



Kong Meetup

Meetup:

Kong 101

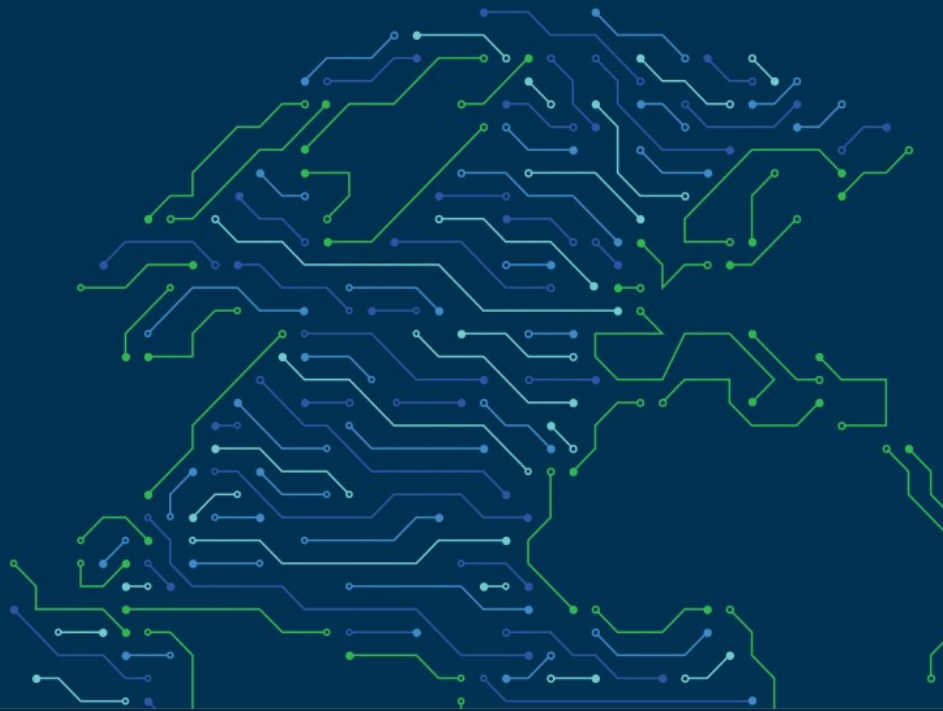
Presentado por:

