

Bir Endüstriyel Nesnelerin İnterneti Mimarisinin ATAM Yaklaşımı ile Değerlendirilmesi

Evaluation of an IIoT System Using ATAM

Kemal Memiş¹, Ersin Seza², Erdem Kaymaz² and Hüseyin Kutluca²

¹ ASELSAN, M. A. Ersoy Mah., 296. C. No:16, 06370 Yenimahalle Ankara / Türkiye
kmemis@aselsan.com.tr

² ICTerra Bilgi ve İletişim Teknolojileri A.Ş., Galyum Blok Kat:2, No:3 ODTÜ-Teknokent
06531
Ankara / Türkiye
ersin.seza@icterra.com
erdem.kaymaz@icterra.com
huseyin.kutluca@icterra.com

Özet. Binlerce uç birimden toplanan verilerin kaydedilmesini, analiz edilmesini ve operatörlere etkin olarak sunulmasını sağlayan izleme ve kontrol yazılımının mimarisi ATAM yöntemi ile analiz edilmiştir. Senaryo tabanlı analizi ile teknik riskler, hassasiyet noktaları ve ödünler belirlenmiş ve proje geliştirme aşamasında bu veriler dikkate alınmıştır.

Abstract. This paper presents the ATAM based architectural analysis of a monitoring and control system that collects data from thousands of end devices for recording, analysis and display. Risks, sensitivity points and tradeoffs identified during scenario based analysis are being handled during development phase of the project.

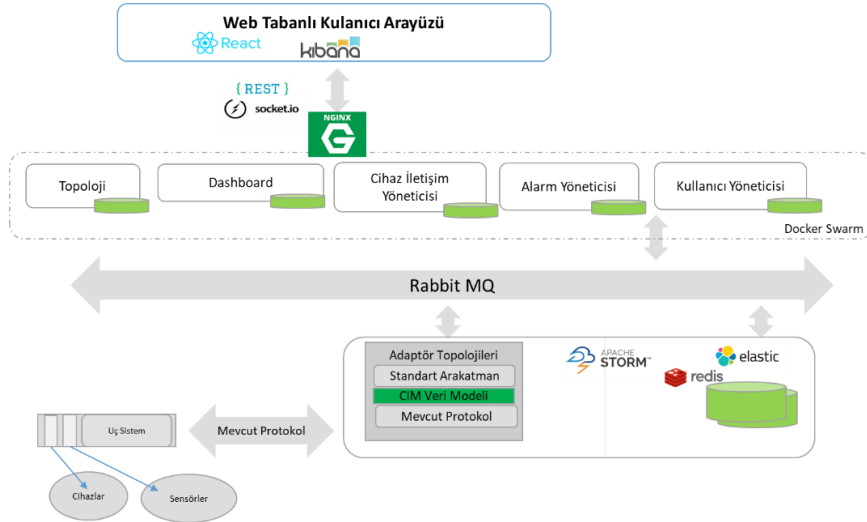
Anahtar Kelimeler: ATAM, Endüstriyel Nesnelerin İnterneti, Mikro Servis

Key Words: ATAM, Industrial Internet of Things, Micro Service

1 Giriş

ASELSAN Ulaşım, Güvenlik, Enerji ve Otomasyon Sistemleri Sektör Başkanlığı (UGES) her tür şebeke yönetimini (enerji, doğal gaz, su vb.) hedeflediği bir altyapı projesini gerçekleştirmektedir. Geliştirilecek mimarinin uygulandığı ilk alan olarak elektrik şebeke yönetimi (Enerji Dağıtım, Üretim, İletim Sistemleri) hedeflenmektedir. Bu kapsamda tasarım, yazılım geliştirme ve testlerin icra edilmesi işleri için yazılım mühendisliği alanında hizmet sağlayıcı olan ICTerra Bilgi ve İletişim Teknolojileri A.Ş ile projenin altyüklenicisi olarak çalışmaktadır.

