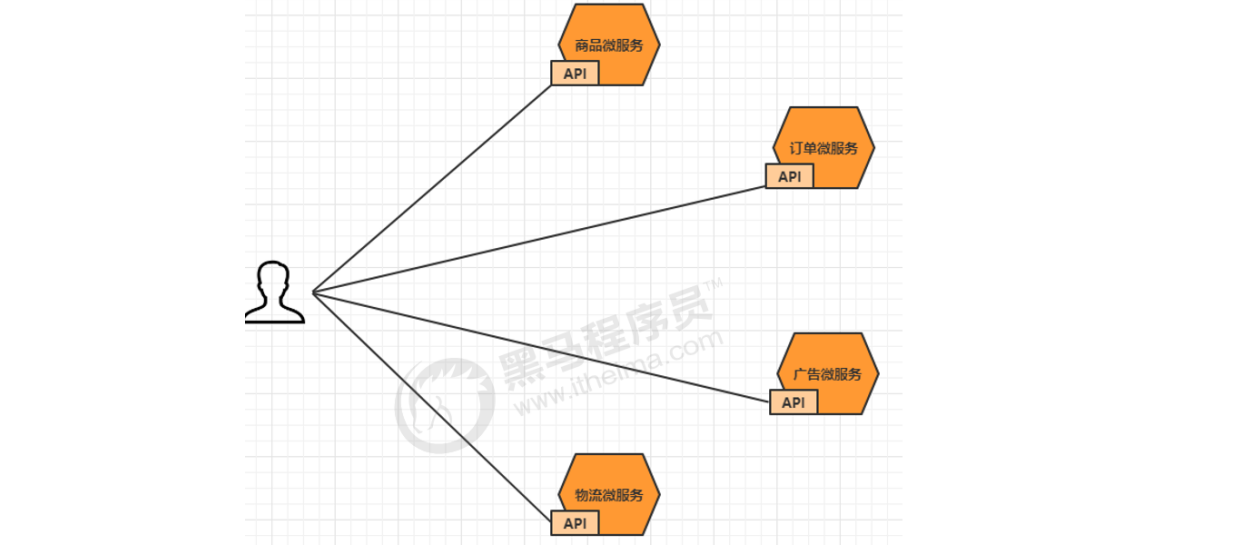
## **网关简介**

大家都都知道在微服务架构中，一个系统会被拆分为很多个微服务。那么作为客户端要如何去调用 这么多的微服务呢？如果没有网关的存在，我们只能在客户端记录每个微服务的地址，然后分别去调 用。



