

T.C.
İSTANBUL OKAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI



İSTANBUL
OKAN ÜNİVERSİTESİ

NESNELERİN İNTERNETİ VE UÇTAN UCA
BAĞLI EV (AKILLI) SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Levent YILDIZ

tarafından

YÜKSEK LİSANS

derecesi şartını sağlamak için hazırlanmıştır.

Mayıs 2019

Program: Bilgisayar Mühendisliği

NESNELERİN İNTERNETİ VE UÇTAN UCA BAĞLI EV (AKILLI)
SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

LEVENT YILDIZ

tarafından

İSTANBUL OKAN ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümüne

Yüksek Lisans

derecesi şartını sağlamak için sunulmuştur.

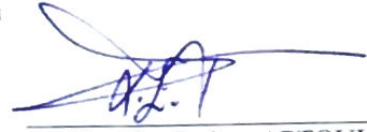
Onaylayan:



Doç.Dr. Pınar YILDIRIM
Danışman



Prof.Dr. Bekir Tevfik AKGÜN
Üye



Doç.Dr. Erchan APTOULA
(Gebze Teknik Üniversitesi)
Üye

Mayıs 2019

Program: Bilgisayar Mühendisliği

ÖZET

Bu tez çalışması ile bağı akıllı evlerin kullanımı ve hayatımıza etkisi araştırılmıştır. Çalışmada bağı akıllı evlerin kurulumu ve kullanımı konusunda geniş yer tutan bulut platformu ve bulut teknolojilerine de ayrıca yer verilmiştir.

Çalışma yapılırken öncelikle daha önceden yapılan akademik araştırmalar incelenmiştir. Daha sonra sektörde ve ülkemizdeki ticari bağı-akıllı ev ürünleri araştırılmıştır. Edinilen bilgiler neticesinde bir bağı akıllı evde olması gereken sensörler ve aktüatörler belirlenmiştir. Bahsi geçen sensörler temin edilerek üzerinde deneme ve test çalışmaları yapılmıştır. Bağı akıllı evde kullanılacak sensörler araştırılıp tek tek implementasyonu yapılmıştır.

Son olarak ise uçtan uca bir bağı akıllı ev tasarımı yapılarak çeşitli sensörler tek bir uygulamada toplanarak bulut platformuyla iletişimi sağlanmıştır. Bu sensörlerin kurulumunu ve kullanımı sağlayacak iOS ve Android işletim sisteminde çalışan yerel(NATIVEAPP) bir mobil uygulama geliştirilmiştir.

Anahtar Kelimeler: Nesnelerin İnterneti, Bağı Akıllı Ev, Ev, Ev Güvenliği

ABSTRACT

In this thesis, the use of connected smart homes and its effect on our lives were investigated. In the study, cloud platform and cloud technologies, which are widely used in the installation and use of connected smart homes, are also included.

When the study was carried out, firstly the researches in the academic community were examined. Then, commercial and smart home products in our country were investigated. As a result of the information obtained,

the connected smart home sensors and actuators were identified. The mentioned sensors were supplied and tested. The sensors used in the connected smart home are researched and implemented individually.

Finally, an end-to-end connected smart home was designed and various sensors were collected in a single application and communicated with the cloud platform. A native mobile application has been developed for iOS and Android operating systems to install and use these sensors.

Keywords: Internet of Things, Connected Smart Home, Home Security, Comfortable Home

İÇİNDEKİLER

ONAY SAYFASI.....	ii
ÖZET	iii
ABSTRACT	iv
İÇİNDEKİLER	v
ŞEKİL LİSTESİ	ix
KISALTMALAR	1
BİRİNCİ BÖLÜM	3
GİRİŞ	3
1.1. Araştırmanın Amacı.....	6
1.2. Araştırmanın Alana Katkıları.....	7
1.3. Tez Yazım Düzeni	8
İKİNCİ BÖLÜM.....	9
İLGİLİ LİTERATÜR.....	9
2.1. Bağlı/Akıllı Ev	9
2.2. Bağlı/Akıllı Ev Çalışma Alanları	13
2.2.1. Akıllı Ev Aletleri.....	13
2.1.2. Çoklu Ekranlar	15
2.1.3. Bulut Tabanlı Servisler.....	15
2.3. Nesnelerin İnterneti	16
2.4. Bulut Bilişim	21
2.4.1. Bulut Servisleri.....	23
2.4.2. Bulut Bilişim Uygulama Modelleri	28
2.4.3. Bulut Bilişimin Kullanımı	28
2.4.4. Bulut Bilişimde Riskler ve Dezavantajlar	29
ÜÇÜNCÜ BÖLÜM	31
NESNELERİN İNTERNETİ VE UÇTAN UCA BAĞLI EV MODELİ	31
3.1. Bağlı Ev Temel Görevleri	32
3.2. Bağlı Ev Kullanıcı ve Sistem Uygulamaları.....	35
3.2.1. Kullanıcı Uygulamaları	35
3.2.2. Sistem Uygulamaları	36
3.3. Bulut Tabanlı Otomasyon	39
3.4. Bağlı/Akıllı Ev Tasarımı Modeli	40
3.4.1. Akıllı Ev İçin IOT Tabanlı Bütünsel Çerçeve	40

3.4.1.1. Bağlı/Akıllı Ev	41
3.4.1.2. Bulut.....	42
3.4.1.3. Yarar.....	43
3.4.1.4. Üçüncü Taraf Uygulamaları	43
3.4.1.5. Kullanıcı Arayüzü	43
3.4.2. Bağlı/Akıllı Ev Yönetim Sistemi.....	43
3.4.2.1. Akıllı Nesnelere/Akıllı Cihazlar.....	44
3.4.2.2. Merkezler (Hubs)	45
3.4.2.3. Bulut.....	45
3.4.2.4. Üçüncü Taraf.....	46
3.4.3. Birlikte Çalışabilirlik.....	46
3.4.4. Güvenlik ve Gizlilik	47
3.5. Bağlı/Akıllı Ev Tasarımı Modeli İle İlgili Bir Senaryo.....	47
DÖRDÜNCÜ BÖLÜM.....	50
MOBİL UYGULAMA VE BULGULAR.....	50
4.1. Senaryo Ve Gerçekleşme Adımları	50
4.1.1. Mobil Uygulamanın Tanıtımı	50
4.1.2. Örnek Kullanım Senaryosu	53
4.1.3. Kurulum ve Kullanım Adımları	54
4.1.4. Verilerden Rapor ve Grafik Oluşturma	55
4.1.5. Grafik Verileri Üzerinden Kural Tanımlama.....	56
4.2. Kurallar Ve Alarmlar.....	56
4.2.1. Kural/Alarm Oluşturma.....	56
4.2.2. Kuralların/Alarmların Faydaları	56
4.3. Envanterler	57
4.3.1. Sensör Ve Aktüatörler	57
4.3.2. Node'lar.....	60
4.3.3. Cihazlar	61
4.4. Geliştirme Aşamaları.....	61
4.4.1. Node Geliştirilmesi.....	61
4.4.2. Gateway Entegrasyonu.....	62
4.4.3. Mobil Uygulama Geliştirilmesi	62
4.4.3.1. Melez(HYBRID) Ve Yerel(NATIVEAPP) Uygulama Farkı	64
4.4.3.2. ES6 Avantajları	65

4.4.3.3. RCAPI ve RDX Yaklaşımları.....	68
4.4.3.4. Projenin Dizin Ağacı(Directory Tree)	71
4.4.3.5. Bulut'a İstek(Request) ve Sonuç(Response) Örnekleri	71
4.4.4. Mobil-Web Farklılıkları	73
4.4.5. Test Geliştirme Ve Doğrulama Adımları.....	74
4.5.Mimariler/Yapılar.....	74
4.5.1. IGNITE Mimarisi.....	74
4.5.2. Mobil Uygulama Mimarisi	75
4.5.3. Node-Gateway Entegrasyon Mimarisi	76
4.5.4. Genel Sistem Mimarisi	76
BEŞİNCİ BÖLÜM.....	78
AĞ GEÇİDİ UYGULAMASI.....	78
5.1.Android İşletim Sistemini Ağ Geçidi Olarak Kullanmak.....	78
5.1.1. Ağ Geçidinin Tanıtımı ve Çalışma Mantığı	78
5.1.2. Ağ Geçidi Ve Ağ Tarama Teknolojisi.....	79
5.2.Android Servis Mimarisi.....	82
5.3.Android Servis Olarak Ağ Geçidi Uygulaması	83
5.4.Android Servis Uygulaması Ve Activity Farkı	85
5.5.Test Kodlarının Yazılması ve Doğrulama Aşamaları.....	85
5.6.İmzalama ve Yükleme İşlemleri.....	87
5.7.SONAR İle Proje Ve Kod Analizi.....	89
5.8.Birim Testleri ve Entegrasyon Testleri.....	90
SONUÇ	92
KAYNAKLAR	95
EKLER.....	108
EK A	108
EK B.....	108
EK C.....	109
EK D	110
EK E.....	110
EK F	111
EK G	112
ÖZGEÇMİŞ	114

ŞEKİL LİSTESİ

Şekil 1: Akıllı ev için sistem mimarisi örneği	11
Şekil 2: Kullanım şeması diyagramı.....	12
Şekil 3 : Nesnelerin internet üzerinden birbirine bağlanması.....	18
Şekil 4: Bulut bilişimin yapısı	22
Şekil 5: Bulut depolama servisleri.....	26
Şekil 6: IBM bağlı-akıllı ev modeli	31
Şekil 7: 2015-2015 yılları arasında bağlı cihaz sayıları(milyon).....	39
Şekil 8: Akıllı bir evin kullanıcıdan bulut'a genel akış diyagramı	42
Şekil 9: Akıllı ev mobil uygulaması - giriş ekranı.....	50
Şekil 10: Akıllı ev mobil uygulaması - kullanım amacının belirlenmesi	51
Şekil 11: Akıllı ev mobil uygulaması - kullanım senaryosuna göre modüller.....	51
Şekil 12: Akıllı ev mobil uygulaması - kurulum senaryosuna göre modüller	52
Şekil 13: Akıllı ev mobil uygulaması - günlere göre ortalama sıcaklık.....	53
Şekil 14: Akıllı ev mobil uygulaması - örnek sensör ekleme ekranı	53
Şekil 15: Envanter – ısı nem sensörü.....	57
Şekil 16: Envanter – m ^q -6 gaz sensörü	57
Şekil 17: Envanter – alev algılayıcı sensör.....	58
Şekil 18: Envanter – manyetik kapı alarm sensörü.....	58
Şekil 19: Envanter – hareketlilik algılayıcı sensör	59
Şekil 20: Envanter – buzzer.....	59
Şekil 21: Envanter – led	59
Şekil 22: Envanter – fan	60
Şekil 23: Envanter – su algılayıcı sensör.....	60
Şekil 24: Akıllı ev mobil uygulaması – örnek düğüm(node) tanımlama verisi(JSON)	62
Şekil 25: Melez(HYBRID), yerel(NATIVEAPP) uygulama ve RN	65
Şekil 26: Promise ve callback kodlama farkı	66
Şekil 27: Örnek şablon değişken(template literal) kullanımı	67
Şekil 28: Örnek ok(arrow) fonksiyonu kullanımı.....	67
Şekil 29: Örnek RCAPİ kullanım diyagramı.....	69
Şekil 30: Örnek RCAPİ oluşturma	70
Şekil 31: RCAPİ' dan TOKEN ve device bilgilerini alma ve FETCH İsteği yapma	71
Şekil 32: Projenin izin ağacı ve SONAR analiz sonucu	71
Şekil 33: Bulut' a yapılan örnek bir http get isteği ve sonucu	72
Şekil 34: Bulut'a yapılan örnek bir http post isteği ve sonucu	73
Şekil 35: IGNITE mimarisi	74
Şekil 36: IGNITE bulut platformu ve mobil uygulama mimarisi	75
Şekil 37: IGNITE düğüm(node) ve ağ geçidi(gateway) mimarisi	76
Şekil 38: Genel sistem mimarisi.....	77
Şekil 39: Ağ geçidi – geliştirilen uygulama	79
Şekil 40: Ağ geçidi – UDNS ve MDNS farkı.....	80
Şekil 41: Ağ geçidi – JMDNS ile ağ tarama.....	81

Şekil 42: Ağ geçidi – JMDNS java implementasyonu	81
Şekil 43: Android servis yaşam döngüsü	83
Şekil 44: Ağ geçidi – GRADLE ile bağımlılık yönetimi.....	84
Şekil 45: Ağ geçidi – uygulamaya ağ ve internet erişim yetkisi.....	85
Şekil 46: Ağ geçidi – guard clause ile test caseleri.....	86
Şekil 47: Ağ geçidi – keytool komutuyla imza oluşturma.....	87
Şekil 48: Ağ geçidi – GRADLE imzalama ayarları	87
Şekil 49: Ağ geçidi – komut satırı üzerinden Android APK oluşturma	88
Şekil 50: Ağ geçidi – bulut platformuna uygulama yükleme	88
Şekil 51: Ağ geçidi – properties dosyasında SONAR tarama ayarları	89
Şekil 52: Ağ geçidi – SONAR tarama komutu.....	89
Şekil 53: Ağ geçidi – başarılı SONAR taraması sonucu	90
Şekil 54: Ağ geçidi – SONAR analizi	90
Şekil 55: Ağ geçidi – mockito ile birim(unit) testi	91
Şekil 56: ES6 ve FETCH API ile bulut’ a giriş yapma	108
Şekil 57: Üretici(provider), tüketici(consumer) ve RAPI kullanımı	109
Şekil 58: JSON formatında çoklu dil desteği İngilizce olarak kullanımı.....	110
Şekil 59: IGNITE ile mq-6 gaz sensörünün düğüme tanıtılması	110
Şekil 60: Mobil uygulamada sensör verilerinin tutulduğu sensordata tablosu	111
Şekil 61: Arduino ide üzerinden “sketch data upload” işlemi	112
Şekil 62: GIT kod gönderme grafiği.....	113
Şekil 63: GIT kod gönderme geçmişi.....	113

KISALTMALAR

IOT	: (Internet of Things) Nesnelerin interneti
WSDL Dili	: (Web Service Description Language) Web Hizmetleri Tanımlama Dili
SNIA Kurumu	: (Storage Network Industry Association) Depolama Endüstri Kurumu
DBMS	: (Database Management Systems) Veri tabanı yönetim sistemleri
DB	: (Database) Veri tabanı
HDFS	: (Hadoop Distributed File System) Hadoop dağıtık dosya sistemi
SOAP	: (Simple Object Access Protocol) Basit Nesne Başarı Protokolü
SHEMS Yönetim Sistemi	: (Smart Home Energy Management System) Akıllı Ev Enerji Yönetim Sistemi
DO-SS Sistemi	: (Document Oriented Storage System) Belge Odaklı Depolama Sistemi
OO-SS	: (Object Oriented Storage System) Nesneye Yönelik Bir Depolama Sistemi
MFM	: Media Follow Me
PPM	: (Prediction by Partial Matching) Kısmi Eşleme ile Öngörü
PAAS	: (Platform as a Service) Hizmet olarak Platform
SAAS	: (Software as a Service) Hizmet olarak Yazılım
IAAS	: (Infrastructure as a service) Hizmet olarak Altyapı
HEMS	: (Home) Ev Enerji Yönetimi Sistemi
EV	: (Electronic Vehicle) Elektrikli araç
UI	: (User Interface) kullanıcı arayüzünü
WSAN	: (Wireless) Kablosuz Ağ Sensör Aktüatör Ağları
FETCH	: Ecmascript 6 ile yapılan http isteği
API	: (Application) Uygulama Programlama Arayüzü
IGNITE	: IOT-ignite
IGNITE-AGENT	: IOT-ignite ajan uygulamaları

IGNITE-API	: IOT-ignite açık API'leri
R-CONTEXT	: React Context
RCAPI	: React Context Api
ES6	: Ecmascript6
ES5	: Ecmascript5
MDNS	: (Multicast DNS) Noktadan Çok Noktaya
UDNS	: (Unicast DNS) Noktadan Noktaya
JMDNS	: Java Multicast DNS
MVN	: Apache Maven
GRADLE	: (Gradle) Groovy tabanlı altyapı otomasyon sistemi
DEBUG	: (Debug) uygulamanın geliştirme esnasındaki hali
RELEASE	: (Release) Uygulamanın canlıya çıkmış hali
JUNIT	: Java Unit Test
JAR	: (Java Archive) Java Aşiv Çıktısı
APK	: (Android Package Kit) Android Uygulama Paketi
SONAR	: Kod analiz tool'u
SONAR-S	: Kod analiz için yapılan tarama
HYBRID	: WEB uygulamasının mobile dönüştürülmesi
NATIVEAPP	: Android işletim sistemi için geliştirilmiş yerel mobil uygulama
RN	: (RN) Mobil uygulama geliştirme kütüphanesi.
RDX	: (Redux) React'ta statelerin tutulduğu teknoloji.
RCMP	: (React Component) React ile geliştirilen bileşen
GIT	: (Github) Web tabanlı versiyon kontrol sistemi
REPO	: Github üzerindeki proje
COMMIT	: Github'a gönderilen değişiklikler
TOKEN	: Sistemlere giriş yapıldıktan sonra alınan erişim anahtarı
JSON	: (Javascript Object Notation) Programlama dilleri arasında veri değişim objeleri biçimi.

BİRİNCİ BÖLÜM

GİRİŞ

Bu çalışmada temelde bağlı ev (connected) ve ilgili diğer kavram olan akıllı ev (smart home) tasarımı incelenmiştir. Bu amaçla nesnelerin interneti (IOT), bulut bilişim gibi konular araştırılmıştır. Araştırmanın son bölümünde bir bağlı/akıllı ev tasarımı modeli örneklendirilmiştir.

Son on yılda, çoklu-medya (multi-medya) kullanımı özellikle internetin gelişmesine paralel olarak çok hızlı bir şekilde artmıştır. Flickr, Youtube ve Facebook gibi sosyal paylaşım sitelerinde tutulan çoklu-medya verileri çok büyük boyutlara ulaşmıştır. Arama motorları da günümüzde çoklu-medya arama özelliği sunduklarından, bu boyutta görüntü verilerinin kaydedilmesini ve işlenmesini yönetmek zorundadırlar.

Bu kadar büyük çoklu-medya verisinin tutulması ve paralel olarak işlenmesi için dağıtık sistemler kullanılmaktadır. Dağıtık sistemin yeni eklenen verilerle güvenli bir şekilde büyüebilmesi ve bu veriler üzerinde paralel işleme yapılabilmesine olanak sağlaması gerekmektedir. Görüntülerin sınıflandırılabilmesi ve içeriğinde çeşitli aramalar yapılabilmesi için görüntü işleme algoritmalarının da bu dağıtık sistem içerisinde hızlı ve paralel bir şekilde çalıştırılabilir olması gerekmektedir.

Ekonominin hızlı gelişimi, insanların yaşam standardı ve kendi kalitelerini geliştirmeleriyle, yaşam ortamına daha yüksek gereksinimler getiriyor ve bilgi ve ağ akıllı evin temel çağrışımı haline gelmesiyle artık arzulanan ev talebi kavramı da değişmektedir. Akıllı ev ağı teknolojisi ve ilgili kablosuz ağ teknolojisi platformu, akıllı ev ağının geliştirilmesinde iyi bir rol oynamıştır. Bunun ana fikri, kişiselleştirilmiş ev bilgi ağı ile kişisel bilgi terminali için akıllı ara bağlantı sağlayan geleneksel kablolar yerine radyo sinyali kullanmaktır (Han ve diğerleri, 2014b).

Son yıllarda, bulut bilişim teknolojinin genel görünümü değiştirmektedir. Çalışma süresinin kısaltılmasının yanı sıra, fiziksel altyapının dağıtılması riskinin en aza indirilmesi ve yeşil bilgi işlemenin (green computing) gerçekleştirilmesi, belki de en çok göze çarpan özellik, bulut tabanlı bir sistemin sağlayabileceği ölçeklenebilirlik olacaktır.

Bulut bilişim, sistem genişletmek için yeterince esnektir, yeni hizmetler veya uygulamalar sunmayı kolaylaştırır. Kullanıcılar, İnternet'e erişebildikleri sürece bu hizmetlere istedikleri zaman erişebilirler. Akıllı ev teknolojileri, yaşam kalitesini iyileştirmek için on yıldan uzun süredir sürekli olarak gelişimini sürdürmektedir. Akıllı ev dediğimiz yapı, çeşitli sensörler ve ağ altyapısı ile donatılmıştır ve bu alandaki en büyük sorun, farklı satıcılardan sağlanan heterojen uygulamalar arasındaki entegrasyon ve etkileşimdir. İhtiyaç duyulan şey, akıllı evlerden üretilen verileri ve akıllı evdeki birlikte çalışabilirliği kullanan dayanıklı ve tutarlı hizmetler ile katma değerli uygulamalardır. Akıllı evdeki sensörler, neredeyse her saniye değerli bilgiler anlamına gelebilecek veri üretiyorlar, ancak bu veri miktarı, kendisinin yerel düzeyde tutulması veya analiz edilmesi için çok fazla olabilir (Kong ve diğerleri 2015).

Akıllı ev teknolojileri, ev ağlarından ve multimedyaadan çeşitli ev otomasyon sistemlerine kadar hızla gelişmiştir. Özellikle, bu teknolojiler çoğunlukla bireysel yaşam alanlarıyla ile sınırlı olmasına rağmen, ev enerji yönetiminde de yaygın olarak kullanılmaktadır. Nesnelerin İnterneti (IOT) sayesinde, akıllı ev teknolojileri, geleneksel sensörler ve uzaktan kumandalardan akıllı ev aletleri ve robot sistemlerine kadar çeşitli akıllı cihazları entegre etmeye başlamıştır. Bu alanda bugüne kadar birçok yenilikçi uygulama geliştirilmiştir. Son zamanlarda, akıllı ev teknolojileri, katma değerli hizmetler, işlemler ve yönetim sağlamak için bulut tabanlı hizmetler ile entegre edilmeye başlanmıştır (Kong ve diğerleri 2015; Han ve diğerleri, 2014a; Han ve diğerleri, 2014b).

Bilim ve teknoloji alanındaki bilgi birikimi geliştikçe, yaygın olarak kullanılan anlamıyla, en erken akıllı ev tasarımı olarak Bill Gates'in geleceğin evini gösterilebilir. Gates'in eviyle ilgili örnek olarak, ev ziyaretinde bir misafir varsa özel elektronik broş kullanılacaktır, broş yoluyla otomatik olarak konuk ilgili bilgileri kaydedebilir ve daha sonra bilgi vermek için bu bilgileri göndermek için broşun kapsamlı bir analiz yoluyla bir merkezi kontrol ile bir bilgisayara bilgi göndererek aydınlatma, akustik, klima ve bunun gibi ilgili ekipmanların yerli yerinde ve zamanında çalışması sağlanabilir.

Basit eylemler yoluyla insan doğasının tam otomatik ayarlanması gerçekleştirilebilir. Nesnelerin interneti, aslında akıllı ev konseptini uzun zaman önce ortaya koymuş ve bu teknolojinin yıllar boyunca gelişmesi sayesinde toplum nezdinde de daha fazla ilgi görmeye başlamıştır. Bununla birlikte, günümüzdeki bazı teknik

kısıtlamalardan dolayı, bu teknoloji piyasada yaygın bir şekilde yayılmadığı gibi, sıradan insanların evlerine beklenen düzeyde etkili olamamıştır. Tabii bu da akıllı ev tasarımlarının gerçek yaşamda yaygınlaşma hızını etkilemektedir. Tüm bunların yanında son yıllarda akıllı mobil terminalin geliştirilmesi ve yaygınlaştırılması ile 3G ve 4G ağı, akıllı ev için daha büyük geliştirme alanını yarattı ve insanların adı geçen teknolojileri sıklıkla kullanması, akıllı evlerin kullanım alanlarının gelişmesi ve yaygınlaşması için nesnelerin internetinin temel teknolojiler olarak kullanılmasına katkı sunmaktadır. Günümüzde akıllı evlerin temel ortamında hem teknoloji hem de kalite daha da gelişmiştir. Sürekli gelişen her yeni teknoloji ile akıllı ev dinamik ve sürekli bir değişimin ve gelişim göstermektedir (Han ve diğerleri, 2014a)

Nesnelerin interneti çağında, temel internet teknolojilerinin gelişmesine paralel olarak, bilim ve teknolojinin donanım ile yazılımın arasındaki etkileşimi hızlı bir biçimde güçlendirdiğini, evlerde kullanılan temel teknolojik aletlerin gün geçtikçe daha iyi işler yapabildiğini (yerel ağ teknolojileri, kablosuz internet, akıllı aletler gibi), akıllı telefonların etki alanlarının gittikçe genişlediğini ve daha da akıllandıklarını ve ilgili diğer teknolojilerin daha da çeşitlenip becerikli hale geldiği görülmektedir. Tüm bunlar bağlı-akıllı ev tasarımlarına önemli kapılar açmaktadırlar.

Sıradan evlere kıyasla, bağlı-akıllı evler yalnızca konaklama, güvenlik sağlama, aile yaşamı için yüksek refah ve konfor sağlama fonksiyonunu icra etmiyorlar, aynı zamanda yaşanan yeri pasif statik yapıdan dinamik akıllı bir yapıya dönüşerek, aile üyeleri için hayatı kolaylaştırmaya dönük uzaktan kontrol olanakları sunmakta ve geniş bir yelpazede ev aletleri arasında etkileşime imkân tanımaktadırlar. Bağlı-akıllı evler sakinlerinin yaşam stillerine ve zamanlarını etkili kullanmalarına yönelik katkılar sunduğu gibi, evin güvenliğini üst seviyeye taşımakta ve enerji tasarrufu bağlamında ev sakinlerine destek olabilmektedir. Ev sakinleri için bahsi geçen akıllı web servisleri kaynak, dönüştürücü ve akıllı uygulama katmanlarını içermektedir (Porter ve diğerleri, 2013).

Kaynak katmanı bağlı-akıllı ev sistemini destekleyen bir algı ve işlem arayüzüdür ki, veriye erişen cihazlar arasında bir geçit (gateway) işlevi görürler. Diğer taraftan kaynak katmanı ayrıca veri uyarılma katmanı bilgilerini de aktarabilir. Kaynak katmanının aile geçidine dayandığı kabul edildiğinde, bu katmanın evdeki cihazların

dinamik arayüz ayarlarına ve standart işlem arayüzlerini bir araya toplayan bir yapı olarak tasarladığı düşünülebilir (Hu ve diğerleri, 2013).

Akıllı ve tamamlayıcı uygulama katmanı; kullanıcıların tercihlerini, alışkanlıklarını ve durumsal geribildirimlerini algılayarak, kullanıcıların kişiselleştirilmiş davranışlarından çıkarımlar yapmak suretiyle akıllı ev hizmet modunu işlevsel hale getirir. Web hizmetleri, temel farklı platformlar arasında, web servis modunu kullanarak, genişletilebilir biçimlendirme dilinde, heterojen verilerin ayrıştırılması ve genel adaptasyonunu tamamlamak için bir veri temsili biçiminde kesintisiz bir etkileşim gerçekleştirebilir, uygulama katmanı arasında işlevsel ve duyarlı kesintisiz etkileşim sağlarlar. Uyarılama katmanı kullanıcı deneyimi geliştirmek ve sistemin tepki hızını artırmak için heterojen verileri göz önünde bulundurmaz, böylelikle diğer iki katman için problemler yaratabilir (Tsai ve Pan, 2015).

Fiziksel bir mekâna yerleştirilmiş akıllı ortam, ses tanıma, bilgisayar görüşü ve etkileşimli duvar projeksiyonu ve diğer modal etkileşim yetenekleri, bilgisayarın bakışından gizlenmiş, insanların niyetlerini belirleyebilir ve insanların daha verimli çalışmasına ya da iyileştirilmesine yardımcı olmak için uygun geri bildirim veya eylemler yapabilir. İnsanların hayatlarının kalitesini artırabilir. Akıllı ortam fiziksel ortama ilişkilendirilme, ses tanıma, bilgisayar bakışı ve dokunmatik duvar projeksiyonu ve diğer etkileşim yetenekleri yoluyla, insanların niyetlerini belirleyebilir ve insanların daha etkili çalışabilmeleri ve hayatlarını daha kaliteli yaşayabilmeleri amacıyla onlara uygun geribildirimler ve eylem tavsiyeleri sunabilir.

Bağlı-akıllı ev sistemleri ev sahiplerine merkezi ısıtma, görüntülü görüşme, ışıklandırma, perde açma kapama, akıllı sinema ve ses sistemleri, havalandırma, arka planda çalan müzikler ve bunun gibi daha birçok olanaklar sunabilir.

1.1. Araştırmanın Amacı

Bu çalışmada, nesnelerin interneti ve uçtan uca akıllı ev sistemi yapımı üzerine araştırmalar yapılmıştır. Günümüzde enformasyon son derece gelişmiştir ve tüm iletişim ve internet teknolojileri insan medeniyeti üzerinde büyük ilerlemeler sağlamıştır. Buradan

hareketle insanların yaşamını daha konforlu, daha etkili hale getirmek amacıyla bağı-akıllı ev konsepti doğmuştur.

Araştırmamızın temel sorusu: Bir evi, bağı ve akıllı kılan nedir? Diğer bir deyişle bir evin, bağı ve akıllı olma durumu hangi koşullara bağıdır. Bu temel soru etrafında “*nesnelerin interneti nedir ve kullanım alanları nelerdir, bulut bilişim ile nesnelerin interneti arasındaki bağılantıyı nasıl anlamalıyız, nesnelerin interneti ve bulut bilişim hizmet alanları nasıl genişlemektedir, bağı/akıllı evler enterkonnekte, zeki ve enstrümanlı yapılar mıdır?*” gibi sorulara da yanıtlar aranmıştır.

1.2. Araştırmanın Alana Katkıları

Bağı ev ve akıllı ev konuları gün geçtikçe insanlar ve toplumlar nezdinde daha fazla karşılık bulmaktadır. Ekonomik refahın artması, bilgi işlem teknolojilerinin gelişmesi, mobil iletişimin yaygınlaşması gibi etkenler bağı ve akıllı ev tasarımlarına ilgiyi artırmaktadır. Bir çalışanın işinden çıkarken telefonundan evdeki yemeği ısıtması için fırına komut vermesi, yıkanmış çamaşırlarının kurutulması için çamaşır makinesini programlaması, eğer hava soğuksa evin ısıtma sisteminin çalıştırılması ve benzeri birçok senaryo bağı ve akıllı ev tasarımları ile gerçeğe dönüşmektedir. Dahası evin kimi bölümlerinin daha fazla kimi bölümlerinin daha az aydınlatılabilmesi, ısıtmanın homojen ya da heterojen dağıtılabilmesi, ses düzenlemelerinin belirli ortamlarda sınırlı tutulabilmesi ya da ev geneline yayılabilmesi, enerji harcamalarının kontrol edilerek mümkün olan en fazla tasarrufun sağlanabilmesi, evin farklı bölümlerinde yaşayan sakinler arasında bilgi, belge, dosya ve çoklu medya paylaşılabilmesi ve sayısı artırılacak birçok fayda bağı/akıllı evleri daha da arzulanabilir kılmaktadır.

Böyle olunca bağı ve akıllı evlere yönelik sektörler de gelişime açık olmaktadır. Teknolojiyi olabildiğince kullanıp mümkün olan en fazla konfora ulaşma isteğı bu sektörün gitgide gelişmesini sağlamaktadır. Bağı/akıllı ev konusu her ne kadar bilimsel alanda yaygın bir biçimde tartışılıyor olsa da birçok insanın bu konudaki bilgi birikimi istenen düzeyde değildir.

Bu araştırmada, bağı/akıllı ev konusu, nesnelerin interneti ve onunla ilişkili olan bulut bilişim ele alınıp bir model önerisi sunulmaktadır. Bu yönüyle araştırmanın

baęlı/akıllı ev konusunu bir bütn olarak ele alması ve ilgili literatre rnekler sunması arařtırmayı nemli kılmaktadır.

1.3. Tez Yazım Dzeni

Bu alıřma beř temel blmde hazırlanmıřtır. İlk blm arařtırmanın konusunu, amacını, ilgili literatre katkısını ve tezin blm yapısını sunan tez yazım dzenini iermektedir. alıřmanın ikinci blm kapsamlı bir literatr taraması sonucu ulařılan bilgilerin derlenmesini iermektedir. Bu blmde baęlı/akıllı ev kavramı, akıllı ev aletleri, oklu ekranlar, bulut tabanlı servisler, nesnelerin interneti, bulut biliřim, bulut biliřim uygulama servisleri ve modelleri, bulut biliřimin kullanım biimi ve bulut biliřimdeki riskler konuları iřlenmiřtir.

Arařtırmanın nc blm nesnelerin interneti ve utan uca baęlı/akıllı ev tasarımı modeli iřlenmiřtir. Bu blmde baęlı evin temel grevleri, ev aletlerini buluta baęlamının avantajları, nesnelerin interneti destekli baęlı/akıllı evler, baęlı/akıllı ev kullanıcı ve sistem uygulamaları ile rnek baęlı/akıllı ev modeli tanıtılmıřtır.

Arařtırmanın drdnc blmnde bir baęlı/akıllı ev konusunda bir geliřtirilen bir uygulama ve bu uygulamanın detayları iřlenmiř, arařtırmanın son blm olan sonu, arařtırmada ulařılan bulguları deęerlendirmekte ve baęlı/akıllı evlerin geleceęini sorgulamaktadır.

İKİNCİ BÖLÜM

İLGİLİ LİTERATÜR

2.1. Bağlı/Akıllı Ev

Bağlı ev, aynı zamanda akıllı ev olarak da ifade edilebilir, gelişmiş bilgisayar teknolojisi, ağ iletişim teknolojisi, sensör teknolojisi, entegre kablolama teknolojisi ile akıllı kontrol teknolojisi, ev ve kontrol merkezi aracılığıyla güvenlik, çevre kontrolü, iletişim, akıllı ev aletleri ve diğer alt sistemler gibi ev hayatının tüm alanlarında elde edilen bilgileri kullanarak tüm bu karmaşık sistemi organik olarak birleştirmektedir.

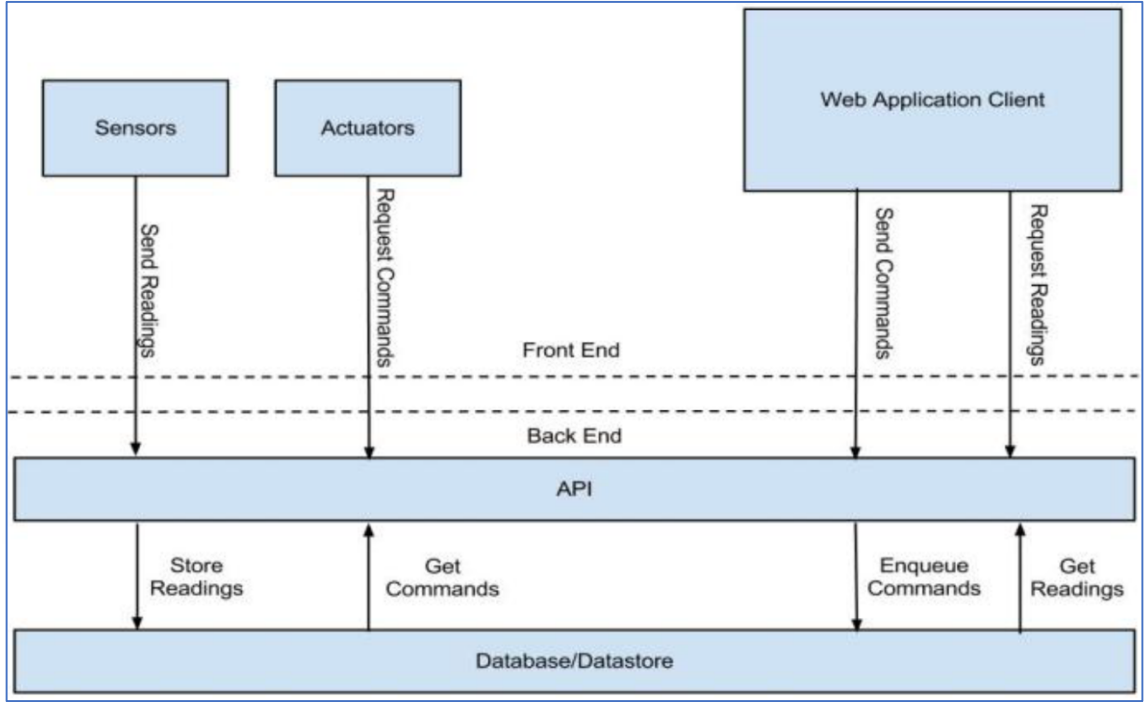
Akıllı Ev (Smart Home), kullanıcının nem, sıcaklık, parlaklık gibi ev koşullarını ölçmesi, ev tipi ısıtma, havalandırma ve klima cihazlarını manipüle etme ve minimum kullanıcı müdahalesi ile durumlarını kontrol etme potansiyeli vaat etmektedir. Araştırmacılar ve uygulayıcılar kavramı kolaylaştırmak için büyük çaba sarf etmektedirler. Örneğin, akıllı ev yönetimi için, ev bağlamını temsil eden bir alan-nesne hiyerarşik modelini kullanan, kaynaktan haberdar bir yönetim sistemi öngörülür. Teknik olarak, mobil cihazlarla ev bilgilerinin uzaktan erişimini sağlamak için Web Hizmetleri Tanımlama Dili (WSDL) ve Basit Nesne Başarı Protokolü (SOAP) kullanılır. Etkin enerji yönetimi için Han ve diğerleri, IEEE802.15.4 ve ZigBee tabanlı yeni bir Akıllı Ev Enerji Yönetim Sistemi (SHEMS) önermektedir (Han ve Lim, 2010).