Referans mimarisi Şekil-1 de verilen sistemde uç sistemler ASELSAN tarafından geliştirildiği gibi, mimari desteklediği uluslararası standartlar sayesinde farklı firmaların uç birimleriyle de veri alış verişi yapabilmektedir. Merkezde çalışacak olan yazılım, sahadan veri alınacak uç birim sayısından ve bu birimlerden alınan verilerin yoğunluğundan bağımsız, kesinti yaşanmadan erişilebilir olmalıdır. Bu isterin sağlanabilmesi için, sistemin gerçek zamanda dinamik olarak, yatayda aşağı ya da yukarı yönde ölçeklendirilebilmesi ve servis hatalarının birbirinden izole tutulabilmesi gerekmektedir. Bunun yanında sistemde bulunan servislerin işlediği verilerin servisler bazında homojen olmaması, her servisin bir diğerinden bağımsız olarak ölçeklendirilebilmesini gerektirmektedir. Mikro servislerin birbirinden bağımsız şekilde konteyner altyapıları kullanılarak kurulumları ve yedekli çalışmaları beklenmektedir. Tüm bu isterlerin yerine getirilebilmesi için, mikro servis[1,2] mimari tasarım kalıbı kullanılmıştır. Uygulamanın genel amacı, uç birimlerden ilgili protokoller üzerinden alınan verilerin, ortak veri modeline dönüştürülerek kaydedilmesi, analiz edilmesi ve ilgili kullanıcılara sunulmasıdır. Uç birimlerin entegrasyonu ile birlikte tüm sistem milli olarak geliştirilen Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things) [3] uygulaması olacak ve büyük çaplı görev kritik projeler için kullanılabilir bir altyapı oluşturacaktır.



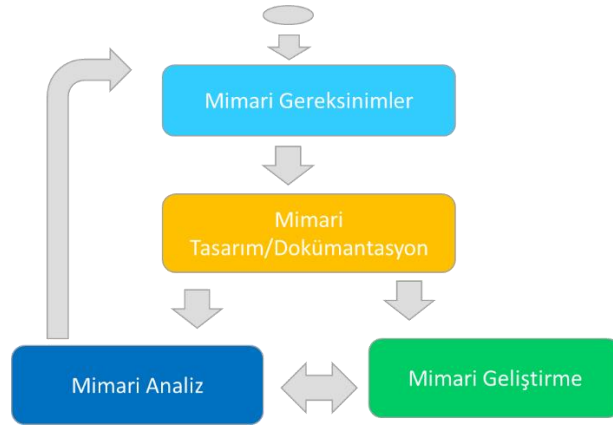
Şekil 1 Üst Seviye Mimarisi

Bölüm 2’de mimari tasarım süreci ve bu sürecin bir parçası olan ATAM tabanlı analiz yaklaşımı anlatılmaktadır. Analiz süreci girdisi olan iş hedefleri Bölüm 3’te verilmiştir. Bölüm 4’te mevcut sistem mimarisinin ATAM sürecine göre analizi anlatılmaktadır. Sonuç bölümünde yapılan çalışma değerlendirilmiş ve gelecek dönem çalışmaları özetlenmiştir.

2 Mimari Tasarım ve Analiz Yaklaşımı

Yazılım mimarisi geliştirme süreci Şekil 2’de verildiği gibi döngülü bir süreçtir. Mimari gereksinimler doğrultusunda tasarlanan yazılım mimarisinin mimari gereksinimleri karşıladığının ve projenin geliştirilebilir olduğunun eş gözden geçirme süreci ile değerlendirilmesi gerekmektedir. Bu kapsamda öne çıkan gözden geçirme yaklaşımı senaryo tabanlı bir mimari analiz metodu olan Mimari Ödün Analizi Metodudur (Architectural Tradeoff Analysis Method-ATAM) [4,5,6]. ATAM metodu endüstride gerek savunma projelerinde gerekse IT projelerinde başarı ile kullanılmıştır [7,8,9,10]. ATAM yaklaşımı savunma sanayinde kullanılan detaylı tasarım gözden geçirme aşamalarında kullanılabilirdiği gibi [7] gibi CMMI sürecinin bir parçası olan mimari Karar Analizi ve Çözümleme süreci kapsamında da kullanılabilir [8].

Çevik geliştirme yönetimi [11, 12] ile geliştirilen ve veri yoğunluğu bakımından kritik olan şebeke yönetim yazılımının mimari kararlarının değerlendirilmesi amacı ile tercih edilmiştir. Bu kapsamda ASELSAN proje teknik lideri, ICTerra proje teknik lideri ve ICTerra Kıdemli Yazılım Mimarı’nın yanı sıra ICTerra Uzman Yazılım Mimarı, ICTerra Grup Yöneticisi ve ASELSAN UGES Yazılım Mühendisliği Müdürü seviyesinde katılım sağlanarak projenin analizi yapılmıştır.



