

# Applying CQRS with .NET Core in Financial Technologies

Murat Dinç





# Murat Dinç

2010 yılında profesyonel kariyer hayatıma atıldım. Bu süre zarfında özellikle Microsoft teknolojileriyle ilgilenerek **Software Developer, Senior Software Developer** ve **Team Lead** gibi pozisyonlarda bulundum. 14 yıllık aktif tecrübeye sahibim.



Şu an Sipay Elektronik Para ve Ödeme Hizmetleri şirketinde **Software Architect** rolünde çalışmaktayım.

- .NET Core (All Versions)
- .NET Framework
- ASP.NET Core MVC, Web API
- Microsoft SQL Server
- MySQL
- MongoDB
- PostgreSQL
- Elastic Search
- Distributed Caching (Redis)
- Docker
- Multi-Tenant Architecture
- Git (Azure, BitBucket, GitLab, Github)
- CI/CD (Jenkins, Octopus, Azure Pipeline, BitBucket)
- Cloud (Azure, Amazon)
- Design Patterns
- SOLID, DRY, YAGNI, KISS Software Principles

## Sertifikasyonlar



## Görev Aldığım Şirketler



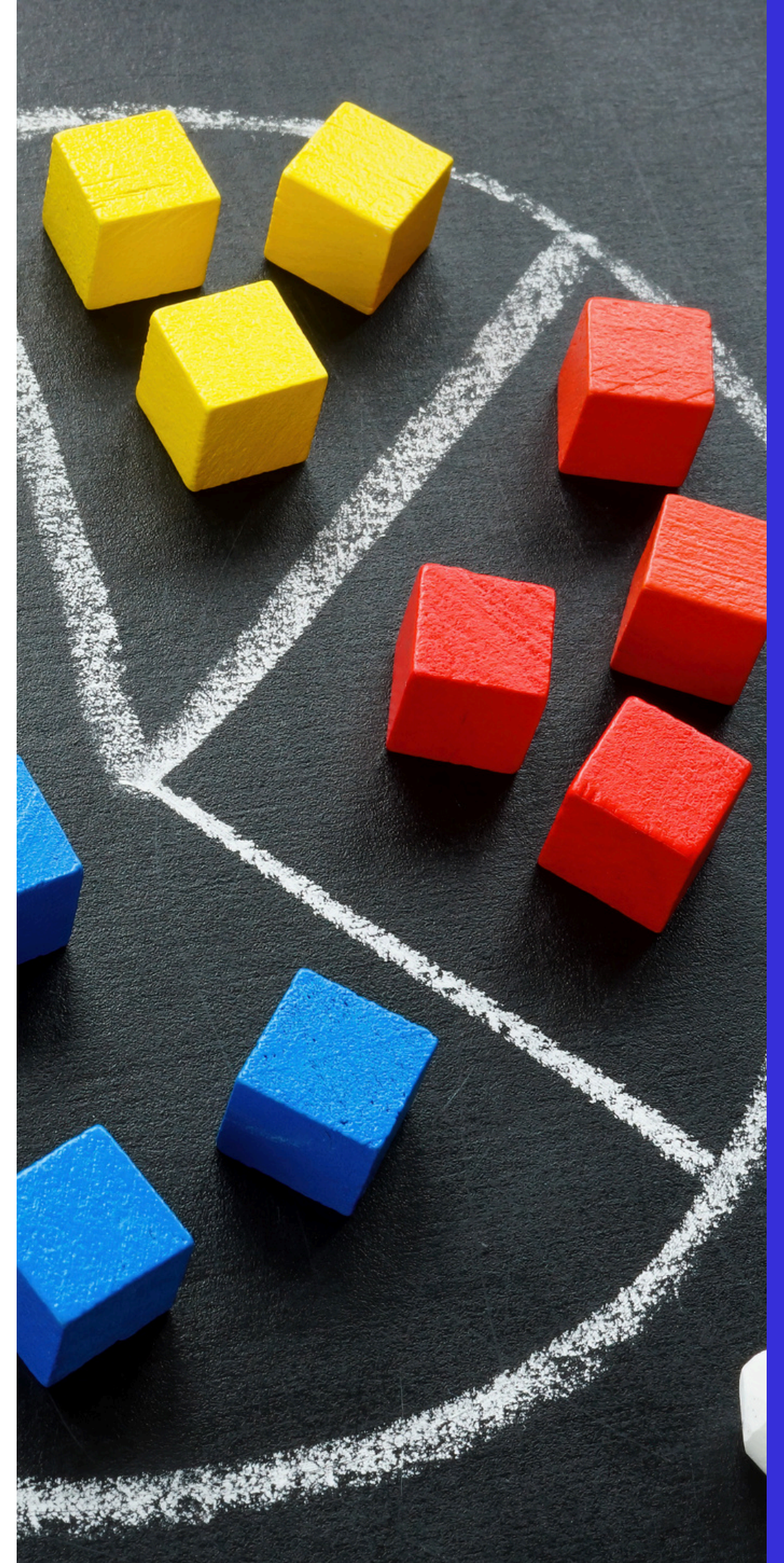


# CQRS Nedir?

## Command Query Responsibility

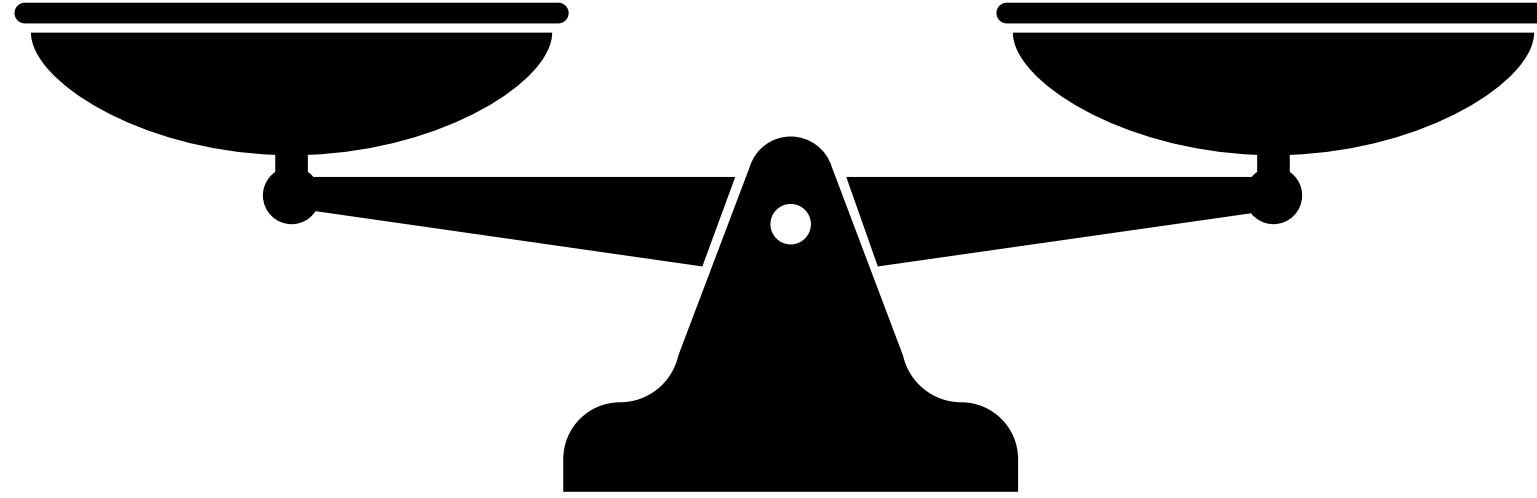
**Segregation**, komut ve sorgu sorumluluklarını ayırma ilkesine dayanır.

Bu yaklaşım, bir uygulamanın komut işleme (yani durum deęiřtirme) ve sorgulama (yani durum okuma) yönlerini birbirinden bağımsız olarak ele almayı önerir. Böylece, her iki tarafın da en iyi performansı ve ölçeklenebilirlięi sunmasını sağlar.





