

A Real Time Price Sharing Architecture for Open Banking

Açık Bankacılık İçin Gerçek Zamanlı Bir Fiyat Paylaşım Mimarisi

Ebru Özdoğru Kandırmaz¹ Ufuk Tiryaki²

^{1,2} IBTECH A.Ş. Tübitak MAM Teknoloji Serbest Bölgesi, Gebze/Kocaeli

¹ebru.ozdogrukandirmaz@ibtech.com.tr

²ufuk.tiryaki@ibtech.com.tr

Abstract. Open banking is a very popular subject nowadays. It is based on sharing financial data through OpenAPI. It provides a wide range of banking service and aims earnings to customers, financial companies, and third party developers. Price management strategies of companies and providing real time prices according to market have deep impact on the earnings of the customers and financial companies. In this paper, alternative solution architectures are mentioned to achieve real time price feeding. As a result, advantages and disadvantages of the implemented hybrid solution architecture is discussed.

Keywords: Open banking, Data polling architecture, Data subscription architecture.

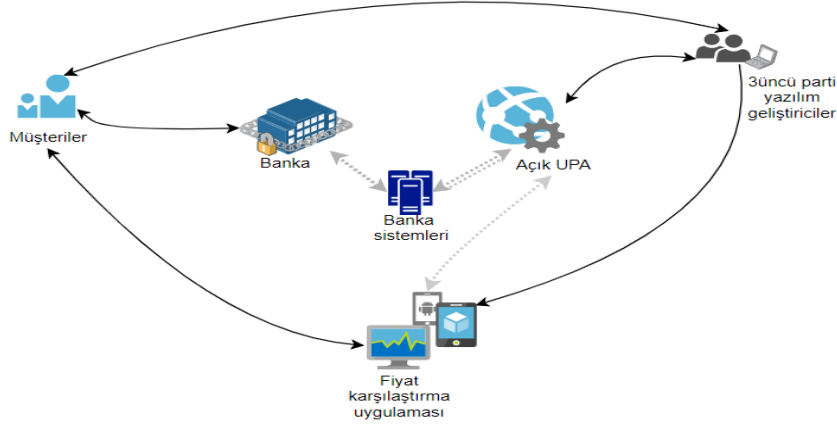
Özet. Açık bankacılık, finansal verilerin paylaşılması ilkesine dayanan son zamanlarda popüler olan bir alandır. Veriler Açık Uygulama Programlama Arayüzleri (UPA) kullanılarak paylaşılır. Geniş bir yelpazede bankacılık hizmetlerinin sunulmasını sağlar. Müşteriler, finansal kurumlar ve üçüncü parti yazılımcılar için kazanımlar hedefler. Kurumların fiyat yönetim stratejisi ve fiyatları piyasaya yakın zamanlı sunması, müşteri ve kurumların kazanımı için önemli etmenlerdir. Bu çalışmada, kurumun fiyatlarının gerçek zamanlı paylaşılması için mimari çözüm alternatiflerine yer verilmiştir. Sonuç olarak uygulanan karma çözümün avantaj ve dezavantajları tartışılmıştır.

Anahtar Sözcükler: Açık bankacılık, Veri Sorgulama Mimarisi, Veri Aboneliği mimarisi.

1 Giriş

Açık bankacılık finansal sistemlerde yeni ortaya çıkmış ve hızla gelişmekte olan bir alandır [1]. Odak konusu Açık Uygulama Programlama Arayüzleri (UPA) kullanarak veriyi paylaşmaktır [2].

Bankacılık verilerini farklı kurumlardan UPA aracılığı ile toplayan ve tek bir platform üzerinde sunabilen uygulamalar geliştirilebilir. Bu uygulamalar, finansal kurum içinden olan yazılımcılar tarafından olabileceği gibi, kurum dışından olan yazılımcılar tarafından da geliştirilebilir. Müşteriler tüm finansal bilgilerini tek bir uygulamada izleyerek kararlarını daha iyi verebilirler.