Iván Tapia

Software engineer at 

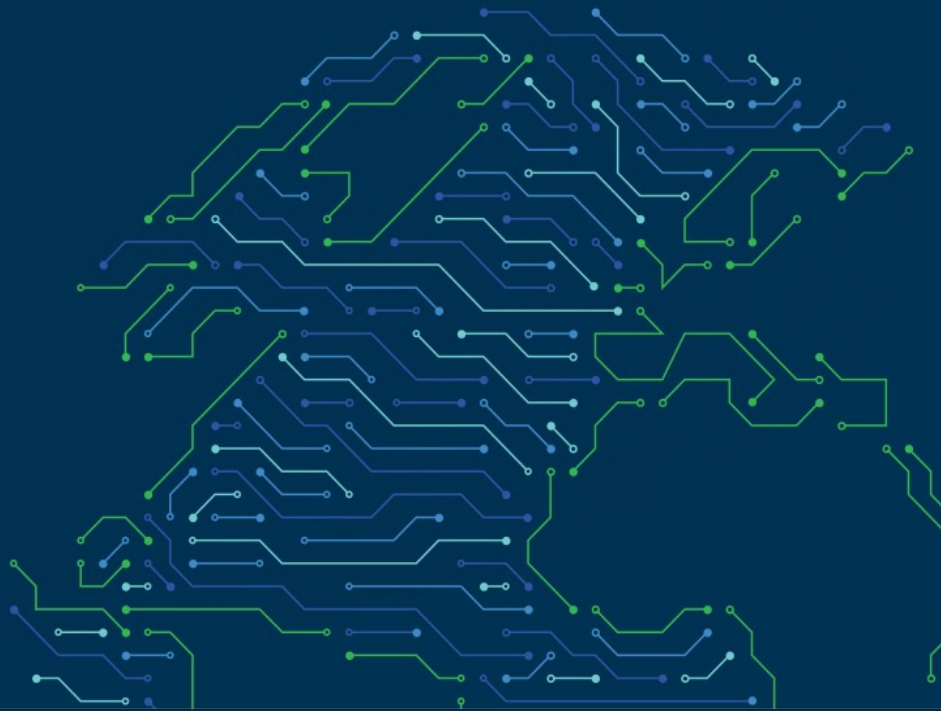
 /silanpa

12 de Octubre, 2018



Agenda

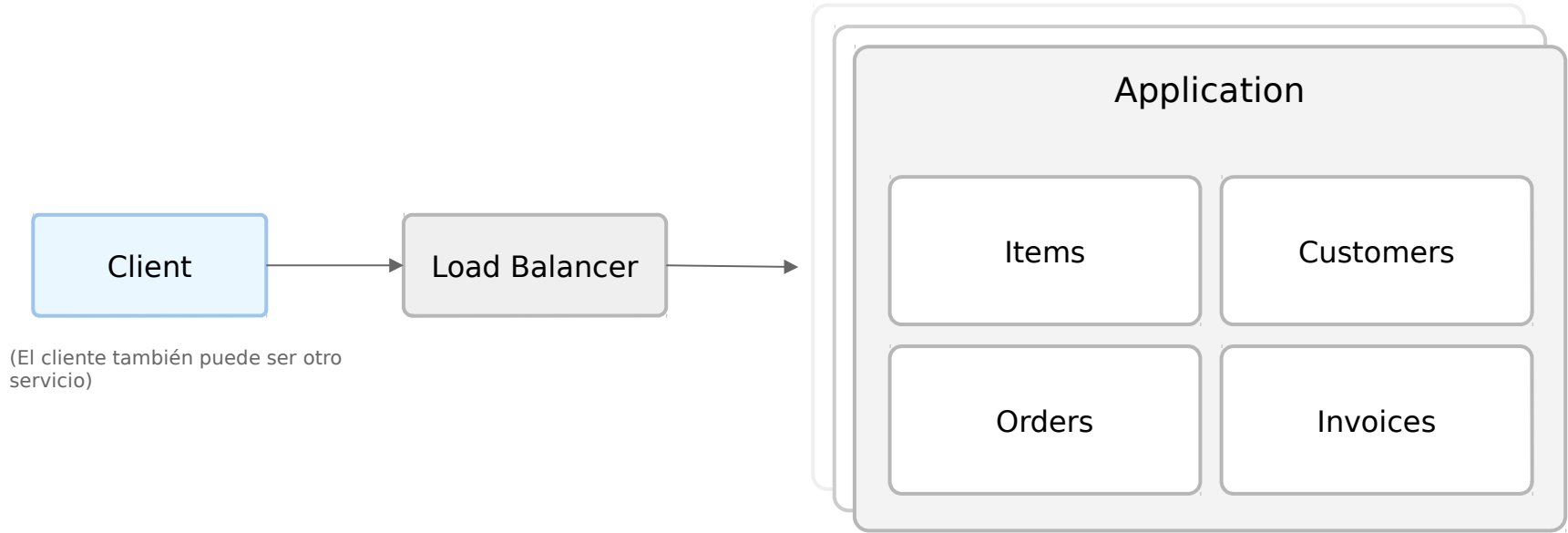
- * Api gateway y servicios
- * Tecnologías que ocupa Kong
- * Instalación y uso de Kong
- * Comunidad y Kong Enterprise