Toplam enerji maliyetini azaltmak için SHEMS tabanlı çoklu algılama ve ışık kontrol uygulaması tasarlanmıştır. Kullanıcılarına hizmet veren akıllı bir evin “ev doğası” ile ilgili olarak Wu ve diğerleri, hizmetler, alanlar ve kullanıcılar arasındaki etkileşim ilişkisini modellemek ve akıllı bir evin insan merkezli etkileşim gereksinimini karşılamak için bir çerçeve sunmuşlardır. Çerçeveyi kullanarak, “Media Follow Me (MFM)” ve “Ubiquitous Skype” gibi iki yaygın uygulama geliştirdiler. Kullanıcı aktivitesini tahmin etmek için, Alam ve diğerleri gelişmiş bölüm keşfi yoluyla dizi tahmini içeren SPEED olarak adlandırdıkları bir algoritma önermişlerdir. SPEED, kullanıcı davranışını dizi etkinliklerinden oluşan ayrı bölümlere sınıflar. İnsan aktivite modellerine dayanarak, sonlu bir sıralama Markov modelinde temsil edilen bölümleri ve kullanıcı eylemlerini ve tahmin doğruluğunu geliştirmek için kısmi eşleme ve öngörü (PPM) algoritmasını çıkarmak için SPEED kullanmışlardır. Chen ve diğerleri de akıllı

evlerde çoklu sensör veri akışlarına dayalı gerçek zamanlı, sürekli etkinlik tanıma için bilgi odaklı bir yaklaşım geliştirdiler. Yaklaşım, bağlam ontolojileri modelleme, durum oluşturma süreci ve bilgi odaklı bir etkinlik tanıma mimarisinden oluşur. Bu şekilde, etkinlik tanıma, düşük seviyeli sensör veri toplama, orta düzey veri füzyonundan yüksek düzey etkinlik tanımaya kadar gerçekleştirilebilir. SemWeb semantik teknolojileri, semantik verilerin oluşturulması, yönetimi ve sorgulanması için kullanılmıştır. Ayrıca bu konuda örnek olarak verebileceğimiz Euler çıkarım motoru semantik akıl yürütme için kullanılmaktadır (Han ve Lim, 2010; Son ve arkadaşları, 2011; Wu ve Fu, 2012; Alam ve diğerleri, 2012; Chen ve diğerleri, 2012). Söz konusu araştırma çabaları, içerik bilincinin, enerji verimliliğinin, doğal etkileşimin ve kullanıcı etkinliğini tanıma gibi bağlı/akıllı ev özelliklerine odaklanmaktadır.

Günümüzde nesnelerin interneti ile bulut bilişim kavramlarının entegre edildiği görülmektedir. Nesnelerin interneti (IOT), bilgisayar zekasını ev cihazlarına yerleştirir ve kullanıcıya ev koşullarını ölçmek ve ev aletlerini izlemek için uygun bir yol sunar. Bulut bilişim, ev hizmetlerini geliştirmek, sürdürmek ve çalıştırmak için ölçeklenebilir bilgi işlem ve depolama gücü sağlar. Buna ek olarak, bulut bilişimin kullanılması, kullanıcının ev cihazlarına her zaman ve her yerde erişmesine (izlemesine ve / veya kontrol etmesine) olanak tanır (Solimani ve diğerleri, 2013).

Solimani ve arkadaşlarının (2013) tasarladıkları akıllı tasarım örneği, akıllı evler için sistem mimarisi ve kullanım örneği sunmaktadır. Örnek incelendiğinde, öncelikle akıllı ev için sistem mimarisinin, ev koşullarının ölçülmesi, enstrümantasyon verilerinin işlenmesi ve ev aletlerinin izlenmesinin gereklerini yerine getirmesi gerektiği söylenebilir. Örnek olarak, ev koşullarını ölçmek için mikrodenetleyici özellikli sensörleri ve ön uçtaki ev aletlerini izlemek için mikrodenetleyici etkin aktüatörleri kullanır. Bu amaçla, arka uçtaki verileri işlemek için PAAS'ı (Hizmet olarak Platform) ve SAAS'ı (Hizmet olarak Yazılım) Bulut bilişimde kullanmaktadır. Şekil 1'de akıllı ev için sistem mimarisini örneği verilmiştir.



Şekil 1: Akıllı ev için sistem mimarisi örneği

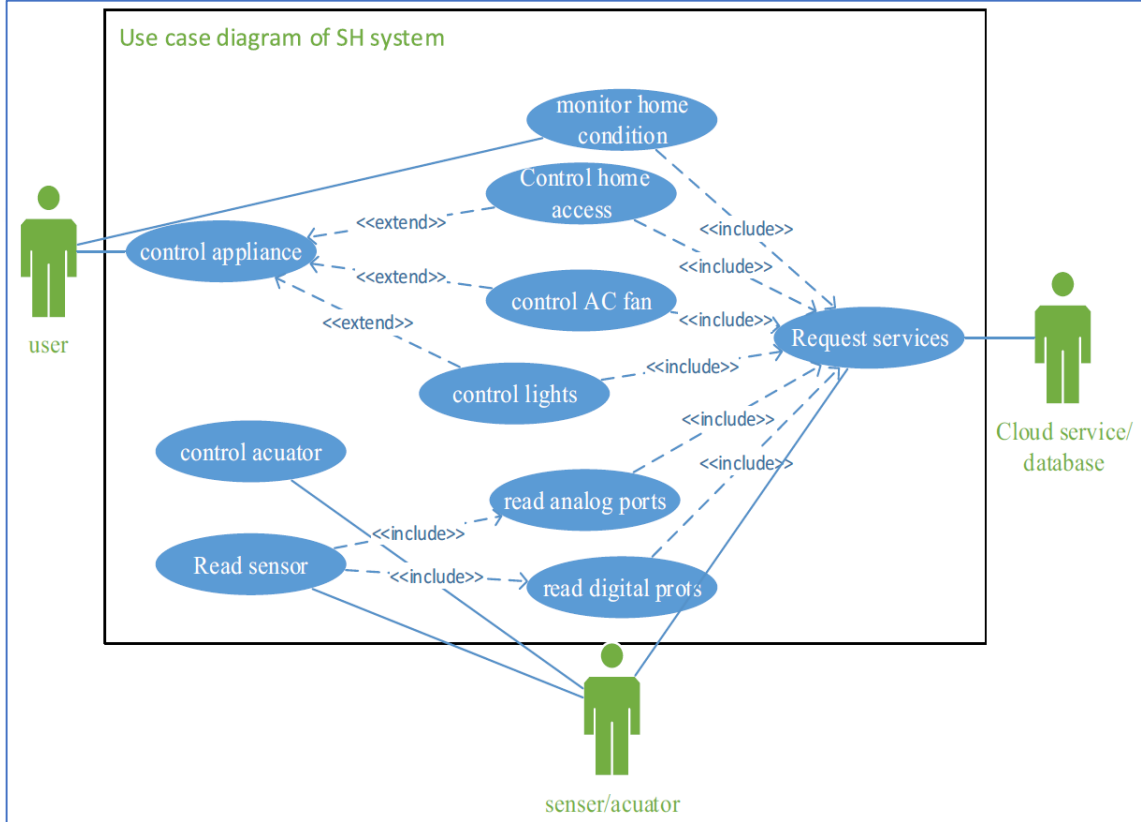
(Solimani ve diğerleri 2013).

Bu örnekte akıllı evin sistem mimarisinin beş bileşeni olduğu görülmektedir (Solimani ve diğerleri, 2013):

- Mikrodenetleyici özellikli sensörler: Mikrodenetleyici, ev koşullarını ölçer; aletlerden gelen verileri yorumlar ve işler.
- Mikrodenetleyici etkin aktüatörler: Belirli eylemleri gerçekleştirmek için mikrodenetleyici tarafından aktarılan komutları alır. Komut, mikrodenetleyici ile Bulut hizmetleri arasındaki etkileşime dayanarak verilir.
- DB / Veri Deposu: Veri analizi ve görselleştirme için mikrodenetleyici özellikli sensörlerden ve bulut hizmetlerinde veri depolar ve aynı zamanda aktüatörlere komut kuyruğu olarak gönderilir.
- Arka uç ile ön uç arasındaki Sunucu / API katmanı: Sensörlerden alınan verilerin işlenmesini ve verilerin DB’ de saklanmasını kolaylaştırır. Ayrıca, aktüatörleri kontrol etmek ve komutları DB’ de saklamak için web uygulama istemcisinden komutlar alır. Aktüatörler, DB’ deki komutları sunucu üzerinden kullanma isteklerini yaparlar.

- Bulut hizmetleri olarak hizmet veren web uygulaması: Sensör verilerini ölçmeyi ve görselleştirmeyi ve mobil cihaz (örneğin, akıllı telefon) kullanarak cihazları kontrol etmeyi sağlar.

Detayları yukarıda verilen mimarinin uygulamadaki örneği Şekil 2’de verilmiştir.



Şekil 2: Kullanım şeması diyagramı

(Solimani ve diğerleri 2013).

Şekil 2’de örnekleri anlatılan sistem mimarisinin kullanım koşullarını görebileceğimiz üç farklı kullanma durumu aşağıda verilmiştir (Solimani ve diğerleri, 2013):

- Ev koşullarının ölçülmesi. Bu birincil kullanım durumlarından biridir. Kullanıcı, ev koşullarının, örneğin sıcaklık ve nemin okumalarını almak için gerçekleştirdiği eylemi açıklar.

- Ev aletleri yönetme. Kullanım durumu, kullanıcının ışıklar, kapılar, fanlar ve klimalar gibi çeşitli ev aletlerini kullanabileceğini gösterir.
- Ev erişimini kontrol etme. Bu kullanım durumu, kullanıcının bir eve erişim sağlamak ve etkinliklerini kaydettirmek için RFID kartlarını kullanabileceğini belirtir. Bu, kullanıcının bir evde bulunmasını izlemeyi sağlar.

2.2. Bağlı/Akıllı Ev Çalışma Alanları

Bağlı/akıllı ev konusunu güçlendirmeye dönük bugüne değin birçok araştırma yapılmıştır. İlgili literatür incelendiğinde, bağlı/akıllı ev sistemleri ile ilgili çalışmaların öncelikli olarak üç alanı vurguladığını göstermiştir: *Akıllı cihazlar, çoklu ekranlar ve bulut tabanlı hizmetler.*

2.2.1. Akıllı Ev Aletleri

1. Ev Enerji Yönetimi Sistemi (HEMS): Akıllı Evlerin Ana Akımı

HEMS konusundaki çalışmalar, enerji-ev yapımı mekanizmalarını araştırmak için basit ev aletleri ölçümlerini içeren ev enerji yönetimini incelemekten geçmiştir. HEMS çalışmalarında bulut hizmet platformlarının çevre veri toplama ile bütünleştirilmesi zaman içerisinde yavaş yavaş vurgulanmıştır (Kong ve diğerleri, 2015; Han ve diğerleri, 2014a, 2014b).

2. Solar Paneller: Yenilenebilir Enerji Üretmek İçin Popüler Kaynaklar

Enerji üretmek için güneş panelleri sıklıkla kamusal alanlarda kullanılır, böylece HEMS uzayları ve topluluk alanları arasında bir çakışma olur. Elektrikli araç (EV) şarj sistemleri ve kamu mülklerinin izlenmesiyle ilgili merkezi yönetim sistemleri gibi bir toplumda çeşitli enerji ile ilgili sistemler bulunabilse de alan çakışması toplum operasyonlarının ve bakımın bütünleştirilmesi ile ilgili endişeleri beraberinde getirmektedir (Lee ve diğerleri, 2014; Tsai ve diğerleri, 2013; Wi ve diğerleri, 2013).

3. Enerji tasarrufu ve yenilenebilir enerji üretiminin ana teknolojilerindeki ilerlemelerle, akıllı evler üzerindeki çalışmalar, akıllı ev sistemlerinde insanların ve olayların, kontrollü ev aletlerinin, kullanıcı etkinliklerinin ve memnuniyetinin ve yaşam

ortamlarının hiyerarşik ilişkileri ve bekleme davranışları gibi konuları incelemeye odaklanmıştır. Artan miktarlarda bilgi barındırmak için, daha önce de belirtildiği gibi çevre bilgisine erişim sağlamak için değil, aynı zamanda kullanıcıların enerji tüketiminin geçmiş verilerini depolamak için de ultra large bulut hizmeti platformları geliştirilmiştir. Bu nedenle, bu tür bulut servis platformları akıllı ev sistemleri ile yakından bağlantılıdır (Byun ve diğerleri, 2013; Jo, Kim ve Joo, 2013; Liu ve diğerleri, 2013).

4. *Ev Güvenliği ve Sağlık Hizmetleri:* Akıllı ev çalışmalarında enerji yönetiminin yanı sıra ev güvenliği ve sağlık hizmetleri de araştırılmıştır. Akıllı ev sistemleri, genellikle ev güvenliği ile ilgili anormal olayları tanımlamak ve değerlendirmek için çeşitli sensörler ve gözetleme kameralarını entegre eder. Evde sağlık bakımında, akıllı ev sistemleri sadece vücuda özgü sensörleri değil, aynı zamanda profesyonel tıbbi ve sağlık hizmetleriyle uzak bulut platformlarından kaynakları birleştirir (Kim ve diğerleri, 2013; Wang ve diğerleri, 2014; Tung ve diğerleri, 2013).

5. İlgili literatür incelendiğinde birçok alanda akıllı ev sistemlerinin uygulandığı görülmektedir. Güvenlik ve gizliliğin yeni uygulamalar ürettiğinden emin olmak için bulut platformlarındaki etiket verilerini depolamak ve analiz etmek amacıyla büyük veri platformları kullanılabilir. Tartışılan alanlara ek olarak, çeşitli alanlar robot yenilikleri ve etiketleme tekniklerini uygulamak için akıllı ev sistemleri kullanılmıştır. Burada iki örnek verilebilir. Birincisi kameralarla birleştirildiğinde, robot sistemleri ortamları ayırt edebilir ve geri bildirim sağlayabilir; bu nedenle, özel, yenilikçi robot tasarımları sıklıkla ev otomasyonuna uygulanır. İkincisi nesne tespiti için etiketlerin uygulanması IOT teknolojilerinin temelidir. (Qiao ve diğerleri, 2013; Huynh ve diğerleri, 2014; Kumar ve Lee, 2015).

6. *Sensörler ve ağ teknolojileri yenilikçi mimarlık uygulamalarını geliştirmek için kullanılmaktadır:* (1) El-jest tanıma özelliğine sahip ev otomasyon sistemleri, (2) fotoğraflardaki aktivitelerin tespit edilmesine dayanan otomatik fotoğraf öneri yaklaşımları, (3) duygu tanıma ve çevre videolar için interaksiyon sistemleri ve (4) aileler için uzun mesafeli görsel paylaşım sağlayan ev ağı sistemleri (Erden ve Çetin, 2014; Tian ve diğerleri, 2013; Kim ve diğerleri, 2013).

Lee ve diğerlerine (2016) göre akıllı cihazlar farklı arayüzlere sahip farklı sensörler arasındaki iletişimi sağlamalıdır. Ancak Mevcut mimarilerin tasarımları,

önceden tanımlanmış ve genişletilebilir iletişim arayüzlerine sahip standart bir arayüz kontrol modülünden (ör., Ev kontrolörü) yoksundur. Ayrıca, akıllı cihazlar sensör fonksiyonları sağlamalı ve akıllı ev sistemlerini topluluk operasyonlarına ve yönetimine entegre etmelidir. Ancak, mevcut mimarilerin tasarımları, topluluk operasyonlarını ve yönetimini çevre bilgilerine göre yerel ortamlarla bütünleştiren sistemle bütünleşik bir bileşen (örneğin, topluluk komisyoncusu-community broker) barındırmamaktadır

2.1.2. Çoklu Ekranlar

1. Akıllı ev sistemleri, kurulduğu ortama göre görüntüleme cihazlarını (örneğin, tablet bilgisayarlar ve akıllı telefonlar) seçebilir ve buna bağlı olarak ilgili cihaz üzerindeki kullanıcı arayüzü (UI-user interface) ile gereken akıllı ev ayarlamaları yapılabilir. Akıllı ev sistemlerinin ilgili cihazlar arasındaki etkileşimleri sağlamasının yanı sıra çevreleyen sensörleri veya uzak sunucuları entegre etmesini sağlayan bir trend geliştirilmiştir. Dahası, televizyonlar geleneksel bir ev ekranı arayüzüdür. Çoklu ekranlarla ilgili yeni çalışmalar, yeni operasyonel yenilikler, sensör entegrasyonu, değerli servislerin uygulamaları ve farklı ekran sistemlerini içermektedir (Costa ve diğerleri, 2013).

Çoklu ekranlar, görüntüleme cihazlarını seçmeli ve kullanıcı arayüzlerini medya ortamlarına göre ayarlamalı, böylece yüksek teknoloji akıllı ev sistemlerini basit ve kullanımı kolay bir eve dönüştürmelidir. Ancak, mevcut mimarilerin tasarımları, çoklu ekranların gerekliliklerini yerine getirebilen, tekdüze bir kullanıcı ara yüzüne (ör., Ev kontrolörü) sahip değildirler.

2.1.3. Bulut Tabanlı Servisler

Ev Sakinlerinin Yaşamına Yönelik Bulut Tabanlı Hizmetler: Yüksek ev otomasyonu elde etmek için, üçüncü taraf sunucular ve yapılandırılmış akıllı ev sistemleri, interhome, çoklu cihaz akıllı ortamlarında veri gizliliği ve kimlik doğrulama sorunlarına yönelik olarak önerilmektedir. Akıllı ev sistemleri hem bina içi hem de açık hava senaryoları ile birlikte akıllı bina sistemlerine genişletilmektedir. Ayrıca, mimarinin dinamik ara tabakalarında bölgesel çevre bilgisi de kullanılmaktadır.

Cabarcos ve diğerklerine (2012) göre bulut tabanlı hizmetler ve akıllı ev sistemlerini entegre etmek kimi zorlukları da beraberinde getirmektedir:

- (1) Kullanıcı iletişim kanalları ve sunucular, veri iletişim kanalları geliřtirmek için hiyerarşik olarak bağlanmalıdır.
- (2) Yazılım güncellemesi bölgesel yazılım dağıtımını dikkate almalıdır.
- (3) Bulut kaynaklarını kullanmak, kişisel ve grup verilerine erişmek için güvenlik işlevleri gerektirmelidir.
- (4) Akıllı ev sistemlerinde arızaları yönetmek için, bulut sunucuları, farklı zaman noktalarındaki olayların karşılaştırılması yoluyla hataları algılamak için yerel sunucuları yöneten çekirdek kaynakla işbirliği yapmalıdır.

Buraya kadar yapılan literatür taraması sonucunda özetle şunlar söylenebilir: Bulut hizmetleri hiyerarşik mimarilerde uygulanmalı ve bölgesel, konum tabanlı hizmetler (örneğin, bilgi hizmetleri ve ev işletmesi ve yönetimi gibi) sağlamak için yapılandırılmış akıllı ev sistemleriyle birleştirilmelidir. Ancak Mevcut mimarilerin tasarımları akıllı bir topluluk yapılandırma modülünden yoksundur. örneğin iletişimin, akıllı ev sistemlerine konum bazlı, çok amaçlı uygulamalar yapmak ve topluluk operasyonlarını ve yönetimini genişletmek gibi kullanılması.

2.3. Nesnelerin İnterneti

Nesnelerin İnterneti (IOT), nesnelerin günlük yaşamdaki bilgilerinin bağlanması ve işlenmesini ifade eder. Daha özel olarak, IOT, sensörler ve aktüatörler gibi çevremizdeki benzersiz olarak tanımlanabilir cihazlara erişmeyi ve bunları kontrol etmeyi mümkün kılan birçok farklı teknolojidir oluşmaktadır. Bu terim ilk olarak 1998 yılında Kevin Ashton tarafından yapılmıştır ve daha sonra “bilgisayarların gerçek dünyayı anlaması için standart bir yol” olarak tanımlanmıştır.

Terim geçmişte temel olarak, günlük nesnelere radyo frekansı kimlik denetleyici (RFKD) etiketleri ve benzer teknoloji ile donatmak anlamına geliyordu. Fikir, bu şeylerin kendileri hakkındaki bilgileri başka bilgisayarlara ve gömülü cihazlara raporlamak için araçlara sahip olması gerektiğiydi, örneğin mikrodalgayı yiyecek pişirmek için gereken

tüm bilgileri sağlayan bir RFKD etiketi içeren dondurulmuş gıda ambalajı gibi. Bu teknoloji, esas olarak, sevkiyatlarını ve ürünlerini takip etmek için çok sayıda lojistik ile uğraşmak zorunda kalan şirketler tarafından benimsenmiştir.

Gündelik nesnelere kullanıldığında, bu teknoloji tek başına, bugün bizi ilgilendiren birçok sorunla başa çıkmak için büyük miktarlarda ek bilgi sağlayarak, örneğin atıkların azaltılması gibi, toplumun üzerinde önemli bir etkisi olabilir. RFKD ile atık ve genel olarak enerji tüketimi (örneğin fotovoltajik bir kollektöre sahip olduğunda, koşullar optimum olduğunda (gün içinde, işyerinde çalışırken) otomatik olarak programlanabilir. Ancak Nesnelerin İnterneti fikri, o zamandan beri, her yerde / Kablosuz Ağ Sensör ve Aktüatör Ağları (WSAN) ile birbirine bağlı olan ve evlerimizdeki kaynaklardan (sensörler, anahtarlar, vb.) bağımsız olarak toplanan verileri işleyecek ve daha güçlü bir biçimde etkinleştirilmiş tümleşik ağ aygıtlarını kapsayacak şekilde evrilmiştir. Ayrıca Nesnelerin İnterneti (özellikle Nesnelerin Web'i), makedeki okunabilir semantik bağlamı sağlamayı amaçlayan Semantic Web denen şeydir, böylece makineler bu verinin bilgi içeriğini daha verimli bir şekilde işleyebilir ve böylece insan girdisi olmadan kendi başına daha fazla bilgi türetebilir.

Zamanla nesnelerin interneti ile ilgili araştırma ve uygulama miktarı giderek daha fazla dikkat çekici hale gelmiştir. Nesnelerin interneti, yeni yüzyılda ülkeler için temel ulusal rekabet gücünün korunması ve ülke gelişiminin hızlıca ilerlemesi için önemli alanlar arasında görülmektedir.



Şekil 3 : Nesnelerin internet üzerinden birbirine bağlanması

(<https://justcreative.com/2018/11/19/internet-of-things-explained>)

Tahmin edilebilir bir şekilde, nesnelerin internetinin gelişmesiyle birlikte geniş bir inovasyon alanı oluşacaktır. Teknik açıdan bakıldığında, nesnelerin interneti doğrudan bilgisayar endüstrisinin, iletişim ağının, sensör endüstrisinin teknik inovasyonunun ve otomatik kontrolün gelişmesini destekleyecektir. Diğer taraftan, endüstriyel tasarım perspektifinden, ürün tasarımının arka planında yer alan nesnelerin interneti geniş bir inovasyon alanına sahiptir.

İnternet, ağ iletişimi temelinde kademeli olarak hızlı bir gelişme göstermektedir. İnternetin gelişme süreci, sistematik tasarım eksikliğinden dolayı ağın verimliliği, güvenlik, ölçeklenebilirlik ve inovasyon yeteneği gibi birçok zorlukla karşı karşıyadır(Liu ve Yang, 2017):

Ağ güvenliği: Akıllı terminal erişimi ve ara bağlantı sahibi geniş kullanıcılar için ağ güvenliği büyük önem taşımaktadır. Aynı zamanda, ticarileşmenin genişlemesiyle

birlikte ağ saldırılarını önleme ilgi alanları arasındadır. İnternet güvenliği riski giderek daha fazla ön plana çıkmaktadır. Bu aşamada, bulut bilişim gibi teknolojinin ve uygulamanın geliştirilmesi de İnternet güvenliği için görece yeni bir konudur. Sadece bulut bilgisinin sınırsız olarak uygulanmasında değil, likidite ve bunun hareketliliği de birçok yeni güvenlik sorununu tetiklemekte ve temel bulut bilişim uygulamalarını kullanan yüksek konsantrasyondaki kullanıcılar ve bilgi kaynaklarının bilgisayar korsanlarının hedefi haline gelme olasılığını artırmaktadır.

Ağ genişletilebilirliği: Ağ genişletilebilirliği için her şeyden önce, adres alanı genişletilebilirliği gösterilebilir. Herhangi bir zamanda ve herhangi bir yerde terminal alma ile herhangi bir kullanıcının bunu kullanması İnternetin gelişiminin temel koşuludur. Çünkü IPv4 adres alanı nispeten dar ve dengesiz tahsisattır. Şu anda, IPv4 adres kaynaklarının çoğu tükenmektedir. IPv6, ağ adresi alanına dayalı olarak, gelecekte İnternet'in uzun vadeli gelişimini karşılayabilir, ancak IPv4 ve IPv6 arasındaki uyumsuzluk nedeniyle, İnternet adresleme alanı dönüşümü uzun ve karmaşık bir süreç olacaktır.

Yenilik yeteneği: İnovasyon, İnternetin birincil canlılık kaynağıdır. Kullanıcıların İnovasyon deneyiminin rehberliği, İnternet'in İnovasyonunun ana gelişim yönünü belirlemektedir. Gelecekte, topluluk ve bireyler için kişiselleştirilmiş İnternet uygulamaları ana akımlar haline gelecek ve verimli bir temel yenilik gereksinimine dönüşecektir.

Ağ performansı: İnternetin popülaritesi toplumun genel enerji tüketimini azaltmaya yardımcı olabilir, ancak aynı zamanda İnternetin yüksek enerji tüketimi sorunu da vardır. Uzun zamandır bu alandaki öncelik ağların kapsamına genişletmek ve artan şebeke inşaatlarını devam ettirmektir. Bu öncelik İnternetin, yüksek enerji tüketimini ve sürdürülebilir kalkınmayı ciddi biçimde sınırlandırmasına yol açmıştır.

Nesnelerin İnterneti, akıllı nesnelere İnternete bağlamak için geliştirilmiştir. Akıllı bir nesne, ağ üzerinden bilgi toplayıp paylaşımlarını sağlayan sensörler, yazılım, elektronik ve bağlantı ile bütünleşmiş her şeyi ifade eder. Bu konuda daha fazla verimlilik sağlayan siber-fiziksel sistemlerin daha doğrudan entegrasyonu için nesnelerin İnterneti (IOT) fırsatları geliştirilmelidir (<https://www.micrium.com/iot/devices>).

IP'nin nesnelerin interneti açısından önemli olması, doğal olarak IP dışı sistemlerin işe yaramayacağı anlamına gelmemektedir. Sadece IP dışı sistemlerin internete ulaşmaları için bir ağ geçidi gerektirdiğini ima ederler. Örneğin, kablosuz sensör düğümleri, ağ üzerinde çok sayıda konuşlandırılmış olan ve internet protokolü bağlanabilirliğini içermeyen düşük maliyetli cihazlardır. Bu nedenle bu araçların sağlıklı bir şekilde işleyebilmesi için kablosuz sensör düğümleri ve IP ağı arasında bir ağ geçidi olarak hareket edebilen bir WSN kenar düğümü gerekmektedir (<https://www.micrium.com/iot/devices>).

Nesnelerin internetine ait temel teknikleri şu şekilde sıralamak mümkündür: akıllı teknoloji, sensör ağı ve radyo frekansı kimlik teknolojisi. Nesnelerin internetinin anahtar teknolojilerinden olan akıllı teknoloji, kullanıcılarıyla aktif ve pasif iletişimi gerçekleştirmek için her türlü yol ve araç bilgisini kullanarak çeşitli kaynaklardan gelen verileri işleyerek yapılan bir tahminin amacına ulaşmasını sağlamaktadır. Sensör ağı ve kablosuz sensör ağları konusunda son yıllarda çok sayıda bilimsel araştırma yapılmaktadır ve bunlar çeşitli protokol katmanı yığınlarındaki sorunlara, enerji tasarrufu, ölçeklenebilirlik, güvenilirlik ve konularında çözüm önerileri geliştirmeye odaklanmışlardır. RFKT (radyo frekansı kimlik teknolojisi) etiketli bir nesne özel okuyucudan geçtiğinde, etiket okuyucusu aktif hale getirilecek ve etiket bilgileri radyo dalgaları tarafından okuyucuya ve bilgi işlem sistemine iletilecek, böylece süreç bilgisinin gereksinimlerine uygun bir biçimde, kontrol ve bilgi işlemden sorumlu olan bilgi toplama çalışmasını tamamlayacaktır (Puustjärvi ve Puustjärvi, 2015).

Yeni bir ağ işlem sistemi olarak, internet mimarisi ve ağ işlem sisteminin yapısı gereği, kendinden beklenen talepleri etkili bir biçimde yerine getirebilmek için öncelikle nesnelerin internetinin tasarım ve uygulaması öncelikle hazırlanmalıdır. Nesnelerin interneti enformasyonun küresel izleme ve paylaşımının yapılmasına olanak veren yeni bir ağ mimarisidir. Günümüzde, nesnelerin interneti teknolojisinin sensör ağına ilişkin araştırmalar şu konularda yoğunlaşmaktadır (Puustjärvi ve Puustjärvi, 2015):

- Sensör ağının, nesnelerin internetinin en altında olması nedeniyle kendisine ait bütünlük ve verimlilik gibi ağ performansı parametleri hayati önemdedir.

- Entegre sensör teknolojisi, gömülü bilgisayar teknolojisi ile, gerçek zamanlı monitörle işbirliği gibi dağıtılmış bilgi işlem teknolojisi, her türlü çevresel veya izleme nesnesi bilgisini algılar ve toplar ve işleme ve teslimatı gerçekleştirir.

- Ağ topolojisi, nod hatası oluştuğunda kendi kendini organize eden, otomatik konfigürasyon kabiliyetini ve genişletilebilir özelliklerini dikkate alarak değişmelidir.

- Sensör ağı, ağdaki tehdidi kontrol ederek, temel bilgi açıklama, kurcalama, tekrar saldırı, hizmet reddi ve diğer tehditlerin genel kablosuz ağına ek olarak, yüz algılayıcı düğümleri de saldırgan fiziği tarafından kolayca manipüle edilir ve sensör düğümünde saklanan tüm bilgileri alır.

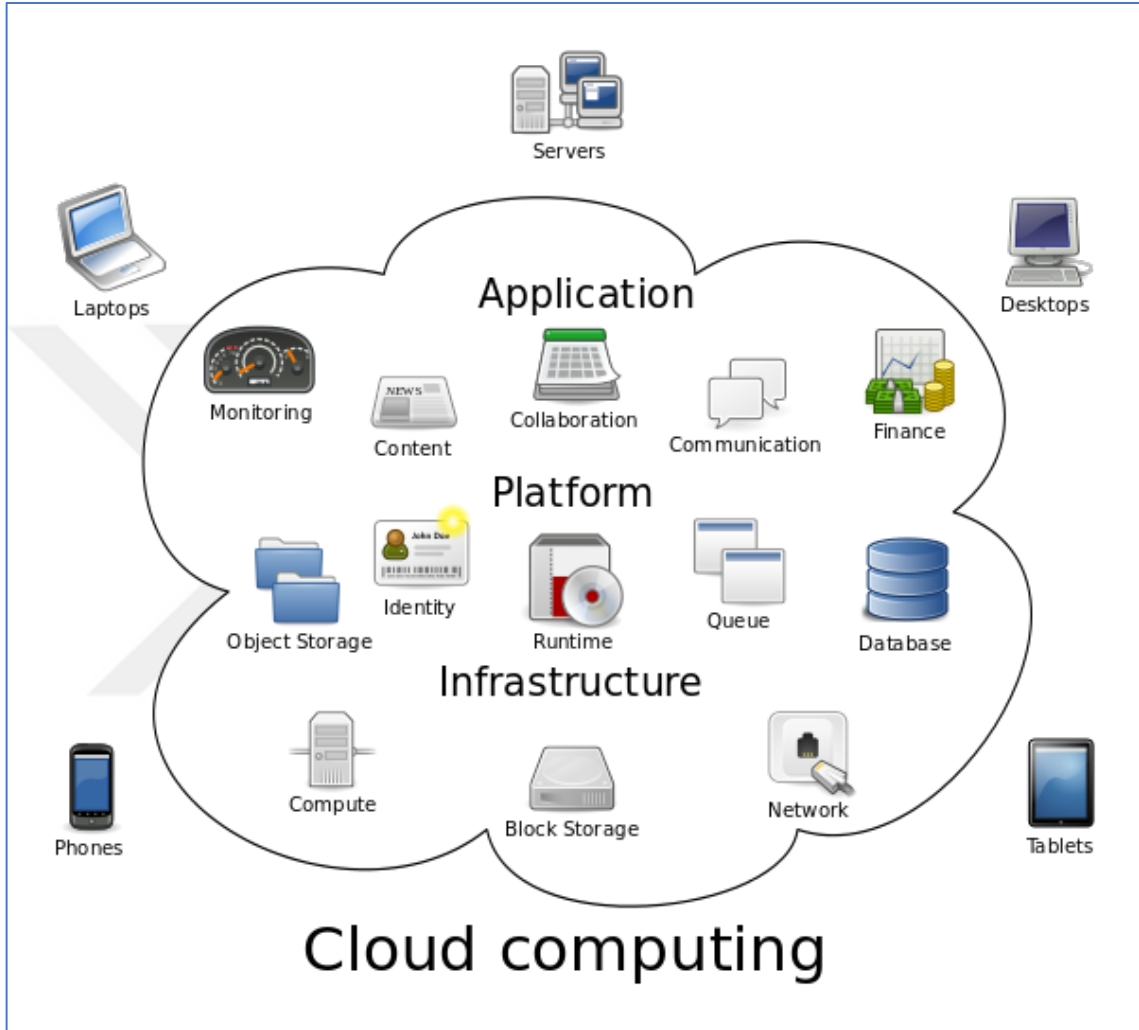
Nesnelerin interneti (IOT) temel öğrenme ve öğretme süreçlerine de katkı sunmaktadır. Sürekli teknoloji geliştirme, öğrencilerin öğretim sürecine daha canlı, yeterli, uyarlanabilir ve sessiz bir şekilde katılmalarını sağlar. Akıllı öğrenme, ileri yaşta öğrenmeyi canlandıran bir fikirdir ve farklı uygulamalarla etkinliği artmış ve genişlemiştir. Akıllı öğretme, sınıf temelli ayrık yönelimi, temelli öğrenmeyi, bireysel-temelli özelleştirilmiş öğrenmeyi ve kitle tabanlı üretken öğrenmeyi bir araya getirir (Shebab ve diğerleri, 2016).

Akıllı öğrenme sistemi, kişiselleştirme için müşteriyle ilişki kurması beklenen gelişmiş geliştirilen bir bileşeni olarak IOT'yi kullanır ve yeni özelleştirilmiş öğrenme araçları sağlar. Çok yönlü işlem geliştirmeleri, kullanıcılarla ilişki kurmak, sürekli alıştırmalarını hissetmek ve akıllı öğrenme sistemi için herhangi bir açık kapı olduğunda onlara destek vermek için kullanılmaktadır. Ayrıca, IOT araçları son zamanlarda veri boyutu, işlem hızı vb. ile ilgili zorlukların yönetilmesini sağlamak için büyük veri tabanlı uygulamalarla birlikte kullanılmaktadır (Digolo ve diğerleri, 2011).

2.4. Bulut Bilişim

Bulut bilişim (cloud computing) veya işlevsel anlamıyla çevrimiçi bilgi dağıtımı; bilişim aygıtları arasında ortak bilgi paylaşımını sağlayan hizmetlere verilen genel addır. Bulut bilişim bu yönüyle bir ürün değil, hizmettir. Temel kaynaktaki yazılım ve bilgilerin paylaşımı sağlanarak, mevcut bilişim hizmetinin bilgisayarlar ve diğer aygıtlardan

elektrik dağıtıcılarına benzer bir biçimde bilişim ağı (tipik olarak İnternet'ten) üzerinden kullanılmasıdır (wikimedia.org). Şekil 4'te bulut bilişimin yapısı örneklendirilmiştir.