Şekil 2 Yazılım Mimari Tasarım Süreci ([7]’ den adapte edilmiştir.)

ATAM incelenen mimarinin potansiyel risklerini, ödünleşim ve duyarlılık noktalarını tespit eden bir yazılım mimarisi değerlendirme ve analiz etme yöntemidir. ATAM, alınan tasarım kararlarının sonuçlarını sistemden beklenen kalite özellikleri [13] doğrultusunda değerlendirmeyi amaçlar.

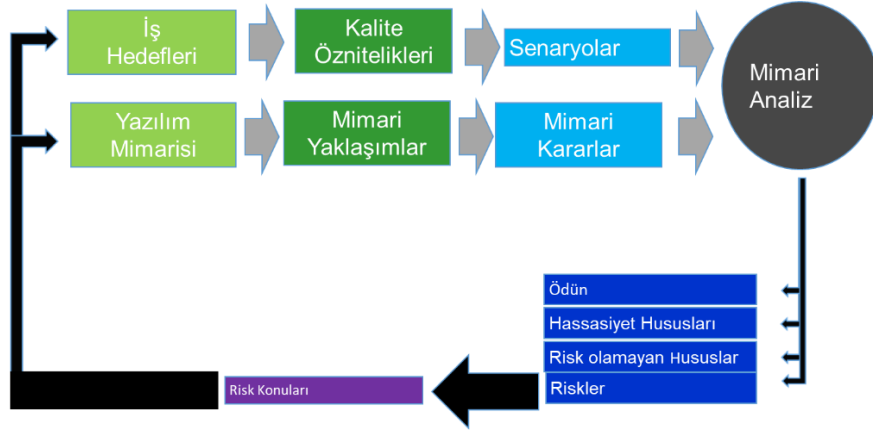
ATAM çalışması sonucunda, değerlendirilen sistemin;

- Kalite öznelikleri, gereksinimleri açıkça tanımlanmış,
- Yazılım mimari dokümantasyonu oluşturulmuş,
- Tasarım kararları gerekçelendirilmiş,
- Mimari riskleri tespit edilmiş,
- Paydaşları arasında iletişim geliştirilmiş olunması hedeflenmektedir.

Kavramsal ATAM süreci akışı Şekil 3'te verilmiştir. Bu süreç öncelikli olarak iş hedeflerinin belirlenmesi ile başlar. Bu aşamada sistemin işin kapsamı, fonksiyonel beklentiler, varsa teknik veya yönetsel kısıtlar ve mimariyi şekillendiren temel kalite gereksinimleri belirlenir. Bununla eş zamanlı olarak planlanan mimarinin ve sistemden beklenen kalite niteliklerinin öngörülen mimari tasarım ile nasıl gerçekleştirileceği belirlenir.

Daha sonraki aşamada sistem için önemli olan kalite nitelikleri belirlenir, bu nitelikler senaryolaştırılır ve senaryolar kendi içlerinde “önem” ve “zorluk” kıstaslarına göre önceliklendirilir.

Sonraki aşamada geliştirilen senaryolardan yüksek öncelikli olanlar analiz edilir. Bu adımda mimari riskler, duyarlılık noktaları ve ödün noktaları tespit edilir. Bu süreç sonucu yapılan çalışma raporlanır ve ihtiyaca göre projenin ileriki aşamalarında güncellenir. Tespit edilen risk konularına göre mimari geliştirme ve değerlendirme süreci tekrarlanır.



Şekil 3 ATAM Tabanlı Mimari Analiz Kavramsal Akışı (SEI web sitesinden adapte edilmiştir.)