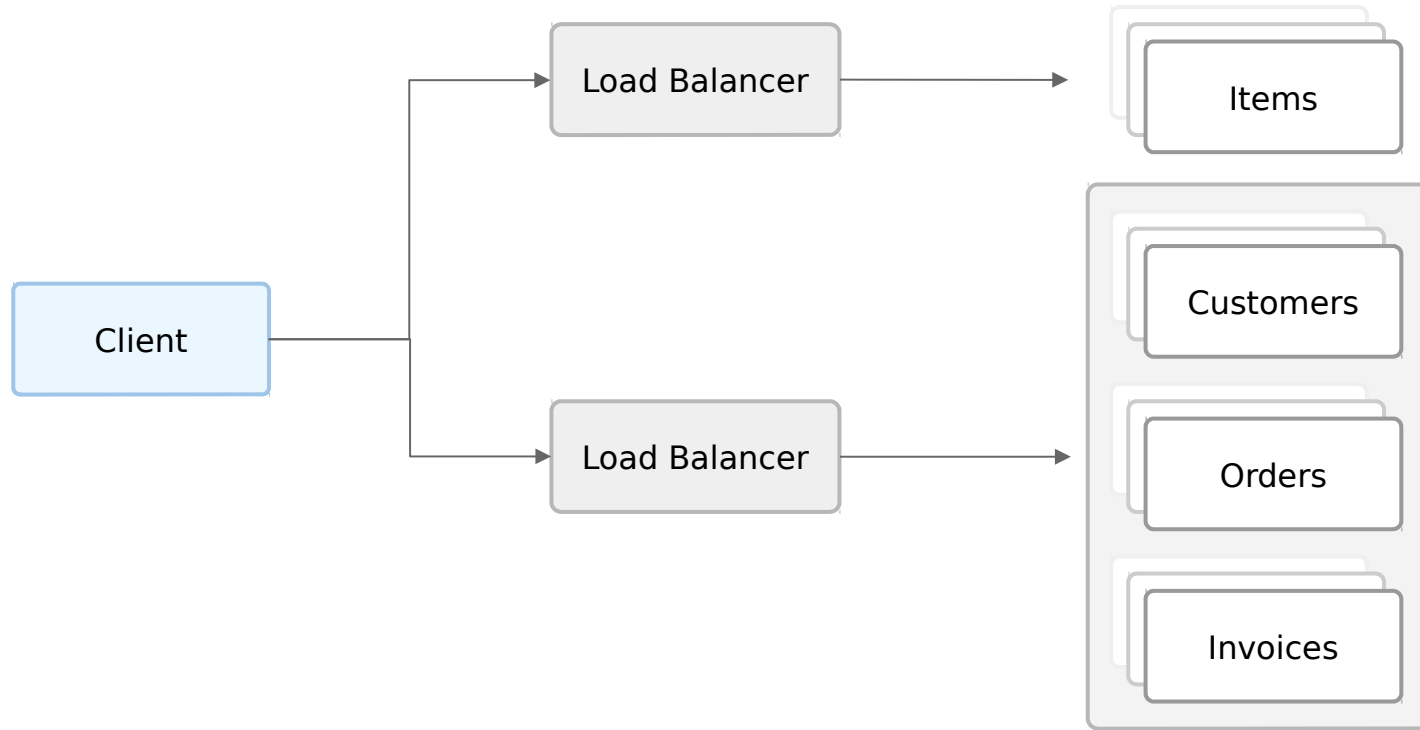


Api Gateway y Servicios

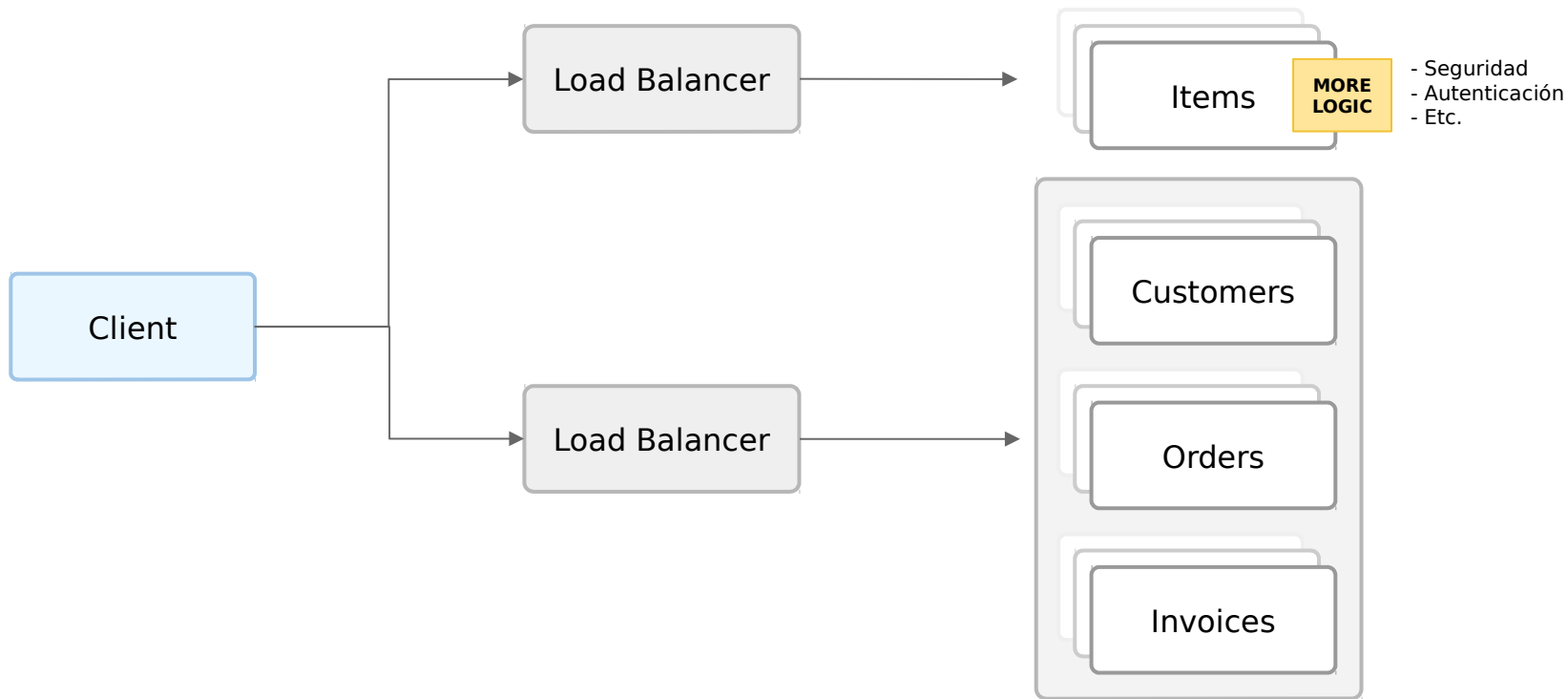
Aplicación de artículos



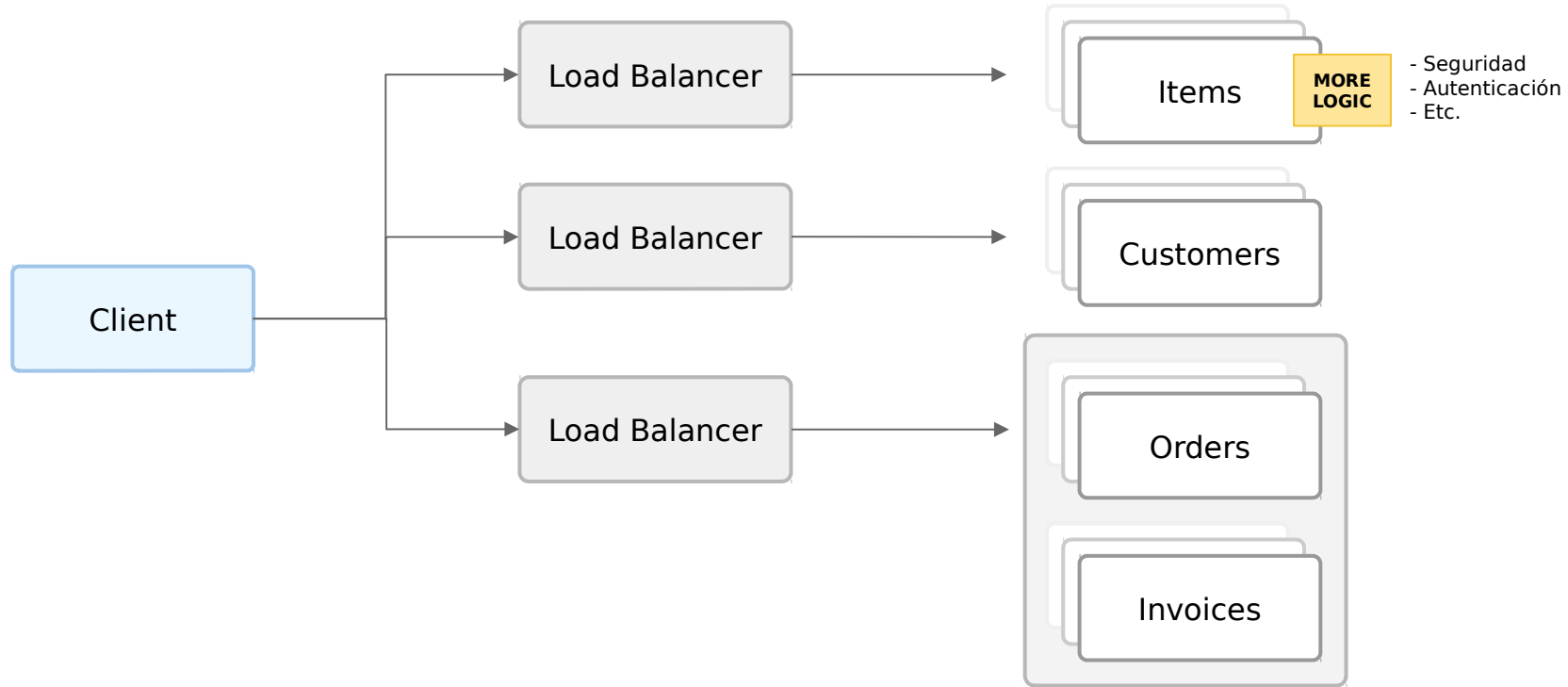