Şekil 4: Bulut bilişimin yapısı

(https://upload.wikimedia.org/wikipedia/commons/b/b5/Cloud_computing.svg)

Bulut bilişimin ve hizmetin tanımı konusunda üzerinde geniş bir mutabakat bulunan tanımlardan birinde; bulut bilişimin dağıtık işlem, paralel işlem ve grid hesaplamaların veya bilgisayar bilimi kavramının ticari olarak uygulanmasının geliştirilmesi olduğu ifade edilmektedir.

Liu ve Yang'a (2017) göre bulut bilişim, ağ üzerinden başvurmak için gereken kaynakları elde etmek için bir çeşit dağıtım ve kaynak kullanımı modudur. Bulut bilişim,

"bulut" üzerindeki müşteri terminalinden hesaplanırken, kullanıcılara internet üzerinden birden fazla bilgisayar tarafından ortak bir işleyişle, dağıtılmış bilgi işlem teknolojileri ile sağlanan bir uygulama gibidir. Kullanıcılar sadece işlevin, uygulamanın uygulanışını önemser ve hizmet sağlayıcının gerçekleştirdiği uygulama ve bakımın gerçekleştirilmesinin gerçekleştirilme şeklini umursamazlar, kullanıcı kendi ihtiyaçlarına uygun uygulamayı seçer. Bulut bilişim bir araç veya mimari değil, bir hesaplama şeklidir.

Bulut hizmeti aşağıdaki teknikleri içerir (Liu ve Yang, 2017):

1. Büyük miktarda veri yönetimi teknolojisi: Bulut bilişimin çok büyük miktarda veri işleme dağıtması gerekir ve bu nedenle veri yönetimi teknolojisi büyük miktarda veriyi verimli bir şekilde yönetebilmelidir.

2. Büyük miktarda veri depolama teknolojisi: Bulut bilişim sistemi, aynı anda çok sayıda kullanıcıya hizmet vermek için çok sayıda sunucudan oluşur; bu sayede bulut bilişim sistemi, verilerin depolanması için daha fazla depolama güvenilirliği sağlayan verileri depolamak için dağıtılmış depolama yöntemi ile çalışır.

3. Programlama modeli: Programlamanın bulut bilgi işlem ortamlarının programlama modelinin katı olması çok basittir. Dağıtılmış bilgi işlemin etkisini elde etmek için çok sayıda bilgisayar işlemine atanmış olan ve daha sonra Reduce programı ile harita programının bloklara bölünmesiyle ilişkili olan, harita ve azaltma işlemlerini gerçekleştiren çıktı sonuçlarını toplayacak grafikler modelidir.

2.4.1. Bulut Servisleri

Bulut bilişim, donanımdan son kullanıcı uygulamalarına kadar tüm ihtiyaçlara yönelik hizmetler sunar. Sunucular, depolar, yönlendiriciler, anahtarlar gibi donanım kaynaklarına kiralık erişim ve ayrıca son kullanıcılar için talep üzerine gerekli uygulamaları sağlarlar. Bu alanda üç servis kategorisi önerilmektedir. Bu kategoriler (SAAS, IAAS ve PAAS) aşağıdaki gibi açıklanmıştır (Reed ve diğerleri, 2007):

- *Hizmet Olarak Yazılım (SAAS):* Eksiksiz bir yazılım uygulaması sağlayan talep hizmetidir. Yazılım, kuruluştaki tek bir bilgisayara kurulur ve birden çok kullanıcı, kuruluş içindeki bulut üzerinden ona erişir.

- *Hizmet olarak Altyapı (IAAS):* Genellikle ekipman, sistem, yazılım ve hizmetlerden oluşan bir bilgi teknolojisi (BT) altyapısı oluşturmak için gereken tüm çözümleri sağlar. Ağda servis olarak depolama ve hesaplama özellikleri sağlar. Donanım sorunları için sunucular, anahtarlar, depolama çözümleri ve yönlendiriciler, vb. ve bilgi işlem amacıyla basit ve yüksek performanslı uygulamalardan her türlü uygulamayı sağlar.

- *Hizmet olarak platform (PAAS):* Üst düzey hizmetler oluşturmak için bir platform olarak platform sağlar. Platform, işletim sistemleri, uygulama yazılımı, güvenlik, ara katman yazılımı, depolama, programlama dili ve geliştirme ortamı gibi gerekli tüm kaynaklarla donatılmıştır.

Nesnelerin İnterneti (IOT) perspektifinde, milyarlarca fiziksel sensör ve cihaz, birçok heterojen, karmaşık ve yapısal olmayan veri sağlamak için internet üzerinden birbirine bağlanır. Sektördeki ve araştırma alanındaki birçok çaba, veri bakım ve analizi için maliyetleri ve performansı dengelemek amacıyla IOT verilerinin depolanmasına odaklanmıştır. Gerçekten de güçlü depolama sistemlerinin tasarımı, büyük veri uygulamalarının gereksinimlerini etkin bir şekilde ele alabilir ve bulut bilişimin IOT paradigmasında önemli bir rol oynaması beklenir. Bulut depolama alanı ölçeklenebilir bir şekilde büyük miktarda depolama ve işleme olanağı sunar. Bu nedenle, akıllı ortamlar (örneğin akıllı şehirler, iç güvenlik, afet önleme, vb.) için pek çok farklı uygulamada kullanışlı uygulama ve hizmetlerin geliştirilmesi için yararlanılabilecek büyük veri depolamak için gözetimli bulut mimarileri hâlihazırda tasarlanmaya devam etmektedir (Fazio ve arkadaşları, 2012; Jiang ve arkadaşları, 2014).

Burada bulut bilişim ile izleme faaliyetlerinden kaynaklanan Büyük Veri konularını analiz etmekte ve veri depolama, sorgulama ve geri alma işlemlerini optimize etmek amacıyla farklı veri türlerini desteklemek için yararlanılabilecek farklı depolama teknolojilerini tartışılmaktadır. Burada örnek olarak sunulan depolama mimarisi, Büyük Veri depolamasında hem belge hem de nesne yönlendirme depolama sistemleri yaklaşımlarını birleştirir, böylece farklı bilgi kaynaklarını ele alan benzersiz bir çözüm sunar. Ayrıca, bulut bilişim teknolojisini ölçeklendirilebilirlik ve güvenilirlikten

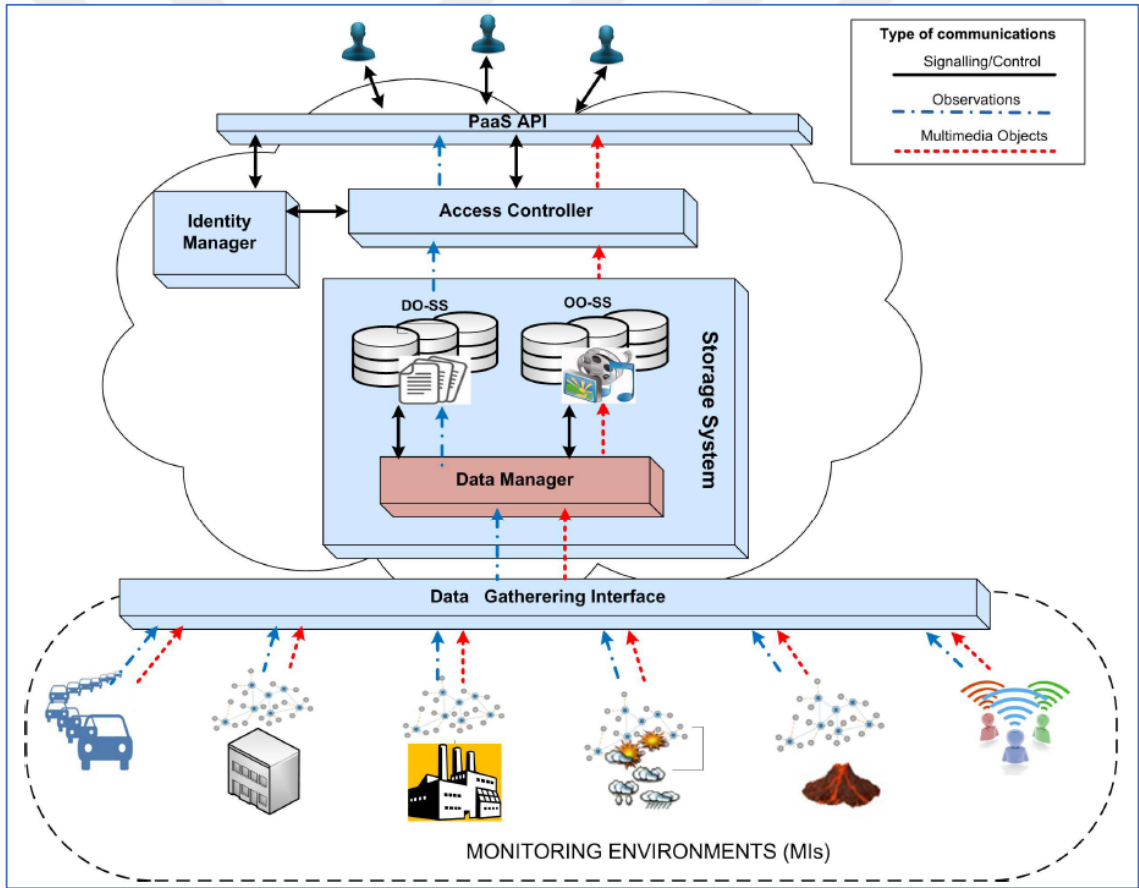
yararlanmak için kullanır. Bulut kullanıcısı açısından, izleme altyapısından toplanan veriler, Açık Geospatial Konsorsiyumu (OGC) tarafından tanımlanan Sensör Web Etkinleştirme (SWE) özelliklerine göre tasarlanan tek tip bir şekilde sunulur (Reed ve arkadaşları, 2007).

Sensörler ve Nesnelerin İnterneti (IOT) ile etkileşim halinde olan yeni bulut bilişim altyapıları, yakın zamanda literatürde daha çok ortaya çıkmaktadır. Tam Bağlı Araç sistemini desteklemek için yararlı bir Bulut Platformu, Hizmet mimarisi olarak çok daha ayrıntılı bir Platform olarak nesnelerin bulutu (CloudThings) olarak adlandırılır. Nesnelerin bulutu, BT pazarı (örneğin, Facebook, GAE), akıllı cihazlar ve gömülü sistemler (örneğin, Kablolama, Sun SPOT, mbed, Arduino) ve Bulut uygulamaları (Heroku, Paraimpu,) gibi geniş bir çerçevede yer alan bulut servislerini içerir (Shah ve diğerleri, 2013).

Uygulama, Finlandiya'nın Oulu şehrinde konuşlandırılan gerçek bir kullanım durumu olan Akıllı Ev senaryoları için uyarlanmış tüm kabul edilmiş çözümleri göstermektedir. Cloud Sensing, buluttaki algılama kaynaklarını yönetmek için iki farklı stratejiyi birleştiren ve son kullanıcının hangi tür bulut hizmetini seçmesi gerektiğini ücretsiz olarak belirleyen bir çerçevedir. Özellikle, çerçeve veri merkezli veya cihaz merkezli bir modele göre hizmet sunmaktadır: Birincisi, müşterilere sağlanan heterojen algılama / harekete geçirme verilerini soyutlayabilen ve depolayabilen bir PAAS (Hizmet olarak Platform) olarak uygulanmaktadır; ikincisi bir IAAS (Servis olarak Altyapı) olarak uygulanmakta, müşterilere bir algılama / harekete geçirme altyapısı sunmaktadır. Diğer bir üst düzey platform, Kablosuz Sensör Ağlarını bulut bilişim ile entegre edebilmektedir. Tüm bu platformlar aynı tip işlevleri ve öğeleri sunar. Bize göre, gerçek ilerleme için, heterojen sistemler arasında birlikte çalışabilirliği hesaba katmak gerekir (Ding ve diğerleri, 2012; Zhou ve diğerleri, 2012; Shah ve diğerleri, 2013).

Buradaki çalışma, şu anda SWE çerçevesini genişletme ve mevcut açık standartlarla uyumlu hale getirme fırsatlarını araştırabilen "IOT için Sensör Ağı" Standards Working Group oluşturmak isteyen OGC (O G Consortium) standardı olan SWE'ye dayanmaktadır. REST protokolünü kullanarak sensör arayüzlerinin ve sensör ağlarının Web-dostu ve verimli uygulamalarını barındırır. Burada Ballari ve diğerleri tarafından tanımlanmaya çalışılan ve gerçekte sorun olan şey şudur ki; yakınlık, bitişiklik

gibi duyuşal verilerin yeniden ölçeklenerek sunulması ve soyut hale getirilmesidir. Hatta onla farklı senaryolarda farklı dinamiklerle veri temsili bağlamlarını tanıtmışlardır. Küresel görüşün nasıl başarılacağını göz ardı ederek dinamik bir birlikte çalışabilirliğe sahip küresel bir model sağlamışlardır. Karar vericilerinin büyük miktarda gelen veriyi işlemeleri istenir, ancak böyle bir problemin (yani ölçeklenebilirlik problemleri) pratik olarak nasıl ele alındığı açık değildir (Ballari ve diğerleri, 2009). Şekil 5'te bulut servislerinin işleyiş prensibine örnek verilmiştir.



Şekil 5: Bulut depolama servisleri

Fazio ve diğerleri, (2015).

Akıllı ortamlardaki altyapıların izlenmesi, dünya çapında bir alana yayılmış farklı kiracılara aittir. Kiracıların verilerini bulut üzerinden paylaşmalarına olanak veren birkaç

olası model vardır. Örneğin, kiracılar, verileri ağ üzerinden açık algılama verileri olarak temin etmektedir. Bu durumda, bulut depolama sağlayıcısı, bu tür verilerin kendi sistemiyle bütünleştirilmesiyle ilgilenir; ya da kiracı hem kaynak sağlayıcı hem de tüketicidir ve bulut altyapısını bulut altyapısı aracılığıyla fiziksel altyapısını genişletmek için kullanır; aksi takdirde, bulut depolama sağlayıcısı ve kiracı şirket ticari anlaşmalar yapar.

İzleme altyapısı kiracıları ve bulut depolama sağlayıcıları arasındaki izleme altyapılarından gelen veriler çok heterojen yapıdadır. Bu tür veriler kabaca iki ana tipte sınıflanabilir:

1. *Gözlemler:* Algılayıcı aygıtlar tarafından gerçekleştirilen fiziksel veya oluşmuş olayların ölçümleri. Gözlemler tuple (anahtar, değer) ile ifade edilebilir ve ağ üzerinden iletilen metin dosyasında saklanabilir.

2. *Nesneler:* Bilgi içeriği işleme cihazları tarafından kaydedilen multimedya içerikleri (ör., Ses, görüntü, video ve animasyon). “Büyük Veri” nin anlamı, günümüzde çok büyük yapılandırılmamış veri kümeleriyle (PetaByte of data), saniyeler içinde verilen cevaplarla hızlı analitik ihtiyaçlarının ele alınmasını sağlamaktadır. Bununla birlikte, büyük verileri yönetme stratejileri, belirli bir veri türüne güçlü bir şekilde bağlıdır. Gözlemler bilgileri büyük veri üretebilir çünkü geniş bir coğrafi bölgedeki faaliyetlerin izlenmesi, kısa zaman aralığında birkaç guruba ait bilgi üretebilir. Böylece uzun dönemlerde (günler, aylar, yıllar) çok miktarda verinin yapılandırılması ve depolanması gerekmektedir. Gözlemler, standartlaştırılmış bir iç formatta saklandığı gibi yine kapsüllendiği belgeler aracılığıyla anlaşılabilir hale getirilip kullanılabilir. Etkili bir Belge Odaklı Depolama Sistemi (DO-SS) (örn., MongoDB, Cassandra, CouchDB), kolayca alınabilmesi için her belgenin içeriğini indeksler. Ayrıca, HTML, XML, JSON ya da en azından dışa aktarım yapabilen veya bunlara dönüşebilen sistemlerde çok sayıda yayın yapılmaktadır.

Nesneler büyük boyutlu dosyalar oluşturabilirler, ancak büyük veri sorunları sadece nesnelerin hacminden değil, aynı zamanda onların heterojen doğasına göre de ortaya çıkarlar. Aslında, belirli tipte verilere göre bir depolama sisteminde bir nesne bulmak için farklı türden sorgular yürütülebilir. Nesneye Yönelik Bir Depolama Sistemi (OO-SS) (örn., AWS S3, SWIFT, Kinetiç), nesneye yönelik programlama dili yetenekleri

ile depolama yeteneklerini (örn., Saydam olarak kalıcı veriler, eşzamanlılık kontrolü, veri kurtarma, birleştirici sorgular) birleştirir. Geleneksel yaklaşımlar, esas olarak, Nesneyi tanımlayan genişletilmiş öznitelikler olan meta verilere dayanır. OO-SS, ek verileri desteklemek için dosya meta verilerini açık bir şekilde ayırır ve çıkarılan meta veriler için kullanılan tipik biçimler XML, YAML ve JSON'dur. Bir nesne ile ilişkili bilgi şeması, belirli OO-SS'a bağlıdır, ancak genellikle, nesnenin oluşturulduğu iç özelliklerine (ör. Görüntü boyutu, sıkıştırma tipi, video süresi, görüntü çözünürlüğü) sıkı sıkıya bağlıdır (Fazio ve diğerleri, 2014).

2.4.2.Bulut Bilişim Uygulama Modelleri

Bulut bilişim farklı yollarla ve farklı mimarilerle uygulanabilir. Temel olarak üç model vardır: kamu, özel ve karma. Bunlar müşteri ihtiyaçlarına göre uygulanabilir. Üç model aşağıda açıklanmıştır (Barga, 2009):

Herkes için açık bulutlar: kullanıcı bağlamında herkese açıktır, internet üzerinden ulaşılabilen büyük miktarda kaynağa erişebilir. Bu bulutlar müşterinin konumundan uzaktadır ve farklı müşterilerden gelen uygulamalar bulut sunucu, ağ ve depolama sistemlerinde karıştırılabilir. Kamu bulutlarında çalışan uygulamalar hem sağlayıcılara hem de kullanıcılara açıktır. Kamu bulutları, diğer modellere kıyasla uygulama için daha ucuzdur.

Özel bulutlar: Özel bir kullanıcı için özel bulutlar oluşturulur ve normalde kendi konumunda bulunur. Müşteri altyapısına sahiptir ve uygulanacak veri, güvenlik ve uygulamalar üzerinde tam bir kontrole sahiptir. Bu, uygulama için daha yüksek maliyet gerektirir.

Hibrit bulutlar: Hem kamu hem de özel bulutların özelliklerini birleştirir. Bunlar, kamu kaynaklarının kullanımıyla özel bir bulut uygulama becerisine sahiptir. Müşteri lokasyonunda konuşlandırılan sunucudur, ancak genel bulut ile iletişim halindedir.

2.4.3.Bulut Bilişimin Kullanımı

Bulut bilişimi neden kullanıyoruz?

Bulut bilişim mimarisi, onu standart kurumsal bilgisayardan ayıran birçok avantaj sunar. Barga (2009) bulut bilişimin üç temel faydasını şu şekilde sınıflandırmaktadır:

1. Çalışma zamanı ve tepki süresinin azaltılması: Bulut bilişim, uygulamaların çok sayıda sunucuya erişmesini sağlar, böylece çalışma zamanını tek bir sunucu üzerinde çalışan uygulama ile karşılaştırıldığında azaltır. İyi bir örnek, New York Times'ın bulut bilişim yardımıyla çalışma süresini kısaltmasıdır.

2. Altyapı riskini en aza indirme: Bulut bilişim, örn. ekipman, sistem, yazılım veya hizmetler; Bulut bilişimde bir şirket veya bir kullanıcı bulut sunucularına erişebildiğinden, fiziksel sunucuları satın almak gerekli değildir. Bu, işleme kapasitesi bağlamındaki iş yükü arttığında, çok sayıda sunucuya ihtiyaç duyulduğunda, hızlı bir şekilde devreye alınabileceğini ima eder. Kullanıcılar kendi sunucularının kurulu olduğu özel bir bulut kullanıyor olsalar bile, iş yükleri arttığında bu yük genel bulutlara kaydırılabilir.

3. Düşük giriş maliyeti: Bulut bilişim yeni pazarlarda kurulum ve giriş maliyetini azaltır; gelişmiş BT altyapısına gerek yoktur. Düşük maliyetin ana nedeni, bulut bilişimde kurulu altyapının kiralanması, dolayısıyla sunucu satın almanın gerekmemesidir, bu nedenle ilk yatırım sıfır olabilir.

2.4.4. Bulut Bilişimde Riskler ve Dezavantajlar

Bulut bilişimdeki risk ve dezavantajları sıralamak istediğimizde ilk iki sraya gösterilen çabanın büyüklüğü ve masraflar ile güvenlik ve gizlilik koyulabilir.

Bulut bilişim kullanımında dikkate alınması gereken önemli bir konu, bulut sağlayıcının kapanması durumunda verilerin taşınmasıdır. Mearian'ın (2011) yazdığı bir makalede, bir bulut sağlayıcısının kişisel verilerini başka bir buluta taşıması için böyle bir yol bulunmadığını vurgulamaktadır. Bir bulut bittiğinde ya da sağlayıcı kapanırsa, verileri doğrudan müşteriye geri göndererek, verileri depolamak için başka bir sağlayıcı bulmalıdır. Ayrıca, veri silme için bilinen tatmin edici bir prosedür yoktur.

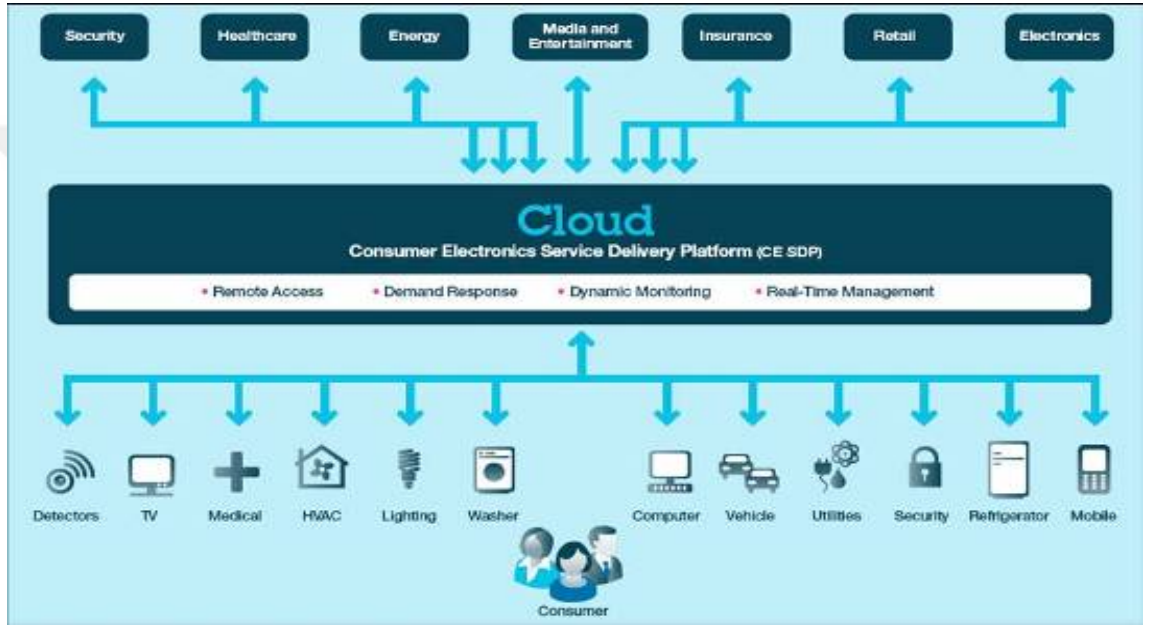
Servis sağlayıcı, silinecek veriyi çöp toplama olarak işaretler ve daha sonra silinmesi aylar alabilir. Bu tür sorunların üstesinden gelmek için, SNIA (Storage Network Industry Association) teknik grubu, verilerin doğrudan bir buluttan diğerine geçişini kolaylaştıracak bir uygulama geliştirmektedir. Artık bulutta verilerin güvenli bir şekilde silinmesini sağlayan şirketler de var, bu yüzden bir buluttan diğerine geçişte veriler tam anlamıyla kurtarılamayabilir. Böyle bir durumdan sakınmak için daha iyi seçenek, verilerin ilk önce müşteri konumunda depolandığı ve ardından bulut depolama sağlayıcısına kopyalandığı hibrit bulut kullanımınıdır.



ÜÇÜNCÜ BÖLÜM

NESNELERİN İNTERNETİ VE UÇTAN UCA BAĞLI EV MODELİ

Bağlı ev denildiğinde, nesnelerin internetini kullanan, büyük veriyi işleyen, bulut tabanlı ve akıllı evler aklımıza gelmektedir. IBM, akıllı ev tasarımına ilişkin Şekil 6'daki bağlantıları göstermektedir:



Şekil 6: IBM bağlı-akıllı ev modeli

<https://www.ibm.com/blogs/cloud-computing/2012/03/15/cloud-an-enabler-for-a-smarter-home/>

Şekil 6 incelendiğinde bağlı evin, güvenlik, sağlık, enerji tasarrufu, medya ve eğlence, sigorta, perakende ve elektronik ürünlere yönelik faydaları görülmektedir. Kullanıcının evinde kullanacağı tüm akıllı aletler bulut bilişim kanalıyla ortak ağa bağlanmakta ve kullanıcı tarafından mekân kısıtlaması olmadan kontrol edilebilmektedir.

3.1.Bağlı Ev Temel Görevleri

Akıllı evler, kullanıcılar tarafından manuel olarak gerçekleştirilmesi gereken görevleri ve işlevleri görünmez bir şekilde gerçekleştirirler. “Nesnelerin İnterneti” vizyonunun yaygın bir şekilde benimsenmesiyle, bağlı cihazların sayısı katlanarak artmaktadır. “Nesnelerin İnterneti” kavramının akıllı bir evinki ile iyi bir uyum içerisine girdiğine inanılır ev, cihaz çeşitliliği nedeniyle başlamak için en iyi yerdir. Hem PC'ler hem de diğer gömülü ve mobil cihazlar tarafından evde HTTP kullanımı giderek artmaktadır.

Tüm bunlar, her yerde bulunan bilgi işlem vizyonu ile uyumludur, ancak akıllı evler gerçekten yaygın hale gelmeden önce bir dizi önemli sorunun çözülmesi gerekmektedir. Mevcut ev düzeni, ev otomasyon sistemlerinin esnek olmayan ve kötü yönetilebilirliği gibi birçok problemle boğuşmaktadır. Ek problemler arasında yüksek enerji tüketimi ve ısı dağılımı, cihaz heterojenliği ve kötü yönetilebilirlik sayılabilir. Daha fazla cihaz eklemek sadece bu sorunları şiddetlendirecektir. Benzer şekilde, ev ağlarının kendi problemleri vardır. Temel hazırlık, karmaşıklık, sorun giderme, güvenlik ve kompozisyon sorunları, bu ağlarda uçtan uca ilkenin kullanımına atfedilebilir.

Ağ yöneticilerinin evdeki yokluğu da ağın yanlış yapılandırılmasına yol açabilir. Ayrıca, ev ağları da Spam, DoS trafiği ve / veya aldatmaca veya kimlik avı saldırılarına karşı hassastır. Bu sorunlara bir çok çözüm önerilmiştir ancak bunların her biri bazı durumlarda özel kaynak ve yoğun hesaplama gerektirir (Nabi ve Alvi, 2014).

Ev aletlerini buluta bağlamanın avantajları

Uygulama Geliştirme: Bulut üzerinde geliştirme, geliştirme ve prototip süresini kısaltacaktır. Ayrıca, ortak bir yürütme platformunun kullanılması ortak programlama uygulamalarına ve tasarım standardizasyonuna yol açacaktır (IBM, 2010).

Cihaz Heterojenitesi: Şu anda, evler uygulama geliştirme ve birlikte çalışabilirliği zorlaştıran çok sayıda farklı cihaz içerir. Tüm hesaplamaların bulut üzerinde gerçekleştirildiği ve yalnızca çıktının evde görüntülediği ince istemci

bilgisinin kullanımı, cihaz homojenliğine yol açacaktır (Dixon ve diğerleri, 2010; Brush ve diğerleri, 2011)

Esneklik ve Ölçeklenebilirlik: Entegre akıllı ev sistemleri esnek değildir, diğer yandan çoklu satıcılar birlikte çalışabilirlik sorunlarına yol açabilir. Böylece, yeni bir cihazın değiştirilmesi veya eklenmesi bir dizi sıkı tasarım seçeneği gerektirir. Evdeki aptal istemci tarafındaki cihazları kullanmak tüm sistemi esnek hale getirir. Ek olarak, yazılım yükseltmeleri, ince istemci tarafında güncellemeye gerek duymadan uygulamalara uygulanabilir. Ayrıca, bulut kaynaklarının ölçeklendirilmesiyle yeni uygulamaların gereksinimleri karşılanabilir (Brush ve diğerleri, 2011).

Enerji Verimliliği: 2005 yılında, hane halkları Avrupa Birliği'nde enerji tüketiminin %42'sini oluşturuyordu. Veri merkezleri, akıllı yük yönetimi, verimli mimari ve enerji farkında yönlendirme uygulamalarını etkin bir şekilde kullandığından elektrik tüketimini azaltabilirler. Hesaplamayı buluta taşıyarak, ısı merkezlerini harekete geçiren yüksek verimli su bazlı soğutma sistemleri gibi güncel soğutma tekniklerden yararlanılabilir (Gyarmati ve Trinh, 2010; Katz, 2009):

Fiyatlandırma: Kullanıcılar fatura fiyatlarını düşürmek için elektrik fiyatlarında hem zamansal hem de coğrafi dalgalanmalardan yararlanabilirler. Çalışmalar, bulutlara taşınan devlet kurumlarının yüzde 25 ila 50 arasında tasarruf sağlayabildiğini göstermiştir. Benzer şekilde, yoğun uygulamaları evden buluta taşımak da benzer kazanımlara yol açacaktır (Qureshi ve diğerleri, 2009; West, 2010).

Hareketlilik: Daha fazla kullanıcı bu modeli benimserken, standartlaştırılmış cihazlar her evin bir parçası olacaktır. Kullanıcılar bir evden başka bir eve taşınırlar bile kendi hesapları / profilleri bulutta saklanacağı için mevcut uygulamalarını kullanmaya başlayabilirler.

Ortak Kullanıcı Arayüzü: Evdeki mevcut kullanıcı arayüzleri ortalama kullanıcı için çok karmaşıktır. Kullanıcıların sistemle etkileşimde bulunabildikleri ve cihazlara girdiklerini verebildikleri ortak bir kullanıcı arayüzünün kullanımı çevreyi herkes için erişilebilir kılar. Buna ek olarak, mevcut masaüstü merkezli arayüzler, insanların merkezi sahne aldığı bulut destekli uygulamalarla uyumsuzdurlar. Hesaplama altyapısının (onu buluta taşıyarak) ve giriş sistemiyle ayrıştırılması, kullanıcı merkezli özelleştirmeye olanak tanıyan birden fazla kullanıcı arabiriminin yan yana olmasını sağlarlar (Calvert ve diğerleri, 2007; Pham ve diğerleri, 2011).

Ağ Yönetimi ve Güvenliği: Ortak bir işlem platformunun kullanılması, kullanıcıların uygun ağ yapılandırmasını sağlayabilen, ağ karmaşıklığını azaltan ve saldırılara ve istenmeyen postalara karşı güvenlik sağlayan uygulamalardan yararlanmasına olanak tanır. Bu uygulamaların sinerjisi, kullanıcıların ağlarından en iyi şekilde yararlanmasını sağlayacaktır (Calvert ve diğerleri, 2007).

Nesnelerin interneti destekli bağlı-akıllı evler

Radyo frekans etiketinde ev aletleri, güç tüketimini azaltan ekipman tasarımları arasındaki iletişimi azaltmak merkezi kontrolör konumu ve nesnenin çalışma durumu herhangi bir zamanda tespit edebilir. Radyo frekans etiketine sahip olan buzdolabındaki yiyecekler, konuşma, okuma ve yazma cihazlarının onlardaki bilgiyi okumaları sayesinde kendisi için uygun sıcaklık ve nemi ayarlayabilirler.

Enformasyon yönetim sistemleri yaşamın ve iş ortamlarının her alanında görülebilir, fakat bağlı-akıllı evlerdeki enformasyon yönetim sistemi yalnızca veriyi toplayıp depolamakla kalmaz, güç tüketimini azaltmak amacıyla elektrik ekipmanları ile elde edilen veriler arasındaki ilişkiyi yapay zekâ yardımıyla analiz ederler. Yapay zekâ akıllı veri analizini yapabilmek, modülleri kontrol edebilmek ve zeki elektrik kontrolünü sağlayabilmek için birleştirilmiş enformasyon platformu üzerinde iş görür. Yanı sıra, birleşik bilgi platformunun, çeşitli elektrikli ekipman türleri, bilgi toplama ekipmanı operasyonları ile iletişim kurmak için standart iletişim ara yüzü sağlaması gerekmektedir.

Artan ev aletleri ve elektrikli cihazların akıllıca geliştirilmesiyle, akıllı cihazların merkezi kontrol bilgi sistemi ile yakından iletişim kurması gerekmektedir. Bununla birlikte elektrikli ev aletlerinin haberleşme dünyasında birleştirilmiş standart bir iletişim protokolü ve genel zeki elektrikli ekipman ya da diğer cihazlarla iletişim kurma becerisi ya da merkezi enformasyon kontrol sistemi bulunmamaktadır. Bu durum da elektrikli cihazlar arasında kolaylıkla iletişim kurulmasına engel olmaktadır. Böylelikle ev aletleri arasında birleştirilmiş iletişim ara yüzünün formüle edilmesi ve iletişim protokolünün geliştirilmesi oldukça önem kazanmaktadır.

3.2. Baęlı Ev Kullanıcı ve Sistem Uygulamaları

Bu kısımda bulut tabanlı bir uygulamadan yararlanabilecek yeni uygulamalar ve literatürdeki mevcut uygulamalara yer verilmiştir. Uygulamalar, bir ev deposu aracılığıyla kullanıcılara sunulabilir. Dixon ve dięerleri (2010) bu uygulamaları iki geniş kategoriye ayırmaktadır: 1) Kullanıcı merkezli ve 2) Sistem seviyesi.

3.2.1. Kullanıcı Uygulamaları

3.2.1.1. Depolama

Ev kullanıcıları, müzik koleksiyonlarını saklamak için kompakt diskler ve özel depolama sürücülerini gibi çoklu ortamlara ihtiyaç duyarlar. Ayrıca en sevdikleri filmleri ve TV şovlarını saklamak için sofistike set üstü kutular ve medya merkezleri satın alırlar. Bu hem donanım açısından hem de son derece esnek olmayan bir şekilde pahalıdır, yani bir cihazın depolama kapasitesi üzerinde bir üst sınır vardır. Öte yandan, tüm bu medyanın bulutta depolanması hem maliyetli hem de ölçeklenebilir.

Uygulamalar, doğrudan favori medyayı buluttan aktarabilmektedirler. İşlem veritabanları, telefon defterleri ve büyük müzik katalogları gibi yapısal veriler gerektiren uygulamaları uygulamak için kullanılabilirler.

3.2.1.2. Oyun

Ye ve Huang'ın (2011) çalışmasına göre, ABD'deki hanelerin üçte ikisi bir oyun konsolu kullanmaktadır. Bunlardan evlerin%65'inde geniş bant bağlantısı vardır. Ayrıca, 2015 yılına kadar 260 milyon konsolun İnternet'e bağlanması beklenmektedir. Bu konsollar yüksek işlem hızı, bellek ve yerel depolama alanına sahiptirler. Örneğin, Xbox 360, 3.2 Ghz, 512 MB DRAM, 500 Mhz GPU ve 500 GB'a kadar isteğe baęlı depolamada çalışan 3 CPU çekirdeğine sahiptir. Bu konsolların bir bulut versiyonu, işlenmesini bulut üzerinde gerçekleştirecek ve çıkışı doğrudan televizyon setlerine aktarılacaktır. Bu bulut konsolu, aę üzerindeki girdisini Kinect denetleyicisinden alır. Çok oyunculu yetenek, buluttaki Sanal LAN'ların kullanılmasıyla etkinleştirilir. Böyle bir yaklaşım, bulut kaynaklarının yeni oyunların ortaya çıkmasıyla kolayca ölçeklendirilebileceği için, yeni yükseltilmiş konsolların

satın alınmasını gereksiz hale getirecektir. Bulut tabanlı bir oyun modelinin, Devasa Çok Oyunculu Çevrimiçi Oyunların (MMOGs) mimarisini büyük ölçüde basitleştirdiğini vurgulamak da önemlidir. Bu tür bulut tabanlı sistemler milyonlarca değere hizmet edebilir.

