



Workshop Kong

Luiz Fernando de Jesus
Portela

Gustavo Hernandes Furtado
de Avelar

Introdução



- API Gateway é um serviço, dispositivo ou proxy que atua como um intermediário que aceita, transforma, encaminha e gerencia o tráfego de APIs para serviços de back-end.
- Fornece um único ponto de entrada para a aplicação.
- Disponibiliza camadas de proteção entre requests e responses ao servidor.
- Permite o uso de técnicas como balanceamento de carga, proxy reverso, autenticação, limite de requisições, etc.

Objetivos



- Trazer segurança, performance e estabilidade para aplicações, gerenciamento, transformando requisições e respostas ao servidores.
- Camada de proteção extra aos serviços internos
- Prevenir ataques de força bruta, requisições não autenticadas, sobrecarga de pods/servidores.
- Logging, Cache, Tradução de protocolos, Rastreabilidade aumentada.

Problema



- Uma grande quantidade de chamadas abusivas a um endpoint.
- Necessidade de múltiplos pontos de entradas com endereços diferentes.
- Segurança entre aplicações, duplicidade de autenticação entre micro-serviços.

Mercado



Empresas como: Netflix, X, Spotify, Airbnb, Amazon e muitas outras, se beneficiam de Api Gateways para fazer balanceamento de carga, limitar requisições e fazer toda a transformação delas.

Ferramentas



API Gateways podem ser implementados de forma manual, pois são aplicações que gerenciam as requisições, respostas, autenticação, etc. Do seu serviço ou servidor. Diante disso, para facilitar este processo, existem diversas ferramentas disponíveis no mercado, tornando a disposição e implantação de um API Gateway muito mais rápido.

Balanceamento de carga estáticos



01

Round Robin

Algoritmo estático bem simples. Sua decisão consiste em atribuir as requisições em sequência com base no grupo de servidores disponíveis.

02

Weigthed Round Robin

Variação do Round Robin, porém os servidores recebem pesos para estabelecer uma proporção no número de requisições que recebem. Desta forma, em um ambiente com servidores não homogêneos, é possível direcionar mais carga para aqueles com maior capacidade computacional.

03

IP Hash

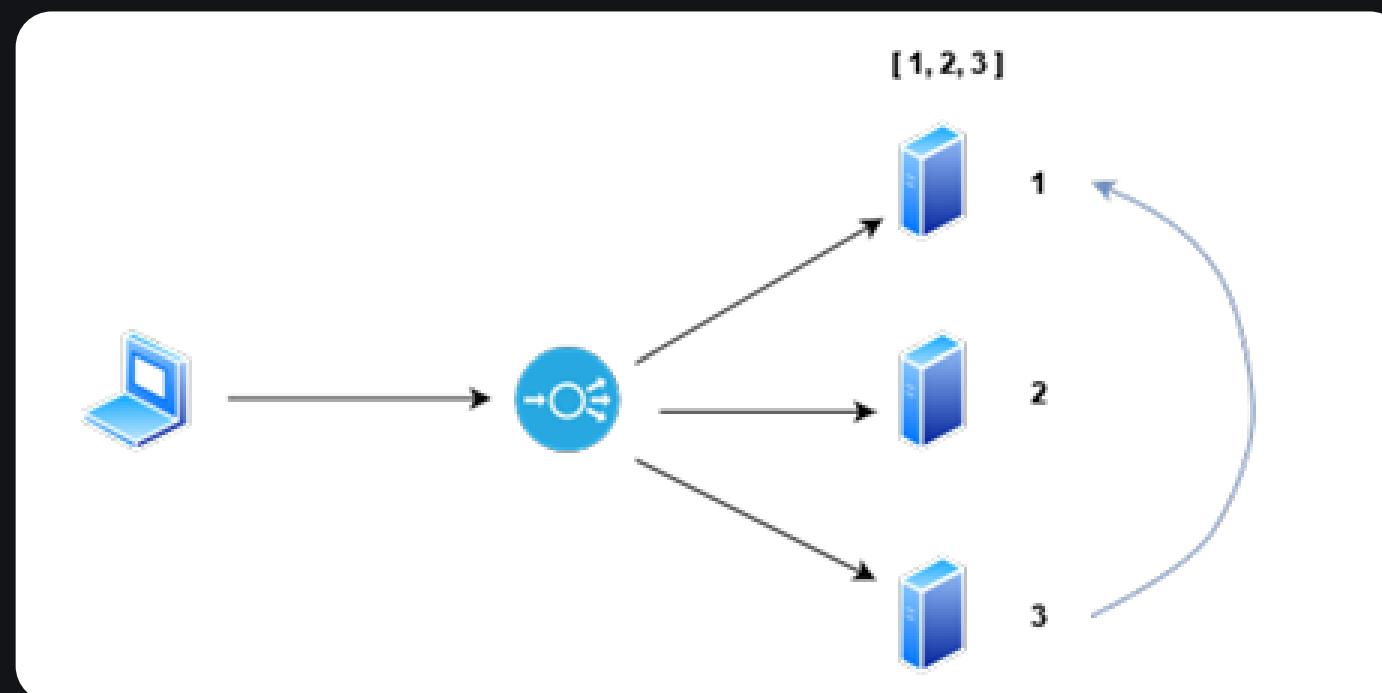
O objetivo deste algoritmo é manter as requisições de um cliente direcionadas sempre para o mesmo servidor. Isso é feito através de um hash calculado com o endereço IP do cliente. Caso o servidor mapeado para um determinado IP se encontrar indisponível, a requisição é direcionada para o seguinte, seguindo a ordem na lista do grupo.

