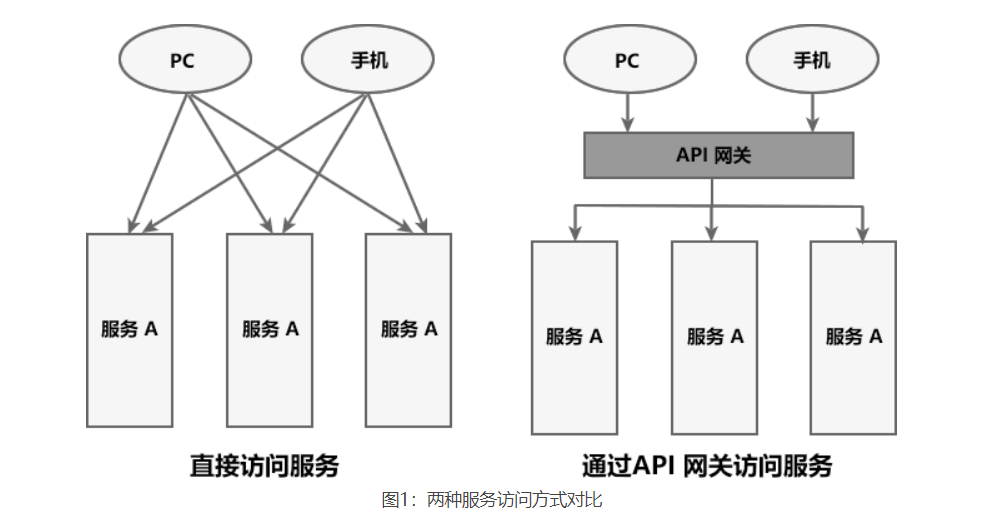
Api网关是搭建在客户端与服务端之间的服务，是微服务系统的门面，可以对外的唯一入口。



使用API的好处：

· 客户端与服务端交互时，不需要请求大量服务端地址，只需要请求网关地址即可。

· 客户端与网关交互，能够减少客户端与各个服务端请求的次数。

· 客户端与后端的耦合降低。

· 网关API提供安全、流控、过滤、缓存、验证、计费、监控等管理功能。

常见的API网关：

· Spring Cloud Gateway

· Spring Cloud Netflix Zuul

· Kong

· Nginx + Lua

· Traefik

**Spring Cloud Gateway**

基于Spring 5.0，Spring Boot 2.0和Project Reactor等技术开发的高性能API网关。

核心概念:

· Route（路由）：网关最基础模块。它由一个id，一个目标uri，一组断言(Predicate)和一组过滤器(Filter)组成。

· Predicate（断言）：路由转发的判断条件，如果请求与断言匹配成功，则将请求转发到相应服务。

· Filter（过滤器）：可以使用过滤器对请求进行拦截，也可以对响应进行处理。

Spring Cloud Gateway的特征：

· 基于Spring 5.0，Spring Boot 2.0和Project Reactor构建

· 能够在任意请求属性上匹配路由

· predicates（断言） 和 filters（过滤器）是特定于路由的

· 集成了Hystrix熔断器

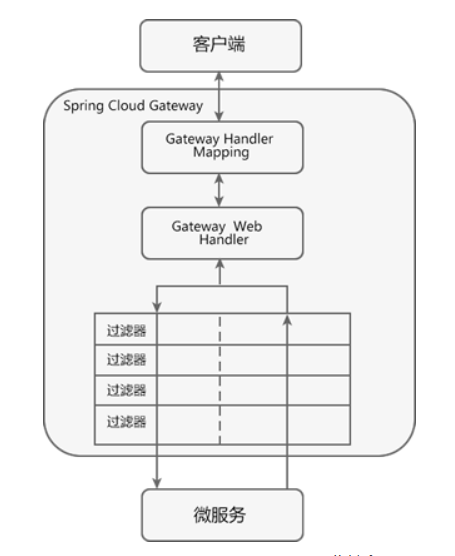
· 集成了Spring Cloud DiscoveryClient（服务发现客户端）

· 易于编写断言和过滤器

· 能够限制请求频率

· 能够重写请求路径

Gateway工作流程：



1. 客户端将请求发送到Spring Cloud Gateway上
2. Spring Cloud Gateway 通过Gateway Handler Mapping找到与请求相匹配的路由，将其发送给Gateway Web Handler
3. Gateway Web Handler 通过指定的过滤器链（Filter Chain），将请求转发到实际的服务节点中，执行业务逻辑返回响应结果
4. 过滤器之间用虚线分开是因为过滤器可能会在转发请求之前（pre）或之后（post）执行业务逻辑
5. 过滤器（Filter）可以在请求被转发到服务端前，对请求进行拦截和修改，例如参数校验、权限校验、流量监控、日志输出以及协议转换等
6. 过滤器可以在响应返回客户端之前，对响应进行拦截和再处理，例如修改响应内容或响应头、日志输出、流量监控等
7. 响应原路返回给客户端

Predicate 断言：

Gateway通过Predicate 实现Route路由的匹配规则。

使用注意：

· Route路由与Predicate 断言的对应关系是一对多，一个路由可以包含多个断言

· 一个请求想要转发到指定的路由，就必须同时匹配多个断言

· 当一个请求同时满足了多个路由条件时，请求只会被首个成功匹配的路由转发



Spring Cloud Gateway 动态路由

会根据注册中心的服务列表，以服务名(spring.application.name)作为路径创建动态路由进行转发，从而实现动态路由功能。



Filter过滤器

微服务的过滤校验，写在gateway的filter中。

使用分类：

· Pre类型：在转发到微服务之前的过滤，用于验证、校验、流量监控等

· Post类型：在微服务对请求做出响应之后过滤，可以进行拦截、再处理等

按照范围分类：

· GatewayFilter：应用在单个路由或一组路由上的过滤

· GlobalFilter：应用在所有的路由器上的过滤



