



# Java ve .Net Tarafında Microservice Bileşenleri

ABDULKERİM KARAMAN | ALBARAKA TECH GLOBAL

**Medium:** @abdulkerimkaraman

**Twitter:** kerimkaraman24

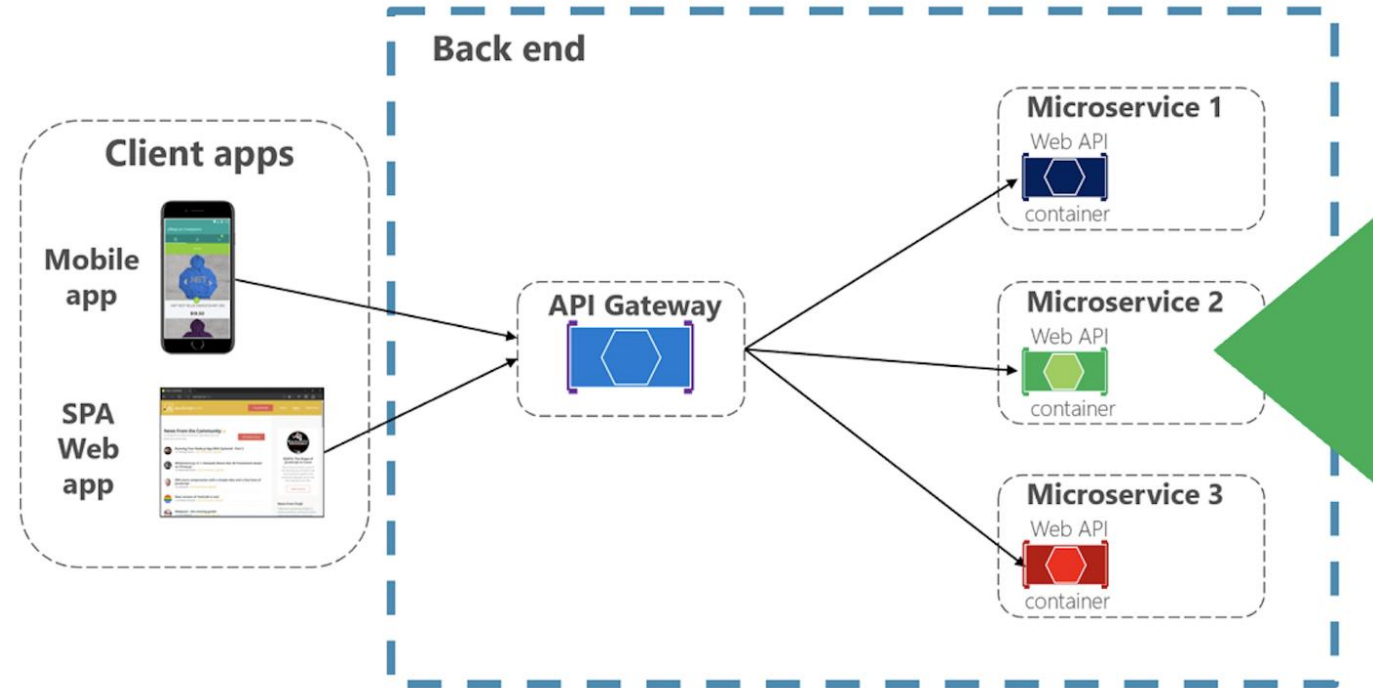
**Mail:** abdulkerimkaraman@gmail.com

# İÇERİK

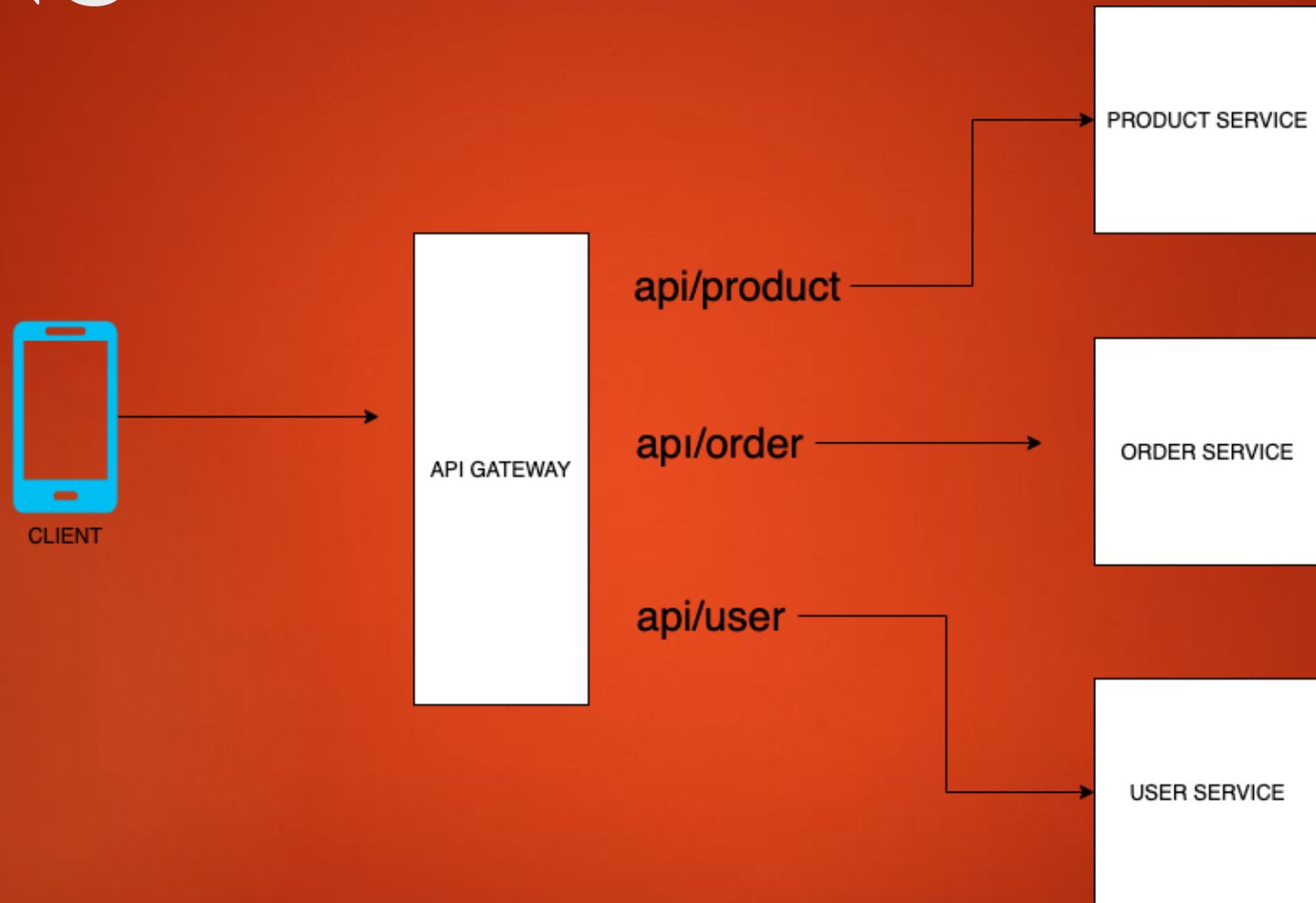
