



A P I s A R

Episodio VI : #APIsARe06

- Preguntas online vía
  - Twitter: **@APIsARgentina** o **#APIsARe06**
  - Facebook: **APIsArgentina**
- ¡¡¡Siempre buscando oradores!!!
- Gracias **Redbee** - <http://redbee.io>



API Gateways & Microservicios con **Kong**

- **Microservicios**
- **API Gateways**
- **¿Que es Kong?**
- **Escenarios**
- **Demo**

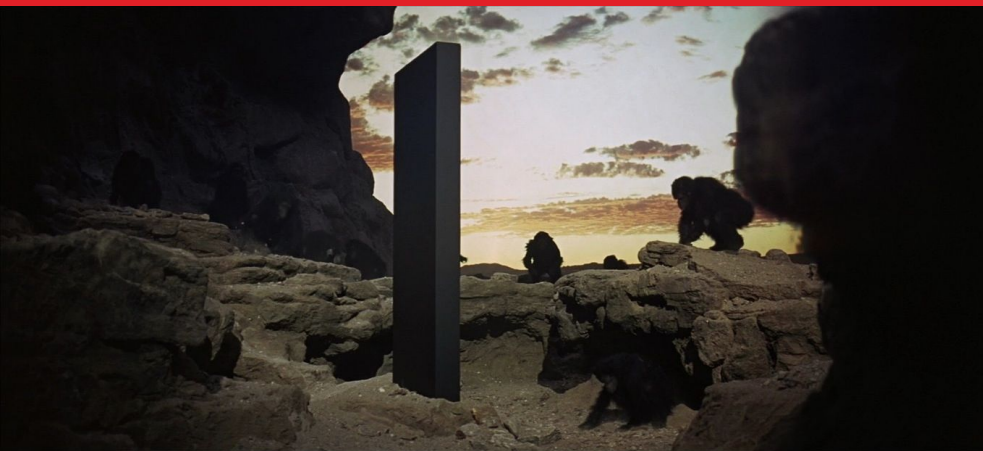


A P I B A R

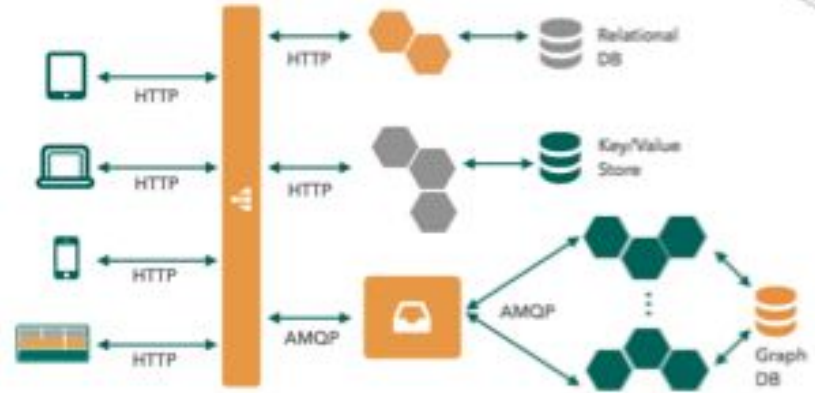
[www.gettingstartedwithapi.bar](http://www.gettingstartedwithapi.bar)

[@APIBARArgentina](https://twitter.com/APIBARArgentina)

# Partiendo el monolito



# Partiendo el monolito



# Arquitecturas monolíticas

- Complejas
- Escalamiento ineficiente
- Fuerte acoplamiento deriva en despliegues poco frecuentes, poca agilidad para el cambio
- Difícil escalar el desarrollo (debido a la gran base de código y el impacto de los cambios)
- El tamaño puede resultar intimidante para los desarrolladores
- Genera un compromiso hacia el stack de desarrollo: Difícil adoptar nuevas tecnologías

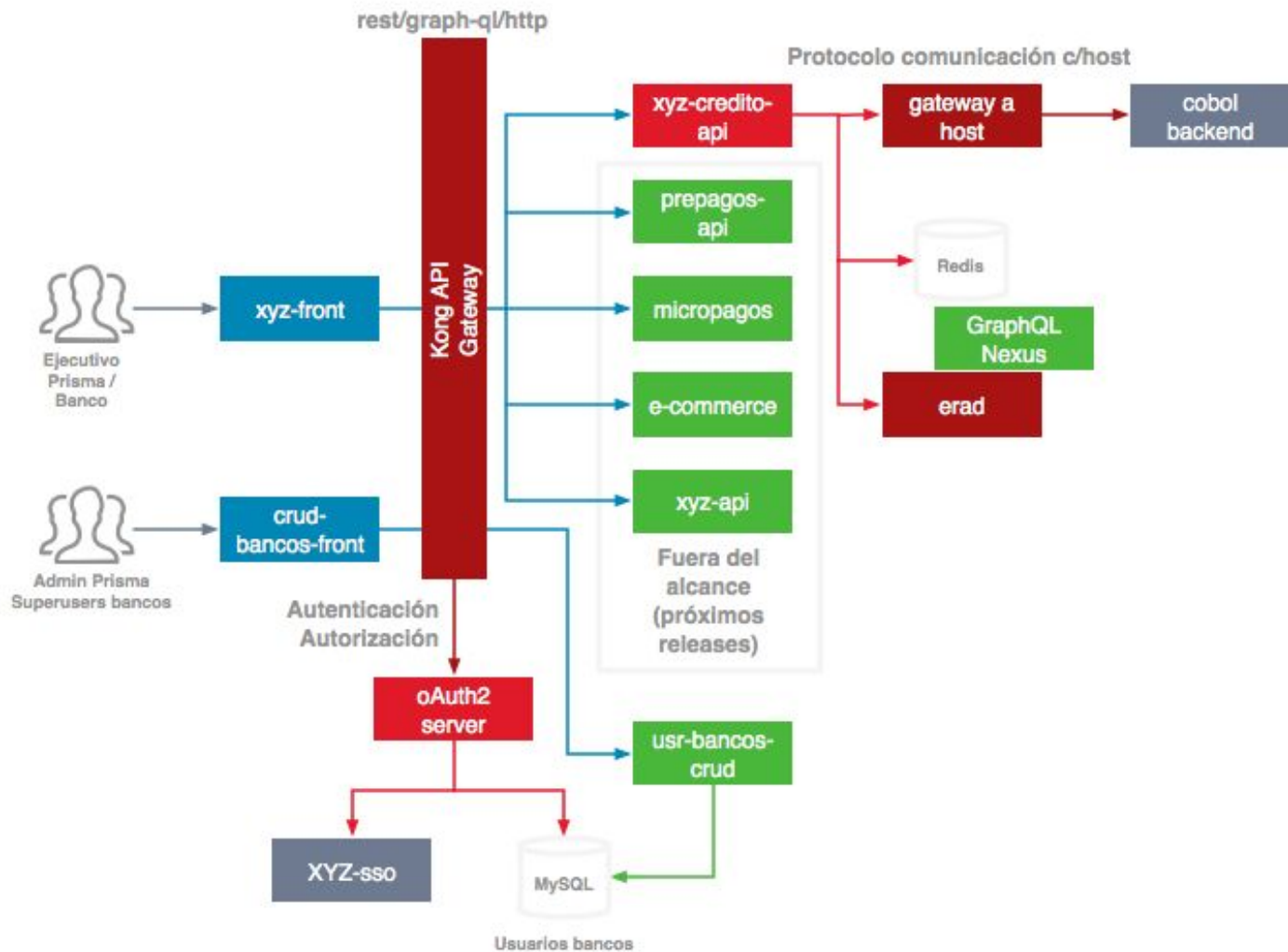
- Todos esos puntos hacen **más lento el avance**, en contraposición con la demanda del negocio de **innovar**.