Şekil 1. Açık bankacılık akışı

Finansal sistemlerde veri paylaşımı riskli bir konu olduğu için yasal düzenlemelere tabidir. Avrupa’da ödeme sistemleri ile ilgili olarak İkinci Ödeme Sistemleri Yönetmeliği (PSD2) bu düzenlemeleri içerir. PSD2’nun çerçevesinde belirlenen açık bankacılık standartları, veri ve uygulama sağlayıcılar için basit, güvenli bir çözüm oluşturmayı amaçlar. Bu sayede geniş bir yelpazede açık bankacılık ürün ve hizmetlerinin oluşturulmasını hedefler [3]. Türkiye’deki sayısal bankacılık sisteminin Avrupa ülkeleri arasında ön sıralarda yer almasından yola çıkarak[4], ileriki yıllarda açık bankacılık alanında çalışmaların görülebileceğinden söz edilebiliriz.

İnternette standartlarla uyumlu bazı üst düzey referans mimariler ve açık UPA belirlenimleri sunulmuştur [5, 6, 7]. Referans mimariler güvenlik, yetkilendirme, iş modülleri gibi pek çok konuyu konumlandırır. Uygulamada her bir konu özelinde detaylı çalışılması gerekir.

Bankalar için kazancı arttırmak, hem müşteri davranışlarındaki hem de piyasadaki hızlı değişikliklere ve yasal düzenlemelere rağmen temel hedef olarak devam eder. Bu hedefe ulaşmanın yollarından önemli bir tanesi, fiyat yönetimidir. Bankaların müşterilerine iyi fiyat sunarak işlem yaptırması ve bu fiyattan kar elde etmesi gerekir. Açık bankacılık UPA kullanarak geliştirilmiş finansal fiyatları karşılaştıran bir uygulama,

müşterilere gördüğü en iyi fiyattan işlem yapabilme olanağı sağlar. Bankanın fiyat yönetim stratejisi ve fiyatlarını etkin sunabileceği bir mimari tasarım ile piyasada bir fırsat yakalanabilir.

Bu çalışmada, Açık Bankacılık UPA'ya uyumlu olmayı hedefleyen ve fiyat verilerini üçüncü partilerle gerçek zamanlı paylaşmak isteyen bir kurum için oluşturulmuş örnek bir çözüm mimarisi konu alınmıştır. Alternatif mimarilerin iyi ve kötü yanları sunularak bir değerlendirme yapılmıştır.

2 Problemin Tanımlanması

Bir bankacılık sisteminde, pek çok değişik türde fiyattan söz edilebilir. Mevduat faiz oranları, endeks oranları gibi fiyatlar günlük, gün içinde bir kaç kez veya ayda bir kaç kez değişen fiyatlar olarak düşünülebilir. Diğer yandan, döviz kurları çok sık değişen fiyatlardır. Çok sık değişen fiyatlara gerçek zamanlı erişilmesi, hem banka hem de müşteriler için piyasaya en yakın fiyat ile işlem yapılabilmesi açısından önemlidir.

Bu amaçla döviz kuru fiyatlarının bankanın fiyatlama stratejisi doğrultusunda gerçek zamanlı olarak sistemde düşük seviyede yük yaratacak şekilde paylaşılması için yeni bir mimari tasarımı oluşturulması hedeflenmektedir. Oluşturulan sistemin dolandırıcılık girişimlerine karşı güvenlik unsurları içermesi gerekmektedir. Kurulan sistem bankanın açık bankacılık faaliyetleri için de temel alınacaktır.

Bankanın fiyatlama stratejisi göreceli olarak karmaşık hesaplamalara sahiptir. Fiyatlama strateji yönetim yazılımı kullanıcılar tarafından değişiklik yapılmasına olanak sağlayacak kadar esnek tasarlanmış bir yazılımdır. Bu esneklik gerçek zamanlı fiyatların sağlanması açısından başarımlı/yük problemleri yaratma potansiyeline sahiptir.