Kavramsal olarak yukarıda verilen ATAM sürecinin Tablo 1'de belirtilen adımlar halinde ilerlemesi beklenmektedir [4]:

Tablo 1. Atam Süreci

Sunum Aşaması:
1. ATAM'ın Sunulması
2. İş Hedefi Girdilerinin Sunulması
3. Mimarinin Sunulması
Araştırma ve Analiz Aşaması:
4. Mimari Yaklaşımların Tanımlanması.
5. Kalite Faktörleri Ağacının Oluşturulması.
6. Mimari Yaklaşımların Analiz Edilmesi
Test Aşaması
7. Senaryo Önceliklendirilmesi
8. Mimari Yaklaşımların Analiz Edilmesi
Raporlama aşaması:
9. Sonuçların Sunulması

Bu sürecin geniş katılımcı paydaşlarla toplantılar halinde yapılması öngörülmekle birlikte [4] bu toplantıların çevik süreç uygulayan projelerde uygulanmasının getirdiği zorluklar değerlendirilerek süreç adapte edilmiş ve ön çalışmalar sonrası yarım günlük bir toplantı ile paydaş analizi yapılmıştır. [6]. Proje kapsamında bu yaklaşım kullanılmış, çekirdek kadro tarafından hazırlanan ATAM analizi yarım günlük bir toplantı ile değerlendirilmiş ve toplantıda alınan görüşlere göre analiz raporu güncellenmiştir. Süre kısıtlarından dolayı sınırlı sayıda senaryo dikkate alınmış ve buna göre çalışma yapılmıştır. Yine çevik model ile uyumlu ATAM sürecinde anlatıldığı üzere daha sonraki proje aşamalarında farklı senaryolar (örneğin güvenlik) ele alınarak ilgili senaryolar ile mimarinin analiz edilmesi hedeflenmektedir.

3 İş Hedefleri Girdileri

3.1 Paydaşlar ve Mimariye Etkileri:

ASELSAN UGES Sektör Başkanlığı: Ürün ve mimari tasarım sahibi. Ürünü, birden fazla projede kullanılabilir bir altyapı olarak hedeflemektedir. Sistem gereksinimlerini belirlemekte, mimariyi oluşturmakta ve yönlendirmektedir. Yazılım geliştirme faaliyetlerini teknik olarak yönlendirmekte ve belirli alanlarda yazılım geliştirme faaliyetlerine katılmaktadır.

ICTerra: Yazılım mimari tasarım faaliyetlerini ASELSAN ile birlikte yürütmekte ve yazılım geliştirme faaliyetlerini yürütmektedir.

Son Müşteri: Şebeke yönetim operatörleridir.

3.2 İşin Kapsamı:

Proje Kapsamında mimari tasarıma girdi olarak sunulan iş hedefleri aşağıda verilmiştir:

1. Referans mimari her tür şebeke yönetimini (enerji, doğal gaz, su vb.) hedeflemektedir.
2. Mimarının ilk uygulandığı alan olarak elektrik şebeke yönetimi (Enerji Dağıtım, Üretim, İletim Sistemleri) hedeflenmektedir.
3. Yeni teknolojinin getirdiği esneklik, birlikte çalışabilirlik ve kolay kurulum özellikleri ürünün pazarlama bilirliliğini arttıracaktır.
4. Ürüne orta vadede gelişmiş yüksek hacim analitiği ve makine öğrenmesi gibi en ileri kabiliyetlerin de eklenebilir olması öngörülmektedir.
5. Yüksek servis sürekliliği sağlayan bir ürün geliştirerek görev kritik sistemler için de kullanılabilir olması planlanmaktadır.

3.3 Mimariyi etkileyen Ana Kullanım Durumları

Proje Kapsamında mimari tasarımı etkileyecek kullanım durumları aşağıda verilmiştir:

1. KD-1 Uç cihazlardan gelen verinin gerçek zamanlı gösterilmesi
2. KD-2 Uç cihazlardan gelen verinin veri tabanına kaydedilmesi
3. KD-3 Kayıtlı veriden grafiklerin oluşturulması
4. KD-4 Gelen verilerden alarm koşullarının oluşturulması ve sergilenmesi
5. KD-5 Uç birimlere ilgili protokolleri kullanarak komut veya konfigürasyon verisi gönderilmesi
6. KD-6 Toplanan veriler üzerinde veri analitiği algoritmaları işletilmesi

3.4 Mimari Kaygılar

Proje paydaşları tarafından dile getirilen mimari kaygılar değerlendirilmiş ve aşağıda listelenmiştir:

