

Monolithic Mimarisi

Bir sistemin, nesnenin veya olgunun tek bir para olarak tasarlanmasıdır. Yani kod tabanı, iş mantığı ve diğ er paraların hepsinin birbirine bağılı ve birbirine bağımlı olduğı bir sistemden oluşur. Bir projenin tüm işlevleri tek bir kod tabanında mevcutsa, bu uygulama monolitik uygulama olarak bilinir.

Tanım olarak, monolitik uygulamalar, "iş için doğıru aracın" kullanılabilirliğini sınırlayabilen tek bir geliştirme yığını (yani, JEE veya .NET) kullanılarak uygulanır.

Monolitik uygulamaların avantajları:

- Yalnızca tek bir jar/war dosyası dağıtıldığından dağıtılması daha kolaydır.
- Mikro hizmet mimarisine kıyasla geliştirmesi nispeten daha kolay ve basittir.
- Ağ gecikmesi ve güvenlik sorunları, mikro hizmet mimarisine kıyasla nispeten daha azdır.
- Daha kolay hata ayıklama ve test etme özelliğine sahiptir.
- Hizmetleri belirlemek ve geliştirmek için yetenekli geliştiricilerin gerekli olduğı mikro hizmetlere göre geliştirmesi kolaydır.

Monolitik uygulamaların dezavantajları

- Uygulama büyüdük e yeni özellik geliştirilmesi ve mevcut kodun bakımı zorlaşır.
- Projede çalışan ekip sayısının ve çalışan sayısının artması ile geliştirme ve bakım daha güç hale gelir.
- Uygulamada yapılan küçük bir değışiklikte bile bütün uygulamanın deploy olması gerekir.
- Birbirlerine olan bağımlılıklarından dolayı, bir fonksiyonalitede yapılan değışiklik diğ er yerleri etkileyebilir.
- Zamanla boyut olarak çok büy r ve bu nedenle yönetilmesi zorlaşır.
- Küçük bir değışiklik için bile tüm uygulamayı yeniden dağıtmamız gerekiyor.
- Uygulamanın boyutu arttık a, başlatma ve dağıtım süresi de artar.
- Projeye katılan herhangi bir yeni geliştirici için, sorumluluğı tek bir işlevsellikle ilgili olsa bile, büyük Monolitik uygulamanın mantığını anlamak çok zordur.

- Monolitik uygulamalar, tek bir geliştiricinin (veya geliştirici grubunun) uygulamanın tamamını anlamadığı bir durum olan "büyük bir çamur topuna" dönüşebilir.
- Monolitik uygulama yapıtlarının tekrarlanan dağıtımında operasyonel çeviklik elde etmek zordur.

Microservice Mimarisi

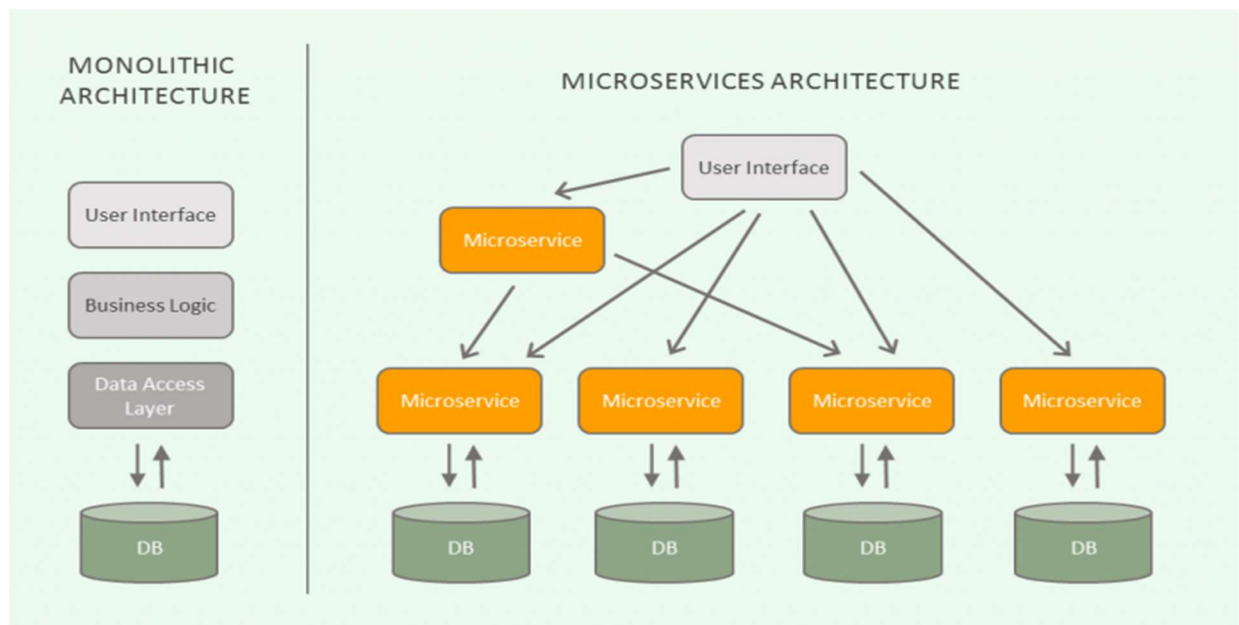
Birbirinden bağımsız olarak çalışan ve birbirleriyle haberleşen bireysel servislerdir. Uygulamanın, HTTP gibi hafif protokoller kullanarak birbirleriyle doğrudan iletişim kuran daha küçük servislerden oluştuğu bir mimari geliştirme stilidir.