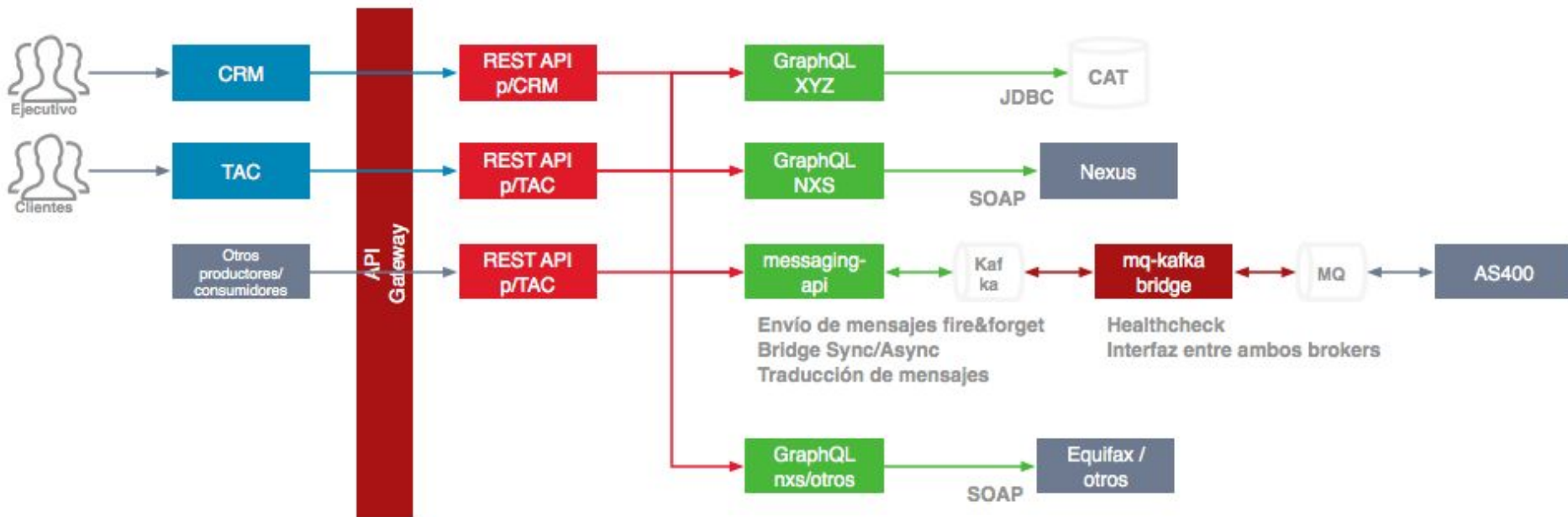
# Microservicios

- Simples (o al menos más simples)
- Modularidad basada en servicios
- Ciclos de cambios de cada componente desacoplados. Permite despliegues frecuentes
- Escalamiento eficiente
- Permite escalar el desarrollo (equipos chicos concentrándose en módulos/microservicios: regla de 2 pizzas por equipo)
- Divide y conquistarás: componentes reducidos resultan menos intimidantes
- Elimina el compromiso fuerte con un stack tecnológico. Cada módulo puede tener su stack

Prima acelerar el **ritmo de cambios** mediante la **eficiencia**







## F5 - Load Balancer

DC/OS



psql

Nueva API Rest  
para comercios

kong api  
gateway

Autenticación  
mediante access  
tokens  
SSL termination  
Rate limiting (esto  
último x ahora desactivado)

resttx

Servicio de replicación  
de datos

kafka

API rest interna

coretx

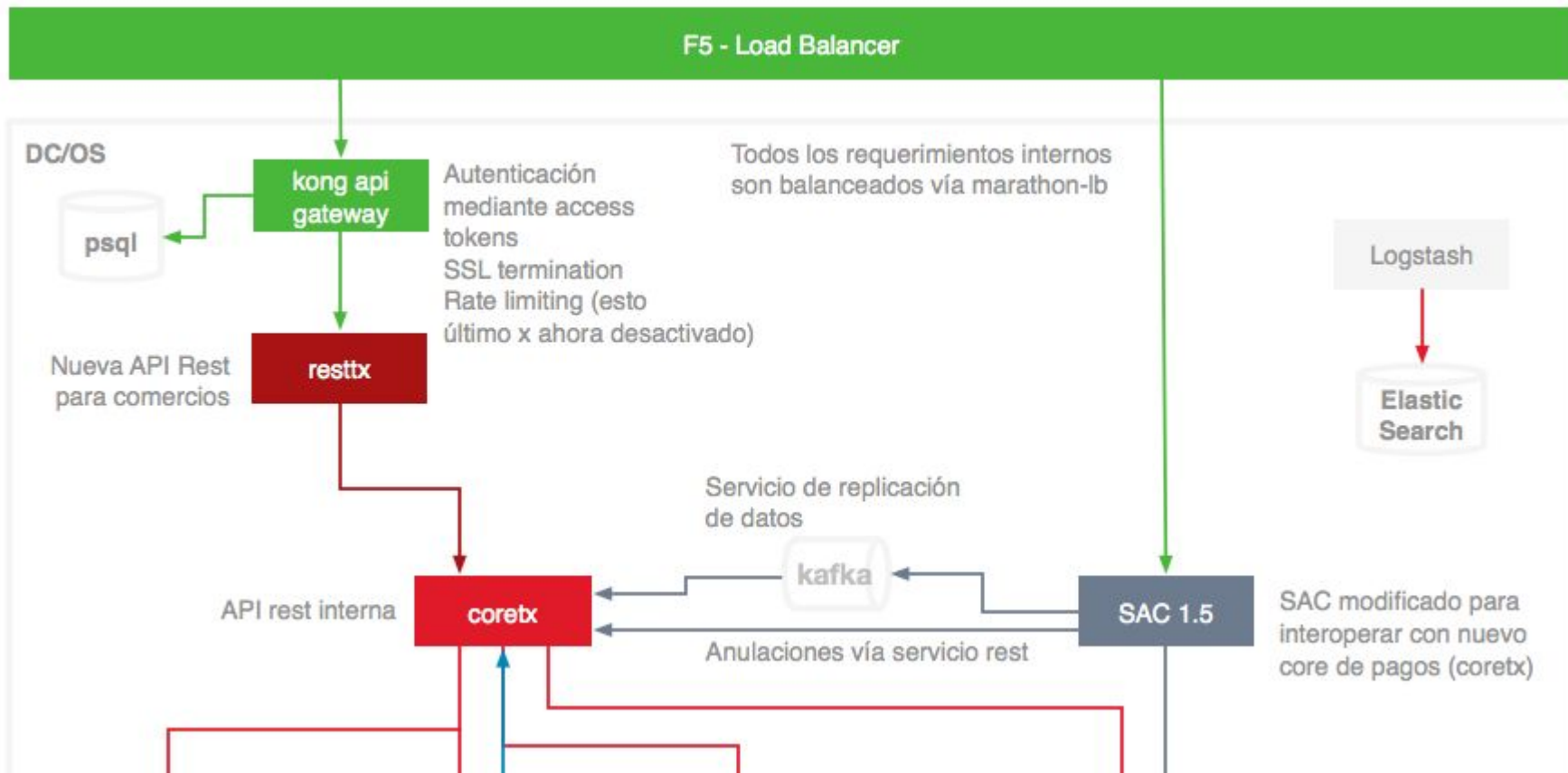
Anulaciones vía servicio rest

SAC 1.5

SAC modificado para  
interoperar con nuevo  
core de pagos (coretx)

Logstash

Elastic  
Search



#### Developer Center

API Console

API Docs

Portal comunidad

Dev manager  
(Alta, claves)

Dev onboarding

SDKs

#### API Gateway

Autenticación

Rate Limiting

Security Policies

Autorización

Caching

Analytics

Logging & Tracing

Transformaciones

Billing

#### Soporte

COE

Diseño

Soporte

Operaciones

Homologación

# API Gateway



A P I 8 A R

[www.getup.com/API8AR](https://www.getup.com/API8AR)

[@API8ARCENTOA](#)