3.2.1.3 Otomatik Gözetleme ve Ses Tanıma

Gözetim sistemleri, akıllı bir evin en popüler bileşenlerinden biridir. Bir ev sahibi kapıda yüz tanıma sistemi kullanarak kapıyı açtığı anda otomatik olarak kapıyı açmak gibi yeni işlevler sağlamak için güvenlik sistemine entegre edilebilirler. Ayrıca, bu sistemler sesle aktifleştirilebilir ve farklı görevler yerine getirmek için doğal insan dilinin inceliklerini ayırt edebilir. Ne yazık ki, bilgisayar vizyonu ve doğal dil işleme algoritmaları yoğun olarak hesaplanmaktadır. Çözüm, konuşma tanıma ve bulut için 3D haritalama dahil olmak üzere CPU'nun yoğun görevlerini azaltan bulut özellikli robotların modelinde yatmaktadır. Böyle bir sistem, yüz tanıma ve diğer bilgisayar görme görevleri için bulutta dağıtılmış veri yoğun bilgi işlem sistemlerini de kullanabilir.

3.2.1.4 Sağlık İzleme

Kişiselleştirilmiş sağlık çözümleri gibi hizmetler. Bu tür hizmetler, her evin bir parçası haline getirilebilir ve bunların işlevleri, bulut kullanılarak geliştirilebilir. Yeni sağlık uygulamaları, evde bulunan çoklu sensörlerden alınan verileri kullanabilirler. Buluttaki uygulamalar, verileri sıra dışı kalıpları aramak için kullanabilirler. Bu uygulamalar ilk müdahale hizmetlerini (paramedik) uyararak için anormal bir şey olduğunda bir bayrak kaldırır. Veriler bulutta olduğundan ve üçüncü tarafların yalnızca uyarı aldığından gizlilik sağlanmış olur.

3.2.2. Sistem Uygulamaları

Bulutla bağlı evlerde kullanılacak sistem uygulamaları arasında güvenlik, yapılandırma, trafik izleme ve akıllı ağı gösterebilir.

3.2.2.1 Güvenlik

Kötü yönetim ve teknik know-how eksikliği, ev ağlarını çok çeşitli saldırılar için kolay hedefler haline getirir. Ayrıca, bu ağlardaki sistemler tehlikeye girebilir ve Spam ve zararlı trafiğin kaynağı olabilir. Bu sorunu çözmek için önerilen bir yaklaşım, kapalı bir geri besleme döngüsünde OpenFlow programlanabilir ağ anahtarlarının ve dağıtılmış çıkarım algoritmalarının kullanılmasını içerir. OpenFlow anahtarı, çıkarım motoruna giden trafiği iletmek ve yeni trafik filtreleme kurallarını uygulamak için kullanılır. Bu kurallar, çeşitli Spam filtreleme ve botnet / kötü amaçlı yazılım algılama algoritmalarını kullanan dağıtılmış çıkarım motoruyla hesaplanır. Bu üçüncü taraf çıkarım motoru - bilgi işlem yoğunluğu nedeniyle - bulutta, açık akışlı ağ geçidi anahtarını kontrol edebileceği yerlere yerleştirilir. Bu uygulamanın kullanıcıları, dağıtılmış çıkarımın bir parçası olmak için bölgesel hiyerarşik Sanal LAN'ları oluşturur. (McKeown ve diğerleri, 2009; Feamster, 2010).

3.2.2.2. Yapılandırma

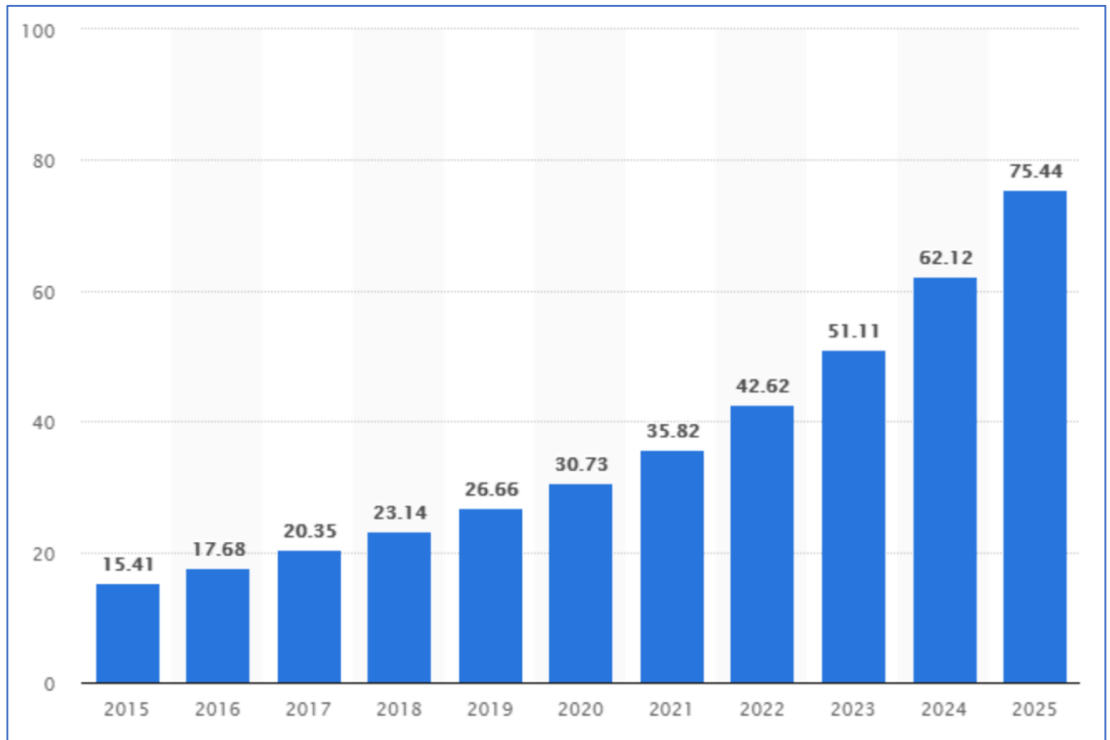
Ev ağı, içinde çok sayıda farklı uygulamayı ve cihazı doğru şekilde yapılandırabilen yerel bir yöneticiden yoksundur. Bu yanlış yapılandırma, ev kullanıcısı deneyimsizliği veya çeşitli bileşenlerin birbirleriyle etkileşimi nedeniyle ortaya çıkan bir sorundur. Bu yanlış yapılandırma, ev ağını kolayca aşağıya çekebilir. NetPrints, ağda en az bir kullanıcının doğru bir yapılandırmayla olması gerektiğinden yararlanır ve bu bilgileri yanlış davranan öğeleri doğru şekilde yapılandırmak için kullanır.

NetPrints, bu paylaşılan bilgiyi otomatik olarak indeksler ve alır. Kullanıcılar konfigürasyonları “iyi” veya “kötü” olarak etiketler. NetPrints, istemcinin ağdaki tüm öğeler üzerinde çalıştığı ve yapılandırma bilgilerini ve bir ağ trafiği izini topladığı bir istemci / sunucu modeline sahiptir. Bu bilgiyi hem çalışma hem de hatalı yapılandırma ayarlarının bir ağacını oluşturmak için ağaç tabanlı öğrenme ve bir mutasyon algoritması kullanan uzak bir sunucuya yükler ve istemcilere önerilen düzeltmeler verir. İstemcilerin sayısı arttıkça, sunucu bir performans darboğazına dönüşebilir (Aggarwal ve diğerleri, 2009).

3.2.2.3 Trafik İzleme

Meraki (2009) bulut yönetimli yönlendiriciler üzerinde şirketlerin popülerliği, merkezi olmayan yönetim ve kontrolün gücünün altı çizilmiştir. Bu yönlendiriciler uygulama başına QoS ve trafik analizi gibi özellikleri etkinleştirirler. Ancak bu yönlendiriciler, 5000'e kadar siteye sahip büyük işletmeler için tasarlanmış olup, bu sayede ev kullanıcıları için hem uygun hem de uygun değildir. İlginç bir şekilde, çalışmalar, kullanıcı başına kullanım bilgilerinin tüm ev sakinlerine sunulmasının, bant genişliğinin daha iyi yönetiminde yardımcı olabileceğini göstermiştir. Ayrıca, merkezi olmayan izleme hem kullanıcıların hem de uygulamaların bant genişliğini otomatik olarak sınırlandırabilir. Böyle bir sistem, istenmeyen içeriklerin bloke edilebildiği ebeveyn kontrolleri uygulamak için de kullanılabilir (Chetty ve diğerleri, 2010).

Trafik ve bant genişliği konusunda da çeşitli araştırmalar yapılarak önümüzdeki yıllarda oluşacak verilerin korkunç düzeyde artacağı ön görülmektedir. Geçmişten günümüze internet' e çıkan cihazların grafiğine bakıldığında eğimin sürekli ve katlanarak artan şekilde olduğu gözlemlenmektedir. Aşağıdaki şekle bakıldığında da statista.com' da yayınlanan rapora göre 2025 yılına kadar 75 Milyon cihazın internet üzerinden birbirinden haberleşmesi öngörülmektedir(Şekil 7).



Şekil 7: 2015-2015 yılları arasında bağlı cihaz sayıları(milyon)

(<https://justcreative.com/2018/11/19/internet-of-things-explained>)

3.2.2.4 Daha Akıllı Ağ

Önceki üç uygulamanın ele aldığı sorunlar, ev ortamı ve internet mimarisi arasındaki doğal uyumsuzluğa da atfedilebilir. Ev ağlarının uçtan uca ilkeye dayanması, önceki üç uygulamanın da ele aldığı gibi bir takım ana sorunlara yol açar. Calvert ve diğerleri (2007) ev ağını bir “kenar ağa” dönüştüren “akıllı bir orta” yaklaşım önermektedir. Bu yaklaşım, paket iletmeyi, cihaz izlemeyi ve aracılık etmeyi ve politika yönetimini kolaylaştırır. Bağlantılar, politika yönetimi ve cihazlar arasında arabuluculuk sağlayan kontrol ögesi olarak merkezi bir portaldan yararlanır. Portal, “ara bağlantı” adı verilen yönetilen bir anahtar aracılığıyla hem iç hem de ev içi iletişimi kontrol eder. Portal ayrıca, cihaz konum ve davranış DB’ sini de korur.

3.3. Bulut Tabanlı Otomasyon

Ev otomasyonunu buluta taşımanın ardındaki temel fikir, müşteriler için mümkün olduğunca fazla kolaylık sağlamaktır. İdeal bulut uygulamasında müşteri evine sadece bir ev otomasyon cihazı kurmalı ve belki de bunu bulut uygulamasına kaydettirmelidir. Başka adımlar, müşteri için bir sıkıntı olabilir. Ayrıca, yeni cihazın hangi üreticiyi ürettiği tamamen alakasız olmalıdır. Elbette böyle bir sistemin bulutu kullanabilmesi için yerine getirilmesi gereken bazı önkoşullar vardır. İletişim kurmak için bir Fieldbus kullanan sistemler için, veri yolu ile internet arasında bir ağ geçidine sahip olmak gerekir. İnternet katmanı protokolü olarak IPv4 kullanılıyorsa, bulut uygulamasının her bir cihazla kendi adresiyle iletişim kurması da mümkün değildir, bunun nedeni NAT'ın kullanılmasıdır veya her ev için olası düzinelerce yeterli IPv4 adresi olmamasıdır.

IPv6 böyle bir sistemi daha uygun hale getirir. Ev otomasyonu için bulut bilişimin kullanılmasının en büyük yararı, bir cihaz İnternet üzerinden iletişim kurulabildiği sürece, kullanıcının kullandığı özel (veya açık) protokolü önemsemesidir. Bulut uygulamasının katman yazılımı, cihazla veri alışverişi yapmalı

ve bilgiyi standart bir şekilde sunmalıdır. Bir bulut hizmeti için temel kullanım durumları şunları içerir:

- Bulutta veri kaydı ve işlenmesi.
- Servisler aracılığıyla cihazların izlenmesi ve kontrol edilmesi.
- Bulut uygulamasında tanımlanmış ve işlenen bazı kural kümelerine dayalı otomatik düzenleme.

3.4. Bağlı/Akıllı Ev Tasarımı Modeli

Bu kısımda alan yazında bağlı ev (nesnelerin internetini kullanan, büyük veriyi işleyen, bulut tabanlı ve akıllı) modelleri incelenerek araştırmamız için bir derleme yapılmıştır. Bu çalışma kapsamında önereceğimiz model akıllı ev için nesnelerin interneti tabanlı bütünsel çerçeve, bağlı/akıllı ev yönetim sistemi, birlikte çalışabilirlik ile güvenlik ve gizlilik bölümlerinden oluşmaktadır.

3.4.1. Akıllı Ev İçin IOT Tabanlı Bütünsel Çerçeve

Akıllı nesnelerin önümüzdeki birkaç yıl içinde piyasada baskın olması ve evlerde, her yerde bulunabilmesi, akıllı evler için yeni ve iyileştirilmiş hizmetlere duyulan ihtiyacı da beraberinde getirmesi beklenmektedir (Karnouskos, 2011). Bu nedenlerden dolayı, IOT tabanlı çözümlere duyulan ihtiyaç tartışılmaz olacaktır. En yeni yayınlar, daha geniş uygulama alanları yelpazesine uygun genel bir IOT çerçevesi geliştirmeye odaklanmaktadır. Lee ve Lee (2015), başarılı IOT çözümleri oluşturmak için beş IOT teknolojisini özdeşleştirirler: Radyo frekansı tanımlama, kablosuz sensör ağları, ara katman yazılımı, bulut bilişim ve uygulama geliştirme yazılımı. Ayrıca, kurumsal uygulamalar için üç IOT kategorisini de tanımlarlar: izleme ve kontrol, büyük veri ve iş analizi, bilgi paylaşımı ve işbirliği.

Da Xu ve diğerleri (2014) de Yakın Alan İletişimi, konum tabanlı hizmetler ve sosyal ağlar ile etkinleştirme teknolojilerinin listesi geliştirmişlerdir. Algılama, ağ oluşturma, servis ve ara yüzden oluşan dört katmanlı bir mimari önermektedirler. Bu öneride bulutun rolü eksik; o nedenle, hizmetlerin nasıl etkinleştirileceği açık değildir. Liu ve diğerleri (2014) adlandırma, adresleme, depolama ve arama servislerini destekleyen bir ara yazılım sunmaktadır. Buradaki fikir, mevcut sistemlerin en üstünde

bir ara katman yazılımı geliřtirmek ve böylece mevcut uygulamaların IOT ortamlarına daha kolay entegrasyonunu saęlamaktır.

Shrouf ve Miragliotta (2015) sanayideki üretim süreçlerinin IOT kullanarak izlenmesi konusunu incelemiřtir. Yazarlar, řirket ii veya bulut tabanlı veri madencilięi ve karar verme olanakları ile enerji yönetimine odaklanmış ayrıntılı bir çereve önermektedir. Üüncü taraf çözüm tasarımcılarının rolü çereve içinde belirtilmemiřtir.

Bu yayınlarla ilgili olarak, bu tezde sunulan çereve, literatürde bulunduęumuz en genel modelin (Da Xu ve dięerleri, 2014), ortadaki bulutla güçlendirilmiş ve belirli bir řeye uyarlanmış bir modifiye edilmiş hali olarak düşünülebilir. Nesnelerin İnterneti üzerine kurulu olan bu çok seviyeli hiyerarşik bütünsel çereve, literatürde tanımlanan akıllı evler için IOT çözümlerinin tüm temel özelliklerinin bir kapsayıcısı veya genellemesi olarak kullanılır.

3.4.1.1. Baęlı/Akıllı Ev

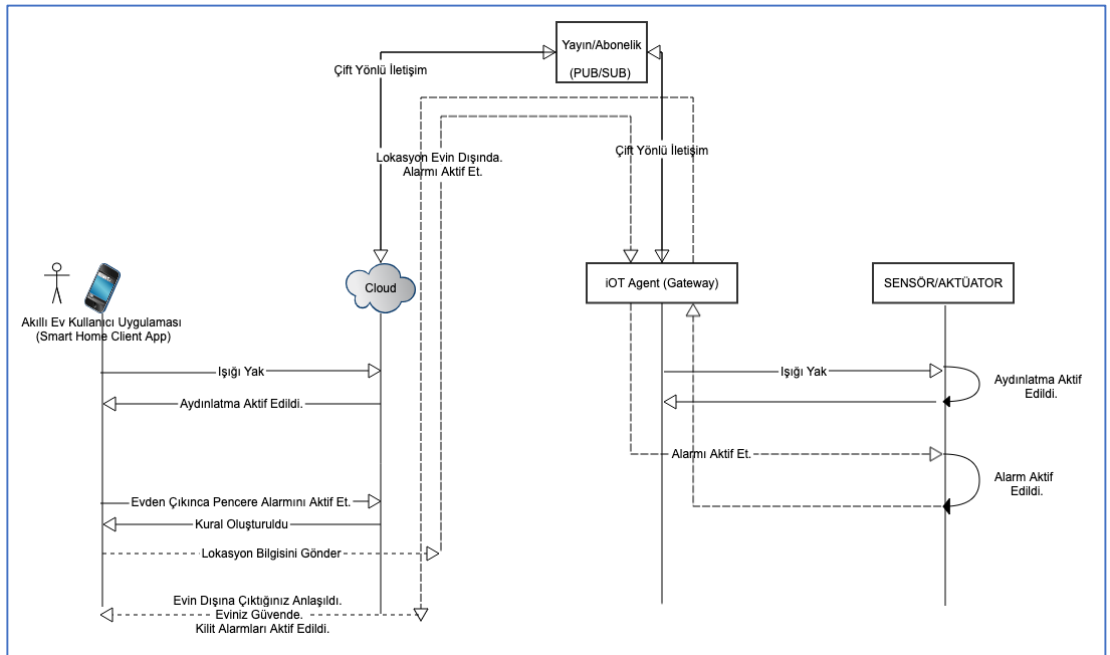
Kablosuz iletişim için arayüzlerle donatılmış tüm ev cihazları, evdeki WSN'yi oluşturur. Her evin bir WSN'si vardır ve her bir cihazdaki algılanan veriler, ev lavabosu ya da ev merkezi olarak adlandırdığımız bir merkez istasyona iletilir. WSN evindeki her düęüm, akıllı bir cihaz olarak kabul edilir ve orta düzeyde hesaplama ve iletişim yeteneklerine sahiptir. Ev merkezi, bazı veri depolama kapasitesine sahip olan, yerel işlemleri gerçekleřtirebilen ve ev WSN'si dışındaki cihazlarla iletişim kurabilen herhangi bir cihaz (akıllı saya, PC, tablet veya akıllı telefon) olabilir. Baęlı/Akıllı ev kompleksleri veya akıllı binalar söz konusu olduęunda, ana merkezin karřılıęı konut veya konut merkezi olarak tanımlanır. Yerleřim merkezinin, ana merkezle karřılařtırıldıęında, paylařımlı daęıtılmış üretim kaynaklarından veriyi yönetmekten sorumlu olduęu ek bir özellięe sahip olması gerekir. Yenilenebilir kaynaklar genellikle tüketiciler arasında paylařıldıęı için bu oldukça önemlidir. PV sisteminin binadaki tüm haneler tarafından kullanıldıęı çatıdaki bir PV sistemine sahip bir konut binası örnek verilebilir. Çerevede, daęıtılan her yenilenebilir enerji kaynaęı akıllı bir cihaz olarak kabul edilir.

3.4.1.2. Bulut

Farklı kaynaklardan elde edilen tüm veriler bulutta (ev sakinlerinin verileri, iletim / dağıtım hatlarından sensör ölçümleri veya üretim sahalarından vb.) toplanmaktadır. Bulut, büyük veri depolama ve işleme altyapısı sağlamaktadır. Çerçevenin en gelişmiş seviyesidir (Da Xu ve diğ., 2014; Lee ve Lee, 2015). Gubbi ve diğerleri (2013), yeni nesil IOT uygulamaları için bulutun *yüksek güvenilirlik, ölçeklenebilirlik ve özerklik* vaat ettiğini ifade etmektedirler. Bulut bu sistemin merkezi kısmıdır, dolayısıyla çerçevemiz “bulut merkezli” veya “bulut tabanlı” olarak kabul edilebilir.

Aşağıdaki şekil incelendiğinde örnek bir akıllı evin birbirine bağlanan parçalarının(things) mesajlaşma akışı görülmektedir. Buradaki mimariye göre kullanıcı cep telefonu uygulaması üzerinden evindeki lambaları açıp kapatabilir. Aç/Kapat komutları öncelikle mobil uygulamadan bulut’a oradan da PUB/SUB aracılığıyla ağ geçidine ve son olarak ta düğüm’e bağlı olan aktüatör’e iletilir(Şekil 8).

Diğer bir senaryoda ise kullanıcı elindeki mobil uygulama aracılığıyla lokasyon bilgilerini bulut’a paylaşarak bulut platformunda kendi lokasyon bilgileri üzerinden kural oluşturabilir. Kısaca örnek verecek olursak evin dışına çıktığında kapı alarmlarını aktif et yada tam tersi olarak eve gelindiğinde kapı alarmlarını pasif’e çek şeklinde kurallar tanımlayabilir.



Şekil 8: Akıllı bir evin kullanıcıdan bulut’a genel akış diyagramı

3.4.1.3. Yarar

Bu seviye, sadece akıllı ev ile kalmaktan ziyade akıllı ev dışında, kurulan akıllı sistemin geri kalan kısımlarına karşılık gelir: Üretim, iletim ve dağıtım. Her bölüm bağımsız olarak verileri doğrudan bulutlara gönderir. Yardımcı programla değiştirilebilecek tipik bilgiler: elektrik fiyatı, hava durumu tahmini, dağıtım/iletim hattı durumu, bir mikrogridin mevcut ve gelecekteki tüketimi, bir mikrogrid ile ilişkili dağıtılmış üretim kaynaklarının mevcut ve gelecekteki üretimi gibi (Sajjad ve arkadaşları, 2014).

3.4.1.4. Üçüncü Taraf Uygulamaları

Üçüncü taraf uygulamaları bulut verileri kullanılarak geliştirilmektedir. Bulut platformundan beslenen diğer uygulamalar, iş uygulamaları, endüstri odaklı uygulamalar kullanıcıya özel birer IOT uygulamasıdır. Yani, üçüncü taraf uygulama geliştiricileri buluttan (özel veya kamusal) veri alır ve bu verileri web tabanlı veya mobil uygulamalar biçiminde çözümler sunmak için kullanırlar (Fan et al., 2010).

3.4.1.5. Kullanıcı Arayüzü

Kullanıcı arayüzü, son kullanıcılara (bildirimler, öneriler, akıllı cihaz kontrolleri, vb.) veri sağlayan kullanıcı arayüzlerini temsil eder (Da Xu ve diğerleri, 2014). Aylık (hatta günlük) hane halkı tüketimine işaret eden ham tablo verilerinin, kullanıcılar tarafından yorumlanması zordur. Sadece genel hane halkı tüketimini değil, aynı zamanda cihaz / cihaz seviyesindeki tüketimi de göstermek için daha sofistike bir görselleştirme aracı gereklidir (Liu ve diğerleri, 2014). Tüketiciler evlerinde farklı cihazlar, özellikle de esnek olmayan cihazlar gibi otomatik olarak kontrol edilemeyen ürünler hakkında daha fazla bilgi edinebilecekleri için, bu sayede tüketicilerin onları tüketen doğasını göz önünde bulundurarak sezgisel olarak kontrol edebilmeleri mümkün olacaktır. Üçüncü taraf uygulamaları, tüketiciler için sezgisel görsel kullanıcı arayüzleri geliştirmek için çaba sarfetmeli ve Kalite Deneyimi (QoE) metrikleri kullananları sıklıkla değerlendirmelidir.

3.4.2. Bağlı/Akıllı Ev Yönetim Sistemi

Bir enerji yönetim sistemi, bir kamu hizmeti şirketi ve güç tüketen akıllı cihazlar arasında bir arayüz olarak tanımlanır. Tüketicilere karşı biraz önyargılı olan

iki tarafa (hizmet ve tüketiciler) fayda sağlamayı amaçlamaktadır. Yaygın olarak kullanılan bir başka terim ise Talep Tarafı Yönetimi'dir (DSM). Talep Tarafı Yönetimi gelecekteki enerji sistemlerinde güç dengelemesi yapmak için tüketici düzeyinde tüketim / üretimin izlenmesini ve kontrol edilmesini sağlayan bir dizi teknolojiyi temsil etmektedir (Atzeni ve diğerleri, 2013; Siano, 2014; Rezvani ve diğerleri, 2015).

Akıllı ev için IOT çözümleri bağlamında, geleneksel DSM modeli bulut merkezli modele doğru kaydırılmıştır. Bulut tabanlı yaklaşım, çok büyük bir parametre kümesini dikkate alan merkezi optimizasyon sunar; dolayısıyla, enerji yönetiminin geleneksel bir yaklaşımla karşılaştırıldığında daha iyi performans göstermesi beklenmektedir.

3.4.2.1. Akıllı Nesnelere/Akıllı Cihazlar

Akıllı şebeke sisteminde üretim veya iletim hatlarına bağlı olan ev aletleri, ışıklar yada daha genel bir ifadeyle aktüatör veya sensörler akıllı nesnelere olarak düşünülebilir. Bu nesnelere, verileri algılayabilir, çalıştırabilir, işleyebilir ve iletişim kurabilirler. Ayrıca algılamak ve harekete geçirmek için A/D ve D/A dönüşümlerini gerçekleştirmeleri gerekir (Byun ve arkadaşları, 2012). Bu cihazlar periyodik olarak algılama yapar ve merkeze (kablolu olarak bağlı) algılanan verileri gönderir. Dahası, protokoller izin veriyorsa, algılanan veriler doğrudan buluta gönderilebilir. Mümkünse, akıllı cihazlar algılanan verileri göndermeden önce temel veri işlemlerini gerçekleştirmelidir (Stojkoska ve arkadaşları, 2012; Viani ve arkadaşları, 2013).

Aktüatörler ayrıca uzaktan kontrol edilebilirler. DSM kapsamında, ev aletleri üç kategoriye ayrılabilirler: esnek olmayan, esnek ve çift doğallı cihazlar (Erol-Kantarci ve Mouftah, 2010). Esnek olmayan cihazlar, temel yükler veya önleyici olmayan görevler (ışık, TV, PC, saç kurutma makinesi gibi) ile bağlantılı olan ve sistem tarafından kontrol edilemeyen cihazlardır (Ullah ve diğerleri, 2013). Esnek cihazlar, düzenli yüklerle veya önleyici görevlerle (ısıtma veya klima gibi) ilişkilendirilir ve sistem tarafından otomatik olarak çalıştırılabilir. Çift doğallı cihazlar bazen esnek, bazen de esnek olmayan (çamaşır makinesi, bulaşık makinesi veya çamaşır gibi) olarak hareket edebilir. Örneğin, bazen tüketici önceden belirlenmiş bir zaman dilimi içinde olduğu sürece bulaşık makinesinin çalışacağı kesin zamanı umursamamaktadır. Bu cihazlar genellikle başlangıç gücü verirler (Khan ve arkadaşları, 2014; Ullah ve arkadaşları, 2013). Akıllı cihazlar (hem esnek hem de çift

doğa), güç tüketimini ölçebilen ve operasyonlarını gerçek zamanlı olarak kontrol edebilen akıllı güç çıkışları ile donatılmıştır.

3.4.2.2. Merkezler (Hubs)

Merkez (Hub), akıllı cihazlardan ham ve / veya işlenmiş verilerin toplanmasından ve bulutlara iletilmesinden sorumlu olan bir cihazdır (Zhu ve diğerleri, 2010). Mümkün olduğunda, merkez, veri akışını buluta doğru azaltmak için yerel veri işlemlerini gerçekleştirmelidir. Viani ve diğerleri, 2013 ve akıllı bir ev senaryosunda, merkez yerel bir zamanlayıcı, regülatör veya yük dengeleyici olarak görev yapan akıllı cihazlara komutlar gönderebilir (Byun ve diğerleri, 2012). Bir konut merkezi söz konusu olduğunda, elektrik akışını nano-benzodan düzenleyen cihazlara komutlar gönderebilir(yani şebekeden elektriğin satın alınması / satılması işlemlerini yönetmek için). Merkez, akıllı cihazların kullandığı iletişim protokollerini anlar. Cihazlar genellikle birbirleriyle iletişim kuramadıklarından, akıllı nesnelere arasındaki birlikte çalışabilirliği sağlamak için merkezlere ihtiyaç vardır (Gubbi ve diğerleri, 2013; Heile, 2010). Dolayısıyla, bazen bir evin birden fazla göbeğe ihtiyacı vardır. Gelecekte, akıllı cihazlar arasında tam birlikte çalışabilirlik sağlandığında, merkezlerin modelde gereksiz olması beklenir.

3.4.2.3. Bulut

Bulut, ev yönetiminin en karmaşık parçasıdır. Bulutun ana görevi verileri saklamaktır (Zhou ve diğerleri, 2013). Yüksek veri hacmi nedeniyle, yeni yaklaşımları karşılamak için geleneksel yaklaşımlar değiştirilmelidir. Makine öğrenme teknikleri, zaman serileri işleme ve ileri analitik temelli yeni yöntemler ve algoritmalar uygulanır (Da Xu ve diğerleri, 2014; Gubbi ve diğerleri, 2013).

Üçüncü taraf uygulamalar tipik olarak, kullanılan verinin değişmeden değiştiğini, yani verilerin gerçek zamanlı olmayan bir şekilde uygulamalar tarafından kullanılabilmesini varsayarlar. Bulutta, olaya dayalı veriler sorguya dayalı işleme dönüştürülür. Bu, gerçek zamanlı IOT ağ iletişimi ve üçüncü taraf uygulama dünyasının farklılıklarını köprülemek için çok önemli bir adımdır. Veriler sürekli olarak depolanmalı ve birden fazla seviyede soyutlanmalı, böylece daha önce depolanmış verilerle kolayca birleştirilebilir, yeniden oluşturulabilir ve / veya toplanabilir, ki bunlar da IOT olmayan kaynaklardan gelen bazı veriler olabilir (Da Xu

ve diğerleri, 2014). Daha da önemlisi, farklı soyutlama seviyeleri uygulamaların gerektirdiği şekilde sunulacağı için uygulama erişimini ve kullanımını basitleştirecektir.

3.4.2.4. Üçüncü Taraf

Üçüncü taraf, son kullanıcılar için programlayıcılar, düzenleyiciler ve yük dengeleyiciler şeklinde uygulamalar geliştirmelidir (Fan ve diğerleri, 2010). Bir zamanlayıcı, ikili doğramanın aktif olacağı zaman aralıklarını tanımlamaktan sorumlu bir araçtır. Bir regülatör, esnek cihazların yönetiminden sorumlu bir araçtır, yani klimaların, ısıtıcıların, nem gidericilerin, vb. çalışmalarını düzenler. Yük dengeleme, piyasadaki mevcut elektrik fiyatını ve üretilen elektrik enerjisini dikkate alarak enerji tüketimini optimize etmelidir ve varsa yerel yenilenebilir kaynakları araştırır. Tüm bu araçlar, ev cihazları tarafından elde edilenlerden çok daha fazla parametre kullanan gelişmiş algoritmalara ihtiyaç duyarlar. Tüketici ve hane halkı profilleri oluşturmak için buluttaki mevcut verilerden madencilik ve bilgi çıkarımının karmaşık görevleri yerine getirilmelidir veya daha basit kelimelerle mevcut akıllı ev verileri tüm kullanıcılar için kişiselleştirilmiş tavsiyelerin oluşturulmasına yol açmalıdır (Liu ve diğerleri, 2014).

3.4.3. Birlikte Çalışabilirlik

Halen, genel bir akıllı ev çözümünün geliştirilmesindeki ana konu, akıllı ev aletlerinin entegrasyonu ile ilgili maliyettir (Ko ve diğerleri, 2011). Birlikte çalışabilirlik, IOT'deki rekabetçi çözümlere piyasa açmak için anahtardır (Lu ve diğerleri, 2011; Misra ve diğerleri, 2015). Dünyada akıllı cihazlar üreten lider şirketler, mevcut internet ile kolay entegrasyon sağlayacak olan tam birlikte çalışabilirliğe ulaşmak için çalışmaktadırlar.

Z-Wave ürünleri ZigBee'nin önceki sürümleri ile zaten birlikte çalışabilirken, ZigBee 3.0 ile ZigBee Alliance bu özelliğin 2015 yılı sonuna kadar uygulanacağını açıklamıştır. ZigBee, farklı satıcılar için birlikte çalışabilen ürün özellikleri tanımlamak amacıyla birçok komite kurdu. Ev Otomasyonu, Sağlık Bakımı, Uzaktan Kumanda vb. gibi farklı genel uygulama profilleri için aygıtlar. Yine de ürünler bu profiller arasında ve bir profil içindeki revizyonlar arasında mutlaka birlikte çalışabilir

değildirler. Diğer tarafta, X10 ve Insteon birbirleriyle tamamen birlikte çalışabilirler (Darbee, 2013).

3.4.4. Güvenlik ve Gizlilik

Akıllı şebeke gelişiminin karşı karşıya olduğu ortaya çıkan gereksinimlerin en önemli konularından biri hem kablosuz hem de sistemlerin kablolu kısımları için siber güvenlik ile ilgilidir (Ning ve diğerleri, 2013; Schneps-Schneppe ve arkadaşları, 2012). Akıllı şebeke, sistem tasarımcıları için kritik bir endişe olarak ortaya çıkan siber teröristler için bir hedef olabilir. Verilerin iletilme şekli nedeniyle, IOT, kablosuz ağların ortak saldırılarının çoğuna doğal olarak savunmasızdır.

Dolayısıyla, IOT bir güvenlik politikası gerektirir, ancak bunu sağlamanın maliyetinin mümkün olduğunca düşük olması gerekir. Güvenlik hafif kriptoprimitifler sağlayan farklı yaklaşımlar, gerçekliği (aygıt zararlı bir nesne değildir) sağlamak için araştırılmalıdır (Altolini ve diğerleri, 2013). Örnek olarak, bütünlük (iletilecek veriler alınan verilerle aynıdır) ve gizlilik (verilerin diğerleri tarafından okunamaz hale getirilmesi) sağlanabilir (Dimitrievski ve arkadaşları, 2006).

3.5. Bağlı/Akıllı Ev Tasarımı Modeli İle İlgili Bir Senaryo

Akıllı evde bulunan cihazlar, sensörler (iç ve dış), mikrodalga fırın gibi elektronik ekipmanlar, buzdolabı, televizyon, PC, cep telefonları ve kontrol cihazları gibi hesaplama cihazlarıdır (kapılar için pencereler). Sabah erkenden bir kişi, bulut sunucusu tarafından belirlenen uygun zamanda fon müziği çalmaya başlamasıyla uyanabilir.

Aynı zamanda yatak odası pencerelerinin perdeleri yavaşça güneş ışığının girmesine izin verir. Ev sakini kahvaltı hazırlarken, mutfakta TV, programın günlüğünü, hatırlatıcılarını ve notlarını otomatik olarak gösterilir. Mevcut gün programına bağlı olarak, görev gerçekleştirilmede herhangi bir gecikme olduğunda, zamanında sesli uyarılar ayarlanır. Birey evden ayrılmak üzereyken, günün hava durumu tahmin edilir.

Bireyler iş için izinli olduğunda ve evde kimse yoksa, AC kapatılır ve kapılar ve pencereler kapalı kilitleme sistemi etkinleştirilir. Herhangi bir zamanda, ev sistemi,

sistemin işleyişine bağlı olarak, günlük rutin yemek ürünlerini ve çamaşırları yıkamak için gözetim moduna girer. Buzdolabındaki sensörler mevcut parçaların miktarını kontrol ederek kullanıcıya durum bilgisi sağlar. Bulut, arkadaşlarla parti gibi özel bir etkinlik olduğunu belirlerse, parti için planlanan menüye, mevcut öğeler için bulut kontrole ve mevcutların miktarını hesaplamayı başlar. Daha sonra, mevcut olmayan veya daha az miktardaki ürünler için sipariş verilecektir. Bir kişi, kirlendikten sonra çamaşırlarını çamaşır makinesine yerleştirecektir, daha sonra çamaşır makinesindeki sensörler, mevcut kıyafetlerin miktarı hakkındaki verileri bulut sunucusuna yönlendirecektir ve daha sonra kullanılmak üzere tersi yönde veri alımı gerçekleşecektir. Bulut sunucusu, enerji tüketimi, zaman gibi çeşitli faktörleri göz önünde bulundurarak giysilerin yıkanmasını yönetecektir.

Akşam saatlerinde dış ortamdaki hava sıcaklığı soğur ve sıcaklık sensörleri ve ışık sensörlerinden gelen veriler kullanılarak bulut sunucusu ilgili aracı bitkilerin ve bahçelerin sulanması için yönlendirir. Kişi evinde bulunuyorsa bulut sıcaklık sensörleri, ışık sensörleri ve ses sensöründen gelen verileri kontrol ederek ona herhangi bir hoş akşam yaşatır ve eğer tüm koşullar bir kişi için elverişliyse, pencereleri açarak soğuk havanın girmesini sağlar.

