Farklı Mobil Platformlar Üzerindeki Servis Tabanlı Mimari (SOA) Yaklaşımı: Elektronik Uçuş Çantası Vaka Çalışması

Emre Barış Aksu, Cem Gündüz, Emrah Ayanoğlu

eaksu@stm.com.tr, cgunduz@stm.com.tr, eayanoglu@stm.com.tr

Özet: Servis Tabanlı Mimari günümüzde birden çok katmanlı, farklı lokasyonlu yazılımların ve yazılım parçalarının entegrasyonu için gerekli yazılımmimarilerininvazgeçilmezseçeneklerinden biri haline gelmiştir. Genellikle farklı mobil platformlar üzerinde yazılım geliştirilirken sunucu tarafında; merkezi ya da dağıtımlı bir mimarikullanımıgerçekleştirilmektedir. Bu tip durumlarda, mobil platformların tamamen bağımsız ve sadece istemci taraflı mimariyle çalışabilmesi servis tabanlı mimari kullanımıyla mümkün olmaktadır. STM A.Ş olarak geliştirmiş olduğumuz farklı mobil platformlar üzerinde çalışan Elektronik Uçuş Çantası projesinde servis tabanlı mimari kullanılarak farklı mobil platformların tamamen sunucu tarafından bağımsız olarak çalışması sağlanmıştır.

Anahtar Sözcükler: Android, iOS, Servis Tabanlı Mimari, Elektronik Uçuş Çantası, Platform Bağımsız, SOA

1.Giriş:

Günümüzde farklı platformlar üzerinde çalışan elektronik cihazların (bilgisayarlar, akıllı telefonlar, otomobiller, elektronik ev eşyaları v.b) birbirleriyle ortak protokoller üzerinden haberleşmeleri beklenmekte ve gerekmektedir. Bu gereklilik cihazların üzerinde çalışan yazılımların da farklı katmanlar içeren mimarilerin ortaya çıkmasına sebep olmuştur.

Mimarilerde kullanılan katmanların daha da fazlalaşması ile bu cihazların birbirleriyle bağlantılarının ve veri aktarımlarının daha bağımlı bir şekilde çözümlenmesine çalışılmaktadır. Bununla birlikte birbirlerine daha bağımlı hale gelen cihaz veri aktarım protokollerinin meydana getirdiği sorunlar da ortaya çıkmaktadır. Örnek olarak; farklı cihazlardaki yazılım sistemlerinde meydana gelecek değişiklikler ile diğer elektronik cihazlarla olan veri aktarımında da farklılaşmalar oluşabileceğinden, diğer cihazlardaki yazılım sistemlerinde de değişiklikler yapmak zorunda kalabilmekteyiz. Bu nedenle platformlardan bağımsız olarak yazılım geliştirme faaliyetlerini gerçekleştirmek oldukça zorlaşmaktadır.

Servis Tabanlı Mimari, iş süreçlerine katılan değişik yazılım sistemlerinin birbirleriyle haberleşmeleri üzerine geliştirilmemiş olmalarına rağmen aynı veri formatında haberleşerek değişik cihazlardaki değişik yazılım sistemlerinin bütünleştirilmesini sağlamaktadır. Örnek olarak bir akıllı mobil cihazdaki yazılım sistemiyle, bir web tabanlı yazılım sistemi ve bir tanktaki yazılım sisteminin haberleşmesinin birleştirilmesi servis tabanlı mimariyle gerçekleştirilirse, bu cihazlar üzerinde kullanılan farklı yazılım sistemlerinin de bağımsız olması sağlanabilir.

Farklı platformlar üzerinde çalışan, bağımsız yazılım sistemlerinin günümüzdeki en iyi örnekleri farklı mobil işletim sistemlerinde geliştirilen yazılım sistemleridir. Bu mobil cihazlardaki yazılım sistemlerinin bağımsız geliştirilmesi ve birbirleriyle gerçekleştirdiği haberleşmeler servis tabanlı mimariyle sağlandığı takdirde, bağımsız yazılım sistemlerine sahip olmak mümkün olmaktadır.

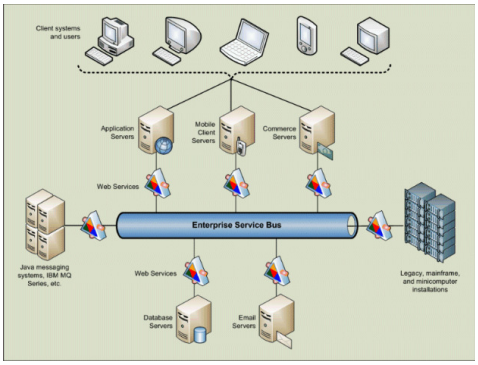
Makalenin servis tabanlı mimari detaylı bir şekilde anlatılacaktır. Ardından, farklı mobil platformlar üzerinde servis tabanlı mimariler nasıl gerçekleştirilebilir sorusu cevaplanmaya çalışılacaktır ve son olarak, firmamızın vaka çalışması olan elektronik uçuş çantası mimari çözümüyle anlatılacaktır.

