



# API Gateway with Kong



# Agenda

- Why API Gateway ?
- What is API Gateway ?
- Architecture
- Topologies and Tools
- Workshop

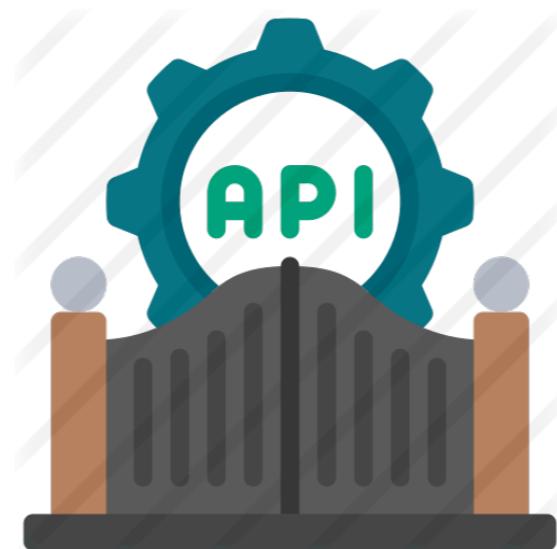


1. Problem

2. Solution

3. Technical solution

4. Improvement



# Why API Gateway ?

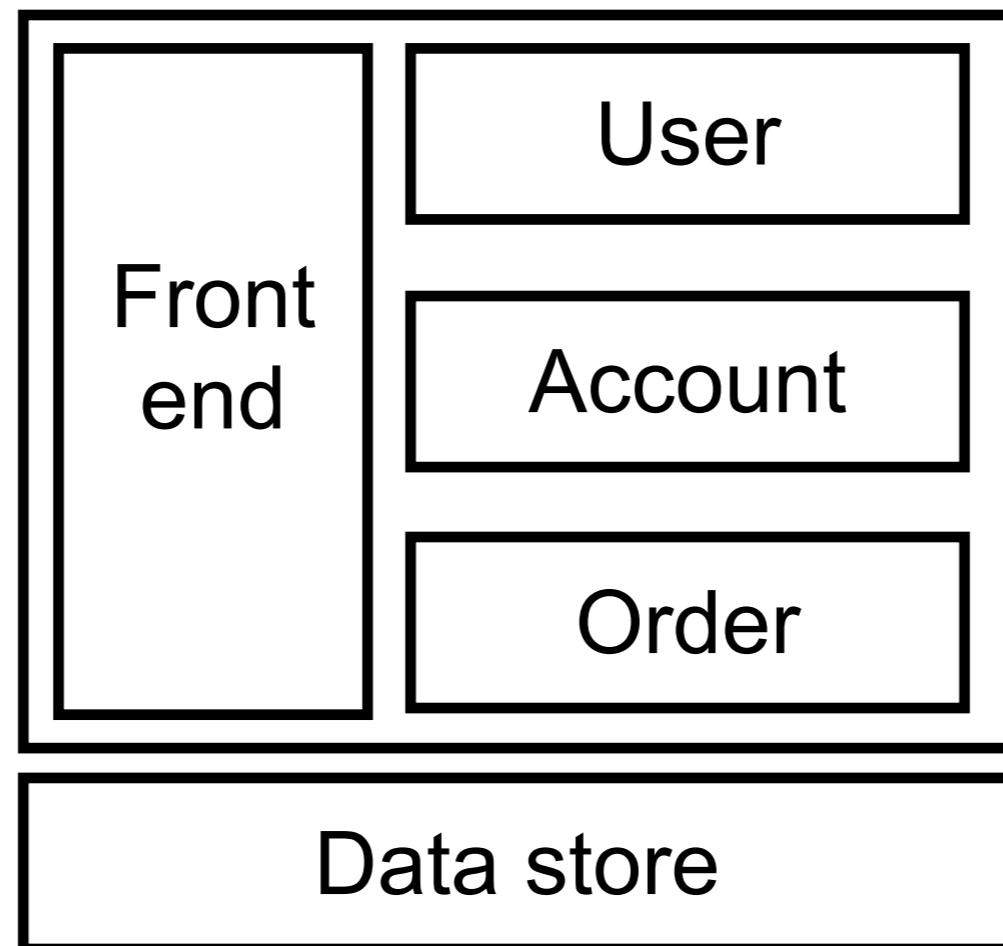


# Monolithic

All business logic in one  
One data store  
Horizontal scale



# Monolithic



# Pros of Monolithic

Simple architecture  
Easy to manage



# Cons of Monolithic

Hard to modify  
Heavy deployment



# Microservices

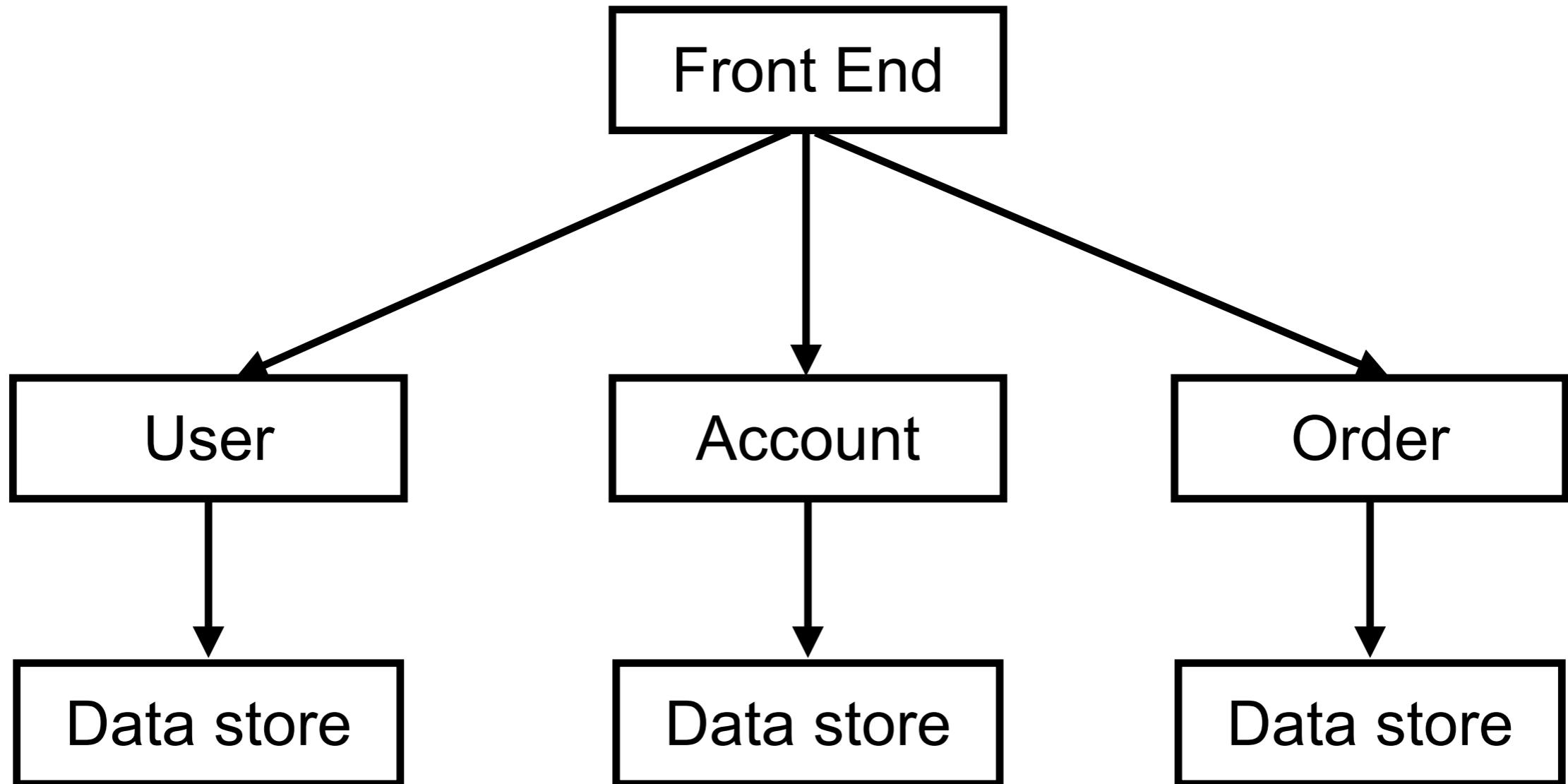
Separate business logic

Each data store per service

Vertical scaling



# Microservices



# Pros of Microservices

Easy to develop service  
Lightweight deployment



# Cons of Microservices

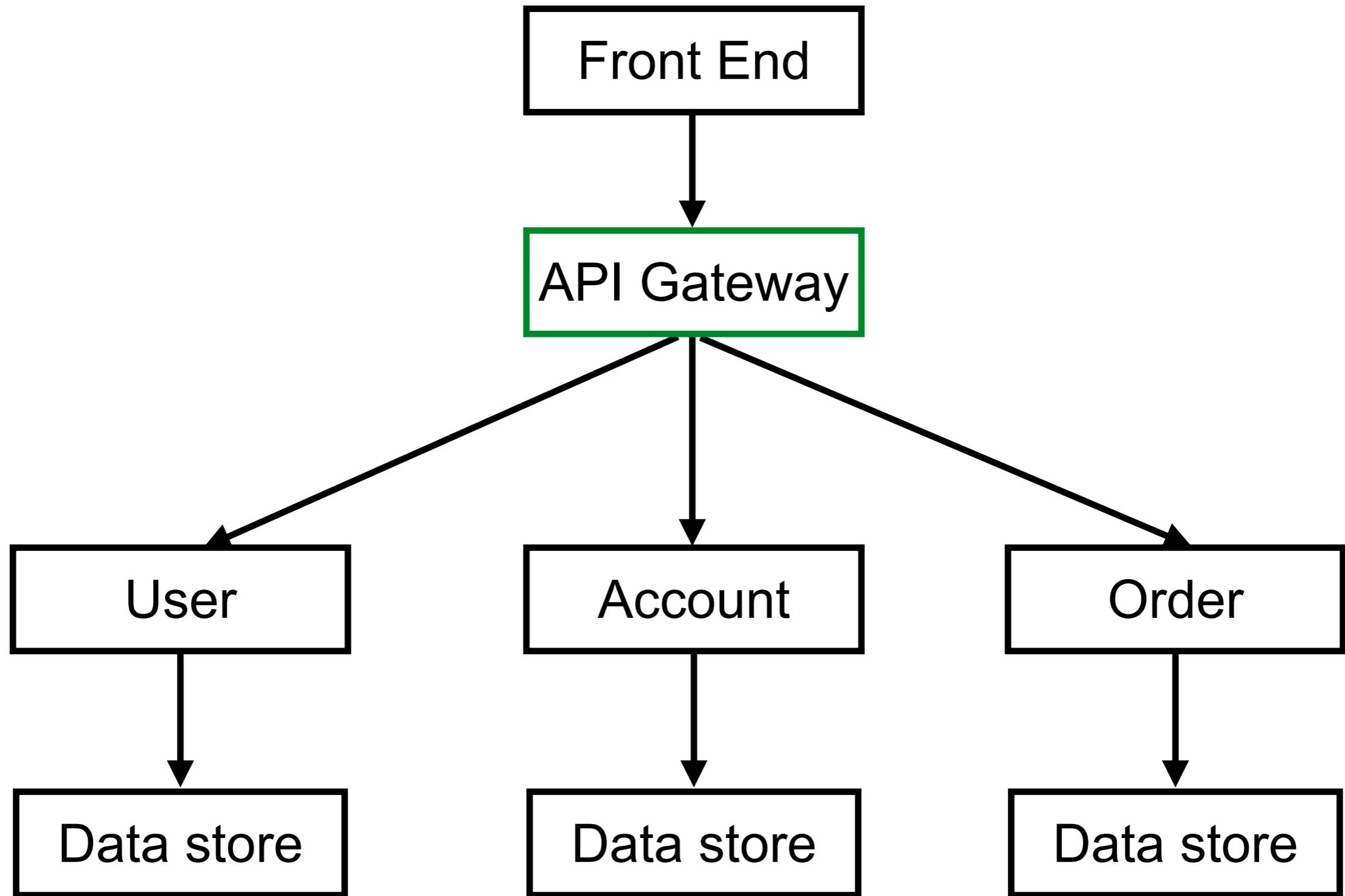
Hard to test and debug  
Many manage points



# What API Gateway ?



# Microservices and API Gateway

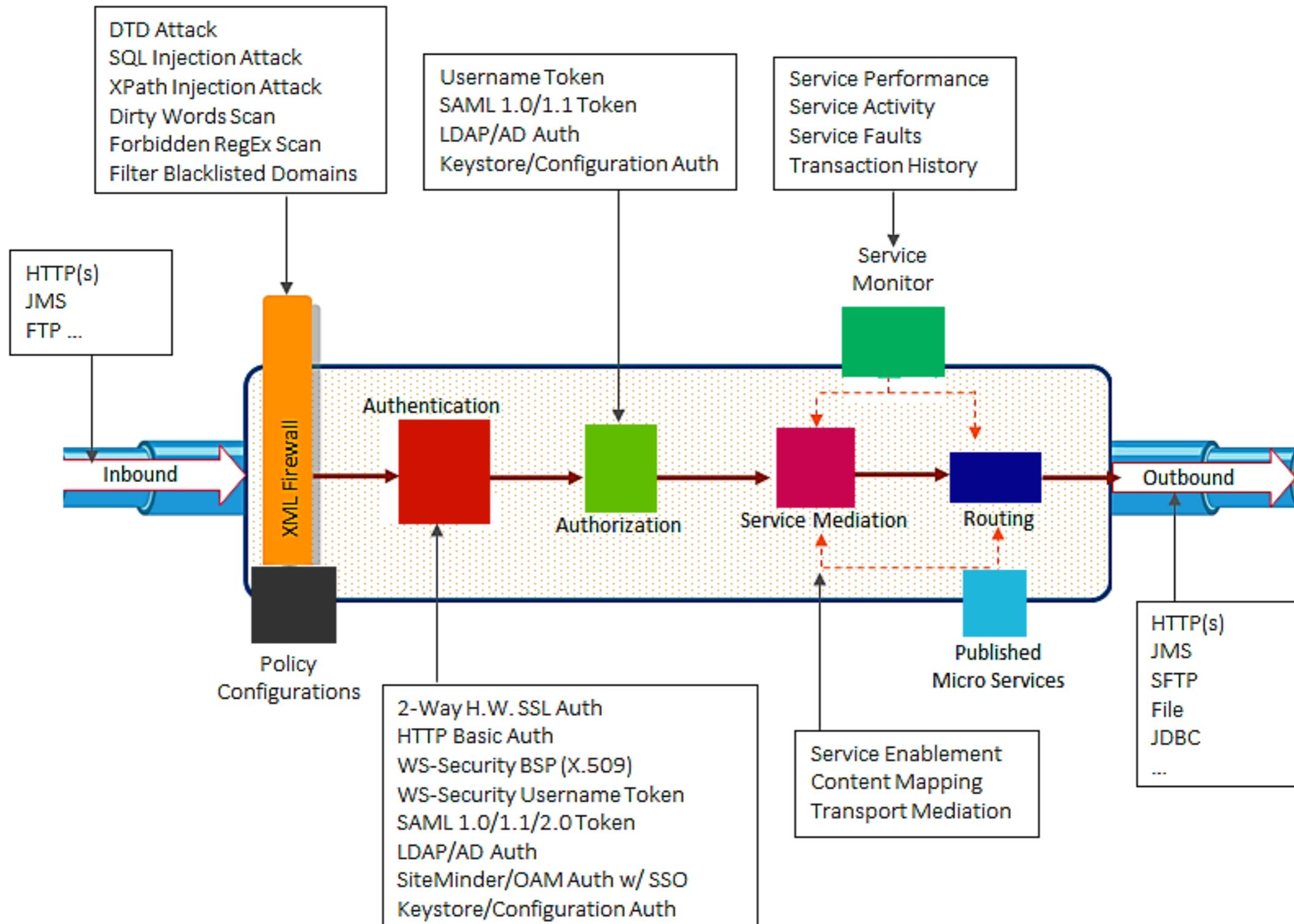


# What API Gateway ?

**Single entry point for API**  
**Middleware for API**  
Authentication  
Security  
Traffic management  
Logging  
Caching  
Load balancing