Kişi seyahat ediyorsa (arabada bulunan etiketin aldığı tanımlayıcı tarafından bilinir), bulut herhangi bir yerde herhangi bir buluşma veya randevu varsa, bireyin zamanlamasını kontrol ederek bulutun bireyi yolu üzerindeki trafik durumu konusunda onu bilgilendirir. Kişi evden ofise veya başka bir yere geri geliyorsa, bulut tahmini varış saatini hesaplar. Ayrıca, sıcaklık sensörlerinden dışarıdaki sıcaklığı da kontrol edecek ve bireyin gelişine bağlı olarak AC'yi açık duruma getirecektir.

Akşam olunca birey işten gelir; evin bina giriş kapısına girerken, orada bulunan etiket, mobil okuyucu tarafından okunmakta ve buluta iletilmektedir. Bulut sunucusu daha sonra bir kişinin varlığını doğrulayarak ilgili sunucusunu yönlendirir. İlgili araç, daha sonra yönlendirildiği gibi uygun sensörlerini başlatır. Birey girerken, oturma odası ışıkları yavaşça açılır ve onun için rahat olan oda sıcaklığını bulur. Yavaş müzik, sadece stresi azaltmak için bir süre için arka planda başlar. Aynı zamanda, bulut sunucusu son 24 saatte enerji tüketimini bildiren bir rapor oluşturacaktır. Raporla ayrıca diyet ve egzersiz için bazı öneriler (giyilebilir sensörlerden alınan veriler kullanılarak) yer alacaktır.

Ev sakininin akşam yemeđini hazırladıđı zamanda mzık otomatik olarak arka planda bařlar. Kiři akřam yemeđi yemek iin oturduđunda, oturma odasında TV, en son programa son noktasını koyduđu yerden otomatik olarak bařlar. Kiři uyumak iin yatak odasına getiđinde loř ıřıklar ile elveriřli bir ortam bulabilir. Oturma odası ve televizyondaki ıřıklar kapanır. Belirtilen sre sonunda ve evde hibir hareket algılanmadıktan sonra, tm cihazlar uyku moduna geer.



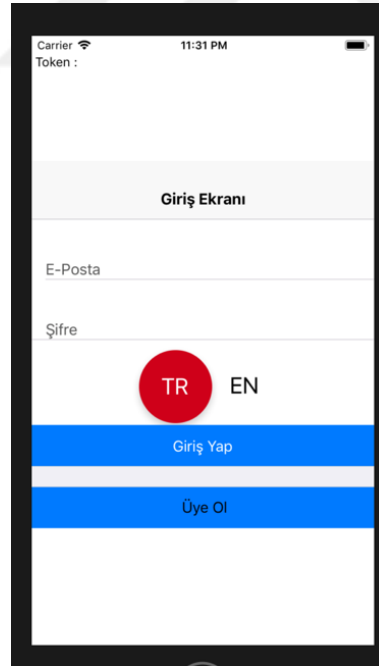
DÖRDÜNCÜ BÖLÜM

MOBİL UYGULAMA VE BULGULAR

4.1.Senaryo Ve Gerçekleşme Adımları

4.1.1. Mobil Uygulamanın Tanıtımı

Bu bölümde ekran görüntüleriyle birlikte akıllı ev sistemi için geliştirilen mobil uygulamanın tanıtımı yapılacaktır. Mobil uygulama akıllı ev sisteminin kurulumu ve kullanımı için geliştirilmiştir. RN teknolojisi kullanılarak yaklaşık %90 oranda aynı kodun hem iOS hem de Android platformu üzerinde çalışması sağlanmıştır. Uygulamanın kullanılabilmesi için sisteme üye olunması gerekmektedir. E-Posta adresi yardımıyla bu sisteme üye olunabilir. Uygulama’da çoklu dil desteği olmakla birlikte İngilizce ve Türkçe olarak dil dosyaları hazırlanmıştır. Çoklu dil desteği ile ilgili açıklamalar EK C’de yapılmıştır. Sisteme e-posta adresiniz ve şifrenizle giriş yaparken dil seçimi de yapılabilmektedir(Şekil 9).



Şekil 9: Akıllı ev mobil uygulaması - giriş ekranı

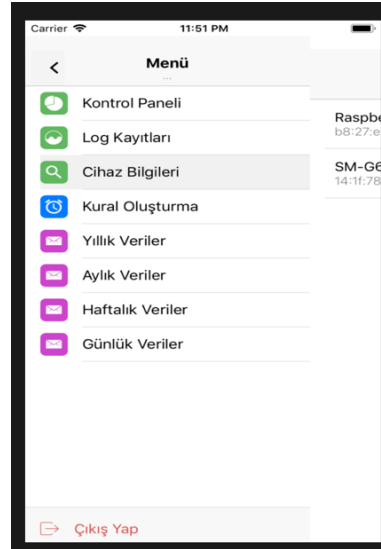
Sisteme giriş yapıldıktan sonra uygulamanın kullanım amacı belirlenir. “Kullanım” ile sensör verileri ve kural ekranları gibi akıllı ev kullanımına uygun modüller listelenir ve erişime açılır. “Kurulum” seçilmesi durumunda ise akıllı bir evin mobil

uygulama aracılığıyla node, sensör ve aktüator ayarları yapılarak bu şeylerden(things) bilgi alınıp verilmesi sağlanır(Şekil 10).



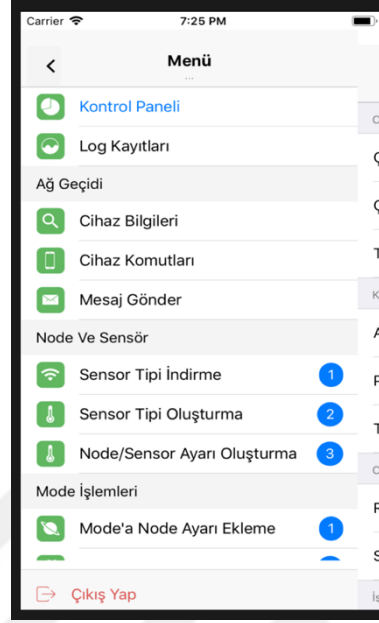
Şekil 10: Akıllı ev mobil uygulaması - kullanım amacının belirlenmesi

Sisteme girerken “Kullanım” amacı seçildikten sonra aşağıdaki gibi modüller listelenecektir.



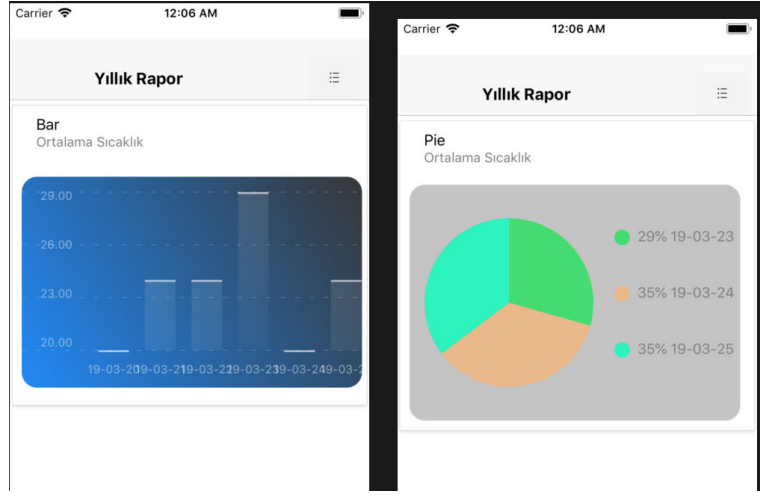
Şekil 11: Akıllı ev mobil uygulaması - kullanım senaryosuna göre modüller

Sisteme girerken “Kurulum” amacı seçildikten sonra ise aşağıdaki modüller listelenecektir(Şekil 11-12).



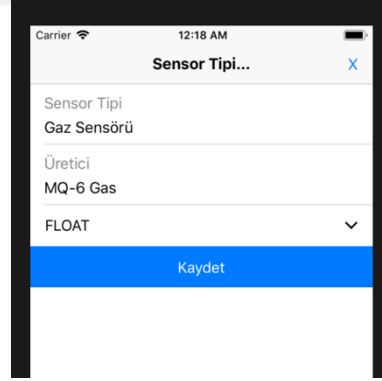
Şekil 12: Akıllı ev mobil uygulaması - kurulum senaryosuna göre modüller

Sensörlerden alınan veriler önce ağ geçidine(gateway) daha sonra da bulut'a gönderilir. Bu işlem için mobil uygulamanın ayakta olmasına gerek yoktur. Alınan bilgiler bağımsız olarak bulut platformunda ilişkisel ve ilişkisel olmayan veritabanlarında depolanır. Daha sonra bu bilgiler REST API'lar aracılığıyla raporlama yapılabilme için tekrar son kullanıcıya sunulur. Sensör verileriyle oluşan uygulama rapor ekranı aşağıdaki gibi olacaktır. Grafikler günlük, haftalık, aylık ve yıllık olarak görüntülenebilir. Ayrıca görüntüleme yapılırken pasta(pie), boru(bar) ve hat(line) şeklinde görseller şekillenebilir(Şekil 13).



Şekil 13: Akıllı ev mobil uygulaması - günlere göre ortalama sıcaklık

Kurulum ve kullanımı kolay mobil uygulama sayesinde kullanıcılar yeni sensör satın aldıklarında bu sensörü kolayca sisteme ekleyebileceklerdir. Mobil uygulama sayesinde ayar profili oluşturularak ağ geçidine(gateway) gönderilen veri tipi ile takılan sensör verileri okunup bulut platforma aktarılır(Şekil 14).



Şekil 14: Akıllı ev mobil uygulaması - örnek sensör ekleme ekranı

4.1.2. Örnek Kullanım Senaryosu

Bu bölümde hem geliştirilen mobil uygulamanın daha iyi anlaşılabilmesi hem de genel mimarinin daha açıklayıcı olması için örnek bir senaryo açıklanacaktır. Bu normal bir evin akıllanması senaryosudur. Bu senaryoda evimizde kapı alarmlarının aktif ve pasif edilmesi işlemi yapılacaktır. Mobil uygulama aracılığıyla alınan konum bilgileri sayesinde geliştirilen mobil uygulamadan “kural oluşturma” seçeneğinden faydalanılarak

ev sahibinin eve gelmesi durumunda her kapı kilidinin tek tek elle pasife çekilmesi yerine otomatik olarak pasif edilmesi gerçekleştirilecektir. Özetle açıklamak gerekirse ev kullanıcısı evine yaklaşınca kapı ve pencerelerde güvenlik amacıyla taktığımız alarm sistemlerinin otomatik olarak pasif olacak yine kullanıcı evinden uzaklaşırsa tekrar bu alarmlar aktif olacaktır.

4.1.3. Kurulum ve Kullanım Adımları

Bu bölümde akıllı bir evin kurulumunu yada normal bir evin akıllandırmak için atılması gereken adımlar açıklanacaktır. İlk olarak ihtiyaca göre sensör ve aktüatör ihtiyaçları belirlenmelidir. Bu sensör ve aktüatörlerin bağlanacağı node'ların da yine temin edilmesi gerekir. Son olarak nodelar'ı da bulut platformuyla haberleştirecek gateway(cihaz) teminiyle sistemin temel gereksinimleri belirlenir. Bu aşamadan sonra geliştirilen iOS yada Android uygulama cep telefonuna indirilerek IGNITE'a üye olunur. Üye bilgileri ile sisteme giriş yaparak uygulama kullanım amacını "Kurulum" olarak seçip envanterimizdeki sensör tipleri bulut'a tanıtılır. Daha sonra bu sensörlerin ne sıklıkla veri okunacağını ve verilerin hangi değeri aşması durumunda bulut'a gönderileceğini belirlenir. Bu konfigürasyonu yapmamızdaki amaç işe yaramayan verilerin daha node aşamasında elenerek bulut platformunun gereksiz verilerle şişmesinin önüne geçmektir. Son olarak eklenen sensör konfigürasyonunu seçilen ağgeçidi(gateway) altında yer alan düğüme(node) gönderilir.

Eklenen sensör veya aktüatör varsayılan olarak IGNITE kütüphanesinde tanımlı ise bilgiler bulut'a otomatik olarak akacaktır. Ancak eğer kütüphane içinde olmayan bir sensör ekleniyorsa ajan(agent) tarafında da kodsız küçük bir geliştirme yapmak gerekir. Yine burada da bulutta yapılan tanımlamanın benzeri yaparak sensör tipi ve veri tipini belirterek tanımlama yapılır. Mobil uygulama aracılığıyla bulutta tanımlanan veri tipiyle ajan tarafında tanımlanan bilgilerin uyuşması rapor ver alarm durumlarının işletilebilmesi için tek kriterdir.

Üstteki tanımlamalar sonrası sensörlerden alınan veriler kablosuz ağ(Wi-Fi) aracılığıyla ağgeçidinde(gateway) aktarılır. Ağ geçidi de aldığı bilgileri bulut'a iletir. Ağ geçidi Android işletim sistemine sahip bir cep telefonu olabileceği gibi RaspberryPi

cihazı da olabilir. Bu örnek çalışmada hem Samsung mobil cihaz hem de RaspberryPi3 üzerinde Android işletim sistemi kurularak bilgi akışı sağlanmıştır.

4.1.4. Verilerden Rapor ve Grafik Oluşturma

Çalışmanın üstteki bölümlerinde akıllı bir evin çeşitli avantajlarından bahsedilmiştir. Bu avantajların gerçekleşebilmesi ve konforun sağlanabilmesi için en temel kriterin bilgilerin toplanmasıdır. Bilgilerin toplanması bu çalışmanın ilk adımındır. Ancak bilgilerin toplanmasıyla birlikte doğru ve güvenli bir şekilde cihazdan bağımsız olarak bir bulut platformunda tutulması gerekmektedir. Bu çalışma kapsamında yapılan örnek uygulamada sensörden verilerin alınıp node'a oradan da ağ geçidine(gateway) ve son olarak ta ağgeçidinden bulut platformuna aktarılması işlenmiştir. Bulut platformu olarak Ardıç firması tarafından geliştirilen ve ücretsiz olarak sınırlı bir şekilde erişime açılan IGNITE kullanılmıştır. (<https://www.iot-ignite.com>) Kullanılan ajanlar(agent) sayesinde N çeşit sensörden veriler alınabilir. Hatta IGNITE üzerinden ve ajanlar üzerinden eşleşecek şekilde tanımlanarak tüm sensörlerle çalışılabilmiştir.

Bilgilerin sensörlerden alınması tarafındaki iki temel kural bulunmaktadır. Sensörden okunan veri belirlenen limit değerinin üzerindeyse ve belirtilen eşik değerinden fazla ise bulut platformuna gönder şeklinde süzme işlemi yapılmaktadır.

Buradaki filtreleme tanımı dışında okunan sensör verilerinin buluta gönderilmesi işleminde bir kısıtlama yoktur. Ancak ilerleyen zamanlarda akıllı bir sensör okuma kütüphanesi geliştirilerek daha anlamlı ve sadece gerçekten ihtiyaç duyulan sensör verilerinin bulut platformuna aktarılması sağlanabilir. Bu sayede bulutun daha efektif kullanımı sağlanarak gereksiz verilerle şişmesinin de önüne geçilmiş olur.

Filtrelenerek bulut platformuna gelen veriler REST API'lar aracılığıyla son raporlama ve analiz için erişime açıktır. Çalışma kapsamında geliştirmesi yapılan mobil uygulamada üstte bahsedilen API'lardan gelen veriler gruplanarak ve sıralanarak günlere göre, haftalara göre, aylara göre ve yıllara göre veriler listelenir. Son olarak ta sıralanan bu verilerden grafik raporları oluşturulur.

4.1.5. Grafik Verileri Üzerinden Kural Tanımlama

Grafik verileri üzerinden kural tanımlama işlemi gerçekleştirilebilir. Eğer grafik olacak şekilde anlamlı veriler varsa bu veriler üzerinde alarm ve kural tanımlaması yapılabilir. Veriler işlenmeden büyük veri olarak ilişkisel olmayan DB' de saklanması durumunda bu raporların oluşturulması çok kolay değildir. Ancak IGNITE hem ilişkisel hem de ilişkisel olmayan DB' de değerleri tutar. Bu sayede ilişkisel DB' deki değerler üzerinden çeşitli kurallar ve alarmlar tanımlamak ta mümkündür. Ayrıca yine bu alt yapıyı kullanarak diğer bir özellik olan ağ geçidi(gateway) üzerinden alarm tanımlaması da yapılabilmektedir. Fakat geliştirilen bu uygulamada ağ geçidi seviyesinde alarm tanımlama özelliği kullanılmamıştır.

4.2.Kurallar Ve Alarmlar

4.2.1. Kural/Alarm Oluşturma

Geliştirilen uygulamanın bir diğer özelliği de kural ve alarm oluşturabilmesidir. Bu uygulama sayesinde tanımlaması yapılan kurallar hem mobil cihazdaki DB' de tutulur hem de bulut platformuna iletilir. Bulut platformunun beklediği formatta oluşturulan kurallar anlık olarak işletilerek senaryonun gerçekleşmesi durumunda çeşitli aksiyonlar alınabilir.

4.2.2. Kuralların/Alarmların Faydaları

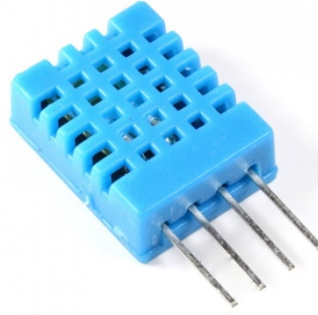
Çalışması yapılan mobil uygulama sayesinde alarm ve kurallar oluşturularak hayatı daha güvenli ve konforlu hale getirebilir. Örneğin evden çıktıktan sonra acaba gaz açık veya musluk açık kaldı mı ? Gibi sorular bizi meşgul etmeden alarmlar vasıtasıyla kontrol altına alınmış olur. Bu senaryonun gerçekleşmesi için mobil uygulamadan gaz sensöründen değer 1 olarak gelirse(gaz kaçağı anlamı taşır) ve e-posta yolla, alarm zilini çal gibi anlık kurallar oluşturulur. Sadece güvenlik değil sağlık ve konfor tarafında da yine alarmlar kullanılabilir. Yine mobil uygulamadan sıcaklık ve nem normal şartların üstüne çıkarsa yada altına düşerse beni haberdar et şeklinde alarmlar tanımlanabilir. Bu senaryolar elimizdeki sensörlerin çeşitliliğine göre farklı kombinasyonlarda çalışarak hayata anlam katabilir.

4.3.Envanterler

4.3.1. Sensör Ve Aktüatörler

Geliştirilen sistemde, gördük ki N çeşit sensör ve aktüatör ile çalışarak bağlı akıllı ev modeli tasarlamak mümkündür. Ancak bu çalışmada aşağıdaki sensör ve aktüatörlerden faydalanarak sistem tasarımı yapılmıştır.

Sıcaklık ve Nem Sensörü olarak DHT11 kullanılmıştır. DHT11 sıcaklık ve nem değerlerini dijital olarak veren gelişmiş bir modüldür ve uzun dönem çalışmaya müsaittir. Bread Board'a takılarak ilgili pinlerden değerler alınmıştır(Şekil 15).



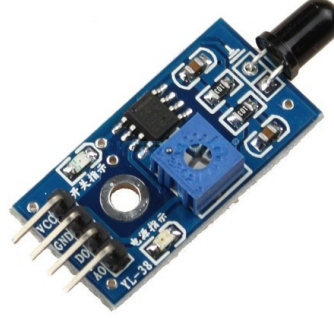
Şekil 15: Envanter – ısı nem sensörü

Gaz algılayıcı sensör olarak MQ6 LPG sensörü kullanılmıştır. Bu sensör belli değer aralıklarında izobütan ve propan algılamda kullanılır. Gaz yoğunluğuna göre analog değer çıkışı verir ve 5V ile çalışmaktadır(Şekil 16).



Şekil 16: Envanter – mq-6 gaz sensörü

Alev algılayıcı sensör(Flame sensör) ateş algılama için kullanılmıştır ve belli aralıklar içindeki yangın algılamasını sağlar. Bu sensör üzerinde bulunan trimpot sayesinde hassasiyeti artırılıp azaltılabilir(Şekil 17).



Şekil 17: Envanter – alev algılayıcı sensör

Kapı alarm sistemi için manyetik kapı alarm sensörü kullanılmıştır ve iki parçadan oluşur. Bu sensör içindeki reed röle ve mıknatıs sayesinde kapı/pencerenin açık veya kapalı olduğunu algılar. Kablo uçlarına ise bu değerleri gönderir(Şekil 18).



Şekil 18: Envanter – manyetik kapı alarm sensörü

Hareket algılayıcı sensör(PIR) ile ortamda hareketlilik olup olmadığı takip edilir. Bu sensör ile algılanan hareketlilik dijital çıktı olarak karta iletilir ve 5M'ye kadar mesafelerdeki hareketliliği algılayabilir(Şekil 19).



Şekil 19: Envanter – hareketlilik algılayıcı sensör

Buzzer ile ses çıkışını algılayıp node'un ayakta olup olmadığı kontrol edilmiştir. Bu aktüatör ile test çalışmalarıyla birlikte olay oluşması durumunda dikkat toplamak amacıyla alarmlarda da kullanılabilir (Şekil 20).



Şekil 20: Envanter – buzzer

Led aktüatörleri çalışmada çokça kullanılmıştır. Bu cihazlar hem test amaçlı hem de aydınlatmayı simüle etme amacıyla çalışmada kullanılmaya karar verilmiştir (Şekil 21).



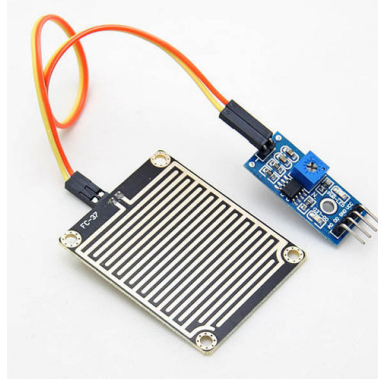
Şekil 21: Envanter – led

Fan aktüatör'ü yine bir sümulasyon oluşturmak amacıyla kullanılmıştır. Fan çalışmada klimayı simule etmektedir(Şekil 22).



Şekil 22: Envanter – fan

Yağmur/Su algılayıcı sensörü ile evde su baskını olup olmadığı tespit edilmeye çalışılmıştır. Bu sensör analog ve dijital çıkış vermektedir. Dijital taraftan yağmurun yağıp yağmadığı anlaşılırken analog taraftan ise de yağmurun şiddeti ölçümlenebilir(Şekil 23).



Şekil 23: Envanter – su algılayıcı sensör

4.3.2. Node'lar

Bir IOT uygulaması geliştirildiğinde yada bağlı akıllı ev çalışması yapıldığında burada node kullanmak önemlidir. Sensör verilerinin nodelara aktarılarak kategorilere ve gruplara ayrılması faydalı olacaktır. Yine sadece salondaki sıcaklık ve nem ilişkisini kurmak için node lardan faydalanmak büyük kolaylık sağlayacaktır. Burada çeşitli

donanım özelliklerine sahip ve çeşitli işletim sistemlerinde kullanılabilir. Bu çalışmada ise maliyetler de göz önünde bulundurularak ESP8266 cihazını node olarak kullanılmıştır. Sensörler bu node lara bağlanmıştır ve node larda ağ geçitlerine(gateway) bağlanarak çalışma gerçekleştirilmiştir.

4.2.3. Cihazlar

Cihaz(gateway) olarak aşağıdaki donanımlar üzerinde çalışma gerçekleştirilmiştir. Iphone Cep telefonu proje kapsamında geliştirilen akıllı ev uygulamasının testi için kullanılmıştır. Bu cihaz üzerine yüklenen uygulama ile Node'lara oradan da sensör ve aktüatörlere değerler gönderilip alınmıştır.

Samsung Cep telefonu (Android) üzerine IGNITE-AGENT kurularak lokasyon bilgilerine erişmek amaçlanmıştır. Bu cihaz diğer ev kullanıcılarını temsil etmektedir. Eve geldiğinde ışığı aç, alarmları kapat gibi senaryolar bu cihazın lokasyon bilgileriyle yönetilmiştir.

RaspberryPi3(Android) cihazı ağ geçidi(gateway) olarak kullanılmıştır. Bu cihaza Android işletim sistemi kurularak çalışma yapılmıştır. Android üzerine de IGNITE-AGENT kurularak node'larla iletişim sağlanmıştır.

4.4.Geliştirme Aşamaları

Uçtan uca akıllı ev geliştirmesi yapılan araştırmalar ve bu araştırmaların uygulamaya geçirilmesi sürecidir.

4.4.1. Node Geliştirmesi

Node tarafında yapılan geliştirme ile sensör ve aktüatörler'e veri akışını sağlamak amaçlanmıştır. İlk olarak node'u kablosuz ağ(Wi-Fi) üzerinden ağ geçidine(gateway) bağlamak gerekir bunu için Arduino ide kullanılmıştır. Arduino ide ile skech data dizininde Kablosuz ağ bilgileri ve node tanımı yapılmaktadır. Buradaki tanım IGNITE üzerinden konfigürasyon göndererek değiştirilebilir. Dosyanın formatı JSON dur. JSON içeriği aşağıdaki şekilde oluşturulmalıdır(Şekil 24).

```
1 {  
2   "nodeId": "Guvencik",  
3   "ssid": "<Wi-Fi SSID>",  
4   "password": "Wi-Fi Password",  
5   "gatewayId": "<GatewayIdWithoutColons>iotigniteagent",  
6   "state": "SEARCH_GATEWAY_STA"  
7 }
```

Ln: 7 Col: 2

Şekil 24: Akıllı ev mobil uygulaması – örnek düğüm(node) tanımlama verisi(JSON)

4.4.2. Gateway Entegrasyonu

Gateway tarafında da Node ve Bulut platform arasında köprü görevi görecek bir Android uygulama kurulmalıdır. Bu çalışmada IGNITE'in sağladığı açık kaynak kodlu olarak geliştirilen Ağ geçidi uygulaması kullanılmıştır(<https://devzone.iotignite.com/documents>).

Node tanımı yapıldıktan sonra ve ağ geçidine Android uygulaması kurulduktan sonra aynı ağ üzerinde bulunan ağ geçidi(gateway) ve Node TCP-IP protokolü üzerinden birbirlerini bulacaktır. Bağlantı sağlandıktan sonra veri akışı çift yönlü olarak sağlanacaktır. Ağ geçidi cihazı Android işletim sistemine sahip bir cihaz olmalıdır. Örnek çalışma kapsamında hem RaspberryPi-3 hem de Samsung Cep telefonu üzerine Android işletim sistemi kurularak veriler alınmıştır.

4.4.3. Mobil Uygulama Geliştirilmesi

Mobil uygulamanın geliştirilmesi süreci içerisinde geliştirme modüllere bölünerek yapılmıştır. Uygulama GIT platformunda açık kaynak kodlu olarak paylaşılmıştır. Tüm değişiklikler anlamlı (yorum) COMMIT mesajıyla GIT'e gönderilmiştir. İlerleyen süreçlerde akıllı evlere olan ilgi ve kullanımın artacağı ve bu artış neticesinde akademik ortamda ya da sektörden üzerinde geliştirme yapmak

isteyenler olabileceği ön görülmektedir. Dolayısıyla projenin açık kaynak olarak paylaşılması ve geliştirilmesi yönünde bir tercih yapılmıştır. GIT ile ilgili grafik ve paylaşım bilgileri EK G' de açıklanmıştır.

Mobil uygulama da ilk olarak kullanılacak DB kararlaştırılmıştır. Kodlama iOS ve Android ortamında %100 çalıştırılmıştır. Ancak aynı kodu WEB platformunda da çalıştırmak istenmesi durumunda DB uyumluluğu için hem mobilde hem de WEB de desteği bulunan SQLite DB tercih edilmiştir. Bu DB'ye uygun tablo yapıları tasarlanmıştır.

DB tercihinden sonra bulut platformunda kullanılacak API listeleri çıkarılmıştır. İlgili API'lar Swagger UI ile test edilerek ilgili sonucu verip vermediği geliştirme öncesinde netleştirilmiştir. Uygulamanın veritabanı tablo yapısı EK E' de örneklenerek açıklanmıştır.

DB ve API entegrasyonu sonrası giriş(login) ekranından rapor ekranlarına kadar tamamı bileşen(RCMP) bazlı olarak geliştirilmiştir. Bu sayede bileşenler üzerinde gerçekleşecek taşıma ve değişiklik süreçleri daha kolay hale gelmiştir.

Akıllı bir evin kurulumu ve kullanımını gerçekleştirmek için iki senaryo oluşturulmuştur. Kullanıcı ilk olarak kurulumlarını gerçekleştirdikten sonra ihtiyaç olmadıkça tekrar kurulum ekranlarını menü listesinden görmeyecektir. Kullanım kolaylığı sağlamak adına menüler iki farklı senaryo altında toplanmıştır.

Kullanıcı geliştirilen mobil uygulamada sensör tanımlarını yapıp ağ geçidine(gateway) oradan da uç düğüm 'e(node) gönderir. Ağ geçidine ve düğüme gönderilen komutlar' ın her biri JSON formatındadır. Bu değerler' in cihaza iletilip iletilmediğini görmek için log mesajları(Action Log) ekranları tasarlanmıştır. Log mesajları ekranından cihaza gönderilen JSON verisinin iletilip iletilmediğini ve gerçekten istenilen mesajın gidip gitmediği takip edilebilmektedir. Projenin geliştirilmesi sürecinde de yine modüllerin ve bu modüllerle gönderilen komutlar log mesajları ekranından takip edilerek ilerleme sağlanmıştır.

Geliştirilen mobil uygulama gerçek iOS işletim sistemine sahip bir cihazda ve Android işletimine ait bir cihazda denenerek kullanıcı testleri yapılmıştır. Ayrıca bulut platformuna yapılan API isteklerini test etmek için entegrasyon testleri de yazılmıştır.

Entegrasyon testleri yazılırken JEST teknolojisinden faydalanılarak API istek sonuçları sanallaştırılmıştır(mocking).

4.4.3.1. Melez(HYBRID) Ve Yerel(NATIVEAPP) Uygulama Farkı

Mobil bir cihaza geliştirme yaparken önceliklerin iyi belirlenmesi gerekmektedir. Eğer sizin için performans, enerji, cihazın özelliklerine erişim gibi durumlar söz konusuysa bu durumda yerel(NATIVEAPP) uygulama geliştirme ile devam etmek en uygun çözüm olacaktır.

Mobil ve Web uygulamaları geliştirirken izlenen yol genellikle şu şekildedir:

İlk olarak Web projesi geliştirilip kullanıma açılır daha sonra kullanıcıdan gelen istekler doğrultusunda Web uygulaması güncellenirken bir yandan da HYBRID yöntemlerle geliştirilen Web uygulaması mobil uyumlu hale getirilmiştir. Bu sayede geliştirilen sistem yüksek oranda hem Web te hem de mobilde çalışmış olabilir. Uygulama isteneni karşılar ancak çeşitli cihaz spesifik özelliklerden mahrum olur yada tam performans çalışamaz.

Son yıllarda ise Facebook tarafından geliştirilen RN projesi ile yeni bir yaklaşım ortaya çıkmış oldu. Mobil uygulamayı Web kadar kolay kodlayarak hem iOS te hem de Android' de çalışacak şekilde yerel uygulamaya dönüştürme yaklaşımı şeklindeydi. Bu yaklaşımın çok fazla benimsenmesinin ve günümüzde bilinirliği yüksek firmaların da mobil geliştirmelerini RN ile yapmasının ardından çeşitli plugin lerle geliştirilen mobil uygulama Web'e de uyarlanmaya başlandı. Bu dönüşüm sayesinde mobil tarafta yüksek performans ve enerji tasarrufu elde ederken aynı zamanda geliştirilen kod Web tarafındaki ekstra geliştirme yükünü de ortadan kaldırmış oldu.

Bu çalışmada RN ile mobil bir uygulama geliştirilerek hem iOS hem de Android platformunda çalıştırmıştır. Araştırmalar neticesinde bir karşılaştırma tablosu hazırlanmıştır. Aşağıdaki karşılaştırma tablosunda HYBRID uygulama, yerel uygulama ve RN ile geliştirilen bir uygulamanın farkı görülmektedir(Şekil 25).

	Yerel(Native) Uygulama	Kırma(Hybrid) Uygulama	React Native
Güvenlik	Yüksek Seviye	Sınırlı	Yüksek Seviye
Performans	Yüksek Seviye	Sınırlı	Yüksek Seviye
Enerji Tüketimi	Düşük	Yüksek	Düşük
Geliştirme	Kısmen Zor	Daha Kolay	Kolay
Öğrenme	Kısmen Zor	Kolay	Kolay
Geliştirme Eforu	Yüksek	Düşük	Düşük
Kodlama	Her Platform İçin Ayrı	iOS ve Android için Aynı Kod	iOS,Android ve Web için tek kod
Cihaza Erişim	Tüm Cihaz Fonksiyonlarına	Sınırlı Sayıda Erişim	Sınırlı Sayıda Erişim
Kodlama Dili	Yerel(Native) Programlama	Yerel Ve Web Programlama	Yerel Ve Bileşen Programlama
Yerel API'lara erişim	Yüksek	Sınırlı	Sınırlı
Hızlı Güncelleme	Kullanıcı Deneyimine Bağlı	Kolay	Kolay Ve Uzaktan Güncelleme İmkânı
Geliştirme Ortamı	Her Platform İçin Ayrı	Tek Kod Tüm Platformlara	Tek Kod iOS ve Android'e
Dükkanlara Dağıtma	Kendi Ortamına	Appstore ve Playstore'a	Appstore ve Playstore'a
Para Kazanma	Dükkanın Komisyonu Dahilinde Mümkün	Dükkanın Komisyonu Dahilinde Mümkün	Dükkanın Komisyonu Dahilinde Mümkün

Şekil 25: Melez(HYBRID), yerel(NATIVEAPP) uygulama ve RN

4.4.3.2. ES6 Avantajları

Bu bölümde geliştirme esnasında kullanılan ES6'nın önceki sürümlerine göre avantajları açıklanacaktır. Aslında ES6 ve ES5 farkı ayrı bir araştırma konusu olacak kadar güncel ve geniş bir alandır. Ancak bu çalışmada elde edilen deneyimler şu şekildedir:

İlk olarak ES6 ile gelen bir özellik olan Promise'leri ele alınırsa; eğer ardı ardına birden fazla asenkron kod çalıştırılması gerekiyorsa(ÖRN: bir REST API'a ardı ardına atılan http isteği) bu işlemi callback ler ile yapılabilir. Öncelikler API'ye bir istek gönderilir daha sonra isteğin callback'ine yerleşerek sonuç bekler gelen sonuç başarılıysa da API'e yeni bir istek daha yapılır. Promise'lerle ise bu asenkron çağrılar senkron bir şekilde tıpkı bir zincirin halkaları şeklinde ardı ardına yapılabilir.

Callback kullanımında kodu okuma daha zor hale gelirdi. Çünkü kod içe doğru kırılarak gidebilir. Ancak promise'te ise aynı iterasyon seviyesinde ilerleyeceği için bu işlemler daha anlaşılır bir hal almıştır. Aşağıdaki şekilde callback ve promise farkını gösteren bir kod örneği görülebilir(Şekil 26).

<pre> p.then((result) => { ... }) .then((result) => { ... }) .then((result) => { ... }) .then((result) => { ... }) .catch((err) => { ...error handle... }); </pre> <p style="text-align: center;">promise</p>	<pre> a.func({ b.func({ c.func({ d.func({ ...etc... }); }); }); }); </pre> <p style="text-align: center;">callback</p>
---	---

Şekil 26: Promise ve callback kodlama farkı

ES6 ile gelen diğer bir avantaj ise blok kapsamlı değişken tanımlanabilmesidir. ES5 te tüm değişkenler global olarak tanımlanabilir. Ancak yeni halinde ise diğer hatırı sayılır dillerde olduğu gibi her blok için ayrı bir değişken tanımlanabilmektedir. Çalışma kapsamında geliştirilen uygulama da bu bilgiler ışığında tutularak kodlama yapılmıştır.