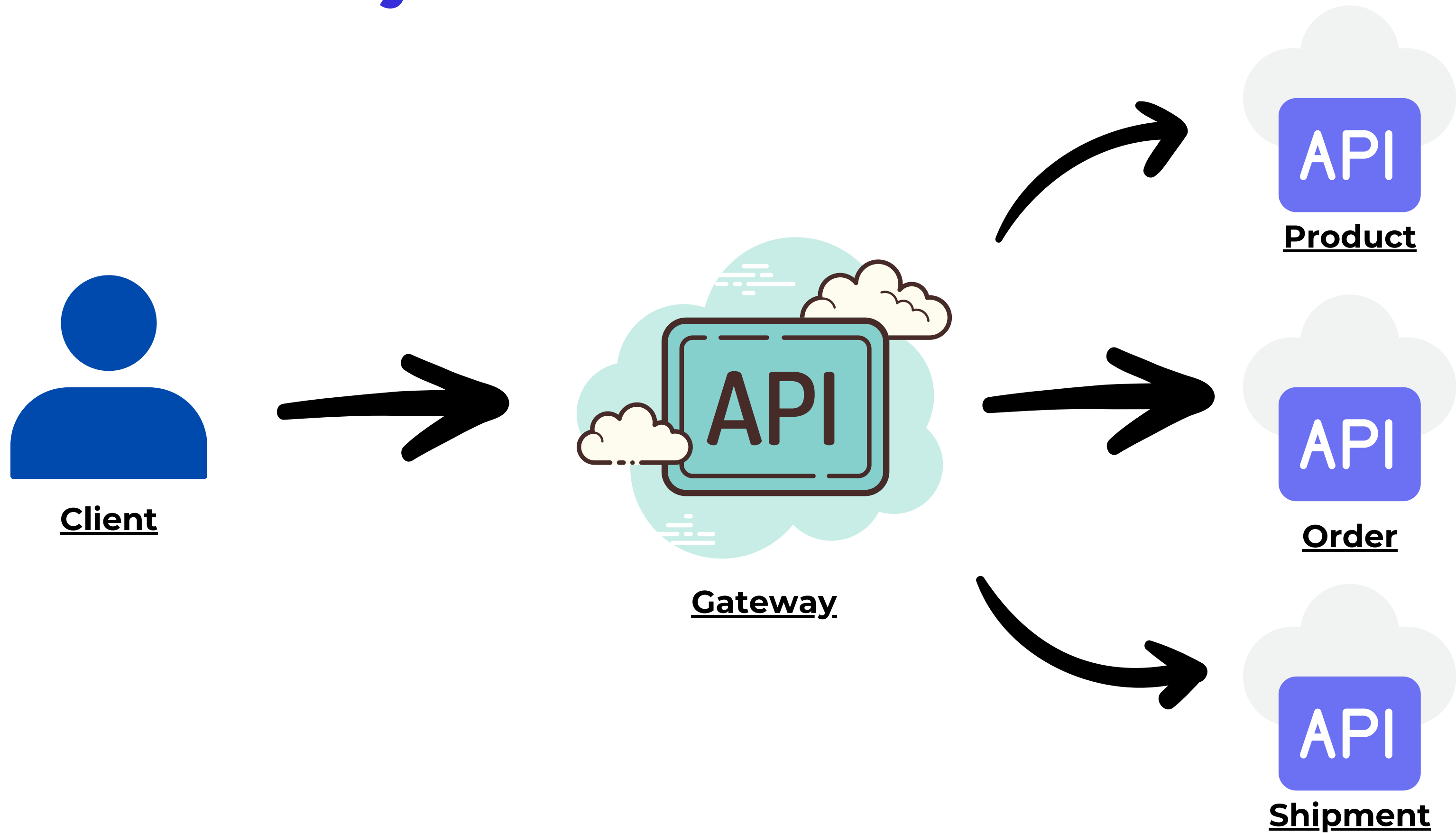
# Avantajları ve Dezavantajları



- Ölçeklenebilirlik
- Esneklik
- Yüksek Performans
- Karmaşıklık
- Geliştirme ve Bakım Maliyeti
- Veri Tutarsızlıkları

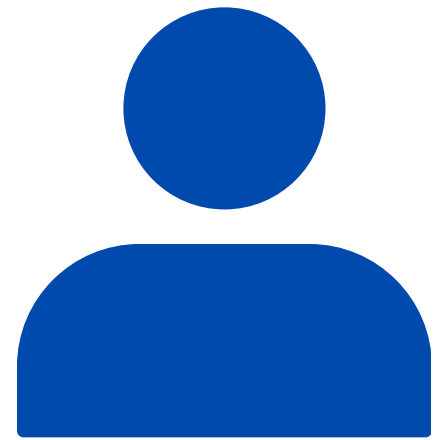


# Örnek Senaryo

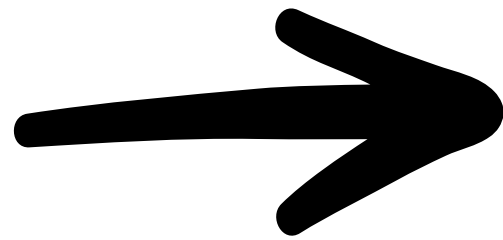




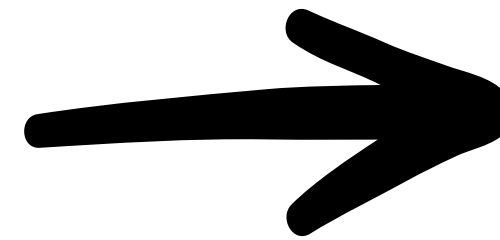
# Product servisi çok yük almaya başladı



Client



Gateway.