# API Gateway

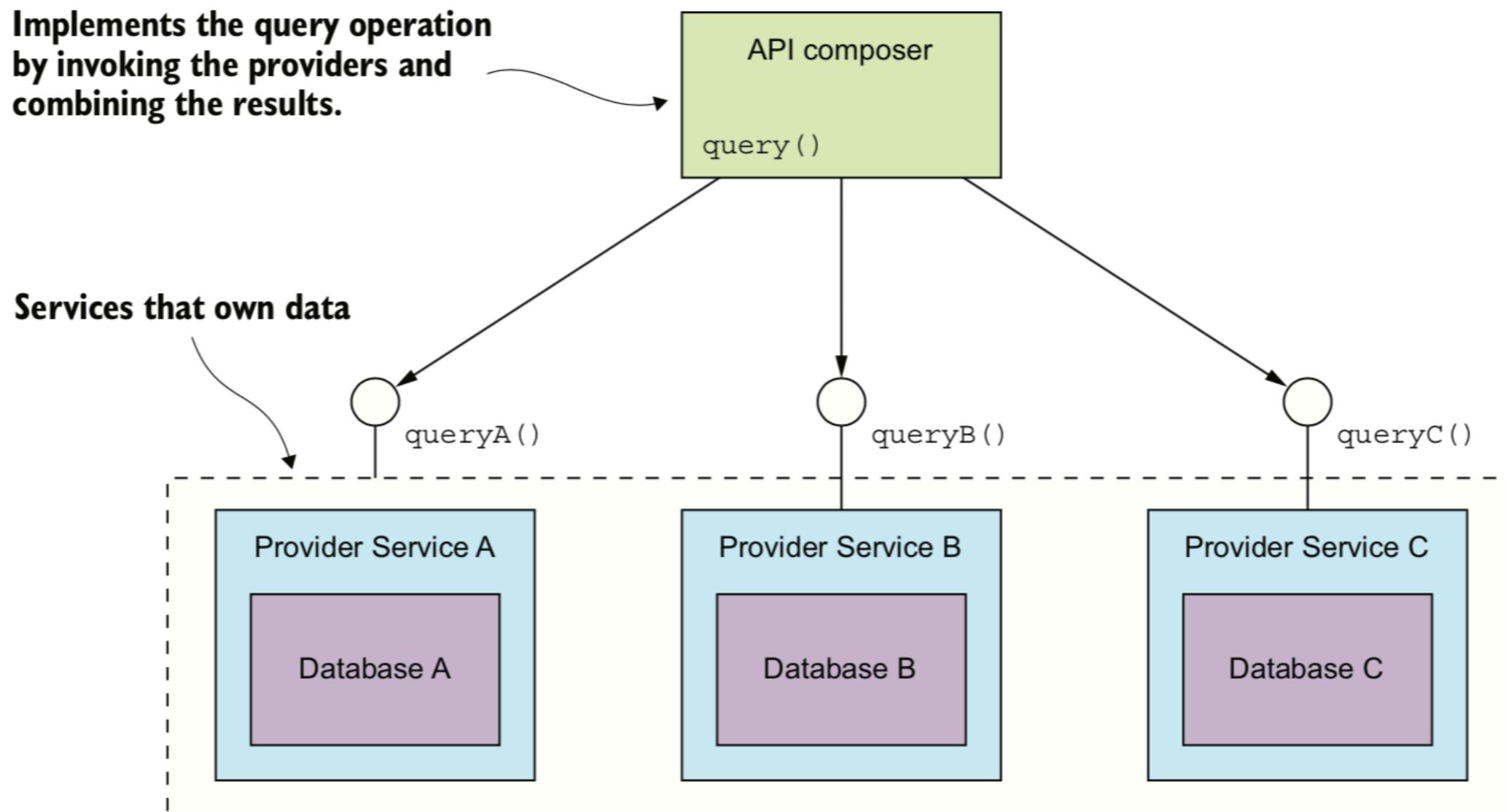


# Responsibility of API Gateway

Request routing  
Composition/aggregation  
Protocol translator



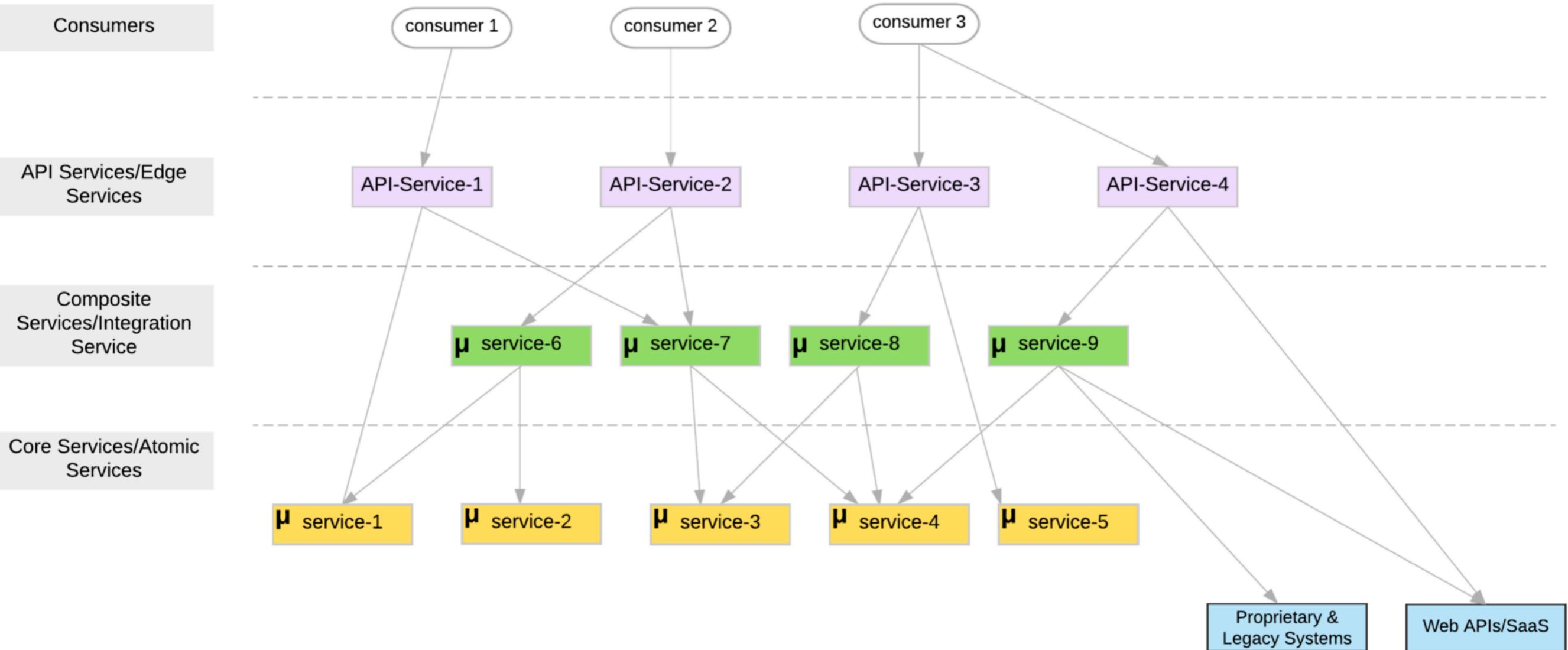
# API Composition



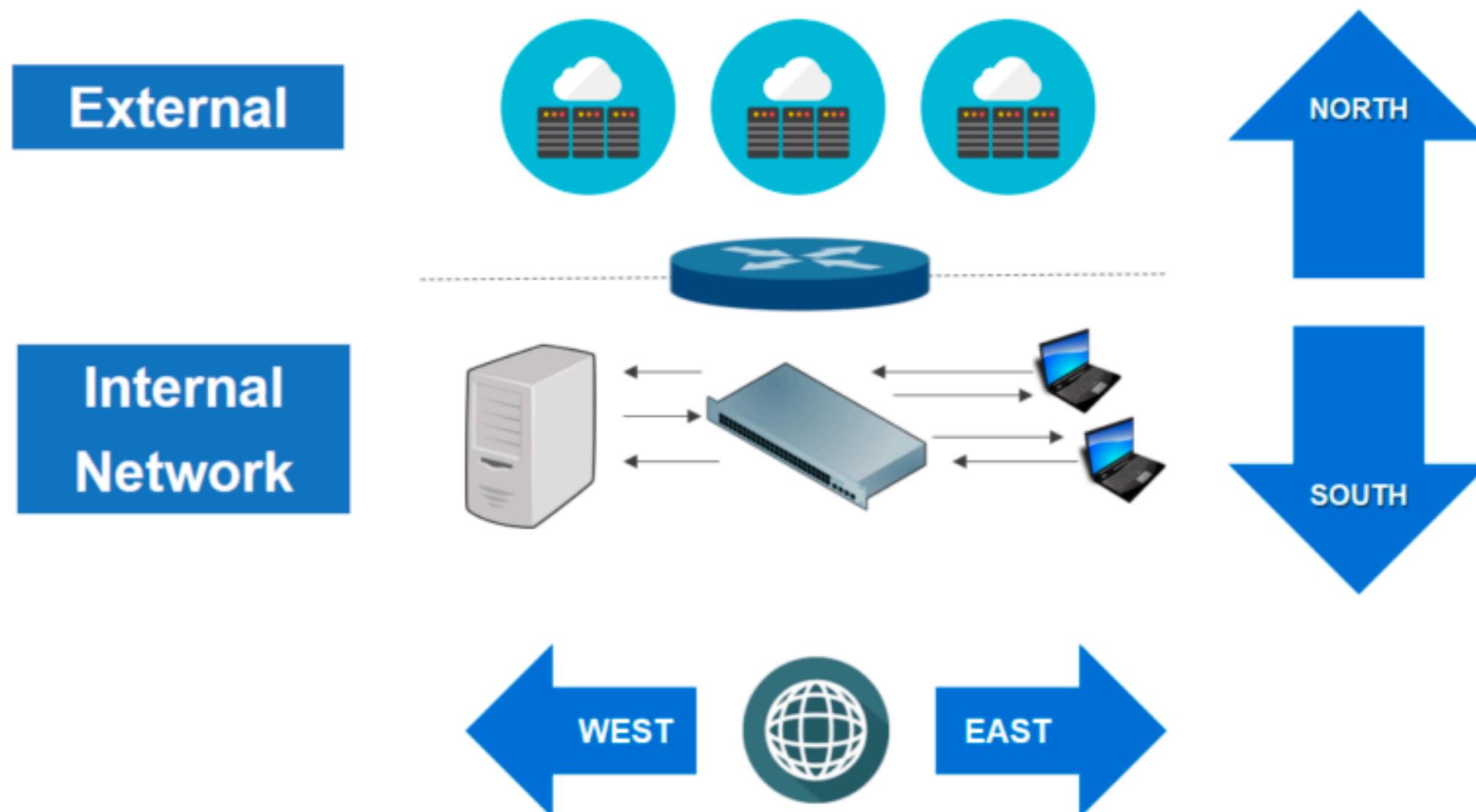
**Figure 7.2** The API composition pattern consists of an API composer and two or more provider services. The API composer implements a query by querying the providers and combining the results.



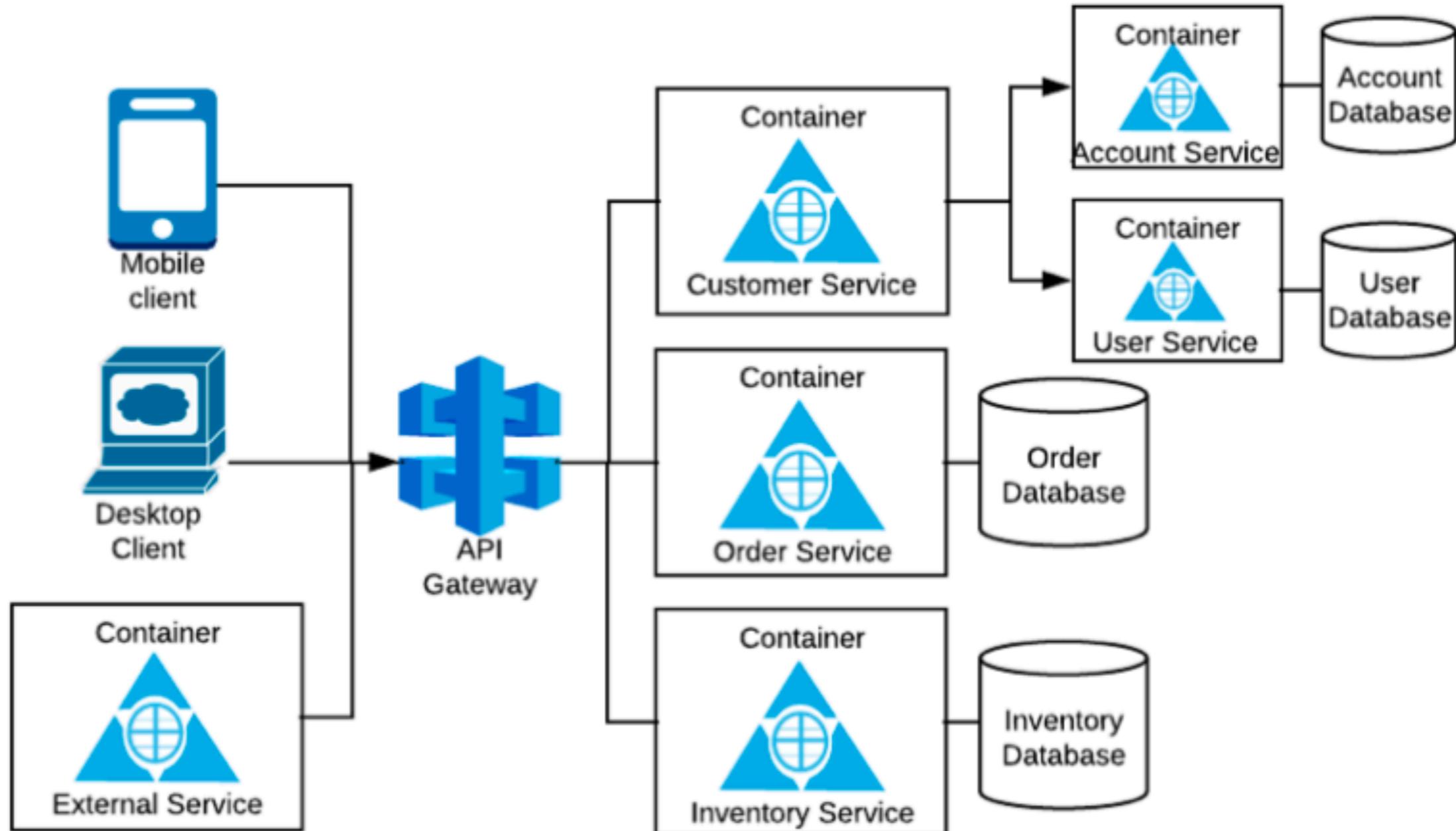
# API Gateway and Composition



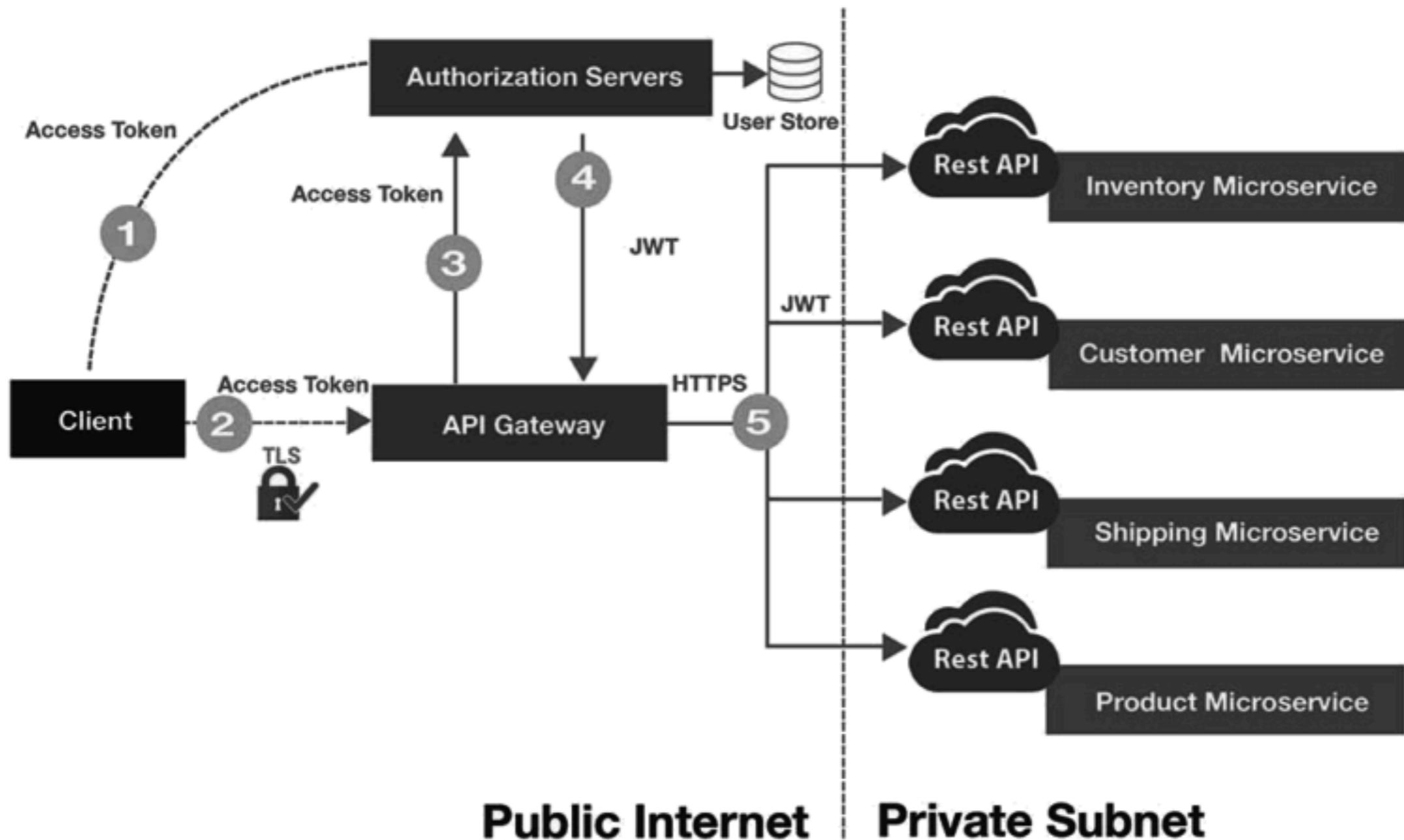
# North-South traffic



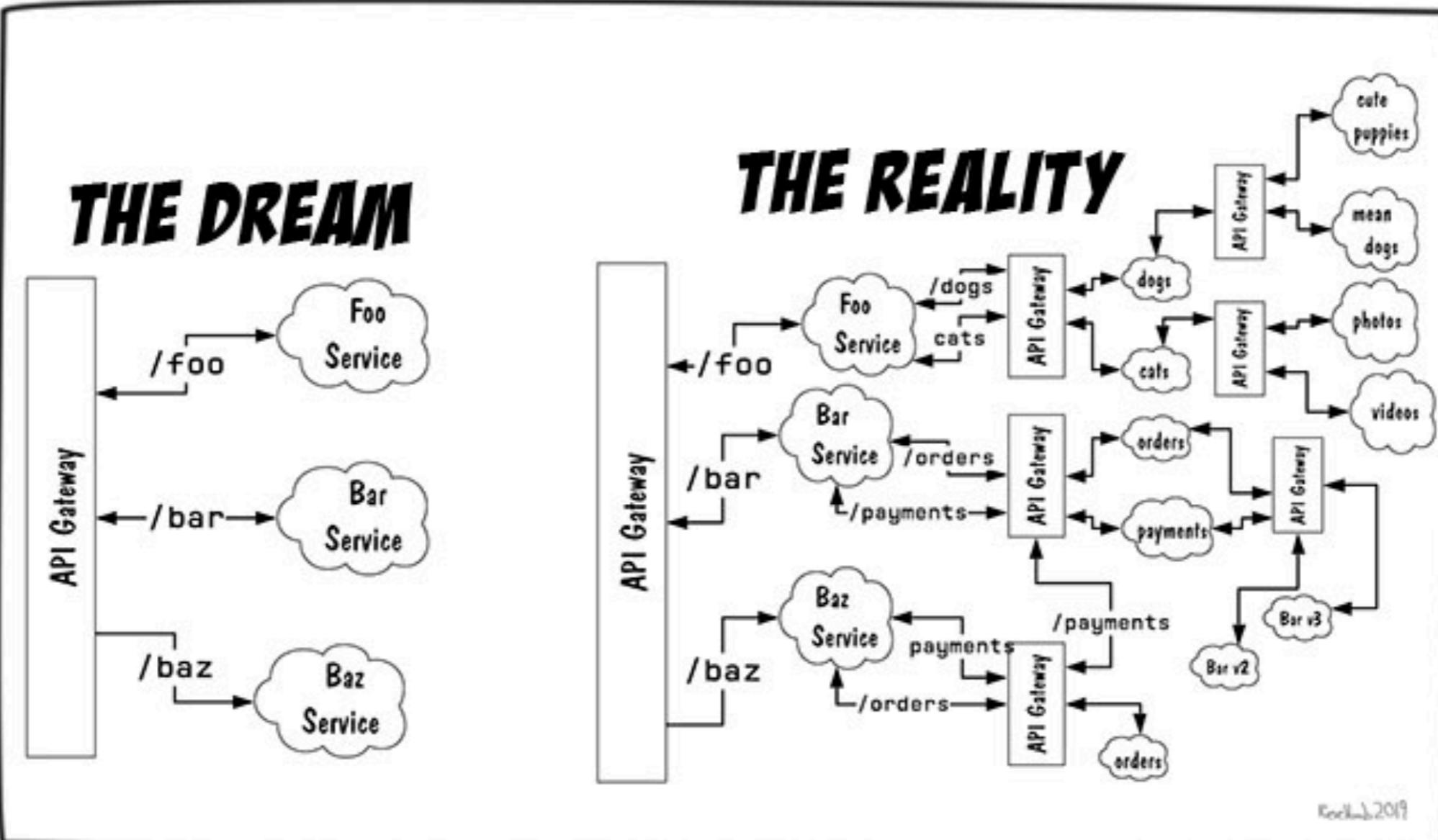
# Example with API Gateway



# Example with Authentication

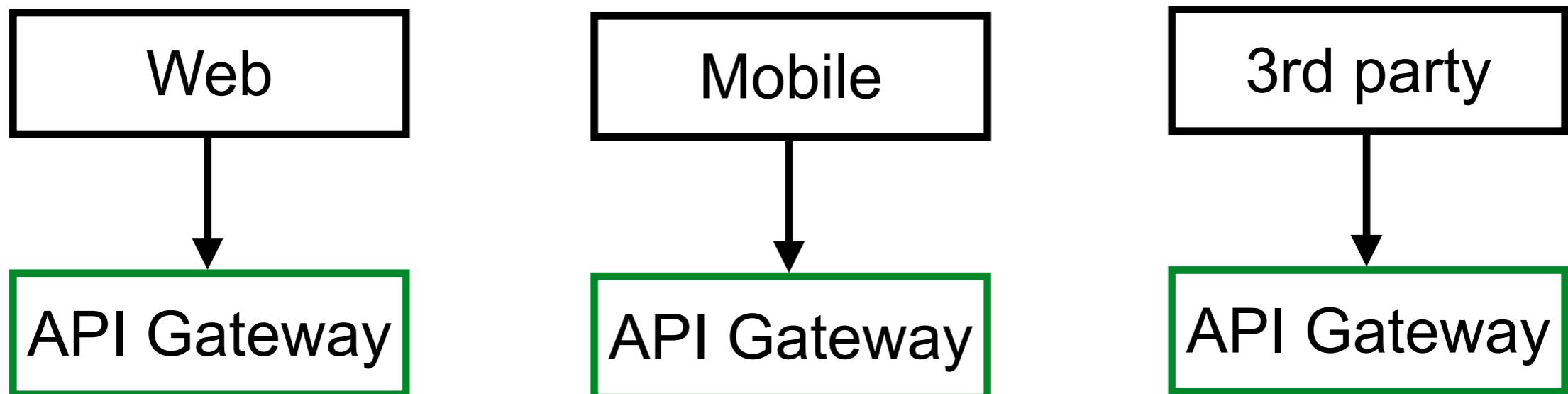


!!!