**ESBs**  
Integration/exposure-centric  
Focus on stability  
Systems of record integration  
Cost reduction enabler

**API MANAGEMENT**  
Consumption-centric  
Focus on agility  
Variability & scale of access  
Top-line growth enabler



Developer Mgmt

Scale

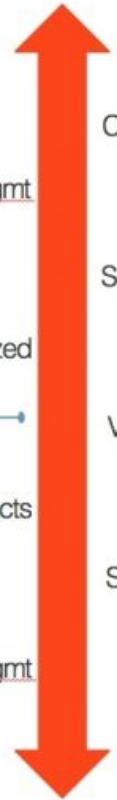
Decentralized

Visibility

API Products

Security

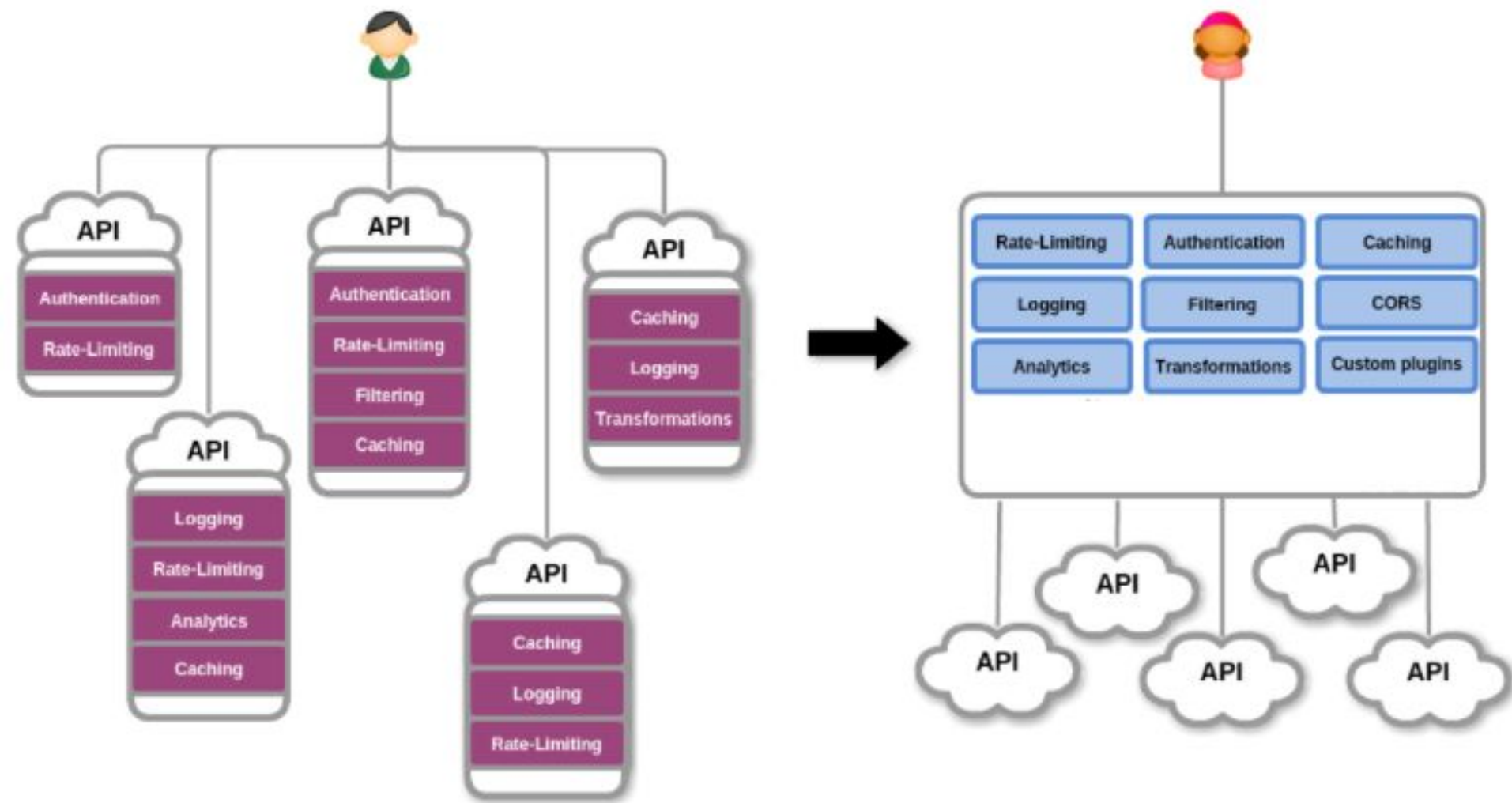
Proxy Mgmt



Cloud Native



**SYSTEMS OF RECORD**



# Kong

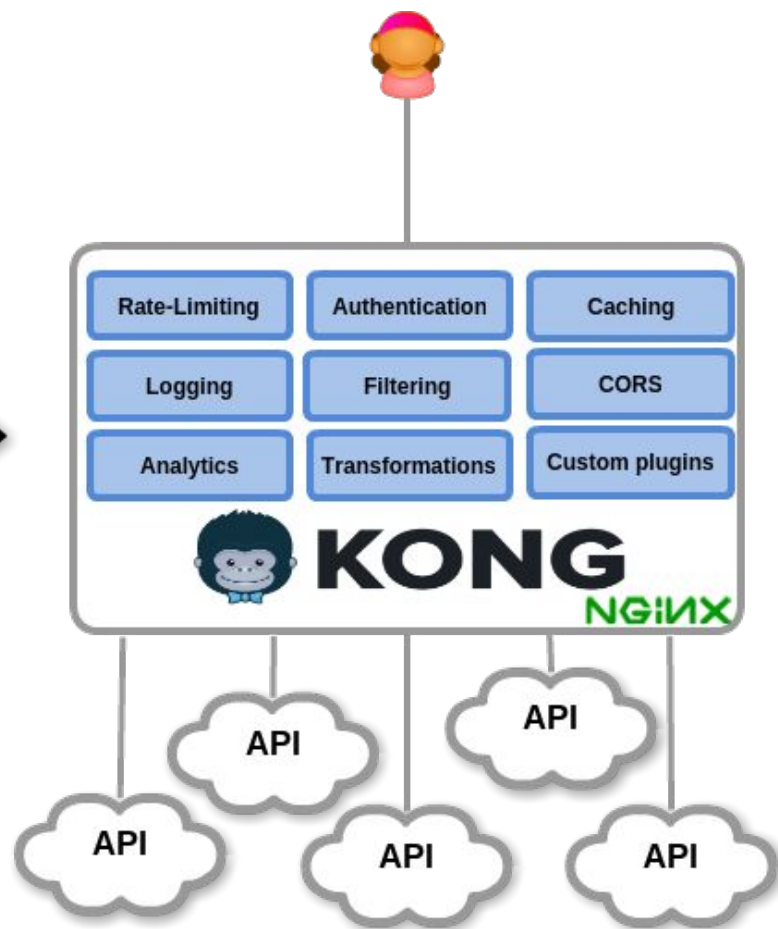
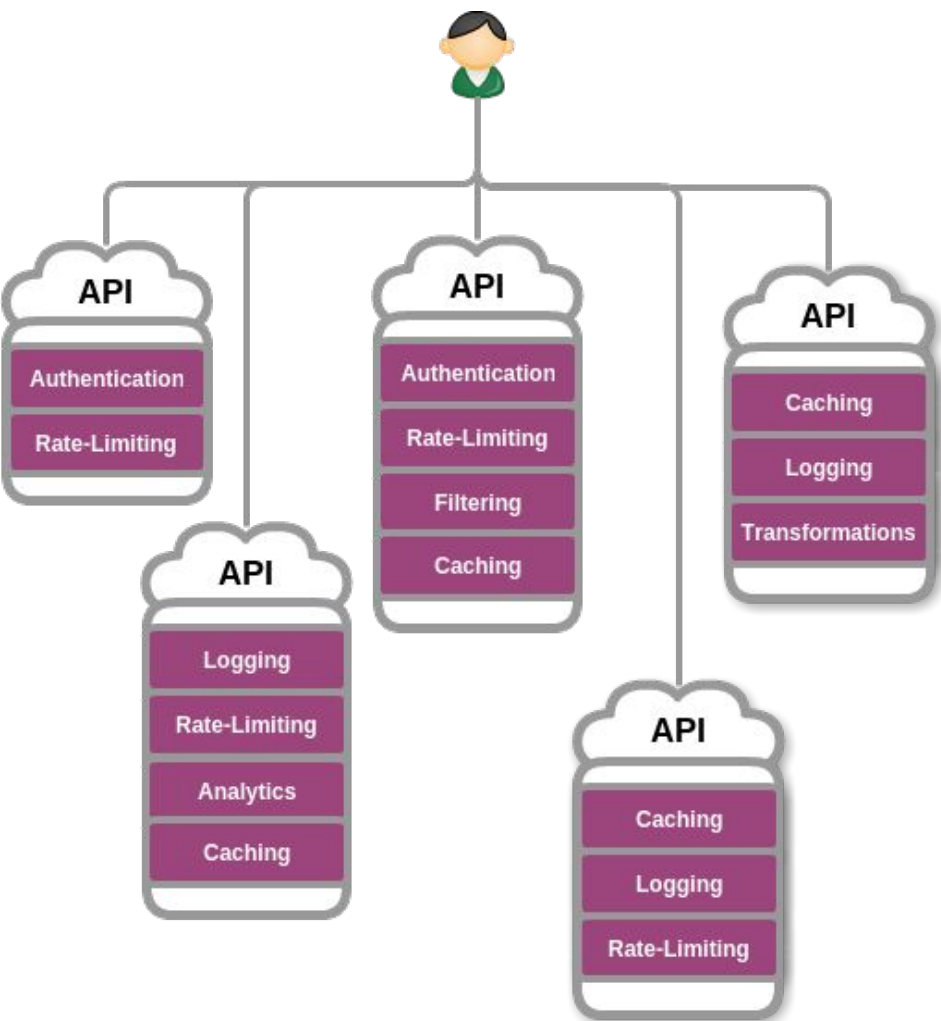


