

构建微服务云原生应用

架构师杨波



扫码试看/订阅

《Spring Boot & Kubernetes 云原生微服务实践》

CHAPTER

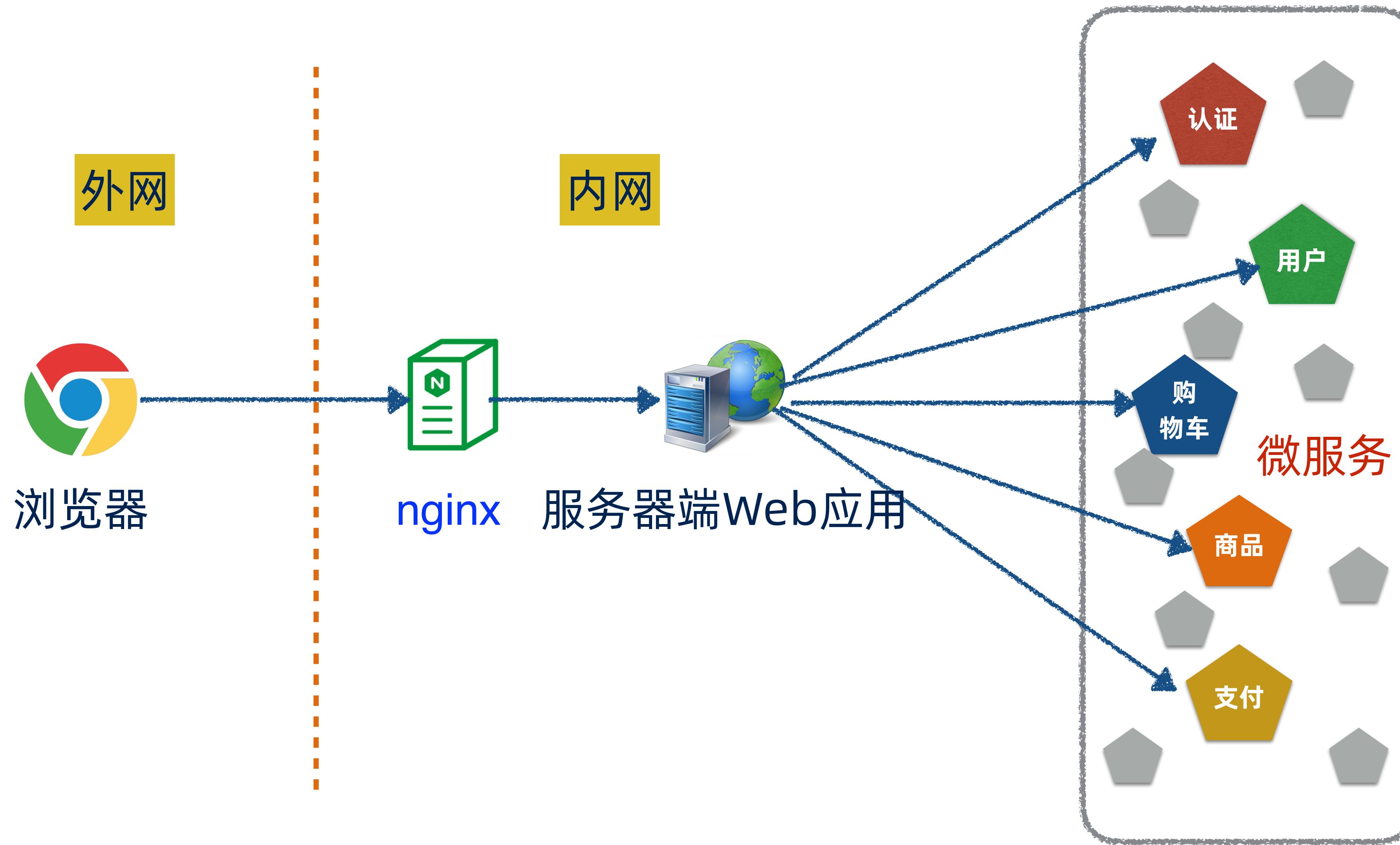