1. Sistemin hem bulut üzerinde hem de yerel sunucularda çalışması beklenmektedir.
2. Farklı büyüklükteki projeler için ölçeklenebilir olmalıdır. (Küçük projeler için maliyet etkin çözüm olması önemlidir.)
3. Sistemin performans sorunu olmadan 7/24 çalışması beklenmektedir.
4. Yedekli çalışabilecek alt yapıda tasarlanması beklenmektedir.
5. Dağıtık (cluster) çalışabilecek alt yapıda tasarlanacaktır. Dinamik olarak genişleyebilecektir/daralacaktır.
6. Dış ara yüzlerde yeni protokolleri destekleyebilecek esnek tasarım altyapısına sahip olacaktır.
7. İzlenebilirlik yüksek olacaktır. Servislerin hata/sağlık vb. durumlarının izlenebilmesi için kayıt altyapısı sağlayacaktır.
8. Farklı projelerde farklı dış ara yüzler ve yapılandırılabilir edilebilir yetenekler olabilecektir. Farklı projelerde yeni yazılım servislerinin eklenmesine olanak tanımlamalı. Bunun için diğer modüllerde değişiklik gerekmemelidir. Kullanılması istenmeyen servislerin sistemde kapatılabilmesi sağlanmalıdır.

3.5 Sınırlamalar

Paydaşların belirttiği ve yazılım mimarisini etkileyebileceği öngörülen proje sınırlamaları aşağıdaki gibidir:

1. Referans mimari ile ilk çalışabilir prototip yazılımın proje tasarım aşamasının başlangıcından itibaren bir sene içinde çıkması gerekmektedir.
2. Ürünün CIM [14] standart veri modelini desteklemesi beklenmektedir.
3. İşletim sisteminden bağımsız olarak çalışması beklenmektedir.
4. Farklı şirketlerde çalışan birden fazla ekip tarafından paralel geliştirilme yapılabilecektir.
5. Lisanslı yazılımlar yerine projede aynen kullanım kısıtı bulunmayan (GPL, LGPL, Apache gibi) açık kaynak yazılım kütüphaneleri tercih edilecektir.

4 Kalite Öznitelikleri Senaryoları ve Mimari Analiz

Mimari analiz çalışması olarak kalite öznitelikleri senaryoları oluşturulmuş ve üzerinde analiz çalışması yapılmıştır. Bu kapsamda ortaya çıkan senaryolar, proje açısından önem ve yazılım geliştirme açısından zorluk derecelerine göre değerlendirilmiştir (Tablo 2).

Tablo 2. Kalite Öznitelikleri

NO	Kalite Özniteliği	Senaryo	Önem Y: Yüksek O: Orta D: Düşük	Zorluk Y: Yüksek O: Orta D: Düşük
Perf-1	Performans-sunum zamanı	Uç birimlerden gelen verilerin kayıpsız işlenmesi.	Y	Y
Sür-1	Süreklilik	Sunucu yazılımları yedekli çalışabilmeli.	Y	Y
Değ-1	Değiştirilebilirlik	Sisteme farklı protokol üzerinden konuşan yeni uç sistem ve/veya farklı merkezler entegre edilebilmesi.	Y	O
Değ-2	Değiştirilebilirlik	Sisteme yeni yetenekler eklenebilecek ve/veya mevcut servisler kapatılabilecektir.	Y	O
Kış-1	Kullanışlılık	Minimum sayıda operatör ile izleme ve yönetim.	O	O
Bkm-1	Bakım yapılabilirlik	Servislerin hata/sağlık vb. durumlarının izlenebilmesi için kayıt altyapısı sağlanması.	O	D

Seçilen örnek senaryolar için detaylı kalite öznitelikleri analizleri aşağıdaki Tablo 3-Tablo 8’de verilmiştir. Her bir senaryoyu gerçeklemek üzere seçilen mimari kararlar dokümanite edilmiştir. Bununla birlikte mimari riskler, duyarlılık noktaları ve ödün noktaları tespit edilmiş ve sunulmuştur.