Balanceamento de carga estáticos

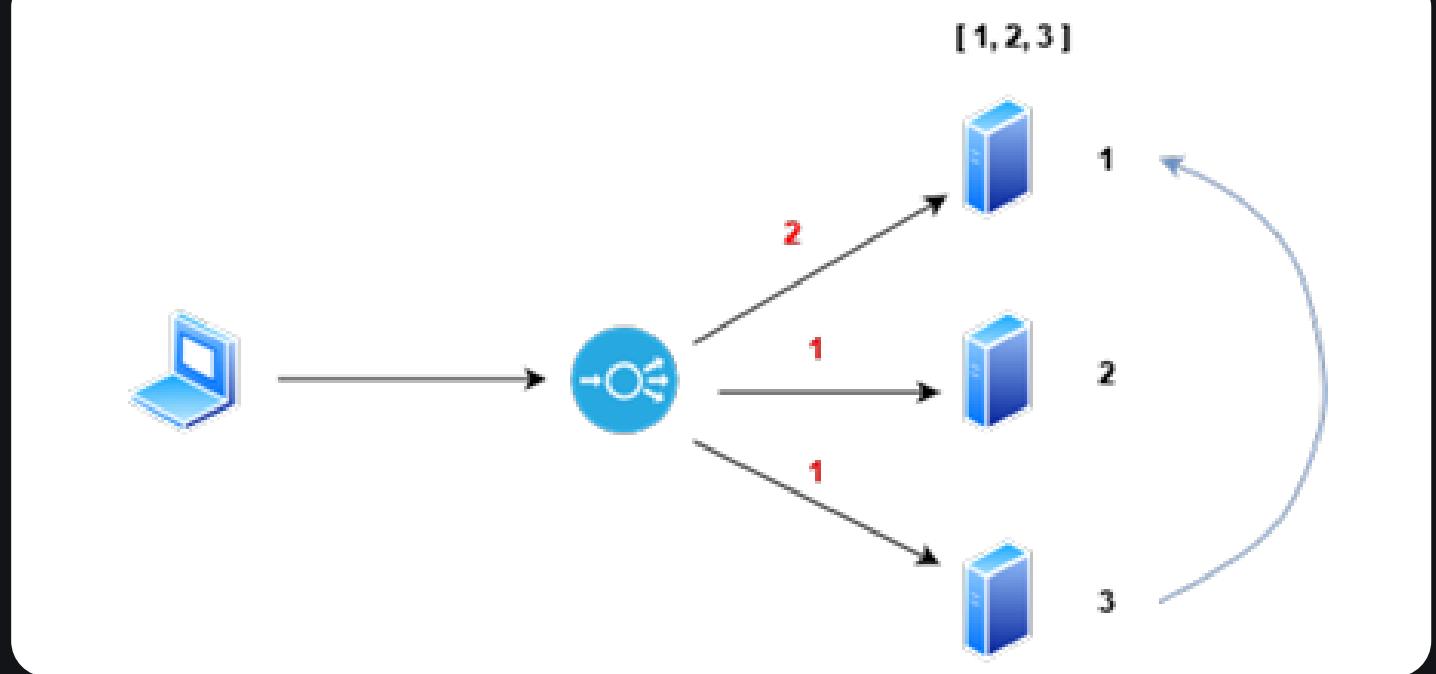
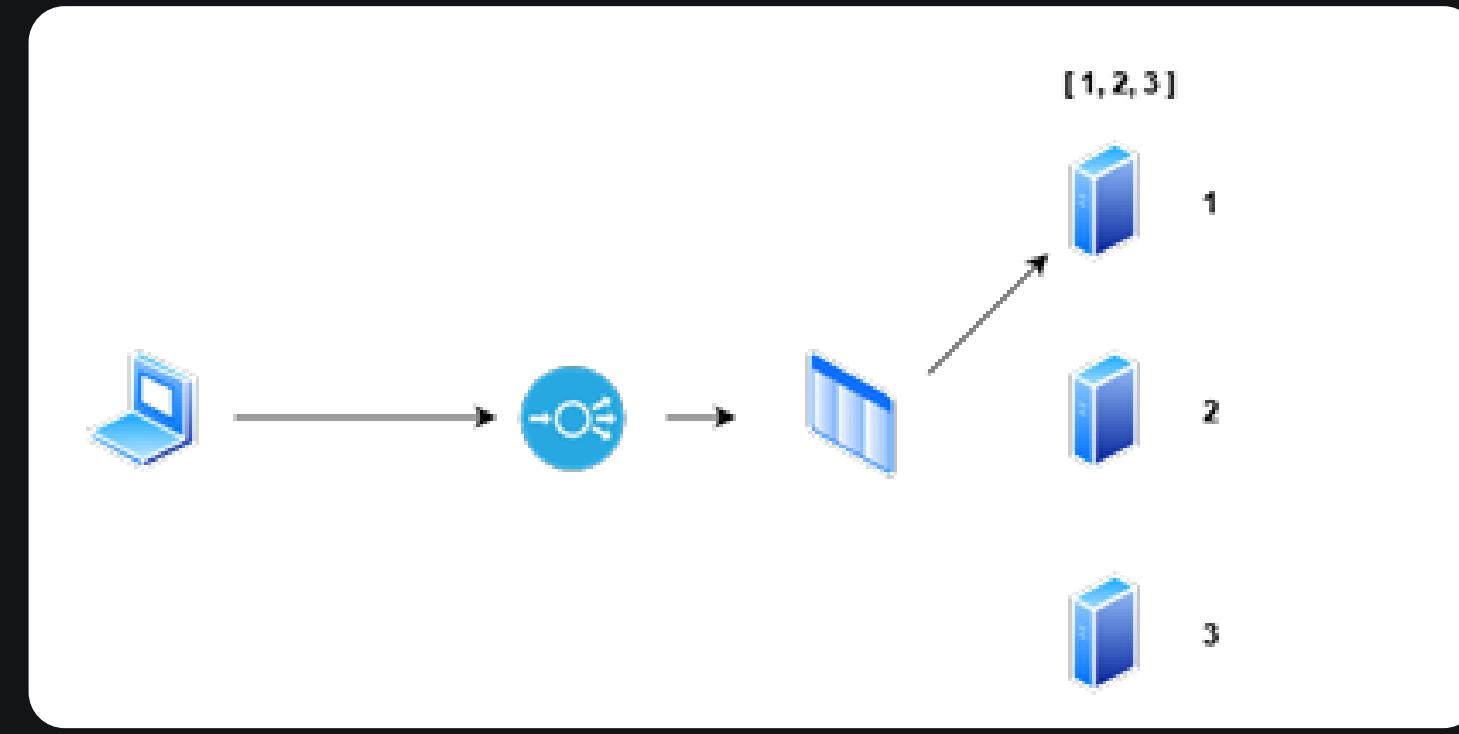
Weigthed
Round Robin



Round Robin



IP Hash



Balanceamento de carga dinâmicos

01

Least Connection

Consiste em consultar as informações sobre o número de conexões ativas em cada servidor disponível. Dessa forma, o balanceador de carga consegue distribuir a próxima requisição para o servidor com a menor carga naquele momento.