Bir **mikroservice yaklaşımı** kullanan DevOps ekipleri, uygulamaları tek bir toplu birim olarak göndermek yerine hizmetleri işlevsel API'lere böler.

Kullanımı kolay API arabirimleri, hizmetleri temel işlevlerle birbirine bağılayarak uygulamaların iletişim kurmasına ve veri paylaşmasına olanak tanır.

Microservice aşağıdaki faydaları beraberinde getirir.

- Kolay testedilebilirlik
- Esnek bağlantı
- Bağımsız deploy edebilme
- Çıktıya odaklanma



Mikroservice, Kubernetes ve Docker gibi kapsayıcı tabanlı düzenleme platformları veya AWS Lambda, Azure İşlevleri ve Google Bulut İşlevleri gibi bulutta yerel bir hizmet olarak işlev (FaaS) teklifleri kullanılarak çalıştırılır.

Mikroservice uygulamaların avantajları

- Boyut olarak nispeten daha küçük olduğu için yönetimi kolaydır.
- Mikro hizmetlerden birinde herhangi bir güncelleme varsa, yalnızca o mikro hizmeti yeniden dağıtmamız gerekir.
- Mikro hizmetler bağımsızdır ve bu nedenle bağımsız olarak dağıtılır. Başlangıç ve dağıtım süreleri nispeten daha kısadır.
- Yeni bir geliştiricinin, tüm sistemi değil, üzerinde çalışacağı işlevselliği sağlayan yalnızca belirli bir mikro hizmeti anlaması gerektiğinden projeye dahil olması çok kolaydır.
- Her mikro hizmet, iş gereksinimlerine göre farklı teknolojiler kullanabilir.
- Belirli bir mikro hizmet bir hata nedeniyle çökerse, diğer mikro hizmetleri etkilemez ve tüm sistem bozulmadan kalır, kullanıcılara başka işlevler sağlamaya devam eder.

Mikroservice uygulamaların dezavantajları

- Dağıtılmış bir sistem olduğundan, monolitik uygulamalardan çok daha karmaşıktır. Mikro hizmetlerin sayısındaki artışla karmaşıklığı artar.
- Nitelikli geliştiricilerin, mikro hizmetleri tanımlayabilen ve aralarındaki iletişimlerini yönetebilen mikro hizmet mimarisiyle çalışması gerekir.
- Mikro hizmetlerin bağımsız dağıtımını karmaşıktır.
- Mikro hizmetler, birbirleriyle etkileşime girmeleri gerektiğinden ağ kullanımı açısından maliyetlidir ve tüm bu uzaktan aramalar ağ gecikmesine neden olur.
- Ağ üzerinden hizmetler arası iletişim nedeniyle mikro hizmetler, monolitik uygulamalara göre daha az güvenlidir.
- Kontrol birçok mikro hizmet üzerinde aktığından ve hatanın neden ve tam olarak nerede meydana geldiğini belirtmek zor bir iştir.