Dapper ve Mikroservisler

Emre Aydoğduoğlu
Gazi Üniversitesi
Teknoloji Fakütesi
Bilgisayar
Mühendisliği
4. Sınıf

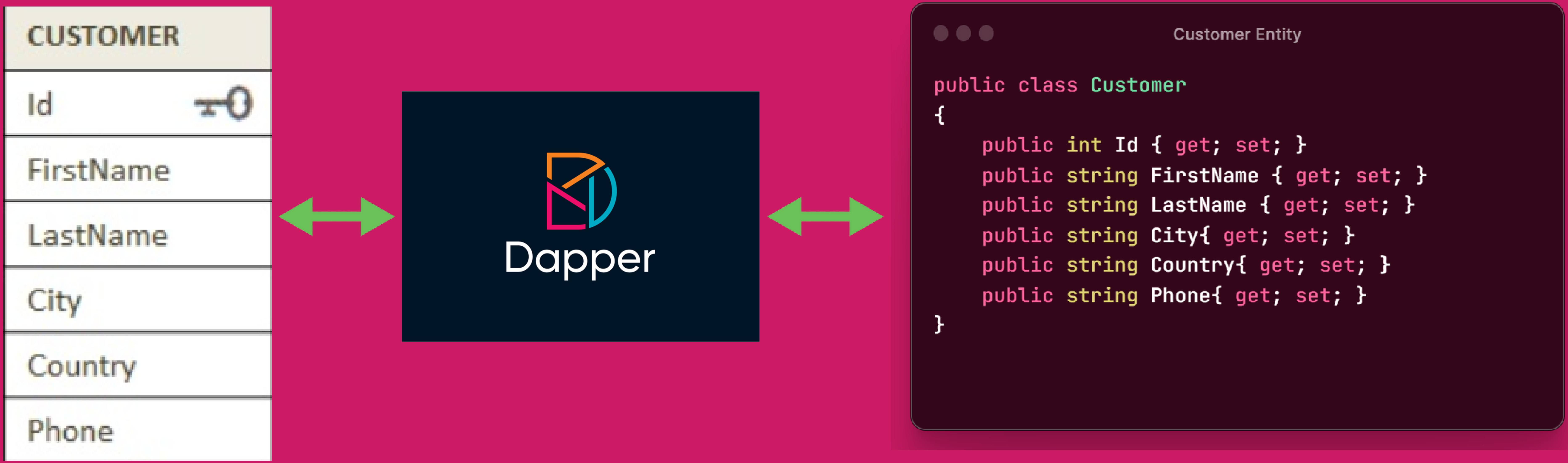
Stack Overflow

Açık Kaynak Kodlu

DAPPER

MICRO ORM FOR .NET

Object Relational Mapping



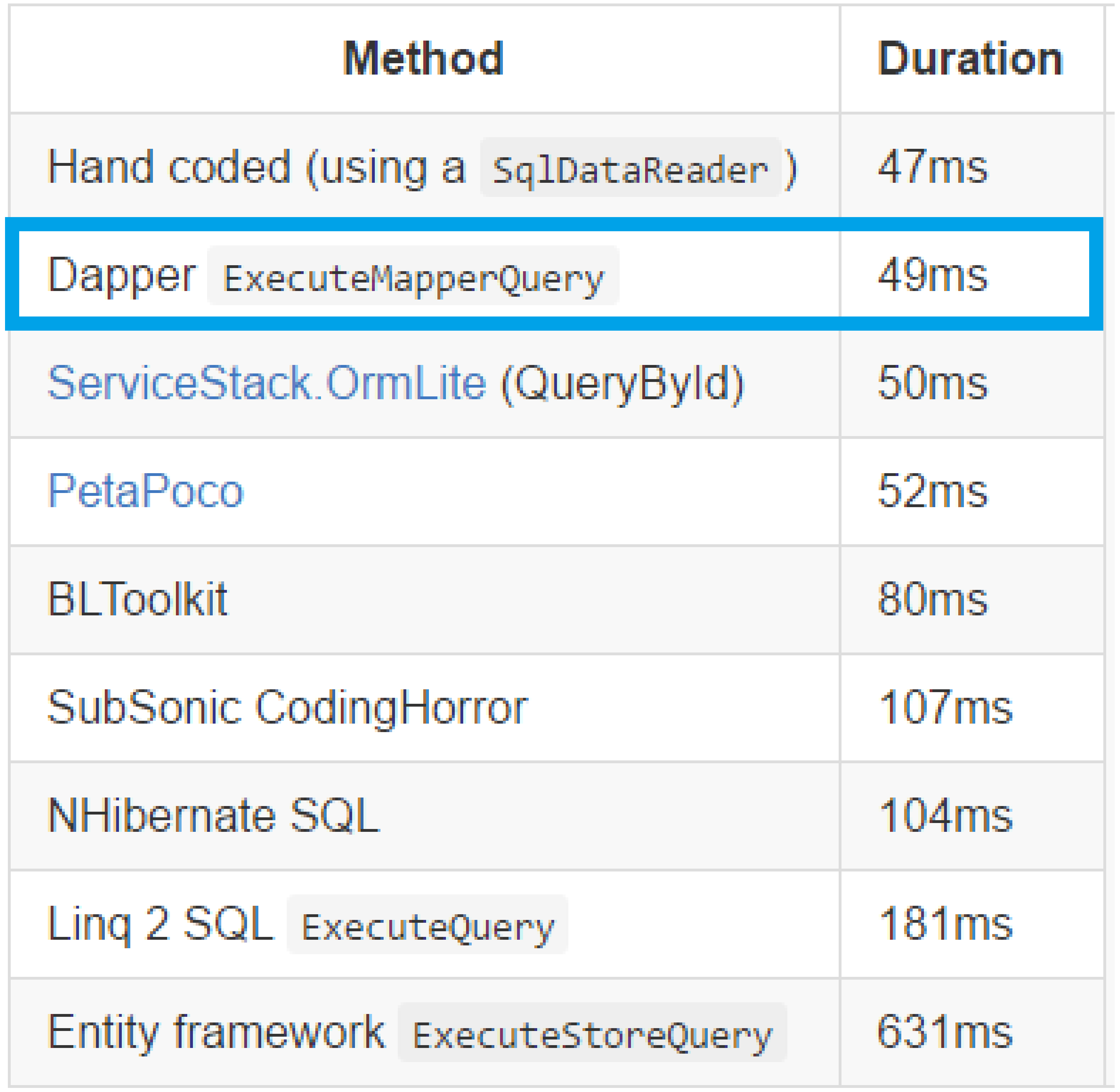
CUSTOMER
Id
FirstName
LastName
City
Country
Phone