Tablo 3. Performans Senaryosu

Senaryo #: Perf-1	Trafolardan gelen verinin kayıpsız işlenmesi			
Ortam: Sistemin entegre sahada çalışma ortamı				
Uyarıcı (Stimulus): Örnek kullanım durumunda verilen sayıda istasyonun sisteme ön-görülen frekansta veri sağlaması				
Tepki (Response): Verilerin kayıpsız şekilde veri tabanına kaydı ve alarm koşullarının kayıpsız şekilde tespiti.				
Tepki Ölçüsü (Response Measure): Kayıpsız veri için referans yük durumu: Örnek bir kullanım durumu: 15000 İstasyon, her istasyonda 3500 ölçüm noktası, her ölçüm noktasından ortalama 64 byte veri (zaman, tipi, ölçüm değeri vb.), 10 dakikada bir güncelleme				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Trade-off)	Risk husus-ları	Risk olma-yan hu-suslar
Verinin Elasticsearch Veritabanına zaman serisi olarak kayıt edilmesi	H1,H2			
Cihaz bağlantılarının, veri tabanı kayıt mekanizmasının ve alarm üretim mekanizmalarının Apache Storm topolojileri olarak tanımlanması	H3			
Gereke: Elasticsearch yüksek hacimdeki verileri zaman etiketli olarak saklayarak etkin sorgulamaya müsade etmektedir. Apache Storm yatay genişleyebilirliğe olanak sağlamaktadır.				
Hassasi-yet/Ödün/Risk/Risk ol-mayan	Açıklama			
H1	Çok yüksek hacimdeki verilerin etkin şekilde saklanabil-mesi için kullanılacak disk sistemi de önemlidir			
H2	Benzer şekilde veri kayıt periyodu dikkate alınarak disk bo-yutu belirlenmeli.			

H3	Performans açısından en kritik olan ölçüm verisinin işlenmesi ve modüller arası aktarılması optimize şekilde yapılmalı.
----	---

Tablo 4. Süreklilik Senaryosu

Senaryo #: Değ-1	Sisteme farklı protokoller üzerinden konuşan yeni uç sistemlerin ve/veya farklı merkezlerin entegre edilmesi.			
Ortam: Yeni Proje/Operasyonel sistem				
Uyarıcı (Stimulus): - Mevcut Altyapının yeni proje için yeniden kullanılması - Mevcut sisteme yeni tür bir uç birim entegre edilmesi - Mevcut sistemin başka bir sistem ile entegre edilmesi				
Tepki (Response): Yeni uç birimin/karşı sistem ile veri alışverişi sağlaması				
Tepki Ölçüsü (Response Measure): Mevcut servislerin en az şekilde etkilenmesi.				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Trade-off)	Risk hususları	Risk olmayan husus
Sistem içinde ortak veri modeli kullanılması (CIM)	H1, H2		R1	RO-1
Kritik bütün verilerin mesaj kuyruğu üzerinden iletilmesi				
Gerekeç: Farklı protokollerden gelen verilerin ortak veri tabanında kaydedilmesi, alarm koşullarının çalıştırılması ve diğer analiz algoritmaları için ortak veri modeli kullanılması öngörülmüştür. Ortak veri modeli olarak CIM veri modeli birçok projede sözleşme maddesi olarak yazılmaktadır.				
Hassasiyet/Ödün/Risk/Risk olmayan	Açıklama			
R1	CIM veri modeli hala gelişmekte olan bir standarttır ve veri modelinde gelişmeler olmaktadır. Uygulanacak projede daha yeni bir CIM versiyonu istenmesi durumunda, yazılım modüllerinde geniş çaplı değişiklik gerekecektir.			
H1	Yeni entegre edilen trafoda daha önceden işlenmemiş veri /kontrol verisi olmaması durumunda birden fazla modül etkilebilecektir. Modüllerin yeni veriler/mesajlar eklenebilir yapıda geliştirilmesi gerekmektedir.			
H2	Yeni protokolleri desteklemek için farklı 3. Parti kütüphane kullanmak gerekebilecektir. Farklı kütüphanelerin entegrasyonu için uygun altyapı kurulması gerekmektedir.			

T1	Storm kullanıldığı için Apache Storm'un desteklediği dillere bağımlılık bulunmaktadır. Diğer dillere mecbur olduğu takdirde mesaj kuyruğuna veri aktaran adaptör tanımlanması gerekecektir.
RO-1	Yeni entegre edilen trafoda daha önce işlenmiş bir veri için bütün alanlar dolu olmayabilir. Bu durumda varsayılan değerler kullanılacaktır.