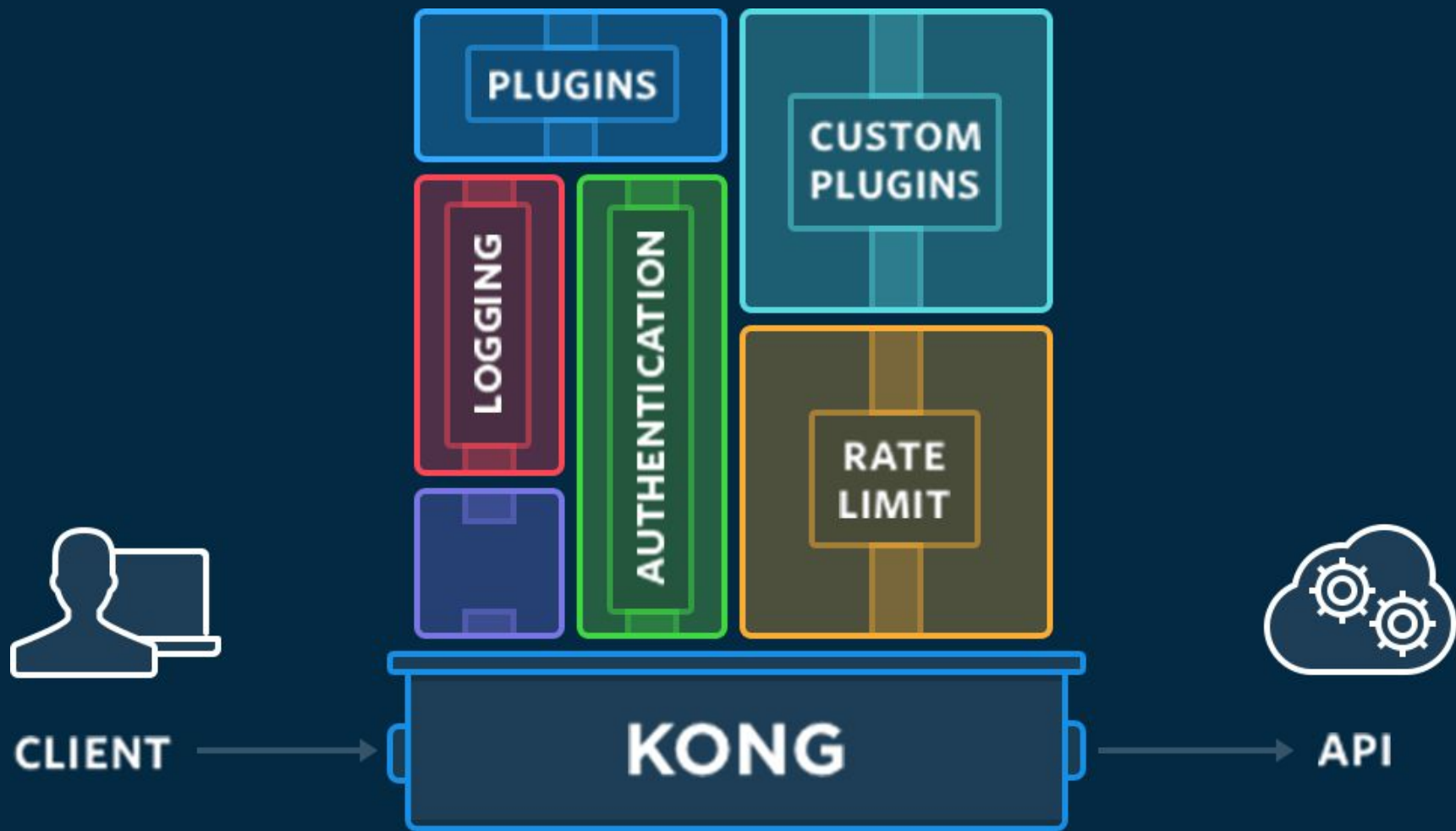
A P I B A R

[WWW.CHEETUP.COM/APIBAR](http://WWW.CHEETUP.COM/APIBAR)

[@APIBARCENTOA](https://twitter.com/APIBARCENTOA)







# Arquitectura

## API REST

Extensible mediante plugins  
Usada para automatización

Cassandra o Postgres

Redis requerido por algunos  
plugins

Cluster single o multi datacenter

Core de Kong

Maneja toda la operatoria de  
bajo nivel

Super sólido y conocido

API de administración

Plugins

Clustering & Datastore

OpenResty

NGINX

LUA como lenguaje de desarrollo  
Interceptan el ciclo de request/response  
Pueden integrarse con servicios de 3ras  
partes

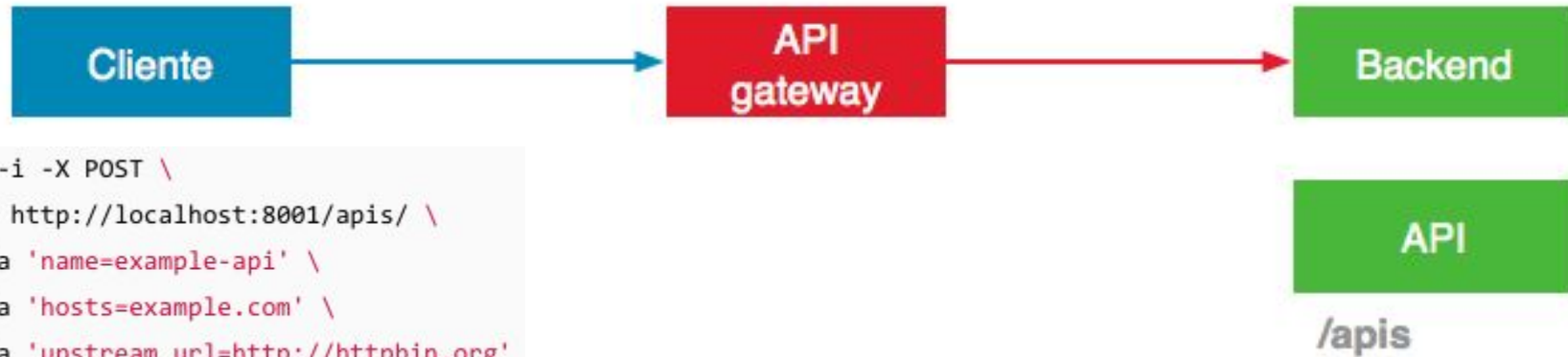
Engine de Kong  
Provee los hooks para la intercepción  
del ciclo req/resp  
Extiende NGINX

```
curl url -o teamWork
```