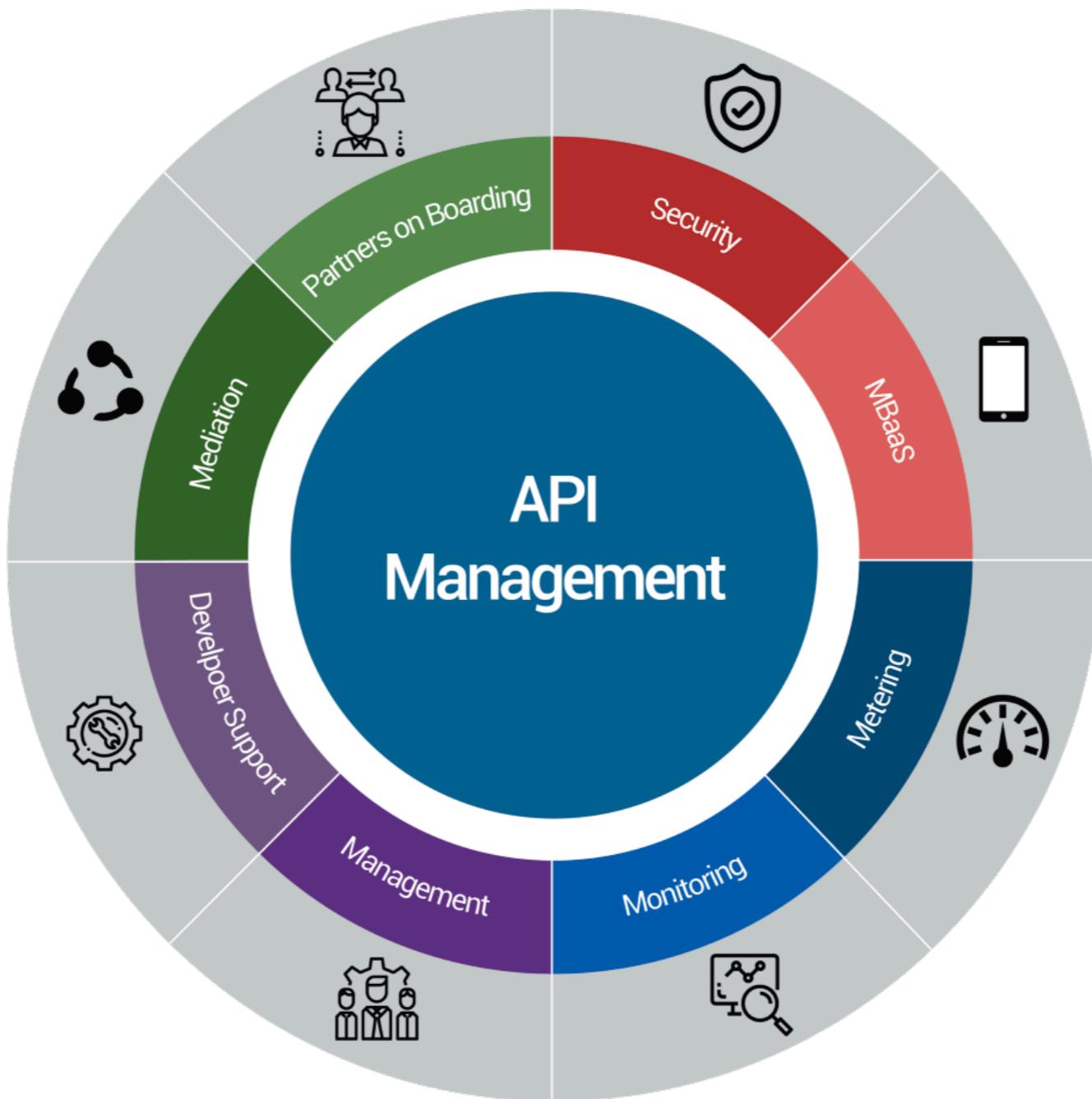


# Backend For Frontend

Separate API gateway for each kind of client



# API Management



# Pros of API Gateway

Simple architecture

Easy to manage

Separation of app layers and requests

Improve development

Increase server load

Buffer zone again attacks



# Cons of API Gateway

Increase complexity

Increase response time



# API Gateway ?

 <b>3SCALE</b> 3Scale Red Hat	 <b>Ambassador</b> Ambassador Ambassador Labs	 <b>APIOAK</b> APIOAK	 <b>APISIX</b> APISIX Apache Software Foundation	 <b>Gloo</b> Gloo Solo.io
 <b>GRAVITEE · IO</b> Gravitee.io Gravitee.io	 <b>Kong</b> Kong Kong	 <b>krakend</b> KrakenD Brutale	 <b>MuleSoft</b> MuleSoft Salesforce	 <b>Reactive Interaction Gateway</b> Reactive Interaction Gateway Accenture
 <b>Sentinel</b> Sentinel Alibaba Cloud	 <b>Tyk</b> Tyk Tyk	 <b>WSO2 API Microgateway</b> WSO2 API Microgateway WSO2		

<https://landscape.cncf.io/category=api-gateway&format=card-mode&grouping=category>



API Gateway with Kong

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

# Working with Kong

<https://konghq.com/kong/>



API Gateway with Kong

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

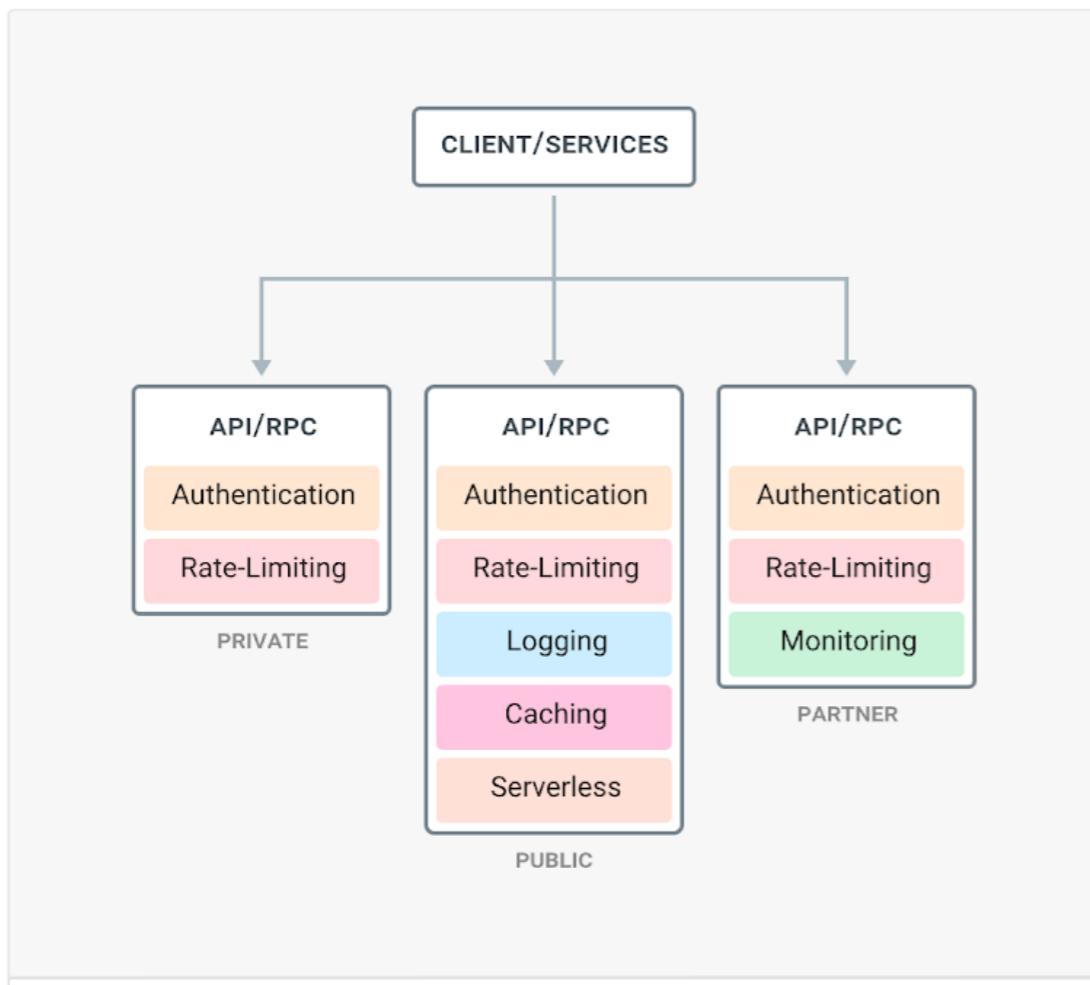
# Why Kong ?

Open source and cloud native  
Scalability (horizontal)  
Performance (based-on NGINX)  
Extensible with plugins  
RESTFul support



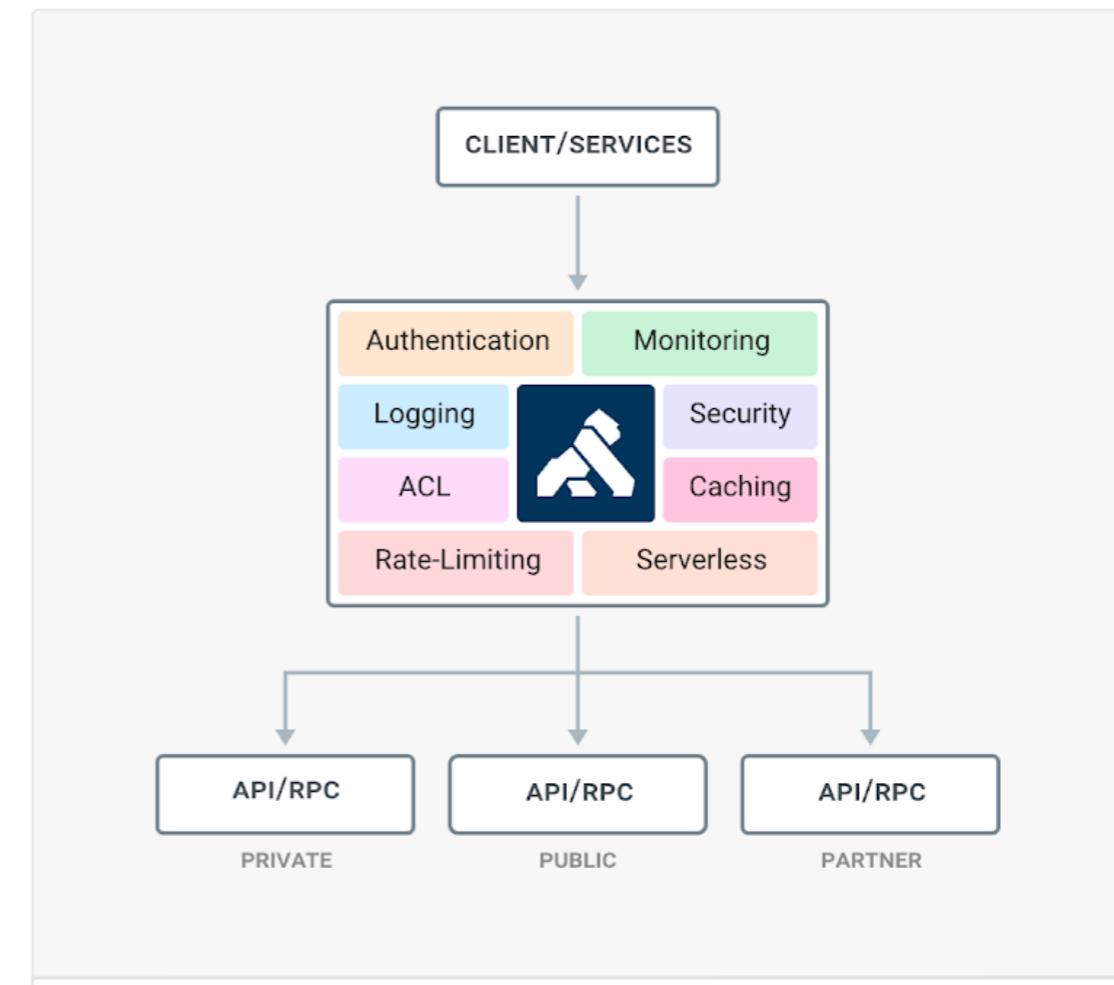
# The Kong way

The Redundant Old Way



- ✗ Common functionality is duplicated across multiple services
- ✗ Systems tend to be monolithic and hard to maintain
- ✗ Difficult to expand without impacting other services
- ✗ Productivity is inefficient because of system constraints

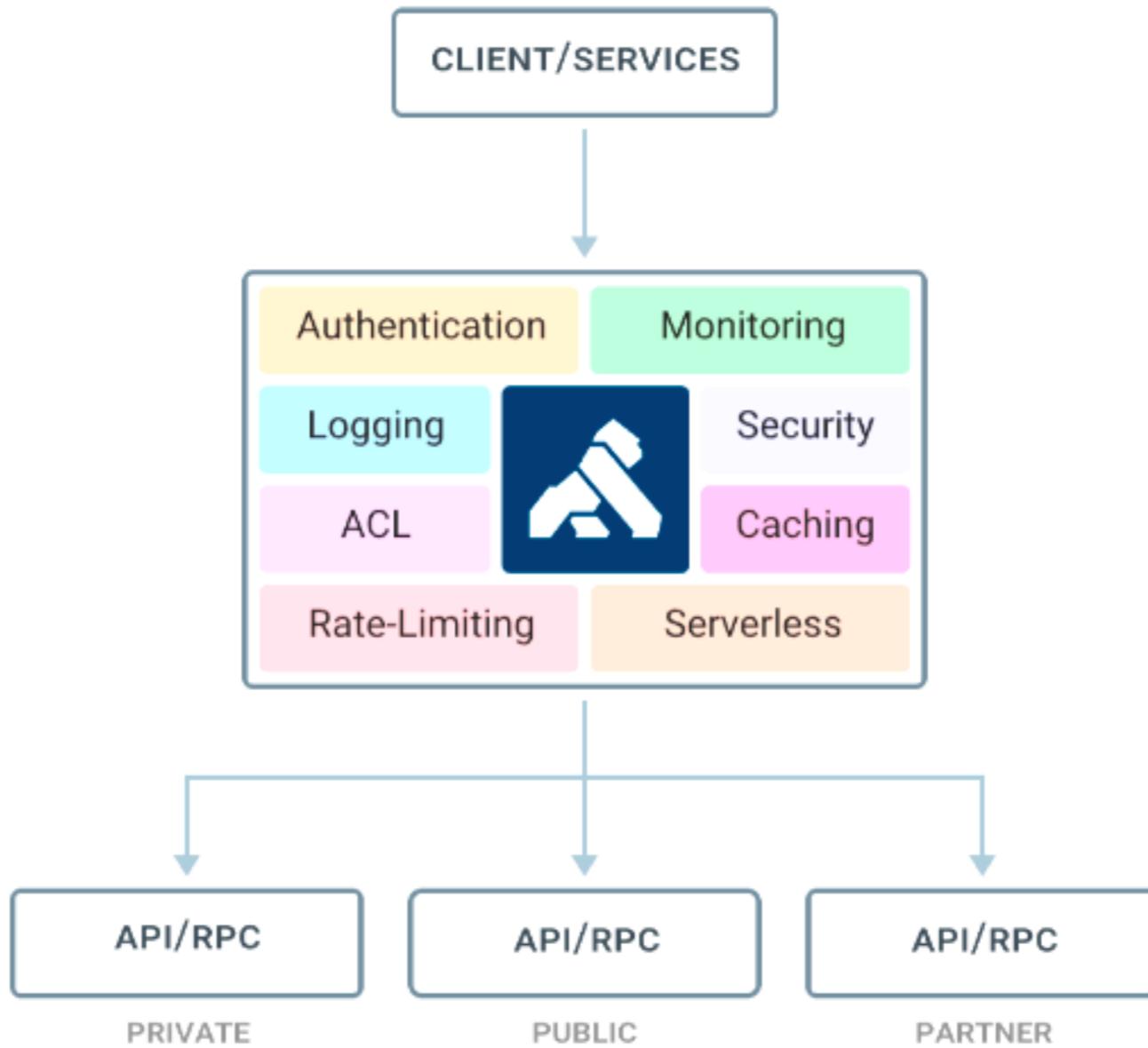
The Kong Way



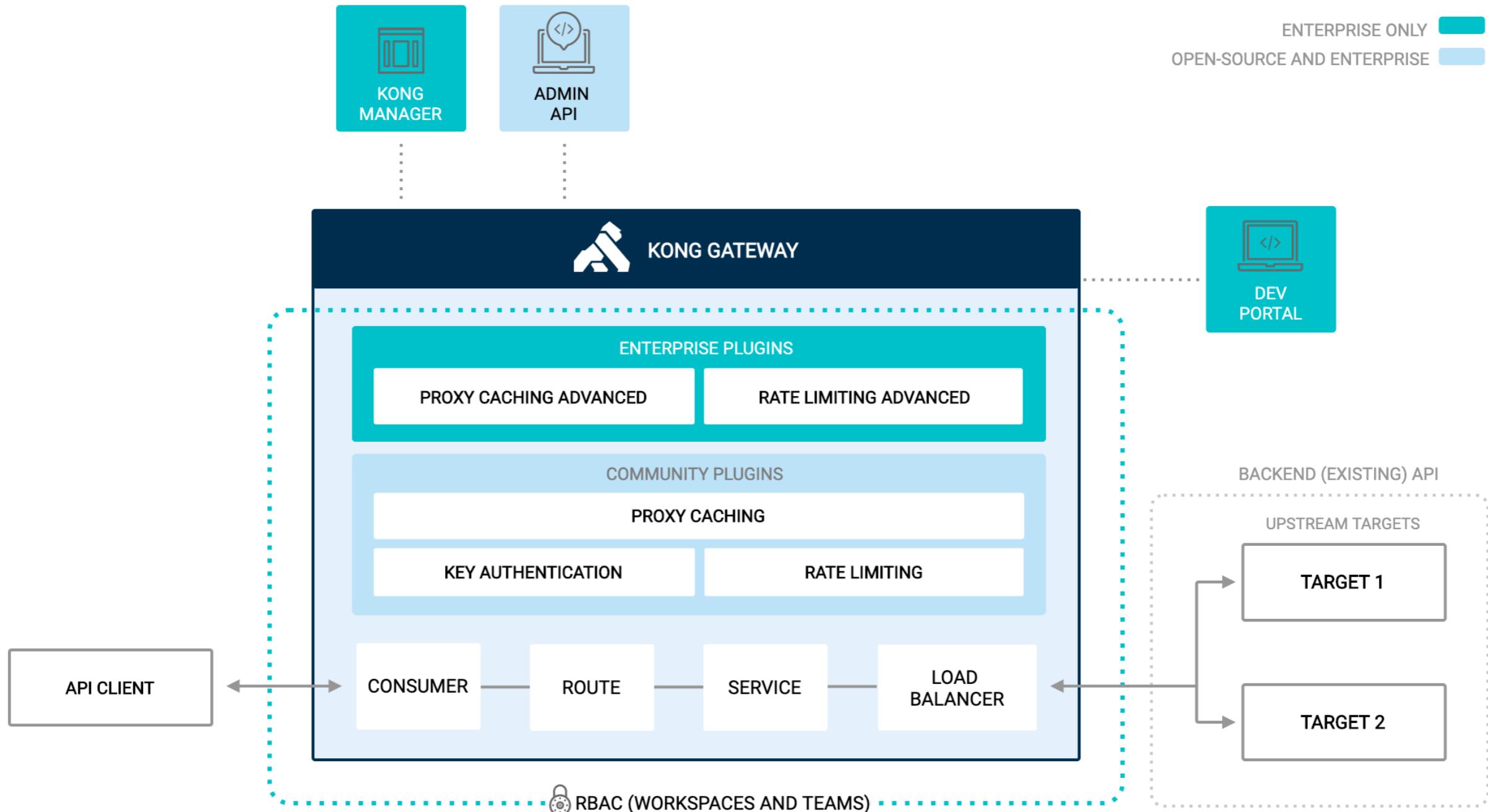
- ✓ Kong orchestrates common functionality
- ✓ Build efficient distributed architectures ready to scale
- ✓ Expand functionality from one place with a simple command
- ✓ Focus on your product and let Kong do the REST