Tablo 5. Değiştirilebilirlik Senaryosu -1

Senaryo #: Sür-2		Sunucu yazılımları yedekli çalışabilmeli		
Ortam: Sahada çalışan istasyon				
Uyarıcı (Stimulus) : Donanım hatası, güç kesilmesi, yazılım hatası				
Tepki (Response) : Sistemin kapanma sonrası yeniden başlatılması				
Tepki Ölçüsü (Response Measure) : Mevcut görevin devam edebilmesi				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Trade-off)	Risk Hususları	Risk Olmayan Hususlar
Mesaj kuyruğu (Message Queue) yapısının kullanılması				
Gerekeç: Yazılımlarda problem olduğu zaman (istenmeden sonlanma, elektrik kesintisi vb.) mesaj kuyruğu veriyi diskte tutarak veri kaybını önleyecektir.				
Hassasiyet /Ödün/ Risk/Risk olmayan		Açıklama		

Tablo 6. Değiştirilebilirlik Senaryosu

Senaryo #: Değ-2		Yeni bir servisi mevcut servisleri değiştirmeden ekleme		
Ortam: Bakım ve garanti dönemi				
Uyarıcı (Stimulus): Yeni bir fonksiyon ihtiyacı				
Tepki (Response): Mevcut servisler değiştirilmeden yeni bir servisin eklenmesi				
Tepki Ölçüsü (Response Measure): Değiştirilen servis sayısı				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Trade-off)	Risk hususları	Risk olmayan husus

Mikro servis mimarisi				
Rabbit MQ Mesaj Kuyruğu	H1			
Konteyner yapısı				
Gerekeç: Yeni servisler mikro servis olarak tasarlanacaktır. Yeni servisler konteyner ile kurularak diğer servisleri olumsuz etkilememesi sağlanacaktır. Yeni servisler RabbitMQ üzerinden verilere erişebilecektir.				
Hassasiyet/Ödün/Risk/Risk olmayan	Açıklama			
H1	Kapatılması muhtemel servislere bağımlı başka servislerin olmamasına özen gösterilecektir.			

Tablo 7. Kullanışlılık Senaryosu

Senaryo #: Klş-1	Minimum sayıda (1-2) operatör ile izleme ve yönetim			
Ortam: Operasyonel sistem				
Uyarıcı (Stimulus): Binlerce uç cihazdan veri gelmekte, alarm koşulları oluşmakta				
Tepki (Response): Alarm durumlarının kaçırılmaması, zamanında operasyonel tepki verebilme				
Tepki Ölçüsü (Response Measure): Hata durumunun kolaylıkla tespiti. Gerekli verinin olması				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Tradeoff)	Risk hususları	Risk olmayan husus
Web Tabanlı Tek sayfa Uygulaması (React Node JS)				RO1
Filtre Tasarımları				
Hızlı ekran tasarımı				
Analiz/alarm altyapısı			R1	
Önceliklendirilmiş alarm altyapısı				
Gerekeç: İzleme sistemlerinde operasyonel maliyeti düşürmek üzere az sayıda operatör ile işlenmesi hedeflenmektedir. Bu durumda mevcut operatörlerin ekranlarının aşırı yüklenmemesi ve oluşan alarmların kayıpsız şekilde operatöre iletilmesi beklenmektedir.				
Hassasiyet/Ödün/Risk/Risk olmayan	Açıklama			
R1	Mevcut algoritmalar verilerin limit değerini incelemekte ve limit dışı her durum için alarm üretmektedir. Bu durumda bir uç			

	<p>sistemde oluşacak ve birden fazla veriyi etkileyecek durum için aynı anda onlarca alarm üretilebilir. Bu durum operatörün aşırı yüklenmesine sebep olabilir.</p> <p>Risk giderme yöntemi 1: Yüksek hacimli analitiği ve makine öğrenmesi gibi en ileri kabiliyetlerin de eklenebilir olması öngörülmektedir.</p>
RO-1	<p>Web tabanlı tek sayfa uygulamaları üzerinde sade ekranlar UX tasarımcıları tarafından tasarlanmakta ve operatöre verinin etkin bir şekilde sunulması öngörülmektedir.</p>