Escalabilidad y desacoplamiento



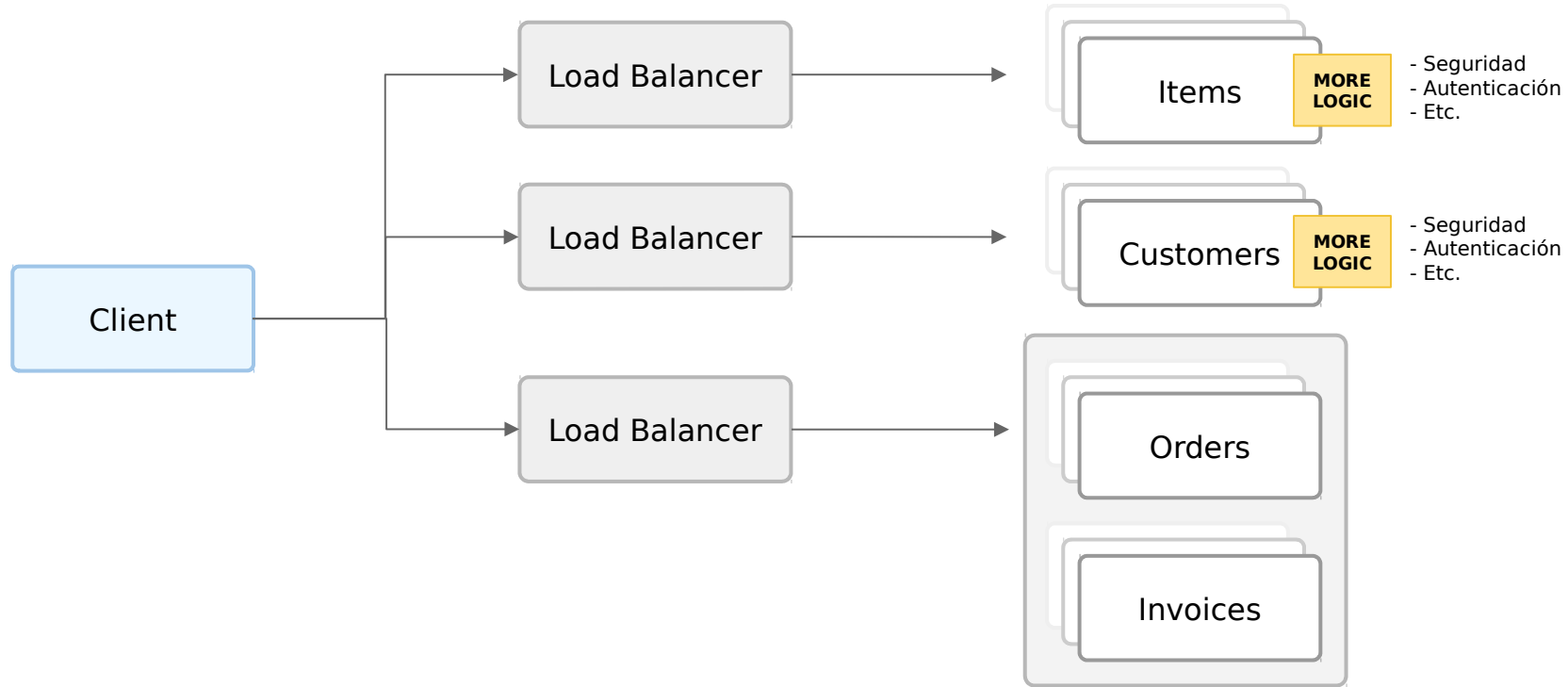
Seguridad, protección, logs, etc



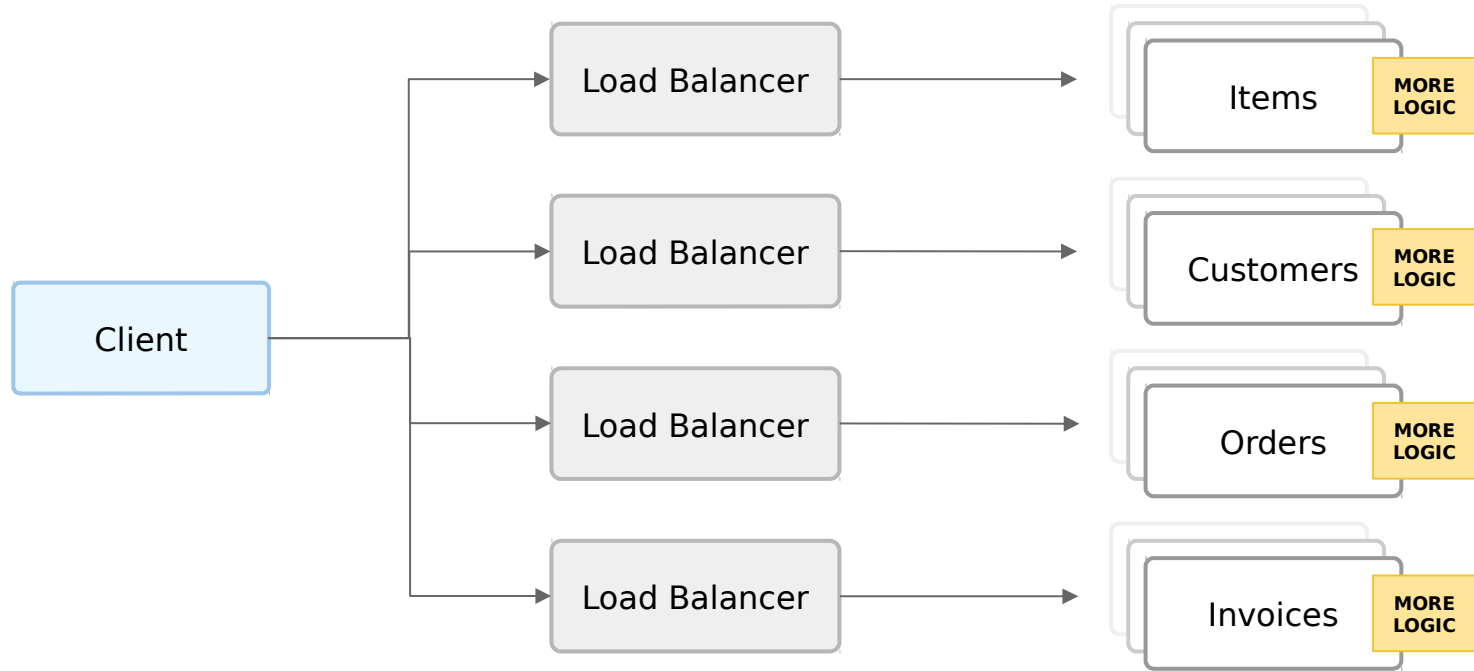
