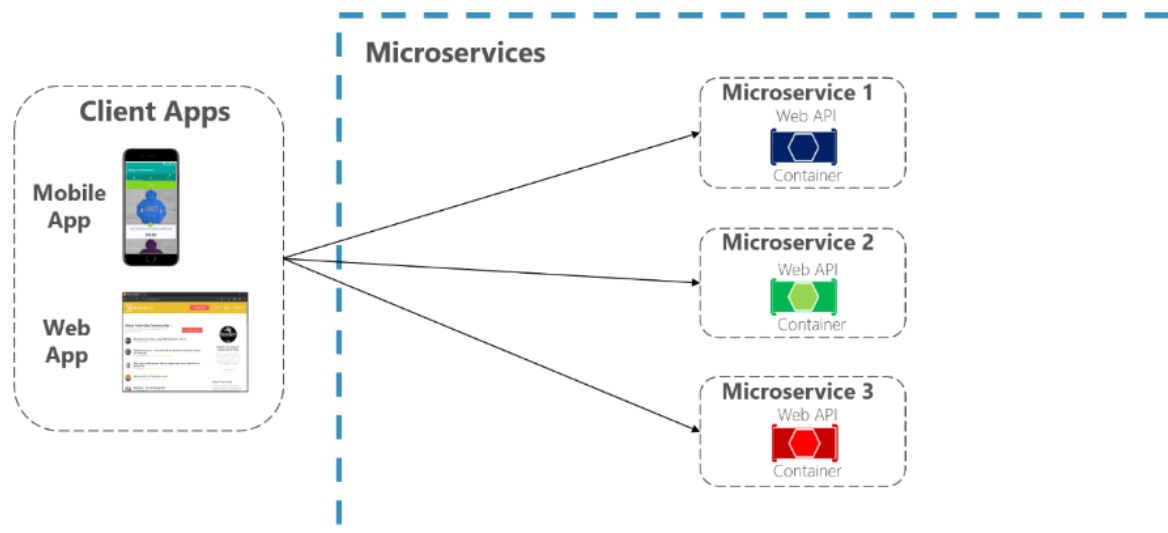


第06期-API网关模式与BFF模式

2020年1月9日 10:16

客户端到微服务直接通信



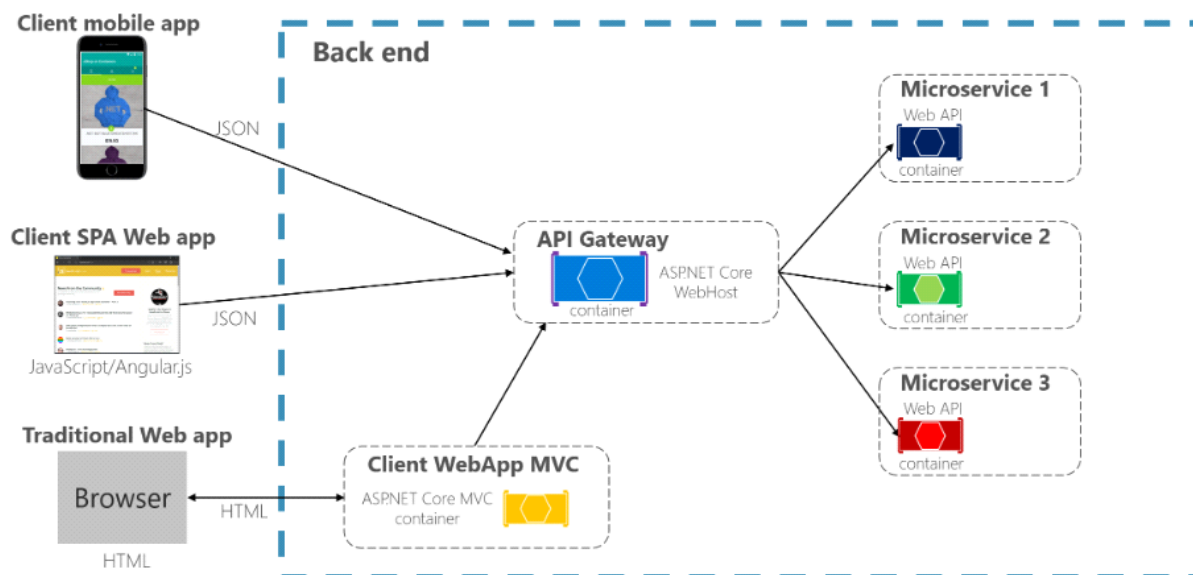
客户端直接与微服务通信，满足基本需求，但对于一些大型复杂的应用程序，需要处理大量的微服务通信，将面临一些问题：