Döviz kurlarında alış fiyatı ile satış fiyatı arasındaki fark, spread oranı olarak adlandırılır ve bu orandan kar sağlanır. Fiyatlama stratejisi gereği, spread oranları müşteri grubu, istek yapılan kanal gibi pek çok sınıflandırmaya göre hesaplanır. Piyasadan gelen anlık değişen çıplak fiyatlara hesaplanmış spread oranları eklenerek müşteriye son fiyat sunulur. Sınıflandırma ölçüt sayısı esnek olarak değiştirilebilmektedir. Her bir sınıflandırma ölçütünün içerdiği çeşit sayısı da esnek olarak fiyatlama stratejisini oluşturan birimler tarafından tanımlanabilmektedir.

Çıplak döviz fiyatları, serbest piyasa kurlarının yayımlandığı üçüncü partilerden alınır. Bu fiyatların sisteme gerçek zamanlı alınması gerekmektedir.

3 Alternatif Mimariler

Mimari çözümü oluştururken verinin değişkenlik karakter özelliği önemli bir ölçüttür. İki temel mimari çözümden söz edilebilir:

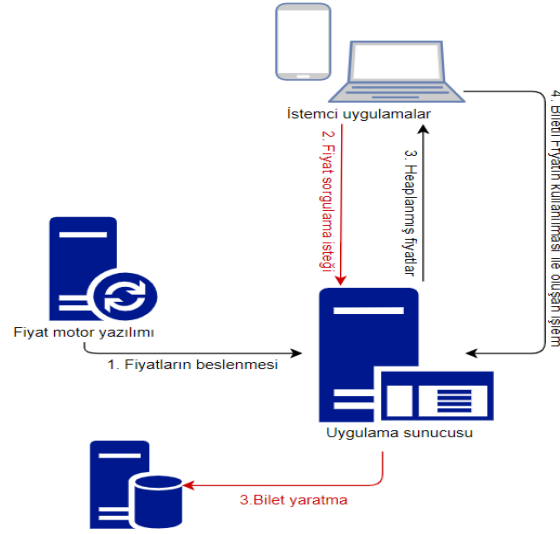
1. Veri Sorgulama: Fiyat verisinin istemciler tarafından talep edilmesi
2. Veri Aboneliği: Fiyat verisinin istemcilere bildirilmesi

3.1 Veri Sorgulama

Veri sorgulama, istemciden sunucu katmanına servis çağrısı ile yapılır. Değişim sıklığı az olan veriler için uygun bir çözümdür.

Bu mimaride, çok sık değişen veriler istemciler tarafından belli zaman aralıkları ile istek gönderilmesi ile alınabilir. Zaman aralıkları hizmet alanların kendi belirlediği periyotlardır. Çözümün şu dezavantajlarından bahsedilebilir:

1. Bu periyotlar gerçekte verinin değişim hızını karşılamayabilir. Sunucu katmanında veri daha sık ya da daha geç değişiyor olabilir. Veri daha sık değişirse gerçek değişimler istemciler tarafından yakalanamaz. Daha az değişirse, gereksiz istekler sunucu katmanına gönderilmiş olur.
2. Birçok istemcinin sunucu katmanına eş zamanlı ve belli zaman aralıkları ile sürekli istek göndermesi sonucunda oluşan yükten dolayı sistem kaynakları hiç bir isteğe cevap veremeyebilir. Açık UPA belirtiminde [7], çok sık veri talep eden istemciler için belirlenen maksimum bir sayıya ulaşıldığında hata verilmesi güvenlik amaçlı olarak önerilmiştir. Sunucu tarafında getirilen limitler, müşteri memnuniyetsizliğine neden olabilir.



