



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOKTORA TEZİ**

**KABLOSUZ ALGILAYICI AĞLARIN BULUT  
HESAPLAMASI KULLANILARAK İNTERNETE  
GENİŞLETİLMESİ**

**Bilg.Yük.Müh. Akhan Akbulut  
Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Programı**

**Danışman  
Prof.Dr. A.Halim Zaim**

**Şubat, 2013**

**İSTANBUL**

2602080070 Öğrenci numaralı Akhan AKBULUT tarafından hazırlanan bu çalışma 23/01/2013 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi

  
Prof. Dr. Abdül Halim ZAIM (Danışman)  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Prof. Dr. Gökhan UZGÖREN  
İstanbul Kültür Üniversitesi  
Mühendislik Fakültesi

  
Yrd. Doç. Dr. Öğr. Üyesi ÖZTAŞ  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Prof. Dr. Ahmet SERTBAŞ  
İstanbul Üniversitesi  
Mühendislik Fakültesi

  
Prof. Dr. Murat TAYLI  
Maltepe Üniversitesi  
Mühendislik Fakültesi

## **ÖNSÖZ**

Tez danışmanım olarak beni her konuda destekleyen, hem ders hem de tez aşamasında her türlü yönlendirmeyi ve desteği sağlayarak önümü açan, önerileriyle bu uzun süreci daha verimli geçirmemi sağlayan değerli hocam Prof. Dr. Abdül Halim Zaim'e minnet ve şükranlarımı sunarım.

Tez izleme komitesi üyelerim Prof. Dr. Ahmet Sertbaş ve Prof. Dr. Gökhan Uzgören'e tezimin gelişimi boyunca verdikleri değerli öneriler için teşekkür ederim.

Tez savunmamda yer alarak, fikir ve düşüncelerini benimle paylaşan Prof. Dr. Murat Taylı'ya ve Yrd. Doç. Dr. Oğuzhan Öztaş'a teşekkür ederim.

Son olarak, her zaman yanımda olup beni destekleyen sevgili eşim Fatma'ya ve bugünlere gelmemi sağlayan aileme içten teşekkürlerimi sunarım.

**Şubat, 2013**

**Akhan Akbulut**

# İÇİNDEKİLER

ÖNSÖZ.....	i
ŞEKİL LİSTESİ.....	v
TABLO LİSTESİ.....	vi
SEMBOL LİSTESİ.....	vii
ÖZET.....	ix
SUMMARY.....	xi
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. BİLİMSEL KATKILAR.....	1
1.2. AMAÇ.....	4
<b>2. GENEL KISIMLAR.....</b>	<b>5</b>
2.1. GİRİŞ.....	5
2.2. KABLOSUZ ALGILAYICI AĞLAR.....	5
2.2.1. Sensör Düğüm (Mote).....	5
2.2.2. Yalın Platformlar (Low-End Platforms).....	8
2.2.3. Üst Düzey Platformlar ( High-End Platforms ).....	8
2.2.4. KAA'ların Kullanım Alanları.....	8
2.2.5. Askeri Uygulamalar.....	9
2.2.6. Çevresel Uygulamalar.....	9
2.2.7. Sağlık Uygulamaları.....	9
2.2.8. Ev Uygulamaları.....	10
2.2.9. Endüstriyel Uygulamalar.....	10
2.3. KABLOSUZ ALGILAYICI AĞLARDAKİ VERİYE ULAŞMAK.....	10
2.3.1. KAA'lar İçin Tasarlanmış Ara Katmanlar (Middleware).....	11
2.3.2. Veritabanından Esinlenilmiş Çözümler.....	11
2.3.3. Yayınla / Katıl Sistemler.....	12
2.3.4. Servis Keşfi Tabanlı Yaklaşımlar.....	14
2.3.5. Demet Alanı Çözümleri.....	14
2.3.6. Sanal Makine Yaklaşımları.....	14
2.3.7. Genel Ara Katman Değerlendirmesi.....	15
2.4. BULUT BİLİŞİM.....	15
2.4.1. Yazılımın Servis Olarak Sunulması (Saas).....	16

