

ibaba</groupId>
o</artifactId>
o.version)</version>

public class DemoService
public String sayHello(S

暨第四届阿里中间件性能挑战赛技术行

Dubbo开发者沙龙

2018.05.12 (周六) 北京·车库咖啡

Dubbo和微店的服务化实践历程分享

演讲者：王欣



- ▶ **微店是谁**
- ▶ **微店的各技术阶段**
- ▶ **挑战&解决方案**
- ▶ **畅想未来**

微店是谁？

微店是帮助卖家在**手机开店**的软件，任何人通过手机号码即可开通自己的店铺

规模
7000万

安全
6亿/天

成本
数千万/年

效率
600+

- 7000万店铺，10亿商品;9P数据
- 支付、搜索、推荐、风控、IM、交易、开放平台、广告、供应链...
- 经受560万次攻击/天；CC、SQLInjection、XSS、CSRF ...
- 每天6亿+爬虫虫访问...
- IDC方方面面，2016/2017年年整体支出数千万
- 600+ 研发一起协同
- 每天有60次上线，400+次发布(含测试/预发)

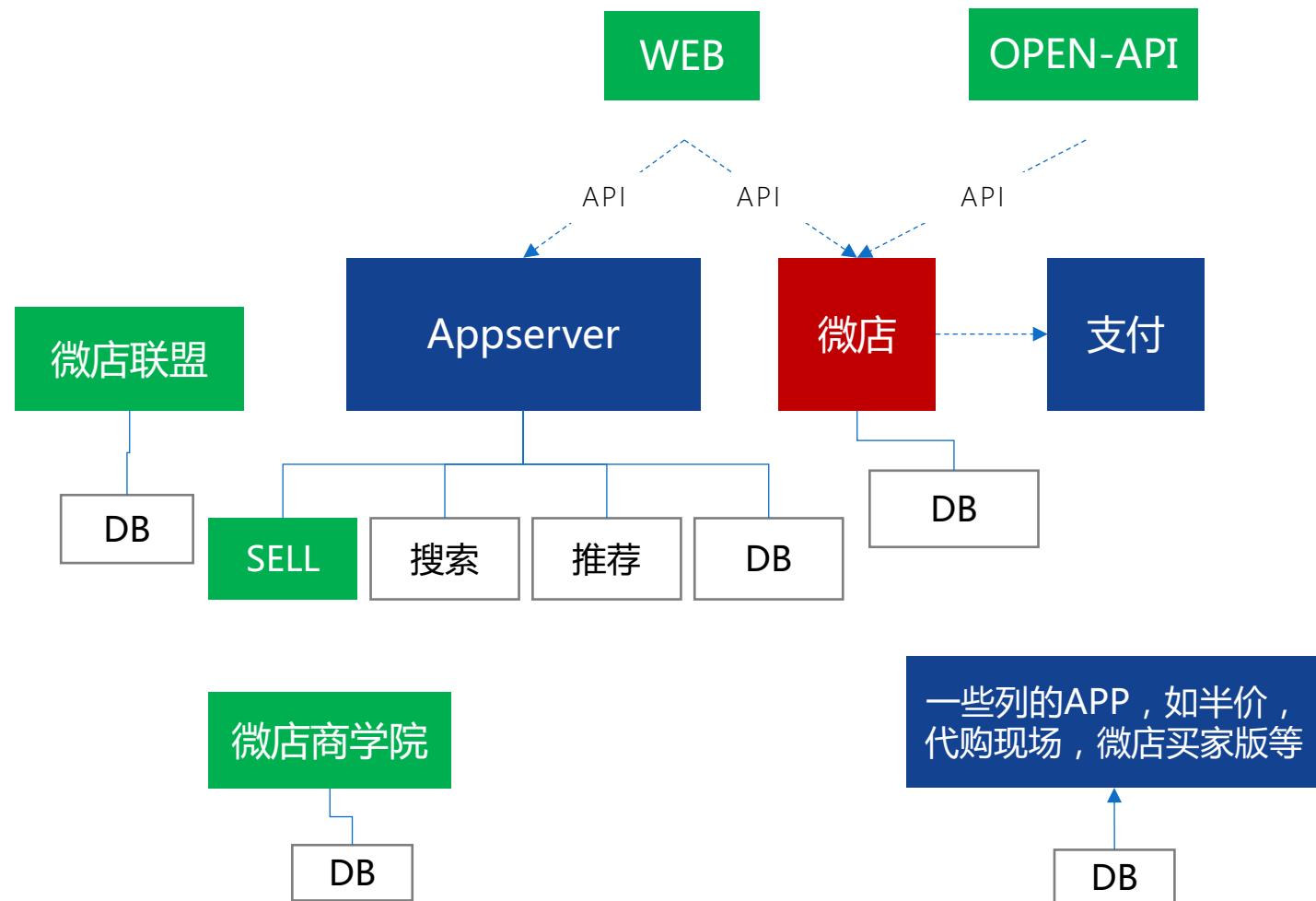
Dubbo和微店的服务化实践历程分享

演讲者：王欣



- ▶ **微店是谁**
- ▶ **微店的各技术阶段**
- ▶ **挑战&解决方案**
- ▶ **畅想未来**

单体应用阶段



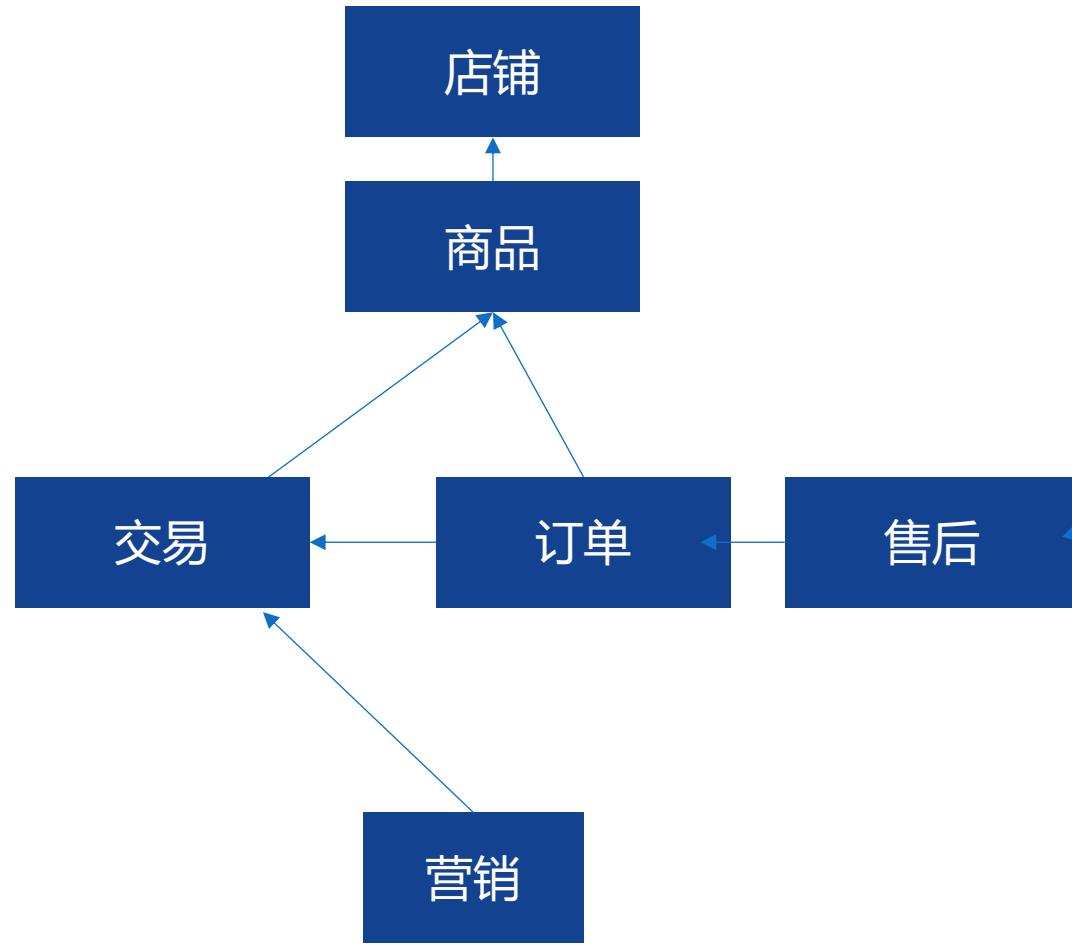
技术栈

- PHP
- Nginx
- Mysql
- Redis

遇到的问题

- 测试困难：需要所有功能回归
- 可伸缩性差
- 边缘BUG，例如死循环、OOM等，拖垮整个站点

多模块阶段



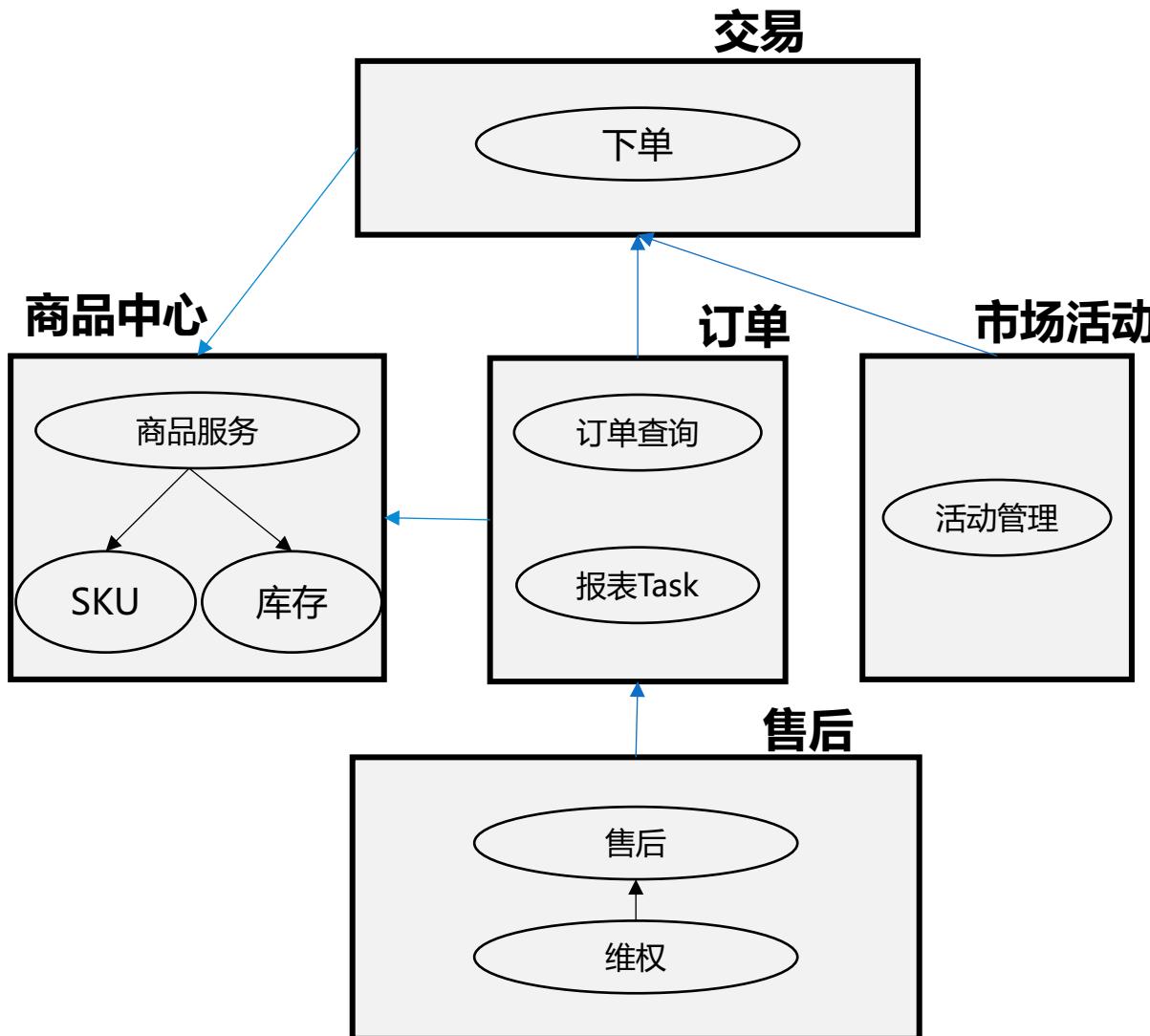
技术栈

- PHP
- Nginx
- Mysql
- Redis

遇到的问题

- 代码拆分，独立部署，流程隔离，
技术栈没有太大变化
- 问题：
 - 升级困难，要全网推动
 - 数据库连接池压力大

服务化阶段



- 服务组出现，订单中心，商品中心，交易中心等
- 技术栈：
 - Dubbo
 - VTrace
 - Vdds
- 存在的问题：
 - 问题追踪困难
 - 上线配置麻烦，运维工作量大
 - 测试麻烦、功能互相干扰