# The Kong way



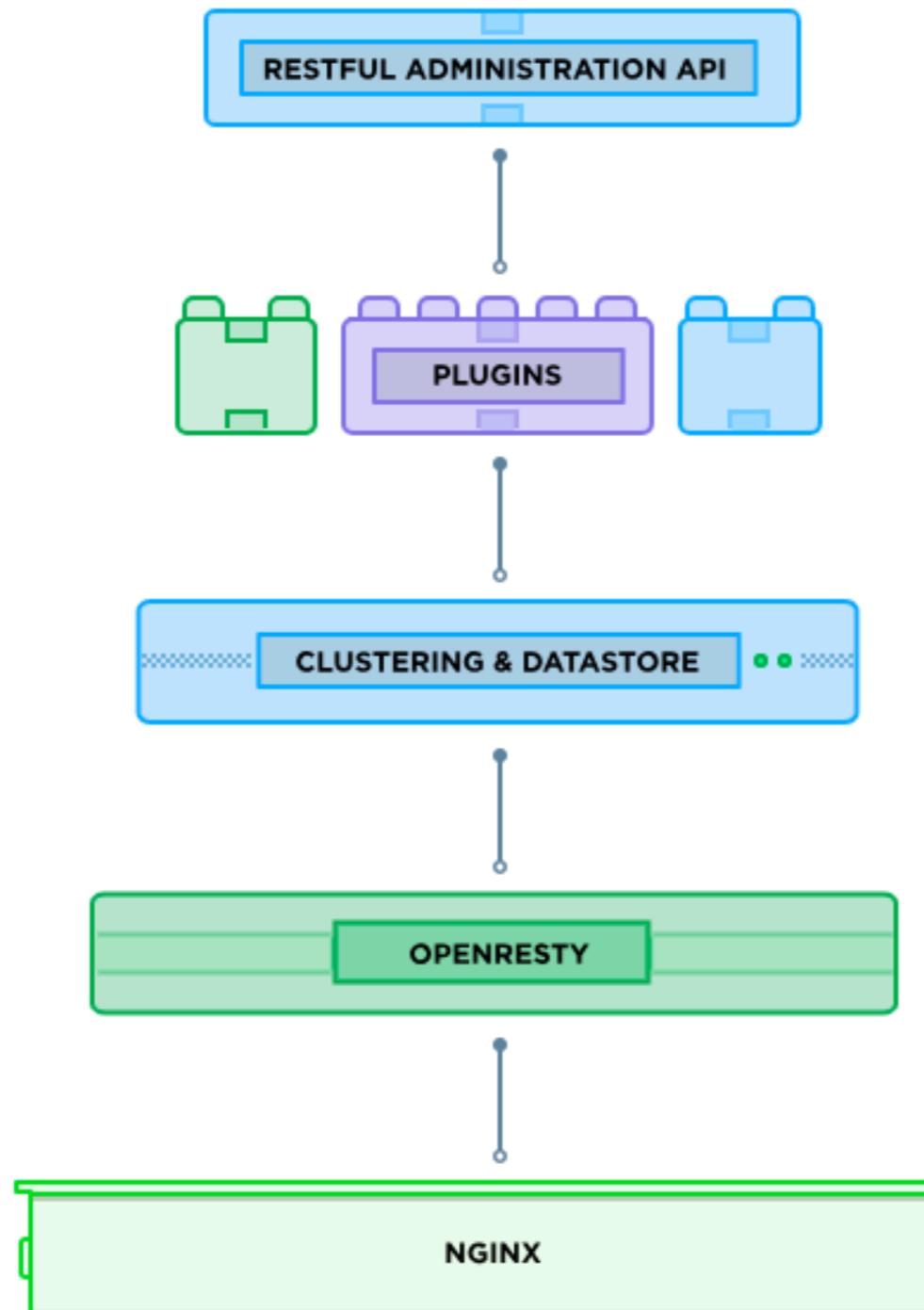
# Kong architecture



<https://docs.konghq.com/getting-started-guide/2.1.x/overview/>



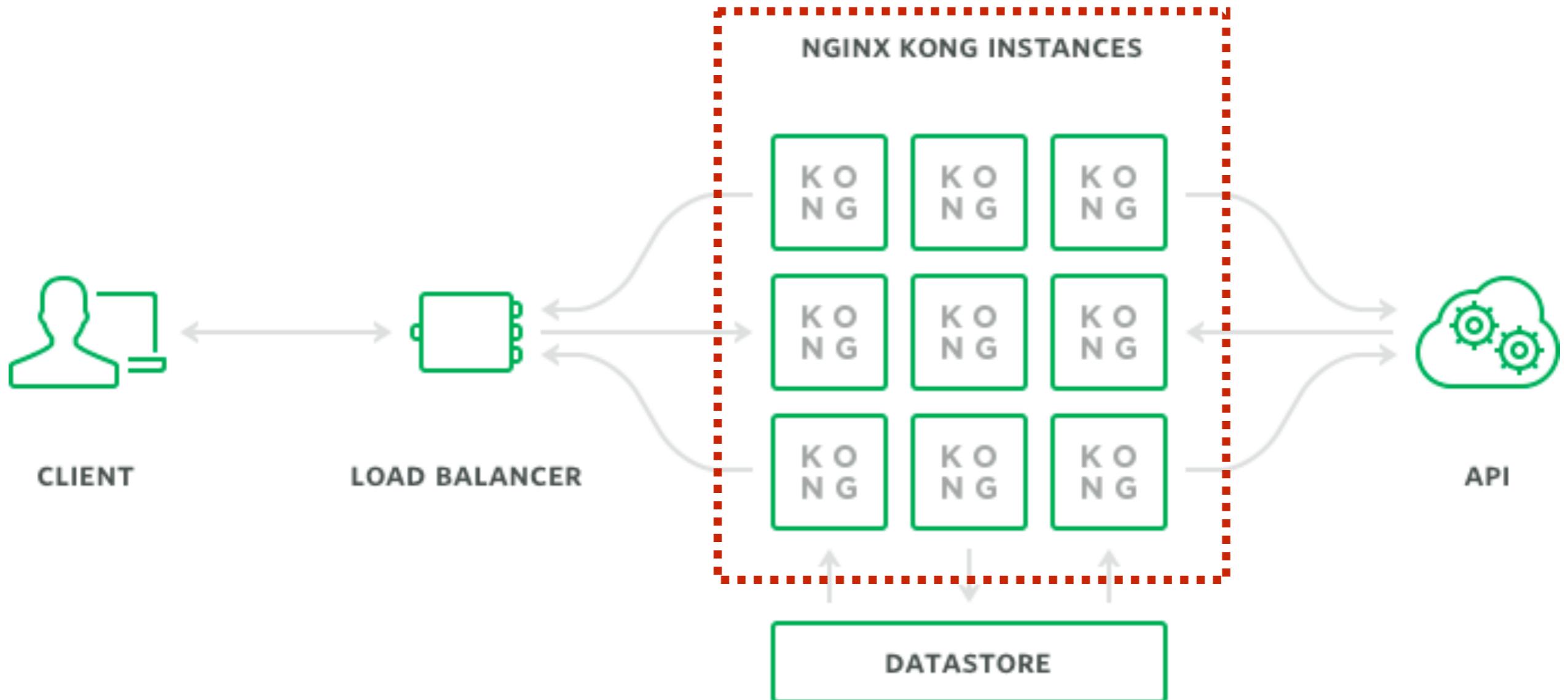
# Kong architecture



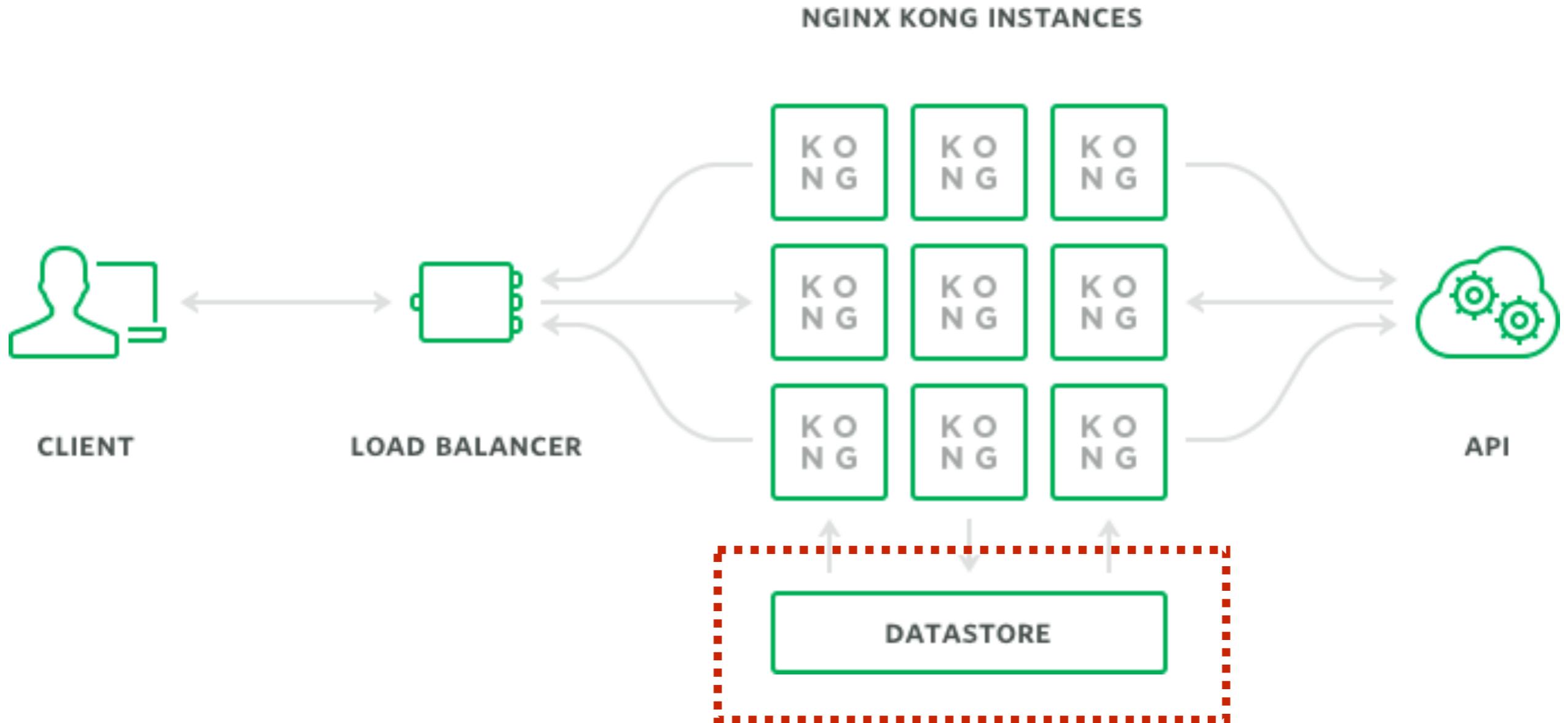
- Administer Kong via RESTful API
- Automate/orchestrate for CI/CD & DevOps
- Extensible with plugins
  
- Create plugins with Lua
- Implement powerful customizations
- Integrate with third-party services
  
- Choice of Cassandra or PostgreSQL
- Scales from laptop to global cluster
- In-memory caching for performance
  
- Intercept Request/Response lifecycle
- Extends underlying NGINX
- Scriptable via Lua
  
- Proven, high-performance foundation
- HTTP and reverse proxy server
- Handles low-level operations



# Kong scaling (Kong server)



# Kong scaling (Kong datastore)



# Features :: Authentication

Basic Auth

KeyAuth

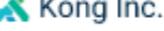
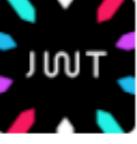
OAuth/OAuth 2

JWT (JSON Web Token)

LDAP



# Features :: Authentication

 <p><b>Portal Application Registration</b> Allow portal developers to register applications against Services Support by:  Kong Inc.</p>	 <p><b>Basic Authentication</b> Add Basic Authentication to your Services Support by:  Kong Inc.</p>	 <p><b>HMAC Authentication</b> Add HMAC Authentication to your Services Support by:  Kong Inc.</p>
 <p><b>Kong JWT Signer</b> Verify and (re-)sign one or two tokens in a request Support by:  Kong Inc.</p>	 <p><b>JWT</b> Verify and authenticate JSON Web Tokens Support by:  Kong Inc.</p>	 <p><b>Key Authentication - Encrypted</b> Add key authentication to your Services Support by:  Kong Inc.</p>

<https://docs.konghq.com/hub/>



# Features :: Security

ACLs

CORS (Cross-Origin Resource Sharing)

Dynamic SSL

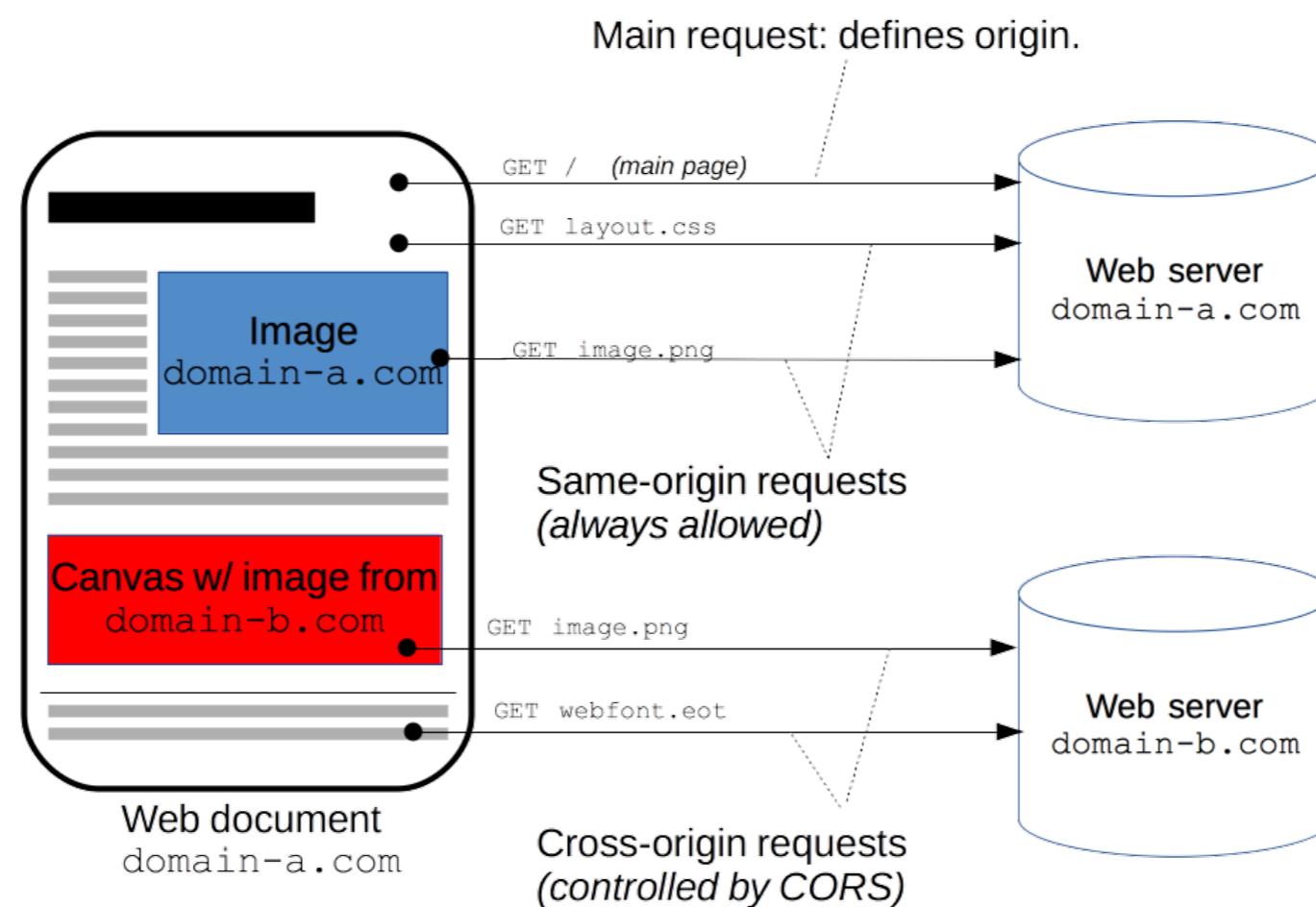
OP Blacklists

Bot detection



# CORS (Cross-Origin Resource Sharing)

One way the server at the other end (not the client code in the browser) can relax the same origin policy.



# Features :: Security



**ACME**  
Let's Encrypt and ACMEv2 integration with Kong  
Support by:  




**Bot Detection**  
Detect and block bots or custom clients  
Support by:  




**CORS**  
Allow developers to make requests from the browser  
Support by:  




**IP Restriction**  
Allow or deny IPs that can make requests to your Services  
Support by:  




**Cleafy plugin for Kong**  
Integrate Kong API GW with Cleafy threat detection & protection for API-based apps  
Support by:  
Cleafy



**Approov API Threat Protection**  
Approov ensures that only genuine and unmodified instances of your mobile app can connect to your server or cloud backend  
Support by:  
CriticalBlue Ltd

<https://docs.konghq.com/hub/>



API Gateway with Kong

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

# Features :: Traffic Control

Rate limiting

Response rate limiting

Request size limiting

Proxy cache



# Features :: Traffic Control



**ACL**  
Control which Consumers can access Services  
  
Support by:  
 Kong Inc.



**Canary Release**  
Slowly roll out software changes to a subset of users  
  
Support by:  
 Kong Inc.