Customer Entity

```
public class Customer
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string City{ get; set; }
    public string Country{ get; set; }
    public string Phone{ get; set; }
}
```

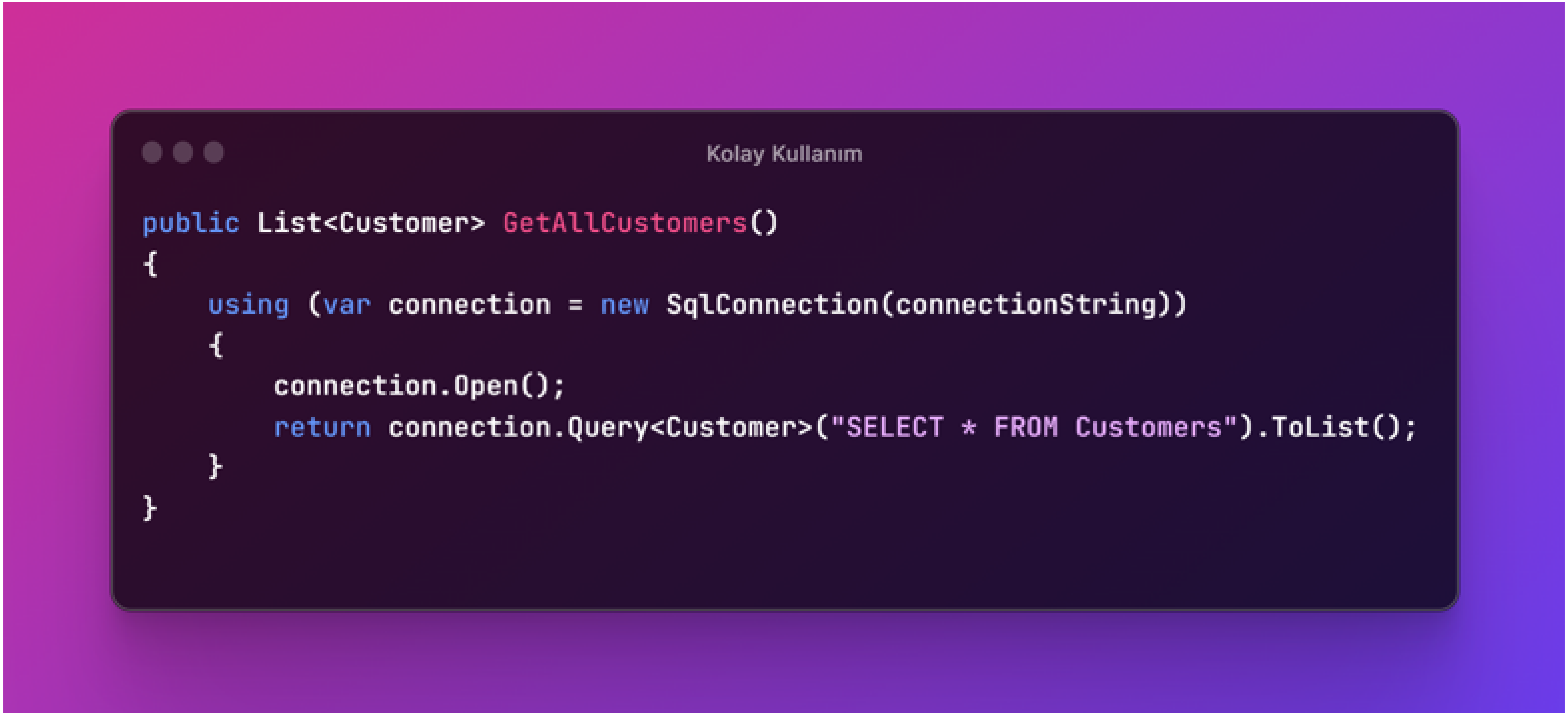
Neden Dapper?