02

Weighted Least Connection

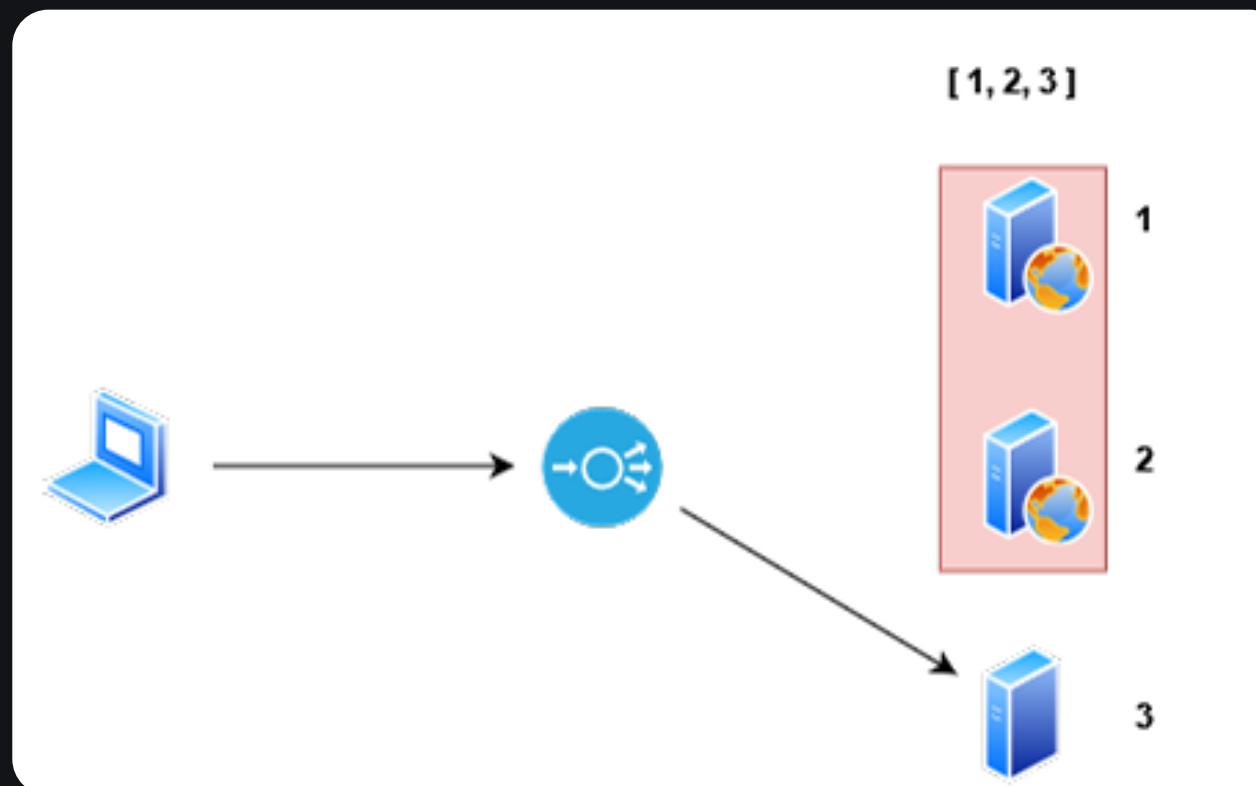
Dispõe de uma configuração inicial, a qual estabelece o quanto cada servidor disponível é proporcionalmente mais capaz de receber requisições. Dessa forma não é escolhido o servidor com o menor número de conexões ativas no momento, necessariamente, mas aquele o qual possui menor valor em proporção. O cálculo desta proporção é a razão entre o número de conexões ativas e o peso atribuído.