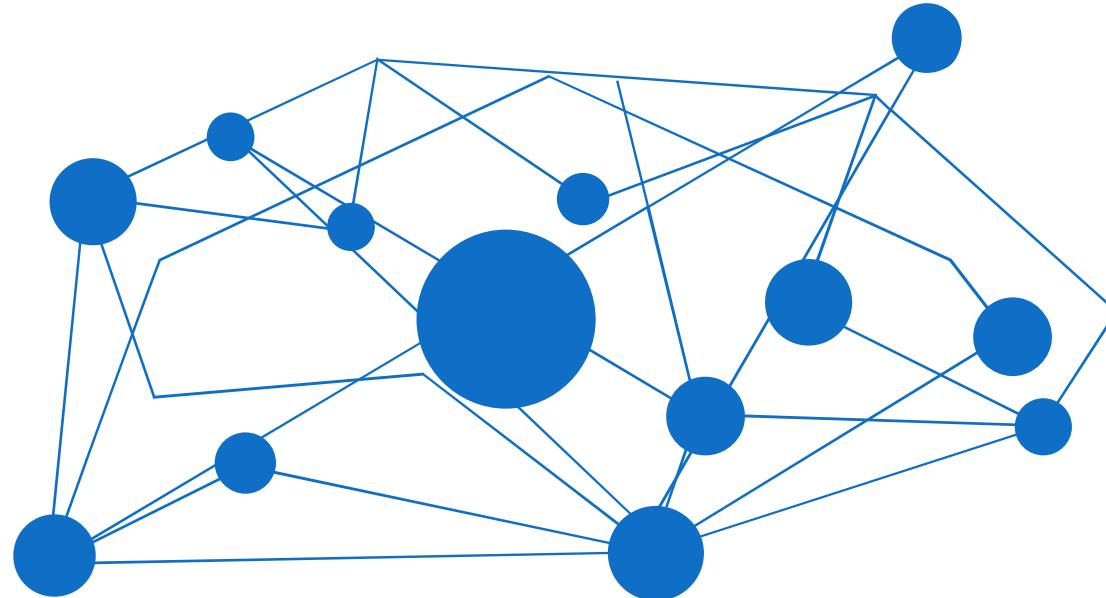
Dubbo和微店的服务化实践历程分享

演讲者：王欣



- ▶ **微店是谁**
- ▶ **微店的各技术阶段**
- ▶ **挑战&解决方案**
- ▶ **畅想未来**

服务化-模块拆分



复杂的业务关系



如何拆分模块？

- 先考虑业务 VS 先考虑数据表？
- 代码行数该不该作为分包的依据？
- 设计、工程、运营等的追求如何均衡？

服务化-模块拆分



Microservices - the new architectural style

Martin Fowler, Mar 2014

◆ 微服务：

- 分布式服务组成的系统
- 按照业务而不是技术来划分组织
- 自动化运维
- 服务快速演化迭代

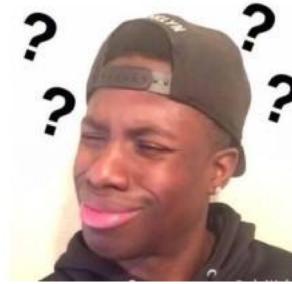
◆ DDD Domain-driven Design 领域驱动设计

- 通过建立领域模型来解决领域中的核心问题
- 通过领域模型驱动代码的实现
- 技术架构设计或数据存储等是在领域模型的外围
- DDD vs SSI (spring+hibernate) :充血模型/失血模型



拆分原则：

- DDD业务驱动
- 设计、工程、运营等的平衡
- 渐进演化

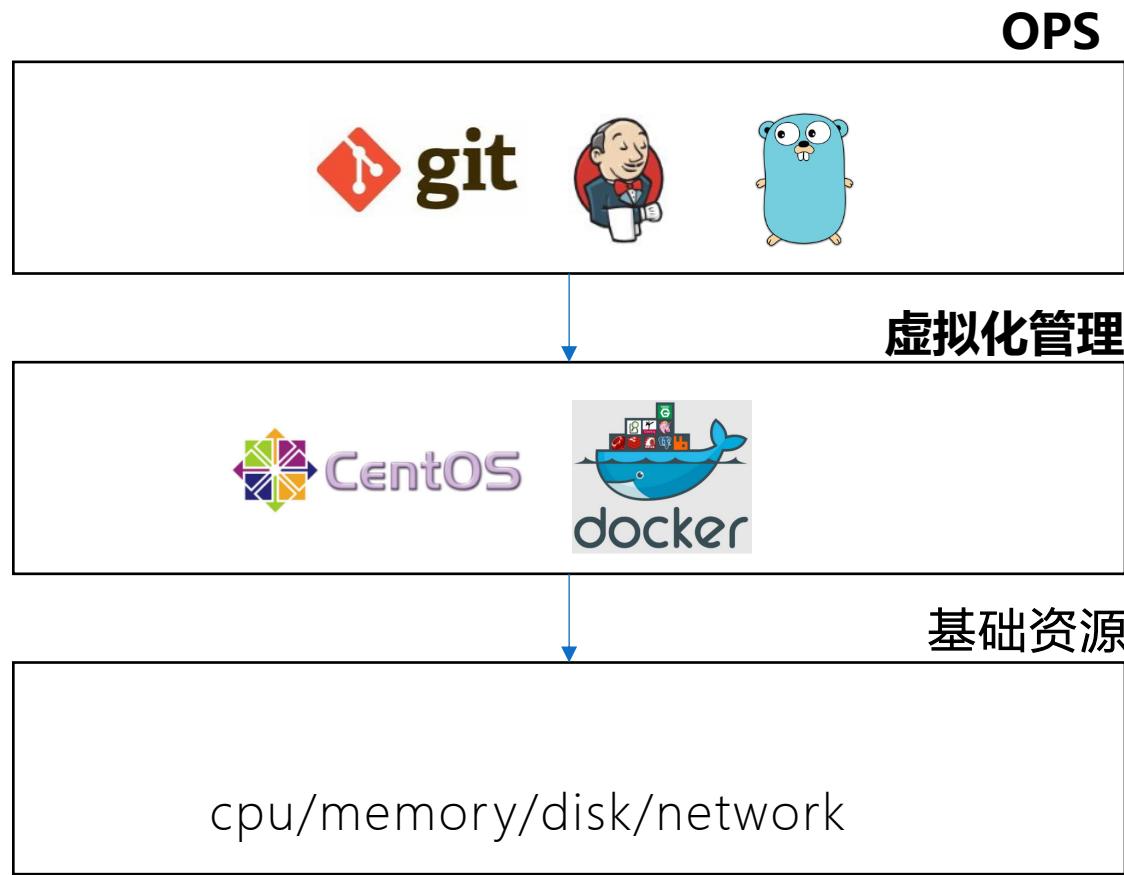


服务化-自动化部署

运维部署 的问题

- 服务变多，配置工作量变大，如何保证不出差错？
- 如何保证测试系统按照业务需求feature独立？
- 如何实施CI/CD的流水线系统？
- 怎么充分压榨基础设施 (CPU/memory/disk/network) 的使用率？