Product

**Read  
5000 TPS**

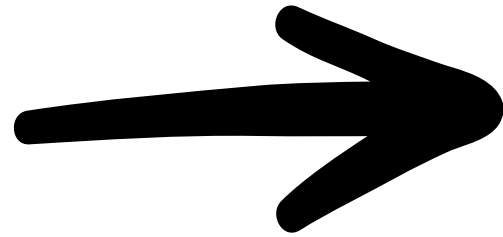
**Write  
50 TPS**



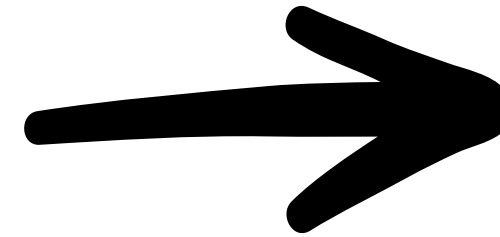
# Ek hizmetler ile servisi rahatlatalım



Client



Gateway



Product

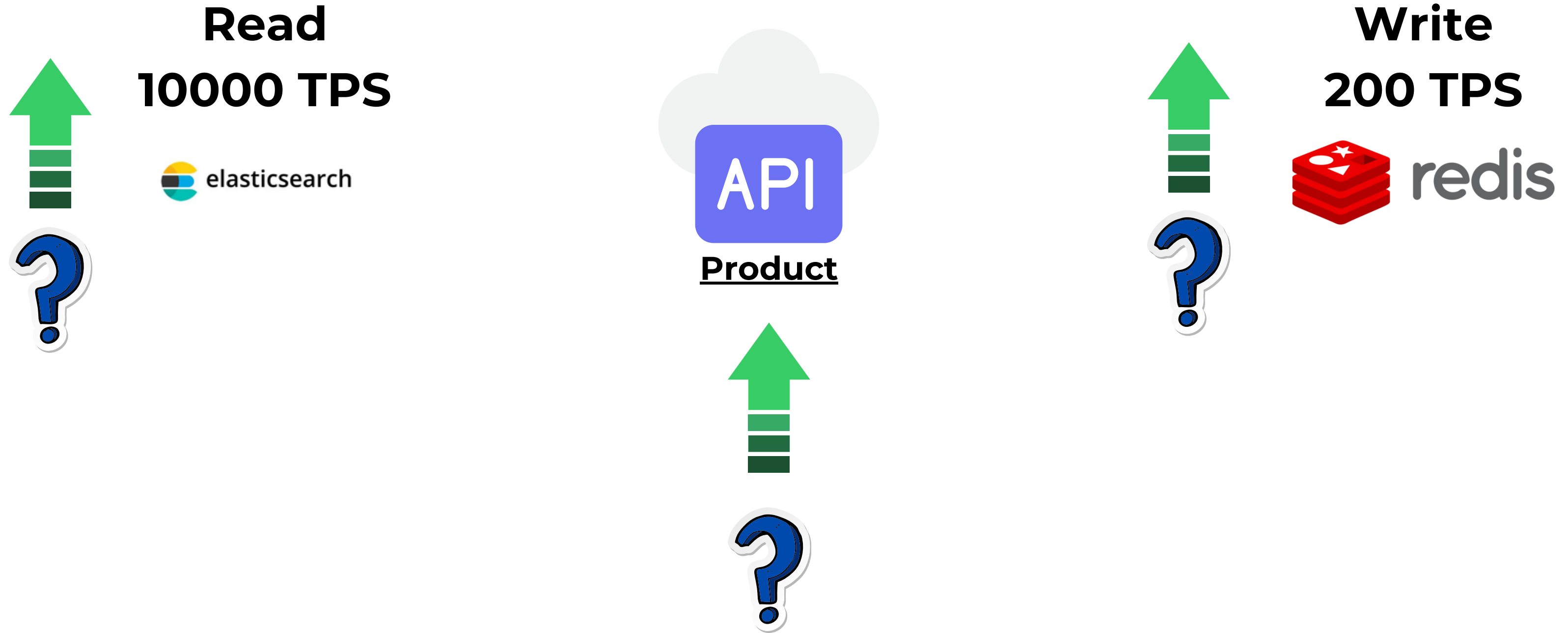


**Write  
200 TPS**

**Read  
10000 TPS**

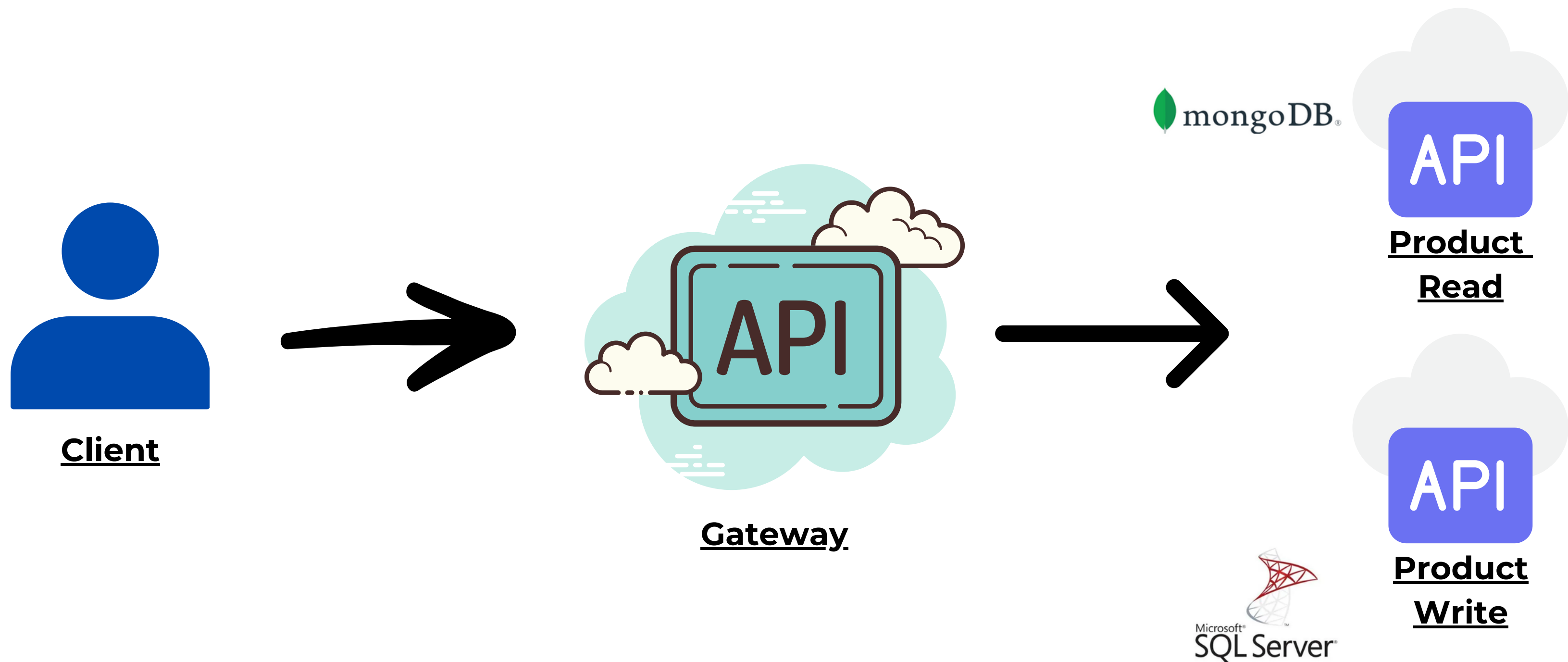


# Nasıl scale up olmamız gerekiyor?



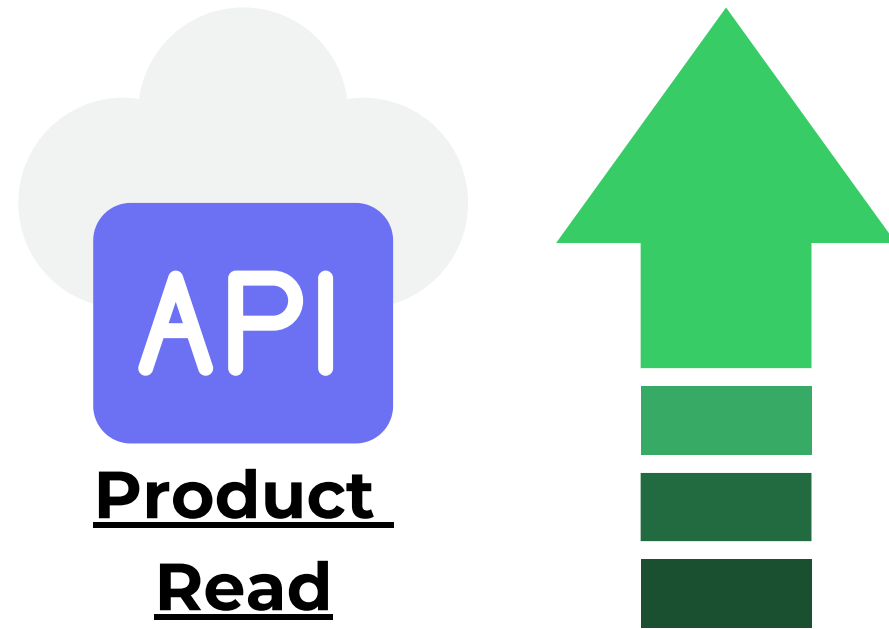


# ideal Senaryo

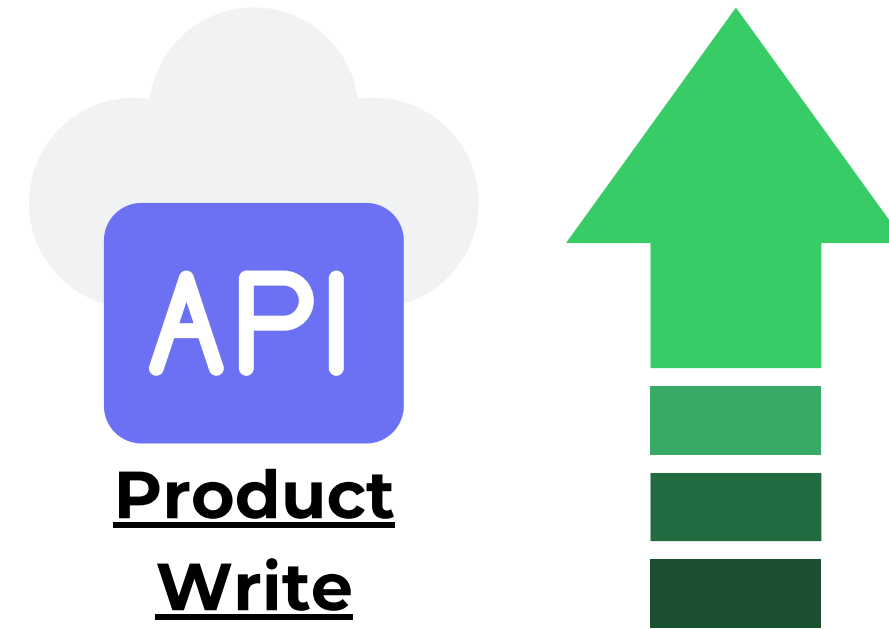


# ideal Senaryo

**Read  
20 Pod**

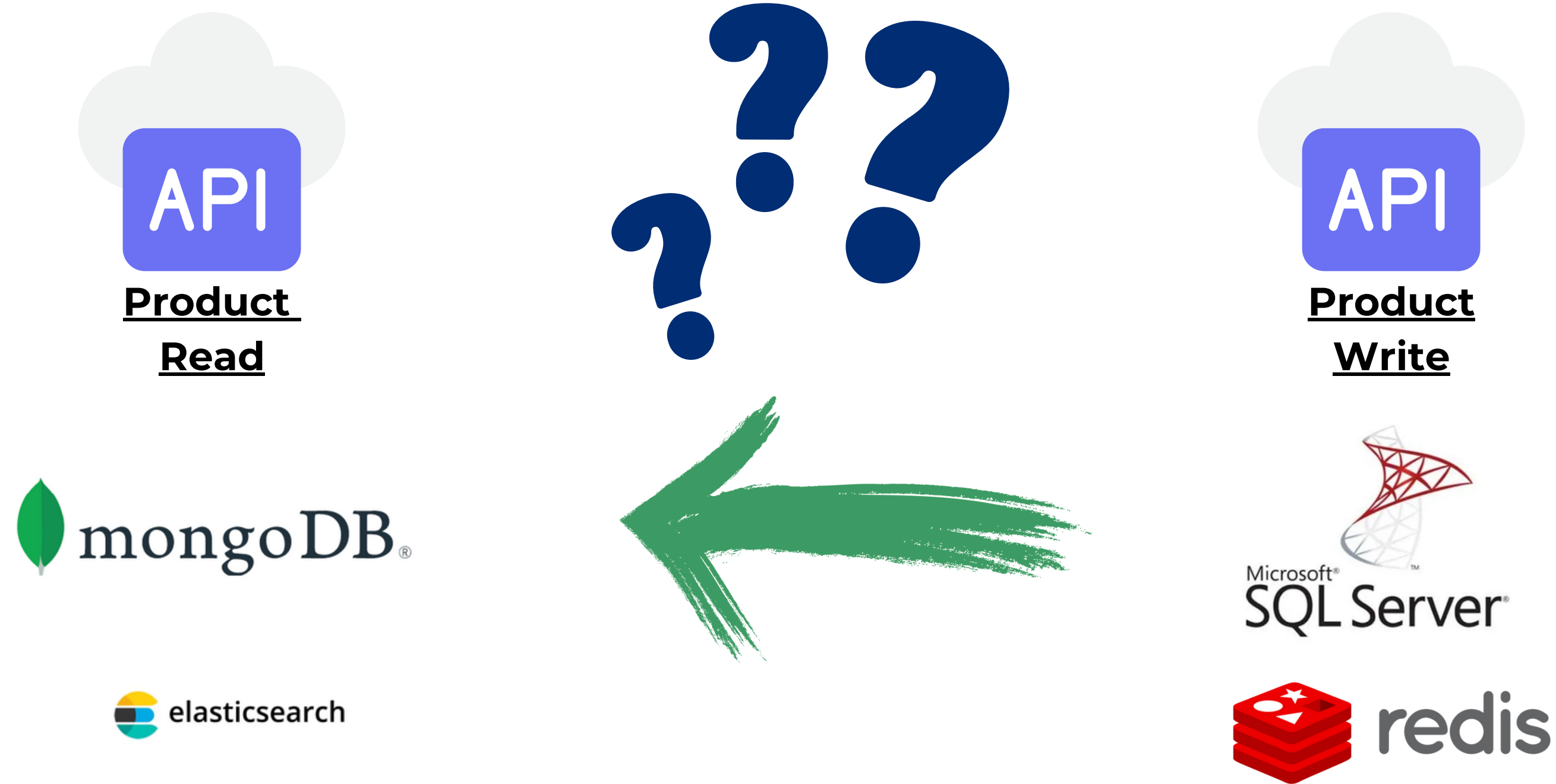


**Write  
5 Pod**





# Consistency nasıl sağlayabiliriz?



# CDC - Change Data Capture



# Event



Product  
Read



Product  
Write





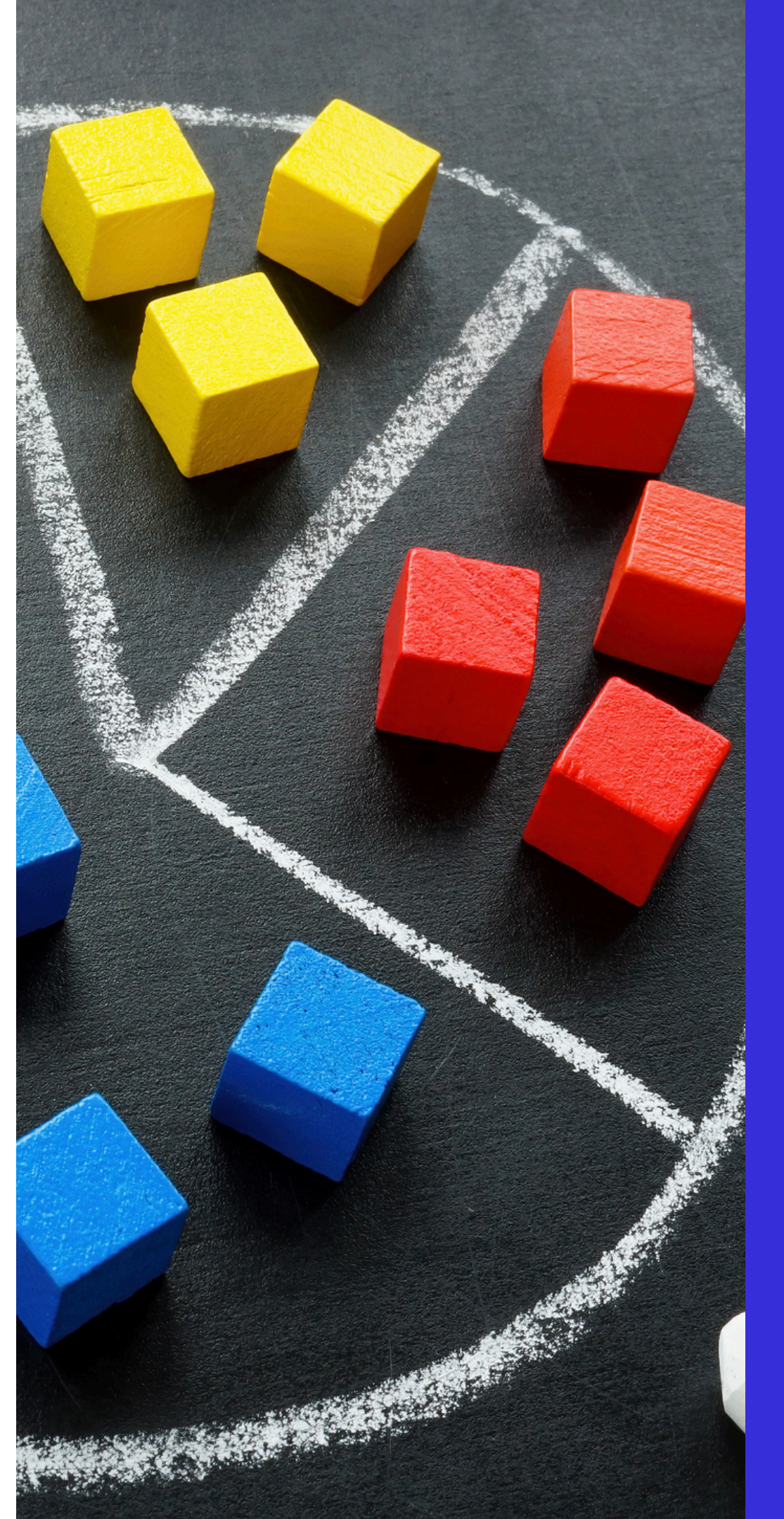