Performans



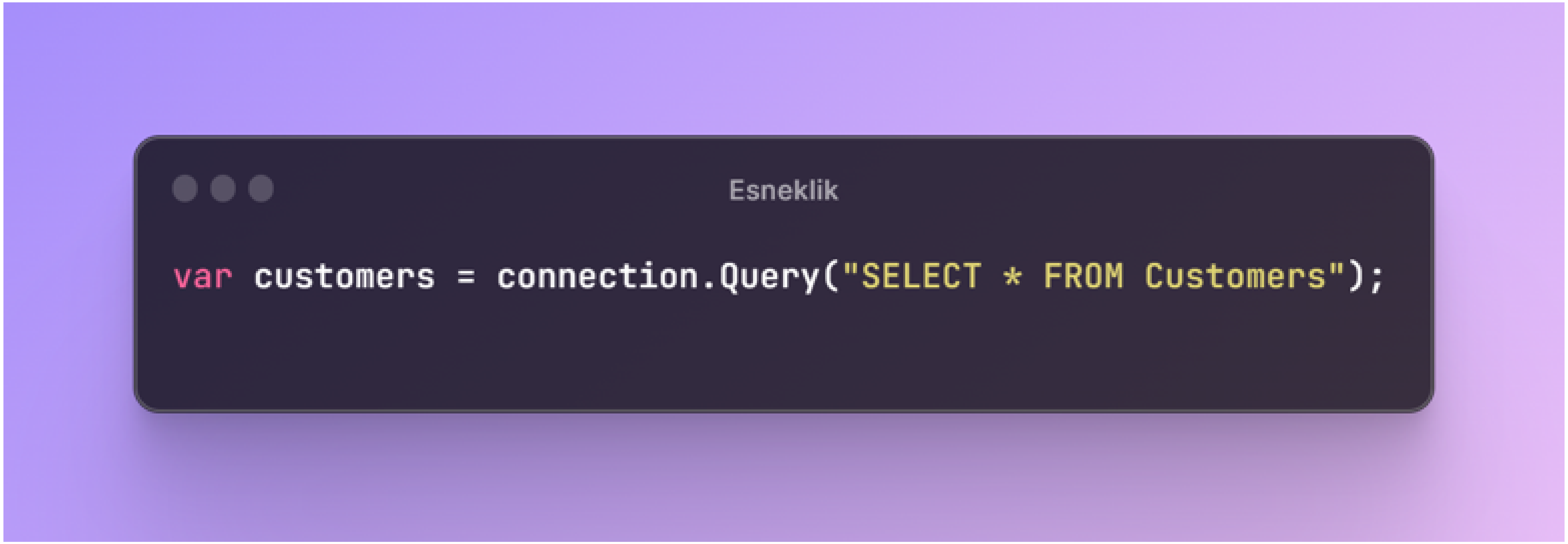
Method	Duration
Hand coded (using a `SqlDataReader`)	47ms
Dapper `ExecuteMapperQuery`	49ms
ServiceStack.OrmLite (QueryById)	50ms
PetaPoco	52ms
BLToolkit	80ms
SubSonic CodingHorror	107ms
NHibernate SQL	104ms
Linq 2 SQL `ExecuteQuery`	181ms
Entity framework `ExecuteStoreQuery`	631ms

Neden Dapper?