服务化-自动化部署



- OS版本

- Centos7.2

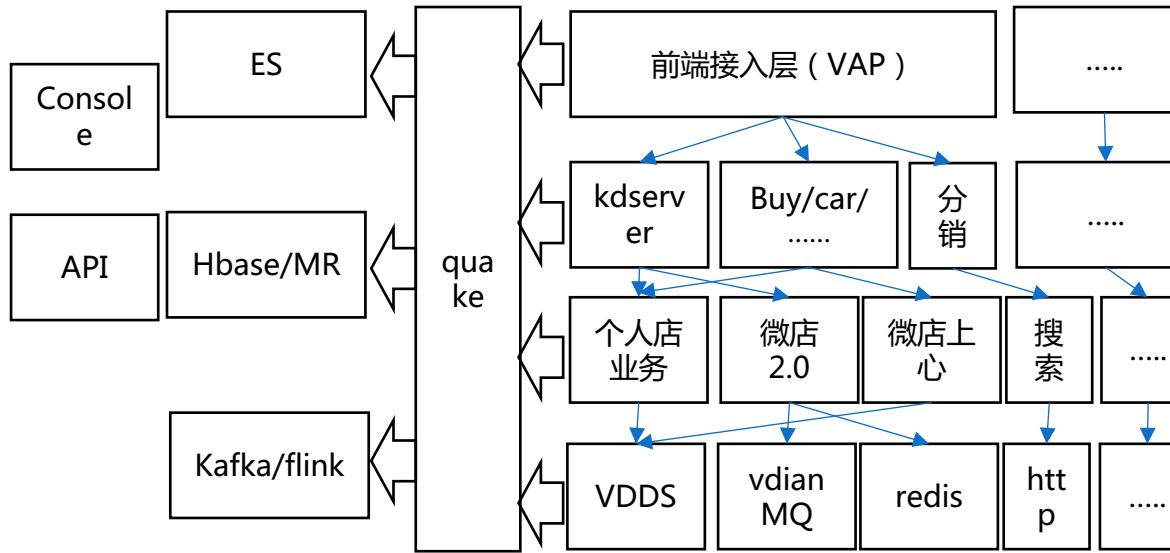
- 虚拟化技术

- KVM
- Docker

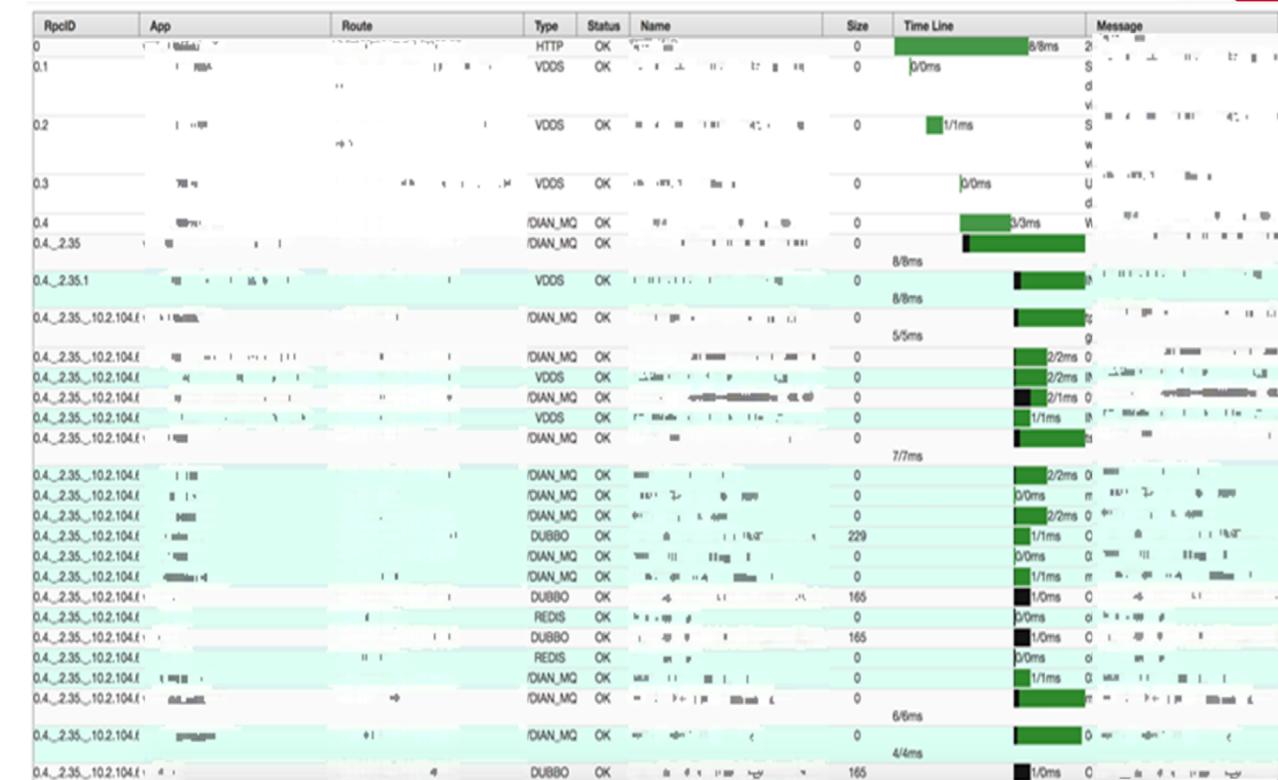
- 网络模式

- 网桥
- ✓ Pipework
- ✓ Libvirt
- 配置唯一IP，且全网互通

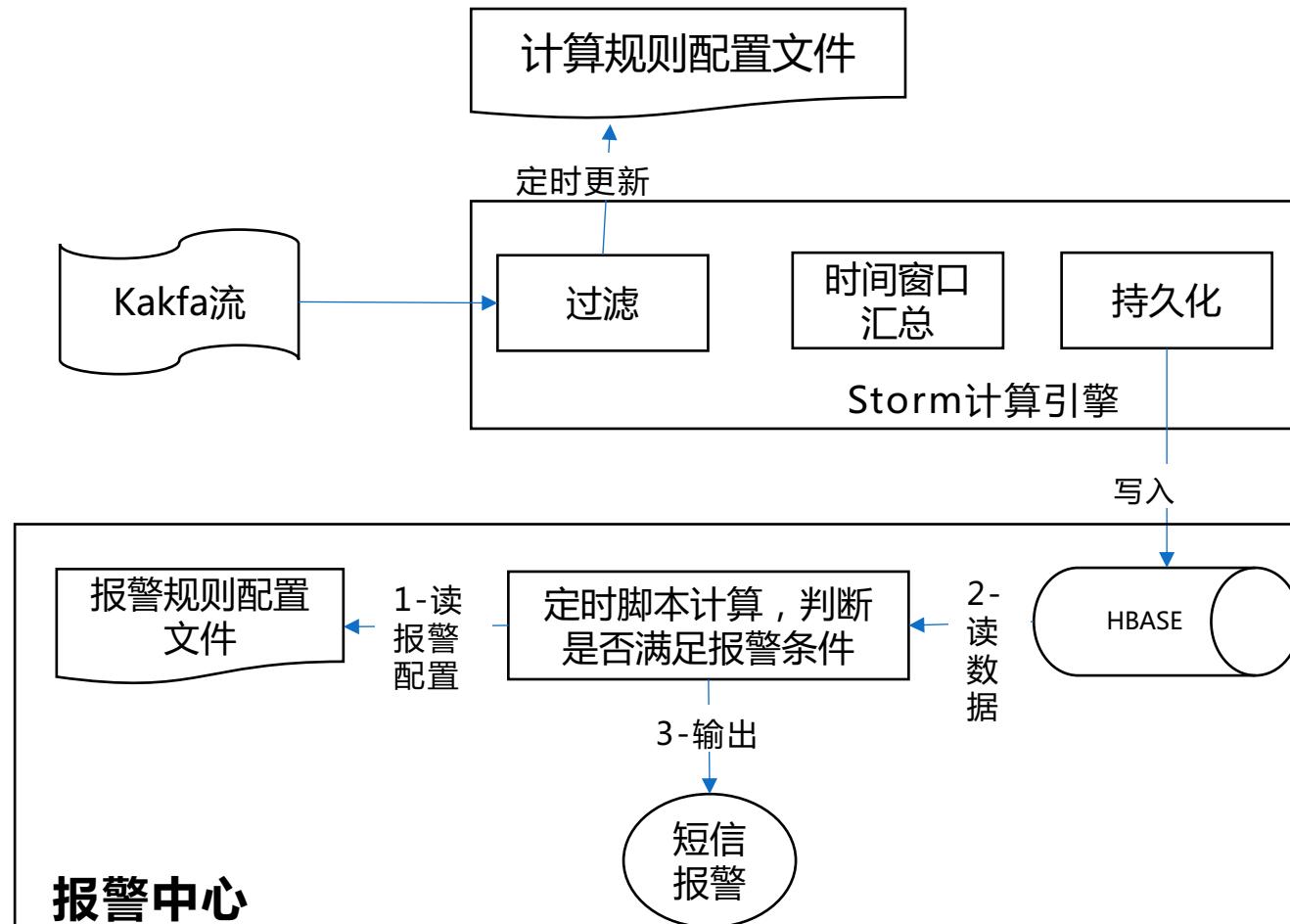
服务化-调用链路追踪



- Dapper : rpc跟踪论文基础
- rpc框架生成、无业务侵入
- trace Id : 初始调用生成uuid串
- 每个rpc调用生成一个span , 同时记录父span
- 其他中间件全面支持 : tomcat/dubbo/vdianMQ/vdds/redisClient