# API Rest

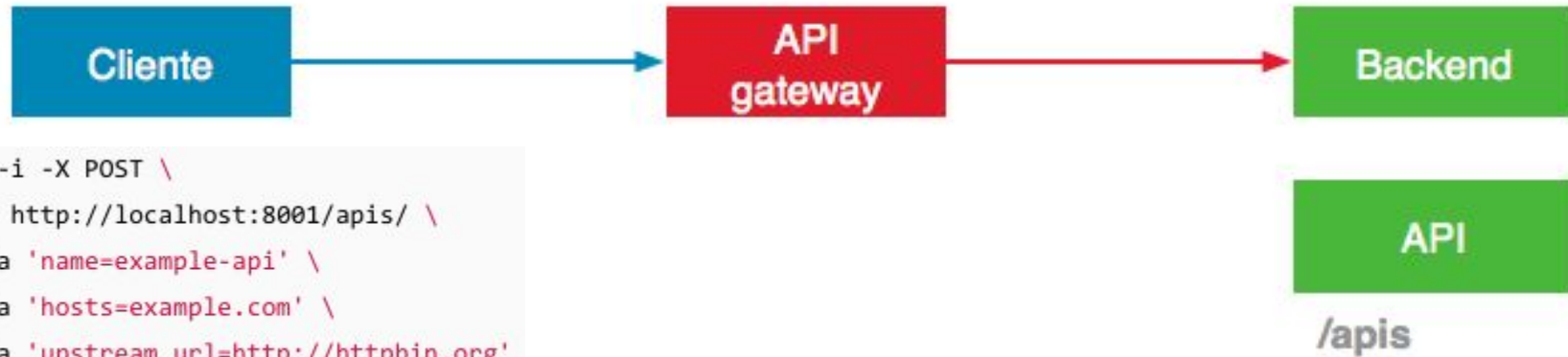


# Agregar una API



```
$ curl -i -X POST \  
--url http://localhost:8001/apis/ \  
--data 'name=example-api' \  
--data 'hosts=example.com' \  
--data 'upstream_url=http://httpbin.org'
```

# Agregar una API



```
$ curl -i -X POST \
--url http://localhost:8001/apis/ \
--data 'name=example-api' \
--data 'hosts=example.com' \
--data 'upstream_url=http://httpbin.org'
```

# Agregar una API

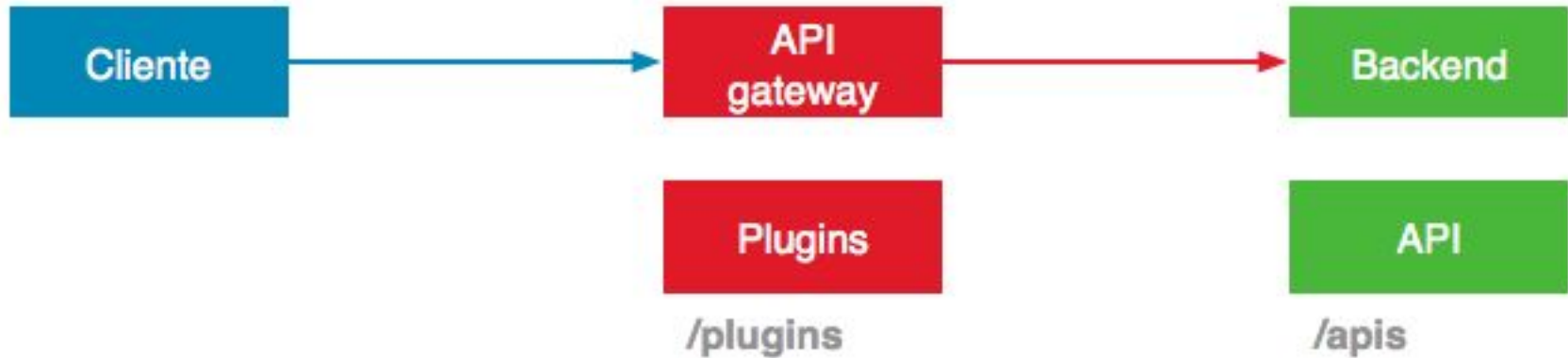
```
$ curl -i -X POST \  
  --url http://localhost:8001/apis/ \  
  --data 'name=example-api' \  
  --data 'hosts=example.com' \  
  --data 'upstream_url=http://httpbin.org'
```

```
HTTP/1.1 201 Created  
Content-Type: application/json  
Connection: keep-alive
```

```
{  
  "created_at": 1488830759000,  
  "hosts": [  
    "example.com"  
  ],  
  "http_if_terminated": true,  
  "https_only": false,  
  "id": "6378122c-a0a1-438d-a5c6-efabae9fb969",  
  "name": "example-api",  
  "preserve_host": false,  
  "retries": 5,  
  "strip_uri": true,  
  "upstream_connect_timeout": 60000,  
  "upstream_read_timeout": 60000,  
  "upstream_send_timeout": 60000,  
  "upstream_url": "http://httpbin.org"  
}
```



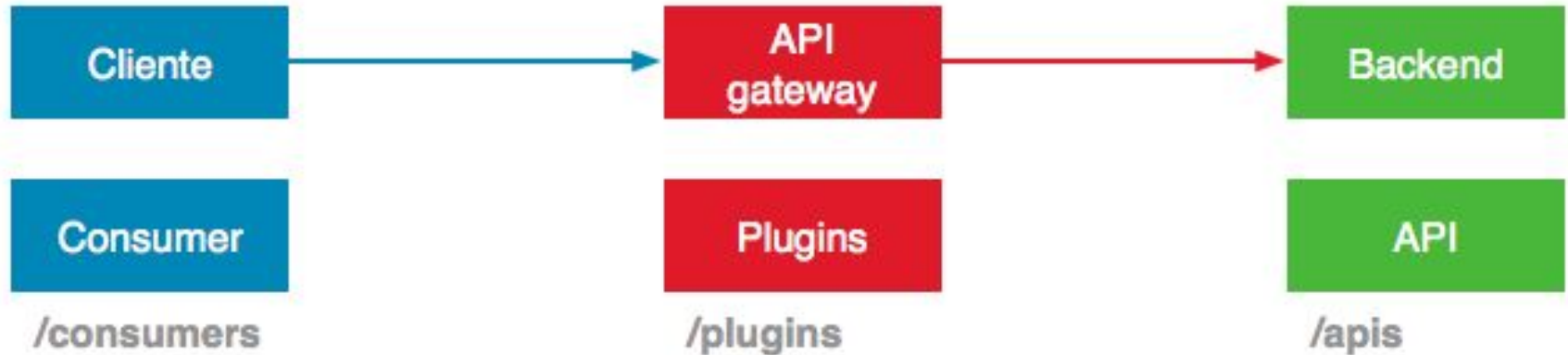
# Configurar un plugin



```
$ curl -i -X POST \  
  --url http://localhost:8001/apis/example-api/plugins/ \  
  --data 'name=key-auth'
```