**Forward Proxy Advanced**  
Allows Kong to connect to intermediary transparent HTTP proxies  
  
Support by:  
 Kong Inc.



**GraphQL Proxy Caching Advanced**  
Cache and serve commonly requested responses in Kong  
  
Support by:  
 Kong Inc.



**GraphQL Rate Limiting Advanced**  
Provide rate limiting for GraphQL queries  
  
Support by:  
 Kong Inc.



**Proxy Caching Advanced**  
Cache and serve commonly requested responses in Kong  
  
Support by:  
 Kong Inc.

<https://docs.konghq.com/hub/>



API Gateway with Kong

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

# Features :: Transforms

Request transformer  
Response transformer  
Correlation ID



# Features :: Transforms

 <b>Correlation ID</b> Correlate requests and responses using a unique ID Support by: 	 <b>DeGraphQL</b> Transform a GraphQL upstream into a REST API Support by: 	 <b>Exit Transformer</b> Customize Kong exit responses sent downstream Support by: 
 <b>gRPC-gateway</b> Access gRPC services through HTTP REST Support by: 	 <b>gRPC-Web</b> Allow browser clients to call gRPC services Support by: 	 <b>Kafka Upstream</b> Transform requests into Kafka messages in a Kafka topic Support by: 

<https://docs.konghq.com/hub/>



# Features :: Visibility

Logs over TCP/UDP/HTTP

Syslog

StatD

DataDog



# Features :: Visibility

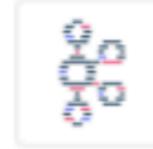


**File Log**  
Append request and response data to a log file  
Support by:  
 Kong Inc.



**HTTP Log**  
Send request and response logs to an HTTP server  
Support by:  
 Kong Inc.

**Enterprise**



**Kafka Log**  
Publish logs to a Kafka topic  
Support by:  
 Kong Inc.



**Loggly**  
Send request and response logs to Loggly  
Support by:  
 Kong Inc.

**Enterprise**



**StatsD Advanced**  
Send metrics to StatsD with more flexible options  
Support by:  
 Kong Inc.



**StatsD**  
Send request and response logs to StatsD  
Support by:  
 Kong Inc.

<https://docs.konghq.com/hub/>



# Features :: Visibility



## Datadog

Visualize metrics on  
Datadog

Support by:  
 Kong Inc.



## Prometheus

Expose metrics related to  
Kong and proxied  
Upstream services in  
Prometheus exposition  
format

Support by:  
 Kong Inc.



## Zipkin

Propagate Zipkin spans  
and report space to a  
Zipkin server

Support by:  
 Kong Inc.



## API Fortress HTTP Log

Record mock responses  
with API Fortress.

Support by:  
API Fortress



## ArecaBay MicroSensor

Discover, Monitor, and  
Secure your APIs at  
object/data level.

Support by:  
ArecaBay



## Moesif API Analytics

User Behavior API  
analytics and observability

Support by:  
Moesif

<https://docs.konghq.com/hub/>



API Gateway with Kong

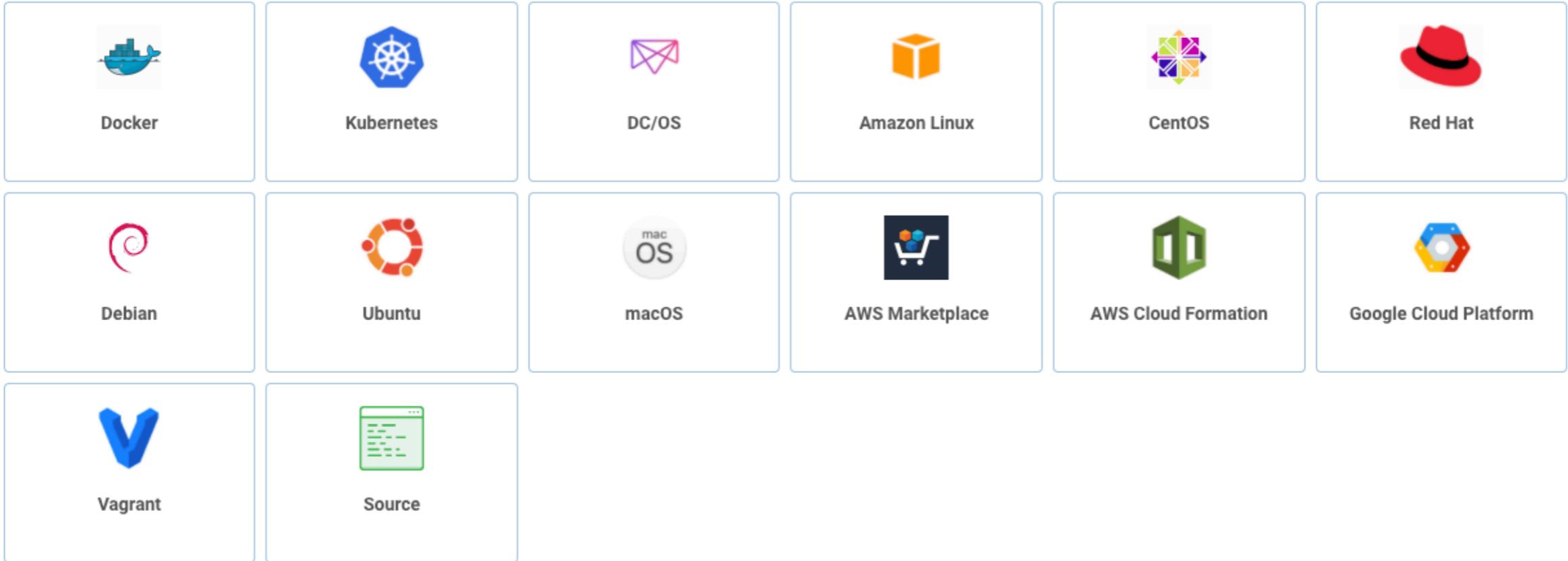
© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

**We can always build out own  
plugin for Kong using Lua**

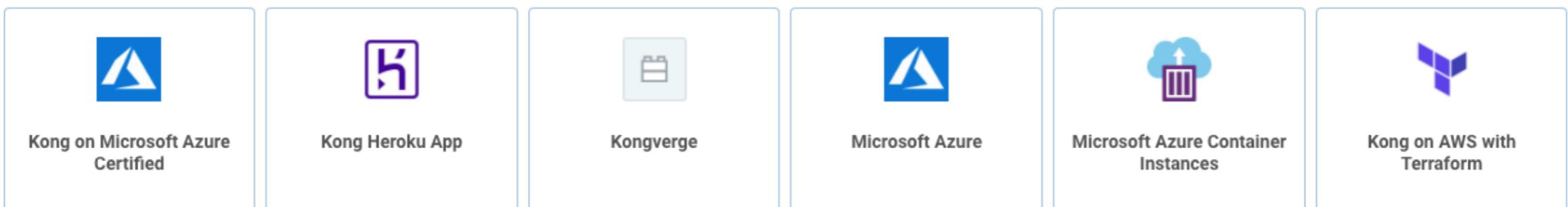


# Install Kong

Supported by Kong Inc



Supported by the Community



<https://konghq.com/get-started/#install>



# Kong with Database

PostgreSQL  
Apache Cassandra  
DB-less



# Installation

```
$ docker-compose up -d kong-database  
$ docker-compose up migrations  
$ docker-compose up kong
```



# **Kong Admin API (8001/8444)**

# **Kong Proxy (8000/8443)**



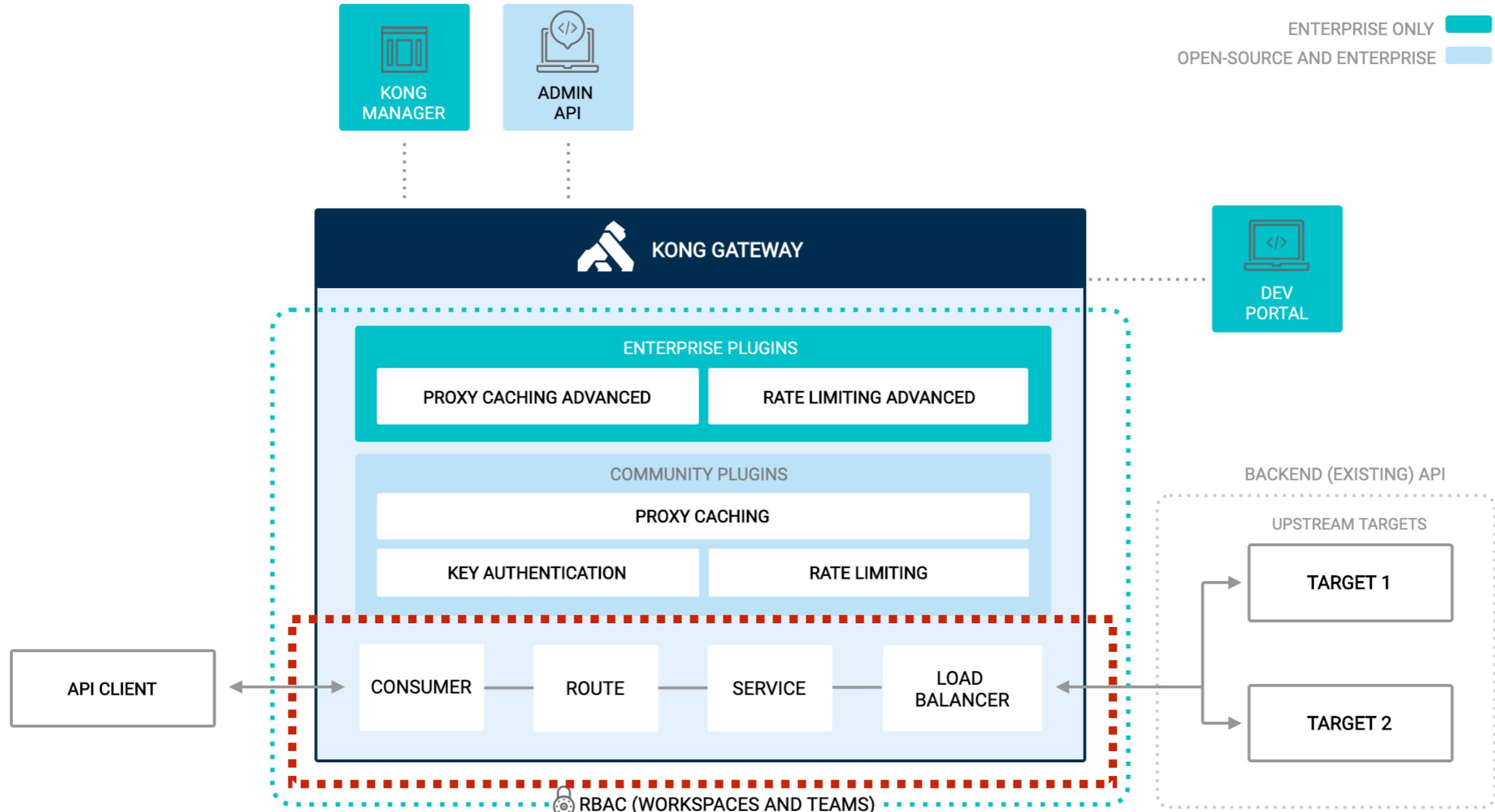
# List of plugins (default)

```
"available_on_server": {  
    "correlation-id": true,  
    "pre-function": true,  
    "cors": true,  
    "ldap-auth": true,  
    "loggly": true,  
    "hmac-auth": true,  
    "zipkin": true,  
    "request-size-limiting": true,  
    "azure-functions": true,  
    "request-transformer": true,  
    "oauth2": true,  
    "response-transformer": true,  
    "ip-restriction": true,  
    "statsd": true,  
    "jwt": true,  
    "proxy-cache": true,  
    "basic-auth": true,  
    "key-auth": true,
```

```
"http-log": true,  
"datadog": true,  
"tcp-log": true,  
"rate-limiting": true,  
"post-function": true,  
"prometheus": true,  
"acl": true,  
"syslog": true,  
"file-log": true,  
"acme": true,  
"udp-log": true,  
"response-ratelimiting": true,  
"aws-lambda": true,  
"session": true,  
"bot-detection": true,
```



# Kong workshop



<https://docs.konghq.com/getting-started-guide/2.1.x/overview/>



# Concepts

Service  
Route  
Consumer  
Admin API  
Plugins  
Load balancing



# Service

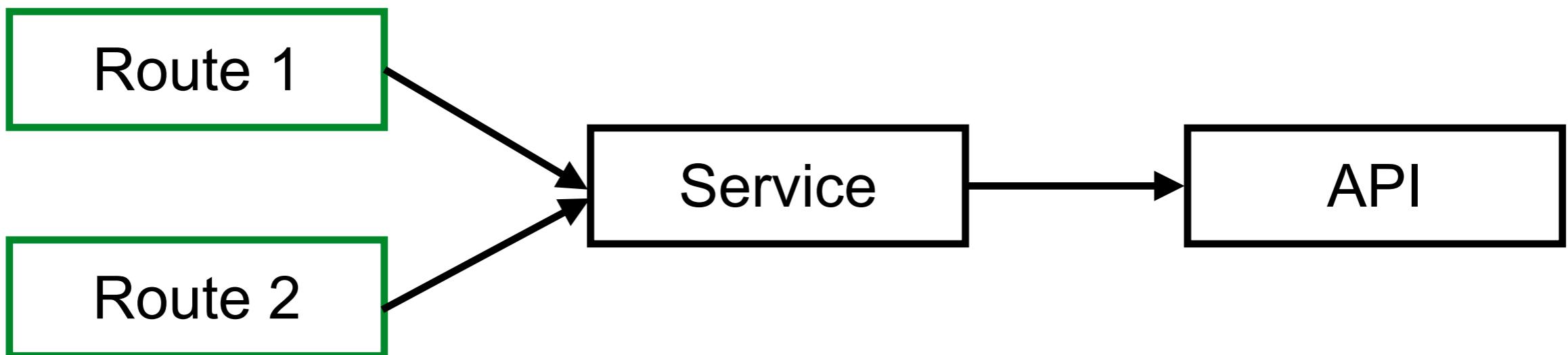
A Service object is the ID Kong Gateway uses to refer to the upstream APIs and microservices it manages.



# Route

Routes specify how requests are sent to their services after they reach the API gateway.

A single service can have many routes.



# Consumer

Consumers represent end users of your API.

Consumer objects let you control who can access  
your APIs.



# Admin API

Kong Gateway comes with an internal RESTful API for administration purposes.

API commands can be run on any node in the cluster, and the configuration will apply consistently on all nodes.



# Plugin

Plugins provide a modular system for modifying and controlling Kong Gateway's capabilities.

*Plugins are one of the most important features of Kong, developed by Lua language*