Şekil. 2. Veri Sorgulama Mimarisi

Zaman Karmaşıklık Analizi. Çözümün zaman karmaşıklığını şu şekilde düşünebiliriz:

N: İstemci sayısı

M: İstemcinin birim zamanda sunucuya gönderdiği ortalama istek sayısı.

Zaman karmaşıklığı, $O(N*M)$ olarak hesaplanır. Sunucu katmanına gelen birim zamandaki istek sayısı arttıkça başarımlar oranı kötüleşir. İstek sayısı, sunucu katmanında

belirlenen maksimum bir sayı ile kısıtlanabilir [7]. M parametresini sabit bir sayı olarak düşünerek karmaşıklığı $O(N)$ 'e indirgeyebiliriz.

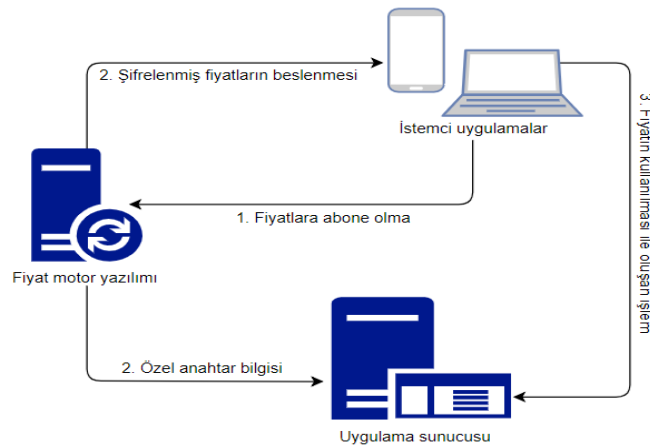
Problemin Mimari Kapsamında Değerlendirmesi. Mimaride fiyat hesaplaması istek temelli yapılıdır. Sunucu tarafında gelen isteğe göre çıplak döviz kuru fiyatının üzerine spread fiyatı eklenerek istemciye son fiyat gönderilir. Spread için, sunucu gelen istekteki müşteriyi içeren sınıflandırma ölçütlerinin değerlerini veri tabanından alarak hesaplama yapar. Belli bir müşteri için bu değerlere ulaşarak son veriyi hesaplama sabit zamanlı bir başarımlı ölçümü demektir. Spread hesaplamasının tek istek üzerinde yarattığı yüksek miktarda yükten bahsedilmez.

Güvenlik Değerlendirmesi. Sunucu fiyatı istemciye gönderirken, oluşturduğu bir bilet numarası ile adresleyerek gönderir. Bilet numarası tüm sistemde tekil belirlenmiş bir numaradır. İstemci işlem amaçlı kendindeki fiyat ile birlikte bilet numarasını sunucu katmanına geri gönderir. İstemciden gelen fiyat ve bilet numarası, sunucu katmanında veri tabanında tutulan verilerle karşılaştırılarak verinin doğruluğu ve güncelliği kontrol edilir. Mimari çözüm fiyatların güvenilirliğini bu yaklaşımla sağlamaktadır.

3.2 Veri Aboneliği

İstemciler, kendilerine bildirilmesini istedikleri veriye açılan ara yüzler (UPA) ile abone olurlar. Fiyatlar değiştikçe, sunucu katmanında çalışan fiyat motor yazılımı abonelere fiyatları bildirir. Değişim sıklığı az olan veriler için kullanılabilir bir yöntemdir. Bununla birlikte, yalnızca değişim sıklığı az olan verilerin paylaşımı için kurulmasına gerek olmayan bir mimari olduğundan bahsedilebilir.

Verinin değişmesinin bir olay olarak belirlenmesi ile veriye abone olan sistemlere değişen veri bildirilir. Bu yapısı ile çözüm Olay Yönelimli Mimari (OYM) ile temellenmektedir.



Şekil 3. Veri Aboneliği Mimarisi

Bu mimari yaklaşımı, çok sık değişen veriler için veri sorgulama çözümünün dezavantajlarını ortadan kaldırır.