服务化-应用服务监控、报警



● 应用监控和报警

- Vmonitor agent 收集rpc跟踪日志
- 报警规则配置（流量异常/服务异常超阈值 ...）

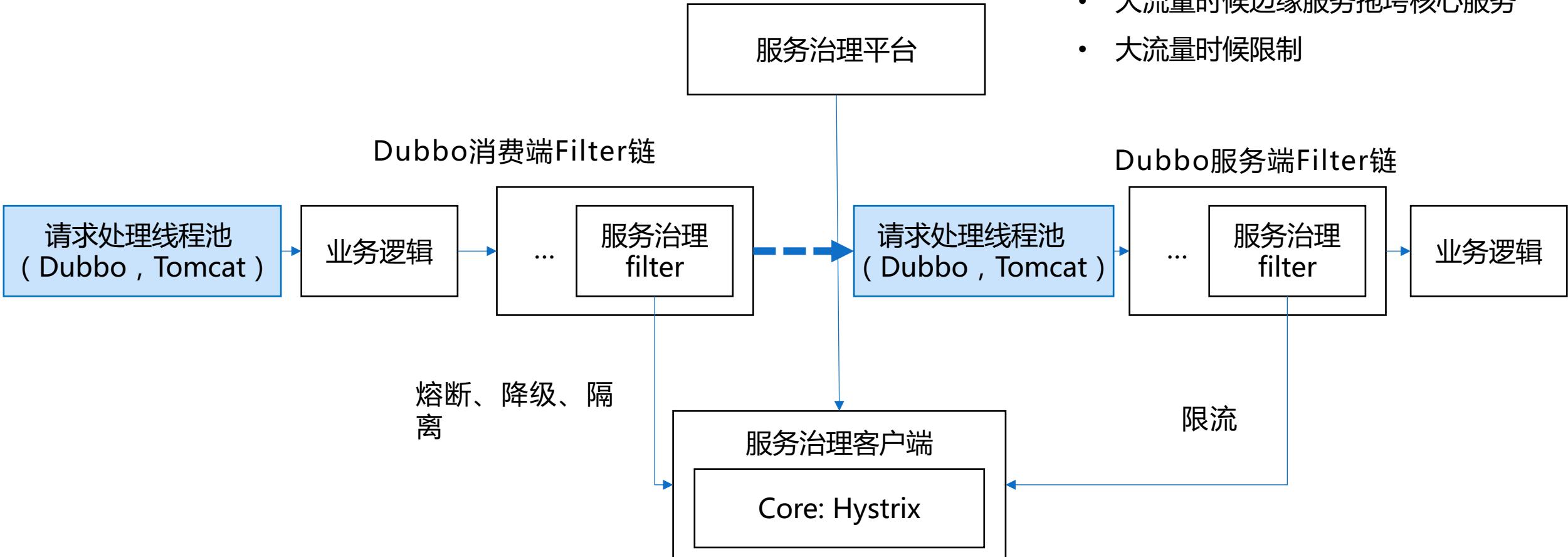
● 解决的问题：

- 异常流量预警
- 错误预警

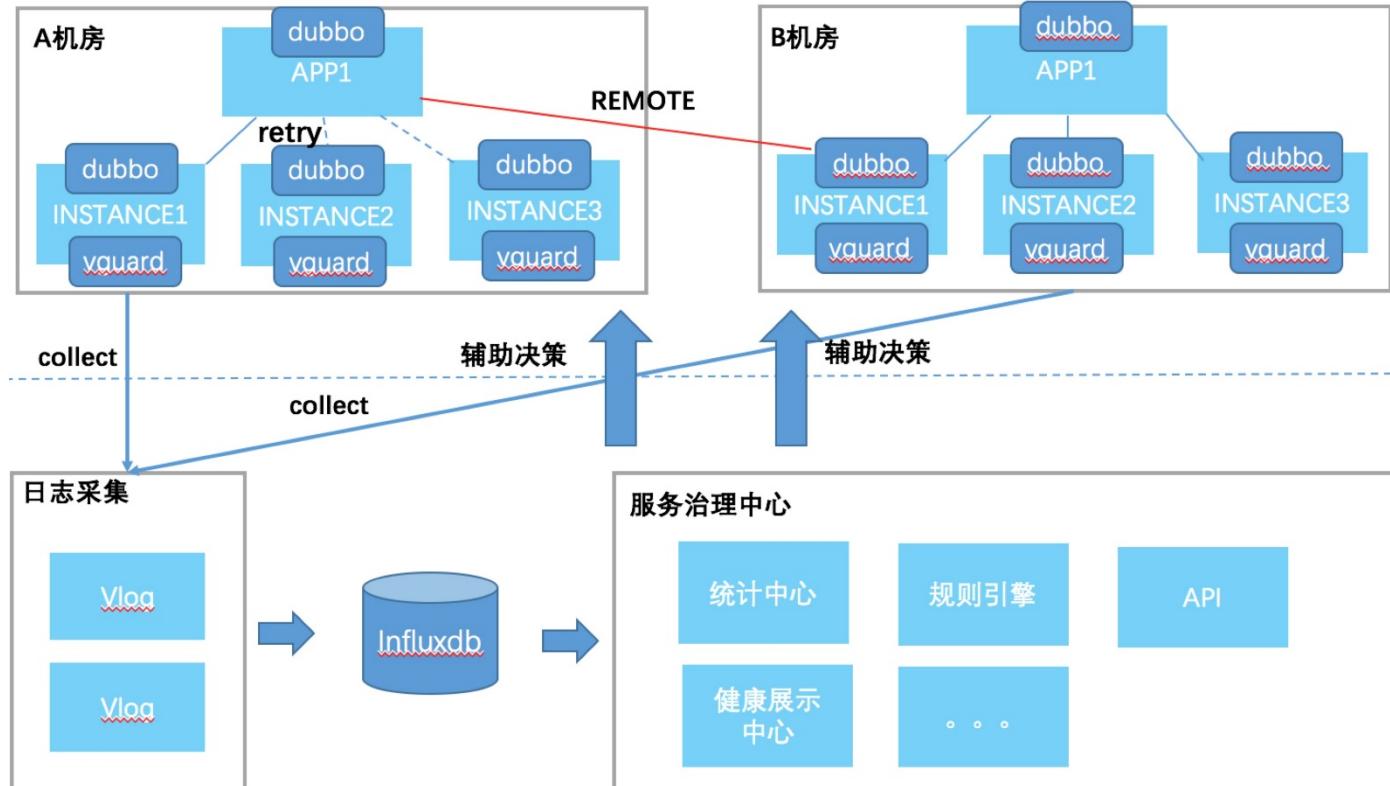
服务化-应用流量熔断限流

● 解决的问题：

- 大流量时候边缘服务拖垮核心服务