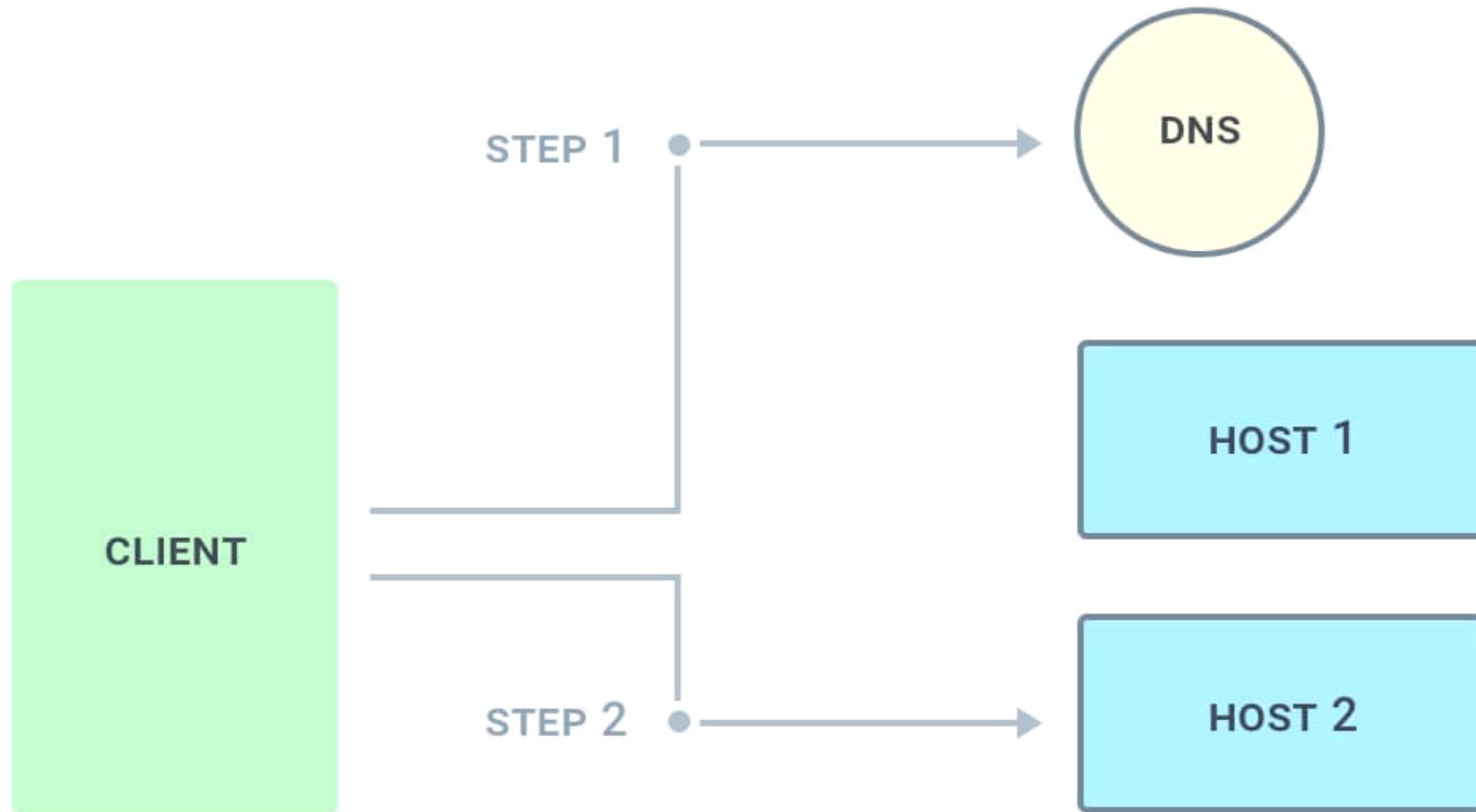
# Load balancing

Kong Gateway provides two methods for load balancing

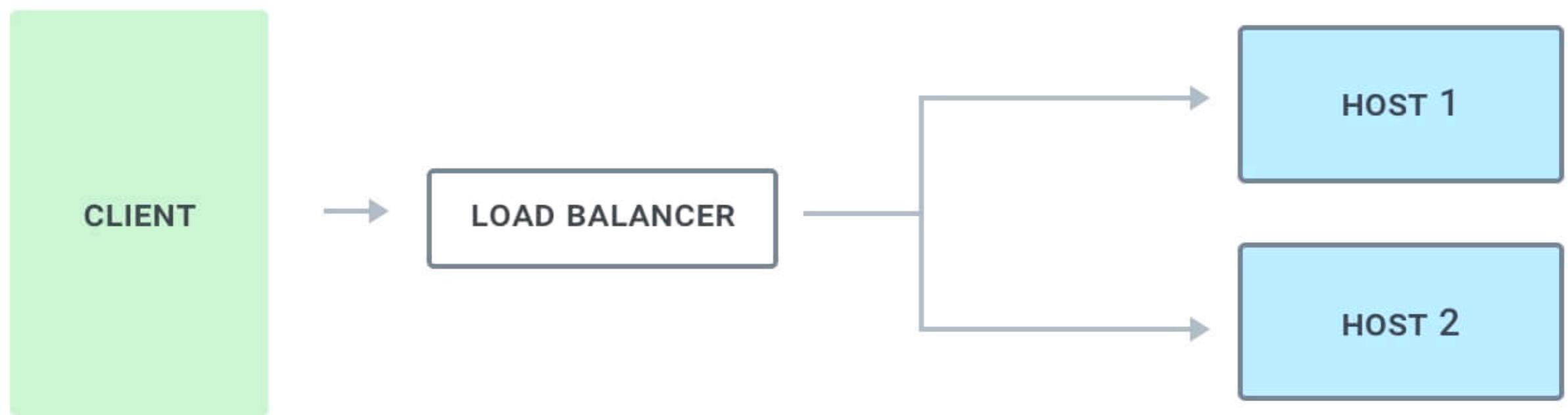
**DNS-based or using a Ring-balancer**



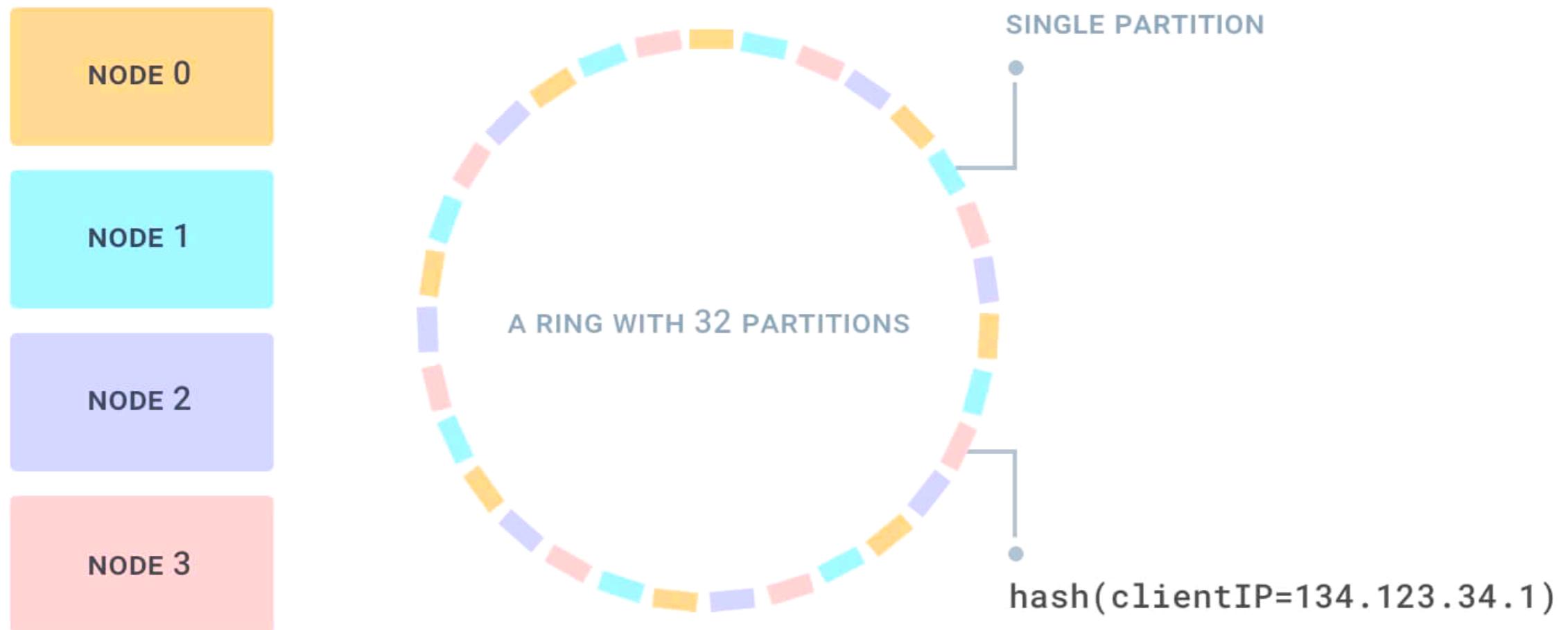
# Domain Name Server (DNS)



# Round Robin



# Ring Balancer



# Health Check

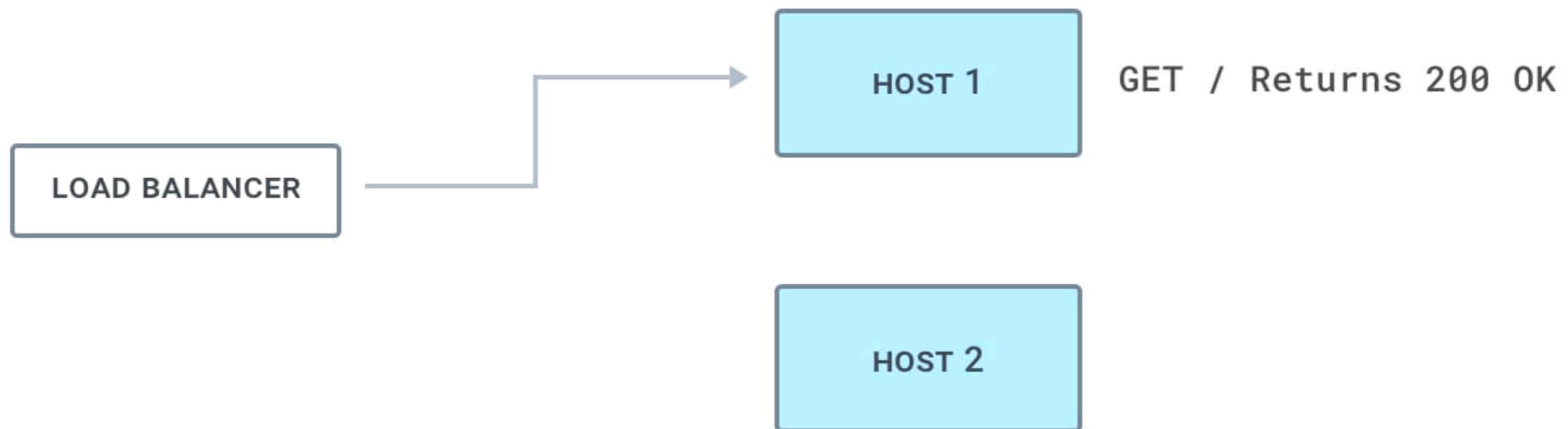
Health checks can help us detect failed hosts so the load balancer can stop requests to them

## Active vs Passive



# Active Health Check

The load balancer periodically “probes” upstream servers by sending a special health check request



# Passive Health Check

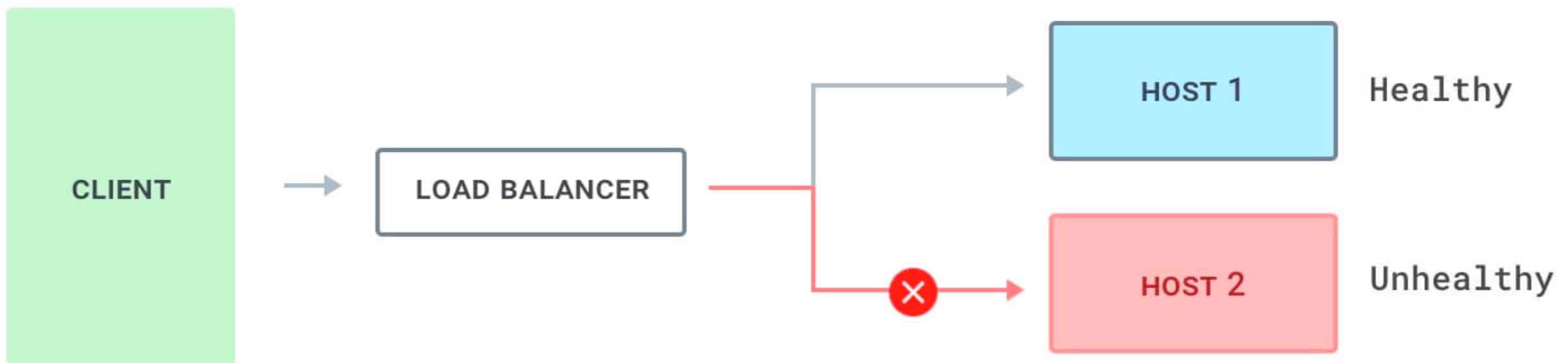
The load balancer monitors real requests as they pass through.

If the number of failed requests exceeds a threshold, it marks the host as unhealthy.



# Circuit Breaker

When you know that a given host is unhealthy, its best to “break the circuit” so that traffic flows to healthy hosts instead



# Kong workshop

Kong Admin APIs

Kong services

Kong routes

Kong consumers

Kong plugins

Kong GUI

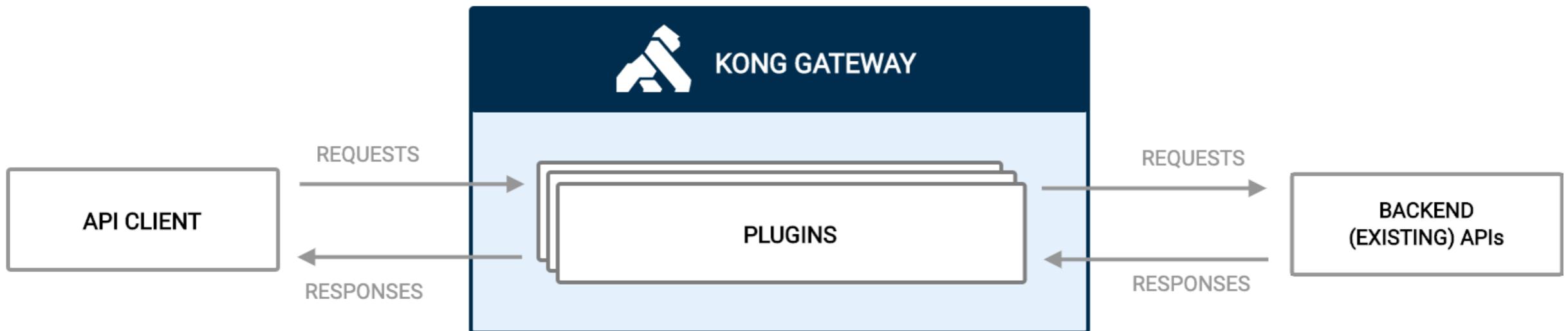


# How to administrate and configure ?

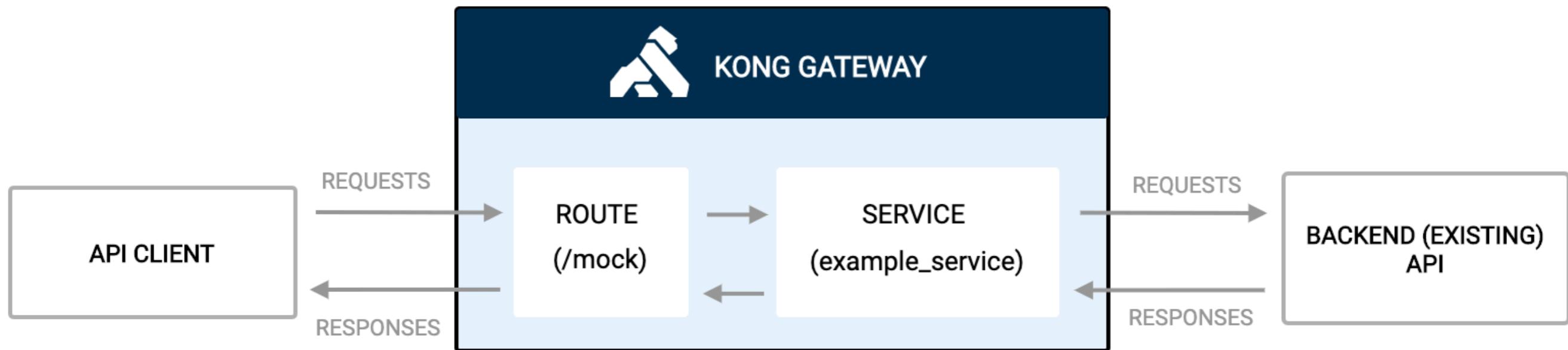
CLI  
Postman  
GUI (e.g. Konga)



# Kong gateway

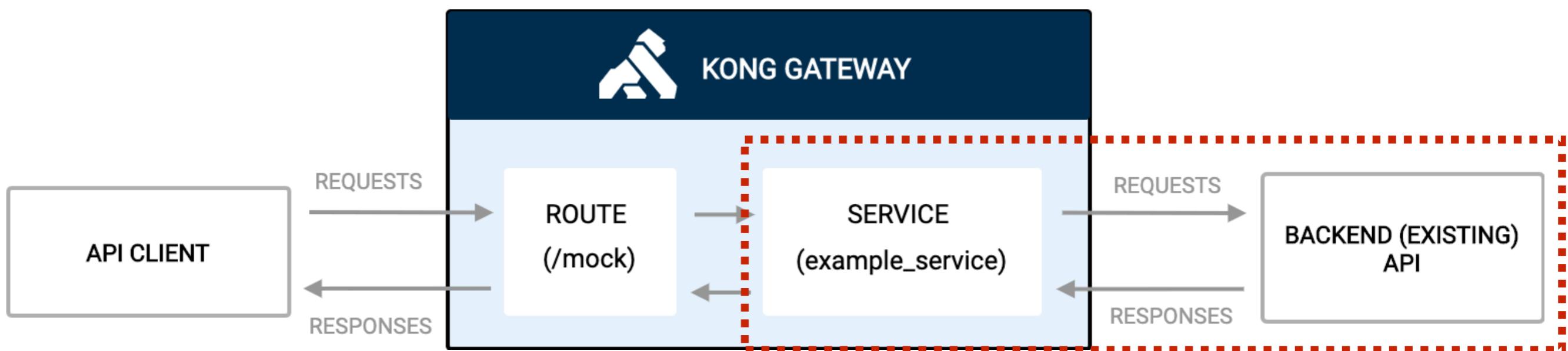


# Services and Routes



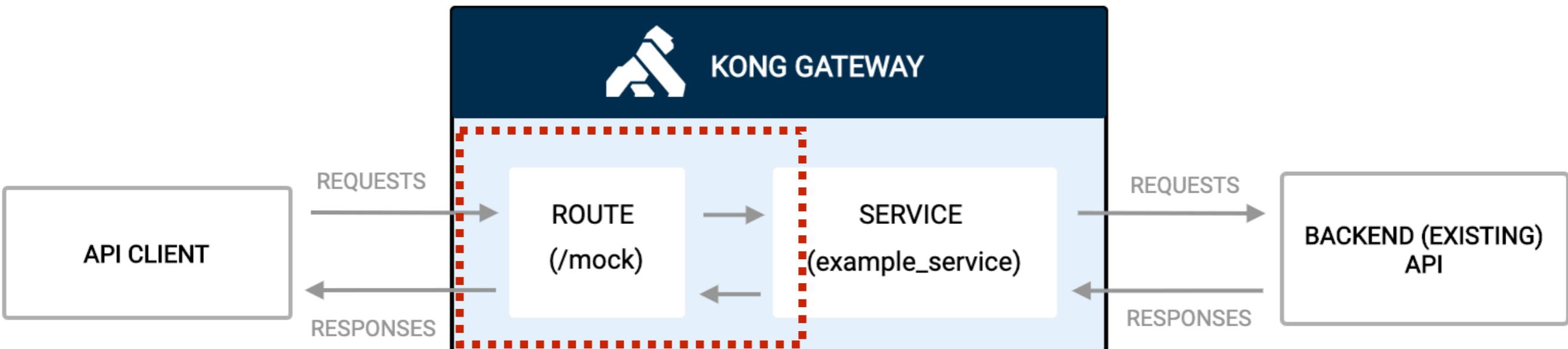
# Services

Entity represent and external upstream API



# Routes

Routes determine how requests are sent to services



*A single service can have many Routes*



# **Workshop services and routes**

## **CRUD**



# Workshop

## Working with {JSON}Placeholder

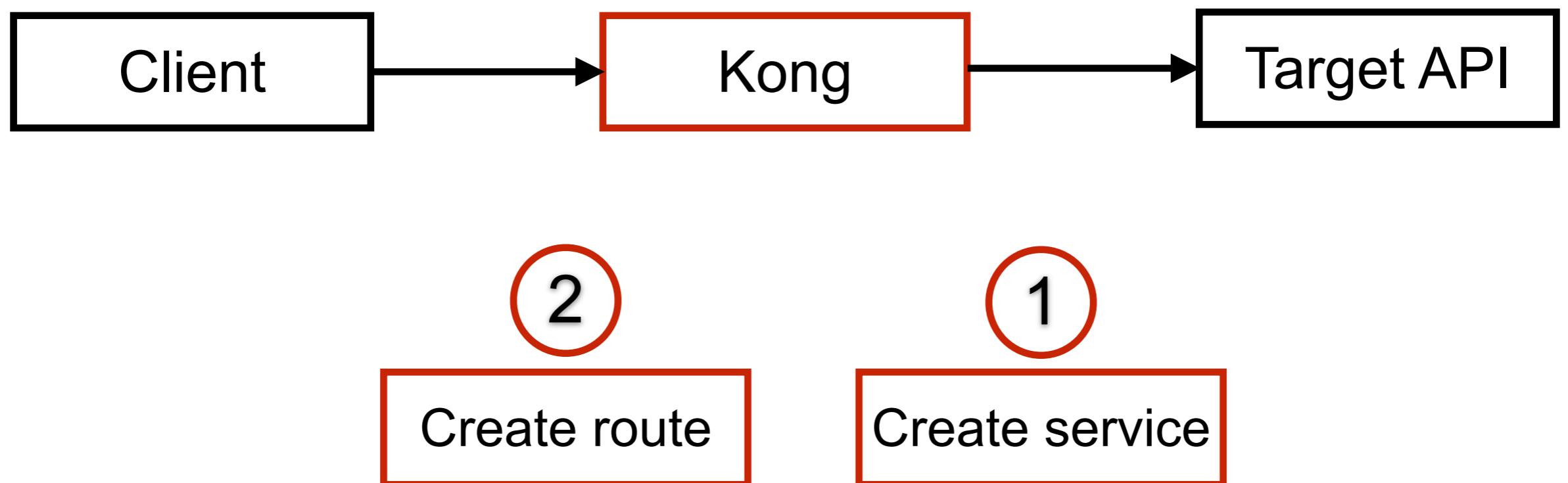


<https://jsonplaceholder.typicode.com/>



# Workshop

## Working with {JSON}Placeholder



<https://jsonplaceholder.typicode.com/>



# Working with Kong plugins

<https://docs.konghq.com/hub/>



# Working with plugins

Add plugin to service

Add plugin to route



# Authentication

Basic auth  
Key auth  
JWT  
OAuth 2



# JWT (JSON Web Token)

## Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.Sf1KxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

## Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```

VERIFY SIGNATURE

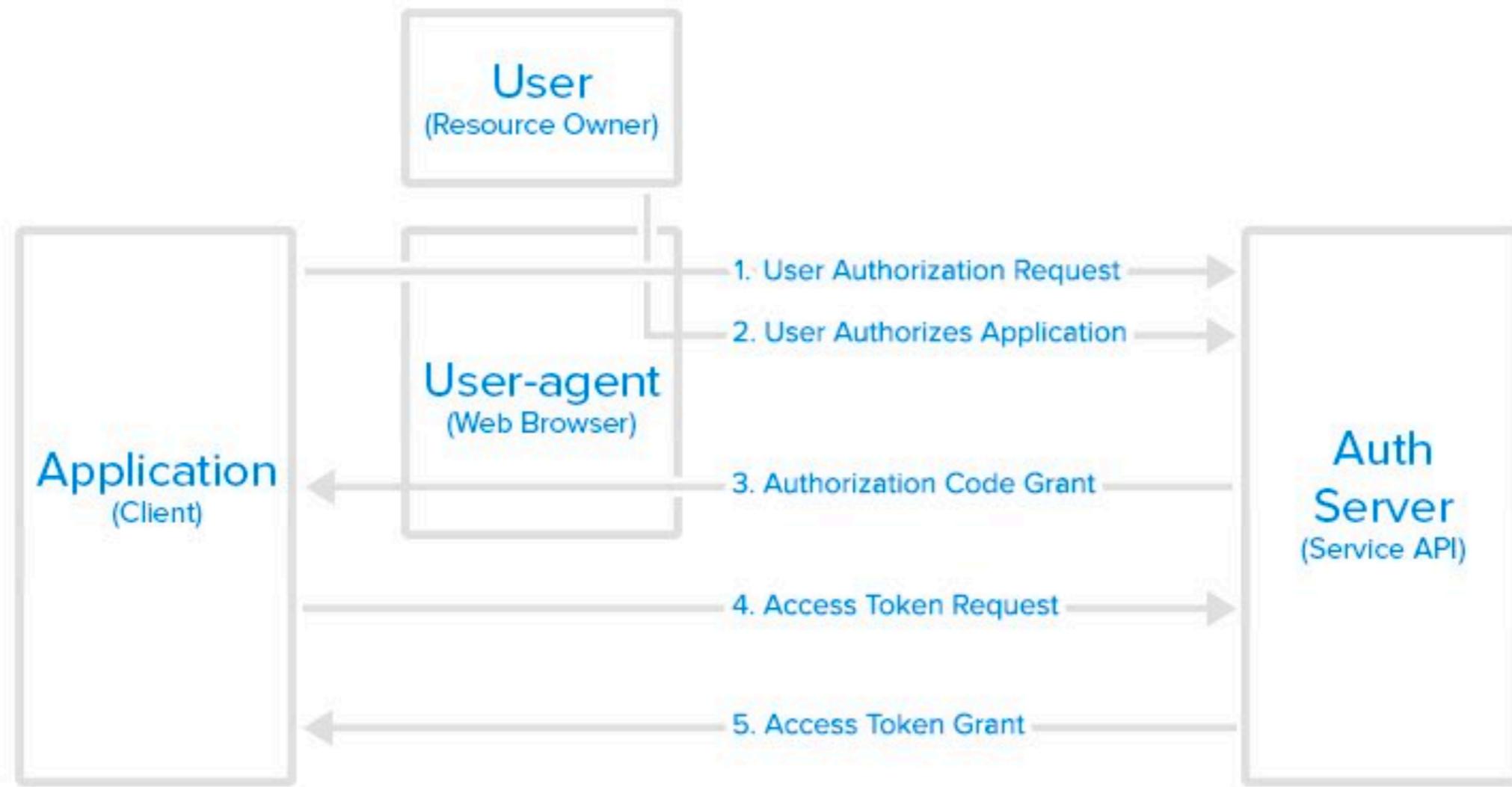
```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
)  secret base64 encoded
```

<https://jwt.io/>



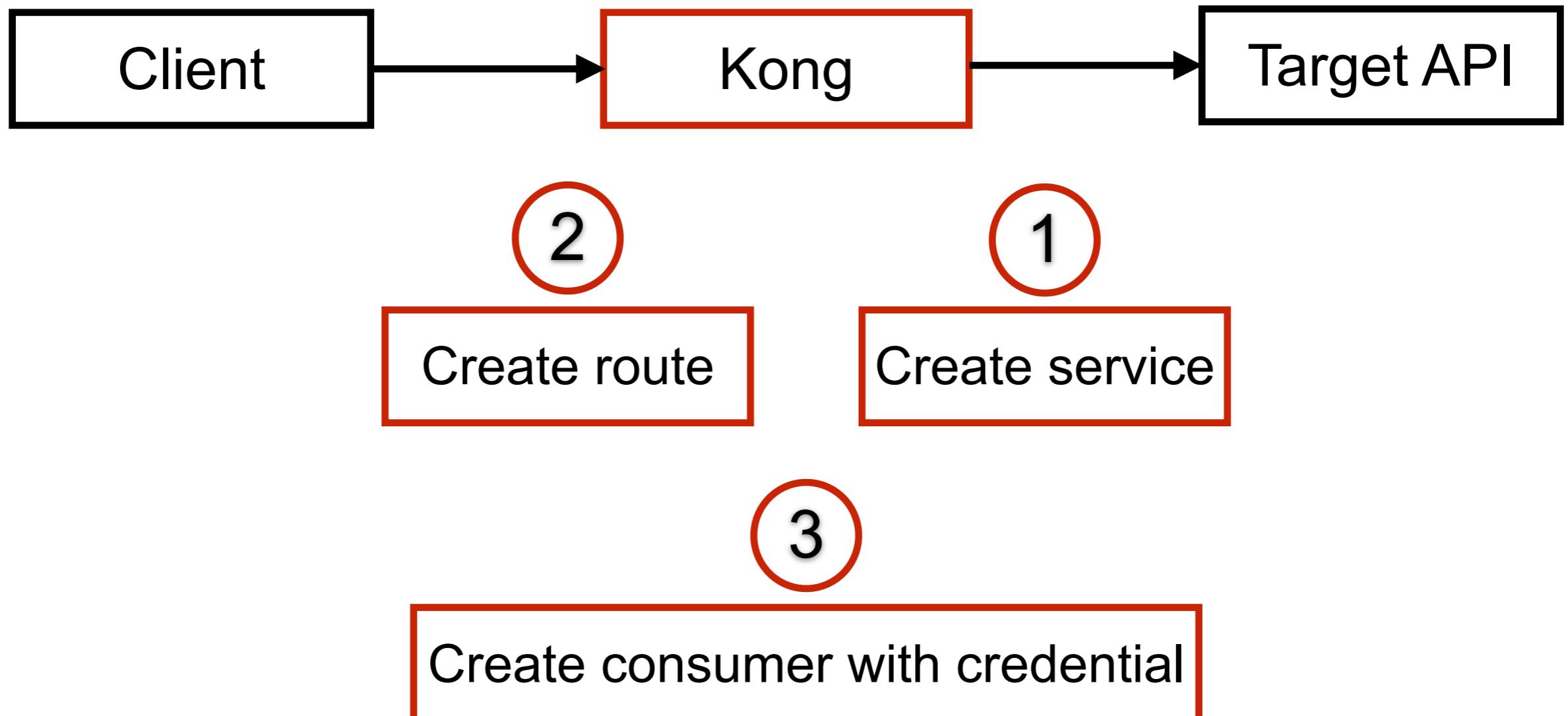
# OAuth 2 Workflow

## Authorization Code Flow



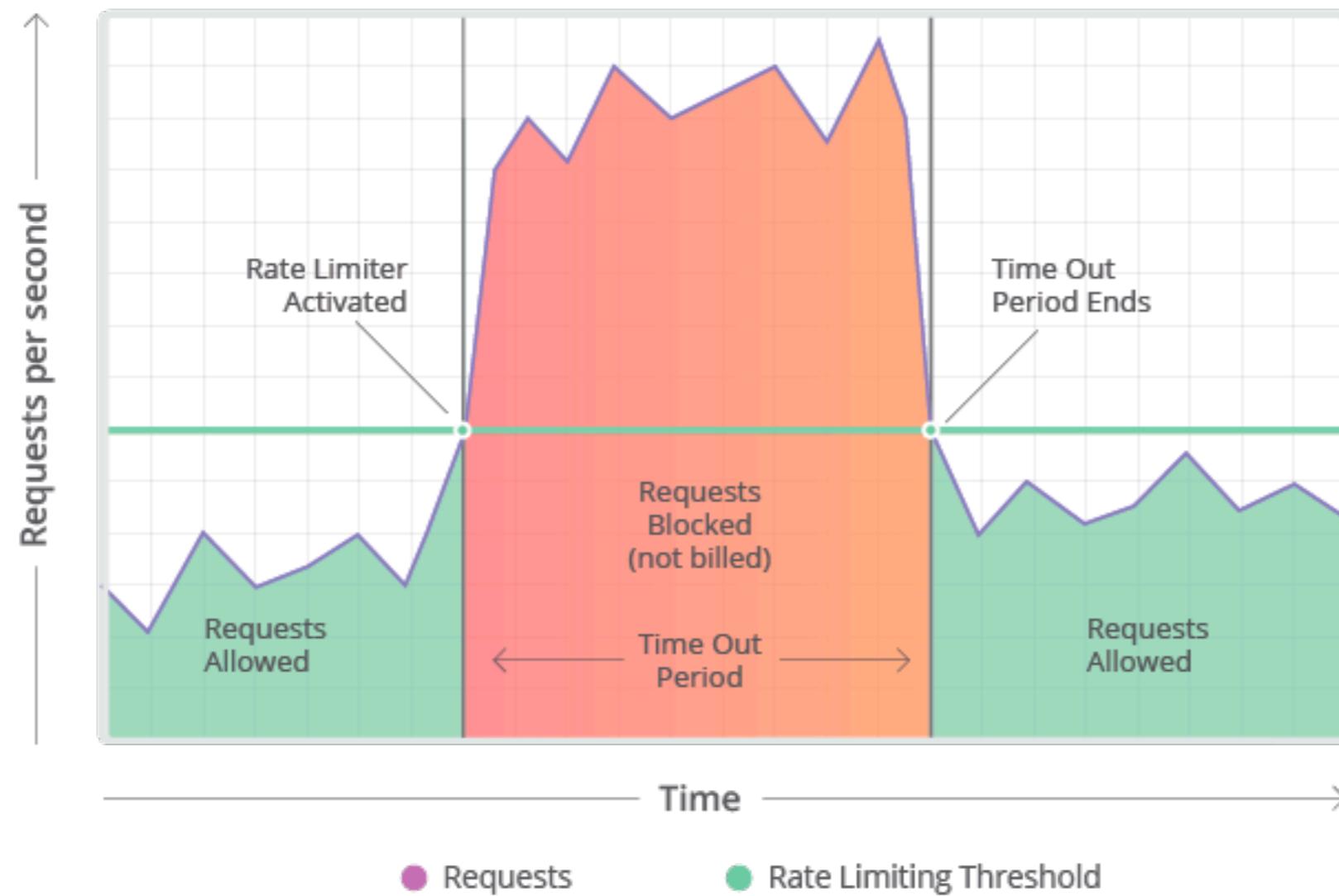
# Workshop

## Authentication with Kong



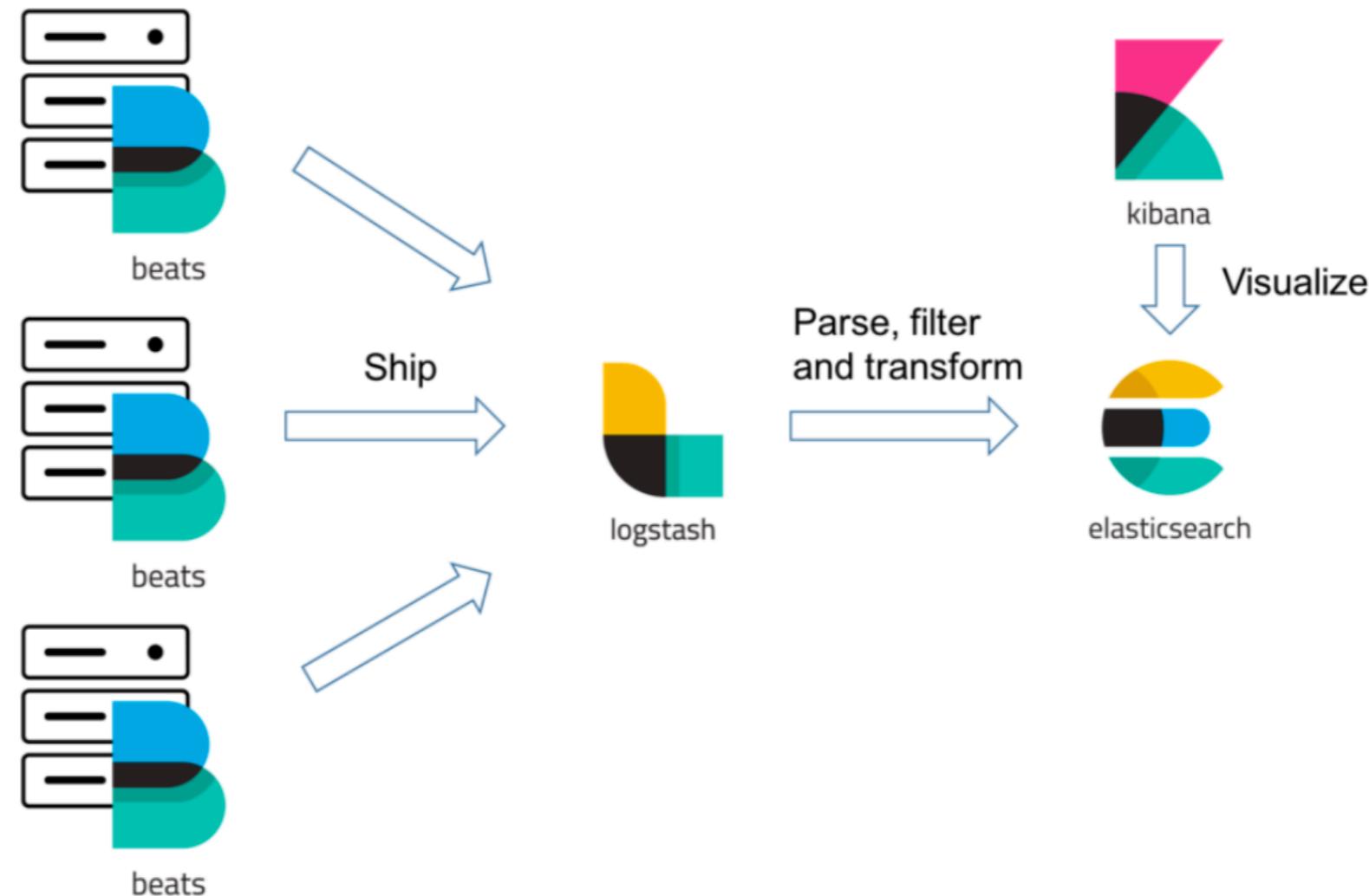
# Control traffic

## Rate limiting



# Logging

## Working with ELK and UDP



# Monitoring

## Working with Prometheus and Grafana



# KonGa

## (Kong Admin GUI)

<https://pantsel.github.io/konga/>



The screenshot shows the Konga dashboard interface. On the left is a sidebar with a dark blue background containing various navigation links. The main area has a light gray background and displays several key metrics and configuration details.

**Connections**

ACTIVE	READING	WRITING	WAITING	ACCEPTED	HANDLED
3	0	2	1	38K+	38K+

Total Requests: 40K+

**Node Info**

HostName	ip-10-0-0-242
Tag Line	Welcome to kong
Version	0.13.1
LUA Version	LuaJIT 2.1.0-beta3
Admin listen	["0.0.0.0:8001","0.0.0.0:8444 ssl"]

**Timers**

**Datastore Info**

	Reachable
DBMS	postgres
Host	kplj4fjnrbhxz.chjr89rj7wah.eu-west-1.rds.amazonaws.com
Database	kong
User	Jp2GxJlEEJlt9p
Port	5432

**Plugins**

A list of available Kong plugins:

- response-transformer
- correlation-id
- statsd
- jwt**
- cors
- basic-auth
- key-auth
- ldap-auth
- http-log
- oauth2
- hmac-auth
- acl
- datadog
- tcp-log
- ip-restriction
- request-transformer
- file-log
- bot-detection
- loggly
- request-size-limiting
- syslog
- udp-log
- response-ratelimiting
- aws-lambda
- runscope
- rate-limiting
- request-termination

<https://pantsel.github.io/konga/>



# Database integration

MySQL  
PostgreSQL  
MongoDB



# Admin GUI

KONGA

DASHBOARD

API GATEWAY

- INFO
- SERVICES
- ROUTES
- APIS (DEPRECATED)
- CONSUMERS
- PLUGINS
- UPSTREAMS
- CERTIFICATES

APPLICATION

- USERS
- CONNECTIONS
- SNAPSHOTS
- SETTINGS

**CONNECTIONS**

ACTIVE	READING	WRITING	WAITING	ACCEPTED	HANDED
3	0	2	1	38K+	38K+

Total Requests: 40K+

**NODE INFO**

HostName	ip-10-0-0-242
Tag Line	Welcome to kong
Version	0.13.1
LUA Version	LuaJIT 2.1.0-beta3
Admin listen	["0.0.0.0:8001","0.0.0.0:8444 ssl"]

**TIMERS**

**DATASTORE INFO**

Reachable	
DBMS	postgres
Host	kp1j4fj1nnfbhxz.chjr89rj7wah.eu-west-1.rds.amazonaws.com
Database	kong
User	Jp2GxJIEEJlt9p
Port	5432

**PLUGINS**

- response-transformer
- correlation-id
- statsd
- jwt**
- cors
- basic-auth
- key-auth
- ldap-auth
- http-log
- oauth2
- hmac-auth
- acl
- datadog
- tcp-log
- ip-restriction
- request-transformer
- file-log
- bot-detection
- loggly
- request-size-limiting
- syslog
- udp-log
- response-ratelimiting
- aws-lambda
- runscope
- rate-limiting
- request-termination

Hello, Arnold ▾

KONGA 0.11.0 GitHub Issues

Connected to kong.imeweb.io



# Installation

```
$docker-compose up -d mongo  
$docker-compose up -d konga
```



# Register admin user



**KONGA**

Welcome to the jungle!

Go ahead and create an administrator account.

Username

Email

Password

Confirm password

**CREATE ADMIN**

<http://127.0.0.1:1337/register>



# Welcome to Konga

**KONGA**

- DASHBOARD
- APPLICATION
- USERS
- CONNECTIONS
- SNAPSHOTS
- SETTINGS



Hello, admin ▾



## Welcome!

First of all, let's setup a connection to Kong Admin.

Select a connection type.

DEFAULT

KEY AUTH

JWT AUTH

BASIC AUTH

Konga will connect directly to Kong's admin API.  
This method is mainly suitable for demo scenarios or internal access (ex. localhost).  
Kong's admin API **should not** be publicly exposed.

Name \*

A unique connection name

Kong Admin URL \*

The URL to Kong's admin API

✓ CREATE CONNECTION



# Create new connection

DEFAULT

KEY AUTH

JWT AUTH

BASIC AUTH

Konga will connect directly to Kong's admin API.  
This method is mainly suitable for demo scenarios or internal access (ex. localhost).  
Kong's admin API **should not** be publicly exposed.