- ▶ API GATEWAY
- ▶ CONFIG SERVER
- ▶ EUREKA SERVER
- ▶ BAĞIMLI SERVİSLER
- ▶ EVENT-BASED RABBITMQ
- ▶ DEMO

# API GATEWAY – PROBLEM ?

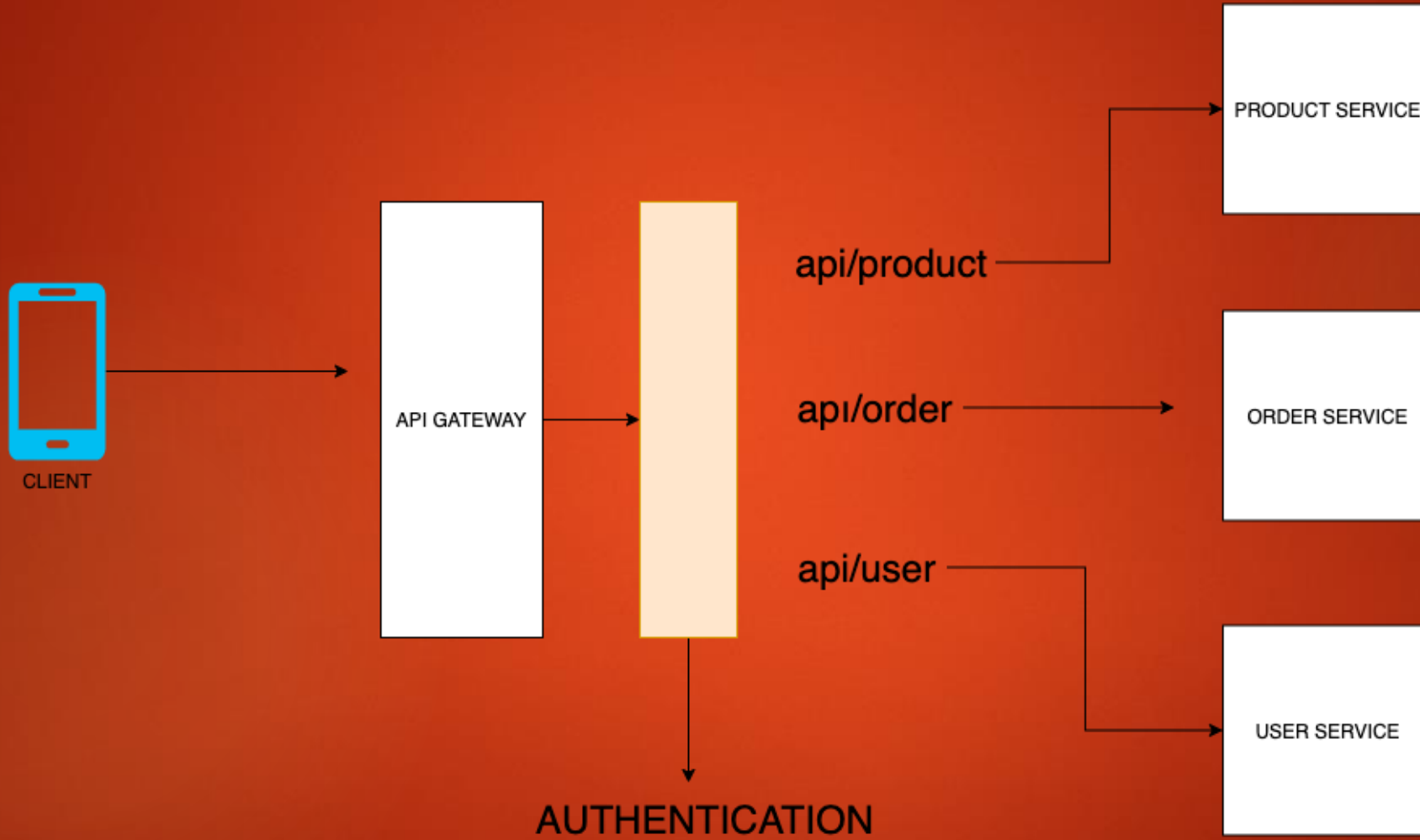
Microservice  
doğası gereği  
birden fazla sunucu  
üzerinde birden  
fazla uygulamaların  
çalıştığı yapılardır.