Zaman Karmaşıklık Analizi. N: Fiyatlara abone olan istemci sayısı olduğu düşünülerek, gerçek zamanlı çıplak fiyat paylaşımı için zaman karmaşıklığı $O(N)$ olarak belirlenir. Değişen fiyatlar abone olan her istemciye gönderilecektir.

Problemimizi bu mimari kapsamında düşünecek olursak, sunucu katmanında çıplak fiyatlar değiştikçe spread fiyatları da hesaplanıp eklenerek son fiyat bilgisi abone olan istemcilere gönderilir. Olası tüm spread fiyatlarının hesaplama algoritması aşağıdaki şekilde örneklenebilir:

Her bir sınıflandırma ölçütünün, karar ağacımızda bir düğüm olduğunu düşünelim. Örnek olarak, üç düğümden oluşan bir karar ağacımız vardır:

Düğüm 1: Müşteri bölüm sayısı (SN)

Düğüm 2: Hizmet kanal sayısı (CN)

Düğüm 3: Kar merkezi sayısı (PN)

Olası sonuçların sayısı düğümlerdeki sayıların çarpımı kadardır.

$$RN = SN * CN * PN \quad (1)$$

Spread hesaplamasının zaman karmaşıklığı $O(P^3)$ olarak belirlenir.

Hizmet kanalları, temel bankacılık sistemi, mobil bankacılık, internet bankacılığı, kurumsal müşteri gibi kanallar olarak düşünülebilir. Bunların belli bir sayıyı geçmeyeceğini düşünebiliriz. Bu durumda CN sabit bir sayı olarak karmaşıklıkta ele alınarak spread hesaplaması zaman karmaşıklığı $O(P^2)$ 'ye indirgenebilir.

Sınıflandırma ölçüt sayısı arttıkça, olası spread fiyatlarını hesaplayan algoritma daha da yavaş çalışacaktır.

Sunucu katmanında çıplak fiyatlara spread fiyatlarının eklenerek istemcilere gönderilmesi için toplam zaman karmaşıklığı şu şekilde düşünülebilir:

$$O(N) + O(P^2) = O(\max(N, P^2)) \quad (2)$$

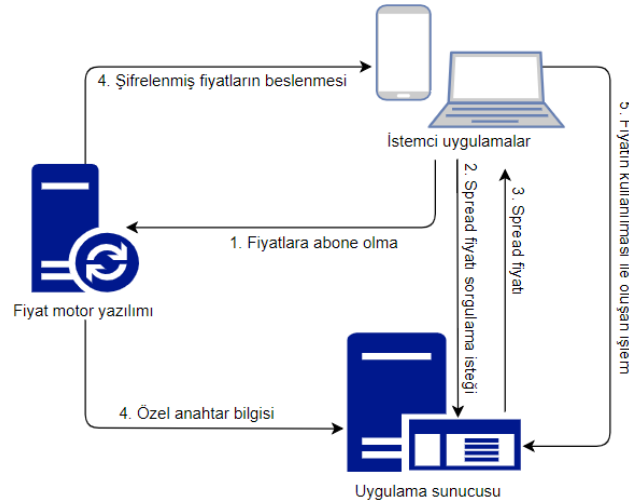
Güvenlik Değerlendirmesi. İstemcilere fiyatlarla birlikte, AES-128 bit ile şifrelenmiş veri gönderilir. Şifrelenmiş veride, fiyat bilgisi, fiyatın piyasalardan sisteme geliş zamanını gösteren zaman damgası, fiyat için tekil olan gelişigüzel üretilmiş şifre anahtarı (SALT) ve sistemde sık periyotlarla değiştirilen gizli şifre anahtarı bilgileri bulunmaktadır. İstemciden işlem sırasında gelen şifrelenmiş veri sunucu katmanda çözümlenerek doğrulama işlemleri yapılır. Bu yöntem ile fiyatların yanlış gönderilmesi ve dolandırıcılık faaliyetleri sorunu çözülmeye çalışılır.