# Finansal Teknolojiler





# FinTech Olmak

Finansal teknoloji şirketleri, teknolojiyi kullanarak finansal hizmetleri daha verimli, erişilebilir ve kullanıcı dostu hale getirmeyi amaçlar.





# FinTech Şirket Tipleri

- Ödeme Sistemleri
- Dijital Bankacılık
- Kredi ve Borç Verme
- Yatırım ve Varlık Yönetimi
- Sigorta Teknolojileri (Insurtech)
- Kitle Fonlama (Crowdfunding)
- Regtech (Regülasyon Teknolojileri)
- Blockchain ve Kripto Para



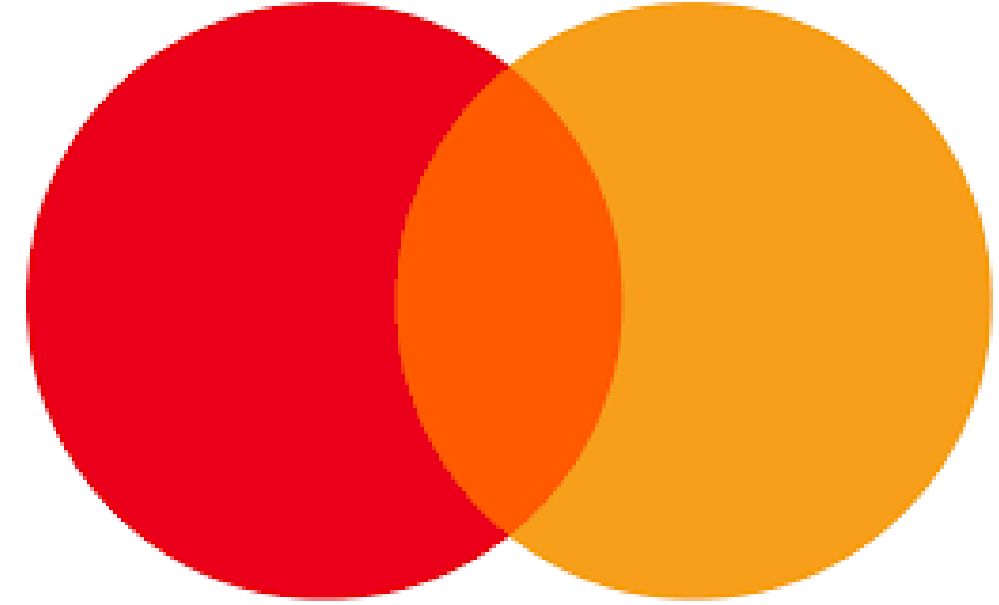


# Regölasyonlar

Regölasyon, belirli bir sektör veya faaliyet alanında uyulması gereken kuralları, standartları ve normları belirleyen yasal düzenlemeler ve denetim mekanizmalarıdır.



TÜRKİYE CUMHURİYET  
MERKEZ BANKASI