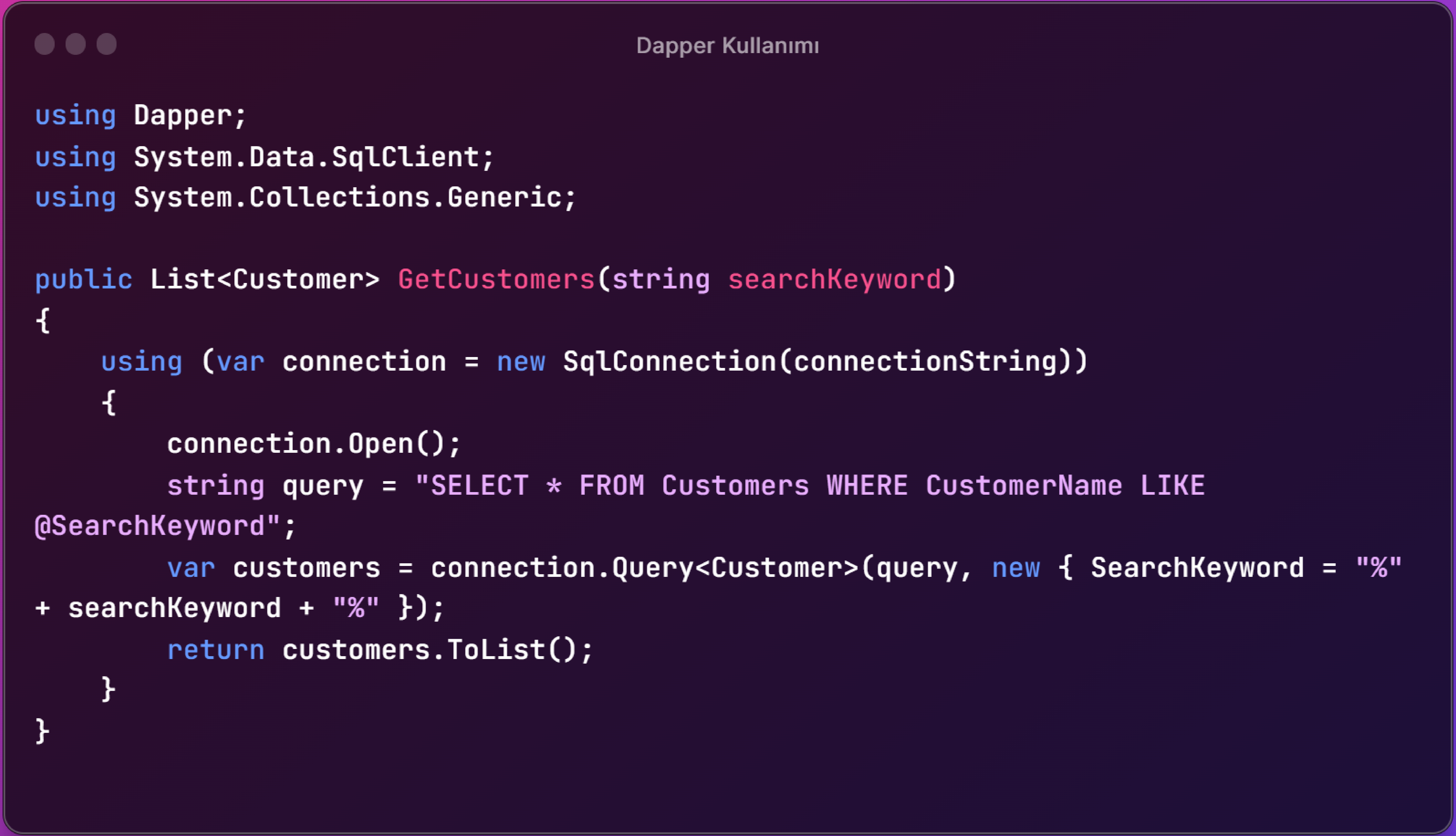


```
public List<Customer> GetAllCustomers()
{
    using (var connection = new SqlConnection(connectionString))
    {
        connection.Open();
        return connection.Query<Customer>("SELECT * FROM Customers").ToList();
    }
}
```

Neden Dapper?