04

可编程网关设计和实践

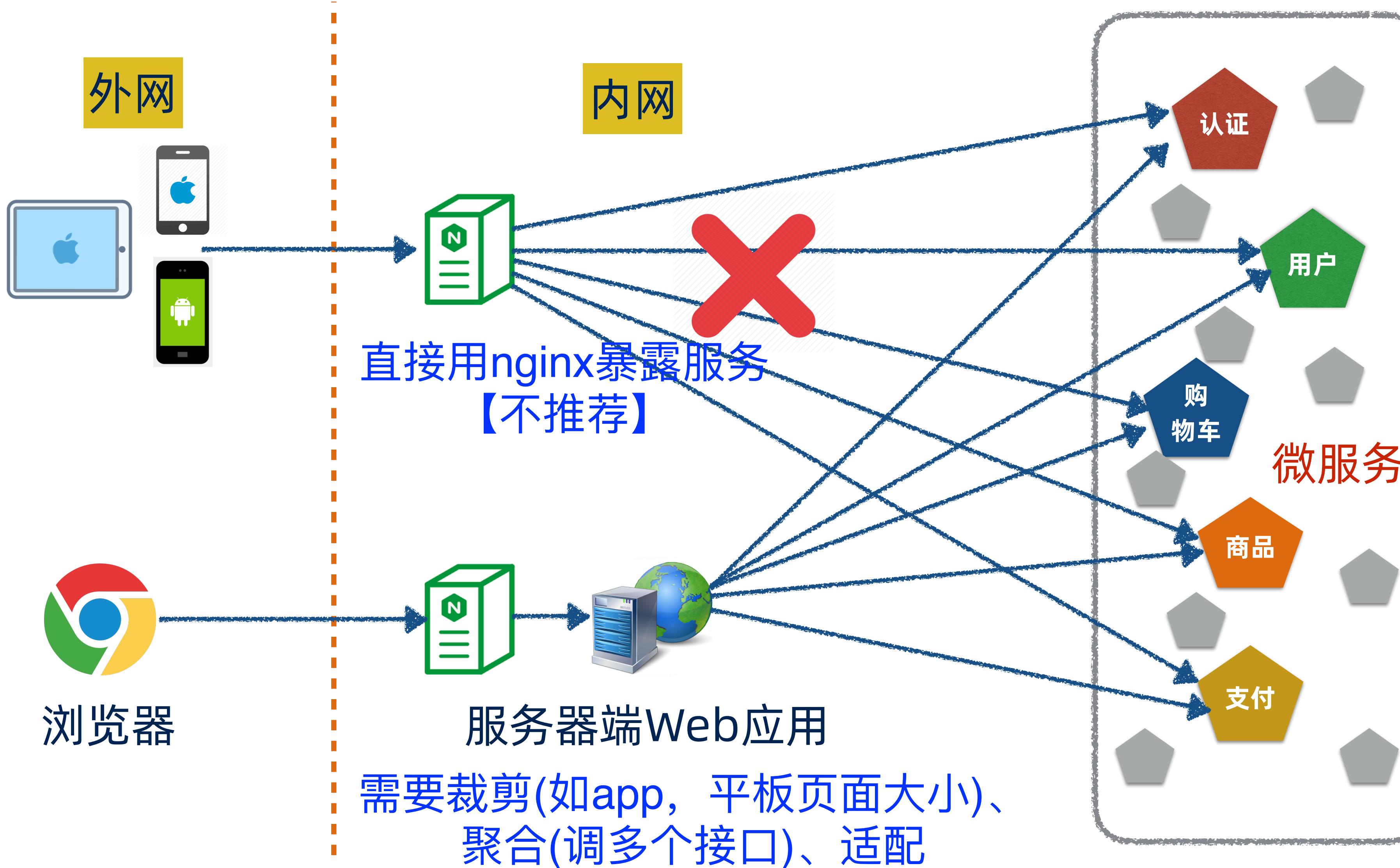
第 1 部分

网关和 BFF 是如何演化出来的(上)