- 大流量时候限制



服务化-业务挑战双活



● 需求：迁移机房(同城双活)

● 策略：

- 新老机房公用同一个dubbo root根节点
- 新机房的服务注册在同一个服务目录之下，url中需增加belongTo属性，标识是哪个机房的服务
- 如果本地服务全部挂了，将路由到远端执行

● 修改点：

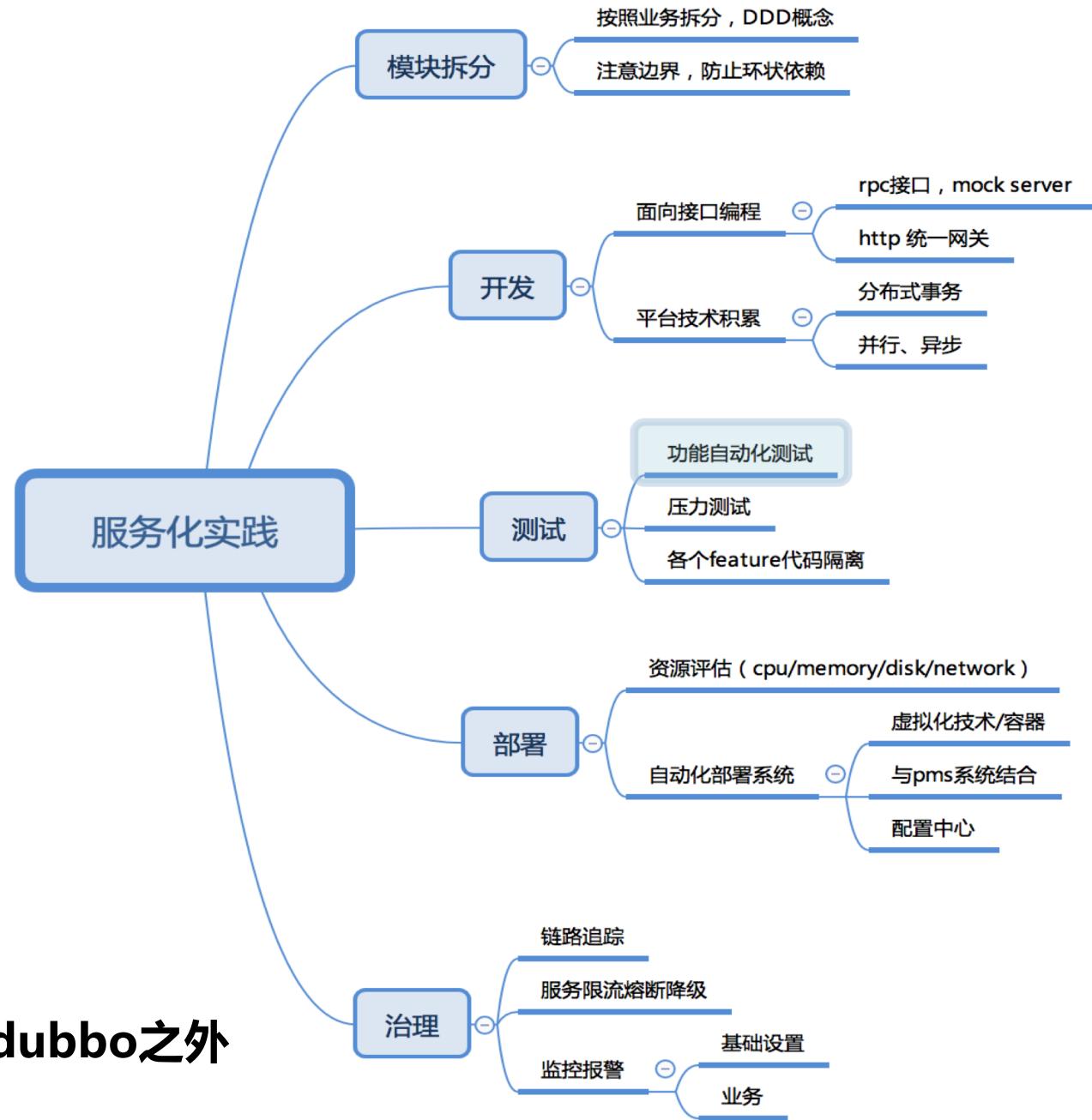
- dubbo router策略需要重新实现
- dubbo 超时重试策略需要重新实现