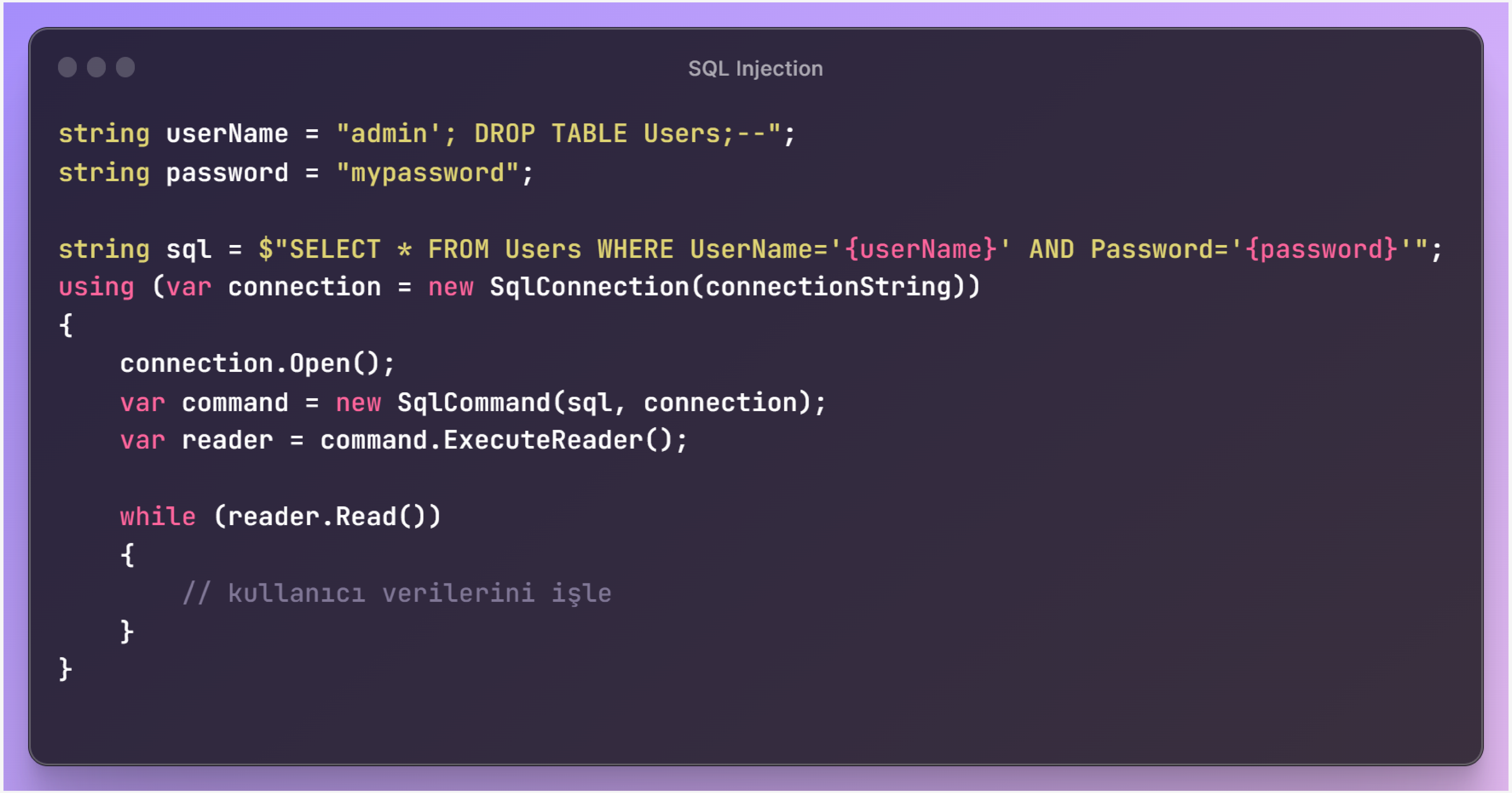
```
var customers = connection.Query("SELECT * FROM Customers");
```


Dapper Kullanımı

```
using Dapper;
using System.Data.SqlClient;
using System.Collections.Generic;

public List<Customer> GetCustomers(string searchKeyword)
{
    using (var connection = new SqlConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT * FROM Customers WHERE CustomerName LIKE
@SearchKeyword";
        var customers = connection.Query<Customer>(query, new { SearchKeyword = "%"
+ searchKeyword + "%" });
        return customers.ToList();
    }
}
```