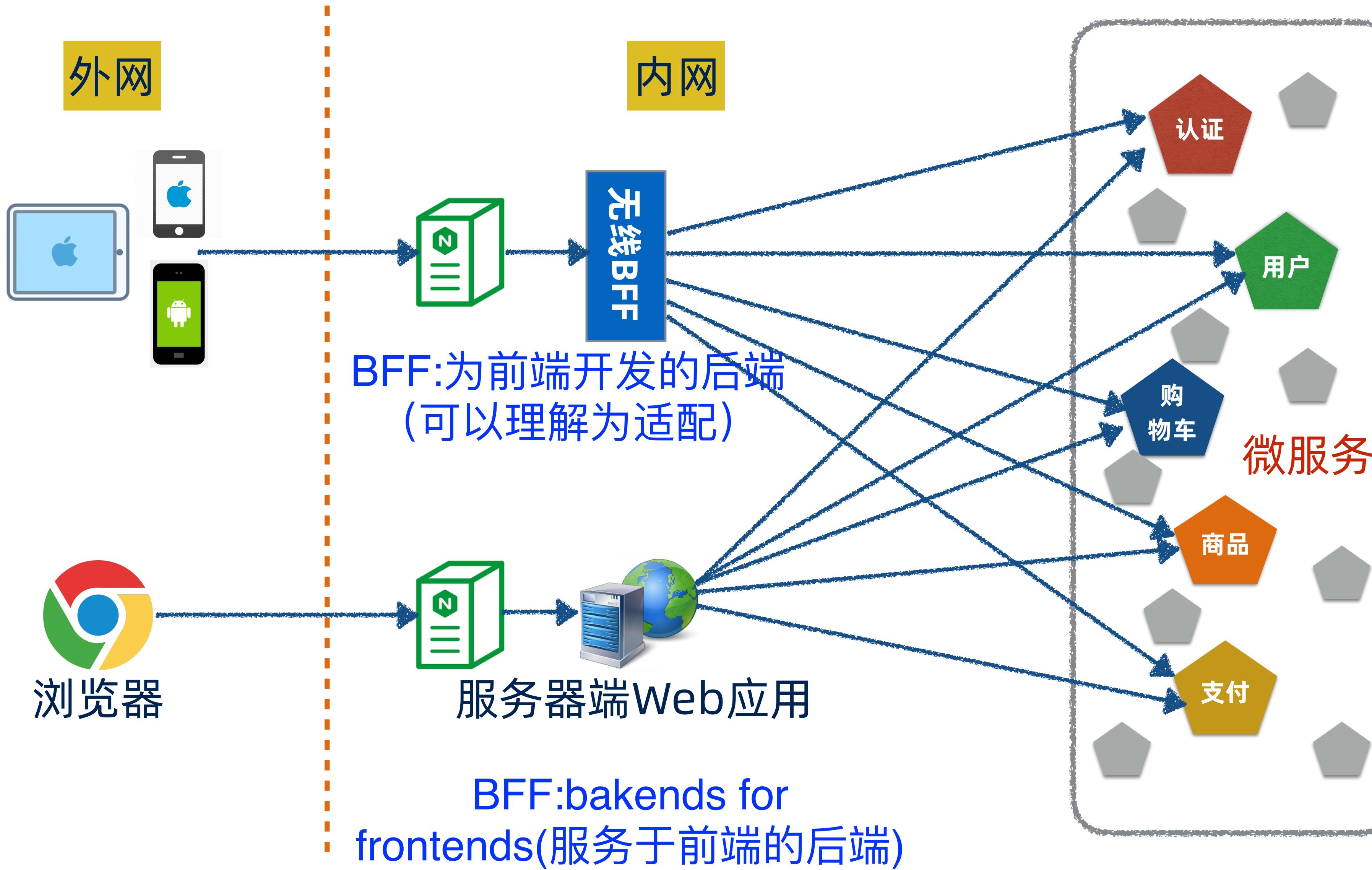
MyShop SOA V1



MyShop SOA V2



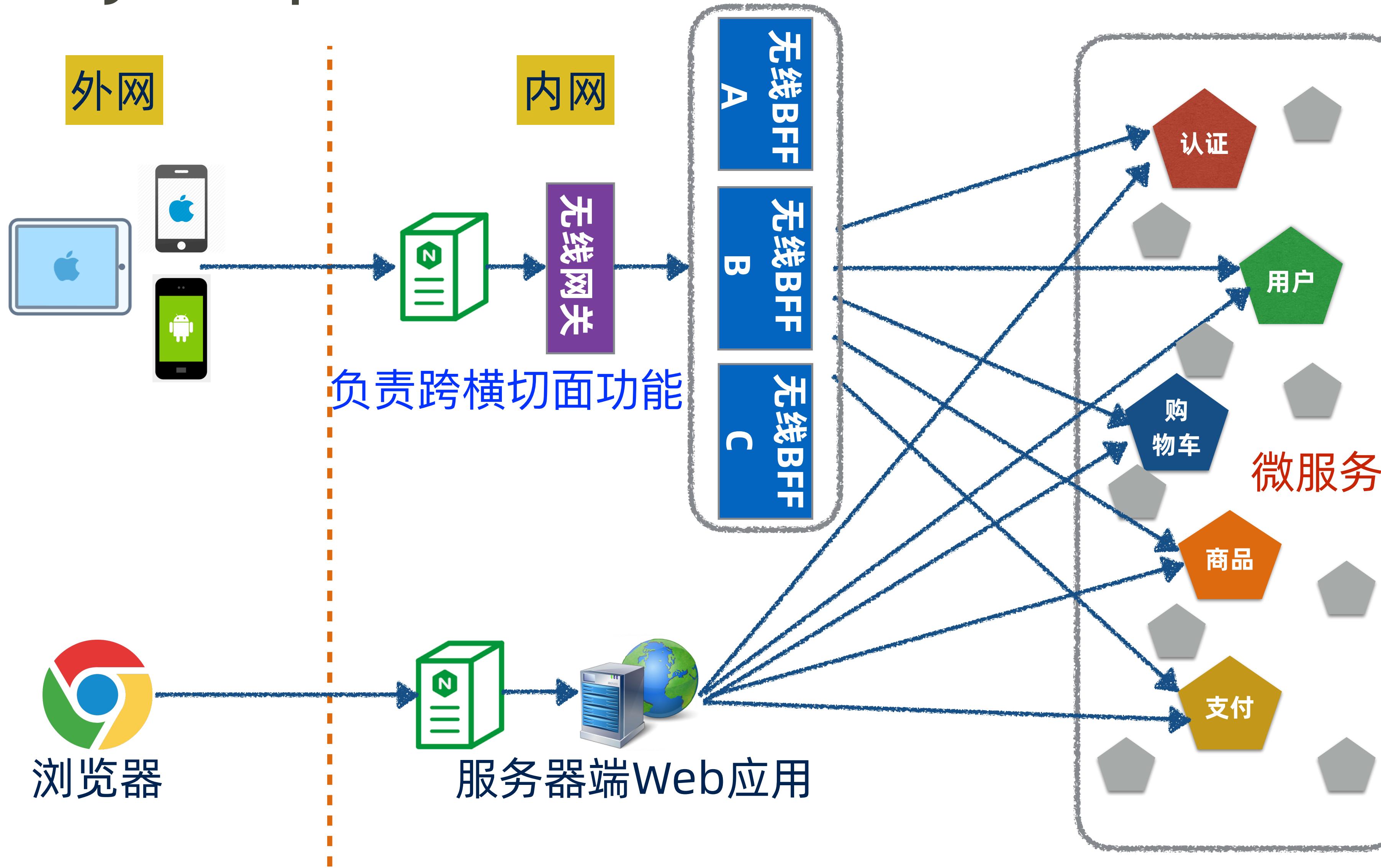
MyShop SOA V2.5



第 2 部分

网关和 BFF 是如何演化出来的(下)