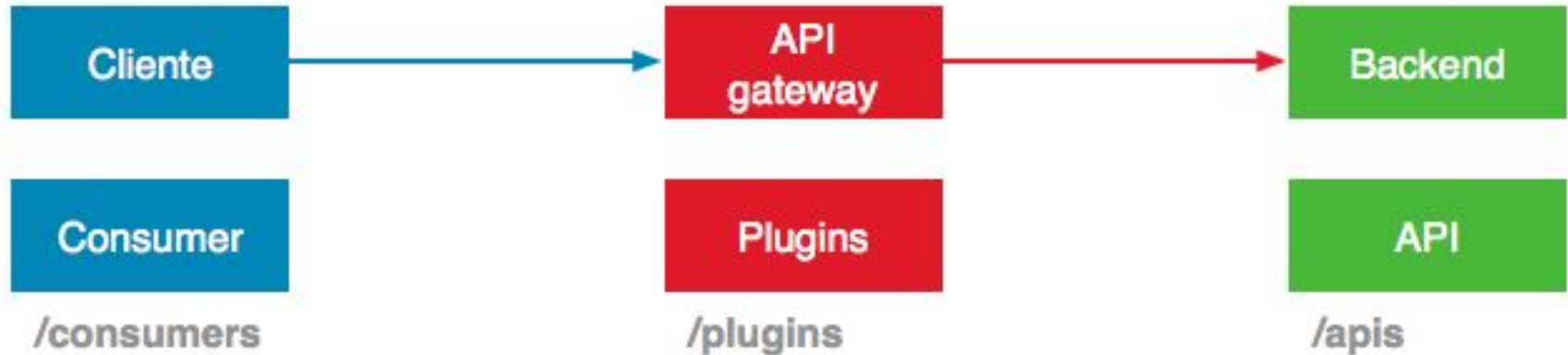


# Configurar un consumidor



```
$ curl -i -X POST \  
  --url http://localhost:8001/consumers/ \  
  --data "username=Jason"
```

# Configurar una clave de acceso para el consumidor



```
$ curl -i -X POST \  
--url http://localhost:8001/consumers/Jason/key-auth/ \  
--data 'key=ENTER_KEY_HERE'
```

# Consolas de administración

## Managing Consumers

The first screenshot shows the 'Consumers' list in the Kong Admin UI. It features a table with columns: Name, External ID, Created, and Status. There are two entries, both with a status of 'OK'. The second screenshot shows the 'Edit Consumer' form, which includes fields for Name, External ID, and a checkbox for 'Enabled'. A green success message is displayed at the top of the form.

## Managin Plugins

The first screenshot shows the 'Plugins' list in the Kong Admin UI. It features a table with columns: Name, API, Consumer, Status, and Created. There are four entries, all with a status of 'OK'. The second screenshot shows the 'Add plugin' form, which includes fields for Name, API, Consumer, and a checkbox for 'Enabled'. A green success message is displayed at the top of the form.

The screenshot shows the Kong Admin UI dashboard. The left sidebar contains a navigation menu with options: DASHBOARD, INFO, CLUSTER, API, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The main content area displays several charts and tables. The 'CONNECTIONS' chart shows a bar graph for Accepted, Active, Handled, Reading, Writing, and Total Requests. The 'TIMERS' chart shows a bar graph for Pending and Running. The 'DATABASE' section lists various database-related items. The 'GATEWAY INFO' section displays details about the gateway, including HostName, Tag Line, Version, and Admin listen. The 'DATABASE INFO' section displays details about the database, including DBMS, Host, Database, User, and Port. The 'PLUGINS' section displays a list of installed plugins.

# Autenticación



## Basic Authentication

Add Basic Authentication to your APIs



## Key Authentication

Add a key authentication to your APIs



## OpenID Connect 1.0 RP

Integrate Kong with a third-party OpenID Connect 1.0 Provider



## OAuth 2.0 Authentication

Add an OAuth 2.0 authentication to your APIs



## OAuth 2.0 Introspection

Integrate Kong with a third-party OAuth 2.0 Authorization Server



## HMAC Authentication

Add HMAC Authentication to your APIs



## JWT

Verify and authenticate JSON Web Tokens



## LDAP Authentication

Integrate Kong with a LDAP server

# Seguridad



## ACL

Control which consumers can access APIs



## CORS

Allow developers to make requests from the browser



## Dynamic SSL

Add an SSL certificate for an underlying service



## IP Restriction

Whitelist or blacklist IPs that can make requests



## Bot Detection

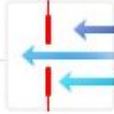
Detects and blocks bots or custom clients

# Control de tráfico



## Rate Limiting

Rate-limit how many HTTP requests a developer can make



## Response Rate Limiting

Rate-Limiting based on a custom response header value



## Request Size Limiting

Block requests with bodies greater than a specific size



## Request Termination

Terminates all requests with a specific response

```
curl url -o teamWork
```

# Serverless



## AWS Lambda

Invoke and manage AWS Lambda functions from Kong



## OpenWhisk

Invoke and manage OpenWhisk actions from Kong

# Analytics & Monitoreo



**Galileo**

Business Intelligence Platform for  
APIs



**Datadog**

Visualize API metrics on Datadog



**Runscope**

API Performance Testing and  
Monitoring



# Manoseo del requerimiento y respuesta (AKA transformaciones)



## Request Transformer

Modify the request before hitting  
the upstream server



## Response Transformer

Modify the upstream response  
before returning it to the client



## Correlation ID

Correlate requests and responses  
using a unique ID

# Logging



## TCP

Send request and response logs to a TCP server



## UDP

Send request and response logs to an UDP server



## HTTP

Send request and response logs to an HTTP server



## File

Append request and response data to a log file on disk



## Syslog

Send request and response logs to Syslog



## StatsD

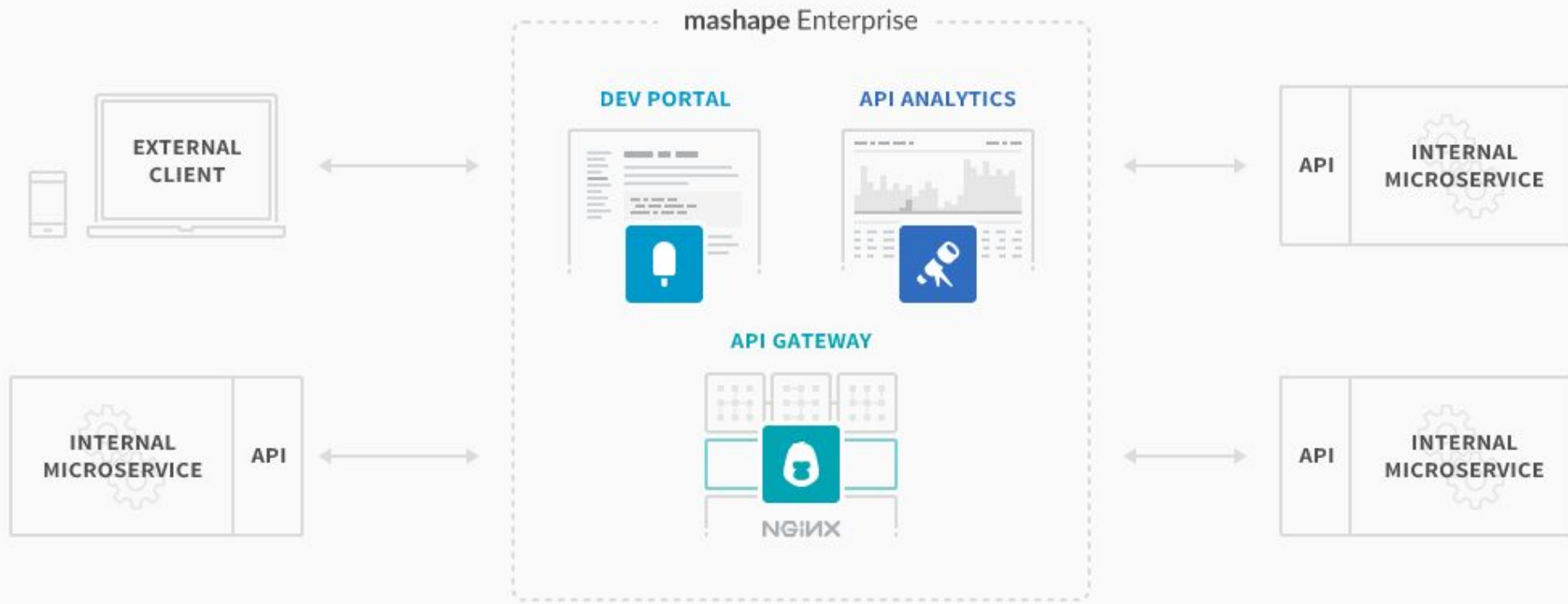
Send request and response logs to StatsD