Tablo 8. Bakım Yapılabilirlik Senaryosu

Senaryo #: Bkm-1	Sistemin hata/sağlık vb. durumlarının izlenebilmesi			
Ortam: Operasyonel sistem				
Uyarıcı (Stimulus): Yazılım hatası/beklenmeyen veri/ durum				
Tepki (Response): Hata durumunun tespiti için kullanıcıya/geliştiriciye gerekli logların sağlanması				
Tepki Ölçüsü (Response Measure): Hata durumunun kolaylıkla tespiti. Gerekli verinin olması				
Mimari kararlar	Hassasiyet (Sensitivity)	Ödün (Tradeoff)	Risk hususları	Risk olmayan husus
Zipkin Kütüphanesi	H1			
ELK Altyapısı				
MetricBeat				
Gerekeç: Tüm konteynerlerin metrikleri MetricBeat ile loglanmakta ve gerçek zamanlı gösterilebilmektedir. ELK altyapısı ile tüm uygulama logları toplanmaktadır. Transaction ID altyapısı ile uçtan uca loglama kabiliyeti sağlanmıştır.				
Hassasiyet/Ödün/Risk/Risk olmayan	Açıklama			
H1	Belirli aralıklarla eski logların yedeklenip silinmesi öngörülmeli.			

5 Sonuç

Yapılan mimari analiz çalışması ile müşteriden gelen iş hedefleri, mimari kaygılar ve teknik kısıtlar dikkate alınarak oluşturulan mimarinin senaryo tabanlı analizi

yapılmıştır. Senaryo seçiminde mikro servis mimarisi dikkate alınarak en kritik görülen performans, değiştirilebilirlik ve süreklilik ön planda tutulmuştur. Bu yaklaşım [11]'de anlatılan endüstri verileri ile de tutarlıdır. Bu çalışmada ayrıca kullanılabilirlik ve bakım yapılabilirlik ön planda tutulmuş ve analiz edilmiştir. Proje kapsamında ortaya çıkartılan teknik riskler, hassasiyet noktaları ve ödümler dikkate alınarak proje prototip çalışması devam etmektedir. Prototip sonrası mevcut mimari analiz dokümanı prototip sonuçlarına göre güncellenecek ve proje yazılım geliştirme faaliyetleri bu dokümana uyumlu bir şekilde yapılacaktır. Önümüzdeki dönemde mevcut senaryoların daha detaylı analizi yapılacak ve güvenlik kalite özneliği senaryoları üzerinde çalışmalar yapılacaktır. Projenin sonraki aşamalarında depolanan verilerin derin öğrenme, büyük veri analizi, makine öğrenmesi gibi tekniklerle analiz edilerek karar destek sistemlerinin ortaya çıkması hedeflenmektedir.

Referanslar

1. Erl, T.: SOA Design Patterns. Pearson Education, Boston, MA, USA (2009).
2. Newman, S.: Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 1st edn. (2015).
3. <https://www.iiconsortium.org/vertical-markets/energy-utility.htm>, Last Accessed November 2018.
4. Kazman, R., Klein, M., Clements, P.: ATAM: Method for Architecture Evaluation, TECHNICAL REPORT CMU/SEI-2000-TR-004.
5. Cervantes H., Kazman R.: Designing Software Architectures A Practical Approach, Addison-Wesley,(2016).
6. Bass L., Clements, P., Kazman, R.: Software Architecture in Practice, 3rd ed., Addison-Wesley, (2012)
7. Byrnes, C., Kyratzoglou, I.: Applying Architecture Tradeoff Assessment Method (ATAM) As Part Of Formal Software Architecture Review, Technical Report 07-0094, The MITRE Corporation, (2007)
8. Kutluca, H. Emre Çetin, İ., Çakır, U., Kılıç, M.: GEMKOMSİS Savaş Yönetim Sistemi Yazılımının AR-GE Projesi Olarak Geliştirilmesi, Deniz Platformları için Sunduğu Ortak Altyapı ve Sahil Güvenlik Arama Kurtarma Gemisi Uygulaması, Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu, İstanbul (2008).
9. Uyanıksoy, G., Oğuztüzün, H., Yazıcı, A.: Bir Çoklu Ortam Veri Yönetim Sistemi Mimarisinin ATAM ile Değerlendirilmesi, Ulusal Yazılım Mühendisliği Sempozyu, İzmir (2013).
10. Toth, S.: An Inverse Evaluation of Netflix Architecture Using ATAM, SEI Architecture Technology User Network Conference, San Diego (2016).
11. Bellomo S., Gorton I., Kazman R.: Toward Agile Architecture: Insights from 15 Years of ATAM Data, IEEE Software , (September 2015).
12. Ambler S., Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise, IBM Press, (2012).
13. ISO IEC 25010:2011, <https://www.iso.org/standard/35733.html>, last visited November 2018.
14. DMTF Common Information Model (CIM). <http://dmtf.org/standards/cim>, Last Accessed November 2018.