Kong做微服务网关的实践

caishu

2018-11

背景

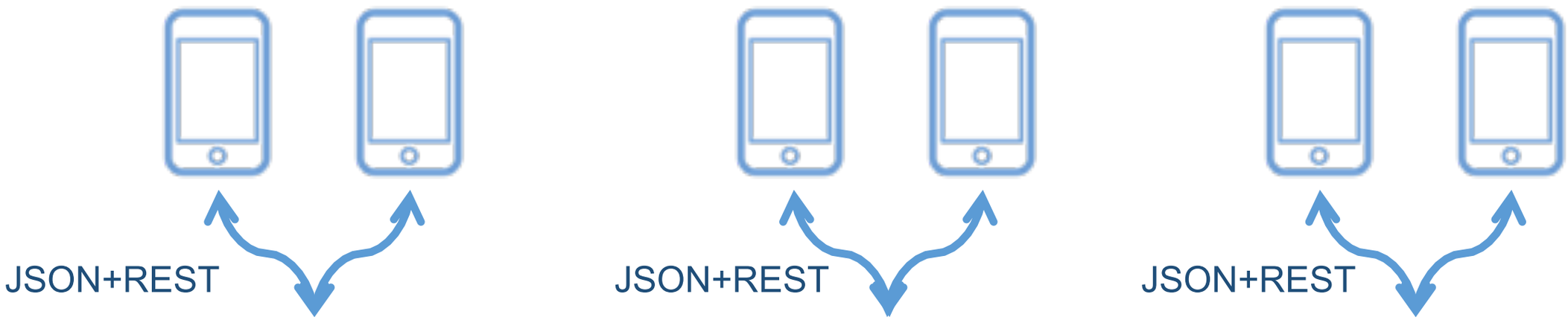
现状：

- 某银行信用卡中心使用spring cloud为基础的微服务架构
- 40左右个服务，生产部署300+ jvm
- 两地三中心
- 新的功能在kubernetes上开发和部署