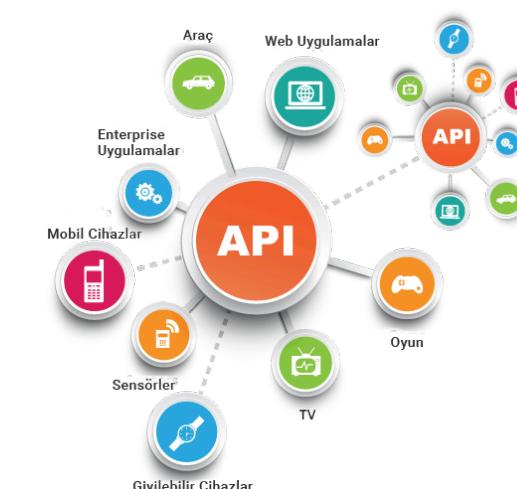
MyShop SOA V3 不同业务解耦



MyShop SOA V4

第三方应用

外网



H5前后分离应用



浏览器



用户体验层

用网关取代 nginx集群

开放平台网关

H5网关

无线网关

Web应用网关

网关层

开放平台BFF层

H5 BFF层

无线BFF层

服务器端Web应用

BFF层

内网

认证

用户

购物车

商品

支付

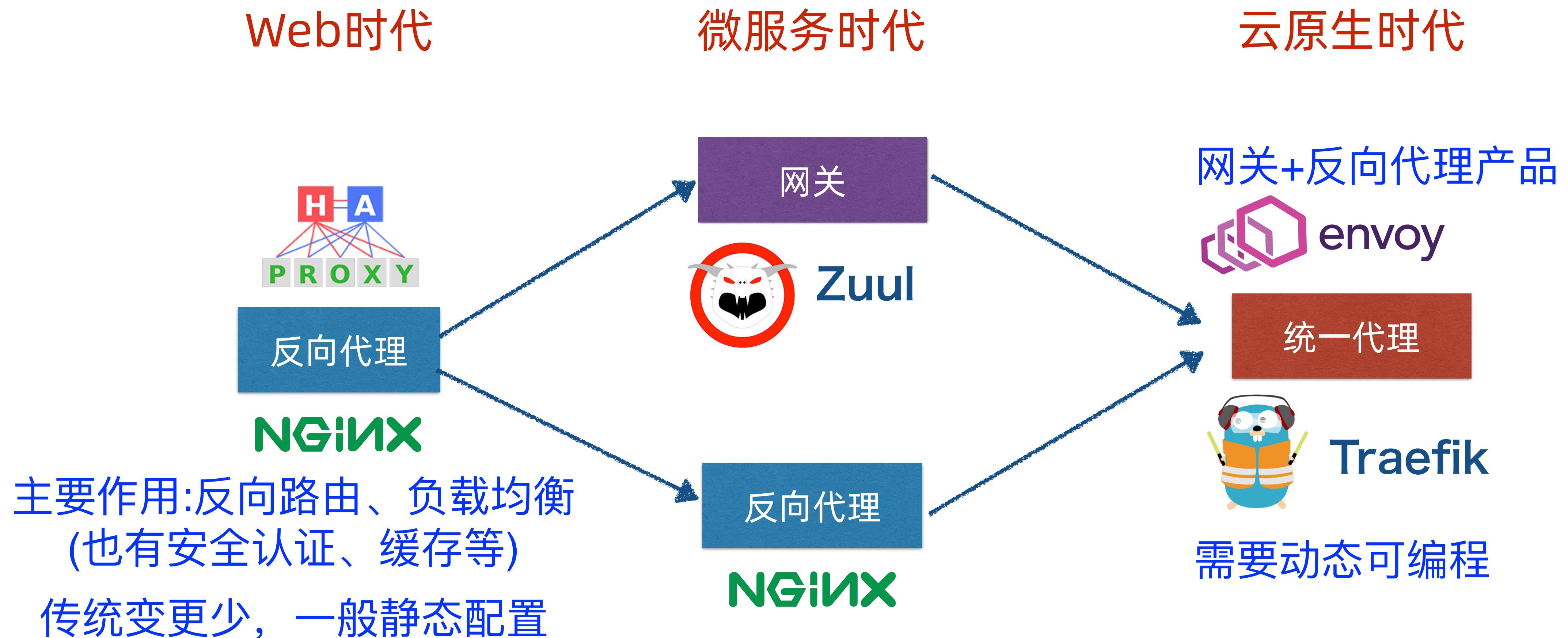
微服务

微服务层

第 3 部分

网关和反向代理是什么关系？

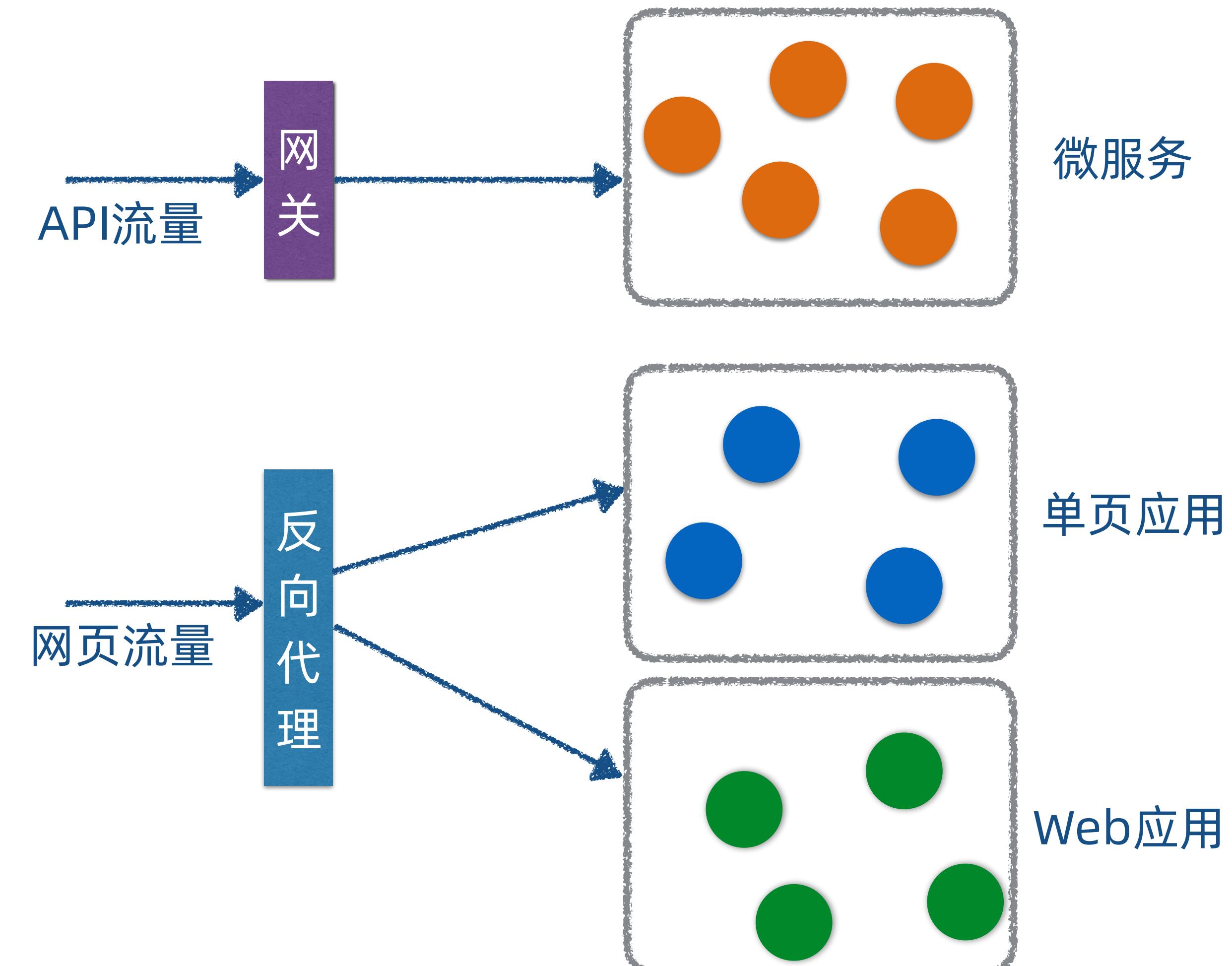
反向代理 vs 网关