Desacoplando más servicios



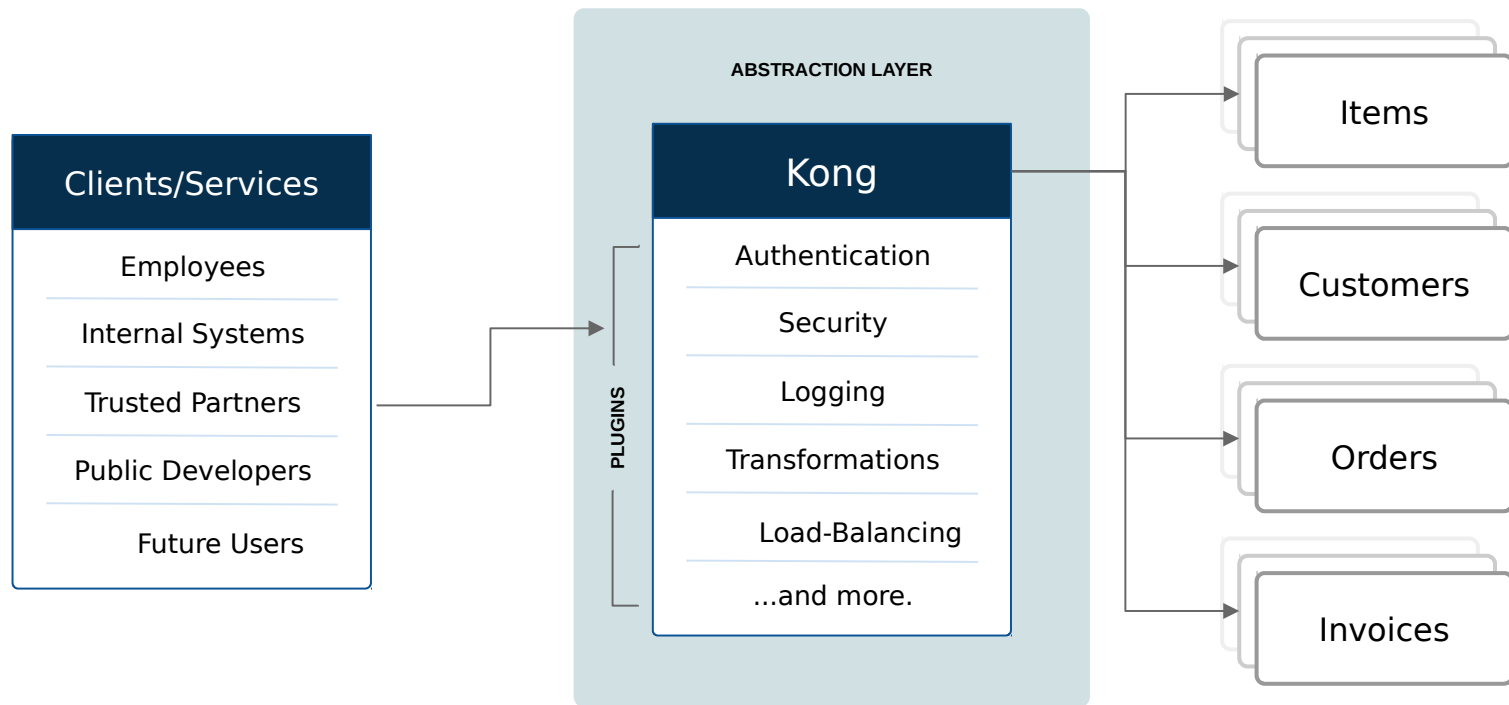
Desacoplando más servicios



Fragmentación



¿Por qué un API Gateway?