2.4.2.	Platformun Servis Olarak Sunulması (Paas).....	16
2.4.3.	Altyapının Servis Olarak Sunulması (IaaS).....	17
2.4.4.	Bulut Bilişim Ortamlarının Değerlendirilmesi .....	18
2.4.5.	Microsoft Azure .....	18
2.4.6.	Google Uygulama Motoru (GAE) .....	20
2.4.7.	Amazon Web Servisleri (AWS).....	21
2.4.8.	Appharbor .....	23
2.4.9.	Seçilen Bulut Bilişim Platformu: Microsoft Azure.....	24
2.4.10.	Windows Azure üzerinde uygulama geliştirme .....	24
<b>3.</b>	<b>MALZEME VE YÖNTEM.....</b>	<b>26</b>
3.1.	CAA'LAR İÇİN KULLANILABİLİR BİR YAYINLA/KATIL MİMARİSİ	26
3.1.1.	Yayınla/Katıl İletişim Modeli .....	26
3.1.2.	Mevcut Yayınla/Katıl Mimarileri .....	27
3.1.3.	Geliştirilen Yayınla/Katıl Modeli .....	29
3.1.3.1.	<i>Altyapı</i> .....	30
3.1.3.2.	<i>Bileşenler</i> .....	31
3.1.3.3.	<i>İletişim modeli</i> .....	32
3.1.4.	Kullanılan Fiziksel CAA Platformu: Sun SPOT .....	35
3.1.4.1.	<i>Sun Spot Uygulama Geliştirme Ortamı</i> .....	36
3.2.	BULUT ORTAMI VE GELİŞTİRİLEN SERVİSLER .....	38
3.2.1.	Bulut Ortamında Veri Saklama.....	38
3.2.2.	Genetik Algoritma Kullanılarak Dağıtımın İyileştirilmesi .....	40
3.2.2.1.	<i>Kapsama Alanı Boşluklarının Tespiti</i> .....	42
3.2.2.2.	<i>Yöntem</i> .....	44
3.2.2.2.1.	<i>Fiziksel Mimari</i> .....	44
3.2.2.2.2.	<i>Genetik Algoritmanın Uygulanması</i> .....	45
3.2.2.2.3.	<i>Yer değişikliği yapılacak düğümlerin belirlenmesi</i> .....	46
3.2.2.2.4.	<i>Genetik algoritma ile mevcut dağıtımın iyileştirilmesi/güncellenmesi</i> .....	47
3.2.3.	Hatalı Ölçüm Yapan Düğümlerin Kestirimi .....	49
3.2.3.1.	<i>Yöntem Seçimi</i> .....	50
3.2.3.2.	<i>Bayes Ağ Tabanlı Yaklaşımlara Ait Gerçekleşmiş Çalışmalar</i> .....	52
3.2.3.3.	<i>Dinamik Bayes Ağı İle Hatalı Sensör Tespiti</i> .....	53

3.2.3.4. Hata Modeli .....	53
3.2.3.5. Kullanılan Veri Kümesinin Oluşturulması .....	55
<b>4. BULGULAR .....</b>	<b>59</b>
4.1. YAYINLA/KATIL MİMARİSİ İLE İLGİLİ YAPILAN ÇALIŞMALAR VE EDİLEN BULGULAR.....	59
4.1.1. Yayınlayıcı Uygulama: WatchMan.....	59
4.1.2. Aracı Uygulama: Mediator .....	60
4.1.3. Geçit Uygulaması: DataAggregator .....	61
4.1.4. Bilgilendirme Uygulaması: Notifier .....	62
4.1.5. Yönetim Uygulaması: Portal.....	62
4.2. GENETİK ALGORİTMA KULLANILARAK YAPILAN DAĞITIMIN İYİLEŞTİRİLMESİ İLE İLGİLİ SİMULASYON ÇALIŞMALARI VE ELDE EDİLEN BULGULAR.....	63
4.3. HATALI ÖLÇÜM YAPAN DÜĞÜMLERİN TESPİTİ İLE İLGİLİ SİMULASYON ÇALIŞMALARI VE ELDE EDİLEN BULGULAR .....	67
<b>5. TARTIŞMA VE SONUÇ .....</b>	<b>69</b>
<b>KAYNAKLAR .....</b>	<b>72</b>
<b>ÖZGEÇMİŞ.....</b>	<b>83</b>