第 4 部分

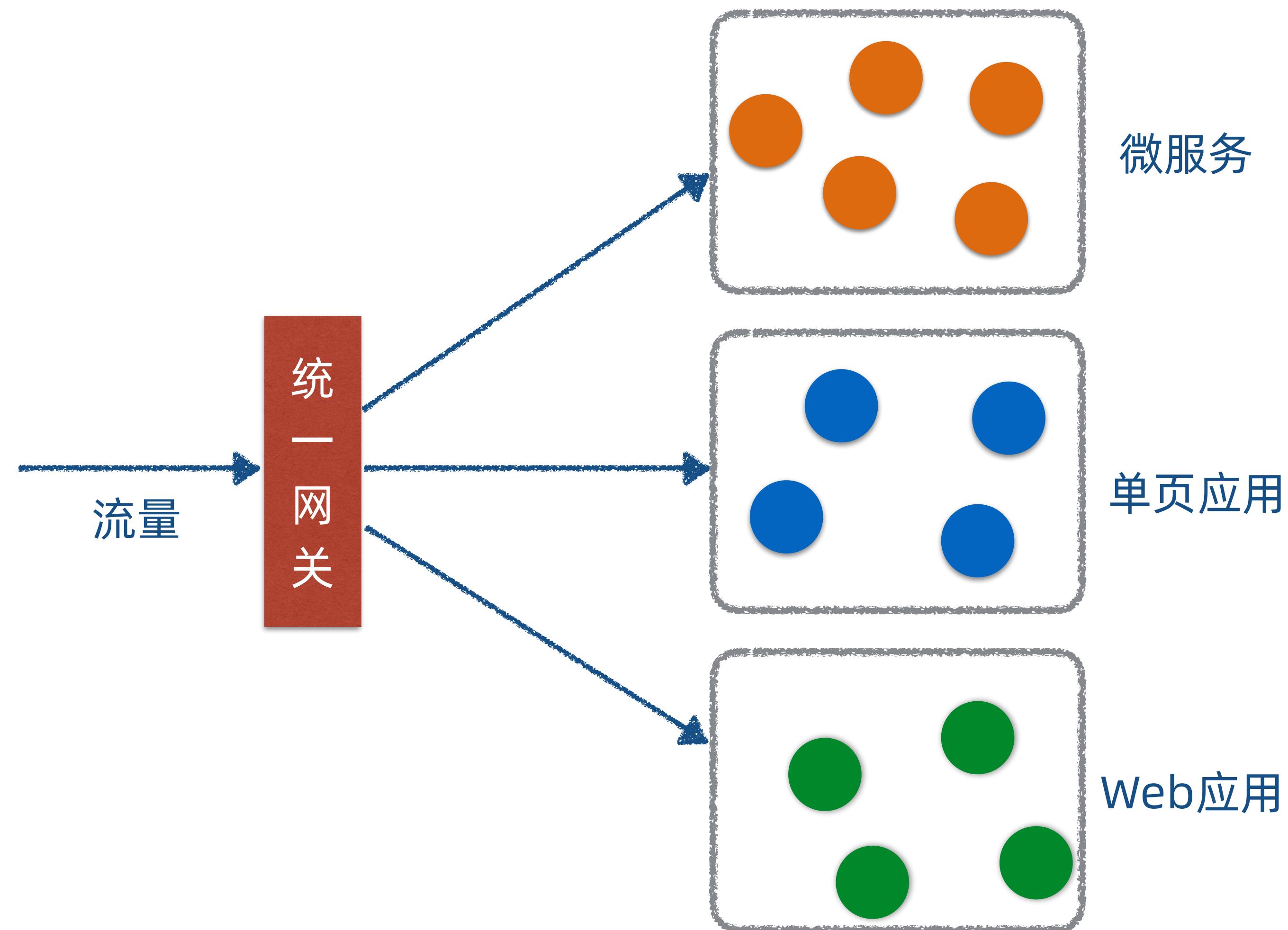
网关需要分集群部署吗？

反向代理+网关部署架构

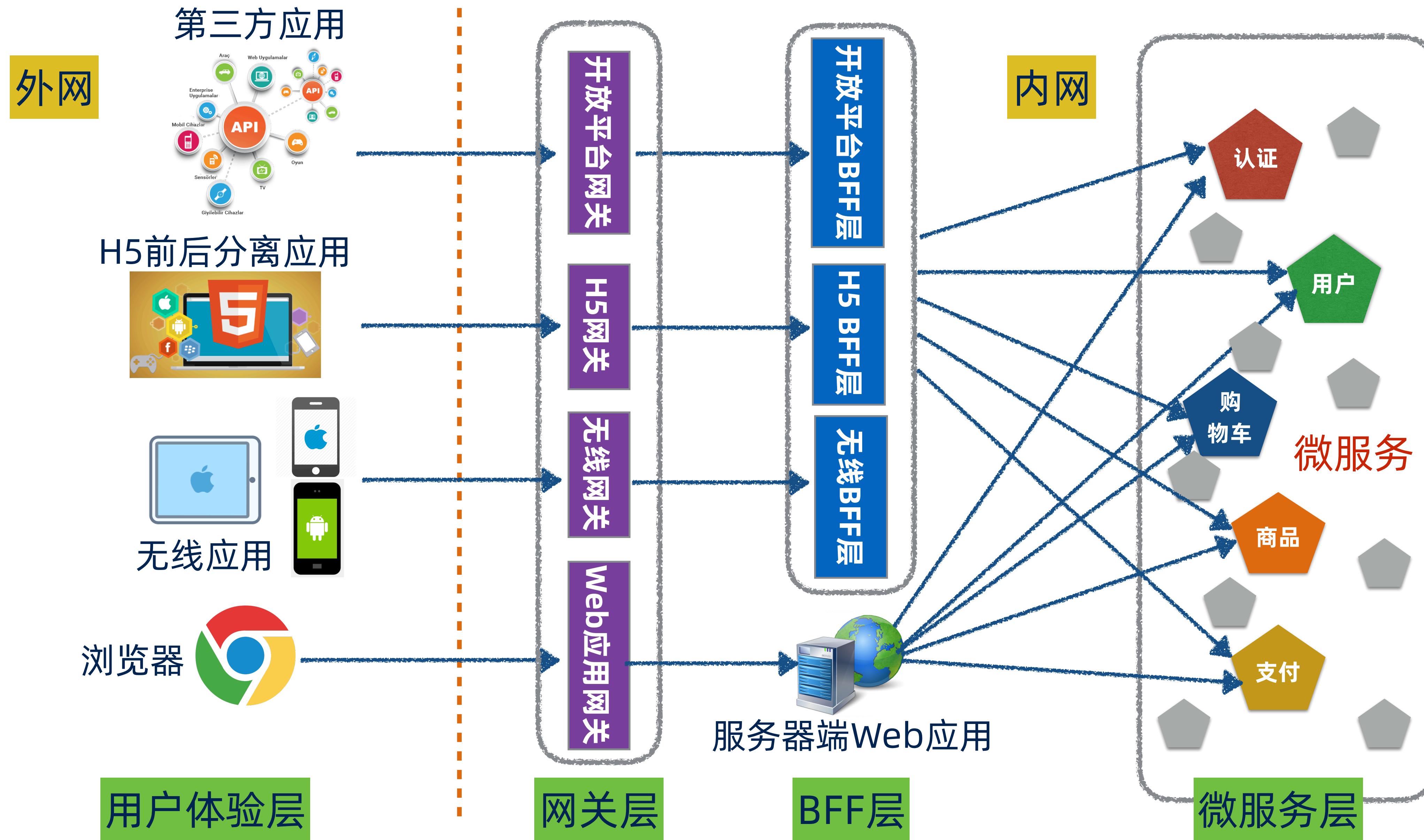


目前很多公司两者并存

统一网关部署架构



统一网关+分集群

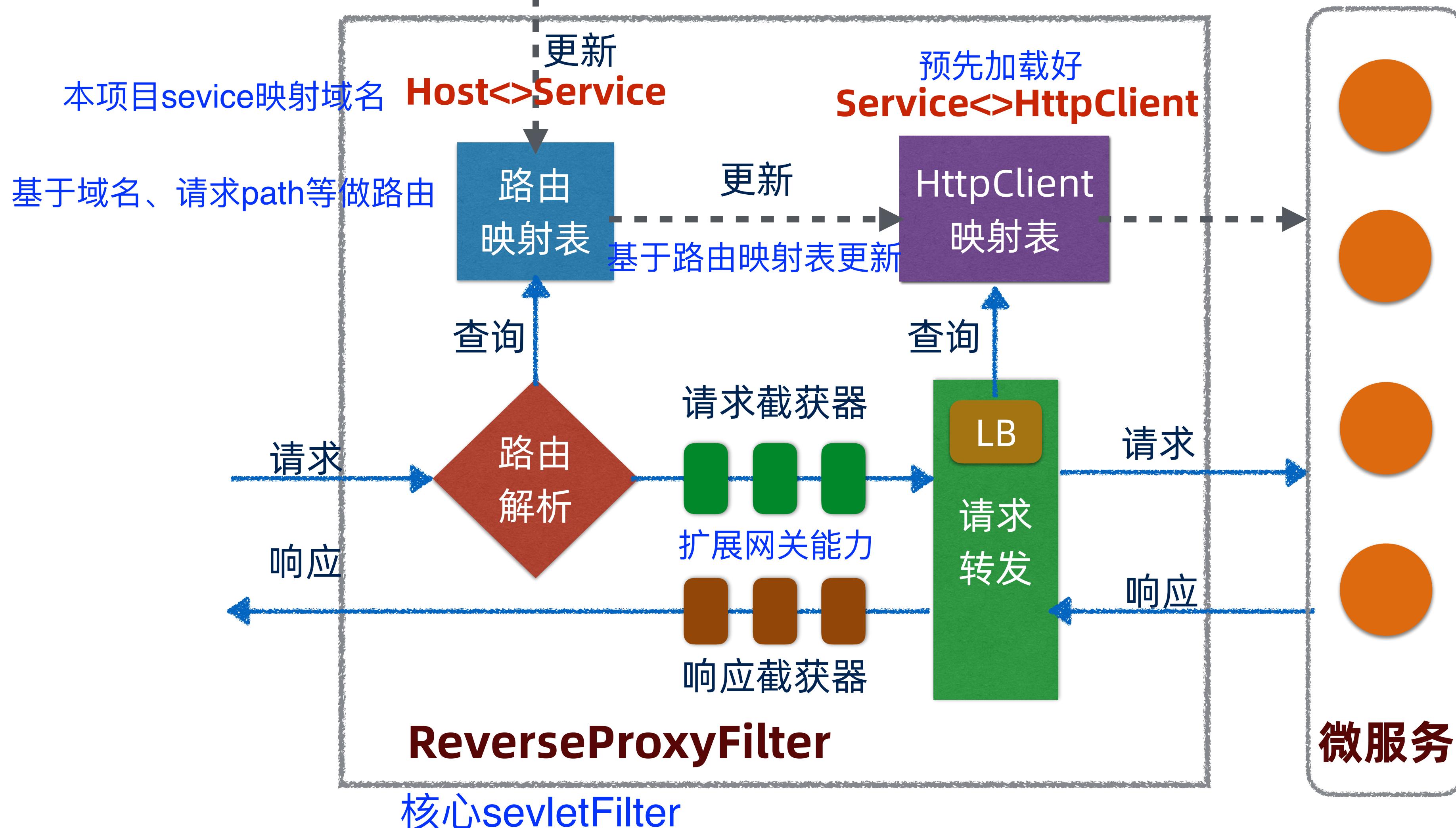


第 5 部分

Faraday 网关内核设计

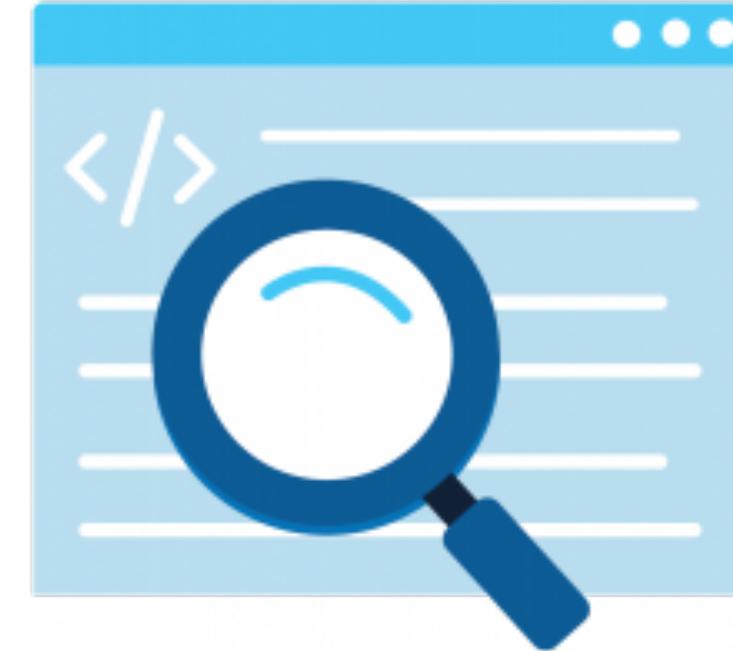
Faraday 网关内核设计

本项目开发自定义可编程轻量级网关



静态路由配置

```
faraday:  
  enable_programmatic_mapping: false  