Name \*

kong

Kong Admin URL \*

http://kong:8001/

✓ UPDATE CONNECTION



# Welcome

## KONGA

- DASHBOARD
- API GATEWAY
- INFO
- SERVICES
- ROUTES
- CONSUMERS
- PLUGINS
- UPSTREAMS
- CERTIFICATES
- APPLICATION
- USERS
- CONNECTIONS
- SNAPSHOTS
- SETTINGS

Total Requests: 1K+

ACTIVE	READING	WRITING	WAITING	ACCEPTED	HANDLED
2	0	1	1	986	986

**NODE INFO**

HostName	22d7384ce7fc
Tag Line	Welcome to kong
Version	2.0.4
LUA Version	LuaJIT 2.1.0-beta3
Admin listen	["0.0.0.0:8001"]

**TIMERS**

**DATASTORE INFO**

DBMS	postgres
Host	kong-database
Database	kong
User	kong
Port	5432

**PLUGINS**

correlation-id pre-function cors ldap-auth loggly hmac-auth zipkin request-size-limiting azure-functions request-transformer oauth2 response-transformer ip-restriction statsd jwt proxy-cache basic-auth key-auth http-log datadog tcp-log rate-limiting post-function prometheus acl syslog file-log acme udp-log response-ratelimiting aws-lambda session bot-detection request-termination

KONGA 0.14.9 GitHub Issues Support the project Connected to kong

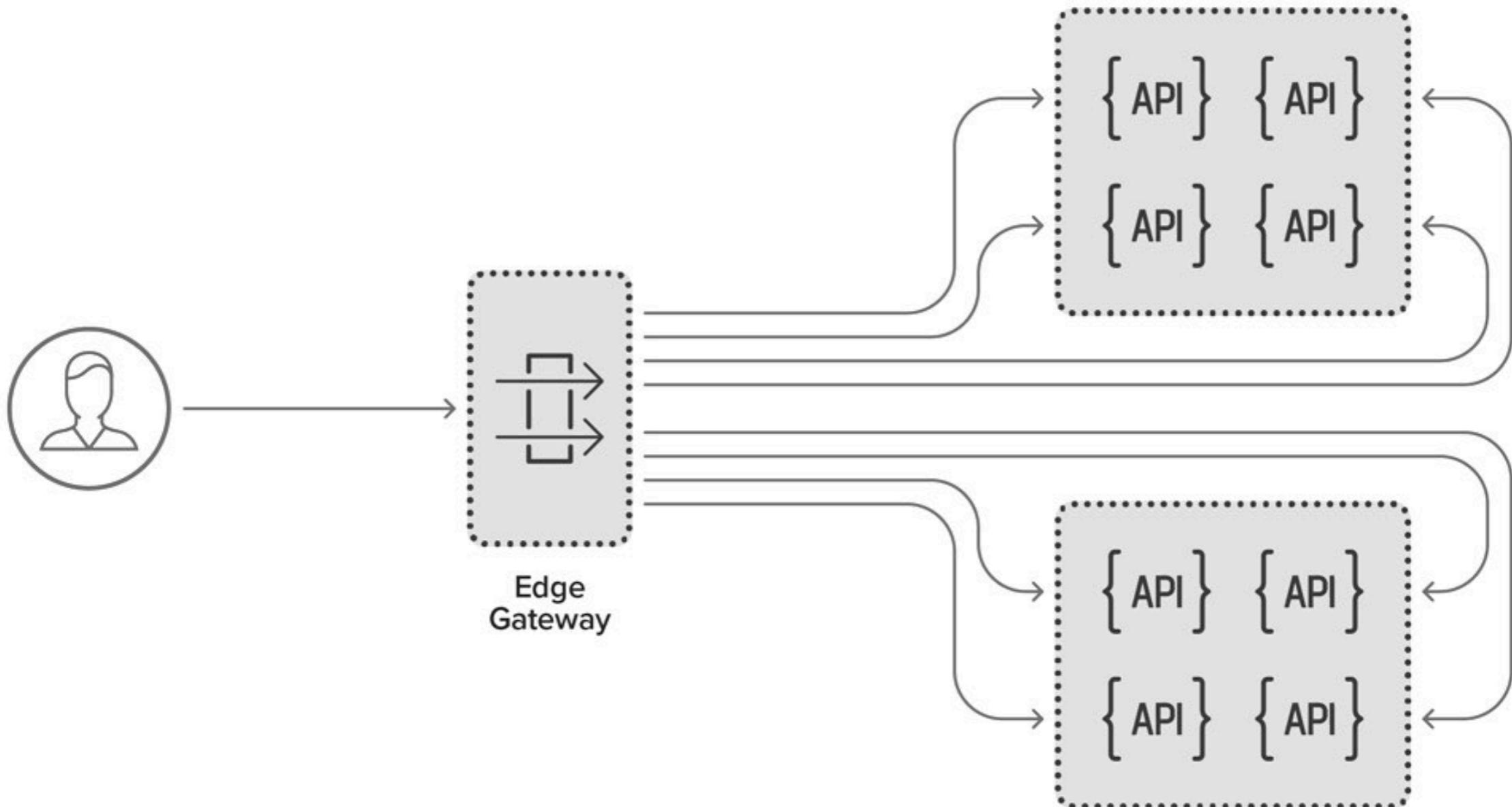
API Gateway with Kong  
© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

96

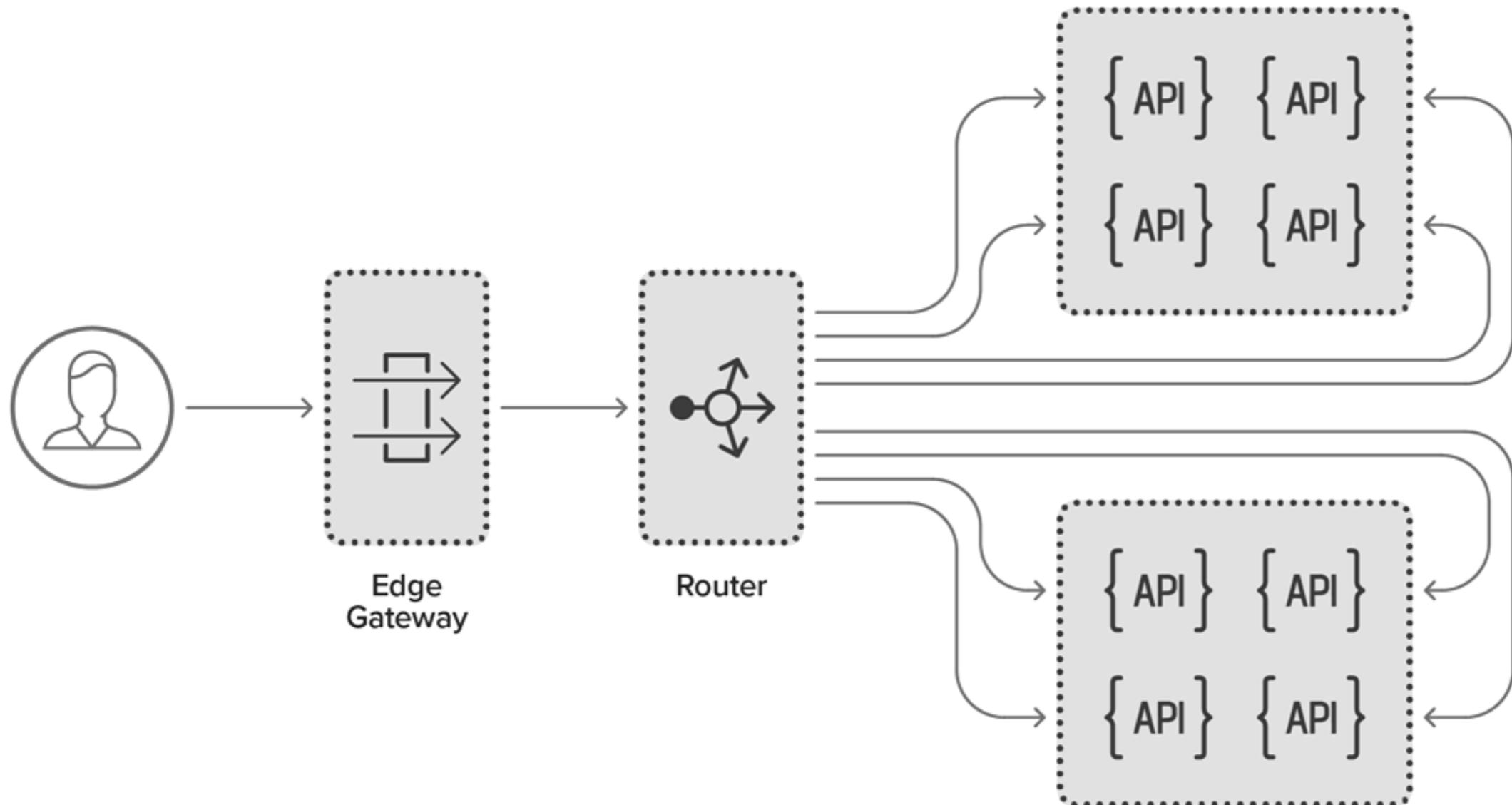
# Deployment patterns



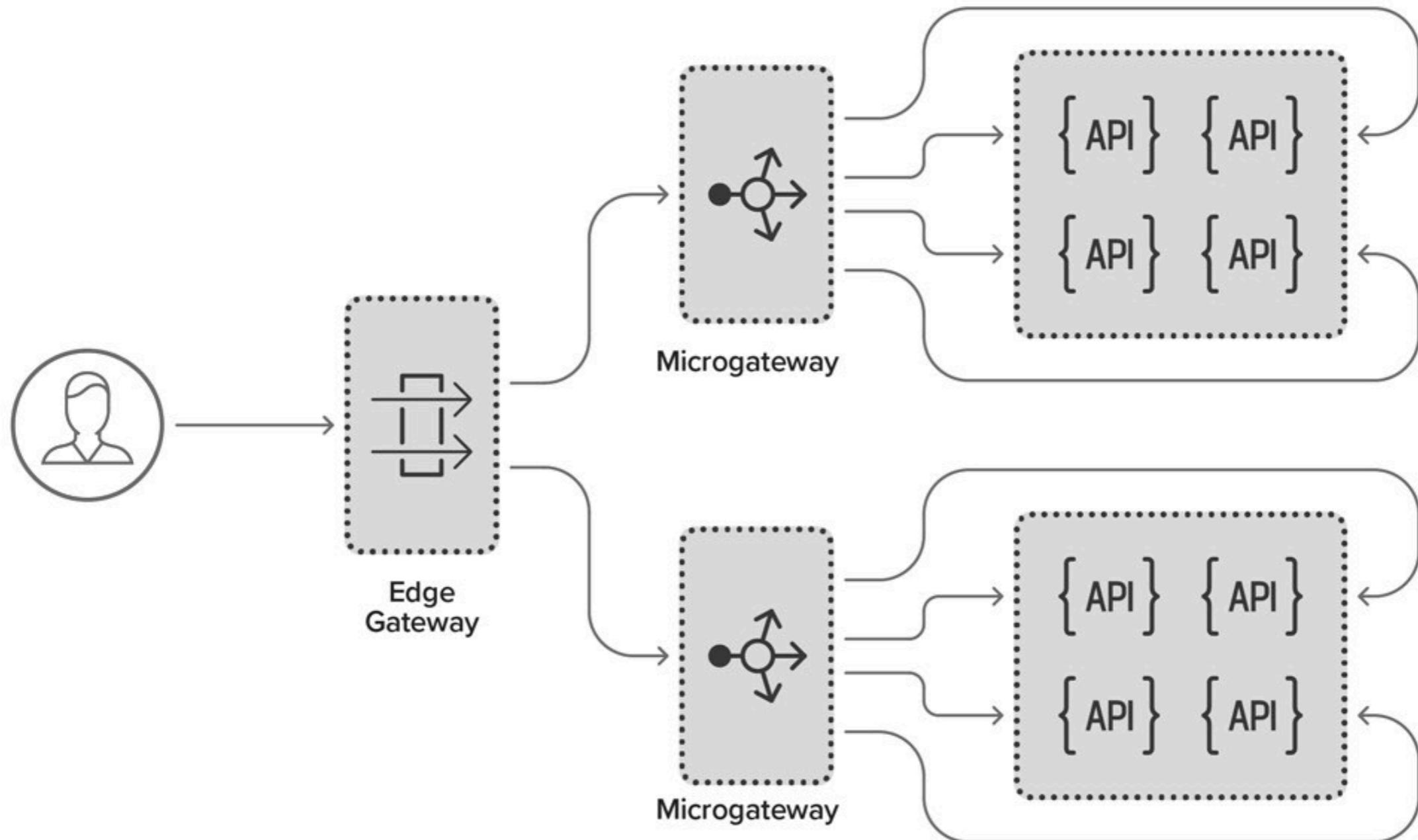
# Edge Gateway



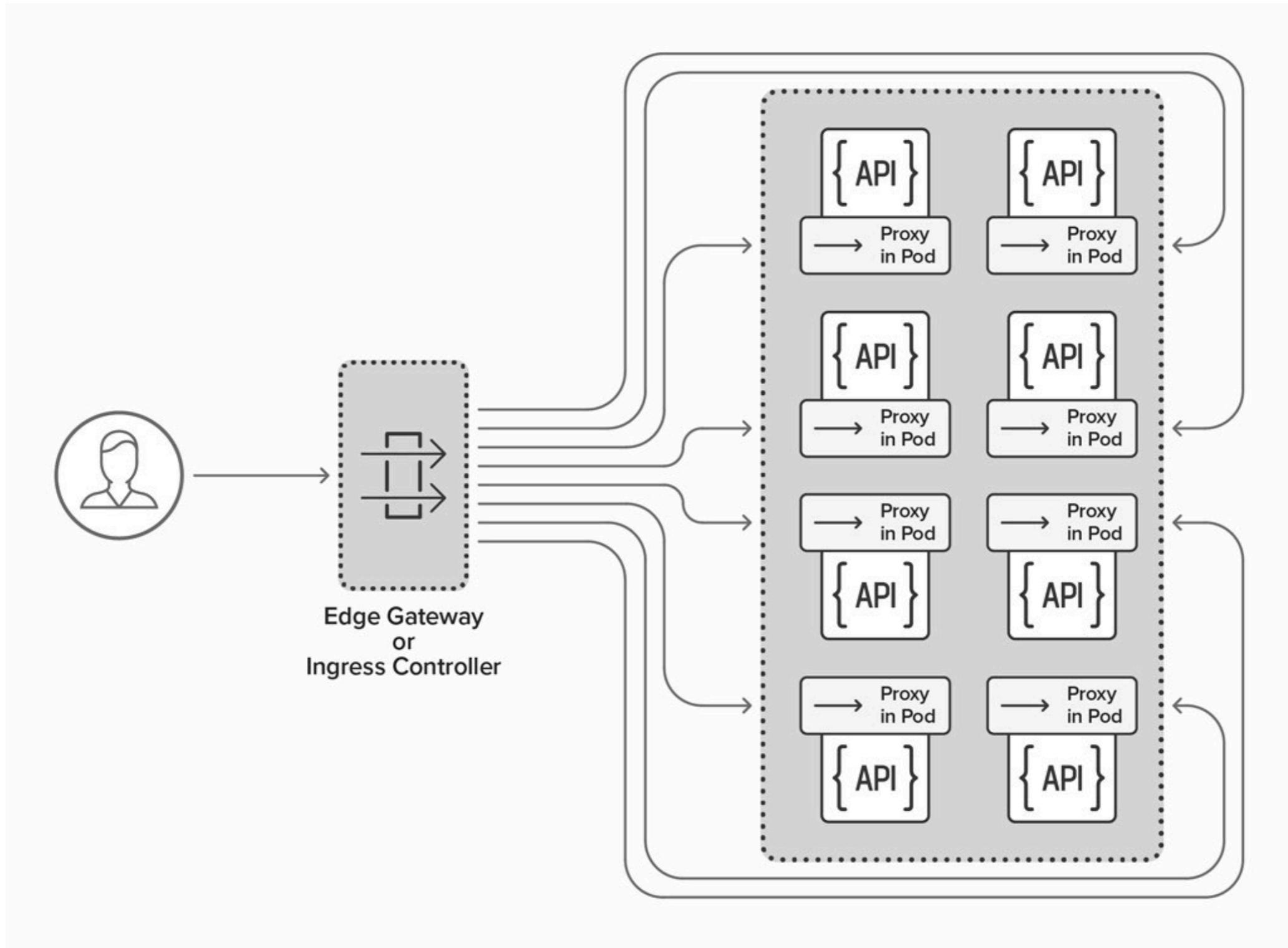
# Two-Tiers Gateway



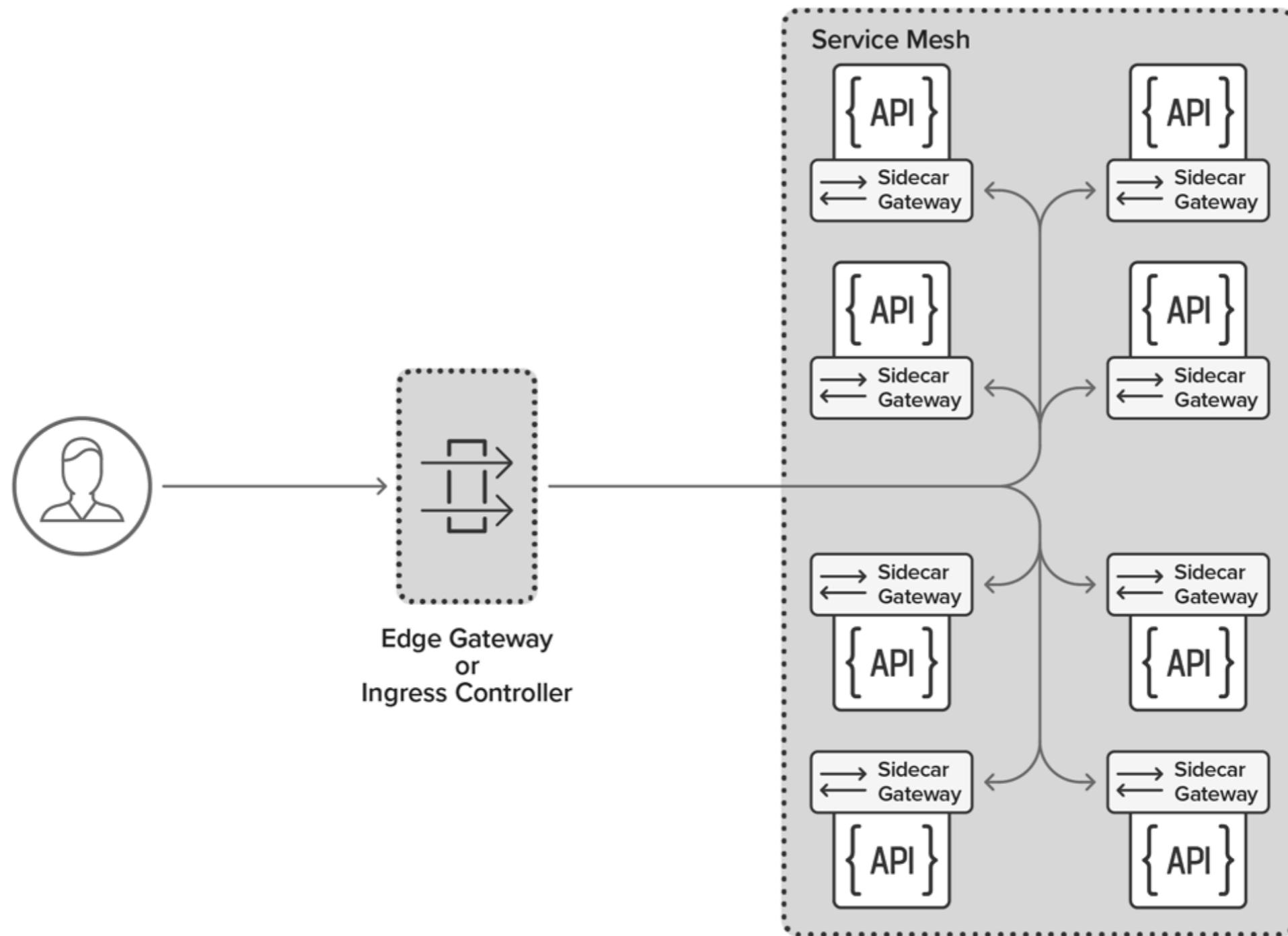
# Micro Gateway



# Per Pod Gateway



# Sidecar



# **Use the right tool for the right job**



# Learn Lua to develop Kong plugins



# Q/A