Dezavantajlar



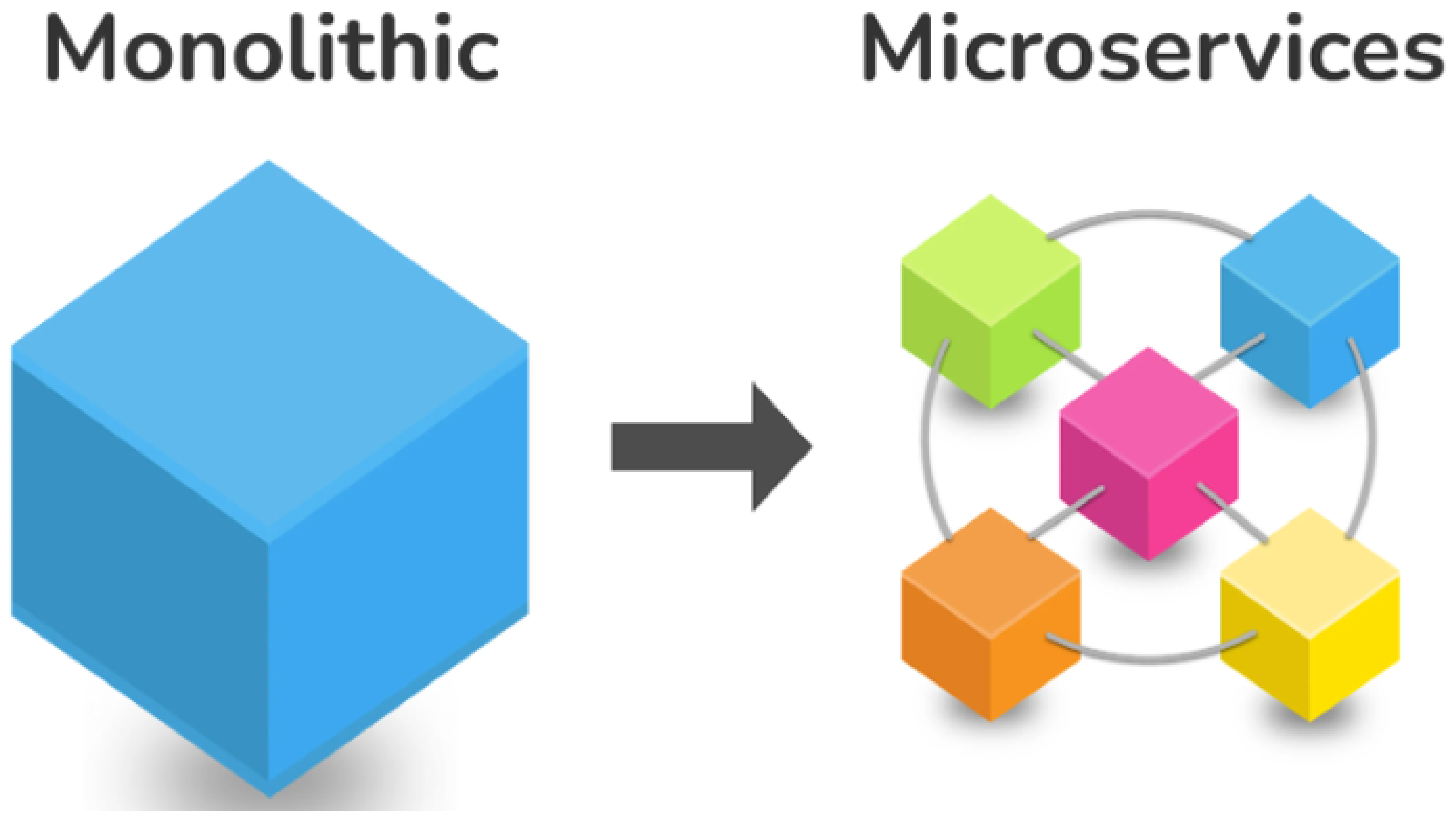
SQL Injection

```
string userName = "admin'; DROP TABLE Users;--";
string password = "mypassword";

string sql = $"SELECT * FROM Users WHERE UserName='{userName}' AND Password='{password}'";
using (var connection = new SqlConnection(connectionString))
{
    connection.Open();
    var command = new SqlCommand(sql, connection);
    var reader = command.ExecuteReader();

    while (reader.Read())
    {
        // kullanıcı verilerini işle
    }
}
```