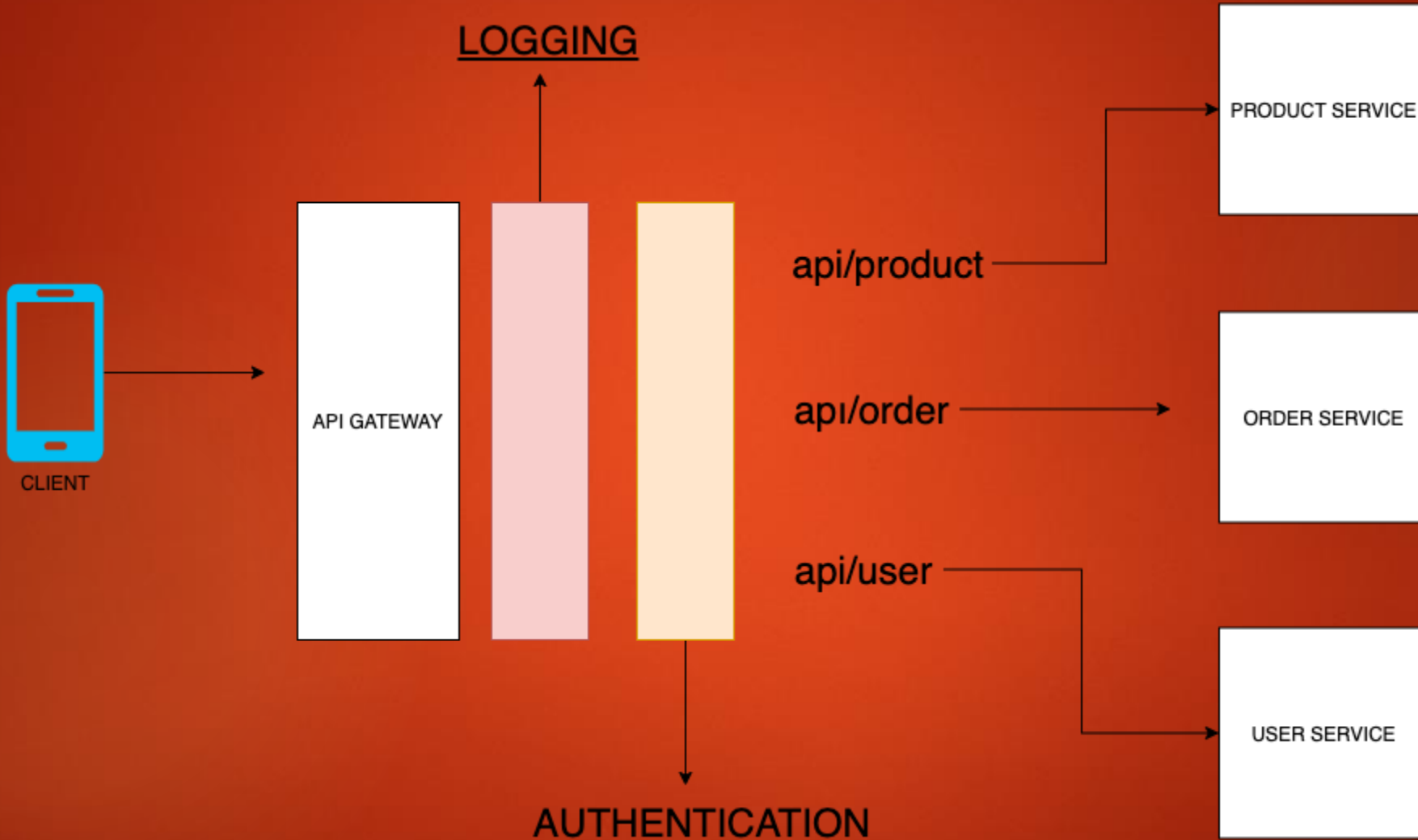
# ROUTING



# AUTHENTICATION



# LOGGING



# DAHASI..

**Request Id eklenerek request uçtan uca izlenebilir.**

Rate Limit tanımlanarak olası fazla isteklerin önüne geçilebilir.

Load Balancing yapılabilir.

Çeşitli filterlar kullanılabilir.

# Ocelot Api Gateway

- ▶ .Net Core
- ▶ Open Source

Repo: <https://github.com/ThreeMammals/Ocelot>

Site: <https://threemammals.com/ocelot>

Nuget Package: Install-Package Ocelot



# Zuul Api Gateway

- ▶ Java Spring Boot
- ▶ Open Source

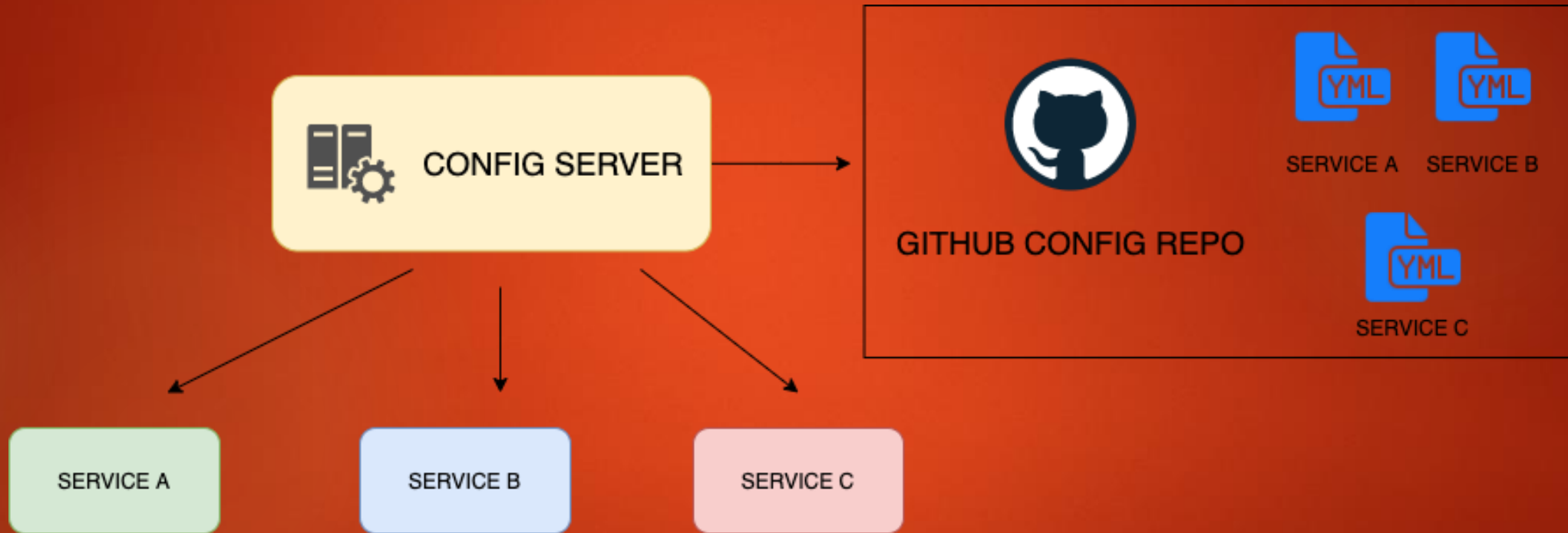
Repo: <https://github.com/Netflix/zuul>

# CONFIG SERVER

- ▶ Microservice mimarisinde her bir microservice kendine ait config bilgilerine ihtiyaç duymaktadır.
- ▶ Servisler çoğaldıkça config yönetimi zorlaşmaktadır.