## Loggly

Send request and response logs to Loggly

# Mashape Enterprise



# Escenarios

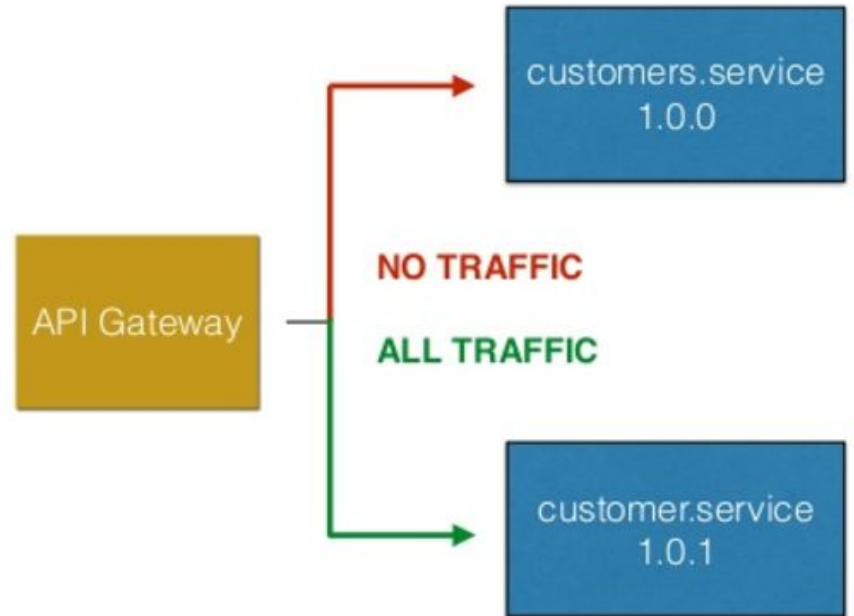
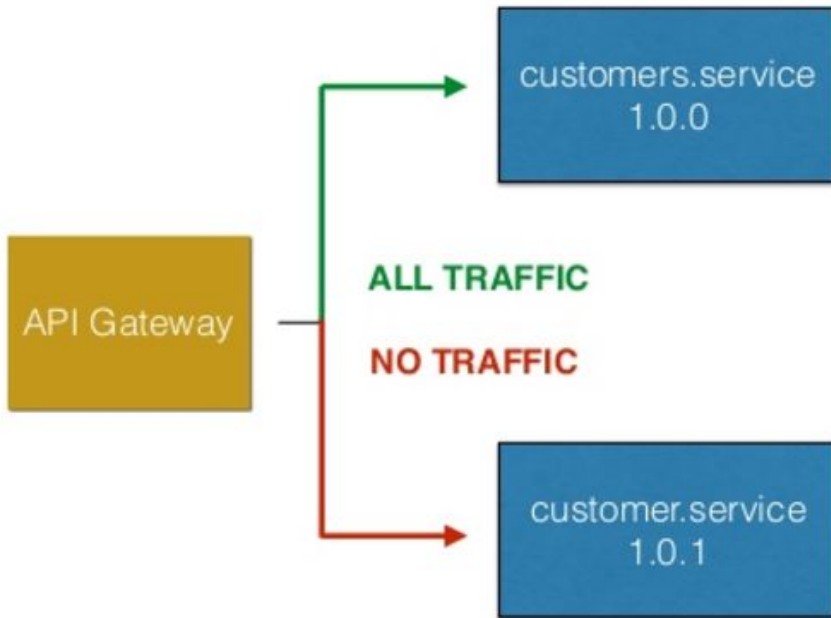


A P I S A R

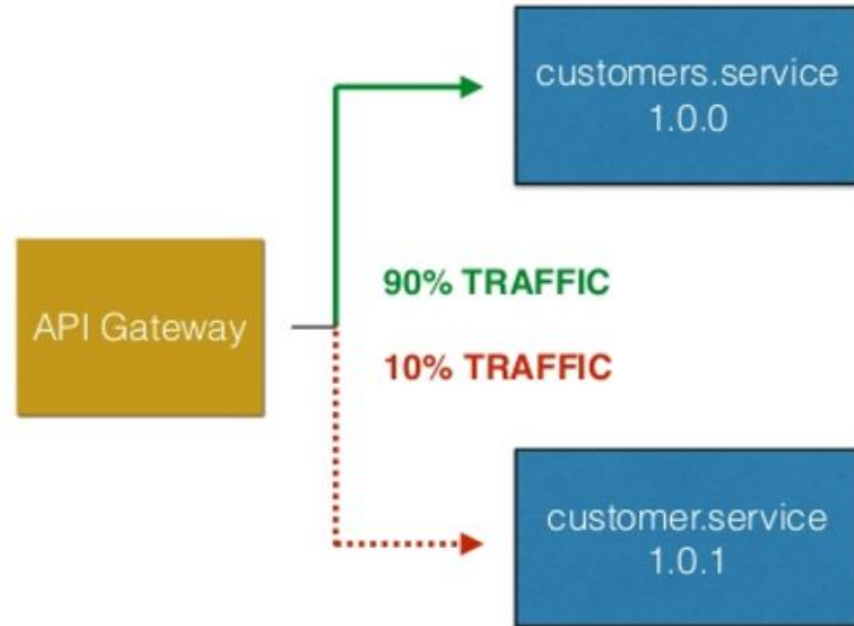
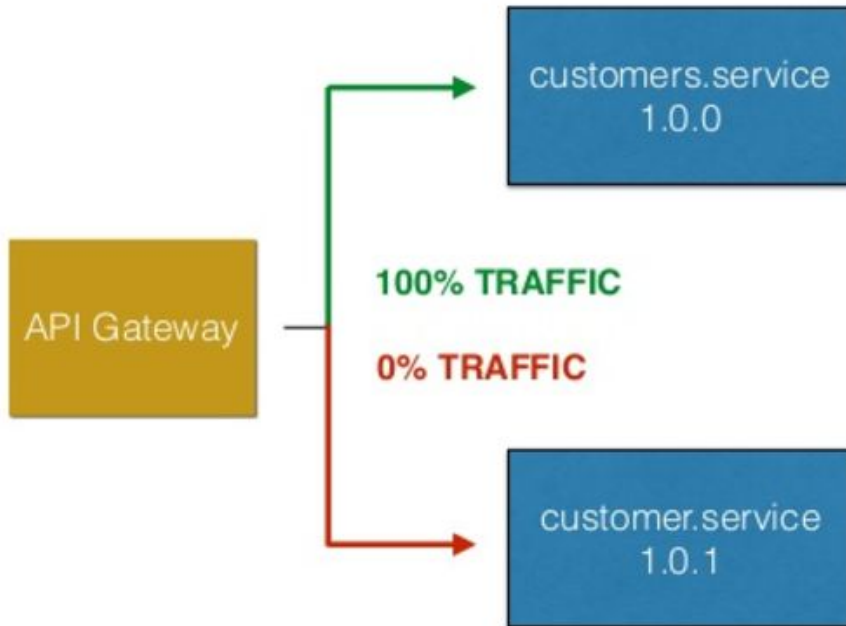
[www.cheetup.com/apisar](http://www.cheetup.com/apisar)

[@APISARArgentina](https://twitter.com/APISARArgentina)