Javascriptte artık gömülü ifadelerine izin veren şablon değişkenler(template literal) tanımlamak mümkündür. Bu sayede çok satırlı dizeleri birbirine bağlamak için ekstra bir çabaya ve koda ihtiyaç yoktur. Ayrıca yine bu dize içinde değişkenler tanımlayarak değişken içinde değişken'e parametre göndermek büyük bir esneklik olacaktır. Aşağıdaki şekilde çalışmada SQL sorguları yazarken kullanılan ve soru işaretleri(?) ile parametre gönderilen şablon değişken örneği görülmektedir(Şekil 27).

```

let sqlQuery = ` SELECT
  (T.total / T.count) as average,
  T.formattedDate
FROM (
  SELECT
    strftime('%H', formattedSensorCreateDate) as formattedDate,
    count(*) as count,
    sum(data) as total
  FROM sensorData
  WHERE formattedDate is not null
    AND formattedSensorCreateDate > (SELECT DATETIME('now', '-1 day'))
    AND deviceId = ?
    AND nodeId = ?
    AND sensorId = ?
  GROUP BY formattedDate
) as T
ORDER BY T.formattedDate
ASC`

```

Şekil 27: Örnek şablon değişken(template literal) kullanımı

Daha sezgisel ve okunabilir bir kullanım olan ok(arrow) fonksiyonu da ES'in yenilikleri arasındadır. Aşağıdaki şekilde çalışmada ok fonksiyonu olarak tanımlama gösterilmiştir(Şekil 28).

```

componentDidMount = () => {
  const {context} = this.props;
  context.showLoading();

  device.getDetail(context.token, context.device).then(device => {
    this.setState({device, modalVisible: true});
    context.hideLoading();
  });
}

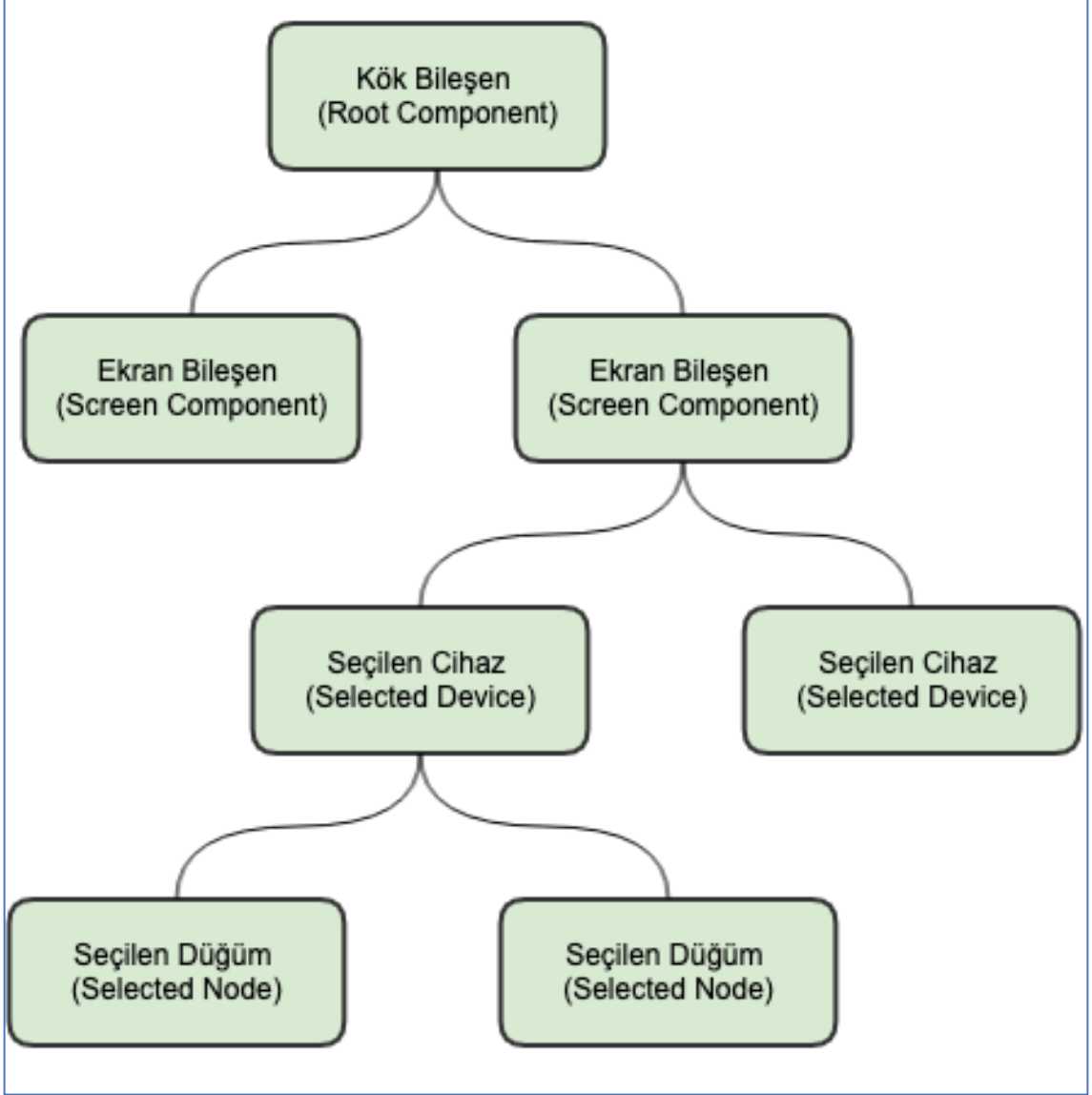
```

Şekil 28: Örnek ok(arrow) fonksiyonu kullanımı

Sınıf tanımlama(class defination), ok haritalama(arrow map) fonksiyonu, filtreleme(filter) gibi yenilikler yine ES6 ile hayatımıza girdi. Üstte de belirtildiği gibi ES6 ile gelen yenilikleri aslında bir araştırma konusu olarak ele almak gerekir. Ancak çalışmada kullanılan ve araştırmalarda öğrenilen kısmı açıklanmıştır. Bundan sonra yapılacak çalışmalarda Ecmascipt6'yı incelerken üstte verilen örnekleri ve açıklamaları da göz önünde bulundurmak faydalı olacaktır. Yapılan çalışmada ES6 ile buluta giriş örneği EK A' da gösterilmiştir.

4.4.3.3. RAPI ve RDX Yaklaşımları

Çalışmada üstteki bölümünde RN ile melez(HYBRID) ve yerel(NATIVEAPP) uygulama karşılaştırması yapılarak uygulamanın RN ile geliştirilmesine karar verilmiştir. Bu başlık altında ise RN ile geliştirme yapılırken yaşanan zorlukların en başında gelen durum(state) değişikliklerini diğer bileşenlere(RCMP) taşıma konusuna değinilecektir. RN bileşen bazlı (RCMP based) bir geliştirme yöntemi olduğu için bileşenler kendi durumlarını(state) tutabileceği gibi birbirlerinin durumundan da haberdar olmak isteyebilirler. Bu durumda arada durum üreten ve tüketen bir mimarinin kullanılması gerekmektedir. RDX yada RAPI bu ihtiyacın karşılığı sonucu doğmuştur. RAPI'nin implementasyonunun ve kullanımının daha kolay olduğu için çalışmada RDX tercih edilmemiştir. Aşağıdaki şekilde RAPI'nin çalışmadaki örneği görülmektedir(Şekil 29).



Şekil 29: Örnek RC API kullanım diyagramı

Şekil 29'a bakarak ilk olarak çalışmada buluta login olduktan sonra alınan TOKEN kök bileşen üzerinde tanımlana RC API üzerinde tutulur. Daha sonra alt bileşenlerden context.token şeklinde bu TOKEN'a erişim mümkündür.

Bir sensör tanımı yapmak için cihaz seçiminin ve düğüm seçiminin yapılması gerekmektedir. Seçim yapılarak alt bileşenlere doğru gidildikçe üstte bileşende seçilen cihazın ve ardından seçilen düğümün unutulmaması gerekir. Çünkü bu bilgileri sensör tanımlaması yapılacak bileşende kullanılması gerekmektedir. Bu akılda tutma işlemini yapabilmek için tıpkı bir oturum(session) mantığında olan RC API'ye seçtikçe değerleri set ederek en alt bileşenden erişim sağlanmıştır.

Ayrıca bir hatırlatma yapmak gerekirse RCAPİ'da tutulacak değerlerin session mantığında sadece diğer bileşenlerde tutulacak değerler olması gerekmektedir. Aksi halde her durum değişikliğinde RN sadece ihtiyaç olanları değil tüm ekrandaki bileşenleri yeniden render ederek çalışma hızının düşmesine sebep olabilir. Bilindiği üzere RN tarafında her durum değişikliği yeniden ekran boyama(rerender) anlamına gelmektedir. RN ile RCAPİ'nin kullanım örneği EK B'de gösterilmiştir.

Aşağıdaki şekilde çalışmada RN ile R-CONTEXT'in oluşturma adımlarını görülmektedir. Üstte bahsedilen kolay implementasyonun tam olarak karşılığı bu şekildeki kod satırlarıdır(Şekil 30).

```
import React from 'react';

const {Provider, Consumer} = React.createContext();

const CtxProvider = Provider;
const CtxConsumer = Consumer;

export {CtxProvider, CtxConsumer}
```

Şekil 30: Örnek RCAPİ oluşturma

Şekil 31'de çalışmadaki cihaz detay bileşeninde (DeviceDetailComponent) R-CONTEXT'te tutulan TOKEN ve cihaz bilgileri çekilerek buluttan bu cihazın bilgilerinin istenmesi(FETCH) görülmektedir.

```
componentDidMount = () => {
  const {context} = this.props;
  context.showLoading();

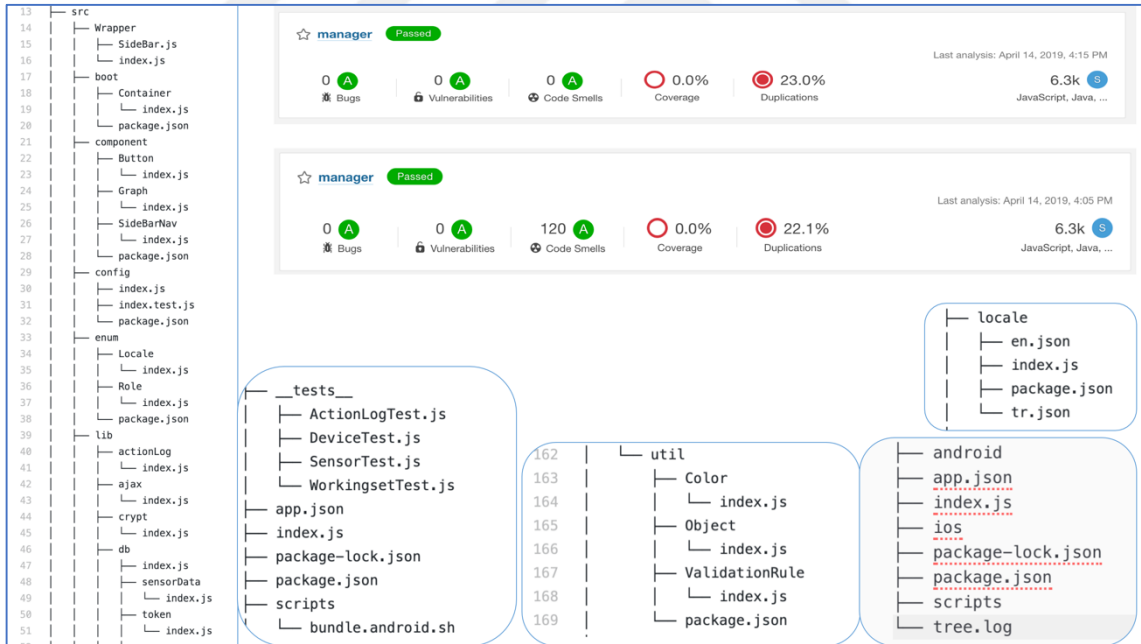
  device.getDetail(context.token, context.device).then(device => {
    this.setState({device, modalVisible: true});
    context.hideLoading();
  });
}
```

Şekil 31: RCAPİ' dan TOKEN ve device bilgilerini alma ve FETCH İsteği yapma

4.4.3.4. Projenin Dizin Ağacı(Directory Tree)

Bu çalışma kapsamında geliştirilen uygulama GIT üzerinden herkese açık(public) olarak paylaşılmıştır. Bundan sonra yapılacak çalışmalara bir ışık tutması adına projenin kodlarını gizleme yoluna gidilmemiştir.

Geliştirmeyi yaparken okunabilirliği artırmak ve kodlamayı kolaylaştırmak adına proje kategorilere ayrılmıştır. Geliştirme tamamlandıktan sonra kodu SONAR analizinden geçirerek farkında olmadan yapılan kötü kodlamalar(code smells) düzeltilmiştir. Aşağıdaki şekil incelenerekte hem projenin hiyerarşik yapısı hem de SONAR analiz sonucunu görülebilir(Şekil 32).



Şekil 32: Projenin dizin ağacı ve SONAR analiz sonucu

4.4.3.5. Bulut'a İstek(Request) ve Sonuç(Response) Örnekleri

Bulut platformu ile geliştirilen mobil uygulamanın iletişimi REST API'lar arasında olduğu üstteki bölümlerde belirtilmiştir. Bu bölümde ise bulut platformuna atılan

isteklerin parametre tipini ve dönen JSON verilerini ele alınacaktır. Bulut platformuna atılan http isteklerini swagger ile incelenerek birkaç örnek ekran görüntüsü elde edilmiştir.

Aşağıdaki ekran görüntüsünde bir cihazın detay bilgilerini sorgulamak için istek yapılacak URL ve yapılan istek sonucunda dönen JSON verisi örneğin görülebilir. Bu istek esnasında her şey yolunda giderse http durum kodu olarak 200 dönülmektedir(Şekil 33).

```
İstek(Request) URL
https://api.ardich.com:443/api/v3/device/summary?device=14%3A1f%3A78%3A2e%3Aae%3Ad8%40iotigniteagent

Sonuç(Response) Body
{
  "content": [
    {
      "deviceId": "14:1f:78:2e:ae:d8@iotigniteagent",
      "imei": "353468081435598",
      "status": "VALID",
      "model": "SM-G610F",
      "modeAppVersion": "AR.IGF.0.8.42",
      "lockStatus": false,
      "lostStatus": false,
      "createdAt": 1542564650872,
      "lastPresenceDate": 1556738941416,
      "presence": {
        "state": "OFFLINE",
        "clientIp": "192.168.1.34"
      },
    },
  ],
  "network": {
    "telephony": {
      "networkRoaming": false,
      "simOperator": null,
    },
  },
}
```

```
Sonuç Kodu(Response Code)
200
```

Şekil 33: Bulut' a yapılan örnek bir http get isteği ve sonucu

Aşağıdaki ekran görüntüsünde ise bulut platformuna yapılan POST isteği örneği görülmektedir(Şekil 34). Burada parametre olarak girilen cihaz kodu ve güvenli mesaj isteği ile cihaza güvenli bir mesaj gönderilebilir. Geliştirilen uygulamada da cihaza mesaj gönder modülü eklenmiştir. Bu modül sayesinde cihazın ayakta olup olmadığından emin olunabilir.

İstek(Request) Parametresi				
Parameter	Value	Description	Parameter Type	Data Type
code	317b887a87274466886bd8d1b1ebe200	code	path	string
genericTrustedMessage	{"message":"Burasi Cihaza Gidecek Mesaj"}	genericTrustedMessage	query	string

Try it out! [Hide Response](#)

İstek(Request) URL

https://api.ardich.com:443/api/v3/device/317b887a87274466886bd8d1b1ebe200/control/genericTrustedMessage?genericTrustedMe

Sonuç(Response Body)

```
{
  "response": "",
  "links": []
}
```

Dönüş Kodu(Response Code)

201

Şekil 34: Bulut'a yapılan örnek bir http post isteği ve sonucu

4.4.4. Mobil-Web Farklılıkları

Gelişen teknolojilerle birlikte kullanıcı alışkanlıklarının da değiştiği gerçeğini göz önünde bulundurulursa; günümüzde mobil kullanımının giderek artacağı görmezden gelinemez. Bu çalışmada da değinildiği üzere daha rahat ve konforlu yaşam giderek yükselen bir tercih olmaktadır.

IGNITE alt yapısı kullanılarak geliştirilen mobil uygulama akıllı ev özelinde ve bu amaç doğrultusunda geliştirilmiş bir uygulamadır. Gereksiz modül karmaşasından uzak sadece ayar rapor ekranlarının olduğu ve sadece ev kullanımında ihtiyaç duyulacak sensör/aktüatörleri kontrol altına alabileceği bir uygulamadır.

Web tarafından baktığımızda ise IGNITE bir MDM çözümdür. İçinde bağlı akıllı ev konseptini de barındırmasına rağmen uzaktan bakıldığına hemen her koşulda içinden bir akıllı ev yönetim ekranlarını çıkarıp anlamının pek kolay olmayacağı bir gerçektir.

Özetlemek gerekirse geliştirilen mobil uygulama web tarafının içinde bulundurduğu özelliklerin sadece akıllı ev katmanlarını barındıran bir geliştirmedir.

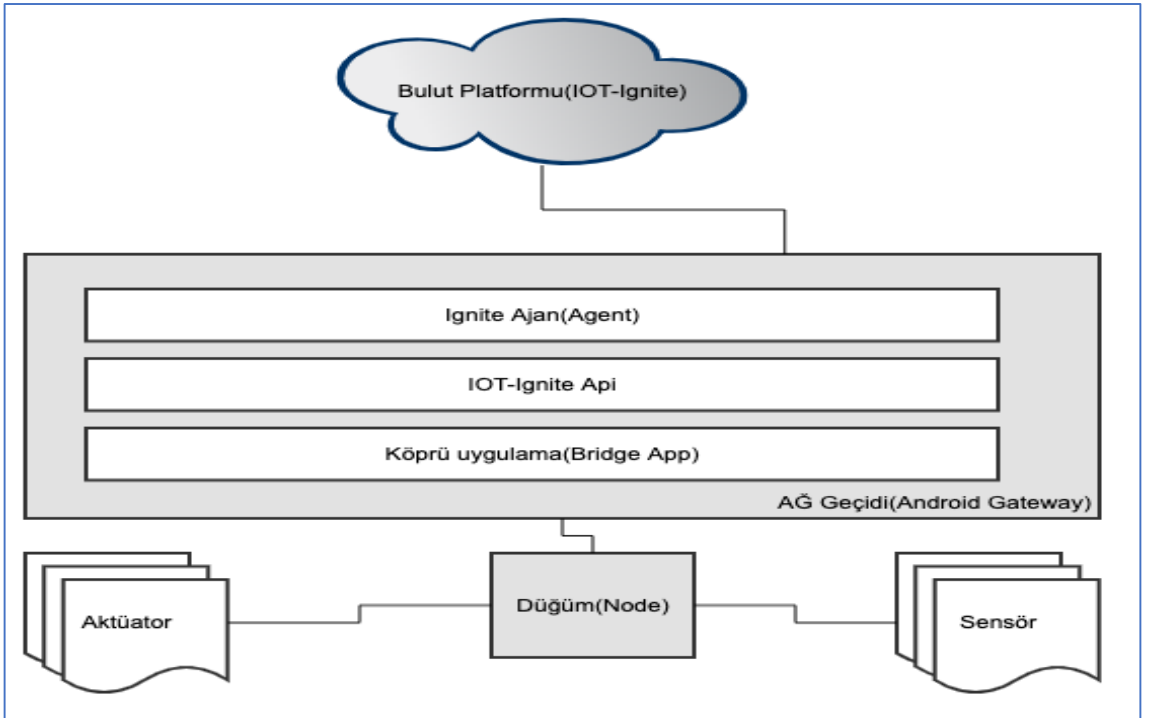
4.4.5. Test Geliştirme Ve Doğrulama Adımları

Akıllı ev uygulaması geliştirme aşamasında çeşitli doğrulama işlemleri yapılmıştır. Bu çalışmada kullanıcı testleri iOS ve Android işletim sistemlerinde ayrı ayrı yapılmıştır. Geliştirme esnasında developer testi yapılmıştır Ancak bu yeterli bir doğrulama yöntemi olmadığı için farklı bir bakış açısı katması adına farklı kullanıcıların telefonuna kurularak kullanımı hakkında gözlemler yapılmıştır.

4.5.Mimariler/Yapılar

4.5.1. IGNITE Mimarisi

Günlük hayatımızın bir parçası haline gelen akıllı evlerin bir örneğini bu çalışma kapsamında tasarlanmıştır. Bu akıllı ev tasarımı yaparken IGNITE bulut platformun kullanılmıştır. IGNITE kütüphanesiyle sensör tanımlama yapma işlemi EK D'de açıklanmıştır. IGNITE bulut platformunun ana hatları ve sistem mimarisini açıklanır; bulut platform, ağ geçidi , düğüm(Node) katmanı ve Şeylerin(things) bulunduğu katmanla birlikte 4 katmanda özetlenebilir(Şekil 35).



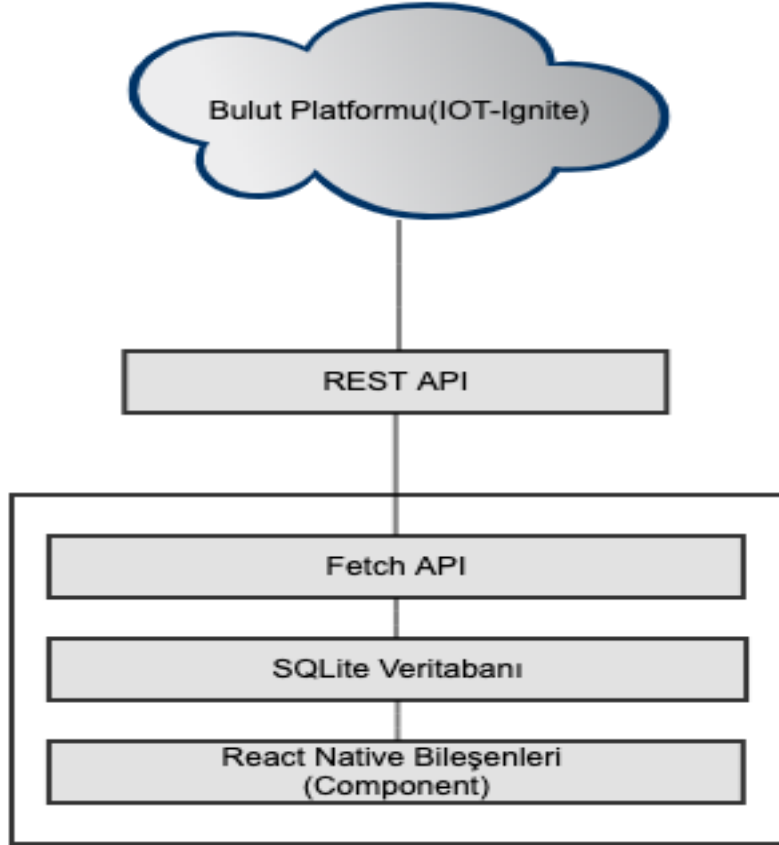
Şekil 35: IGNITE mimarisi

4.5.2. Mobil Uygulama Mimarisi

Geliştirilen mobil uygulamanın tanıtımın yukarıda yapılmıştır. Bu bölümde ise mobil uygulamanın nasıl bir sitem üzerinde geliştirildiği açıklanacaktır.

Akıllı ev uygulaması ağ geçidi ve düğümlerle bulut platform üzerinden haberleşir. IOT uygulamalarında meydana gelen güvenlik açıklarına çözüm üretmek için böyle bir geliştirme yapılmıştır.

Bulut platform, akıllı ev uygulaması ve evimize yerleştirdiğimiz ağ geçidi tamamıyla güvenli olan HTTPS protokolü üzerinden güvenlik doğrulamalarından geçerek gerçekleşir. Bulut platform ve akıllı ev uygulaması REST API üzerinden haberleşme sağlar(Şekil 35-36).

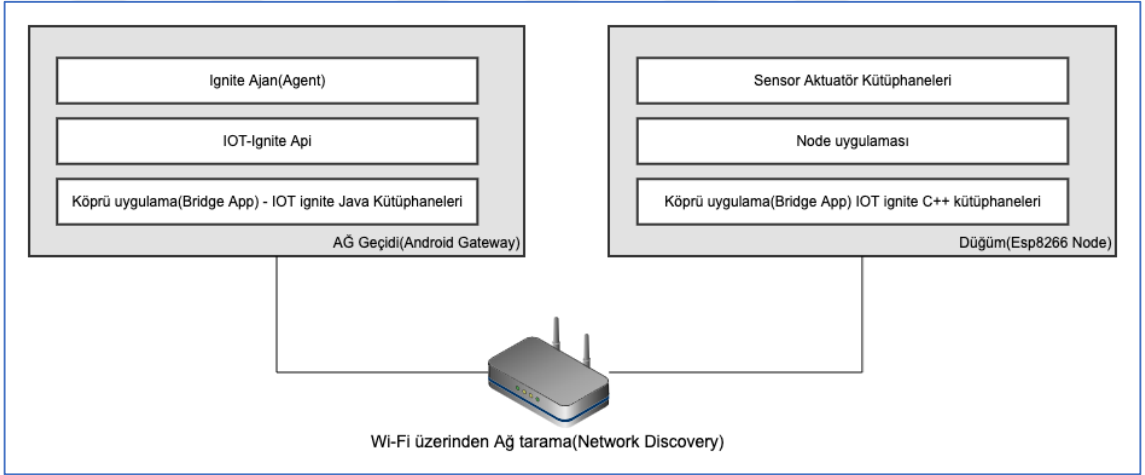


Şekil 36: IGNITE bulut platformu ve mobil uygulama mimarisi

4.5.3. Node-Gateway Entegrasyon Mimarisi

Evde kurulan akıllı ev sisteminin her bileşeni önemlidir ancak düğüm katmanı daha da önemlidir. Zira bu katman bulut ile mobil uygulama ve bulut ile düğümler arasında bir köprü görevi görmektedir.

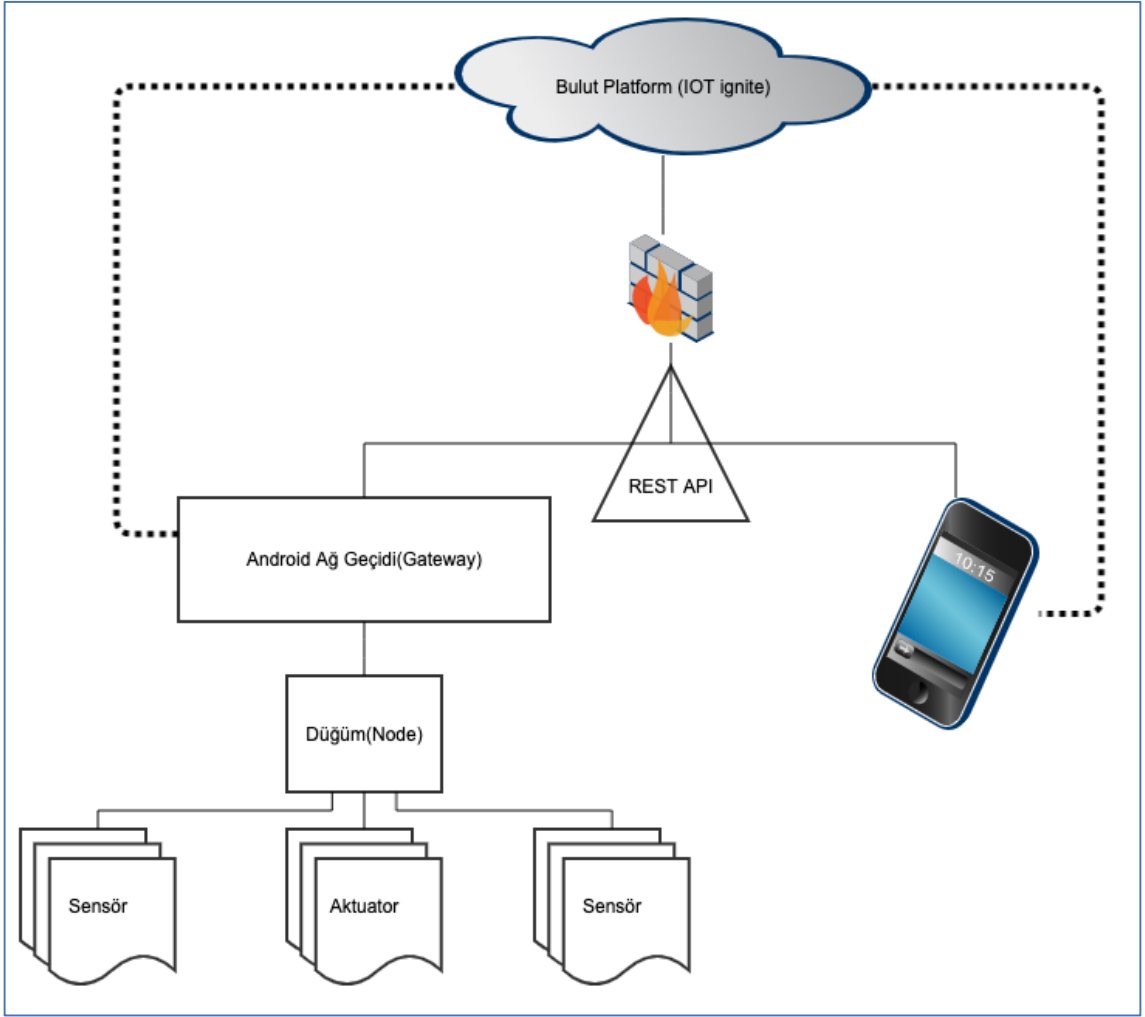
Çalışmada IGNITE-AGENT ve IGNITE-API'ler kullanılarak düğüm katmanının ağ tarama ile ağ geçidine bağlanması sağlanmıştır. Düğüm olarak ESP8266 kullanıldığı için ekstra bir Wi-Fi(Kablosuz Ağ) modülü takmaya gerek kalmamıştır. Hem maliyetler hem de bu modülü üzerinde bulundurması sebebiyle ESP8266 kartı tercih edilmiştir. Şekil 37'de aynı ağ üzerindeki düğüm ve ağ geçidinin konumları görülmektedir. Düğüm(node) geliştirmesinde kullanılan arçalar EK F de açıklanmıştır.



Şekil 37: IGNITE düğüm(node) ve ağ geçidi(gateway) mimarisi

4.5.4. Genel Sistem Mimarisi

Şuana kadar yapılan çalışmayla akıllı bir sistem tasarlanmıştır. Yazılanlardan ve kodlamalardan biraz uzaklaşarak sisteme yukardan bakıldığında nasıl bir mimarinin oluşturulduğunu anlamak çok güç olmayacaktır. Tasarlanan bu sistemde küçük bileşenler bir araya gelerek daha büyük bileşenleri oluştururlar. Son olarak ta uçtan uca bir akıllı evin tüm bileşenleri ve hatları meydana çıkmış olur (Şekil 38).



Şekil 38: Genel sistem mimarisi

BEŞİNCİ BÖLÜM

AĞ GEÇİDİ UYGULAMASI

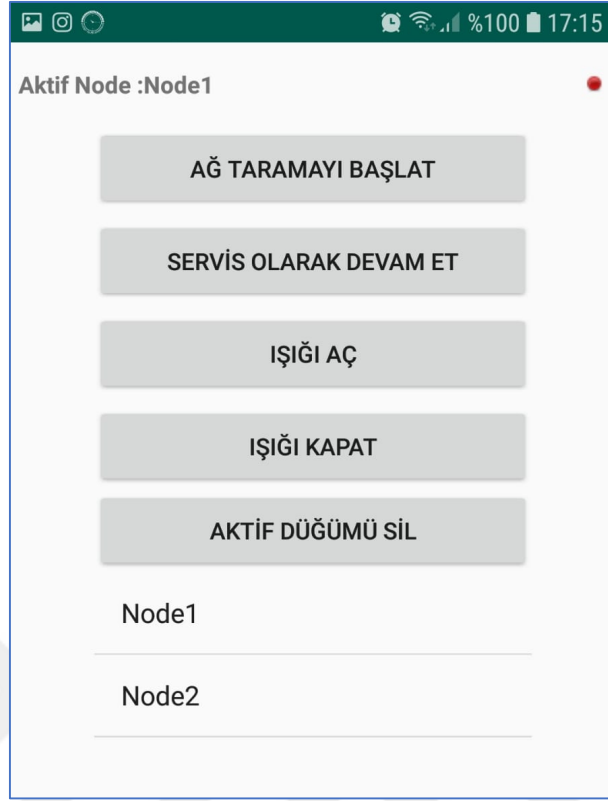
5.1. Android İşletim Sistemini Ağ Geçidi Olarak Kullanmak

5.1.1. Ağ Geçidinin Tanıtımı ve Çalışma Mantığı

Ağ geçidi(gateway) farklı ağ iletişim kurallarını kullanan iki farklı bilgisayarın arasındaki veri iletimini sağlar. Çalışmadaki yapıda kullanılan her bir düğüm(node) aslında bir bilgisayardır. Keza bulut platformu da birbirine bağlı birkaç bilgisayardan oluşmaktadır. Kullanılacak Android ağ geçidi(gateway) ise düğümünden buluta direkt iletişim olamayacağı için bu iki bilgisayar kümesi arasında köprü görevi görmektedir.

Ağ geçidini geliştirirken IGNITE kütüphanelerinin kullanılmasındaki diğer maksat çeşitli güvenlik gerekçeleridir. Bu durumda buluttan düğüme ve düğümünden bulut'a olan veri iletişimi ve güvenliği kapalı kütüphaneler ve arkasındaki profesyonel ekip aracılığıyla sağlamaktadır.

Belirtilen üzere bu çalışmada IGNITE ile geliştirilen Android uygulamayı ağ geçidi olarak kullanılacaktır. Bu uygulama düğümlerden aldığı verileri güvenli bir şekilde bulut platformuna aktarmaktadır. Benzer şekilde buluttan düğüme doğru olan iletişim de yine aynı ağ geçidi üzerinden olacaktır. Uygulamanın çıktısı aşağıdaki gibidir(Şekil 39).

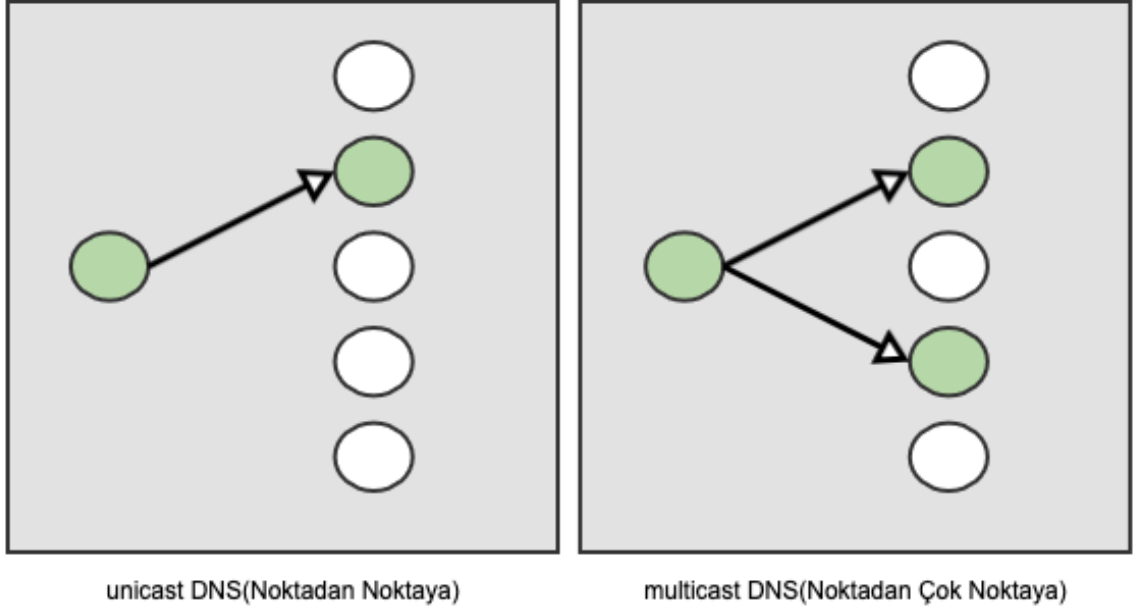


Şekil 39: Ağ geçidi – geliştirilen uygulama

5.1.2. Ağ Geçidi Ve Ağ Tarama Teknolojisi

Ağ tarama (Network Discovery) bir cihazın(Bilgisayar, Telefon...) ağ üzerinde kendisini aktif olarak işaretlemesiyle aynı ağı ve protokolü kullanan diğer cihazların bu aktif cihazı görmesi ve bağlantı kurması işlemidir. Ağ üzerindeki cihaz kendisini ağ taramaya açabilir ya da kapatabilir. Ağ taramasına açık olan cihazlar birbirlerini iletişime geçebilirler. Sektörde bu teknoloji Zeroconf ya da bonjour olarak bilinir.

Ağ taraması yapabilmek için geleneksel bir UDNS(noktadan noktaya) sunucusuna ihtiyaç yoktur. Amaç zaten internete çıkmadan aynı ağda bulunan cihazların iç ağda(intranet) haberleşebilmesini sağlamaktır. Dolayısıyla DNS benzeri işlemleri yürütmek için MDNS(çok noktaya) kullanılır. Aşağıda DNS ve MDNS farkını gösteren şekli inceleyebilirsiniz(Şekil 40).