服务化-总结

感谢dubbo对业务的支撑



总结：用Dubbo做服务化治理，要发功在dubbo之外



Dubbo和微店的服务化实践历程分享

演讲者：王欣

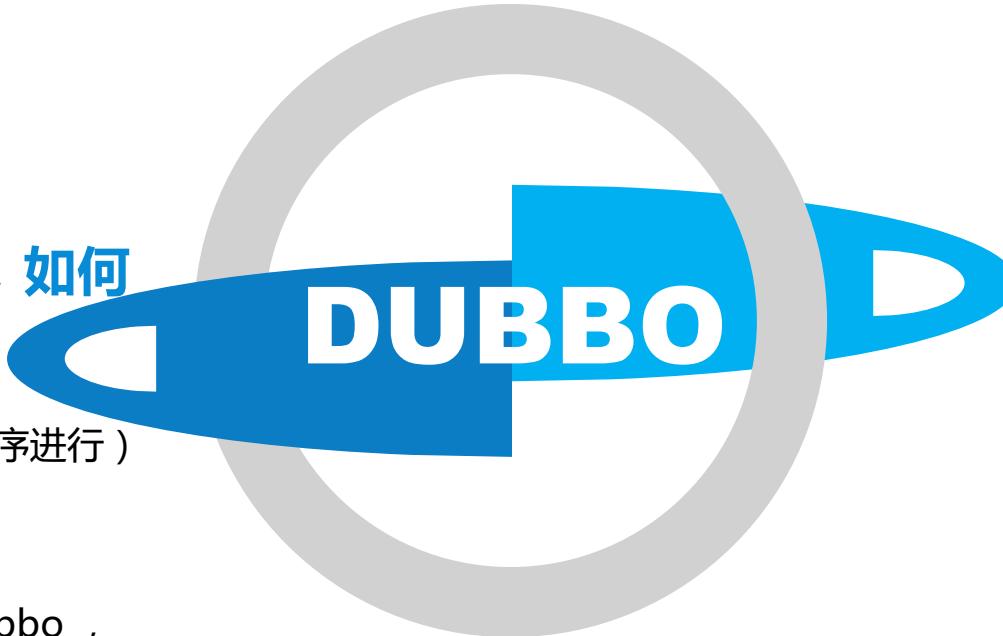


- ▶ **微店是谁**
- ▶ **微店的各技术阶段**
- ▶ **挑战&解决方案**
- ▶ **畅想未来**

畅想未来-投桃报李，回馈社区

业务公司维护私有dubbo版本，如何
更换成最新dubbo版本？

- 序列化兼容性（在滚动升级中保证业务有序进行）
- 在apache社区毕业后，包会由
`com.alibaba.dubbo`变成`org.apache.dubbo`，
考虑提供一个扩展插件



公司维护私有dubbo版本,如
何与dubbo官方版本融合？

- 提交自有的扩展包给dubbo社区
- dubbo官方在rpc跟踪、限流熔断、多
语言支持等生态上做好扩展和最佳实
践问题

畅想未来-千唤万唤，多语言支持

● 方法：

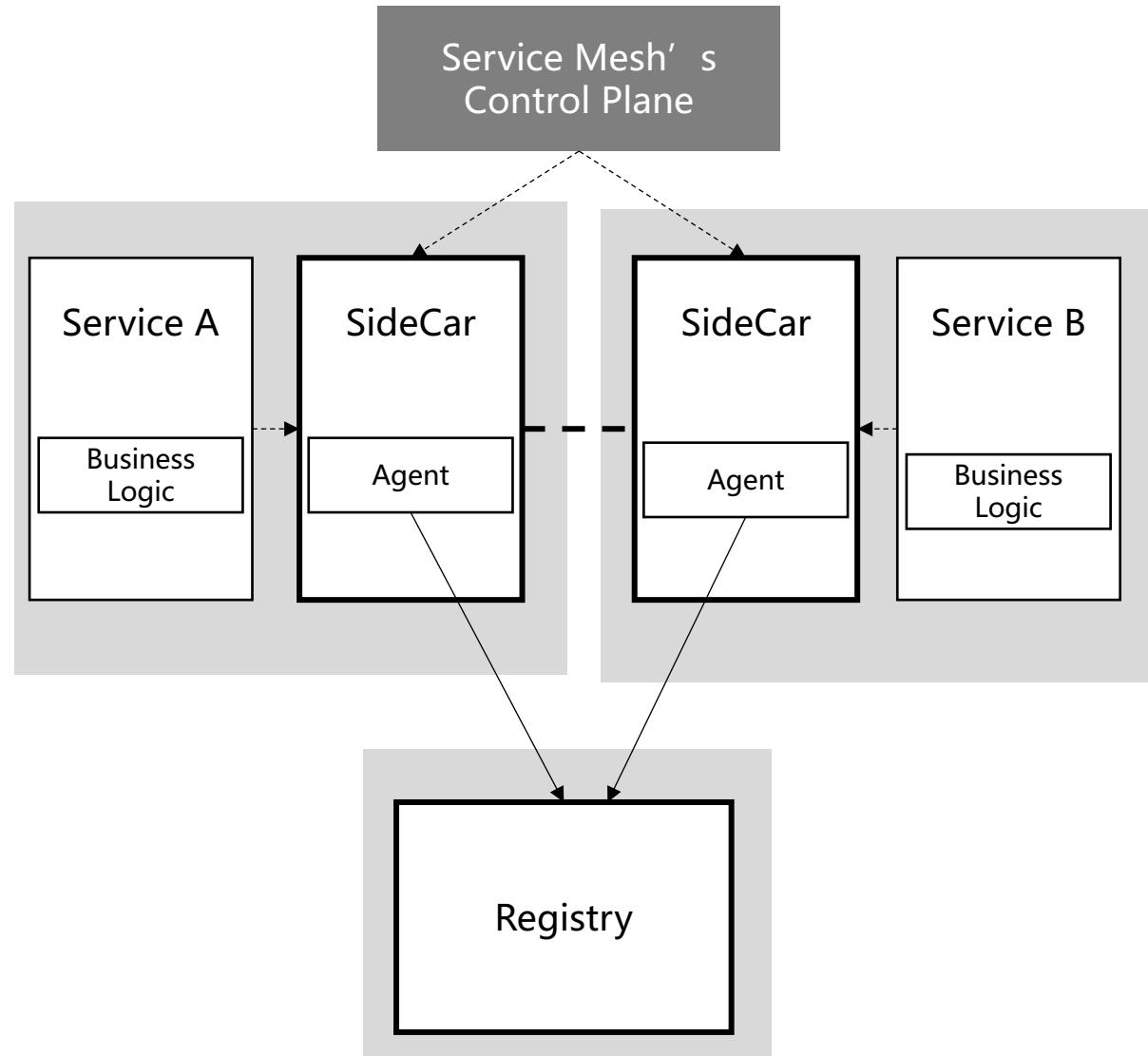
用一种高效的语言开发一个agent，java以外的语言可以通过此agent进程和注册中心进行交互。（Service mesh 边车模式）