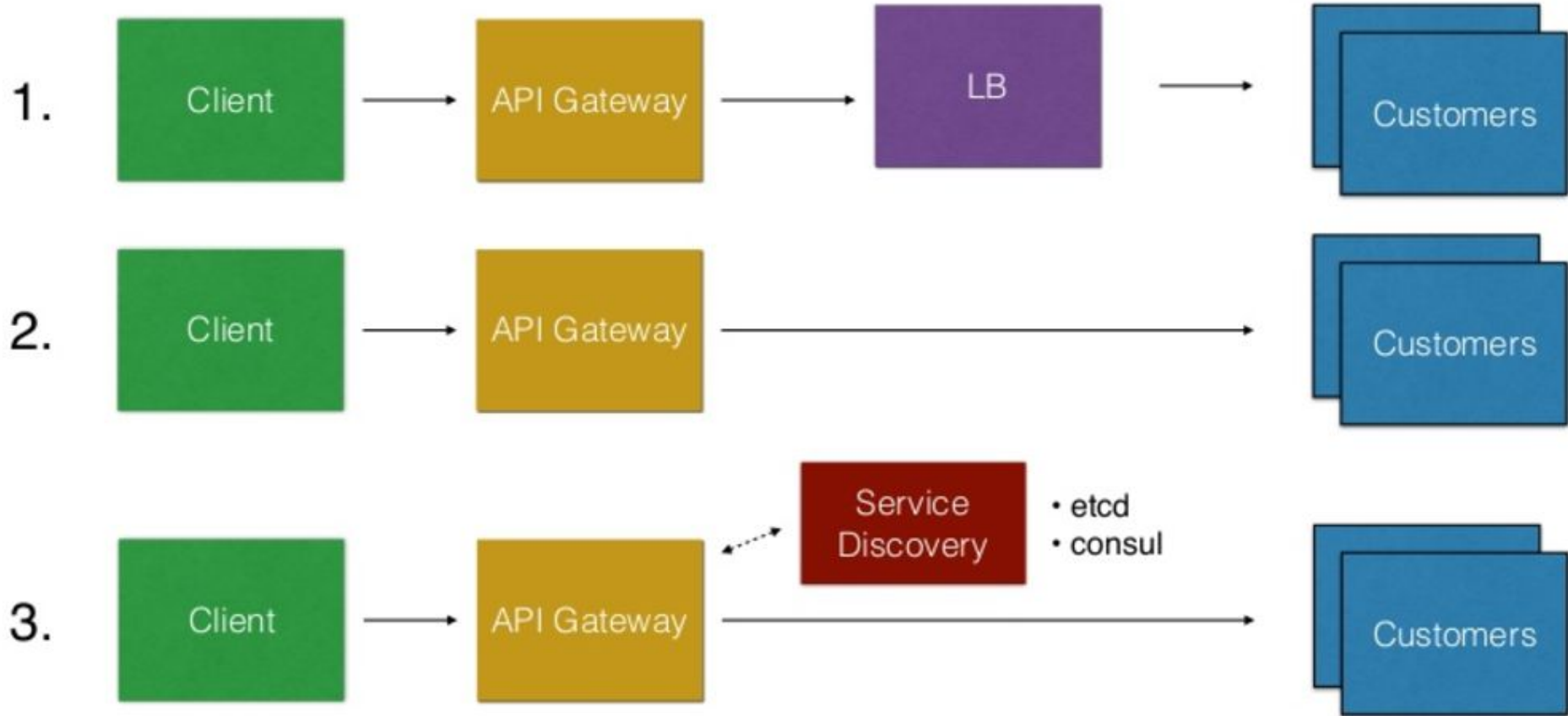
# Despliegues Blue/Green



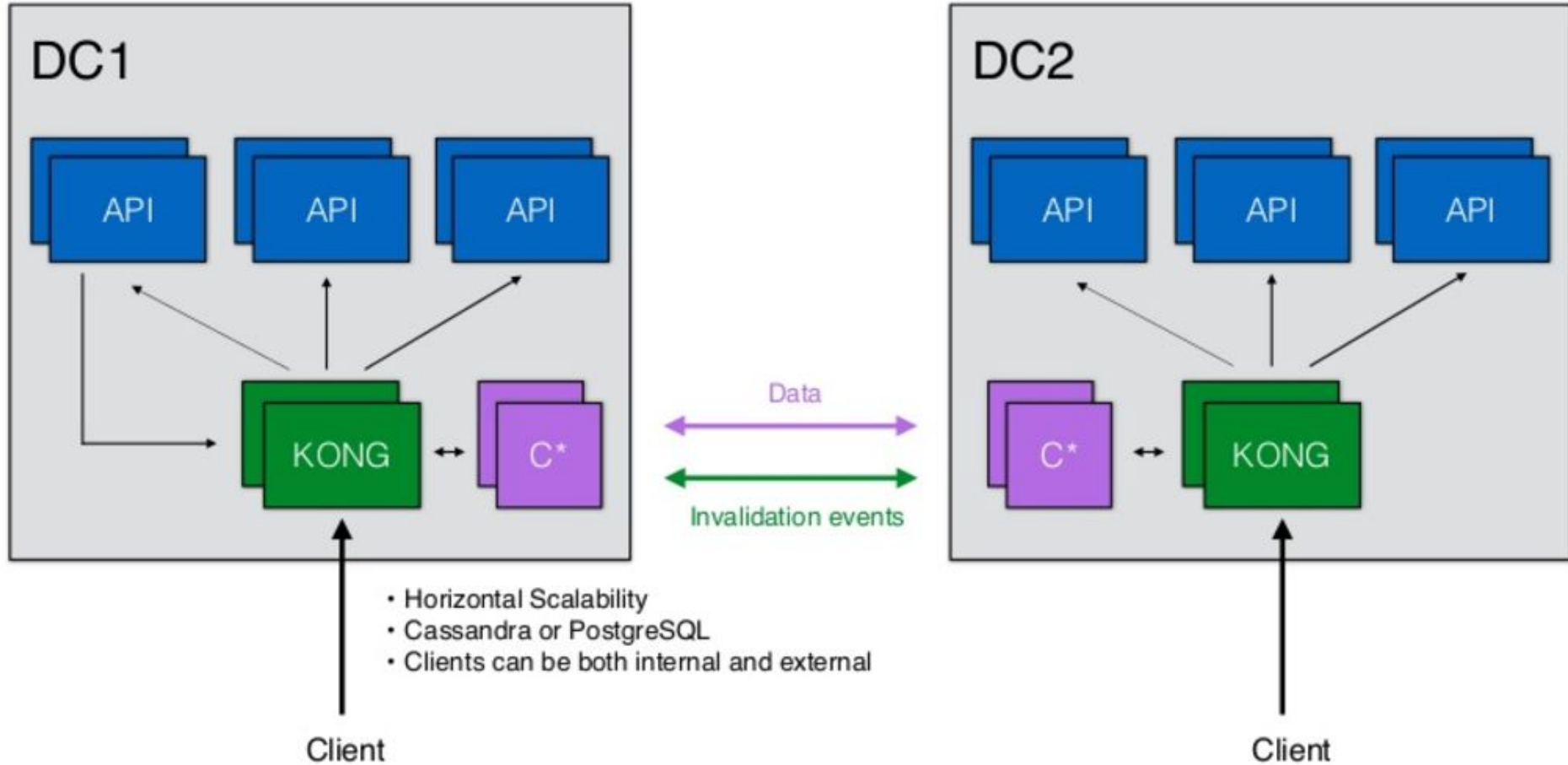
# Releases canary



# Load balancing



# Clustering





# Demo



A P I B A R

[www.cheetup.com/APIBAR](http://www.cheetup.com/APIBAR)

[@APIBARCENTOA](#)



**Thank You.**



A P I S A R

[WWW.HOOTUNGOVAPISAR](http://WWW.HOOTUNGOVAPISAR)

 @APISARARGENTINA