mastercard



BANKACILIK  
DÜZENLEME VE DENETLEME  
KURUMU

# SLA (Service Level Agreement) Gerçeđi

SLA, Hizmet Düzeyi Anlaşması anlamına gelir ve bir hizmet sağlayıcı ile bir müşteri arasında belirlenen hizmet standartlarını ve bu hizmetlerin nasıl ölçüleceđini tanımlayan resmi bir sözleşmedir.





# Güvenlik Politikaları

Katı güvenlik politikaları ile düzenli sızma testleri yaptırılması gerekmektedir. Uygulayacağınız yöntemler güvenlik riski barındırmamalıdır.



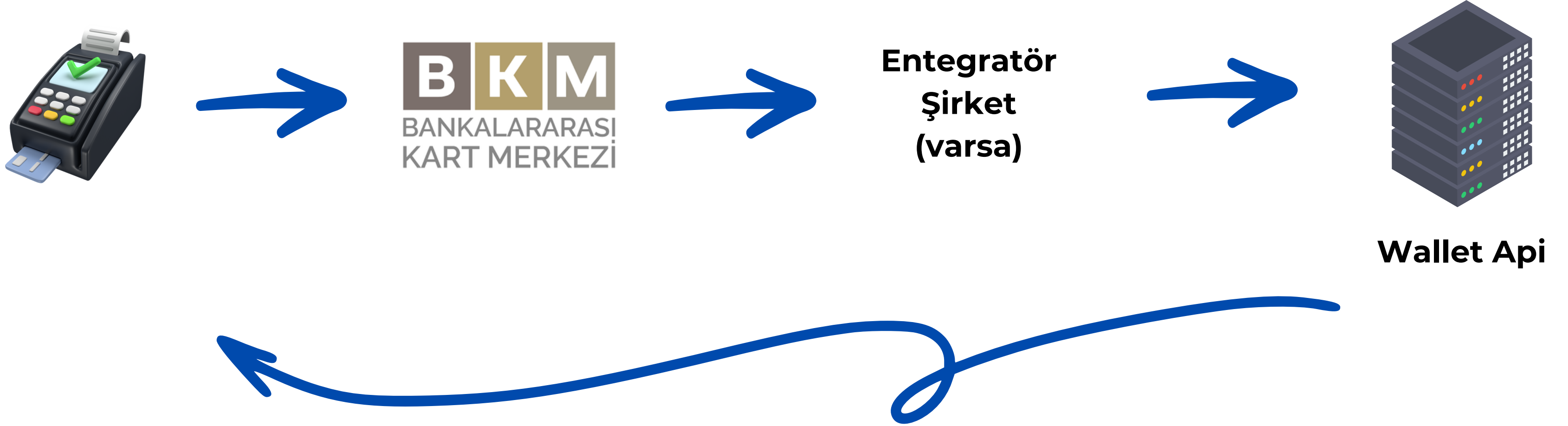
# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

- Sanal Cüzdan uygulamamız mevcut
- Fiziki basılı kartlarımız ile ödeme alıyoruz
- TCMB kurallarına uygun bir biçimde akışı bozmamamız gerekiyor