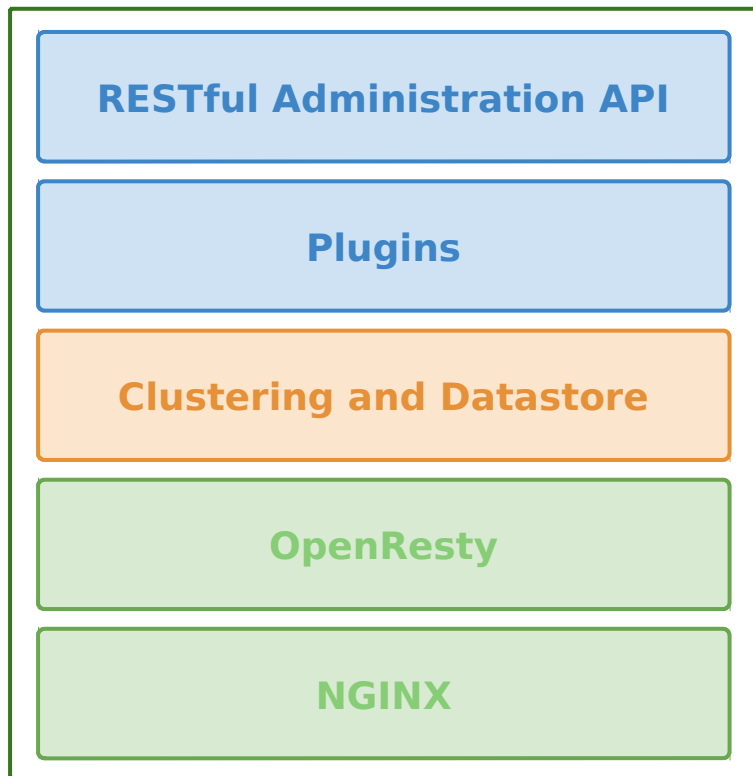
The background is a dark blue field filled with a complex, abstract pattern of glowing circuit lines. These lines, in shades of light blue and green, meander and branch out across the entire frame, creating a sense of digital connectivity and flow. Small dots of the same colors are scattered along the lines, resembling solder points or data nodes.

¿Preguntas?



¿Que tecnologías ocupa Kong?

Kong OpenResty & NGINX



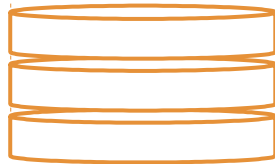
- JSON HTTP API
- Extensibilidad por Plugins
- Puede automatizarse
- Plugins creados con LUA
- Interceptar Request/Response
- Cassandra o PostgreSQL
- Redis para algunos plugins
- Single o Mult-DC clustering
- Extensibilidad provista junto con Nginx
- Dependencia principal
- Solido reconocimiento en el mercado

Kong Node Communication



- Los nodos de Kong manejan los eventos de invalidación de caché a través de un mecanismo de polling por medio de la base de datos
- Los nodos de Kong apuntando al mismo datastore formaran un Cluster
- Todos los nodos compartirán la misma configuración
- Todas las entidades principales de Kong son almacenadas en un cache en memoria
- No hay comunicación directa entre los nodos de Kong

- Cassandra o PostgreSQL como datastore
- Un cluster de bases de datos es diferente a un cluster de Kong



DB



- Los puertos 8000 y 8443 (SSL) son los puntos de entradas para los consumidores
- El puerto 8001 es el puerto de Administracion

Más information: [Documentation](https://docs.konghq.com)

The background is a dark blue field filled with a complex, abstract pattern of glowing circuit lines. These lines, in shades of light blue and green, meander and branch out across the frame, resembling a high-tech electronic layout. Small, solid-colored dots in blue and green are scattered along these lines, adding to the digital aesthetic.

Formas de instalación

Kong – Deployment Agnostic



Instalación de Kong con Docker

```
$ docker network create kong-net

$ docker run -d --name kong-database \
  --network=kong-net \
  -p 5432:5432 \
  -e "POSTGRES_USER=kong" \
  -e "POSTGRES_DB=kong" \
  postgres:9.6
```

Instalación de Kong con Docker

```
$ docker run --rm \  
  --network=kong-net \  
  -e "KONG_DATABASE=postgres" \  
  -e "KONG_PG_HOST=kong-database" \  
  -e "KONG_CASSANDRA_CONTACT_POINTS=kong-database" \  
  kong:latest kong migrations up
```