Microservices



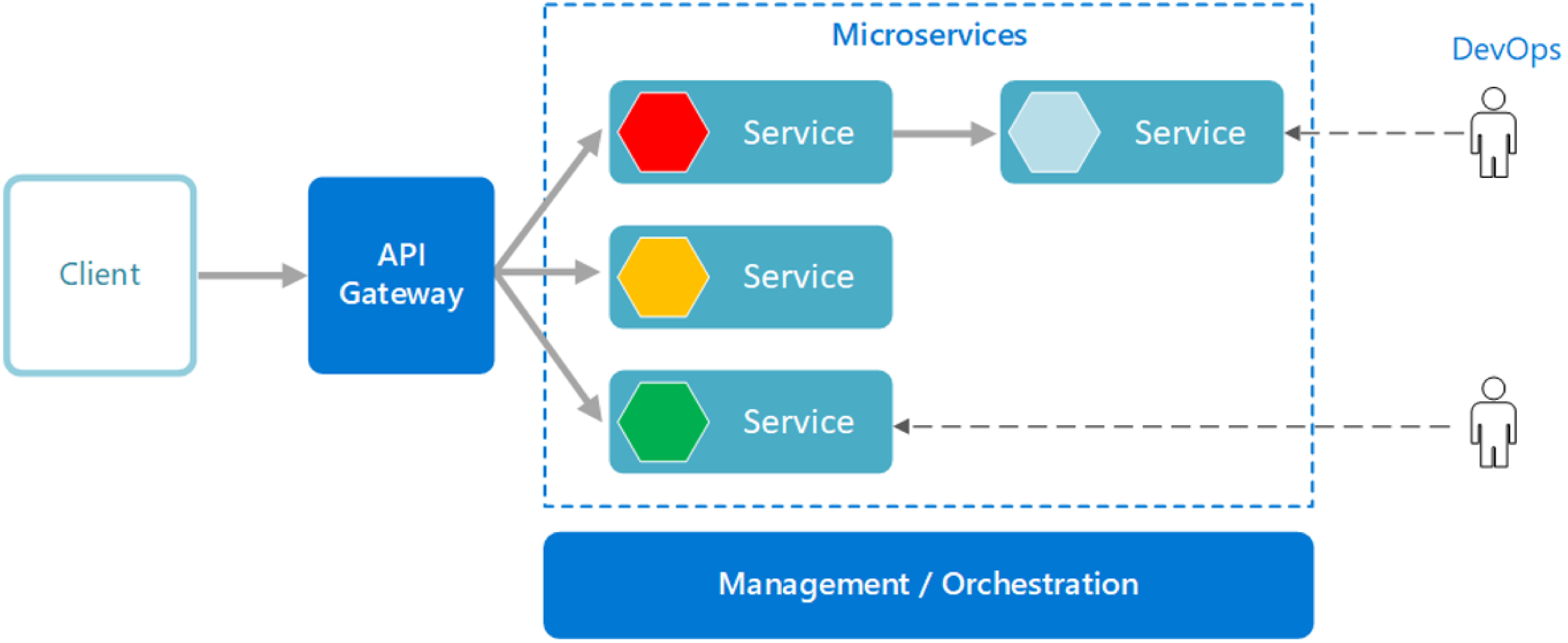
Microservice Mimarisinin Özellikleri

API'ler aracılığıyla iletişim

Karmaşıklığın azaltılması

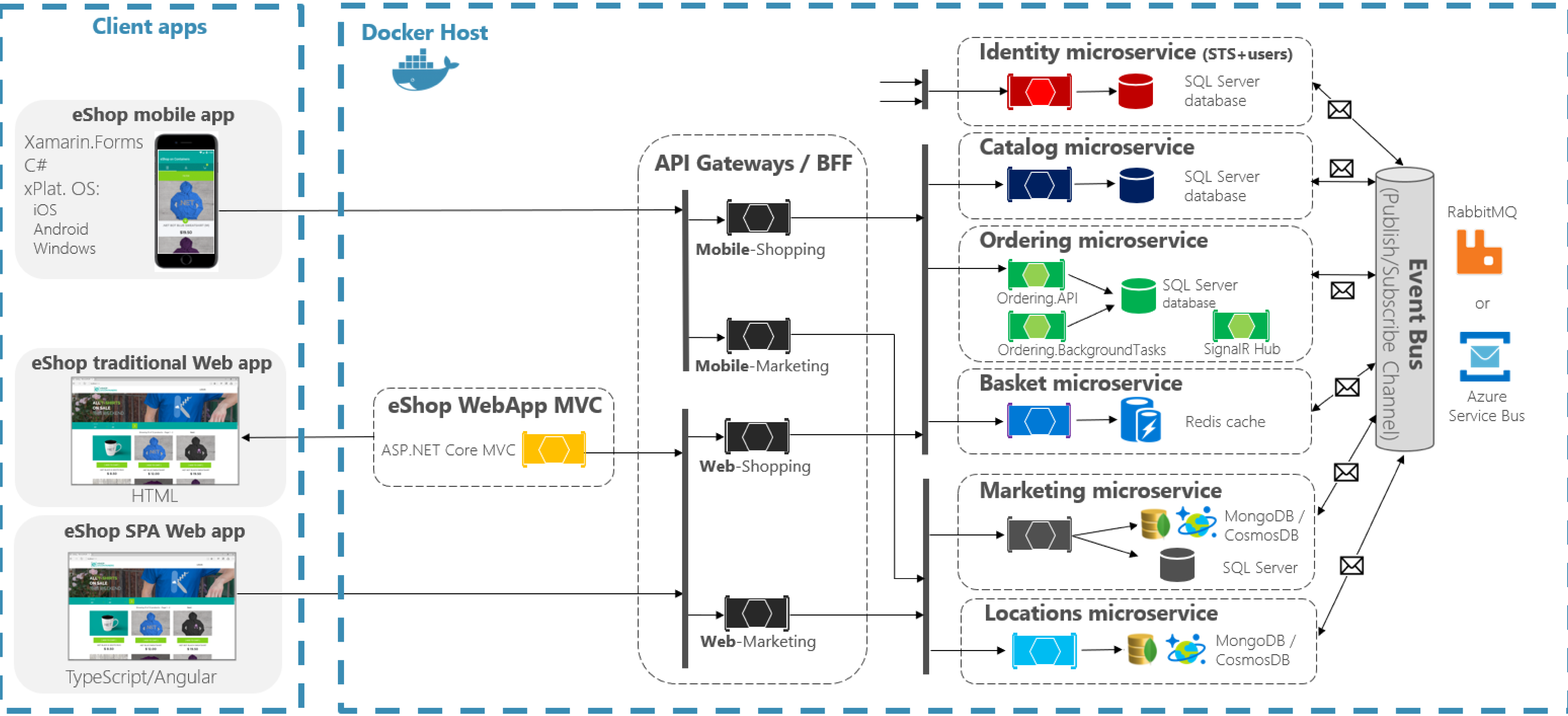
Uygulama geliştirme süreci daha hızlı ve verimli