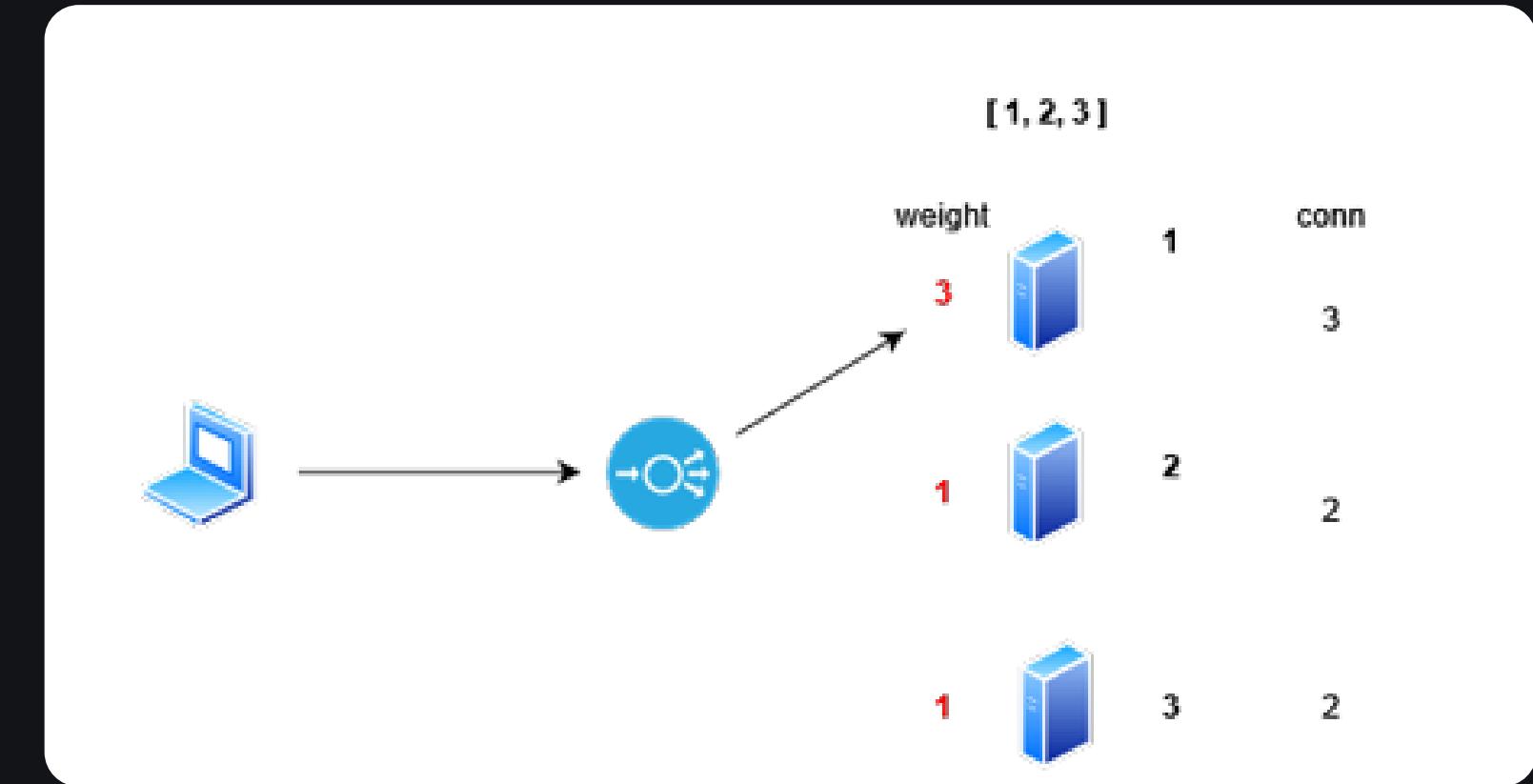


Balanceamento de carga estáticos

Least Connection



Weighted Least Connection



Fonte: <https://www.ic.unicamp.br/~reltech/PFG/2021/PFG-21-49.pdf>

NGINX



O predecessor e mais utilizado por debaixo dos panos, o NGINX é um super servidor, que além de gerenciar, também pode ser utilizado como API Gateway, lidando com balanceamento de carga, proxy reverso, rate limiting, e outras diversas funcionalidades.



Altamente configurável, o que pode ser uma vantagem e desvantagem ao mesmo tempo, pois suas configurações devem ser feitas de forma manual através de arquivos conf, muito mais trabalhoso e manual.

Porém é uma ótima ferramenta para quem quer ter total controle do que está sendo feito, e quer entender o que está acontecendo por de trás de grandes aplicações e servidores, levando em consideração que boa parte da Web está construída em cima dele.