这样的架构，会存在着诸多的问题：

客户端多次请求不同的微服务，增加客户端代码或配置编写的复杂性

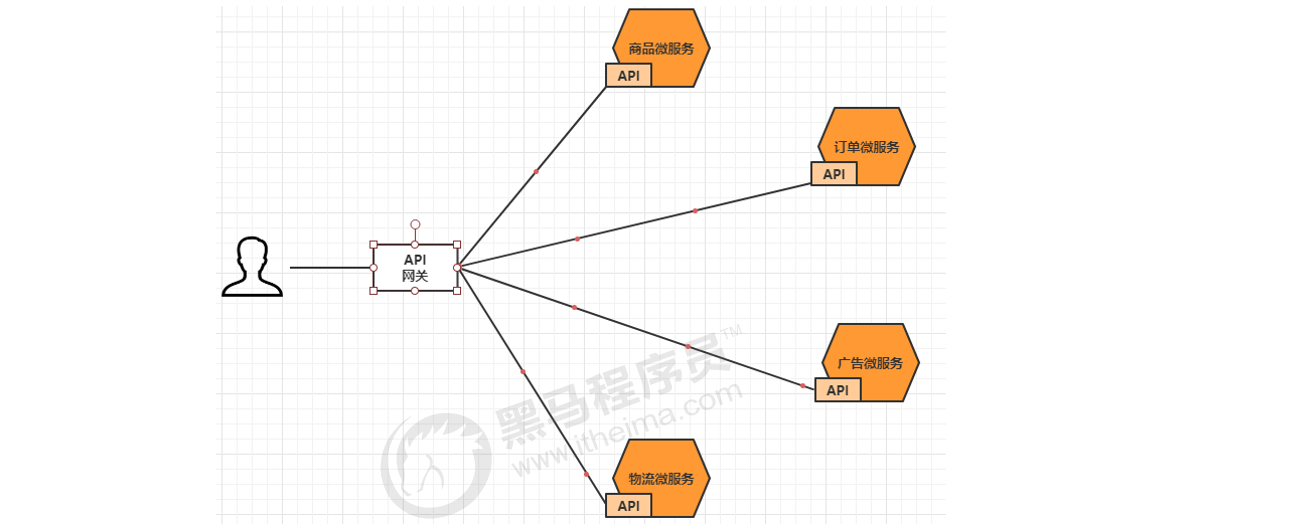
认证复杂，每个服务都需要独立认证。

存在跨域请求，在一定场景下处理相对复杂。

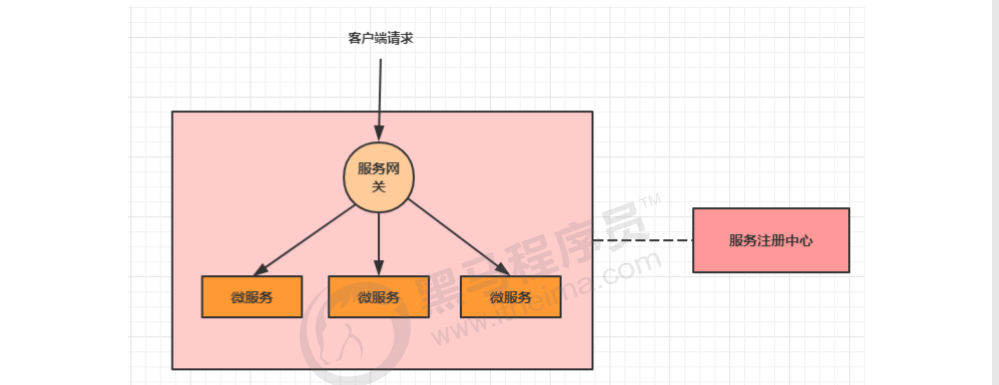
上面的这些问题可以借助**API网关**来解决。

所谓的API网关，就是指系统的**统一入口**，它封装了应用程序的内部结构，为客户端提供统一服 务，一些与业务本身功能无关的公共逻辑可以在这里实现，诸如认证、鉴权、监控、路由转发等等。

添加上API网关之后，系统的架构图变成了如下所示：



我们也可以观察下，我们现在的整体架构图：



在业界比较流行的网关，有下面这些：

**Ngnix+lua**

使用nginx的反向代理和负载均衡可实现对api服务器的负载均衡及高可用

lua是一种脚本语言,可以来编写一些简单的逻辑, nginx支持lua脚本

**Kong**

基于Nginx+Lua开发，性能高，稳定，有多个可用的插件(限流、鉴权等等)可以开箱即用。 问题： 只支持Http协议；二次开发，自由扩展困难；提供管理API，缺乏更易用的管控、配置方式。

**Zuul Netﬂix**

开源的网关，功能丰富，使用JAVA开发，易于二次开发 问题：缺乏管控，无法动态配 置；依赖组件较多；处理Http请求依赖的是Web容器，性能不如Nginx

**Spring Cloud Gateway**

Spring公司为了替换Zuul而开发的网关服务，将在下面具体介绍。

**注意：SpringCloud alibaba技术栈中并没有提供自己的网关，我们可以采用Spring Cloud Gateway 来做网关**

## Gateway简介

Spring Cloud Gateway是Spring公司基于Spring 5.0，Spring Boot 2.0 和 Project Reactor 等技术 开发的网关，它旨在为微服务架构提供一种简单有效的统一的 API 路由管理方式。它的目标是替代 Netﬂix Zuul，其不仅提供统一的路由方式，并且基于 Filter 链的方式提供了网关基本的功能，例如：安 全，监控和限流。

**优点：**

性能强劲：是第一代网关Zuul的1.6倍

功能强大：内置了很多实用的功能，例如转发、监控、限流等

设计优雅，容易扩展

**缺点：**

其实现依赖Netty与WebFlux，不是传统的Servlet编程模型，学习成本高

不能将其部署在Tomcat、Jetty等Servlet容器里，只能打成jar包执行

需要Spring Boot 2.0及以上的版本，才支持