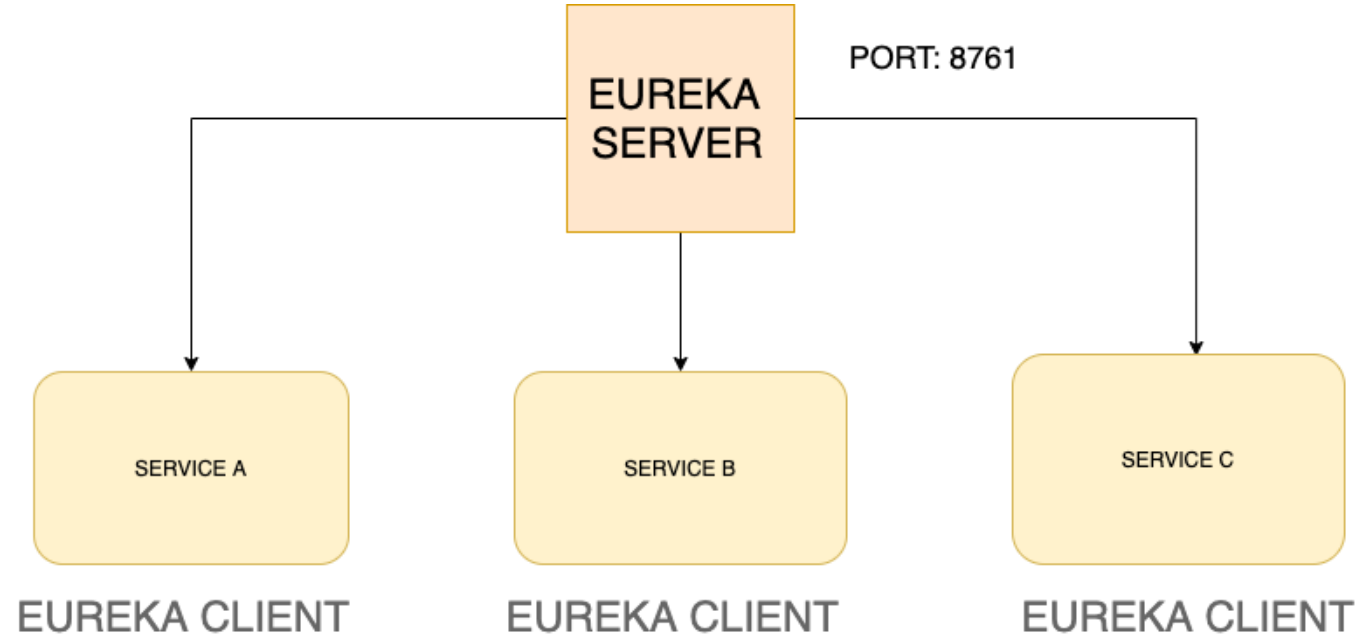
ÇÖZÜM: Merkezi bir config server yapısının kurulumu ile mümkündür.

# Spring Cloud Config Server



# Eureka Service Discovery

- ▶ Adından da anlaşılacağı üzere bir service registration bileşenidir.
- ▶ Server ve client olarak iki kısımdan oluşmaktadır.
- ▶ Her bir microservice eureka server'ın clientı şeklinde tanımlanır.
- ▶ İlgili microservice ayağa kalkarken kendini eureka service'e kayıt eder.
- ▶ Ayrıca api gateway üzerinden gelen istekleri eşit olarak dağıtır.



# DİĞER DISCOVERY SERVICE UYGULAMALARI



Kubernetes ve Cloud Foundry tarafında kullanılan bir service registration uygulamasıdır.



Consul; service discovery, distributed key-value store, health checking özelliklerini bünyesinde barındır piyasada sıkça kullanılan bir araçtır.



Önceden Hadoop içerisinde bir alt proje olan ZooKeeper, zamanla üst seviye bir Apache projesi haline gelmiştir. İlk olarak Yahoo şirketinde kullanılmıştır, günümüzde de kullanılmaktadır.