Service Oriented Architecture (SOA):

Günümüzdeki rekabet ortamının getirdiği temel sorunlar olan pazarda etkin kalabilme ve çabuk değişebilme sorunları, bilişim teknolojilerinde ve özellikle de yazılım sistemleri için mevcut en büyük problem olarak gözükmektedir. Bu nedenle yapıların esnek bir hale dönüştürülmesi kaçınılmaz bir sondur. Ayrıca birbirine entegre olan yazılım sistemlerinin ve birden fazla cihazlarda ve platformlarda çalışan yazılım sistemlerinin fazlalaşması, bu esnekliğin daha da artmasının önemini ortaya koymaktadır.

İşte tam bu noktada Servis Odakli Mimari(SOA) yaklaşımı ortaya çıkmaktadır. SOA'da önemli olan farklı platformlardaki yazılım sistemlerinin farklılıkları

değil bunların birbirleriyle entegrasyonudur. Yani, buradaki durum şu şekilde özetlenebilir: her bir yapı kendi içerisinde karmaşık ve birbirinden farklı bir mimariye sahip birer yazılım sistemi içerse de, önemli olan yazılım sistemleri arasındaki entegrasyon sürecidir. Bu anlatımlardan sonra artık SOA'yı tanımlamak mümkündür: SOA, iş fonksiyonlarınınbirlikteçalışmayıdestekleyecek esneklikte ve tekrar kullanımı geliştirecek şekilde, iyi-tanımlanmış ve gevşek-bağlı bileşenler halinde oluşturulduğu bir sistem mimarisidir.



Resim 1: Servis Tabanlı Mimari

SOA'nın faydalarını sıraladığımızda; öncelikle farklı yazılım sistemlerinin birbirleriyle bağlarını en aza indirger(loose-coupling), böylece daha esnek bir mimariye sahip olunur. Ayrıca SOA ile daha önce geliştirilen bir servisin farklı yazılım sistemlerinde de kullanılması mümkün olur. Bilinen en önemli özelliği ise, çoklu platform entegrasyonuna imkan sağlamasıdır. Mesela, elimizdeki bir yazılım sisteminin barındığı işletim sistemleri Windows, MacOS, Linux, Android, iOS olabilir. Bu platformlarda geliştirilebilecek yazılım sistemleri arasındaki entegrasyon SOA yapısıyla tamamen mümkün olabilmektedir. Aynı zamanda farklı karmaşık yapıları ve karmaşık servisleri bir arada entegre ederek daha basite indirgemek SOA ile mümkün olmaktadır. Son olarak, SOA'da her bir servis kendi işinden sorumlu olduğu için uygulamalarda daha iyi performans, kolay bakım, rahat okunabilirlik gibi birçok özelliği de beraberinde getirir.

Mobil Platformlar Üzerinde SOA:

Mobil platformlarda geliştirilen yazılım sistemlerinin temel sorunlarını incelediğimizde; özellikle mobil platformlarda daha az kaynak kullanımının mevcut olduğunu görmekteyiz. Ayrıca, piyasada farklı farklı mobil platformların oluşması da bir diğer büyük problemdir. Mesela İOS, Android, Windows Mobile gibi farklı platformların bulunması esnek yazılım sistemleri geliştirmek için oldukça büyük bir sorun teşkil etmektedir. Ayrıca, ölçeklenebilir bir yazılım sistemini geliştirmek mobil sistemlerin kısıtlı kaynaklarıyla oldukça zordur.

Bu nedenlerle mobil yazılım sistemlerinde SOA'yı kullanmayı düşündüğümüzde SOA'nın mobil yazılım sistemleri üzerinde birçok avantajı olduğu görülebilir. Öncelikle SOA'nın sağlamış olduğu farklı platformların entegrasyonunu kolaylaştırma yeteneği, mobil yazılım sistemlerinin de esnek bir hale dönüşmesini sağlamaktadır. Ayrıca, SOA sayesinde farklı mobil platformlar arasında entegrasyonu sağlamak da mümkün olabilmektedir. Kısıtlı bir kaynağa sahip olan mobil cihazlar, SOA altyapısıyla kaynaklarını en iyi şekilde kullanabilen cihazlara dönüşmektedir. Örnek olarak, Instagram gibi bir fotoğraf çekme uygulamasında, mobil cihazların sadece fotoğraf çekme özelliklerinden yararlanıp diğer fotoğraf işlemleri ve sosyalleşme gibi işlemlerin tamamen servisler tarafından gerçekleştirilmesi sayesinde mobil cihazların kaynaklarını en yüksek şekilde etken ve etkin olarak kullanmak SOA ile mümkün olmaktadır.