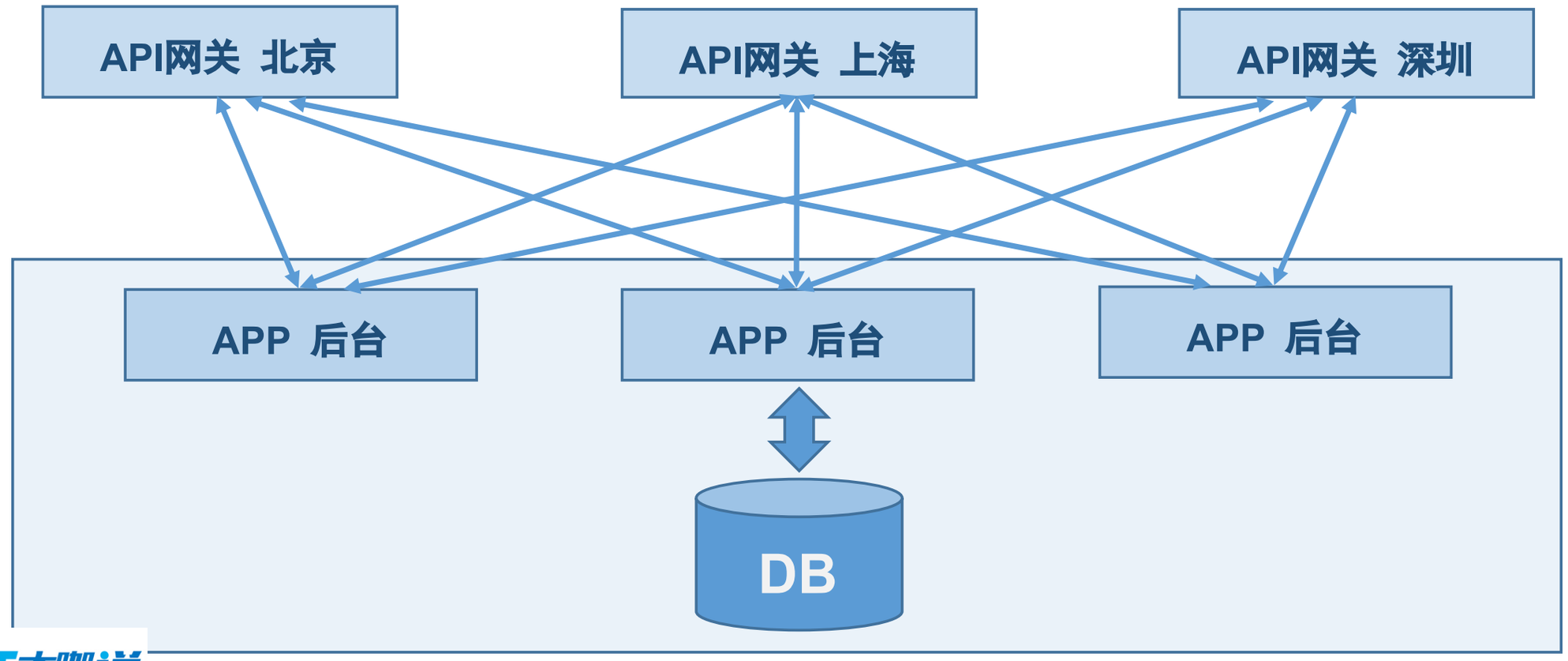
需要解决的问题：

- 入口流量需要清洗，恶意访问、机器人、注册机、薅羊毛
- 多认证渠道，CRM、微信、微博、QQ、百度账号等
- 有些内部服务希望可以快速便捷的开放给外部
- 服务在虚拟机和容器平台之间互相访问
- 服务注册和发现，目前用Eureka
- 服务网关，目前用Zuul
- 服务质量管理：延迟，错误率，访问统计
- 服务访问控制：内网也需要访问控制
- 服务间调用链和拓扑
- 精细路由

应用拓扑

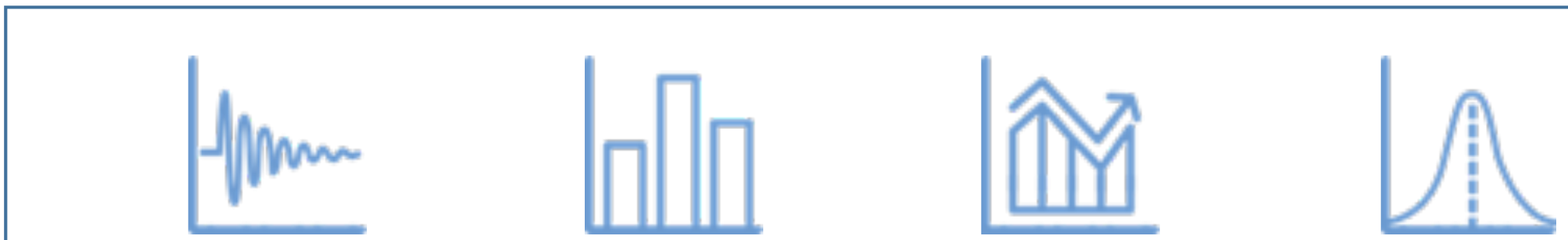


网关



OpenResty™ | IT大咖说 知识共享平台

部署拓扑



流量清洗

问题：

- 机器人，注册机，爬虫，恶意（SQL注入、脚本注入），“薅羊毛”