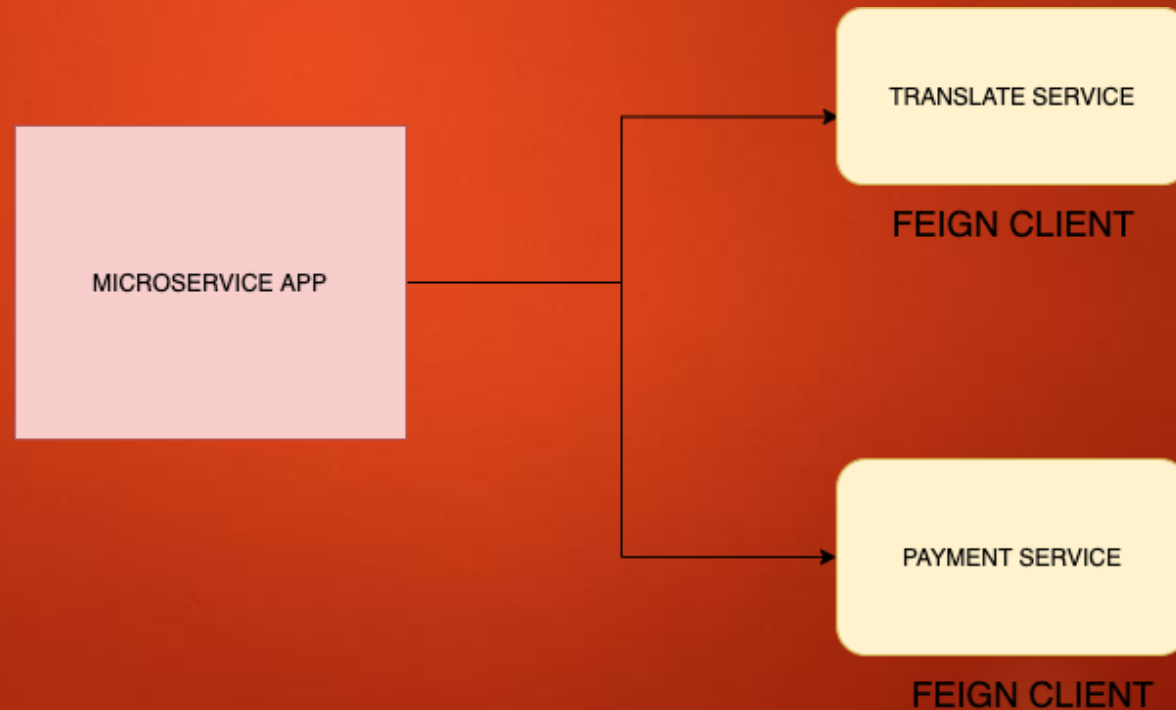
# BAĞIMLI OLDUĞUMUZ DİĞER SERVİSLER

- ▶ Monolitik uygulamalarımız
- ▶ Third party uygulamalarımız
- ▶ Filter ile loglama
- ▶ Hata yakalama

# Feign Client ve Hystrix

Repo: <https://github.com/OpenFeign/feign>

- Feign Client java http istemcilerinin yazılmasını ve yönetilmesini kolaylaştıran bir bileşendir.