Resim 2: Mobil Platformlar Üzerinde SOA

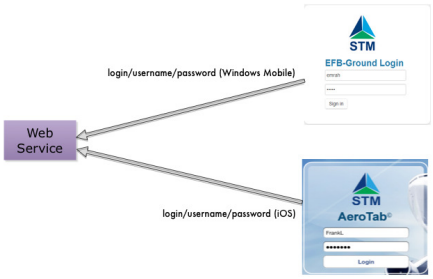
Vaka Çalışması; Elektronik Uçuş Çantası:

Firmamız olan STM AŞ'nin geliştirmiş olduğu Elektronik Uçuş Çantası (Electronic Flight Bag -EFB) projesi, farklı mobil platformlar üzerinde çalışan, uçuş yapan ekibin, uçuş yönetimini daha kolay ve daha verimli olarak kağıtsız bir ortamda gerçekleştirebildikleri mobil yazılım sistemlerini kapsamaktadır.

EFB projesinde birden çok mobil platform bulunmaktadır. Mevcut kullanılan EFB yazılım sistemi; İOS, Windows 8 Mobile ve web platformları üzerinde bulunmaktadır. Bu yazılım sistemlerinin tamamen birbirlerinden bağımsız ve esnek olması en önemli yazılım geliştirme kriterlerindendir. Ayrıca, mobil platformların kısıtlarının da bir nevi üstesinden gelebilmek, EFBprojesiningerekliliklerindendir.

Resim 3: Elektronik Uçuş Çantası

SOA altyapısı EFB projesinin birçok sıkıntısını ortadan kaldırmaktadır. Başlangıçta, farklı platformlar üzerinde bulunan EFB'nin yazılım sistemlerinin bağımsız ve esnek olabilmesi için, bağımsız bir servis altyapısı ile birbirleri arasında entegrasyon sağlanmıştır. Örnek olarak, kimlik denetimi işlemi her bir platform için "login" servisiyle bağdaşlanmıştır. "login" servisi üzerinden gelen cevaplar farklı platformlarda farklı kullanıcı arayüzleri ile görüntülenebilmiştir.



Resim 4: Login Servisleri

Bir diğer örnekte; EFB üzerinde kullanılan doküman yönetim sistemi verilebilir. EFB üzerindeki doküman yönetim sisteminin ağaç yapısı içerisindeki dokümanlar ve kategorilerinher bir platform üzerindeki entegrasyonu "getCategoryList" ve "getDocumentList" gibi servisler aracılığıyla sağlanmaktadır.

Aynı şekilde, mobil platformlar üzerindeki kaynak kısıtından dolayı doküman arama veritabanının oluşturulması oldukça zaman almaktadır. Bu nedenle, bu işlemlerin mobil platformlarda gerçekleştirmek zor olduğundan SOA sayesinde bu işlemler EFB projesinde "processDocument" servisiyle sağlanmıştır.

Görüldüğü üzere, SOA altyapısı sayesinde EFB projesinde, mobil platform farklılıkları ve yazılım sistemlerinin esnekliği sağlanmıştır. Aynı zamanda mobil platformlarda oluşacak kaynak kısıtlarının da üstüne SOA yardımıyla çıkılmıştır.

Sonuç:

Artık günümüz yazılım sistemlerinin anlık değişimleri ve birden çok platformda geliştirilmiş olması nedeniyle, birbirleri arasındaki entegrasyon süreci ve bağımsız, yalın halde kalması önemli bir problemdir. SOA altyapısıyla günümüz yazılım sistemlerinin problemlerinin bir çoğuna cevap bulabilmek mümkündür. Firmamız tarafından geliştirilen EFB projesinde farklı mobil platformlar üzerinde geliştirmiş olduğumuz yazılım sistemimizin problemleri, SOA altyapısıyla çözüm bulmuştur. Farklı platformlardaki EFB yazılımının bağımsızlık ve esneklik sıkıntıları, yazılım parçalarının birbirleriyle entegrasyonu ve kaynak kısıtları SOA kullanımı ile çözüme kavuşturulmuştur.

Kaynaklar:

[1] Bell, Michael (2008). "Introduction to Service-Oriented Modeling". *Service-OrientedModeling: Service Analysis, Design, and Architecture.* VViley & Sons. p. 3.

[2] Bell_, Michael (2010). *SOA Modeling Patterns for Service-Oriented Discovery and Analysis.* VViley & Sons. p. 390.

[3] Nevvcomer, Eric; Lomovv, Greg (2005). *Understanding SOA with Web Services.* Addison VVesley.

[4] Eri, Thomas (2005). *Service-oriented Architecture: Concepts, Technology, and Design.* Upper Saddle River: Prentice Hail PTR

[5] Yuri Natchetoi (2008). Service-oriented architecture for mobile applications. SAM 08. p. 27-32. ACM

[6] Do Van Thanh (2005). A service-oriented architecture framevvork for mobile services. Telecommunications 2005. p. 65-70. IEEE

[7] Tergujeff, R. (2007). Mobile SOA: Service Orientation on Lightvveight Mobile Devices. VVeb Services 2007. p. 1224-1225. IEE