Ejecucion de Kong con Docker

```
$ docker run -d --name kong --network=kong-net \  
-e "KONG_DATABASE=postgres" \  
-e "KONG_PG_HOST=kong-database" \  
-e "KONG_CASSANDRA_CONTACT_POINTS=kong-database" \  
-e "KONG_PROXY_ACCESS_LOG=/dev/stdout" \  
-e "KONG_ADMIN_ACCESS_LOG=/dev/stdout" \  
-e "KONG_PROXY_ERROR_LOG=/dev/stderr" \  
-e "KONG_ADMIN_ERROR_LOG=/dev/stderr" \  
-e "KONG_ADMIN_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl" \  
-p 8000:8000 -p 8443:8443 -p 8001:8001 -p 8444:8444 \  
kong:latest
```



Entidades

Entidades Core de Kong

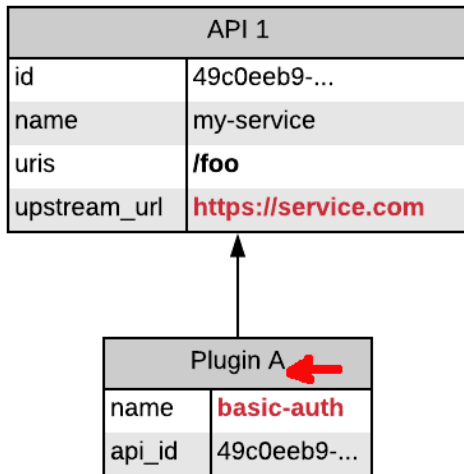
Con la llegada de la versión v0.13.x, aparecen nuevas entidades:

- Services (Nueva)
- Routes (Nueva)
- Plugins
- Consumers

Kong Entities

Core Entities	/services	/plugins	/consumers
	/routes		
Plugin Entities	/consumers/{id}/key-auth		
	/consumers/{id}/oauth2		
	...		

Plugins



Un plugin representa una configuración que se ejecutará durante el ciclo de vida de un HTTP request / response.

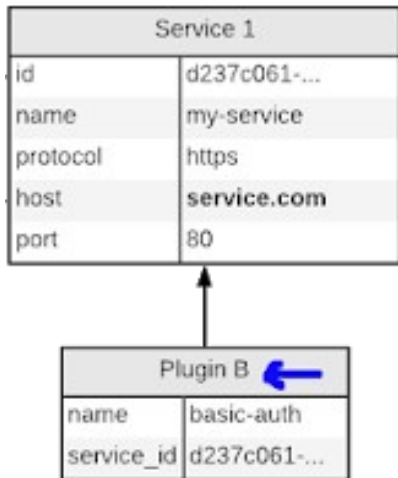
Así puedes agregar funcionalidades a las API que se ejecutan en Kong, como Autenticación o Rate-Limit , por ejemplo.

Tenemos una amplia lista de plugins disponibles, pero también podemos crear algunos ad-doc.

El endpoint utilizado para habilitar los plugins es:

/plugins

Services



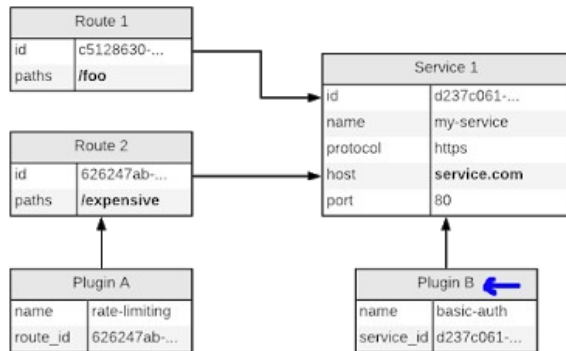
Services, como su nombre lo indica, son abstracciones de cada uno de sus propios servicios. Ejemplos de servicios serían un microservicio de transformación de datos, una API de facturación, etc.

Services están asociados a Routes (Los Services puede tener muchas Routes asociadas). Una vez que Route es encontrada, Kong envía la solicitud a su Servicio asociado.

El endpoint utilizado para crear los services es:

/services

Routes



Las Routes definen las reglas para que coincidan con las solicitudes de los clientes.

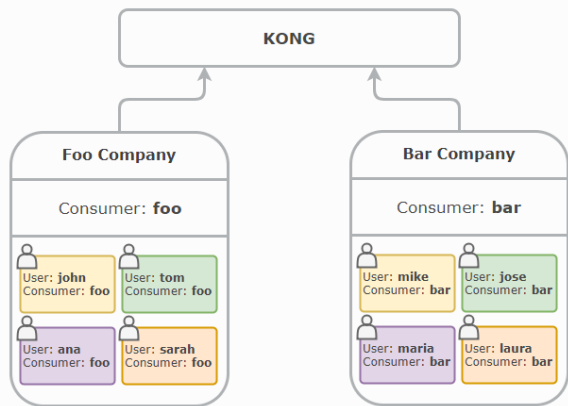
Cada ruta está asociada a un Service , y un Service puede tener múltiples rutas asociadas.

La combinación de Routes y Sevices ofrece un poderoso mecanismo de enrutamiento con el que es posible definir puntos de entrada detallados en Kong que conducen a diferentes servicios.

El endpoint utilizado para crear los routes es:

/routes o /services/<service>/routes

Consumers



El Consumer representa un consumidor, o un usuario, de una API. Estos funcionan en conjunto con los plugins y los distintos métodos de autorización.