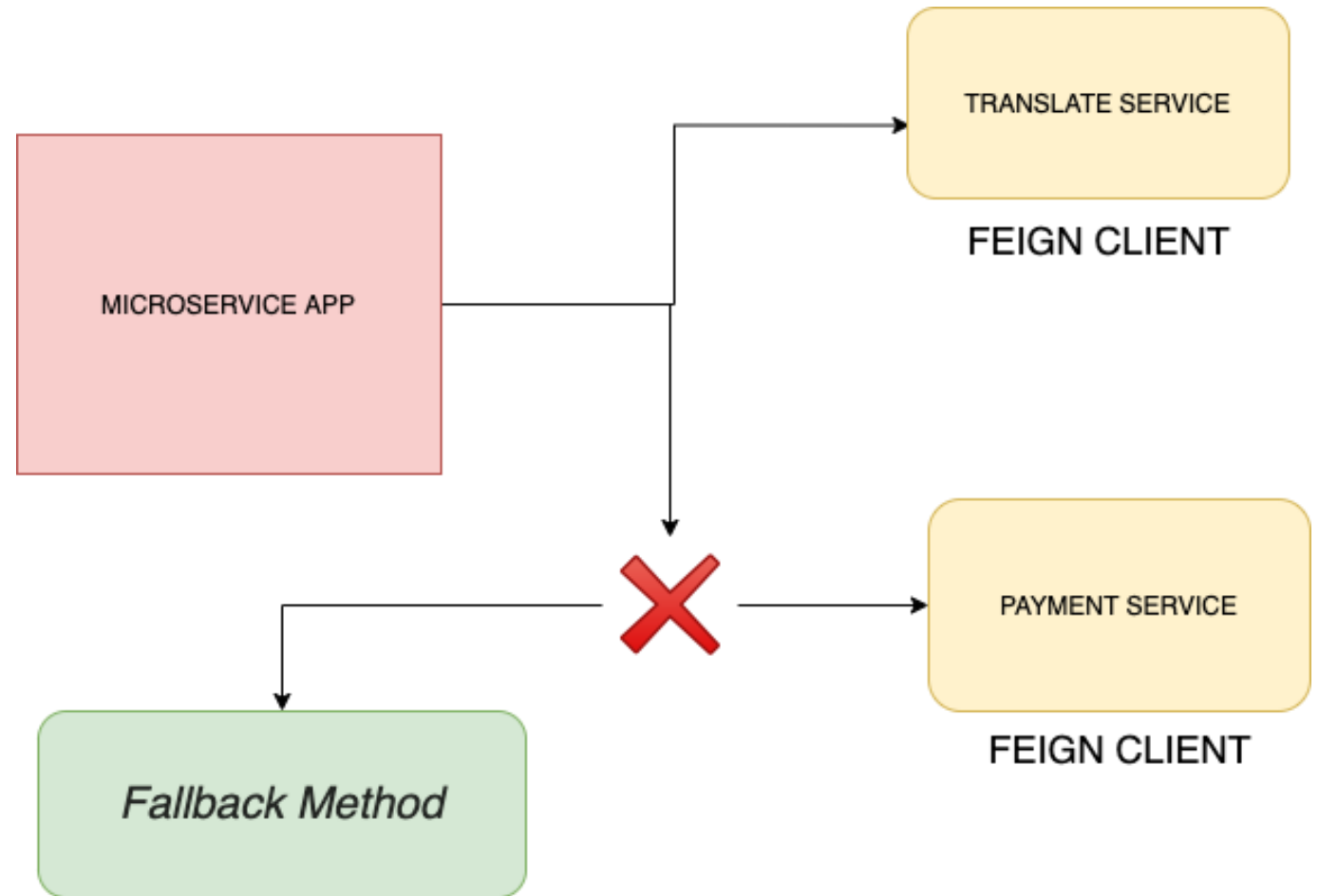
# TEMEL ÖZELLİKLER

- ▶ HTTP DESTEĞİ
- ▶ SOAP
- ▶ MOCK DATA
- ▶ ÇOKLU SERVER TANIMI İLE RIBBON LOAD BALANCING
- ▶ LOGGING
- ▶ TIMEOUT
- ▶ **REQUEST INTERCEPTORS**
- ▶ **HYSTRIX**



# HYSTRIX

- Feign client üzerinden yapacağımız servis çağırım işleminin fail olması durumunda kurtarıcı method.



# Servisler Arası İletişim

- ▶ Microservice gibi dağıtık mimarilerde servisler arası iletişim çok önemlidir.
- ▶ Özellikle bir işin birden fazla paydaşı olduğu seneryolarda transaction yönetimi bir hayli zordur.

Örneğin: Bir müşteri takip sistemi düşünelim. Bu sistemde müşteri kayıt edildiğinde ilgili müşteriye mail, sms ve sözleşmesinin bir nushasının pdf olarak iletilmesi gibi farklı işler olsun.