- 单个 UI 需要请求多个微服务完成功能，多次请求增加延迟时间和复杂性。
- 每个微服务需要单独授权，无法统一授权。
- 无法根据客户端灵活转换传输协议。
- 客户端无法灵活定制界面呈现，针对web和不同移动端所需的功能和优化是不同的。
- 耦合度高，协调性复杂，难以维护，微服务终结点变化直接影响用户客户端。

客户端到 API 网关的通信

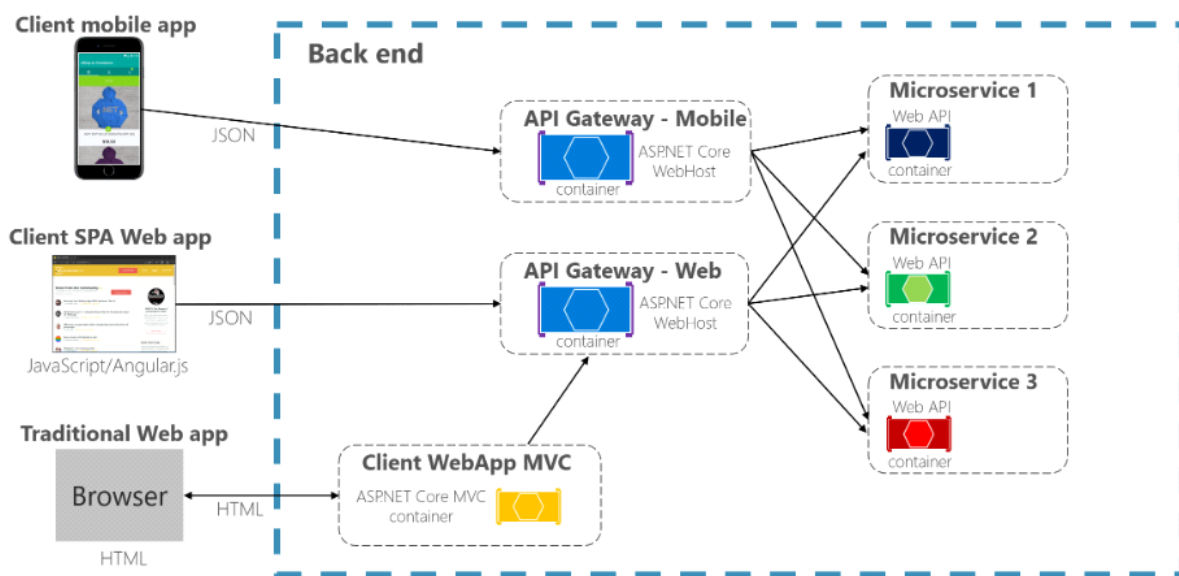
基于微服务的应用程序来说，有一个间接的中间网关层将非常方便，网关使得微服务调用具有单一入口，外观模式，重点为客户端提供定制需求，所以 API 网关模式有时也称为“基于前端的后端模式” 模式(BFF)。

Backend For Frontend



API 网关将作为自定义 ASP.NET Core WebHost 服务实现，并作为容器运行。

业务增长，API 网关压力增大，建议将 API 网关拆分成多个更小粒度的网关（根据业务），最好不要将整个项目的所有微服务都挂到一个网关，这违法微服务的自主性。



API 网关模式中的主要功能

- 反向代理和网关路由，Nginx 也算一种反向代理网关路由模式的实现。
- 请求聚合，将多个微服务聚合到单一请求。
- 横切关注点，身份验证和授权、服务发现集成、响应缓存、重试策略、断路器和 QoS、速率限制和遏制、负载均衡、日志记录、IP 允许列表。