El endpoint utilizado para crear los consumers es:

/consumers

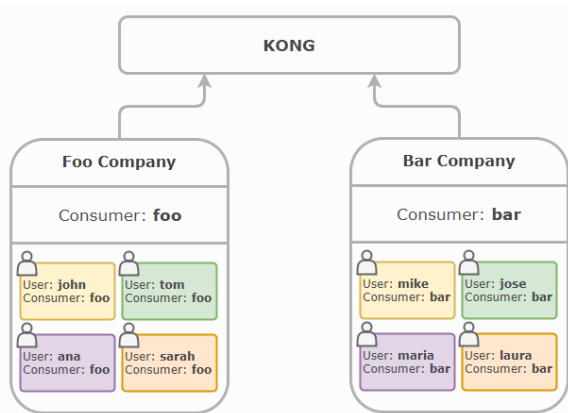


¿Preguntas?

The background is a dark blue field filled with a complex, abstract pattern of glowing circuit lines. These lines, in shades of light blue and green, meander and branch out across the frame, punctuated by small, bright circular nodes. The overall effect is one of high-tech connectivity and digital flow.

Upstream y Target

Upstreams



Un upstream representa un nombre de virtualhost.

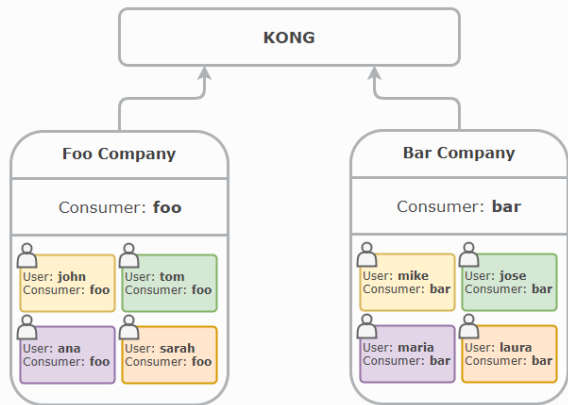
Un upstream también incluye un health checker, que puede habilitar y deshabilitar Targets en función de su capacidad o incapacidad para atender solicitudes.

El endpoint utilizado para crear los upstreams es:

/upstreams

Upstream Object : <https://getkong.org/docs/0.13.x/admin-api/#upstream-objects>

Targets



Un Target es una dirección IP / nombre de host con un puerto que identifica una instancia de un servicio de back-end.

Cada Upstream puede tener muchos targets, y estos se pueden agregar dinámicamente.

Como el Upstream mantiene un historial de cambios, los Target no se pueden eliminar ni modificar. Para deshabilitar un Target, publique uno nuevo con `width=0`; alternatively, use el método DELETE para lograr lo mismo.

El endpoint utilizado para crear los targets es:

`//upstreams/{name or id}/targets`

Target Object : <https://getkong.org/docs/0.13.x/admin-api/#target-object>

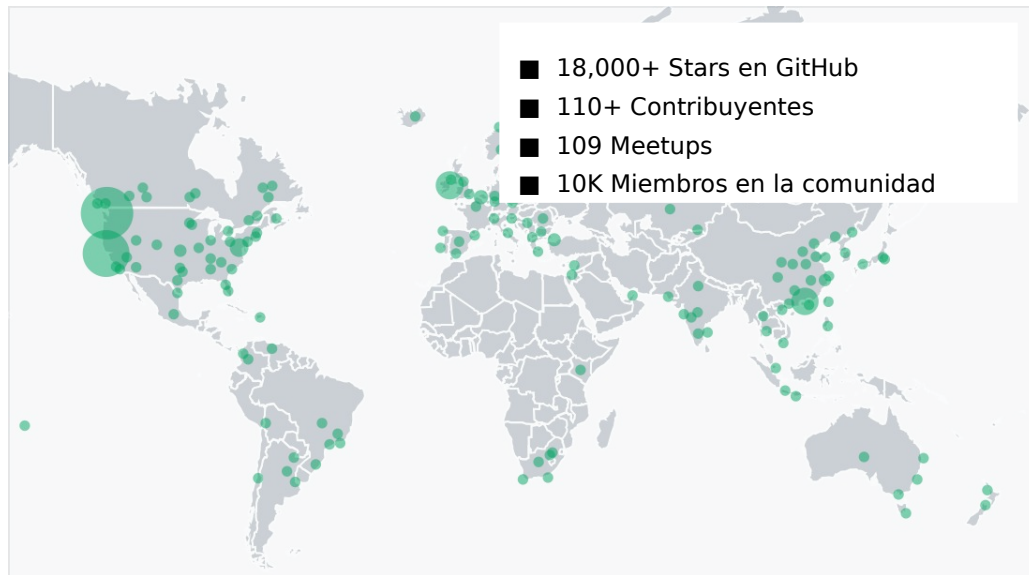


Comunidad y Kong Enterprise

Community Edition & Enterprise Edition

Millones de descargas Kong + Kong Enterprise a través de múltiples plataformas.

- Open-Source
- **15M+** Descargas desde 2015
- Construido sobre NGINX + Lua + LuaJIT
- Extensible con Plugins (**60+** disponibles)
- Latencia de sub-milisegundos en la mayoría de los casos de uso
- Cloud-Native y Platform Agnostic
- Rapido y escalable. Up and running en solo unos minutos



Enterprise Edition



Admin GUI



Security



Dev Portal



Vitals & Analytics



Scalability



Customer Success & Support

The background of the slide is a dark blue field filled with a complex, abstract pattern of white and light blue lines. These lines, representing circuit traces, crisscross the entire area in various directions. Interspersed along these lines are numerous small circles in white, light blue, and black, resembling solder points or electronic components. The overall effect is a high-tech, digital aesthetic.

Gracias!

<https://github.com/Kong/kong>

<https://konghq.com>

Meetup:

Kong 101

Presentado por:

Iván Tapia

Software engineer at 

 /silanpa

12 de Octubre, 2018