# Servisler Arası İletişim

Feign  
Client

Two-Phase  
Commit  
(2PC)

Event  
Based

# EVENT BASED

- ▶ Microservice uygulamalarının kendi aralarında olay tabanlı haberleşmesidir.
- ▶ Publisher, Subscriber ve **Queue** 'nun görevleri şu şekildedir:

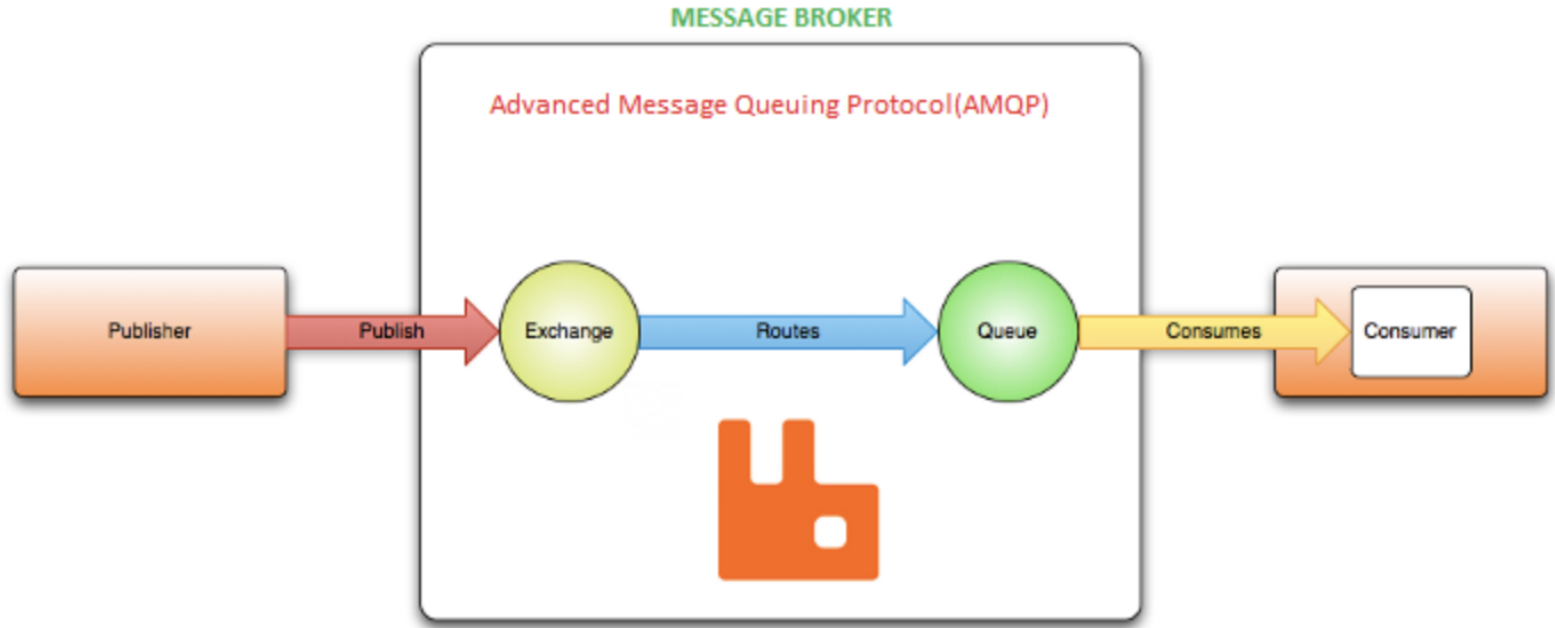
**Publisher** olarak tanımlanan servis message broker'a (rabbitmq, kafka v.s.) bir event gönderir. (örn: sendNotification)

**Subscriber** olarak tanımlanan servise message broker üzerinde bağlı olduğu channel'a ait eventleri dinler. Kuruktan alınan event'e gönder üzerine düşen işlemi gerçekleştirir. (örn: Notification gönderir.)