实现带 API 网关功能的第三方库

Nginx

一个高性能的 HTTP 反向代理 WEB 服务器。

正向代理和反向代理

Ocelot

轻量级、简单、开源、基于 .NET Core 平台、容器化，跨平台。

<https://github.com/ThreeMammals/Ocelot>

Envoy

Envoy 是专为大型现代微服务架构设计的代理和通信总线。C++编写。

<https://github.com/envoyproxy/envoy>

Azure API Management

API 管理 (APIM) 是一种为现有后端服务创建一致且现代化的 API 网关的方法。

<https://azure.microsoft.com/zh-cn/services/api-management>

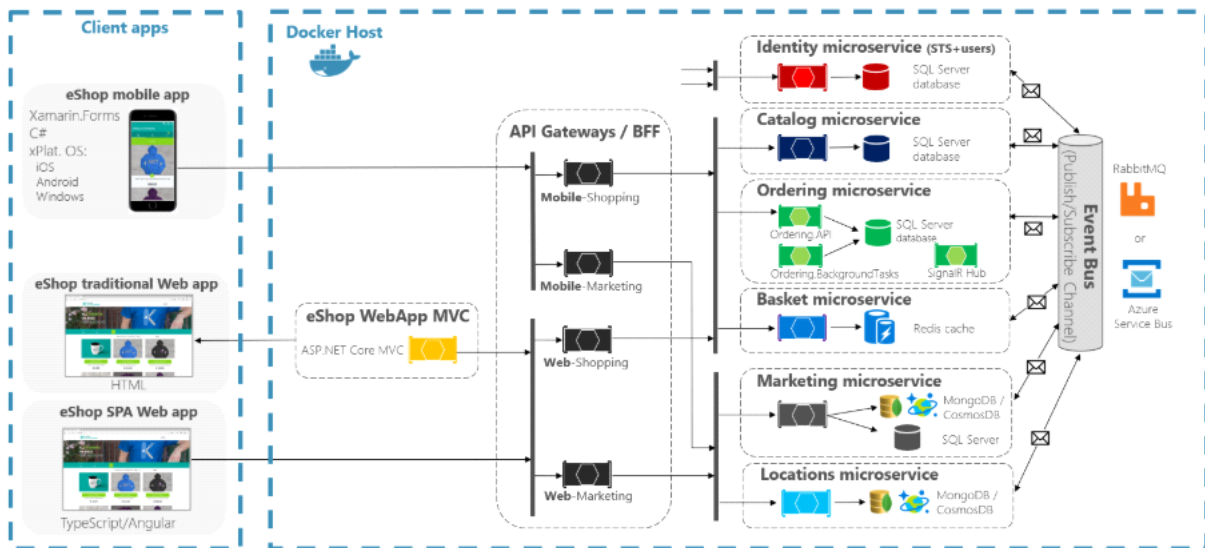
其它 API 网关框架

比如 Apigee、Kong、MuleSoft、WSO2，以及提供服务网格入口控制器功能的 Linkerd 和 Istio 等其他产品。