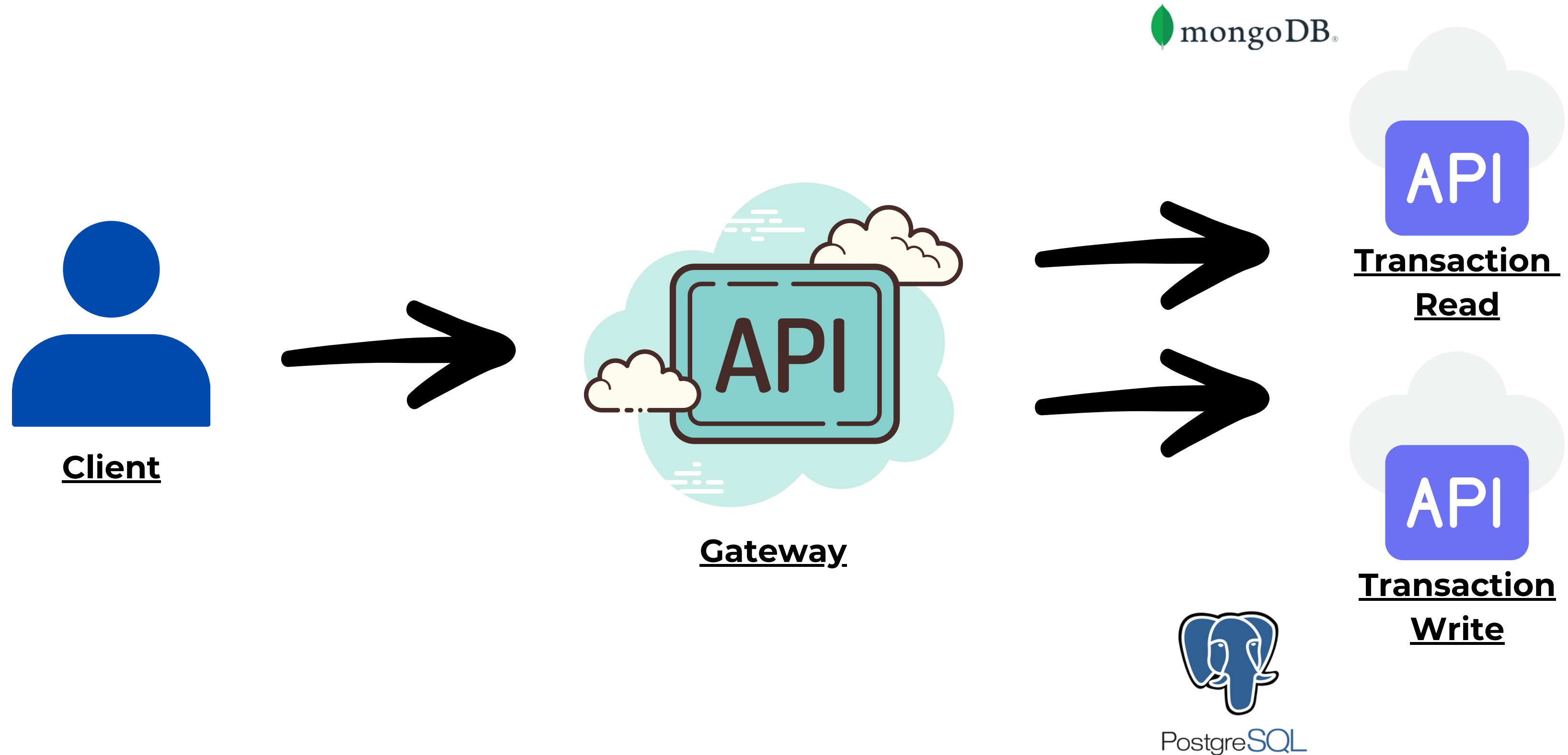


# Nasıl Çalışmaktadır?



**4 Saniye Kuralı**

# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

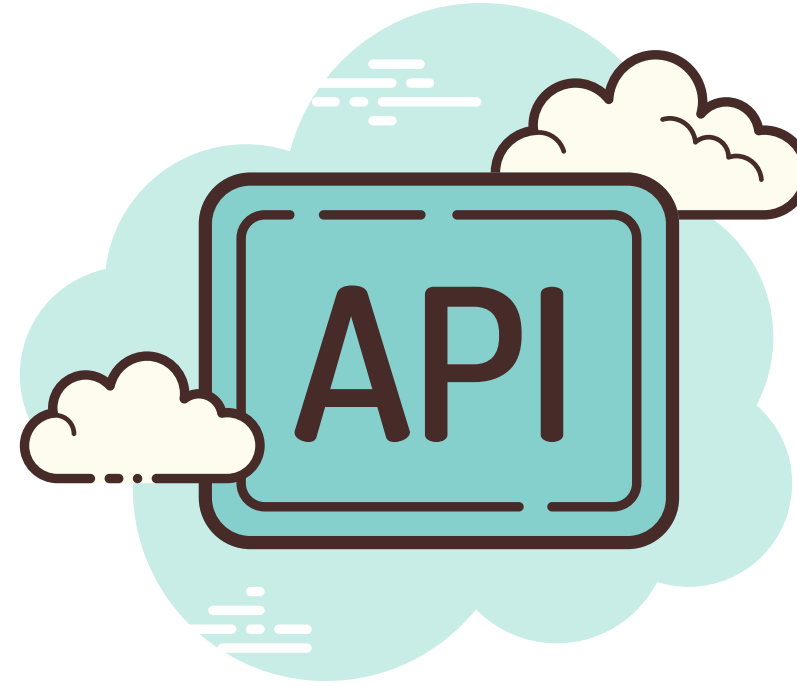


# Proje Yapısı



Transaction  
Read

.NET Core Api  
Entity  
Framework



Gateway

.NET Core Api  
YARP



Transaction  
Write

.NET Core Api  
Entity  
Framework





# Kurallara göre nasıl olmalı? - Server

## Server 1



Gateway.

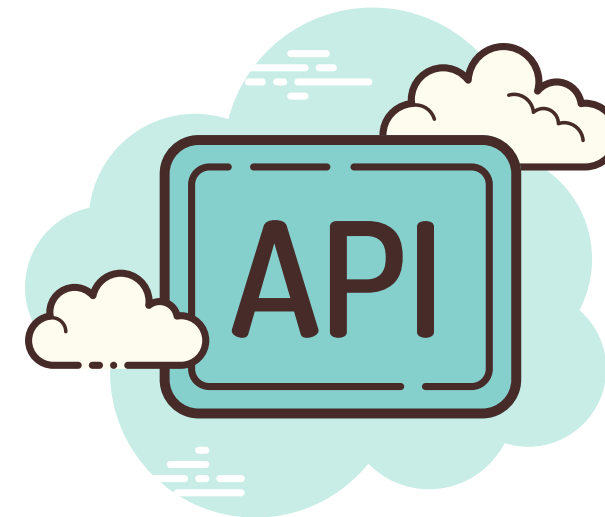


Transaction  
Read



Transaction  
Write

## Server 2



Gateway.



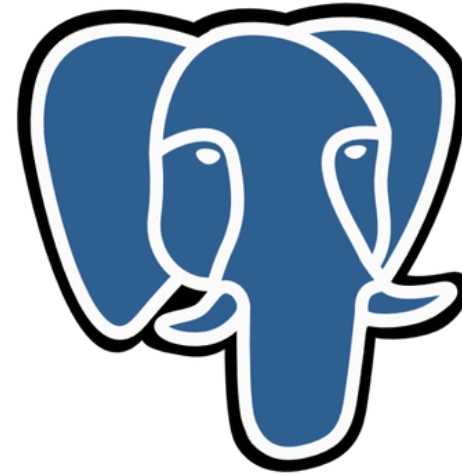
Transaction  
Read



Transaction  
Write

# Kurallara göre nasıl olmalı? - Database

## Master Node

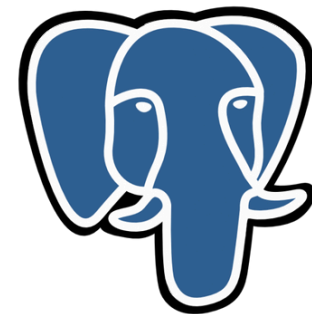


PostgreSQL



PostgreSQL

## Replication



PostgreSQL

## Replication

# Kurallara göre nasıl olmalı? - Database

Primary.