Ek Kazanım Alanları. Bu çözümde abone olan istemcilere fiyat besleme sıklıklarının istemciye ve fiyatın türüne göre belirlenerek algoritmanın yapılandırılması mümkündür. Kazandırılan esneklik ile açık bankacılıkta farklı veri lisanslama ve ücretlendirme politikaları geliştirilebilir.

4 Çözüm Mimarisi

Problemimiz kapsamında, veri aboneliği mimarisi fiyatları değiştikçe besleyebilme özelliği ile başlangıçta uygun görünse de, spread hesaplamalarının sunucu katmanında yapılması gerçek zamanlı fiyat beslenmesinden uzaklaştıran bir çözüm olabilmektedir. Az değişkenlik gösteren spread hesaplamalarının, veri sorgulama mimarisinde müşteri temelli hesaplanması, tüm olası spread fiyatlarının sunucu katmanında hesaplanmasına göre daha etkin bir başarımlı/yük oranı sunmaktadır.

Bu bakış açısı ile, çok sık değişen çıplak fiyatların istemcilere sunulmasında veri aboneliği ve az değişen spread fiyatları için de veri sorgulama mimarilerini bir arada kullanmayı hedefleyen karma bir mimari çözüm uygulanmıştır.



Şekil 4. Karma Çözüm Mimarisi

İstemciler müşteri oturum açığında, müşteri için uygulanacak spread fiyatını sorgularlar. Veri sorgulama alternatif mimarisinde bahsedildiği üzere, spread fiyatlarının istek temelinde müşteriye göre hesaplanması az maliyetli bir işlemdir.

Abone olunan çıplak fiyatlar fiyatlama motor yazılımından istemcilere gönderildikçe, istemciler çıplak fiyatlara spread fiyatını ekleyerek son fiyatı müşterilere sunabilir.

Zaman Karmaşıklık Analizi. Bu çözüm mimarisinde, her istemci kendi spread fiyatını başlangıçta sunucudan sorgulamaktadır.

N: İstemci sayısı (Fiyatlara abone olan ve spread fiyatı sorgulayan)

M: İstemcinin birim zamanda sunucuya gönderdiği ortalama istek sayısı.

Spread fiyatlarının karmaşıklık analizi için, veri sorgulama mimarisinin zaman karmaşıklık analizinden yola çıkılabilir. Tüm istemcilerin aynı anda spread fiyatlarını sorguladığı ve M parametresini sabit bir sayı olarak düşünerek karmaşıklık $O(N)$ olarak hesaplanabilir.

Sık değişen fiyatların bildirilmesi için, veri aboneliği mimarisinde bahsedilen çıplak fiyatların N sayıda abone olan sisteme gönderileceği düşünülerek karmaşıklık $O(N)$ olarak belirlenir.

Toplamda sistemdeki karmaşıklık aşağıdaki şekilde özetlenebilir:

$$O(N) + O(N) = O(N) \quad (3)$$

İstemcilerde ise, iki fiyatın toplanması ile son fiyata ulaşıldığından sabit zamanda bir karmaşıklıktan bahsedilebilir.

Güvenlik Değerlendirmesi. Son fiyatın istemci tarafında hesaplanması ile uygulama hatası ya da dolandırıcılık kaynaklı fiyat kontrolü yapılması daha da önemli hale gelmiştir. Bu sorunun üstesinden gelmek için, veri aboneliği mimarisinde bahsedilen şifreleme algoritmaları kullanılmıştır. Şifrelenmiş veride ek olarak, o anda sistemde bilinen spread fiyatlarının versiyonları da bulunur.