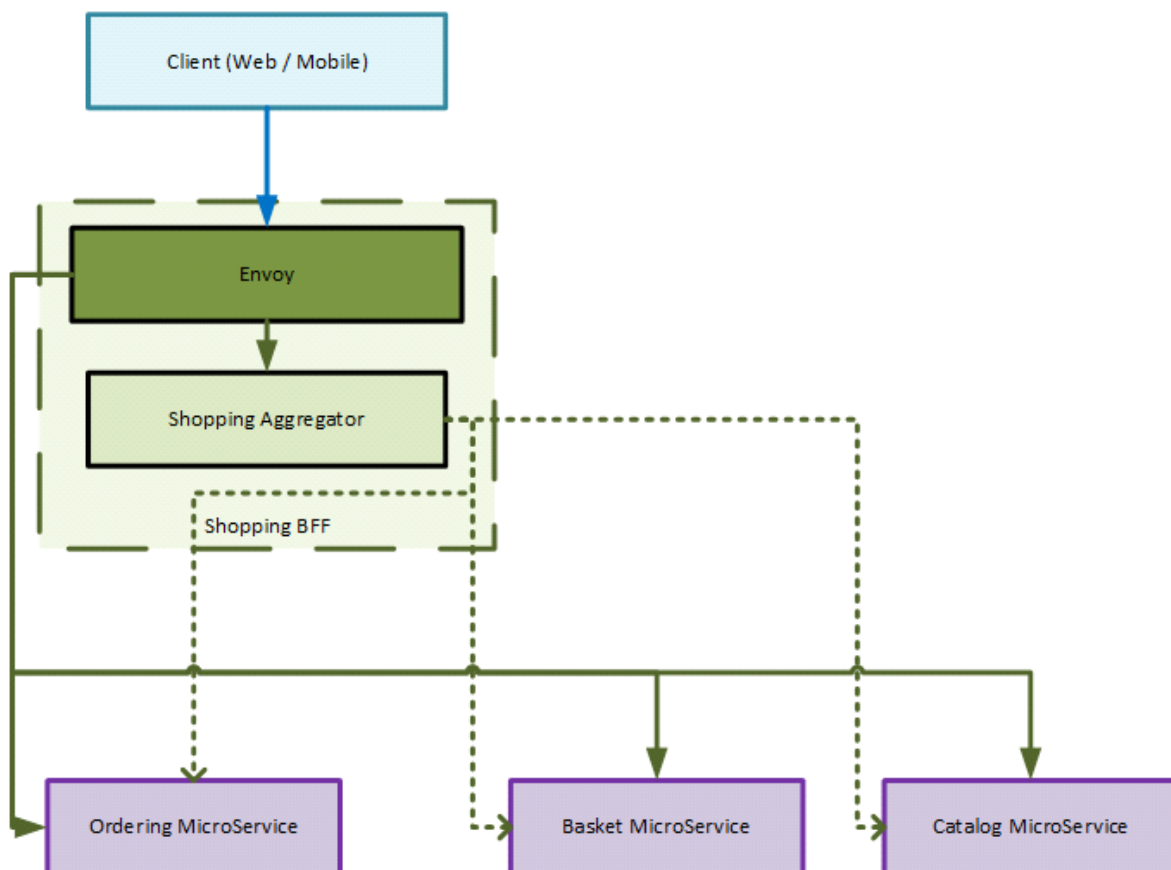
API 网关模式的缺点

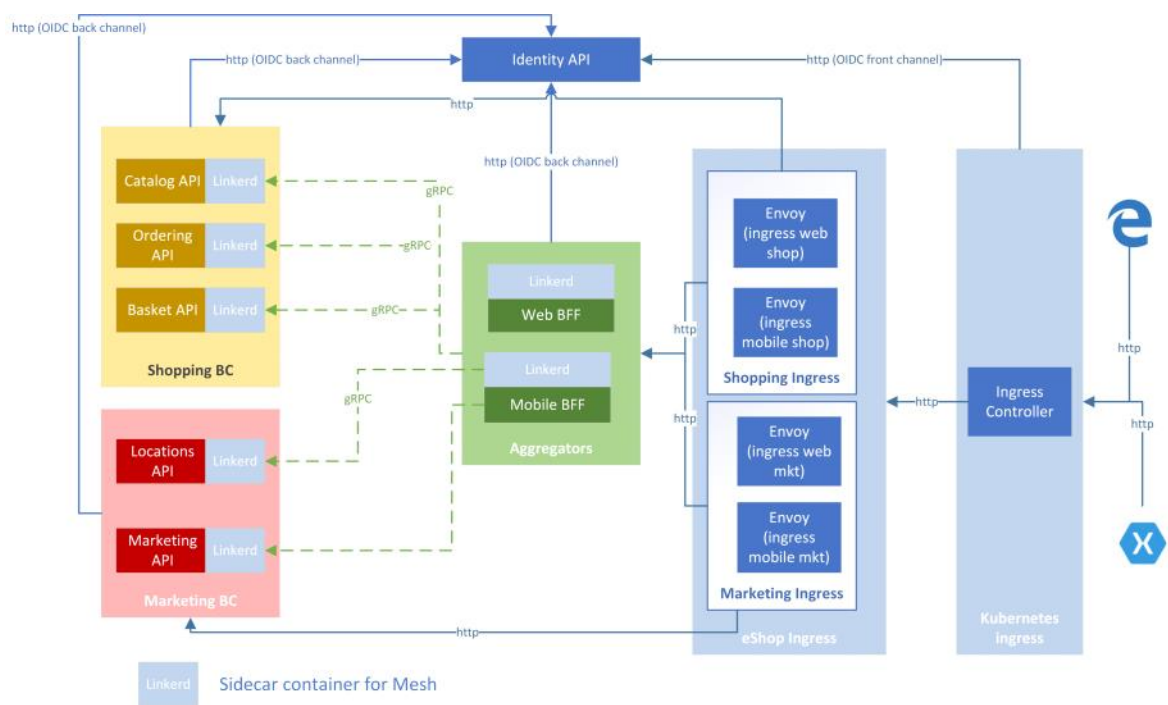
- 网关与微服务耦合，配置复杂性
- 网关单点故障导致其它微服务无法使用
- 网络调用，API 网关可能会导致响应时间加长，但可以忽略不记。
- 网关设计不当，无法正确扩展，成为性能瓶颈。
- 网关如果包含自定义逻辑和数据，需要单独人员来维护。

学习 eShopOnContainers 2.X 中的 API + BFF 网关模式



学习 eShopOnContainers 3.X 中的 API + BFF 网关模式





参考链接

[API 网关模式与客户端到微服务直接通信](#)

[通过 Ocelot 实现 API 网关](#)

[eShopOnContainers API 网关模式](#)

[eShopOnContainers BFF 基于前端的后端模式](#)