Secondary.



Secondary.



**Replica-Set Modunda Olmalıdır**



# Kurallara göre nasıl olmalı? - Kafka

Broker 1



Broker 2



Broker 3



# Kurallara göre nasıl olmalı? - Log

Logger

graylog

**Tübitak imzalı 10 yıl saklanabilmelidir**

# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

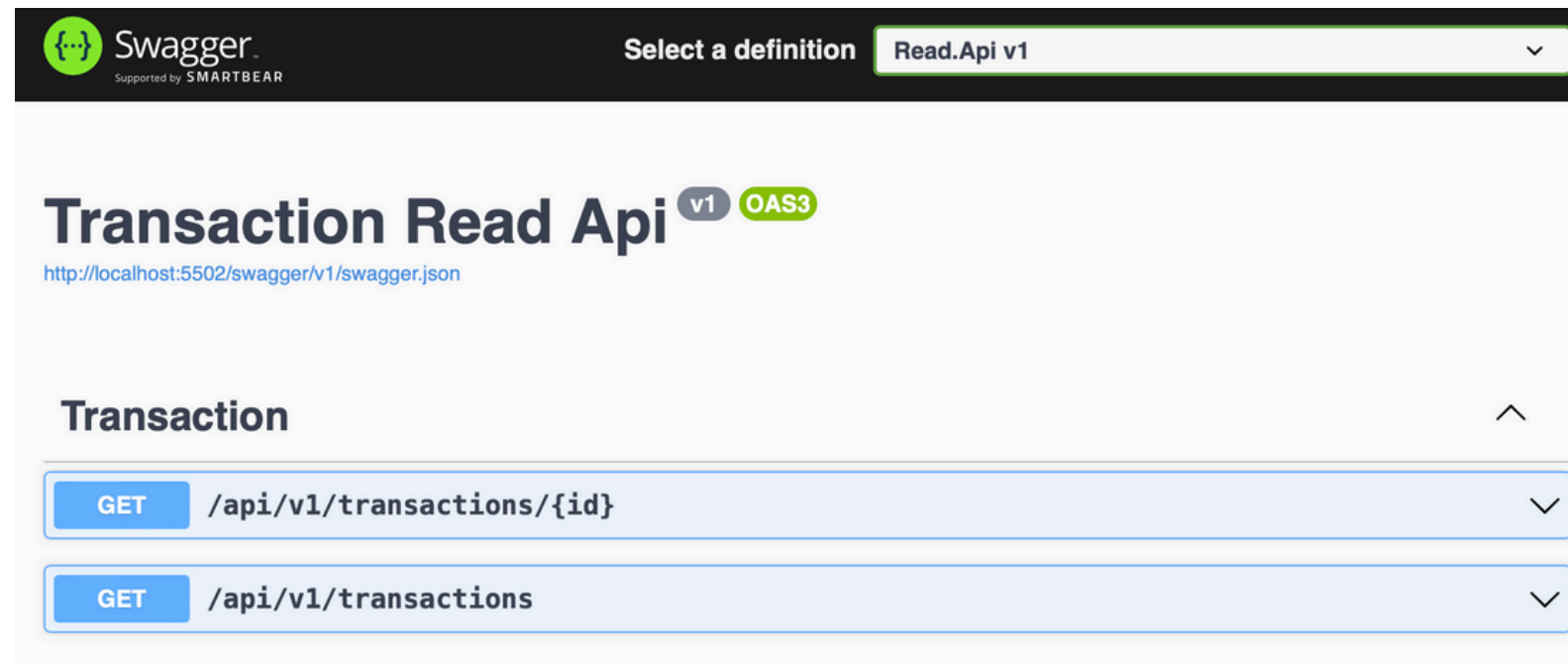




# Sanal Cüzdan Uygulamasına CQRS Uygulayalım



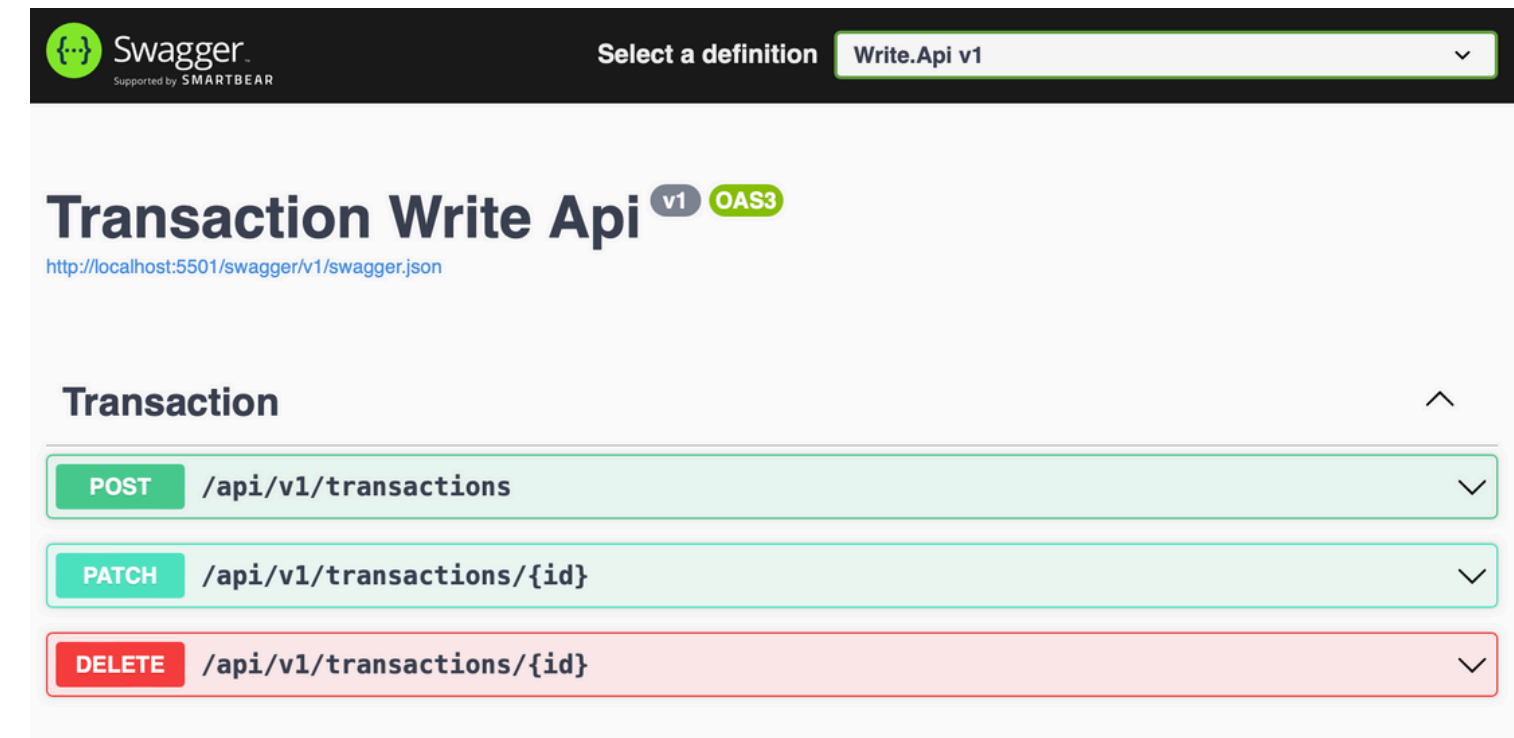
Transaction  
Read



localhost:5502



Transaction  
Write



localhost:5501

# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

## YARP



Gateway

```
YARP Config
1 "ReverseProxy": {
2   "Routes": {
3     "write_routes": {
4       "ClusterId": "write_api",
5       "Match": {
6         "Path": "/api/v1/transactions/{**catchall}",
7         "Methods" : [ "POST", "PUT", "DELETE" ]
8       }
9     },
10    "read_routes": {
11      "ClusterId": "read_api",
12      "Match": {
13        "Path": "/api/v1/transactions/{**catchall}",
14        "Methods" : [ "GET" ]
15      }
16    }
17  },
18  "Clusters": {
19    "write_api": {
20      "Destinations": {
21        "destination1": {
22          "Address": "http://host.docker.internal:5501"
23        }
24      }
25    },
26    "read_api": {
27      "Destinations": {
28        "destination1": {
29          "Address": "http://host.docker.internal:5502"
30        }
31      }
32    }
33  }
34 }
```

# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

## Job Config



```
YARP Config
1 {
2   "name": "transaction-activity",
3   "config": {
4     "plugin.name": "pgoutput",
5     "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
6     "database.hostname": "host.docker.internal",
7     "database.port": "5432",
8     "database.user": "postgres",
9     "database.password": "123456",
10    "database.dbname": "transaction_write_db",
11    "table.include.list": "public.transaction",
12    "topic.prefix": "data",
13    "decimal.handling.mode": "double",
14    "time.precision.mode": "connect",
15    "key.converter": "org.apache.kafka.connect.storage.StringConverter",
16    "key.converter.schemas.enable": "false",
17    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
18    "value.converter.schemas.enable": "false",
19    "include.schema.changes": "false",
20    "transforms.filter.condition": "value.op == 'c' || value.op == 'u' || value.op == 'd'"
21  }
22 }
```

**localhost:8083**

# Sanal Cüzdan Uygulamasına CQRS Uygulayalım

## Fake Data

```
var transactionFaker = new Faker<TransactionDto>("tr")
    .RuleFor(u => u.FromWalletId, f => f.Random.Long(min: 10000000))
    .RuleFor(u => u.ToWalletId, f => f.Random.Long(min: 10000000))
    .RuleFor(u => u.CardNumber, f => f.Finance.CreditCardNumberObfuscated())
    .RuleFor(u => u.FirstName, f => f.Name.FirstName())
    .RuleFor(u => u.LastName, f => f.Name.LastName())
    .RuleFor(u => u.Email, f => f.Internet.Email())
    .RuleFor(u => u.Total, f => f.Finance.Amount(18, 4));

var transactionList = transactionFaker.Generate(1000);
```

**.NET**  
Core



# Örneğimizi İnceleyelim

```
1 TransactionId: 1849
2 -----
3 FromWalletId: 735568650687334653
4 ToWalletId: 2840414453314588378
5 CardNumber: ****-****-****-8524
6 FirstName: Boğa
7 LastName: Sinanoğlu
8 Email: Bekecarslan19@gmail.com
9 Total: 7,90
10 -----
11 -----
12 TransactionId: 1850
13 -----
14 FromWalletId: 8126400926783529631
15 ToWalletId: 6446623710851364772
16 CardNumber: ****-****-****-2501
17 FirstName: Evin
18 LastName: Okur
19 Email: Aydemir60@gmail.com
20 Total: 15,27
21 -----
```

.NET  
Core





# Debezium ile CQRS Eđitimi





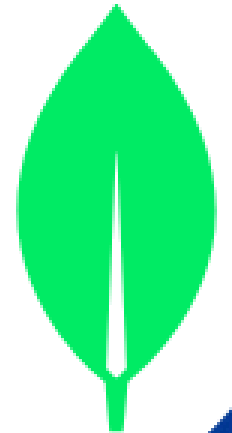
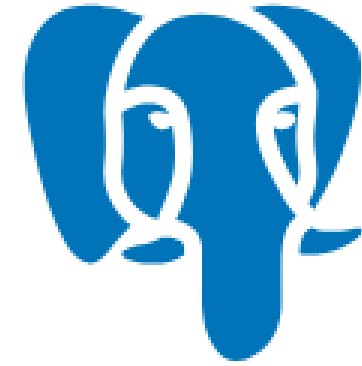
# .NET CORE WEB API İLE CQRS UYGULAMAK

- CQRS Nedir? Neden İhtiyaç Duyarız? Avantajları Nelerdir?
- Kafka Debezium Nedir? CDC Nedir?
- PostgreSQL Change Data Capture (CDC) Aktifleştirme ve Yönetimi
- Docker üzerinde Kafka Debezium Kurulumu ve Yapılandırması
- CQRS uygulayarak .NET Core Web Api projesi geliştirelim!

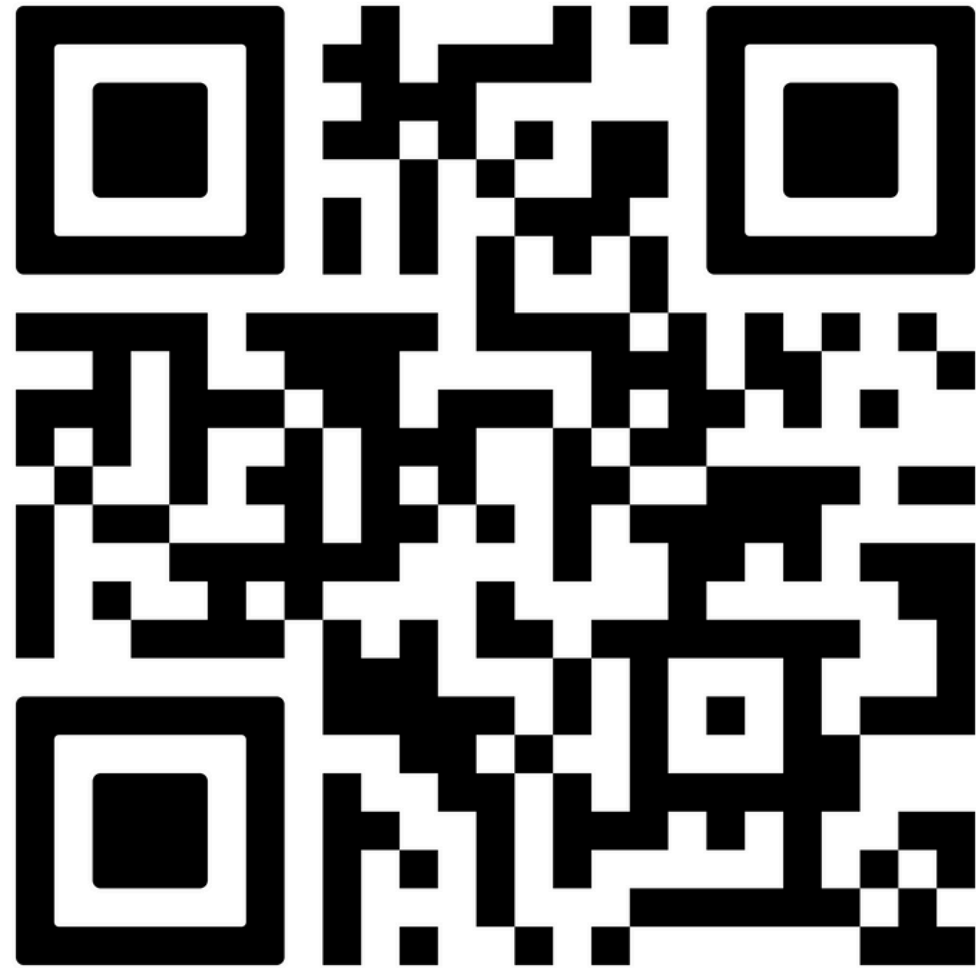
.NET  
Core



debezium



# Youtube Kanalım



# Eğitim







**Dotnet Konf 2024**

Applying CQRS with .NET Core in Financial Technologies

# SORU - CEVAP





**Dotnet Konf 2024**

Applying CQRS with .NET Core in Financial Technologies

# TEŞEKKÜRLER