## ŞEKİL LİSTESİ

Şekil 1.1 : Araştırma/Geliştirme Süreci .....	3
Şekil 2.1 : Sensör Düğüm Yapısı .....	6
Şekil 2.2 : TinyDB Sorgu Örneği.....	11
Şekil 2.3 : Yayınla Katıl İletişim Modeli .....	13
Şekil 2.4 : Bulut Bilişim Modelleri .....	18
Şekil 2.5 : Azure Mimarisi .....	19
Şekil 2.6 : GAE Mimarisi .....	21
Şekil 2.7 : AWS Mimarisi.....	22
Şekil 2.8 : AppHarbor Mimarisi .....	23
Şekil 3.1 : Yayınla / Katıl Sisteminin Genel Yapısı .....	26
Şekil 3.2 : Geliştirilen Yayınla/Katıl Modelinin İskeleti .....	30
Şekil 3.3 : Geliştirilen Yayınla/Katıl Modelinin Fiziksel Yapısı.....	32
Şekil 3.4 : Sun SPOT cihazı radyo iletişim yığıtı (stack) .....	33
Şekil 3.5 : Deney için Tercih Edilen KAA Platformu: Sun SPOT .....	35
Şekil 3.6 : Solarium emülatöründeki sanal düğümlerin görüntüsü.....	37
Şekil 3.7 : SQL Azure Şema Görünümü.....	39
Şekil 3.8 : Genetik Algoritmanın Akış Diyagramı (Flow Diagram).....	46
Şekil 3.9 : Konumundan dolayı ölçüm kapasitesi efektif olarak kullanılamayan düğümler .....	48
Şekil 3.10 : Birbirine çok yakın konumlandırılmış düğümlerin çakışması durumu .....	48
Şekil 3.11 : Aykırı gözlem tespit teknikleri .....	50
Şekil 3.12 : Düğümlerin zamandaki değişimini gösteren kesitler.....	54
Şekil 3.13 : Deniz Suyu Sıcaklıklarının Uydu Görüntüleri .....	56
Şekil 3.14 : Veri Kümesini Üreten Simülasyon Yazılımın Ekran Görüntüsü.....	57
Şekil 4.1 : Portal Uygulaması Ekran Görüntüsü .....	63
Şekil 4.2 : Simülasyon dağıtımını görüntüsü .....	64
Şekil 4.3 : Simülasyon dağıtımının iyileştirilmiş halinin görüntüsü.....	64

## **TABLO LİSTESİ**

Tablo 2.1 : Sensör Düğüm Özellikleri .....	7
Tablo 3.1 : Yayınla/Katıl Modelinin Bileşenleri.....	32
Tablo 4.1 : Test Sonuçları .....	65
Tablo 4.2 : Hatalı Düğüm Kestirim Deney Sonuçları.....	68



## SEMBOL LİSTESİ

<b>CAA</b>	: Kablosuz Algılayıcı Ağlar
<b>IP</b>	: İnternet Protokol
<b>DCOM</b>	: Distributed Component Object Model (Dağıtılmış Bileşen Nesne Modeli)
<b>CORBA</b>	: The Common Object Request Broker Architecture (Ortak Nesne İstem Aracısı Mimarisi)
<b>PVM</b>	: Parallel Virtual Machine (Sanal Paralel Makine)
<b>MPI</b>	: Message Passing Interface (Mesaj Geçirme Arayüzü)
<b>SQLT</b>	: Sensor Query and Tasking Language (Sensör Sorgulama Dili)
<b>GSN</b>	: The Global Sensor Network (Küresel Sensör Ağı)
<b>SIP</b>	: Session Initiator Protocol (Oturum Başlatıcı Protokol)
<b>MILAN</b>	: Middleware Linking Applications and Networks ()
<b>API</b>	: An application programming interface (Uygulama Programlama Arayüzü)
<b>QoS</b>	: Quality of Service (Ağ İletişimi Hizmet Kalitesi)
<b>SOA</b>	: Service-Oriented Architecture (Servis Odaklı Mimari )
<b>CRM</b>	: Customer relationship management (Müşteri İlişkileri Yönetimi )
<b>SOAP</b>	: Simple Object Access Protocol (Basit Nesne Erişim Protokolü)
<b>SaaS</b>	: Software As A Service (Yazılımın Servis Olarak Sunulması )
<b>PaaS</b>	: Platform As A Service (Platformun Servis Olarak Sunulması )
<b>IaaS</b>	: Infrastructure As A Service (Altyapının Servis Olarak Sunulması )
<b>BT</b>	: Bilişim Teknolojileri
<b>IIS</b>	: İnternet Information Services (İnternet Bilgi Servisleri)
<b>VM</b>	: Virtual Machine (Sanal Makine)
<b>CDN</b>	: A content delivery network (İçerik Dağıtım Ağı)
<b>NTFS</b>	: New Technology File System (Yeni Teknoloji Dosya Sistemi)
<b>CPU</b>	: Central Processing Unit (Merkezi İşlem Birimi)
<b>GAE</b>	: Google App Engine (Google Uygulama Motoru)
<b>EC2</b>	: Amazon Elastic Compute Cloud (Amazon Esnek Hesaplama Bulutu)
<b>ELB</b>	: Elastic Load Balancing ()
<b>AWS</b>	: Amazon Web Services (Amazon Web Servisi)