Şekil 40: Ağ geçidi – UDNS ve MDNS farkı

Çalışmada IGNITE kütüphanesiyle birlikte MDNS sorgusu yaparak Android cihaz ile ESP8266'nın haberleşmesi sağlanacaktır. Android cihaz ağ geçidi(gateway) olacağı için düğümlerin(Node) bu ağ geçidinin IP adresini tespit edebilmesi gerekmektedir.

Eğer bir ağ tarama yöntemi kullanılmazsa tek tek her eklenen düğüm'e(node) ağ geçidinin IP adresinin girilmesi gerekmektedir. Bu ise çalışmadaki yaklaşımın dışına çıkılarak manuel konfigürasyon tanımlanmasını gerektirir.

Aşağıdaki şekildeki(Şekil 41) kod örneğinde JMDNS kütüphanesi kullanarak Java ile nasıl servis taramasının yapıldığı görülmektedir. Taramayı JMDNS yaparak Implementasyonundaki serviceAdded(...) ve serviceRemoved(...) methodları içerisinde yeni servis eklenme olayı(event) ve servisin kaldırılma olayı görülebilir (Şekil 42).

```
@Slf4j
public class DiscoveryService {

    public static void main(String[] args) throws InterruptedException {

        try {
            // Create a JmDNS instance
            JmDNS jmdns = JmDNS.create(InetAddress.getLocalHost());
            log.info("JmDNS created. JmDNS : {} ", jmdns);

            // Add a service listener
            jmdns.addServiceListener(Constants.TYPE, new ListenerHandler());
            log.info("Service Listener Added To JmDNS");

            // Wait a bit
            Thread.sleep( millis: 60000);

        } catch (UnknownHostException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Şekil 41: Ağ geçidi – JMDNS ile ağ tarama

```
@Slf4j
public class ListenerHandler implements ServiceListener {

    @Override
    public void serviceAdded(ServiceEvent event) {
        log.info("new service added. info : {} ", event.getInfo());
    }

    @Override
    public void serviceRemoved(ServiceEvent event) {
        log.info("Service removed! info : {} ", event.getInfo());
    }

    @Override
    public void serviceResolved(ServiceEvent event) {
        log.info("Service Name Resolved. info : {}", event.getInfo());
    }
}
```

Şekil 42: Ağ geçidi – JMDNS java implementasyonu

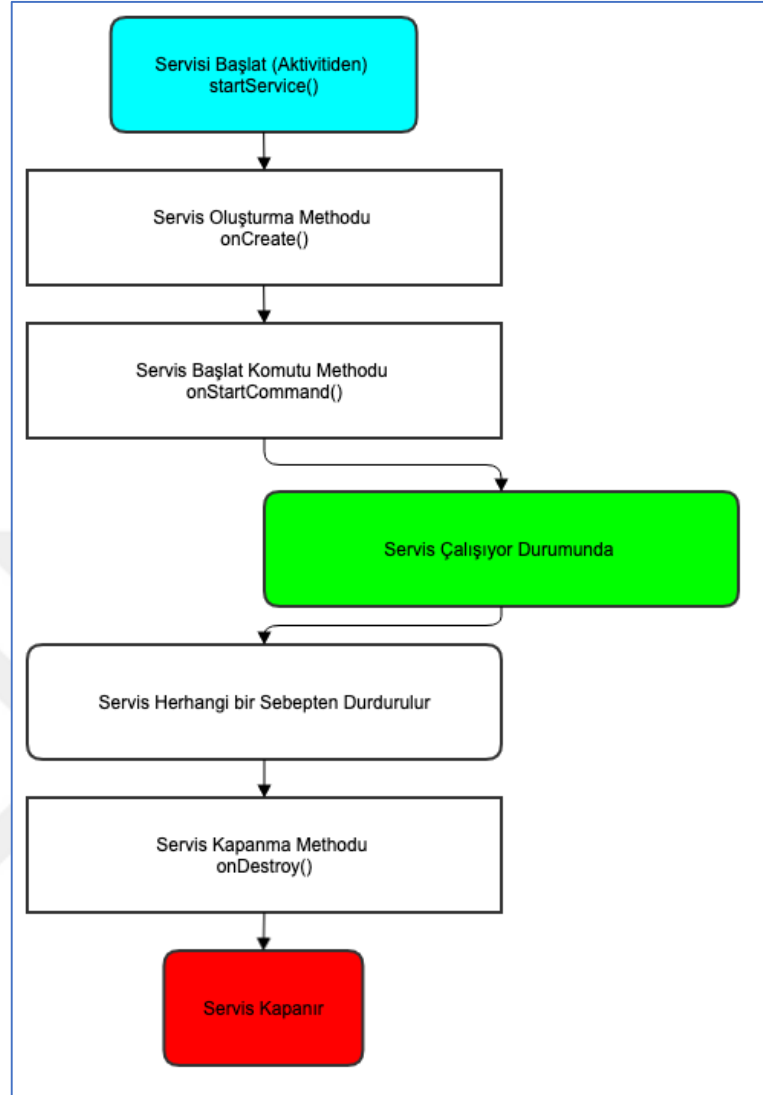
5.2.Android Servis Mimarisi

Android işletim sisteminde bir uygulama çalıştığında sistem bir Linux görevi(process) oluşturur. Varsayılan olarak tüm komponent ve uygulamalar Main Thread üzerinde gerçekleşir.

Servisler ise Aktiviteler üzerinden tetiklenerek bir process üzerinde ayrı bir thread olarak çalışırlar. Bu çalışmada da bir aktive oluşturarak MDNS ile ağ taraması yapacak servisi tetiklenmiştir. Bu sayede ağ geçidi(gateway) ekranı olmadan yoluna devam etmektedir.

Androidde aktivite'den tetikleme yapmaksızın servis mimarisi tasarlamakta mümkün dür. Google Play servisi buna en iyi örnektir. Ancak bu şekilde bir tasarım yapabilmek için yani işletim sistemi açılır açılmaz aktiviteden bağımsız bir servis başlatabilmek için Android'in kaynak kodunun derlenerek işletim sistemi içine bu tasarlanan servisi gömmek gerekmektedir. Dolayısıyla çalışmada ilk açıklaması yapılan yöntem ile servis geliştirmesi yapılmıştır.

Aşağıdaki şekli inceleyerek aktivite üzerinden başlatılan servisin yaşam döngüsü görülebilir(Şekil 43).



Şekil 43: Android servis yaşam döngüsü

5.3. Android Servis Olarak Ağ Geçidi Uygulaması

Android uygulamasını servis olarak geliştirmek için bir ide aracılığıyla Android uygulaması oluşturuldu. Yukarıda da açıklaması yapıldığı üzere çalışma ilk aşamada uygulama olarak açıp daha sonra arka planda çalışan ve kullanıcı ile etkileşime girmeyen bir servis olarak yoluna devam etmektedir.

Bu çalışmanın geliştirilmesi aşamasında Android Studio ide kullanılmıştır. Projenin bağımlılıklarını manuel yönetmeden kaçmak için çalışmada GRADLE ve MVN birlikte kullanılmıştır. Yine imzalama ve APK oluşturma adımlarını kolaylaştırmak için GRADLE üzerinde çeşitli tanımlamalar yapılmıştır.

Aşağıdaki şekil incelenerek projenin son halindeki bağımlılık listesi görülebilir(Şekil 44).

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation 'com.android.support:support-annotations:28.0.0'
    testImplementation 'junit:junit:4.12'
    implementation 'com.google.code.gson:gson:2.6.2'
    implementation 'com.ardic.android:IoTIgnite:0.8.2'
    implementation 'com.ardic.android:utilitylib:0.1'
    implementation 'com.ardic.android:connectivitylib:0.2'
    implementation 'org.apache.commons:commons-lang3:3.4'
    implementation 'org.jmdns:jmdns:3.5.1'

    testImplementation 'junit:junit:4.12'
    testImplementation 'org.mockito:mockito-core:2.7.22'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'org.mockito:mockito-android:2.7.22'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

    implementation 'com.afollestad.material-dialogs:core:0.9.1.0'
    implementation 'com.ardic.android.iot:HwNodeAppTemplates:1.0'

    configurations {
        all*.exclude group: 'com.android.support', module: 'support-v13'
    }
}
```

Şekil 44: Ağ geçidi – GRADLE ile bağımlılık yönetimi

Eğer çalışmada GRADLE veya MVN gibi bir araç kullanılmıyaydı bu listedeki tüm JAR'ları tek tek indirip çalışmaya elle eklemek durumunda kalınacaktı. Bu da farklı ortam ve dizinlerde proje çalıştırılmak istendiğinde bu işlemi her defasında yaparak gereksiz zaman kaybı oluşacağı anlamına gelmektedir. Ayrıca üstteki bağımlılık listesinde projenin test kodunu yazarken kullanılan JUNIT ve Mockito kütüphaneleri de tanımlanmış bulunmaktadır.

Uygulamanın Servis olarak çalışabilmesi için Android tarafında bir takım izinlere ihtiyaç duyulur. Android işletim sistemi güvenlik gerekçeleri kapsamında başlangıçta internet erişimi, galeri erişimi gibi bir takım izinleri uygulamaya kapatır. Uygulama geliştirirken “AndroidManifest.xml” dosyası üzerinde ihtiyaç duyulan izinlerin tanımlanması gerekmektedir.

Çalışma kapsamında geliştirilen ağ geçidi uygulaması Wi-Fi erişimi, İnternet erişimi ve ağ erişimi iznine ayrı ayrı ihtiyaç duymaktadır. Uygulama cihaza

yüklendiğinde kullanıcı bu izinler için bilgilendirilebilir. Aşağıdaki şekilde manifest dosyasında tanımlanan izinler görülmektedir(Şekil 45).

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Şekil 45: Ağ geçidi – uygulamaya ağ ve internet erişim yetkisi

5.4.Android Servis Uygulaması Ve Activity Farkı

Android işletim sistemi üzerinde uygulama geliştirirken bu iki kavramın iyi bilinmesinde fayda olacaktır. Aksi halde uygulama geliştirmeye daha başlamadan temel iki özellikten eksik kalmış bilgilerle ilerlemesi güç olacak ve buda geliştirilen yazılımın kabiliyetlerini etkileyecektir.

Android tarafında aktivite(Activity) kullanıcı ile etkileşimi sağlayan yapıdır en özet ifadeyle. Yani kullanıcının bastığı butonları ve girdiği form verilerini alıp işlemeye yarayan bir kullanıcı ara yüzüdür.

Servis kısmı ise; kullanıcıdan alınacak bilgilere ihtiyaç duyulmaksızın ve yine bir ekrana gerek kalmadan(headless) arka planda çalışan yapıdır. Örnek olarak Android.Media kütüphanesi ile arka planda tablet ve telefonlarda müzik çalma işlemi bu yapıyı çok net açıklamaktadır.

Bu iki yapı arasındaki farkı ele alınacak olursa Servisler Thread yaşam döngüsüne bağlıdır ancak Aktiviteler bu yaşam döngüsüne bağlı değildir.

5.5.Test Kodlarının Yazılması ve Doğrulama Aşamaları

Yazılım geliştirme süreçlerinin bir parçası olan birim testleri ve entegrasyon testlerinin öneminden bahsedilmiştir. Test kodunun önemine kısaca değinilirse yazılan her koşulun doğrulanması işlemidir.

Yine entegrasyon süreçlerini değil de birim testlerini yazmayı bu yöntemlerle doğrulama yaparak ilerleneceği yukarıda belirtilmişti.

Bir uygulama geliştirme sürecinde çeşitli sınıflar, methodlar ve değişkenler olacaktır. Bu değişkenlerin, sınıfların ve diğer yazılım parçalarını bilinen bir işi yaptıktan sonra beklenen değeri alması ya da beklenen değeri dönmesi aksi halde yolunda gitmeyen bir durum varsa bunun daha geliştirme sürecinde bilinmesi çok önemlidir.

Aşağıda paylaşılan şekil proje kapsamında geliştirilen servis uygulamasında HomeActivity içinde butonlara tıklama olaylarının(click event) tespiti görülmektedir. Buradaki kod örneğinde kullanıcının bastığı butona bakarak ilgili butona bağlı olan işlem tetiklenmektedir. Doğrulama aşamalarında test kodu yazılırken de yine buradaki her *if*'in istenen sonucu verip vermediği kontrol edilmelidir(Şekil 46).

```
@Override
public void onClick(View v) {

    if (v.equals(startNDSBtnView)) {
        startApplicationAsService();
        return;
    }

    if (v.equals(hideActivityBtnView)) {
        return;
    }

    if (v.equals(ledOnBtnViewLvt)) {
        doLedOnAction();
        return;
    }

    if (v.equals(ledOffBtnViewLvt)) {
        doLedOffAction();
        return;
    }

    if (v.equals(removeActiveNodeViewLvt)) {
        removeActiveNode();
        showMessage("Aktif Düğüm Silindi");
        return;
    }

}
```

Şekil 46: Ağ geçidi – guard clause ile test caseleri

Üstteki kod örneğinde iç içe *if* ler kullanmak yerine okunabilirliği artırmak adına guard clause kullanılmıştır. İlişkili durumların iç içe yazılması yerine her bir koşulun bağımsız yazılması işlemidir guard clause(Şekil 46).

5.6.İmzalama ve Yükleme İşlemleri

Android uygulaması olarak geliştirilen ağ geçidinin IGNITE-API'leri ile konuşabilmesi için öncelikle bir anahtar(key) ile imzalanıp bulut'a yüklenmesi gerekmektedir. İmzalama işlemi play.store'a uygulama yüklemek için takip edilmesi gereken süreç ile aynıdır. Uygulama geliştirilir, anahtar oluşturulur ve dükkana(store) yüklenir.

Aşağıdaki keytool komutu Linux tabanlı bir işletim sisteminde terminal üzerinde çalıştırarak bir anahtar oluşturulmuştur(Şekil 47).

```
keytool -genkey -v -keystore imza.keystore -alias imza-kapsami -keyalg RSA -keysize 2048 -validity 10000
```

Şekil 47: Ağ geçidi – keytool komutuyla imza oluşturma

İmzalama işlemi GRADLE'a yaptırılacağı için imza bilgilerini(alias,password) gradle.properties dosyasına ekleyerek yine bu tanımlama ortak bir yere alınmış oldu. Bundan sonraki aşamada build.gradle dosyasında properties dosyasında değişken olarak tanımlanan bu değerleri kullanarak RELEASE ve DEBUG APK'sı oluşturulabilir.

Aşağıdaki şekilde ise build.gradle dosyasında imzalama tanımının nasıl yapıldığını görülebilir. Bu tanım ile değerler gradle.properties dosyasından alınmaktadır(Şekil 48).

```
signingConfigs {  
    //APK sign parameters  
    release {  
        storeFile file(RELEASE_STORE_FILE)  
        storePassword RELEASE_STORE_PASSWORD  
        keyAlias RELEASE_KEY_ALIAS  
        keyPassword RELEASE_KEY_PASSWORD  
    }  
    debug {  
        storeFile file(RELEASE_STORE_FILE)  
        storePassword RELEASE_STORE_PASSWORD  
        keyAlias RELEASE_KEY_ALIAS  
        keyPassword RELEASE_KEY_PASSWORD  
    }  
}
```

Şekil 48: Ağ geçidi – GRADLE imzalama ayarları

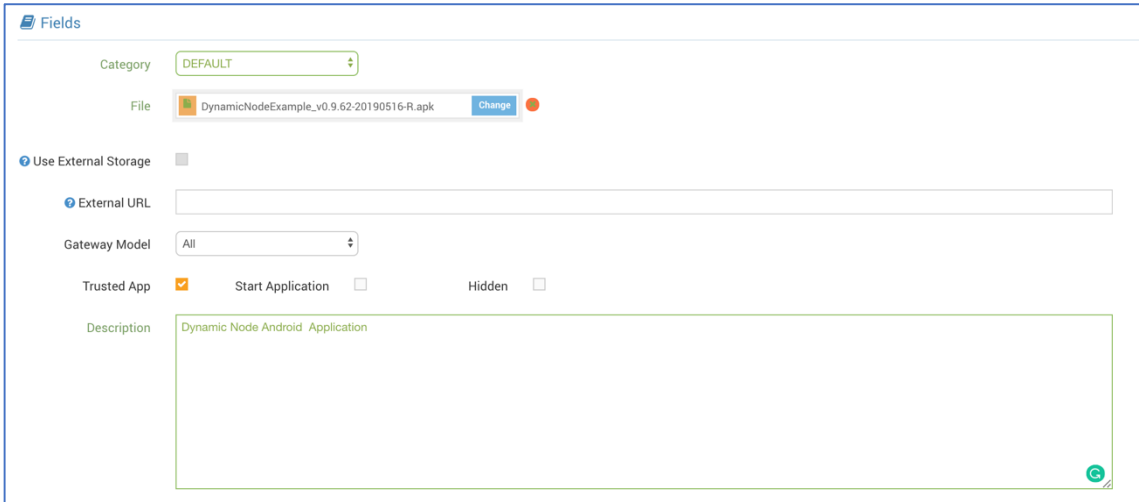
Çalışma kapsamında üstte oluşturulan imza kullanılarak imzalı bir Android uygulama oluşturulmuştur. Aşağıdaki şekilde Linux terminal üzerinden GRADLE aracılığıyla Android APK'nın nasıl çıkarılacağı ve sonuç mesajı görülmektedir(Şekil 49).

```
leventyildiz:AlBideBurdanYak leventyildiz$ ./gradlew assembleRelease
Starting a Gradle Daemon, 1 busy Daemon could not be reused, use --status for details
leventyildiz:iot-ignite-smart-home leventyildiz$ GIT
BUILD SUCCESSFUL in 28s
27 actionable tasks: 12 executed, 15 up-to-date
leventyildiz:AlBideBurdanYak leventyildiz$
```

Şekil 49: Ağ geçidi – komut satırı üzerinden Android APK oluşturma

İmzalama işleminden sonra oluşan doğrulanabilir APK bulut platformuna yüklenerek Android cihaza gönderilebilir hale getirilmiştir. IGNITE bulut platformuna yükleme esnasında uygulamanın güvenli(trusted) olarak işaretlenmesi sağlanmıştır. Güvenli olarak işaretlenmeyen uygulamalar düğümlerle(node) iletişim kurmayacaktır. Yine doğrulama adımlarından biri de güvenli olarak yüklenen uygulamanın sertifikasının da cihaza gönderilen profile eklenmesi gerekmektedir. Bu işlemler IOT literatüründe büyük sorun teşkil eden güvenlik adımları için gereklidir.

Aşağıdaki şekilde bulut platformuna(<https://www.iot-ignite.com>) eklenen uygulamanın yüklemeden önce güvenli olarak işaretlenmesi görülmektedir(Şekil 50).



The screenshot shows the 'Fields' section of the IOT Ignite web interface. It includes a 'Category' dropdown set to 'DEFAULT', a 'File' field with the filename 'DynamicNodeExample_v0.9.62-20190516-R.apk' and a 'Change' button, a 'Use External Storage' toggle, an 'External URL' field, a 'Gateway Model' dropdown set to 'All', a 'Trusted App' checkbox which is checked, and two sub-checkboxes 'Start Application' and 'Hidden' both of which are unchecked. A 'Description' field contains the text 'Dynamic Node Android Application'.

Şekil 50: Ağ geçidi – bulut platformuna uygulama yükleme

5.7.SONAR İle Proje Ve Kod Analizi

Çalışmadaki mobil uygulama ayağını SONAR ile analiz edilerek üstte projenin durumu paylaşılmıştır. Bu kısımda ise geliştirilen ağ geçidi uygulamasının geliştirme süreçlerine dahil edilen kod analiz kısmı açıklanacaktır.

SONAR veya benzeri kod analiz araçlarıyla geliştiriciler ve şirketler projeleri tararlar. Bu sayede gözden kaçan ve kodlamanın hem performansını hem okunabilirliğini azaltan durumlardan kaçılmış olunur.

Çalışmanın bu kısmında da kodlamaya başlarken kod kalitesi ve okunabilirlik kriterini göz önünde bulundurarak SONAR analizi işlemi yapılmıştır. İlk olarak SONAR bir Docker container'ı olarak çalıştırıldı. Daha sonra uzak docker container'ı üzerinden proje ve TOKEN oluşturularak projeyi uzak erişime açıldı. SONAR-S properties dosyasına oluşturulan projenin anahtarı(key) ve projenin bulunduğu dizinin yolu yazıldı(Şekil 51).

```
sonar.projectKey=gateway
sonar.projectName=gateway
sonar.projectVersion=1.0
sonar.sources=~/.DEVELOPMENT/GIT/iot-ignite-smart-home/gateway/headlessGateway
```

Şekil 51: Ağ geçidi – properties dosyasında SONAR tarama ayarları

Linux işletim sistemi üzerinde terminal açılıp içinde TOKEN ve proje bilgilerini tutan komut çalıştırılıp projenin analizi komut satırı üzerinden başlatılmıştır. Bu tarama komutunun tamamlanması 2-3 dakika kadar sürebilmektedir(Şekil 52).

```
leventyildiz:headlessGateway leventyildiz$ docker start sonarqube && sleep 60s && sonar-scanner -Dsonar.projectKey=gateway -Dsonar.sources=. -Dsonar.host.url=http://127.0.0.1:9000 -Dsonar.login=b3c5c22bc8b403f1ee30501f403bb8f0f49dc0b7 -Dsonar.java.binaries=.
```

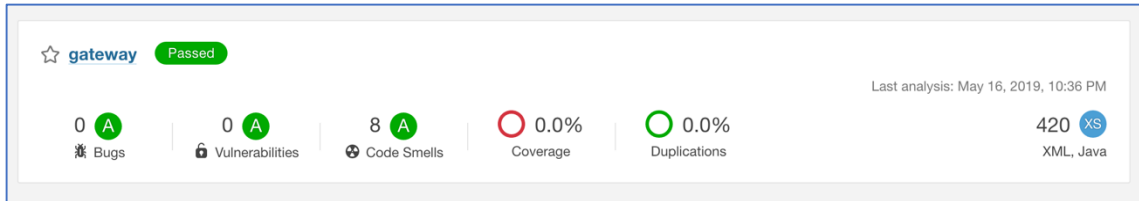
Şekil 52: Ağ geçidi – SONAR tarama komutu

Terminal üzerinden çalıştırılan SONAR-S komutunun başarıyla tamamlanması sonucunda “EXECUTION SUCCESS” mesajı görülmelidir. Eğer tarama esnasında herhangi bir hata olursa ya da Exception atan bir durum oluşursa SONAR ekrana hata mesajını yazmaktadır.

```
INFO: ----- Run sensors on project(done) | time=22ms
INFO: Sensor Zero Coverage Sensor project is: git
INFO: Sensor Zero Coverage Sensor (done) | time=9ms
INFO: Sensor Java CPD Block Indexer
INFO: Sensor Java CPD Block Indexer (done) | time=18ms
INFO: 2 files had no CPD blocks
INFO: Calculating CPD for 2 files
INFO: CPD calculation finished
INFO: Analysis report generated in 78ms, dir size=106 KB
INFO: Analysis report compressed in 53ms, zip size=31 KB
INFO: Analysis report uploaded in 76ms, zip size=32 KB
INFO: ANALYSIS SUCCESSFUL, you can browse http://127.0.0.1:9000/dashboard?id=gateway
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://127.0.0.1:9000/api/ce/task?id=AWrCMC09oa-rZsJYe31Q
INFO: Analysis total time: 7.868 s
INFO: ----- Report processing at http://127.0.0.1:9000/api/ce/task?id=AWrCJWFioa-rZsJYe31P
INFO: EXECUTION SUCCESS time: 9.080 s
INFO: -----
INFO: Total time: 8.883s
INFO: Final Memory: 22M/407M
INFO: -----
```

Şekil 53: Ağ geçidi – başarılı SONAR taraması sonucu

Üstteki aşamalar tamamlandıktan sonra SONAR kendi içinde tuttuğu H2 DB’ de bu bilgileri saklamaktadır. SONAR varsayılan olarak H2 DB kullanmaktadır. Dolayısıyla her kapanıp açılmada veriler sıfırlanmaktadır. Geçmişe yönelik bilgilerin saklanması ve raporlama olanağı için SONAR’ın Oracle, MySQL gibi ilişkisel veri tabanlarıyla bağlantısının yapılması gerekmektedir. Daha sonra web tarayıcı üzerinden <http://127.0.0.1:9000/dashboard?id=gateway> adresine giderek SONAR analiz sonucu gözlemlenmiştir. Bu incelemeler sonucunda SONAR’ın tavsiyelerini dikkate alarak çeşitli kodsız düzenlemeler yapılmıştır(Şekil 54).



Şekil 54: Ağ geçidi – SONAR analizi

5.8. Birim Testleri ve Entegrasyon Testleri

Bu çalışma kapsamında geliştirilen ağ geçidi uygulamasının doğrulamasını birim testleri yaparak gerçekleştirilmiştir. Doğrulama düzeyini artırmak için yazılım projelerinde birim testleriyle birlikte otomasyon ve entegrasyon testleri de yapılmaktadır. Ancak bu çalışma kapsamında birim testlerinin yeterli olacağını kabul ederek sadece birim testlerine yoğunlaşmıştır.

Aşağıdaki şekilde geliştirilen ağ geçidi uygulamasının testi yazılırken birim sınıflarının sanallaştırılması(mocking) yapılmıştır. Sanallaştırma işlemi sonunda her sınıftan ve servisten dönen bilgiler beklenen değerlere çekilebilmektedir(Şekil 55).

Yine Mockito ile birlikte kullanılan JUNIT sayesinde methodların dönen değerleri de doğrulanabilmektedir. Doğrulama aşamasında herhangi bir hata oluşup oluşmadığı da kontrol edilebilmektedir.

Mockito kütüphanesini kapsayan(Encapsule) PowerMock kütüphanesi kullanılarak statik methodlar da sanallaştırılabilir. Yine PowerMock sayesinde private methodlar içinde yaratılan ve servislere parametre olarak geçilen Java Sınıfları da ArgumentCaptor ile yakalanarak private methodlar içerisindeki durumlar(caseler) de doğrulanabilmektedir.

```
@RunWith(MockitoJUnitRunner.class)
public class DiscoveryServiceTest {

    @InjectMocks
    private DiscoveryService service;

    @Mock
    private FromView view;

    private List<Object> esps;

    @Before
    public void setUp() throws Exception {
        esps = new ArrayList<>();
        when(view.getEspList()).thenReturn(esps);
    }
}
```

Şekil 55: Ağ geçidi – mockito ile birim(unit) testi

SONUÇ

Bu çalışmada, nesnelerin interneti ve uçtan uca akıllı ev sistemi yapımı üzerine yürütülen araştırmalardan elde edilen bilgiler derlenmiştir ve çalışmanın sonunda bir bağlı/akıllı ev modeli sunulmuştur.

Araştırma nesnelerin interneti, bulut bilişim, bağlı/akıllı evi temel kavramlar olarak ele almış ve her birini detaylı olarak incelemiştir. Şurası bir gerçektir ki; internet, bireylerin iş yapma, bağlantı kurma ve küresel ölçekte öğrenme şeklini değiştiren dünya çapında bir harikaya dönüşmüştür. İnternet dünya çapında insanların ve toplumların yaşamlarını tamamen değiştirmiştir ve öyle görünüyor ki, bireylerin yaşama biçimini yine yeniden şekillendirmeye devam edecektir. İnsanlığın ve nesnelerin birbirleriyle yeni iletişim biçimlerinin ve etkileşimin her yerde bulunacağı yeni bir çağa girilmektedir. Nesnelerin internetindeki her şey, bir ağ üzerinden veri iletebilen ve alabilen yerleşik bir bilgiişlem cihazı olarak tanımlanabilir. Bilgi işlem cihazını internete bağlamanın temel amaçlarından bir tanesi, teknolojinin onlar için çalışmasını sağlayarak insanların hayatlarını kolaylaştırmaktır.

Bizi bu amaca ulaştırma yollarından en belirginini buluta sürekli bağlı olan akıllı evlerdir. Akıllı Ev, kullanıcıların evlerini farklı şekillerde uzaktan kontrol edebildiği Nesnelerin İnternetine doğru bir adımdır. Örneğin, eve gelmeden önce oda sıcaklığını azaltmak için bir Akıllı telefon kullanarak klimalar internet üzerinden uzaktan açılabilir. İnternet üzerinden uzaktan kontrol edilebilen akıllı ev, işleri kolaylaştırır ve kullanıcılar için büyük kolaylık sağlar. Ayrıca, bu tür sistemler kullanıcılar için güvenlik ve gizlilik sağlar. Bulut bilişim sağlayıcıları tarafından sağlanan hizmetlerin yükselmesiyle bu tür uygulamaların geliştirilmesi daha kolay hale getirilmiştir.

Bulut bilişim kullanıcıların bir bulut havuzunun (örneğin, işlemciler, hizmetler, depolar) paylaşılan havuzunu isteğe bağlı bir şekilde esnek bir şekilde kullanmalarını sağlayan yeni bir bilgi işlem türüdür. Bulut bilişimin amacı, kullanıcıların her teknoloji ile ilgili derin bilgiye veya uzmanlığa ihtiyaç duymadan fayda elde etmelerine izin vermektir. Bulut, maliyetleri azaltmayı ve kullanıcıların bilişim teknolojileri alanındaki isteklerinin engellenmemesini sağlayarak ana işlerine odaklanmalarına yardımcı olmayı

amaçlamaktadır. Bulut bilişim, kullanıcılara hizmet anlamında kaynak sağlar ve standartlaştırılmış bir şekilde bulut hizmetlerine küresel ve kolay erişim sağlamak için belirlenen standart ve uygulamaları kullanır.

İlgili çalışma takip edildiğinde bulut bilişim ve nesnelerin internetinin başlangıcı birbirinden bağımsız olduklarını fakat zaman içerisinde koşulların gerektirmesiyle birlikte geliştikleri görülmektedir. Nesnelerin internetinin, teknolojik kısıtlamalarını (örneğin işleme, depolama) aşmak için bulutun neredeyse sınırsız yeteneklerinden ve kaynaklarından yararlanabilir. Özellikle, bulut bilişim nesnelerin internetine ilişkin servis yönetimini uygulamak ve bu şeyler tarafından üretilen çok büyük miktarda veri için depolama alanı sağlamak için güçlü bir çözüm sunabilir. Ayrıca bulut, gerçek dünyadaki şeyleri daha dinamik ve dağıtılmış bir şekilde ele almak için kapsamını genişleterek nesnelerin internetinde de faydalanabilir. Böylelikle iki yönlü sürekli bir etkileşimden söz edilebilir.

Bağlı-akıllı evler konusunda her ne kadar iyi niyetli araştırmalar ve destekler devam ediyor olsa da ilgili sektör henüz ortak kararlarla edilmiş endüstri standartlarına sahip değildir ve hali hazırda ürünler arasında da iletişim ve uyumluluk konusunda tam anlamıyla birlik sağlanabildiğini söylemek olanaklı değildir.

Nesnelerin internetinin geliştirilmesi yeni kavramların ortaya çıkmasına bağlı-akıllı evler konusunda yeni alanlar oluşmasına katkı sunmaktadır. Yanı sıra pek çok ev sakini nesnelerin internetini bir tür önemli uygulama olarak görmektedir. Nesnelerin interneti konusundaki çalışmaları internete bağlanma, ev aletlerini izleme ve yönetim gibi alanlarda gelişmeleri hızlandırmaktadır ve ev halkına bu konularda faydalar sunmaktadır.

Nesnelerin interneti geniş bir uygulama ve ekipman alanında ürün desteğini ve entegrasyonu gerektiren, birleştirilmiş standartların güçlendirilmesini devam ettiren bir anlayışı gerektirmektedir. Bu konuda gelecekte geniş çaplı bir gelişim yaşamak için, konu ile ilgili olan bağlı-akıllı ev sektöründeki işletmelerin, aile sigorta firmalarının, finans ve ilgili diğer sektörlerin ekipman ve yöntemlerin iyileştirilmesi ve entegrasyonu konusunda çözümler geliştirmeleri gerekir.

Nesnelerin interneti araçları yalnızca ev aletleri arasında bir terminal görevi görmez, yanı sıra her türlü veriyi toplama, güçlü bir bilgi işleme ve kontrol yeteneği de sunar. Bağlı-akıllı ev, televizyon setleri, buzdolapları, klimalar, su ısıtıcılar, aydınlatma gibi elektrikli ev aletlerinin kontrolünün yanında başka işlevlere de sahip olabilirler. Örnek olarak, akıllı tasarımları nedeniyle birer aile veri merkezi, aile güvenlik merkezi, aile eğlence merkezi ve aile iş merkezi olarak işlev görebilir.

Bugün akıllı telefonlara, tabletlere ve benzeri teknolojik cihazlara sahip olan kişi sayısı muazzam bir şekilde artmaktadır. Çoğu akıllı telefon internete erişebilir, ofis işlerinin üstesinden gelebilir ve burada sayılamayacak kadar çok başka görevleri de yerine getirebilir. Ayrıca 2012'den üretilen akıllı telefonlar da yüksek hızlı mobil geniş bantlı 4G LTE İnternete sahiptir. Bu tür teknolojilerin icadı ile, önümüzdeki yıllarda, nüfusun çoğunun neredeyse her zaman internete bağlanacağı söylenebilir. Her zaman, her yerde internet kullanımı için talep, daha yüksek bant genişliği teknolojilerinin geliştirilmesine yol açacaktır. Dolayısıyla, bulut bilişim ve mobil cihazların entegrasyonu, bu nedenle, herhangi bir sistemi uzaktan kontrol etmek için kolay ve kullanışlı alanlara yol açacaktır. Tüm bunlar bağlı/akıllı evlerin konsept olmaktan gerçekliğe doğru yol almasına destek olmaktadır. Bu konuda henüz Türkiye'de büyük bir orandan bahsedilemese de halkımızın teknoloji merakı ve çeşitli teknolojik cihazları kullanım oranları bağlı/akıllı evlerin ülkemizdeki geleceği açısından ümit vermektedir.

Araştırmada ulaşılan kaynakların çeşitliliği, nesnelerin interneti, bulut bilişim, büyük verinin işlenmesi ve tüm bunların bağlı/akıllı ev tasarımları bağlamında ele alınması ve bir bağlı/akıllı ev modeli sunulması araştırmayı ilgili çalışma için önemli ve değerli kılmaktadır. Sonuç olarak şöyle bir öngöründe bulunabiliriz: Bağlı-akıllı evler, birbirine bağlı akıllı şehirlerin oluşumuna katkı sağlayacaktır.

KAYNAKLAR

Aggarwal, B., Bhagwan, R., Das, T., Eswaran, S., Padmanabhan, V. N., And Voelker, G. M. (2009). NetPrints: diagnosing home network misconfigurations using shared knowledge. In Proceedings of the 6th USENIX symposium on Networked systems design and implementation.

Alam, M. R. ve diğeri (2012). "SPEED: An Inhabitant Activity Prediction Algorithm for Smart Homes," IEEE Trans. on Systems, Man and Cybernetics, vol. 42, pp. 985–990

Baker, C.R., Markovsky, Y., Greunen, J.V., Rabaey, J., Wawrzynek, J., and Wolisz, A. (2007). ZUMA: A Platform for Smart-Home Environments. University of California, Berkeley, USA

Ballari, D, M. Wachowicz, M. A. Manso (2009). Metadata behind the interoperability of wireless sensor network, Sensors 9 (2009) 3635–3651.

Barga, R. (2009). Cloud computing architecture and application programming. ACM SIGACT News, 40(2), p.94.

Botta, A., de Donato, W., Persico, V., Pescapé, A., 2016. Integration of cloud computing and internet of things: a survey. Future Gener. Comput. Syst. 56, 684–700.

Brush, A., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., And Dixon, C. (2011). Home Automation in the Wild: Challenges and Opportunities. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)

Byun, J, S. Park, B. Kang, I. Hong, and S. Park (2013). "Design and implementation of an intelligent energy saving system based on standby power reduction for a future zero-energy home environment," IEEE Trans. Consumer Electron., vol. 59, no. 3, pp. 507-514

Byun, J., Jeon, B., Noh, J., Kim, Y., Park, S., 2012. An intelligent self-adjusting sensor for smart home services based on ZigBee communications. IEEE Trans. Consum. Electron. 58, 794–802.

Cabarcos, PA, F. A. Mendoza, R. S. Guerrero, A. M. Lopez, and D. Diaz-Sanchez (2012). "SuSSo: Seamless and ubiquitous single sign-on for cloud service continuity across devices," *IEEE Trans. Consumer Electron.*, vol. 58, no. 4, pp. 1425-1433

Calvert, K. L., Keith, W., Rebecca, E., And Grinter, (2007). E. Moving Toward the Middle: The Case Against the End-to-End Argument. In Sixth Workshop on Hot Topics in Networks

Cardenas, J.A., Gemoets, L., Rosas, J.H.A., Sarfi, R., 2014. A literature survey on smart grid distribution: an analytical approach. *J. Clean. Prod.* 65, 202–216.

