

# Monolith ve Microservice

Bu yazıda Monolith ve Microservice mimarileri karşılaştırmaları yapıp, avantaj ve dezavantajları hakkında bilgi sahibi olacağız.

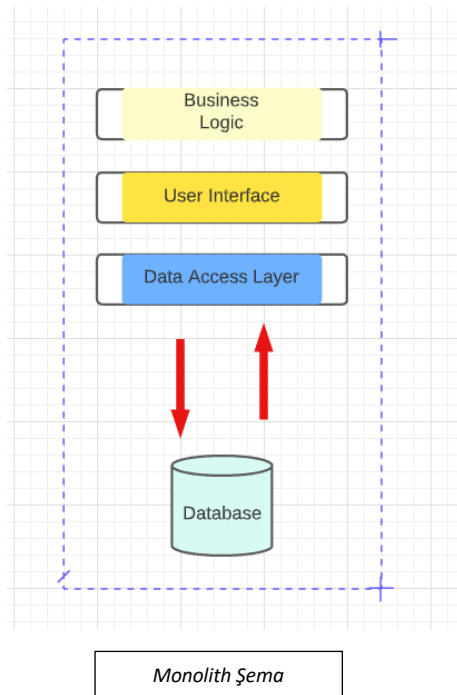
İlk olarak, Microservice mimarilerini anlamak için Monolith mimariler hakkında bilgi sahibi olmamız gerekiyor. Monolith mimari yapısı, mono kelimesini araştırdığımız zaman Latince sayılardan olan “1” anlamına geliyor. Buradan monolitik kelimesini tek parça olarak ifade edebiliriz. Zaten yazılım mimari yapısı bakımından bu anlamda kullanılıyor. Monolith mimariyi araştırdığımız zaman kendi kendine yetebilecek şekilde tasarlanan uygulamaların tek bir parça olarak oluşturulması olduğunu görüyoruz.

Monolith uygulama oluşturulurken, sistem içerisinde bulunan bileşenler birbirlerine bağlı (interdependent) olarak ve kendine yetecek bir sistem içerisinde oluşturulur. Sonucunda bir bütün olarak ortaya çıkar. Uygulama derlenirken tek bir derleme sistemine sahiptir. Uygulama ayağa kalktığı zaman tek parça olarak hareket eder. Monolith uygulamaların avantajları ve dezavantajları vardır.

Avantajlarından bahsedecek olursak, tek bir bütün olarak ortaya çıktığı için sistemin yönetilebilmesi ve geliştirilebilmesi kolaydır. Bakımı ve izlenmesi de bu bakımdan daha kolay şekilde gerçekleşebilir. Küçük çaplı geliştirilecek projelerde maliyet olarak hesaplıdır ve hızlı şekilde geliştirilebilir. Fakat gelişen teknoloji ile birlikte uygulamaların kullanıcı sayısının artması ve geliştirilen sistemlerin büyümesi ile birlikte Monolith uygulamalar avantajlarının yanında dezavantajlarını getirmiştir.

Dezavantajları anlattığımız gibi uygulamanın tek bir yapı halinde olması dolayısı ile karşımıza çıkar. Bunlar arasında, uygulama için gereken güncelleme, geliştirme gibi durumlarda sistemin tamamına müdahale edilmesi gerekir ve tekrardan derlenerek kullanıcılara ulaşması beklenir. Yani uygulamamız büyükçe yönetmesi zorlaşacak hatta geliştiricileri kısıtlayacaktır. Sistem içerisinde bulunan bağımlılıklardan kaynaklı olarak bileşenlerde yapılan bir değişiklik farklı bileşenleri etkileyebilir. Bu dezavantajlar yüzünden Monolith mimariler sürdürülebilir olmaktan çıkmıştır.

Aşağıda bir Monolith mimari şeması görüyoruz. Bu şemada görüldüğü üzere tek bir bütünlük ile çalışma mimarisi gösteriliyor. Uygulamaya ait olan tüm oluşumlar bir arada bulunuyor.