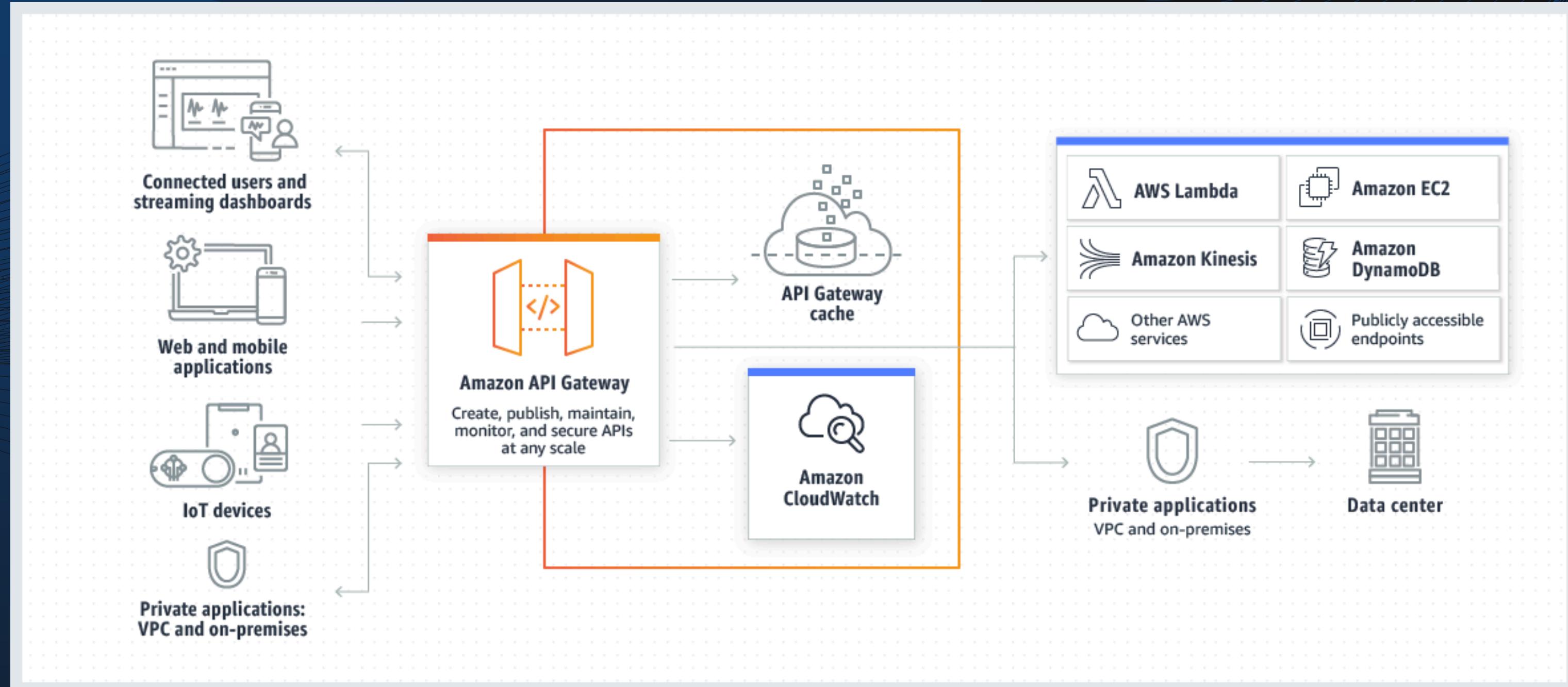
AWS API Gateway & GCP API Gateway



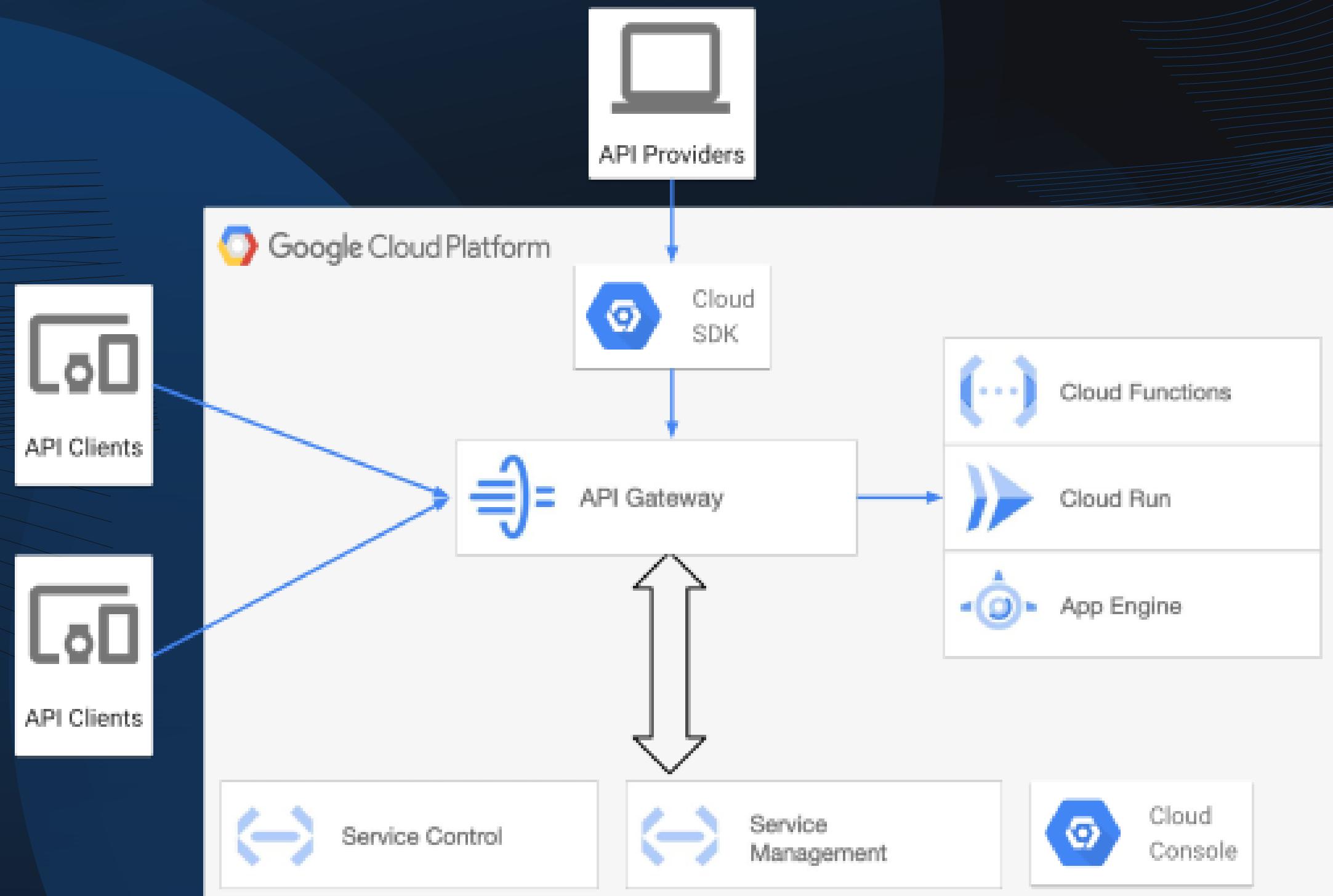
São serviços fornecido pela Amazon AWS e Google Cloud Platform, contendo funcionalidades como:

- Caching
- Rate Limiting
- Authentication
- Throttling
- Monitoramento e suporte a diferentes protocolos, como REST e WebSockets
- Canary Releases.

