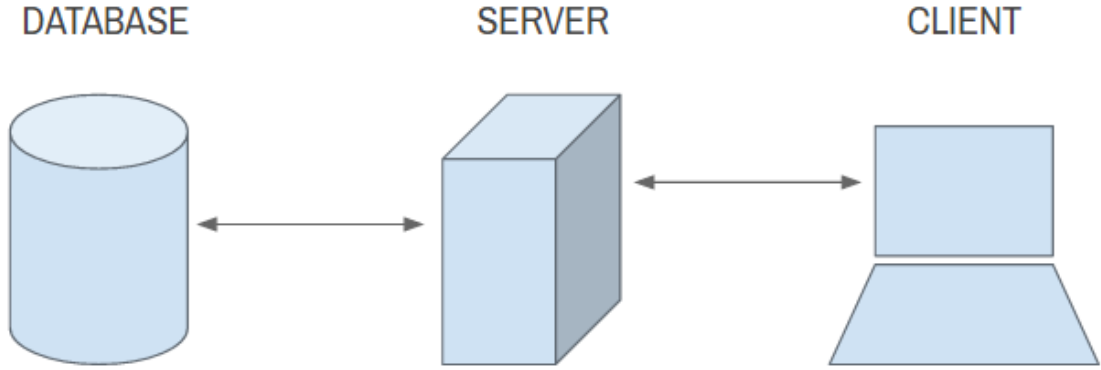


## Tekil Mimari (Monolithic Architecture)

Bir uygulamanın gerekli bütün parçalarının tek bir çatı altında tasarlandığı ve sunulduğu yazılım mimarisini ifade eder. Parçalar her ne kadar farklı özellik ve işlevselliğe sahip olsalar da bir bütün olarak çalışmaktadırlar. Tüm uygulama tek bir derleyici tarafından derlenir. Sunucu tarafında ise tek bir dosya üzerinde çalıştırılır ve sunulur. Günümüzde yazılım dünyasında en fazla tercih edilen mimari türüdür.

Üç ana katmandan oluşmaktadır.



İstemci katmanı(Client): Bu katman sunucuya ulaşarak isteklerin gönderildiği katmandır.

Sunucu (Server): Veritabanı ile iletişime geçerek talep edilen verileri alan ve istemciye sunan katmandır. Monolitik mimari kullanımında sunucu içerisinde üç ayrı katman bulunmaktadır.

Veri tabanı katmanı: Verilerin depolandığı ve istenildiği zaman sunucuya verilerin sunulduğu katmandır.

### Sunucu içerisinde bulunan katmanlar:

Presentation Layer: Son kullanıcının (istemci) gördüğü kodları içeren sunum(UI) katmanıdır.

Business Layer: Mantıksal işlem kodlarının bulunduğu katmandır.

Data Access Layer: Projenin sunucu katmanı içerisinde veri tabanı ile iletişime geçen katmandır.

### Artıları:

- Küçük ve orta ölçekli projeler hızlı ve düşük maliyetli şekilde geliştirilebilir/yönetilebilir.
- Tek aşamada uçtan uca kolayca test edilebilir.
- Tek bir dosya üzerinden yayınlandığından yayına alması kolaydır.
- Yatay olarak kopyalanarak kolayca ölçeklendirilebilir.

### Eksileri:

- Uygulama bir bütün olduğundan büyüdükçe başlama süresi artacaktır.
- Proje büyüdükçe bakımı zorlaşacaktır.
- Herhangi bir güncelleme yapılması gerektiğinde uygulamanın tamamen durdurulması ve güncellenerek tekrar yayına alınması gerekmektedir.
- Herhangi bir satırdaki yanlış kod bütün projenin anlık olarak kullanılamaz duruma gelmesine sebebiyet verir.
- Uygulama bir bütün olarak çalıştığından ölçeklendirme bütün projeyi kapsamak zorundadır.

## **Mikrohizmet Mimarisi(Microservice Architecture)**

Mikroservis mimarisi monolitik mimariden farklı olarak bir uygulamanın birbirleri ile internet üzerinden haberleşen ve kendilerine özel farklı sunucuları/veritabanları bulunan minik servisler kullanarak geliştirmeyi hedefler. Bu mimariyi kullanarak farklı amaçlar için tanımlanacak olan mikroservisler için farklı yazılım dilleri kullanılabilir.

### **Artıları:**

- Uygulama içerisinde bulunan mikroservisler ihtiyaca göre farklı yazılım dilleri ve teknolojiler kullanılarak yazılabilir.
- Uygulama içerisinde bulunan herhangi bir mikroserviste hata meydana gelmesi uygulamanın tamamını kullanılamaz kılmaz, sadece hata barındıran mikroservisin fonksiyonallitesi kullanılamaz hale gelir.
- Her bir mikroservis ihtiyaca göre ayrı ayrı ölçeklendirilebilir. Ölçeklendirme ihtiyacı duyulduğunda tüm uygulamanın ölçeklendirilmesine gerek kalmaz.
- Monolitik sistemlere oranla daha kolay bir şekilde yayına alınabilirler.
- Her bir mikroservis farklı takımlar tarafından geliştirilerek farklı alanlarda uzman ekipler elde edilebilir.

### **Eksileri:**

- Test edilmesi zordur.
- Mikroservisler arası iletişimi sorunsuz ve güvenli şekilde sürdürmek zordur.