解决办法：

- WAF插件
- 机器人识别，机器学习
- 限速
- 一次性token

多渠道认证

问题：

- 各种登录认证被加入：客服系统，微信、微博、QQ、百度等

解决办法：

- 不同认证方式的插件
- 内部使用统一的用户标识

服务快速开放

问题：

- 内部服务需要对外开放——键开放服务

解决办法：

- 访问控制插件
- 默认IP白名单

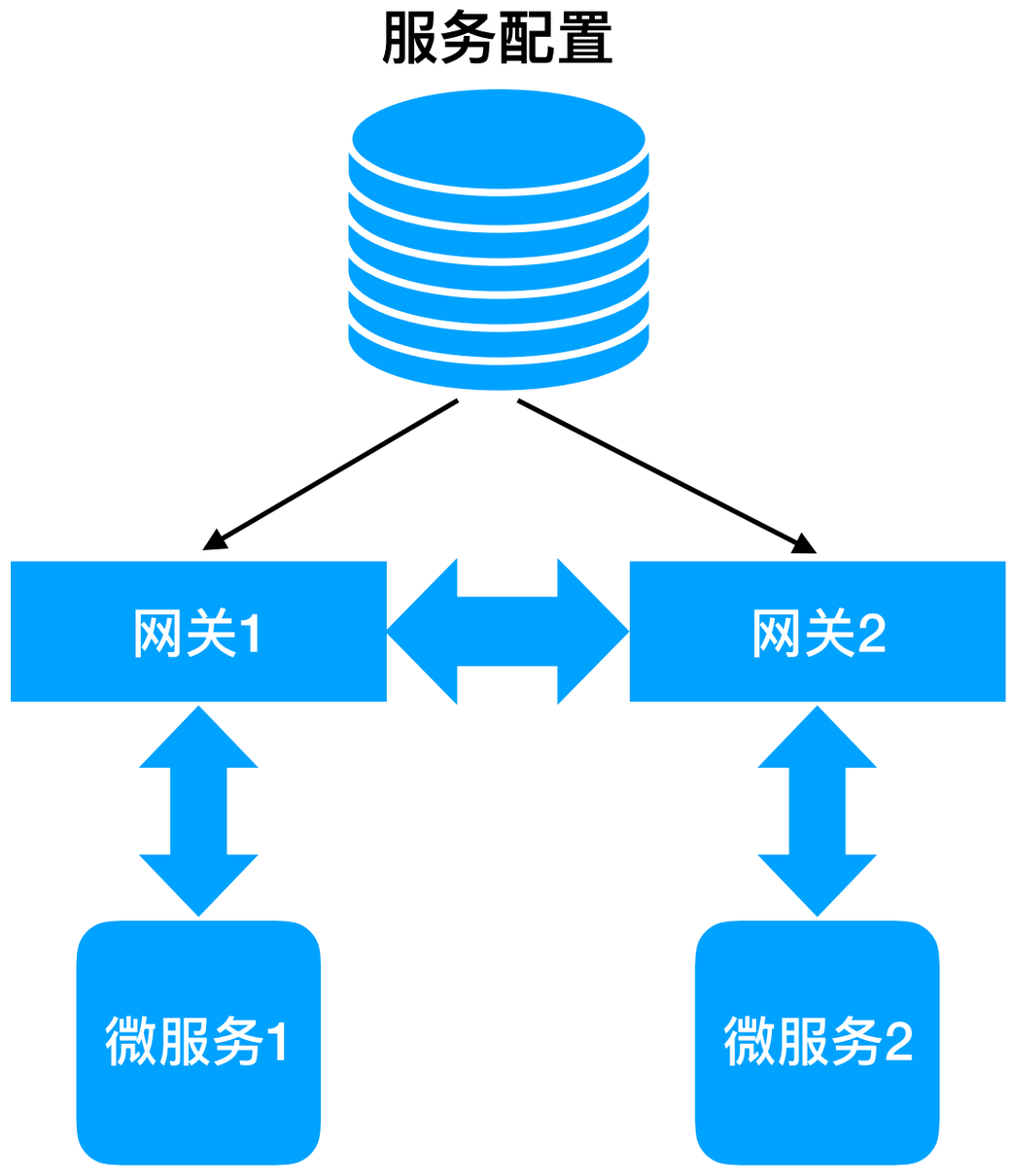
多平台微服务互访

问题：

- 基于虚拟机的微服务平台
- 基于容器的微服务平台

解决办法：

- 网关集联
- 自动路由



服务注册与发现

问题：

- 之前用Eureka，遇到可靠性问题
- 服务的信息，实例的信息，如何展示和可视化

解决办法：

- 代理Eureka的请求，通过数据库管理服务注册信息
- 服务发现通过kong的service+route+upstream实现

服务网关

问题：

- Zuul，需要很多定制

解决办法：

- Kong替代Zuul
- 多种插件开箱即用

服务质量管理

问题：

- 核心的服务质量指标，访问量、延迟、相应时间等
- 服务的细粒度统计。以endpoint为依据，而不是部署单元为依据

解决办法：

- “子服务”的自动发现，人工审批
- 替代prometheus的实现。用SQL做统计分析，兼容多种BI工具，比如superset

服务访问控制

问题：

- 服务之间调用需要权限控制
- 内审和内部风控

解决办法：

- 访问控制
- 机器学习

调用链和拓扑

问题：

- APM实施遇到困难，效果没有达到设计目的
- 需要更简洁的方案解决调用链的整理和展示

解决办法：

- Open tracing插件
- 日志结合open tracing数据
- 可视化展示

精细路由

问题：

- 基于特定的信息作路由：地理位置、运营商、用户等级
- 根据来源IP地理位置升级版本
- VIP用户处理

解决办法：

- 改进kong的route->service的识别和判断逻辑
- http请求信息的“增强”：比如IP到GEO的转换
- 根据“标签”做路由匹配：比如Request contains “SH”, route to Service with tag “SH”

Q&A

OpenResty™ | IT大咖说 知识共享平台