Arquitetura AWS



Arquitetura GCP



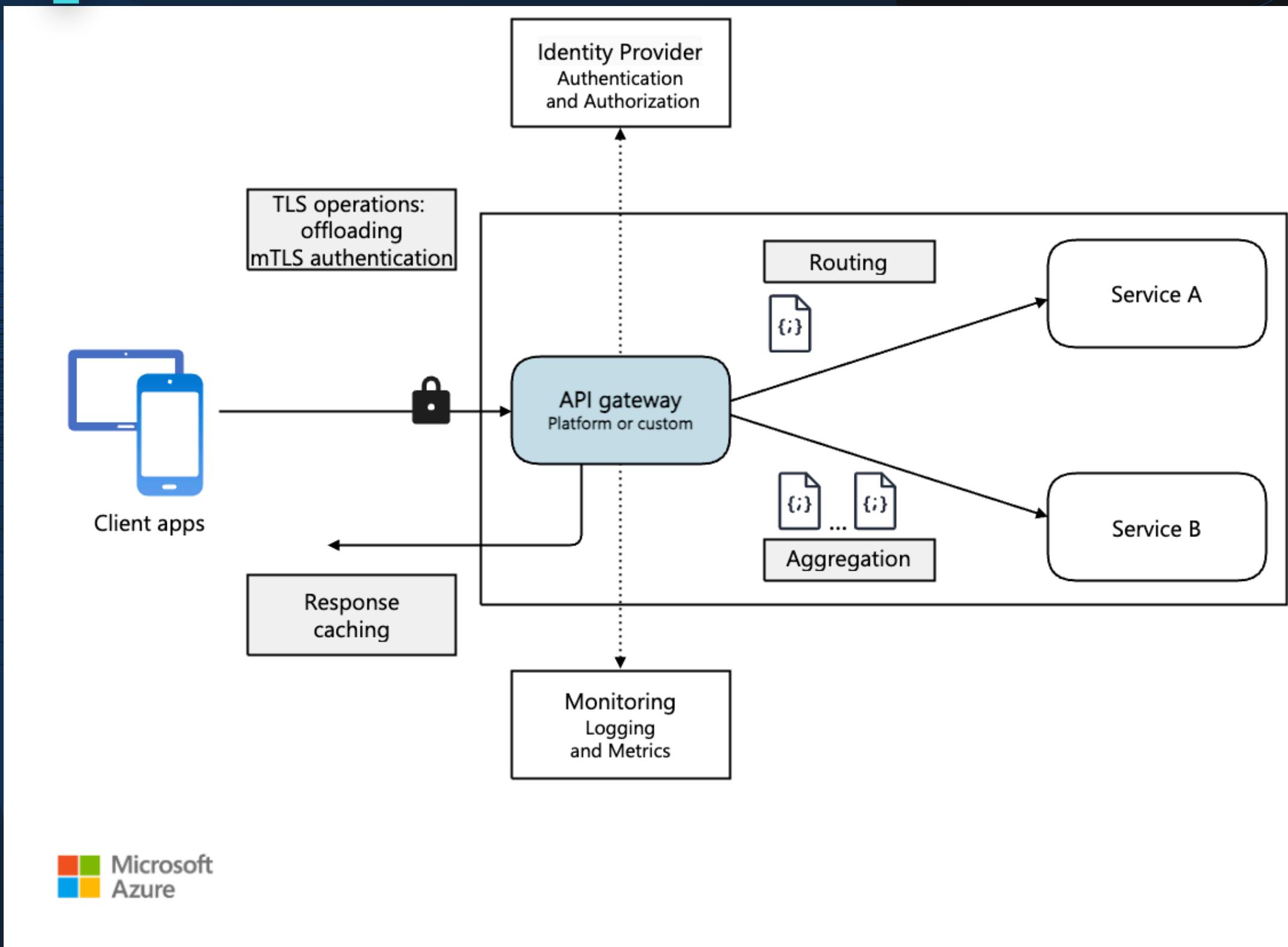
Azure API Gateway

Serviço de API Gateway fornecido pela Microsoft Azure com funcionalidades como:

- SSL termination
- Mutual TLS
- Authentication
- IP allowlist or blocklist
- Client rate limiting (throttling)
- Logging and monitoring
- Response caching
- Web application firewall
- GZIP compression
- Servicing static content



Arquitetura Azure



E o Kong?



Kong API Gateway



- Open Source
- UI Integrada
- Altamente extensível
- Plugins criados pela comunidade
- Plugins pagos
- Enterprise API Gateway
- Já falei Open Source?
- Autenticação
- Proxy Reverso
- Rate Limiting
- Caching
- Request and Response Transformation
- Utiliza LUA para plugins



Kong API Gateway

Criado em cima do NGINX e OpenResty o Kong API Gateway da Kong, que hoje também está por trás do Insomnia. É um API Gateway completo e robusto, de fácil implementação e implantação. Altamente escalável para Microgateways, leve e rápido.

Ele é perfeito para projetos pessoais, enterprises, self-hosted e para quem quer ter controle sobre sua aplicação e seu gateway.



Vamos praticar?!