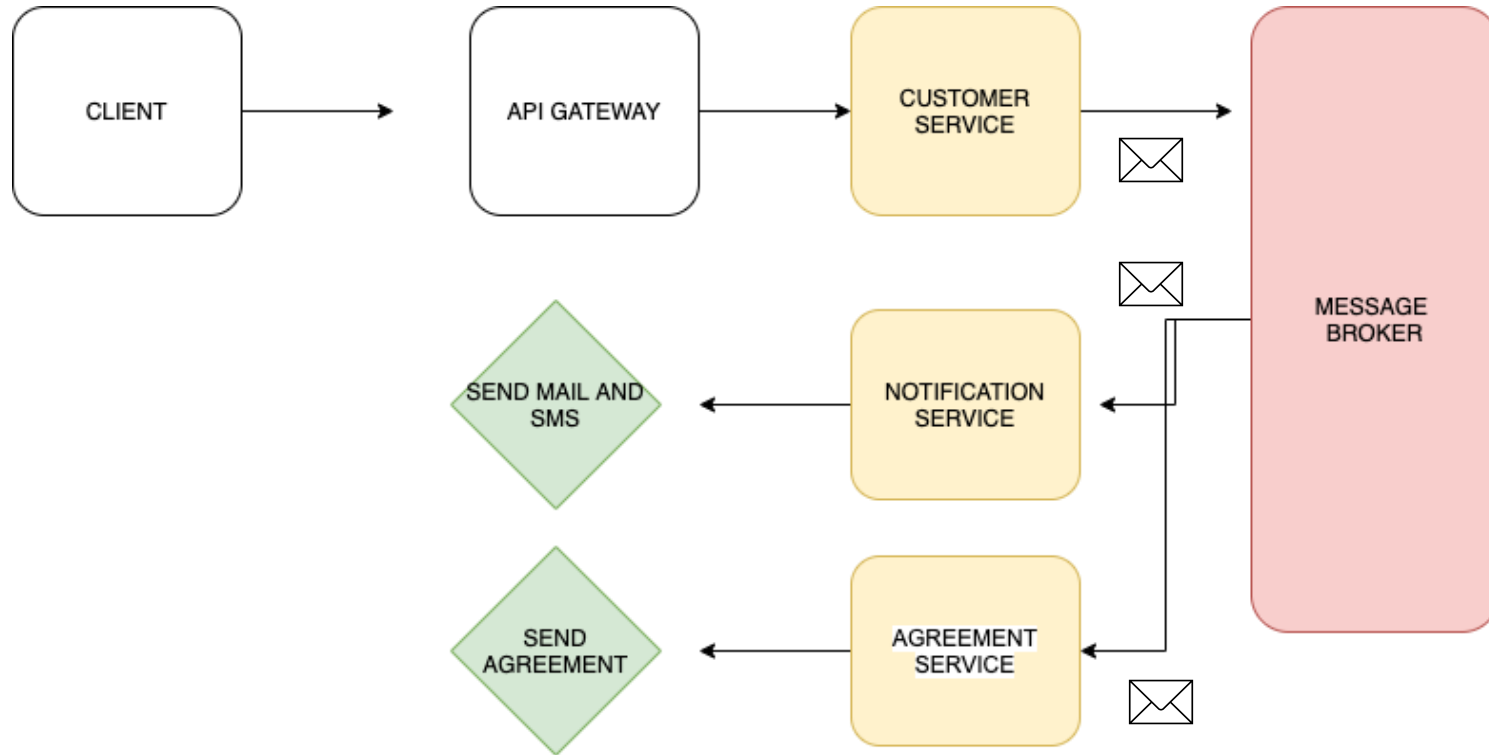
**Queue:** Broker üzerinde bekleyen kuyruk mesajlarını temsil eder.

# RabbitMQ

- ▶ Open source olarak geliştirilmiş bir mesaj kuyruk sistemidir. Basit anlamda herhangi bir uygulamadan aldığı mesajı başka bir uygulamaya iletir.
- ▶ Erlang ile geliştirilmiştir.
- ▶ Farklı işletim sistemlerinde host edilebilir.
- ▶ Bir çok cloud provider tarafından hizmet olarak sunulmaktadır.



# EVENT BASED



# SENARYO

DEMO..



TEŞEKKÜRLER.

ABDULKERİM KARAMAN