Chang F., Dean J., Ghemawat S., Hsieh W. C., Wallach D. A., Burrows M., Chandra T., Fikes A. and Gruber R. (2006). Bigtable: a distributed storage system for structured data. In OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation

Chen, L, C. D. Nugent and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *IEEE Trans. on Knowledge and Data Eng.*, pp. 961–974. Euler Proof Mechanism. <http://www.agfa.com/w3c/euler/>

Chen, M., Wan, J., Li, F., 2012. Machine-to-machine communications: architectures, standards and applications. *KSII Trans. Internet Inf. Syst.* TIIS 6, 480–497.

Chetty, M., Banks, R., Harper, R., Regan, T., Sellen, A., Gkantsidis, C., Karagiannis, T., And Key, P. Who's hogging the bandwidth: the consequences of revealing the invisible in the home. In Proceedings of the 28th international conference on Human factors in computing systems (2010), CHI '10.

Ciau, M., Acevedo, J., Poot, R., Chi, V., Narváez, L., and Lianes, E. (2010). Design and Development of a Prototype for Illuminating a Smart Home using a Client-Server Application. *IJCSNS*, 10(10).

Costa, LCP, N. S. Almeida, A. G. D. Correa, R. D. Lopes, and M. K. Zuffo (2013). "Accessible display design to control home area networks," *IEEE Trans. Consumer Electron.*, vol. 59, no. 2, pp. 422-427

Da Xu, L., He, W., Li, S., 2014. Internet of things in industries: a survey. *IEEE Trans. Ind. Inf.* 10, 2233–2243.

- Darbee, P., 2013. INSTEON: Compared
- Dean J. ve Ghemawat S., (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51, 107–113.
- Diaz, M, G. Juan, O. Lucas, A. Ryuga (2012). Big data on the internet of things: An example for the e-health, in: *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012)*, 2012, pp. 898–900.
- Digolo, B.A., Andang'o, E.A., Katuli, J. (2011). E-learning as a strategy for enhancing access to music education. *Int. J. Bus. Soc. Sci.* 2(11), 135–139
- Dimitrievski, A., Stojkoska, B., Trivodaliev, K., Davcev, D., 2006. *Securing Communication in WSN Trough Use of Cryptography*. NATO-ARW, Suceava.
- Ding, Y, M. Neumann, D. Gordon, T. Riedel, T. Miyaki, M. Beigl, W. Zhang, L. Zhang (2012). A platform-as-a-service for in-situ development of wireless sensor network applications, in: *Networked Sensing Systems (INSS)*, 2012 Ninth International Conference on, 2012, pp. 1–8.
- Dixon, C., Mahajan, R., Agarwal, S., Brush, A. J., Lee, B., Saroiu, S., And Bahl, V. (2010). The home needs an operating system (and an app store). In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks (2010)*, Hotnets '10.
- Dobre, C., Xhafa, F., 2014. Intelligent services for big data science. *Future Gener. Comput. Syst.* 37, 267–281. EcoGrid EU, <http://www.eu-ecogrid.net/>
- Erden, F and A. Çetin (2014). "Hand gesture based remote control system using infrared sensors and a camera," *IEEE Trans. Consumer Electron.*, vol. 60, no. 4, pp. 675-680
- Erol-Kantarci, M., Mouftah, H.T., 2010. Wireless sensor networks for domestic energy management in smart grids. In: *Communications (QBSC), 2010 25th Biennial Symposium on. IEEE*, pp. 63–66.
- Fan, Z., Kalogridis, G., Efthymiou, C., Sooriyabandara, M., Serizawa, M., McGeehan, J., 2010. The new frontier of communications research: smart grid and smart metering. In: *Proceedings of the 1st International Conference on Energy-efficient Computing and Networking*. ACM, pp. 115–118.

Fazio, M, A. Celesti, A. Puliafito, M. Villari (2015). Big Data Storage in ent Monitoringthe Cloud for Smart Environm. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015). *Procedia Computer Science* 52 (2015) 500-506

Fazio, M, M. Paone, A. Puliafito, M. Villari (2012). Huge amount of heterogeneous sensed data needs the cloud, in: *International Multi-Conference on Systems, Signals and Devices (SSD 2012)*, Chemnitz, Germany, 2012.

Feamster, N. (2010). Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks, HomeNets '10*.

Feldman B, Martin EM, Skotnes T. Big Data in Healthcare Hype and Hope. Retrieved from <http://www.west-info.eu/files/big-data-in-healthcare.pdf> [Erişim Tarihi: 22.09.2018]

Finn, P., Fitzpatrick, C., 2014. Demand side management of industrial electricity consumption: promoting the use of renewable energy through real-time pricing. *Appl. Energy* 113, 11–21.

Groves P, Kayyali B, Knott D, Van Kuiken S. (2013). The “big data” revolution in healthcare. Accelerating value and innovation. Center for US Health system Reform. Business Technology Office

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of Things (IOT): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* 29, 1645–1660.

Gyarmati, L., And Trinh, T. A. (2010). How can architecture help to reduce energy consumption in data center networking? In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (2010)*, e-Energy '10.

Han, D.-M. ve J.-H. Lim (2010). "Design and implementation of smart home energy management systems based on zigbee," *IEEE Trans. on Consumer Electronics*, vol. 56, pp. 1417–1425

He, D, N. Kumar, and J. H. Lee (2015). "Secure pseudonym-based near field communication protocol for the consumer internet of things," *IEEE Trans. Consumer Electron.*, vol. 61, no. 1, pp. 56-62

Heile, B., 2010. Smart grids for green communications [Industry Perspectives]. IEEE Wirel. Commun. 17, 4–6.

<http://cassandra.apache.org/> [Erişim Tarihi: 22.08.2018]

<http://devveri.com>, Erişim Tarihi: 10.05.2015.

<http://hadoop.apache.org> [Erişim Tarihi: 25.08.2018]

https://en.wikipedia.org/wiki/Big_data [Erişim Tarihi: 14.09.2018]

https://upload.wikimedia.org/wikipedia/commons/b/b5/Cloud_computing.svg
[Erişim Tarihi: 14.09.2018]

Hu, Qinran, ve Fangxing Li. (2013). "Hardware design of smart home energy management system with dynamic price response." Smart Grid, IEEE Transactions on 4.4 (2013): 1878-1887.

Hu, Qinran, ve Fangxing Li. (2013). "Hardware design of smart home energy management system with dynamic price response." Smart Grid, IEEE Transactions on 4.4 (2013): 1878-1887.

Huynh, S., D. Parry, A. C. M. Fong, and J. Tang (2014). "Novel RFID and ontology-based home localization system for misplaced objects," IEEE Trans. Consumer Electron., vol. 60, no. 3, pp. 402-410

J. Han, C. S. Choi, W. K. Park, I. Lee, and S. H. Kim (2014a). "Smart home energy management system including renewable energy based on ZigBee and PLC," IEEE Trans. Consumer Electron., vol. 60, no. 2, pp. 198-202

J. Han, C. S. Choi, W. K. Park, I. Lee, and S. H. Kim (2014b). "PLC-based photovoltaic system management for smart home energy management system," IEEE Trans. Consumer Electron., vol. 60, no. 2, pp. 184-189

Jiang, L, L. D. Xu, H. Cai, Z. Jiang, F. Bu, B. Xu (2014). An iot-oriented data storage framework in cloud computing platform, Industrial Informatics, IEEE Transactions on 10 (2) (2014) 1443–1451.

Jo, HC, S. Kim, and S. K. Joo (2013). "Smart heating and air conditioning scheduling method incorporating customer convenience for home energy management system," IEEE Trans. Consumer Electron., vol. 59, no. 2, pp. 316-322.

Jo, J., S. Lee, and J. Kim (2014). "Software-defined home networking devices for multi-home visual sharing," IEEE Trans. Consumer Electron., vol. 60, no. 3, pp. 534-539.

Kamilaris, A., Trifa, V., Pitsillides, A., 2011. HomeWeb: an application framework for Web-based smart homes. In: Telecommunications (ICT), 2011 18th International Conference on. IEEE, pp. 134–139.

Karnouskos, S., 2011. Crowdsourcing information via mobile devices as a migration enabler towards the smartgrid. In: Smart Grid Communications (SmartGrid- Comm), 2011 IEEE International Conference on. IEEE, pp. 67–72.

Katz, R. (2009). Tech Titans Building Boom. IEEE Spectrum 46 (February 2009).

Khan, M., Javaid, N., Arif, M., Saud, S., Qasim, U., Khan, Z., 2014. Peak load scheduling in smart grid communication environment. In: 28th International Conference on Advanced Information Networking and Applications. IEEE, pp. 1025–1032.

Kim, D., S. Ahn, S. Park, and M. Whang (2013). "Interactive emotional lighting system using physiological signals," IEEE Trans. Consumer Electron., vol. 59, no. 4, pp. 765-771

Kim, T., H. Park, S. H. Hong, and Y. Chung (2013). "Integrated system of face recognition and sound localization for a smart door phone," IEEE Trans. Consumer Electron., vol. 59, no. 3, pp. 598-603

Ko, J., Terzis, A., Dawson-Haggerty, S., Culler, D.E., Hui, J.W., Levis, P., 2011. Connecting low-power and lossy networks to the internet. IEEE Commun. Mag. 49, 96–101.

Krishna M., Kannan B., Ramani A., Sathish S. J., (2010). Implementation and Performance Evaluation of a HYBRID Distributed System for Storing and Processing Images from the Web. Cloudcom, 2010 IEEE Second International Conference on Cloud Computing Technology and Science, 762-767, 2010

Lee, B, J. Byun, M. I. Choi, B. Kang, and S. Park (2014). "Degradation diagnosis system of photovoltaic panels with mobile application," IEEE Trans. Consumer Electron., vol. 60, no. 3, pp. 338-346

Lee, I., Lee, K., 2015. The Internet of Things (IOT): applications, investments, and challenges for enterprises. *Bus. Horiz.* 58, 431–440.

Liu X., Han J., Zhong Y., Han C. and He X. (2009). “Implementing WebGIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS,” *Proc. of the 2009 IEEE Conf.on Cluster Computing*, DOI: 10.1109/CLUSTR.2009.5289196.

Liu, C., Yang, C., Zhang, X., Chen, J., 2015. External integrity verification for outsourced big data in cloud and IOT: a big picture. *Future Gener. Comput. Syst.* 49, 58–67.

Liu, C.H., Yang, B., Liu, T., 2014. Efficient naming, addressing and profile services in Internet-of-Things sensory environments. *Ad Hoc Netw.* 18, 85–101.

Liu, Q, G. Cooper, N. Linge, H. Takruri, and R. Sowden (2013). "DEHEMS: Creating a digital environment for large-scale energy management at homes," *IEEE Trans. Consumer Electron.*, vol. 59, no. 1, pp. 62-69

Lu, C.-W., Li, S.-C., Wu, Q., 2011. Interconnecting ZigBee and 6LoWPAN wireless sensor networks for smart grid applications. In: *Sensing Technology (ICST), 2011 Fifth International Conference on*. IEEE, pp. 267–272.

Lund, P.D., Mikkola, J., Ypyä, J., 2015. Smart energy system design for large clean power schemes in urban areas. *J. Clean. Prod.* 103, 437–445.

Mainetti, L., Patrono, L., Vilei, A., 2011. Evolution of wireless sensor networks towards the internet of things: a survey. In: *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*. IEEE, pp. 1–6.

Mckeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., And Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* 38

McKinsey Global Institute. Big Data: The next frontier for innovation, competition, and productivity; 2011. Available from: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation [Erişim Tarihi: 22.09.2018]

Mearian, L (2011). What happens to data when your cloud provider evaporates? [online] Available at: http://www.computerworld.com/s/article/9216159/What_happens_to_data_when_your_cloud_provider_evaporates_ [Erişim Tarihi: 22 Eylül, 2018]

Merabti, M., (2006). Networked appliances in home entertainment. Proceedings of the 2006 international conference on, pp.288-293.

Meraki MX Cloud Managed Routers. http://meraki.com/products_services/mx/.

Misra, P., Rajaraman, V., Dhotrad, K., Warrior, J., Simmhan, Y., 2015. An Interoperable Realization of Smart Cities with Plug and Play Based Device Management. arXiv preprint arXiv:1503.00923.

Ning, H., Liu, H., Yang, L.T., 2013. Cyberentity security in the internet of things. Computer 46, 46–53.

Palensky, P., Dietrich, D., 2011. Demand side management: demand response, intelligent energy systems, and smart loads. IEEE Trans. Ind. Inf. 7, 381–388.

Panna, R., R. Thesrumluk, ve C. Chantrapornchai. (2013). "Development of the Energy Saving Smart Home Prototype." International Journal of Smart Home

Panna, R., R. Thesrumluk, ve C. Chantrapornchai. (2013). "Development of the Energy Saving Smart Home Prototype." International Journal of Smart Home

Perumal, T., Ramli, A.R., Leong, C.Y., and Mansor, S. (2008). Interoperability for Smart Home Environment Using Web Services. International Journal of Smart Home, 2(4), p.1–16.

Pham, H., Paluska, J. M., And Ward, S. (2011). User interface models for the cloud. In ACM CHI Workshop on Designing Interaction for the Cloud, CHI '11.

Portet, François ve diğerleri. (2013). "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects." Personal and Ubiquitous Computing 17.1 (2013): 127-144.

Portet, François, ve diğerleri. (2013). "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects." *Personal and Ubiquitous Computing* 17.1 (2013): 127-144.

Puustjärvi, J ve Puustjärvi, L 2015, ' The Role of Smart Data in Smart Home : Health Monitoring Case ' *Procedia Computer Science*, vol. 69, pp. 143-151. DOI: 10.1016/j.procs.2015.10.015

Puustjärvi, J ve Puustjärvi, L 2015, ' The Role of Smart Data in Smart Home : Health Monitoring Case ' *Procedia Computer Science*, vol. 69, pp. 143-151. DOI: 10.1016/j.procs.2015.10.015

Qiao, G, G. Song, Y. Wang, J. Zhang, and W. Wang (2013). "Autonomous network repairing of a home security system using modular selfreconfigurable robots," *IEEE Trans. Consumer Electron.*, vol. 59, no. 3, pp. 562-570

Qiu J., Zheng Q., Zhong X., Li J., Li Y. Dong B. (2010). "A Novel Approach to Improving the Efficiency of Storing and Accessing Small Files on Hadoop: A Case Study by PowerPoint Files," , 65-72, 2010.," in *Services Computing (SCC), 2010 IEEE International Conference*

Qiu X., Ekanayake J., Beason S., Gunarathne T., Fox G., Barga R. and Gannon D.(2009). "Cloud technologies for bioinformatics applications," in *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers. ACM*, 1-10

Qureshi, A., Weber, R., Balakrishnan, H., Gutttag, J., And Maggs, B. (2009). Cutting the electric bill for internet-scale systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication (2009), SIGCOMM '09*.

Rao, S, R. Ramakrishnan, A. Silberstein, M. Ovsiannikov, D. Reeves (2012). Sailfish: A framework for large scale data processing, in: *Proceedings of the Third ACM Symposium on Cloud Computing, SoCC '12, ACM, New York, NY, USA, 2012, pp. 4:1–4:14*.

Reed, C., M. Botts, J. Davidson, G. Percivall (2007). *OGC Sensor Web Enablement: Overview and High-Level Architecture*, *IEEE Autotestcon (2007)* 372–380.

Rezvani, A., Gandomkar, M., Izadbakhsh, M., Ahmadi, A., 2015. Environmental/economic scheduling of a micro-grid with renewable energy resources. *J. Clean. Prod.* 87, 216–226.

Risteska Stojkoska, B., Popovska Avramova, A., Chatzimisios, P., 2014. Application of wireless sensor networks for indoor temperature regulation. *Int. J. Distrib. Sens. Netw.* 2014.

S. Kong, Y. Kim, R. Ko, and S. K. Joo, "Home appliance load disaggregation using cepstrum-smoothing-based method," *IEEE Trans. Consumer Electron.*, vol. 61, no. 1, pp. 24-30, Feb. 2015.

Sajjad, I.A., Napoli, R., Chicco, G., 2014. Future business model for cellular microgrids. In: 4th International Symposium on Business Modeling and Software Design (BMSD). pp. 209–216.

Santini, S., Romer, K., 2006. An adaptive strategy for quality-based data reduction in wireless sensor networks. In: Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006). pp. 29–36.

Schneps-Schneppe, M., Maximenko, A., Namiot, D., Malov, D., 2012. Wired Smart Home: energy metering, security, and emergency issues. In: Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on. IEEE, pp. 405–410.

Shah, SH, F. K. Khan, W. Ali, J. Khan (2013). A new framework to integrate wireless sensor networks with cloud computing, in: Aerospace Conference, 2013 IEEE, 2013, pp. 1–6.

Shehab, A., Elhoseny, M., Hassanien, A. (2016). A HYBRID scheme for automated essay grading based on LVQ and NLP techniques. In: Proceedings of 12th International Computer Engineering Conference (ICENCO), pp. 65–70. IEEE (2016). <https://doi.org/10.1109/ICENCO>.

Shrouf, F., Miragliotta, G., 2015. Energy management based on Internet of Things: practices and framework for adoption in production management. *J. Clean. Prod.* 100, 235–246.

Siano, P., 2014. Demand response and smart grids—A survey. *Renew. Sustain. Energy Rev.* 30, 461–478.

Solimani, Moataz, Tobi Abiodun, Tarek Hamouda, Jiehan Zhou, Chung-Horng Lung (2013). IEEE International Conference on Cloud Computing Technology and Science. udCom.2013.155DOI 10.1109/Clo

Son J.-Y. ve diğerleri (2011). "Resource-aware smart home management system by constructing resource relation graph," IEEE Trans. On Consumer Electronics, vol. 57, pp. 1112–1119

Stojkoska, B.R., Solev, D., Davcev, D., 2012. Variable step size LMS Algorithm for Data Prediction in wireless sensor networks. Sens. Transducers 14, 111.

The IBM vision of a smarter home enabled by cloud technology. White paper, IBM Sales & Distribution, January 2010

Tian, J., W. Wang, X. Gong, X. Que, and J. Ma (2013). "An enhanced personal photo recommendation system by fusing contextual and textual features on mobile device," IEEE Trans. Consumer Electron., vol. 59, no. 1, pp. 220-228

Tsai, CH, Y. W. Bai, M. B. Lin, J. Rong, and Y. W. Lin (2013). "Design and implementation of a PIR luminaire with zero standby power using a photovoltaic array in enough daylight," IEEE Trans. Consumer Electron., vol. 59, no. 3, pp. 499-506

Tsai, Pohsiang, ve Bojun Pan. (2015). "Intelligent electricity bill saving system for smart home." Fuzzy Theory and Its Applications (iFUZZY), 2015 International Conference on. IEEE, 2015.

Tsai, Pohsiang, ve Bojun Pan. (2015). "Intelligent electricity bill saving system for smart home." Fuzzy Theory and Its Applications (iFUZZY), 2015 International Conference on. IEEE, 2015.

Tung, HY, K. F. Tsang, H. C. Tung, K. T. Chui, and H. R. Chi (2013). "The design of dual radio ZigBee homecare gateway for remote patient monitoring," IEEE Trans. Consumer Electron., vol. 59, no. 4, pp. 756-764,

Ullah, M., Mahmood, A., Razzaq, S., Ilahi, M., Khan, R., Javaid, N., 2013. A Survey of Different Residential Energy Consumption Controlling Techniques for Autonomous DSM in Future Smart Grid Communications. arXiv preprint arXiv:1306.1134.

Viani, F., Robol, F., Polo, A., Rocca, P., Oliveri, G., Massa, A., 2013. Wireless architectures for heterogeneous sensing in smart home applications: concepts and real implementation. Proc. IEEE 101, 2381–2396.

Vimarlund, V., Wass, S. (2014). Big Data, Smart Homes and Ambient Assisted Living. IMIA Yearbook of Medical Informatics. <http://dx.doi.org/10.15265/IY-2014-0011>

Wang, J., Z. Zhang, B. Li, S. Lee, and R. Sherratt (2014). "An enhanced fall detection system for elderly person monitoring using consumer home networks," IEEE Trans. Consumer Electron., vol. 60, no. 1, pp. 23-29

West, D. (2010). Saving Money Through Cloud Computing. Brookings Institution, 2010.

Wi, YM, J. U. Lee, and S. K. Joo (2013). "Electric vehicle charging method for smart homes/buildings with a photovoltaic system," IEEE Trans. Consumer Electron., vol. 59, no. 2, pp. 323-328.

Wu, C.-L. ve L.-C. Fu (2012). "Design and Realization of a Framework for Human–System Interaction in Smart Homes," IEEE Trans. on Systems, Man and Cybernetics, vol. 42, pp. 15–31.

Xu, B., Da Xu, L., Cai, H., Xie, C., Hu, J., Bu, F., 2014. Ubiquitous data accessing method in IOT-based information system for emergency medical services. IEEE Trans. Ind. Inf. 10, 1578–1586.

Ya Liu and Chun Yang (2017). Analysis on the Development of Modern Smart Home Industry and Cloud Computing Technique with the Influence on Transforming the Internet Startup Orientation. International Journal of Multimedia and Ubiquitous Engineering Vol.12, No.1. pp.43-56. <http://dx.doi.org/10.14257/ijmue.2017.12.1.04>.

Ying-Tsung Lee, Wei-Hsuan Hsiao, Chin-Meng Huang and Seng-Cho T. Chou (2016). An Integrated Cloud-Based Smart Home Management System with Community Hierarchy. IEEE Transactions on Consumer Electronics Vol. 62, No. 1,

Zaslavsky, A., Perera, C., Georgakopoulos, D., 2013. Sensing as a Service and Big Data. arXiv preprint arXiv:1301.0159.

Zhou, J., Leppänen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., Jin, H., 2013. Cloudthings: a common architecture for integrating the internet of things with cloud computing. In: Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on. IEEE, pp. 651–657.

Zhou, J., T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, L. Yang (2013). Cloudthings: A common architecture for integrating the internet of things with cloud computing, in: Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on, 2013, pp. 651–657.

Zhu, Q., Wang, R., Chen, Q., Liu, Y., Qin, W., 2010. Iot gateway: bridging wireless sensor networks into internet of things. In: Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. IEEE, pp. 347–352.

ZigBee, <http://www.zigbee.org/> [Erişim Tarihi: 22.09.2018]

Z-Wave Alliance, <http://products.z-wavealliance.org/> [Erişim Tarihi: 22.09.2018]

EKLER

Bu bölümde çalışmada geliştirilen mobil uygulama ve diğer katmanlardan özet şeklinde kod blokları paylaşılmıştır.

EK A

Aşağıdaki kod örneğinde ES6 ile bulut platformu üzerinde kimlik doğrulaması yapılmaktadır. Kullanıcı adı ve Şifre base64 encoded olarak buluta gönderilir ve buradan Access_TOKEN alınarak sonraki bulut isteklerinde kullanılır (Şekil 56).

```
export const loginToCloud = (username, password) => {
  return new Promise((resolve, reject) => {
    ajax.doFetch("/login/oauth", {
      grant_type: "password",
      username: username,
      password: password
    })
    .header({
      'Authorization': 'Basic ' + encodeAuth(username, password)
    })
    .formData()
    .then(r => {
      LOG("Api Login Response ").info(r);
      if (r && r.access_token) {
        resolve(r.access_token);
      }
      reject(r);
    });
  });
}
```

Şekil 56: ES6 ve FETCH API ile bulut' a giriş yapma

EK B

Aşağıdaki şekilde RN ile RAPI kullanımı gösterilmektedir. Geliştirilen çalışmada state karmaşasını çözebilmek ve stateleri yönetebilmek için RAPI tercih edilmiştir. Aslında RDX kullanımı da mümkündür ancak kolay implementasyonu ve güncel teknoloji olması sebebiyle tercih RAPI dan yana kullanılmıştır(Şekil 57).

```

render() {
  return (
    <AppProvider>
      <View style={{s.container, {borderColor: 'black', top: 20}}>
        <CtxConsumer>
          {(context) => {
            return (
              <Drawer
                openDrawerOffset={0.07}
                open={context.sideBarOpen}
                content={<SideBar
                  changeScreenByType={(d) => this.changeScreenByType(d)}>
              >
                {this.loadMessageBar(context.toast)}
                {this.loadActivityIndicator(context.loading)}
                <Text style={{color: '#FFF'}}> Token : {context.token}</Text>
                {this.loadScreenByType(context.screenType)}
              </Drawer>
            )//return
          }}//context
        </CtxConsumer>
      </View>
    </AppProvider>
  ); //return
} //render

```

Şekil 57: Üretici(provider), tüketici(consumer) ve RCAPİ kullanımı

EK C

Aşağıdaki şekilde de çalışmaya eklenen çoklu dil desteğinin tanımlamasının nasıl yapılacağını görülmektedir. Çalışmada N tane çoklu dil dosyası eklemek mümkündür. Tr.JSON yada En.JSON şeklinde tanımlanan dosyalara ilgili etiketler karşılığında dile göre açıklama yazılabilmektedir(Şekil 58).

```
{
  "name": "Name",
  "input": {
    "email": "E-Mail",
    "password": "Password"
  },
  "button": {
    "login": "Login",
    "signup": "SignUp"
  },
  "screen": {
    "login": "Login Screen"
  },
  "menu": {
    "dashboard": "Dashboard",
    "actionLog": "Action Log"
  }
}
```

Şekil 58: JSON formatında çoklu dil desteği İngilizce olarak kullanımı

EK D

Aşağıdaki şekilde IGNITE kütüphanelerine parametre olarak verilen pin numarası, sensör tipi , üretici bilgileri, veri tipi ve zamanlayıcıyla eklenen veri okuma methoduyla MQ6 gaz sensörünün verileri alınabilir(Şekil 59).

```
addThingToInventory(SENSOR_MQ6_GAS, TYPE_MQ6_GAS,
  PIN_DATA_MQ6_GAS_SENSOR, NOT_ACTUATOR, VENDOR_MQ6_GAS,
  DATA_TYPE_INTEGER,
  new IgniteEsp8266Timer(readMq6Gas));
```

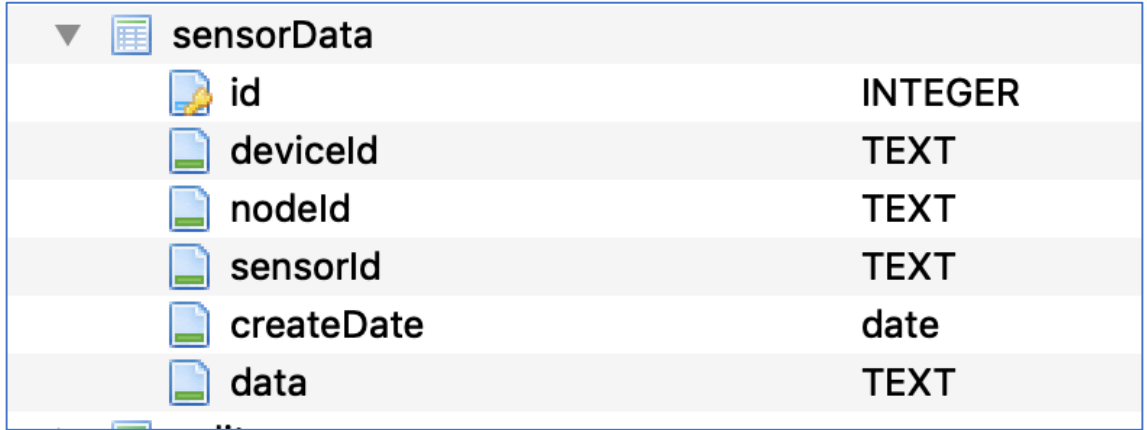
Şekil 59: IGNITE ile mq-6 gaz sensörünün düğüme tanıtılması

EK E

Aşağıdaki şekilde de çalışma kapsamında geliştirilen mobil uygulamanın DB tablolarından bir örnek gösterilmektedir. Bu tabloda bulut platformundan gelen bilgiler arşivlenmektedir. Projenin ilerleyen günlerde çevrim dışı çalışmasına da ihtiyaç

duyulabileceği düşüncesiyle API den gelen sonuçlar anlamlı halde lokal DB’de da tutulmaktadır(Şekil 60).

Diğer bir avantajı ise özetlerin ve grafik bilgilerinin DB aracılığıyla oluşturulması büyük kolaylıktır.

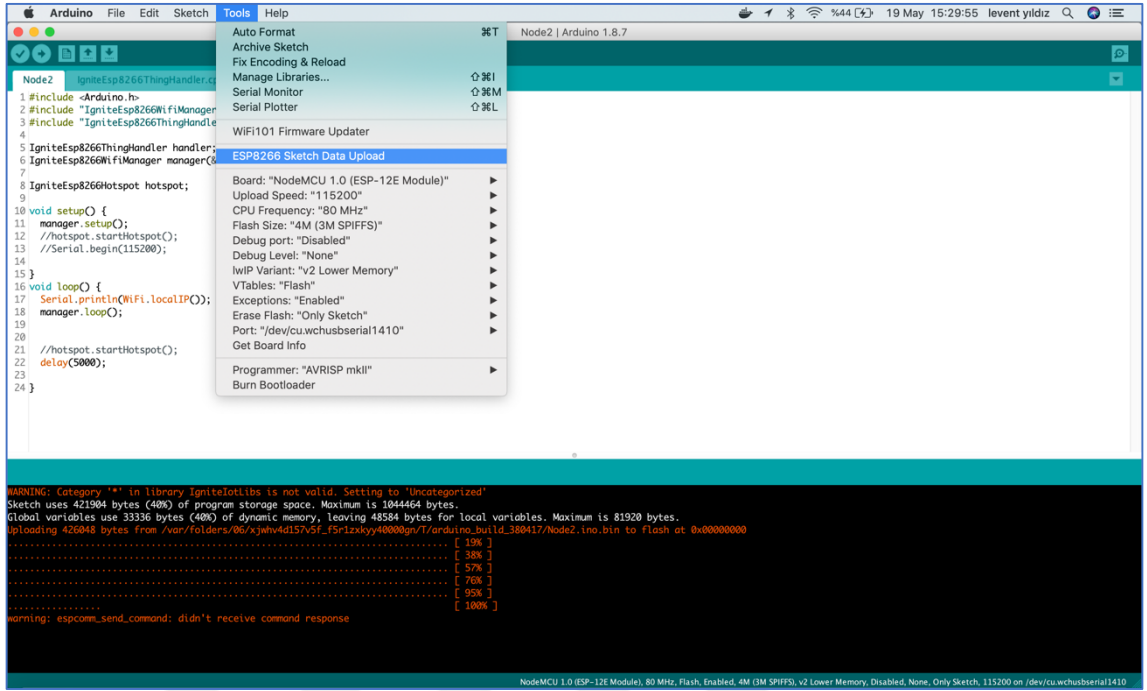


id	dataType
id	INTEGER
deviceId	TEXT
nodeId	TEXT
sensorId	TEXT
createDate	date
data	TEXT

Şekil 60: Mobil uygulamada sensör verilerinin tutulduğu sensordata tablosu

EK F

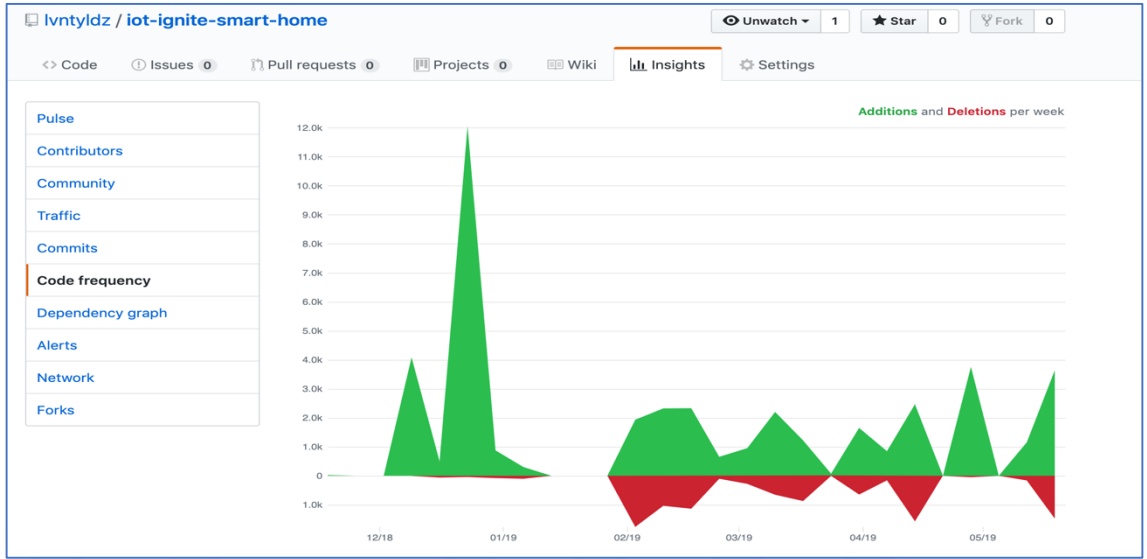
Çalışmada düğüm(node) tarafını geliştirirken ağırlıklı olarak Arduino IDE kullanılmıştır. Bu kapsamda düğümlerin ağ geçidine bağlanabilmesi için bir konfigürasyon göndermek gerekmektedir. Bu konfigürasyon dosyası içinde Wi-Fi erişim bilgileri, bağlanılacak ağ geçidi ID’si ve düğümün tanımı bulunmaktadır. Ağ geçidi ve düğümler bu sayede birbirlerini tanımaktadırlar. Bahsedilen dosyayı Node/data dizini altına eklenerek Arduino IDE üzerinden “tools > ESP8266 Sketch Data Upload” butonu aracılığıyla bu bilgiler düğümlere gönderilir. Aşağıdaki şekilde gösterilmiştir(Şekil 61).



Şekil 61: Arduino ide üzerinden “sketch data upload” işlemi

EKG

Şuana kadar yapılan çalışmalar GIT üzerinde bir REPO açılarak herkese açık olarak paylaşılmıştır. Bu aşamadan sonra üzerinde yeni geliştirmeler yapılabilir. Projenin çıktısı olan uygulamalar ve düğümlere gönderilen kodlar incelenebilir. Daha sonra yapılacak olan akıllı ev çalışmalarına bir ışık tutabilir. Projenin COMMIT girilme grafiği ve COMMIT geçmişi aşağıdaki şekillerden incelenebilir(Şekil 62-63).



Şekil 62: GIT kod gönderme grafiği

Commit Date	Commit Message	Commit Hash
Commits on May 18, 2019	PowerMock added to dependency levent yıldız committed a day ago	6107409
	Add WiFi, Network and Internet permission to manifest.xml levent yıldız committed a day ago	9a1671f
Commits on May 17, 2019	remove duplicated dependency from gradle levent yıldız committed 3 days ago	d3b7982
Commits on May 16, 2019	Add Unimplemented Methods levent yıldız committed 4 days ago	c295c1b
Commits on May 12, 2019	add iot-ignite dependencies levent yıldız committed 7 days ago	828cb1d
	Remove unnecessary test and ad sample joint test levent yıldız committed 7 days ago	7863b9d
Commits on Apr 28, 2019	remove unnecessary libs levent yıldız committed 21 days ago	4d81dbd

Şekil 63: GIT kod gönderme geçmişi

ÖZGEÇMİŞ

Levent YILDIZ, 2010 yılında Azerbaycan Teknik Üniversitesi Bilgisayar Mühendisliği bölümünü tamamladıktan sonra, Ağ ve İnternet güvenliği alanında yazılımlar geliştiren Anet yazılım ve danışmanlık şirketinde Teknik destek uzmanı olarak iş hayatına başladı. İş sözleşmesi imzalama aşamasında yazılıma olan ilgisini ve bu tarafa geçmek itediğini belirten ve sözünü alan Sn. YILDIZ ilerleyen zamanlarda Yazılım geliştirme departmanına geçerek 3 yıl kadar bir süreyle Anet'te çeşitli yazılımlar geliştirmiştir.

2014 yılında ise Türkkep Kayıtlı Elektronik Posta Hizmetleri tarafından yapılan teklifi değerlendirerek Türkkep'e geçen Sn. YILDIZ burada da 1 yıl kadar Java ile fullstack Web geliştirmesi yapmıştır.

Bir yandan özel sektörde çalışmalarına devam eden Sn. YILDIZ diğer yandan da Eğitim hayatında da ilerlemeyi hedeflemektedir. 2014 yılında Okan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği programında tezli yüksek lisans eğitimine başlamıştır.

2015 yılında IOT alanında geliştirme yapan Ardıç Arge ile yoluna devam etme kararı alan Sn. YILDIZ 3 yıl kadar bir süre de Ardıç için çeşitli geliştirmelerde bulunmuştur. Ardıç Arge de Bulut tarafında, Cihaz tarafında ve Mobil uygulama tarafında ana başlık olarak Html5, Java ve RN ile yazılımlar geliştirmiştir.

2018 yılı ortalarında Kıdemli yazılım mühendisi olarak Bilge adam bünyesine ve N11.COM projesine geçen Sn. YILDIZ şuan hala N11.COM için yazılım geliştirmesi yapmaktadır.