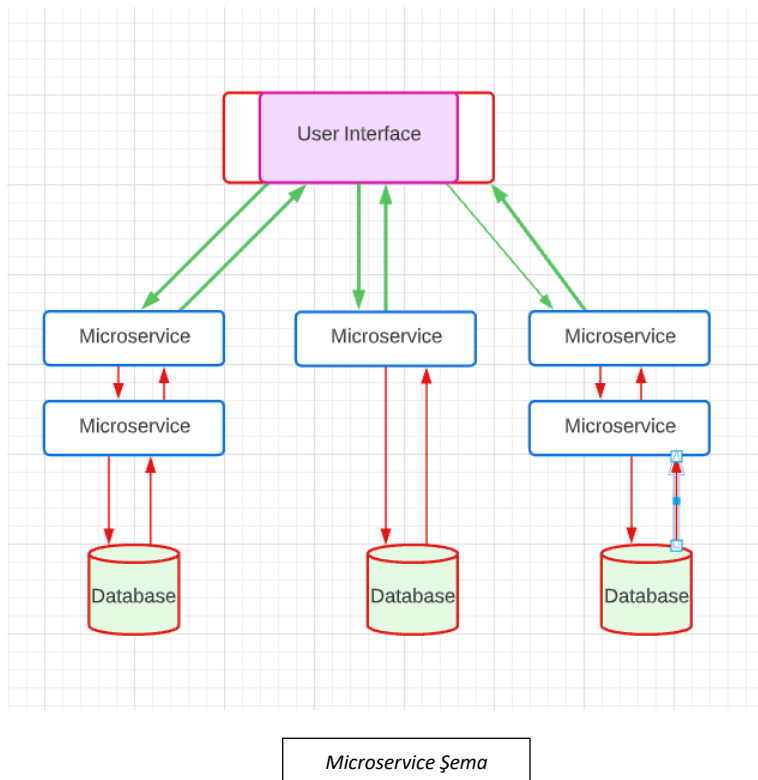
Teknolojinin gelişmesi ile birlikte büyüyen ve gelişen uygulamalar sonucunda yukarıda anlattığımız Monolith mimarilerin dezavantajları sonucunda çözüm için Microservice mimari yapısı gelişti. Microservice mimariler, geliştiricilere birbirlerinden bağımsız olarak çalışan ve birbirleriyle haberleşen servisler olarak ortaya çıktı. Microservice için kısaca küçük parçalar halinde geliştirilerek bir bütün inşa etme sanatı diyebiliriz.

Bağımsız olarak çalışırken aynı zamanda bir bütün olarak hareket ettiklerini unutmamalıyız. Her servis kendisi için geliştirilmiş ve istenen yapıya göre çalışır. Şöyle küçük bir örnek vermek gerekirse, bir okul sistemi düşünelim. Ben bu okul sistemi içerisinde öğretmenlerin verilerine ulaşmak ve onların ders programı üzerine bir iş yapmak istiyorum. Bu isteğim karşısında okulda bulunan tüm öğrencilerin veri tabanına erişmem ve orada bulunan bilgilere ulaşmam biraz saçma ve yorucu olabilir. Bunun için sistemde farklı servisler ile bunları ayırabiliriz. Böylece spesifik amacım için daha kolay erişim ve hız kazanabilirim. Bu tarz örnekleri günlük yaşantımızda kullandığımız birçok sistemde görebiliriz. Bankacılık, e-ticaret ve sosyal medya gibi platformları düşünebiliriz.

Microservice mimariler içerisinde bütün olarak çalıştığı fakat yaptıkları iş akışlarından dolayı ayrıldıkları için birçok avantajı beraberinde getirir. Örneğin bir Microservice tarafında çıkan hata, güncelleme vb. olaylar karşısında uygulamanın tamamı etkilenmeyeceği için kısa sürelerde müdahale edilme ve bu sırada hala çalışabilmesi ile sürdürülebilirlik örneği sağlar.

Microservice avantajlarından bahsedecek olursak, uygulamanın yukarıda anlatılanlardan dolayı bakımı kolay ve hızlı şekilde gerçekleşir. Uygulamaya yeni özellik ekleme, çıkarma veya güncelleme gibi ihtiyaçları daha kolay yapabiliriz. Ekip çalışmalarında kolaylık sağlar. Farklı servisler farklı dil ve frameworkler ile geliştirilebilir. İyi ve güzel olan herşey gibi Microservice mimarilerinde zorluk çıkarttığı dezavantajlarının olduğu unutulmamalıdır.

Microservice mimarilerde birden fazla veri tabanı ile çalışılacağı için işlemler zorlaşacaktır. Bunun yanında kurulan sistemin yönetilmesi ve izlenmesi kolay olmayacaktır. Test etme süreçleri artacaktır. Aşağıdaki şekilde bulunan şema ile Microservice mimarilerin çalışma mantığı anlatılmak istenmiştir.



Geliřtirme yaparken sırf yapmak için deęil ihtiyalarımız doęrultusunda ilerlemeliyiz. Yani sadece tek bir mimariye baęımlı olmamak gerekir. İhtiyalarımıza yönelik mimariyi semek birok konuda bizlere kazanç saęlar.