Spread Güncellenme Sorununun Çözümü. Bir müşteri için açık olan oturum sırasında spread fiyatlarının güncellenmesi durumu bu çözüm mimarisinde düşünülmesi gereken bir diğer konudur. Spread fiyatları sunucu katmanında güncellendiği sırada, istemciden sunucuya işlem isteği eski spread fiyatı ile gönderilirse sunucu katmanındaki şifreleme yazılımındaki fiyat kontrolleri hatalı fiyat olduğunu anlayarak istemciyi bilgilendirebilir. Diğer yandan bu yaklaşım, müşteriler için güzel bir deneyim oluşturmayabilir.

Alternatif çözüm olarak spread fiyatları sürüm oluşturma çözümü uygulanmıştır. Sunucu katmanında spread fiyatları değiştiğinde yeni bir sürüm numarası oluşturulur. Sunucu gerçek zamanlı çıplak fiyatlarla birlikte o andaki geçerli spread fiyatının sürüm numarasını gönderir. İstemci daha önce sorgulamış olduğu spread fiyatının sürüm numarası ile gerçek zamanlı fiyatlarla birlikte gönderilen sürüm numarasını karşılaştırır. Eğer fark varsa, sunucuya yeni spread fiyat isteği gönderir.

5 Çözüm Mimarisinin Değerlendirmesi

Bankanın döviz fiyatları paylaşımı için üretim ortamında uzun süredir kullanılan veri sorgulama mimarisinde temellenen bir sistemi bulunmaktadır. Değişen piyasa koşullarını anlık takip edebilmek ve açık bankacılık uygulamalarını destekleyebilmek için gerçek zamanlı fiyat verme ihtiyacı oluşmuştur. Bu ihtiyacı karşılamak amacıyla mevcut

veri sorgulama mimarisinin kullanımı sırasında, aynı anda gelen çok sayıda isteğe sunucu katmanının cevap verebilme çabası, sistemde başarımlı/yük problemlerini gündeme getirmiştir. Öte yandan, açık bankacılık UPA referans belirtilerinde sistemde çok istek yapan uygulamalar için bir üst sınır tanımlaması öngörölmüş ve bu tür uygulamalar için olay bildirimini UPA yayımlanmıştır [7]. Bu nedenle, yeni bir mimari çözüm çalışması başlamıştır.

İhtiyacı karşılamak üzere oluşturulan karma mimari çözümün uygulandıđı sitem şu an test aşamasındadır. Testte en fazla bir saniye gecikme ile gerçek zamanlı veri paylaşımı yapılabildiđi görölmüştür. Sistemin Açık UPA ile dışarıya açılması çalışmaları, sistemi üretim ortamına taşıdıktan sonra başlayacaktır.

Benimsenen mimarinin bir çözümü olarak sunulan son fiyatın istemci katmanında gösterilmesi mantıđı, ince istemci niteliğinden bir miktar sapmaktadır. Çıplak döviz fiyatı ile spread fiyatının toplanması ve spread versiyonlarını karşılaştırıp gerektiğinde sorgulanması, istemci katmanı sorumluluğuna bırakılmıştır. Bu sorumluluklar büyük iş mantıkları içermemektedir. Kurum içinde geliştirilen istemci uygulamaları için çok büyük problemler oluşturmamaktadır. Bununla birlikte, Açık UPA ile geliştirilecek yazılımlar açısından bu sorumluluđu vermek biraz daha problemli olabilir. Geliştirilen yazılımlarda hata yapma olasılıđını arttırmaktadır. Bu tür hataların tespit edilebilmesi, mimaride tasarlanan sunucu katmanında uygulanan şifreleme kontrol yazılımları ile mümkündür. Yanlış gelen fiyat verileri kontrol edilerek, istemcilere hata durumu olarak bildirilecektir.