● 收益：

不用每种语言都再实现一个dubbo client（在多语言沟通时候，thrift、gRPC等基于tcp协议之上的序列化框架会比http1+json的方式性能更好）序列化和spring cloud对比（选型时候）

● 问题：

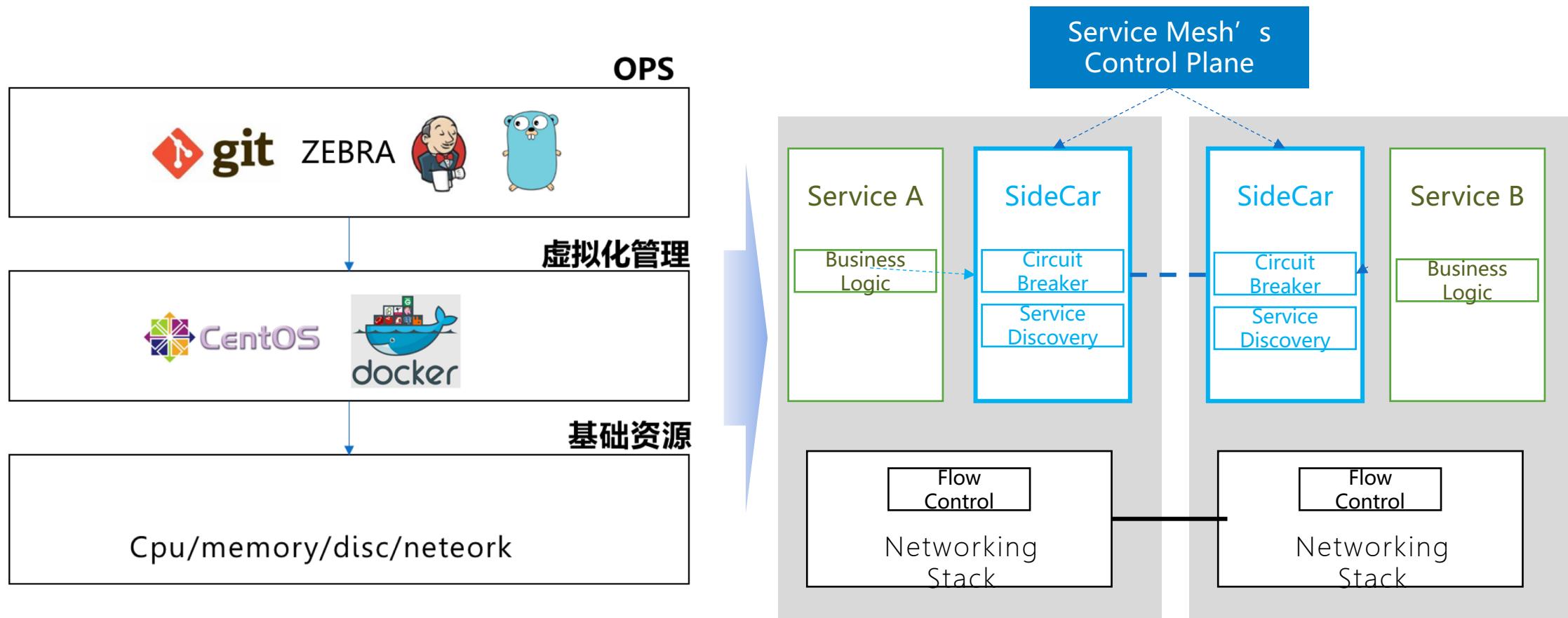
增加了一个节点：在agent进程出异常或者挂掉的时候，此agent服务的业务进程都得摘掉流量



畅想未来- 银弹 or not , Service mesh 之于 dubbo/SC

观点： 1、对于技术的使用要看业务发展阶段

2、对于新的生态：利用他的一些理念，例如sidecar，比如这一次多语言客户端的实践



畅想未来- 重磅炸弹，Dubbo+响应式编程

Finagle基本思想：

- 1、Service as a function;
- 2、function是符合函数式编程范式的：
透明引用、无副作用、可组合、可变换等；
- 3、function的返回值统一都是
Future[Xxx]/Try[Xxx]/Option[Xxx].
Try等又是一个Monad 提示：参照函
数式编程里Monad的基本概念；
- 4、基于上述三点，整个系统可像搭积木
一样构建；

响应式编程：

微服务需要考虑其响应能力，即

- 1、弹性(面对失败的响应)；
 - 2、伸缩性(不同负载情况下响应)
 - 3、消息驱动(松耦合、通过异步消息隔离)
- 而响应式编程要求：消息驱动、是为失败
而设计，异步。故需要响应式编程；

第四届阿里中间件性能挑战赛

语言不限，500,000w现金大奖，40,000w西行游学基金



扫码挑战世界级技术难题

社区地址：

- dubbo.apache.org
- dubbo.io

Repos :

- github.com/apache/incubator-dubbo
- github.com/dubbo

Email list

- dev@dubbo.apache.org

About me :

- Github: <https://github.com/lovepoem>
- Wechat: lovepoem1