  tracing:  
    enabled: false  
  mappings:  
    -  
      name: faraday_route  
      host: faraday.staffjoy-v2.local  
      destinations: httpbin.org  
    -  
      name: account_route  
      host: account.staffjoy-v2.local  
      destinations: localhost:8081  
    -  
      name: company_route  
      host: company.staffjoy-v2.local  
      destinations: localhost:8082  
    -  
      name: ical_route  
      host: ical.staffjoy-v2.local  
      destinations: localhost:8083  
    -  
      name: whoami_route  
      host: whoami.staffjoy-v2.local  
      destinations: localhost:8084  
      timeout:  
        connect: 10000  
        read: 10000
```

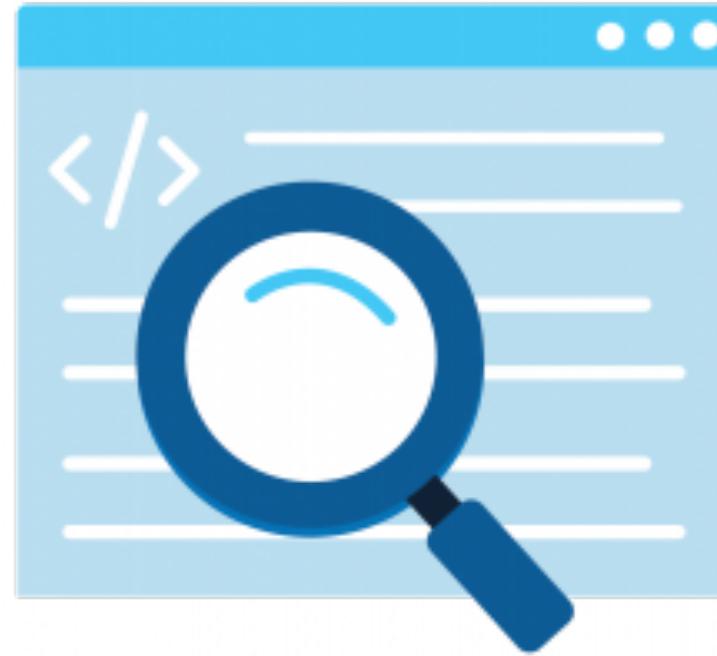


第 6 部分

Faraday网关代码解析(上)

