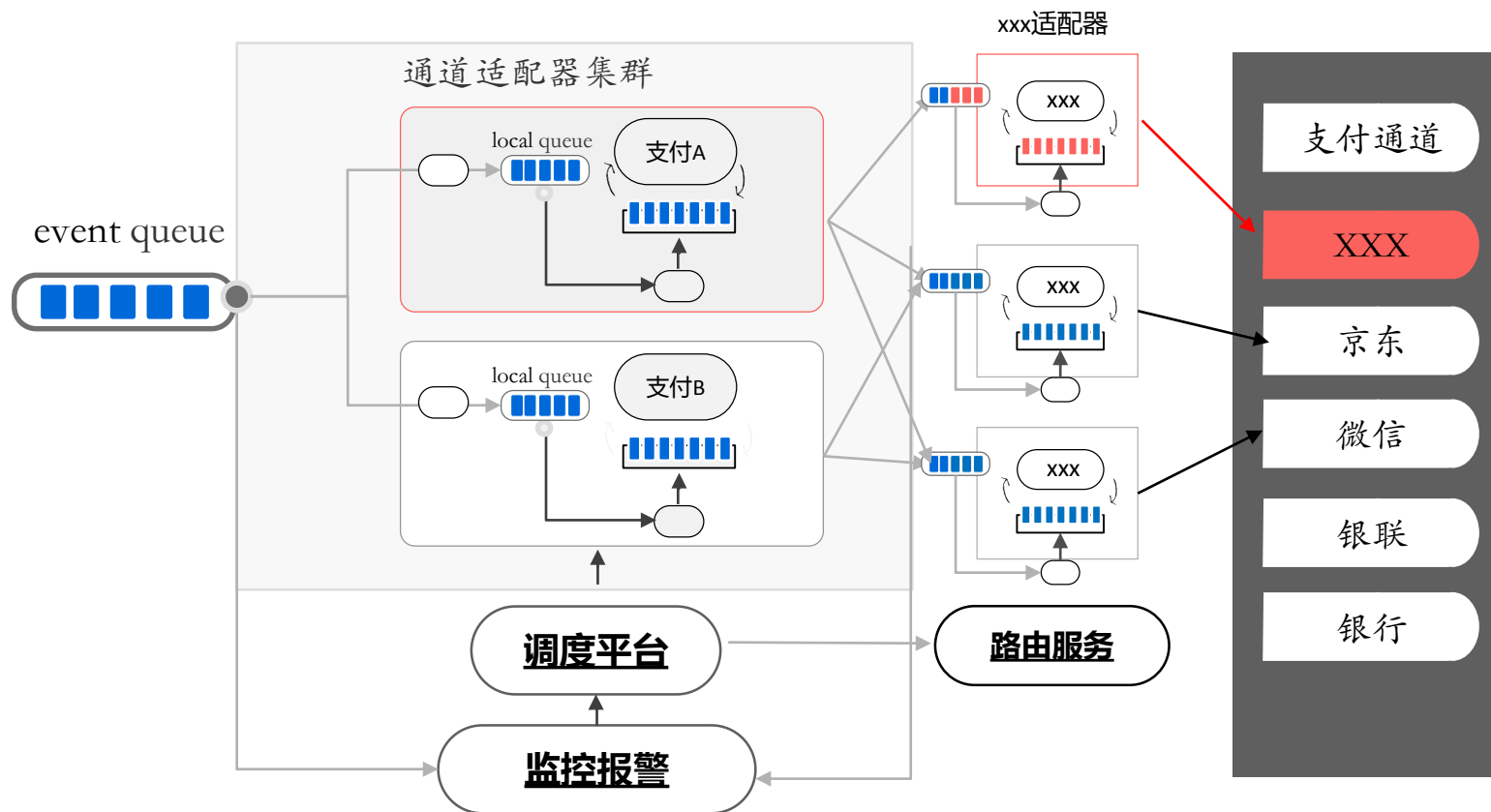
服务降级-秒杀服务系统

- 流量削峰
- 资源调节

雪崩效应-外部依赖失效-熔断



雪崩效应-外部依赖失效-熔断



资源分配管理难题

传播架构方案

合适的业务发展阶段
选择适合的架构模型

谢谢！

微信群



科技让金融更简单