Farklı platformlar ve teknolojiler


Client
API Gateway
Microservices
Service
Service
Service
Service
DevOps
Management / Orchestration

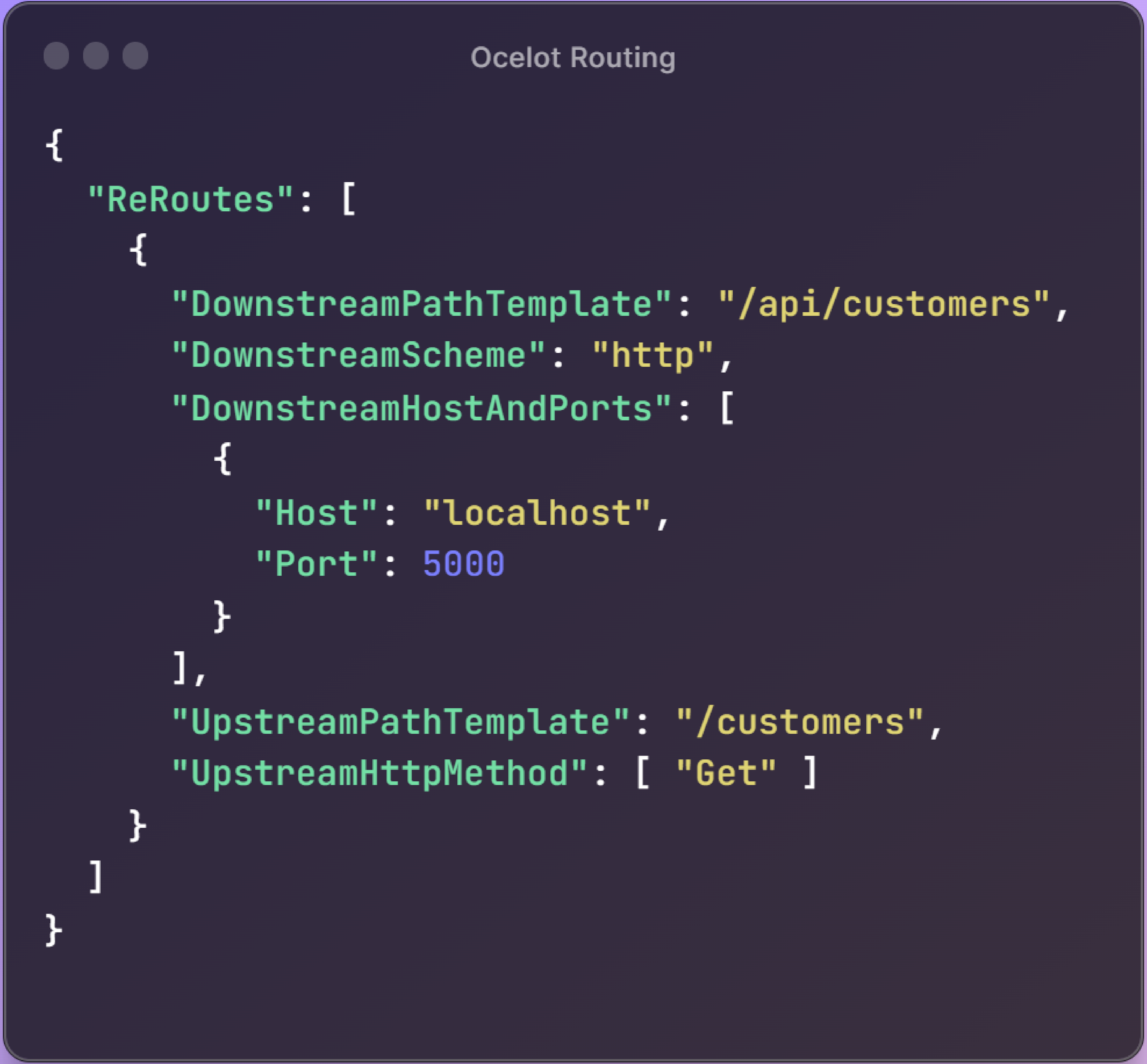
eShopOnContainers reference application

(Development environment architecture)



OCELOT

API Gateway patterni
için bir framework



Ocelot Routing

```
{
  "ReRoutes": [
    {
      "DownstreamPathTemplate": "/api/customers",
      "DownstreamScheme": "http",
      "DownstreamHostAndPorts": [
        {
          "Host": "localhost",
          "Port": 5000
        }
      ],
      "UpstreamPathTemplate": "/customers",
      "UpstreamHttpMethod": [ "Get" ]
    }
  ]
}
```

MICROSERVICES
MONOLITH