路由映射表代码:mapping目录下
httpClient映射表代码:http目录下

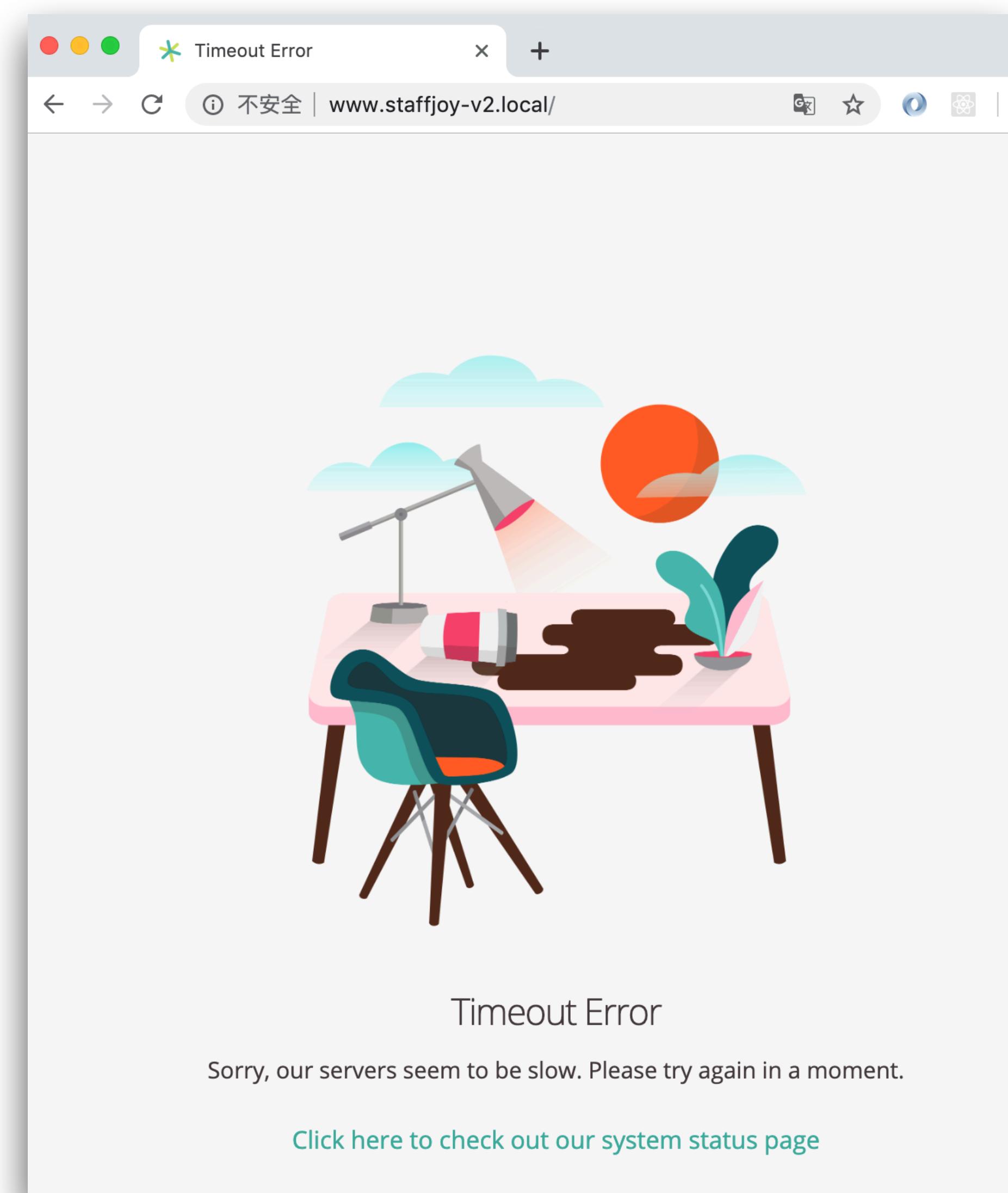
一个请求发起
==> 1. 调用 **MappingsProvider** 的 **resolveMapping()** 方法获取主机头
=====> 2. 调用 **HttpClientProvider** 的 **updateHttpClients()**



第 7 部分

Faraday网关代码解析(下)

统一异常处理



第 8 部分

生产级网关需要考虑哪些环节？

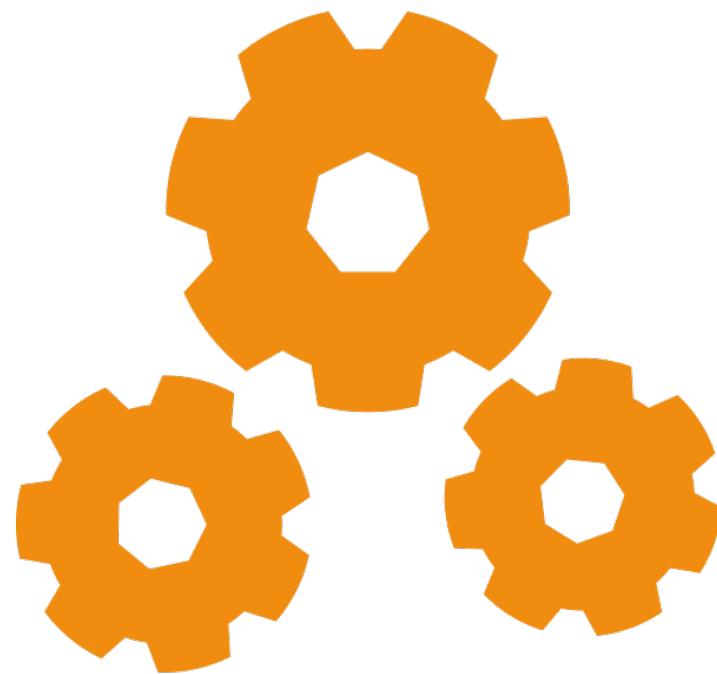
生产扩展点

需要根据自己的业务扩展

开源框架:[resilience4j](#)

- 限流熔断
- 动态路由和负载均衡
- 基于 Path 的路由
好处是只需要一个域名
 - `api.xxx.com/pathx`
- 截获器链
- 日志采集和 Metrics 埋点
- 响应流优化

建议用于大文件，从内核态拷贝到用户态，消耗性能，参见 zuul



第 9 部分

主流开源网关概览

主流开源网关概览

	支持公司	实现语言	亮点	不足
Nginx (2004)	Nginx Inc	C/Lua	高性能，成熟稳定	门槛高，偏运维，可编程弱
Kong (2014)	Kong Inc	OpenResty/Lua	高性能，可编程API	门槛较高
Zuul (2012)	Netflix/Pivotal	Java 本质是SevletFilter	成熟，简单门槛低	性能一般，可编程一般
Spring Cloud Gateway (2016)	Pivotal	Java	异步，配置灵活	早期产品
Envoy (2016)	Lyft	C++	云原生 高性能，可编程API/ ServiceMesh集成	门槛较高
Traefik (2015)	Containous	Golang	云原生，可编程API/对接各 种服务发现	生产案例不多

参考链接

1. **Nginx** <https://www.nginx.com/>
2. **HAProxy** <http://www.haproxy.org/>
3. **Kong** <https://konghq.com/kong/>
4. **Envoy** <https://www.envoyproxy.io/>
5. **Traefik** <https://traefik.io/>
6. **Zuul** <https://github.com/Netflix/zuul>
7. **Spring Cloud Gateway** <https://spring.io/projects/spring-cloud-gateway>
8. **Resilience4j** <https://github.com/resilience4j/resilience4j>
9. **Hystrix** <https://github.com/Netflix/Hystrix>
10. **Spring Cloud Netflix** <https://spring.io/projects/spring-cloud-netflix>
11. **Micrometer** <https://micrometer.io/>



扫码试看/订阅

《Spring Boot & Kubernetes 云原生微服务实践》