6 Sonuç

Açık bankacılık uygulamalarının geliştirilmesinde sistemler arasında veri paylaşımı önemli bir konudur. Bu çalışmada fiyat paylaşımları için mimari çözüm seçenekleri tartışılmıştır. Geliştirilen üçüncü parti yazılımlar, birçok bankadan fiyat alarak müşterilerin daha iyi fiyattan işlem yapabilmesine olanak sağlayabilecektir. Veriyi sağlayan sistemlerin yük/başarımlı oranları ile müşteri memnuniyetleri arasında doğrudan bir ilişki vardır. İyi bir fiyat stratejisi ve mimari çözümle, müşterilere piyasaya en yakın zamanlı fiyatları sağlayarak işlem yaptırabilmek rekabet ortamında öne çıkılmasını sağlayabilecektir.

Yapılan çalışmada, tarif edilen karmaşık ve esnek fiyat stratejisinin uygulanarak fiyatları gerçek zamanlı sunabilmek için veri sorgulama ve veri aboneliđi mimarilerini temel alan karma bir mimari çözüm oluşturulmuştur.

Çalışmanın sonucu olarak, veri paylaşımını sağlayan etkin bir mimari tasarımı ve verinin karakteristik özelliğinin yönlendirdiđi söylenebilir. Az deđişkenlik gösteren veriler için istemci-sunucu tabanlı veri sorgulama mimarileri gereksinimleri karşılayabilmekte iken, çok sık deđişen verilerin paylaşımında yetersiz kalmış ve yük/başarımlı problemlerine neden olmuştur. Verinin deđişiminin bir olay olarak belirlenmesi ve bu olayın abone olan sistemlere bildirilmesi ile gerçek zamanlı paylaşımı sağlanabilmiştir. Mimari çözüm seçeneklerinin deđerlendirilmesinde verinin karakteristik özelliđi bir ölçüt olarak belirlenmiştir.

Mimari çözüm seçeneklerinin sistemlere yaratacağı yük/başarım oranları diğer bir değerlendirme ölçütü olmuştur. Gerçek zamanlı veri paylaşım isterini mimari çözümlerin karşılama değerlendirmesi için zaman karmaşıklık analizlerinden faydalanılmıştır.

Dış sistemlerle veri paylaşımı ve paylaşılan verinin kullanımı söz konusu olduğunda, doğru verinin kullanılması, verinin doğru kullanılması ve verinin değiştirilmesi /dolandırıcılık faaliyetlerinin önlenmesi gibi konulardan bahsedilebilir. Mimari seçeneklerin ortaya çıkardığı güvenlik sorunları değerlendirilerek çözümler üretilmiştir. Etkin bir mimari bu sorunlar için çözümler barındırmalıdır.

Yukarıda bahsedilen ölçütler bu çalışmada her ne kadar açık bankacılık istemlerinde ele alınsa da, verinin paylaşıldığı diğer sistemler için de söz konusu olabilecek ölçütler olarak değerlendirilebilirler.

Kaynakça

1. Open banking homepage, <https://www.openbanking.org.uk/customers/what-is-open-banking>, son erişim 2018/09.
2. Apiacademy homepage, <https://www.apiacademy.co/lessons/2015/04/api-strategy-lesson-101-what-is-an-api>, son erişim 2018/09.
3. Open banking standards homepage, <https://www.openbanking.org.uk/providers/standards>, son erişim 2018/09.
4. <https://www2.deloitte.com/content/dam/Deloitte/global/Documents/About-Deloitte/central-europe/ce-digital-banking-maturity-study-emea.pdf?nc=1>, son erişim 2018/09.
5. Github homepage: <https://github.com/OpenBankProject/OBP-API/wiki/Open-Bank-Project-Architecture>, son erişim 2018/09.
6. IBM homepage: https://developer.ibm.com/apiconnect/2018/08/21/psd2_architecture/, son erişim 2018/09.
7. Open banking specification homepage: <https://openbanking.atlassian.net/wiki/spaces/DZ/pages/23363964/Read+Write+Data+API+Specifications>, son erişim 2018/09.