<b>EBS</b>	: Amazon Elastic Block Store (Amazon Elastic Blok Mağazası)
<b>Amazon S3</b>	: Amazon Simple Storage Service (Amazon Basit Depolama Servisi)
<b>SQS</b>	: Software Quality Systems (Yazılım Kalite Sistemleri)
<b>DNS</b>	: Domain Name System (Alan Adı Sistemi)
<b>SVN</b>	: Support Vector Machine (Destek Vektör Makineleri)
<b>TFS</b>	: Team Foundation Server ()
<b>J2ME</b>	: Java Platform, Micro Edition (JAvA platformu)
<b>GCF</b>	: Generic Connection Framework (Genel Bağlantı İskeleti)
<b>SPOT</b>	: Small Programmable Object Technology (Küçük Programlanabilir Nesne Teknolojisi )
<b>CDS</b>	: Minimal Connected Dominating Set ()
<b>RNG</b>	: Relative Neighbor Graph (Yakın Komşuluk Grafiği )
<b>BN</b>	: Bayesian Network (Bayes Ağı)
<b>DBA</b>	: Dynamic Bayesian Network (Dinamik Bayes Ağı)

## ÖZET

### KABLOSUZ ALGILAYICI AĞLARIN BULUT HESAPLAMASI KULLANILARAK İNTERNETE GENİŞLETİLMESİ

Kablosuz Algılayıcı Ağlar (KAA) uzun yıllar boyunca araştırmalara konu olmuş bir mimaridir. KAA sistemleri, yeni çıkan ve çekici çözümleriyle; ulaşım, iş, sağlık, askeri, endüstriyel otomasyon, çevresel görüntüleme gibi geniş bir yelpazede kullanılmaktadır. Gözlemlenmek istenen bölge üzerine konuşlandırılmış, ortam karakteristiklerini ölçümleme kabiliyetine sahip, kablosuz iletişim yapabilen çok sayıda duyurga düğümünden oluşmakta olan bu ağların; duyurga düğümlerinin birim maliyetleri azdır ve düşük kurulum maliyetleri bulunmaktadır. Öte yandan; güç kaynağının sınırlı olması, hesaplama gücünün çok az oluşu ve üst seviye (IP) ağ protokolünden yoksun olmaları en temel yapısal kısıtlamalarıdır.

Günümüzde geleneksel KAA verilerin daha geniş alanlara taşınabilmesi, üzerinde çalışılmakta olan önemli bir araştırma konusudur. Duyurga düğümlerinin kullanılmakta olduğu IEEE 802.15.4 iletişim protokolünün IP ile farklı olmasından, akademik ve sektörel çalışmalar iki yöne odaklanmaktadır. Birinci yaklaşım KAA iletişimi için, düşük tüketimli bir IP hizmeti geliştirilmesidir; 6LoWPAN isminde tanımlanmış olan uluslararası proje bu konu dâhilindeki araştırmalarını sürdürmektedir. Diğer yaklaşım ise; türdeş olmayan ağlar arasında iletişime imkân veren bir geçit yapısının sunulmasıdır. Araştırmamın ilk kısmında, KAA ölçümlerinin internet üzerinden erişilmesine olanak sağlayacak bir Yayınla/Katıl altyapısı geliştirilmiştir. Yayınla/Katıl modelinde mesajlaşma yapan sistemler, veri-merkezli iletişimin en bilindik örneklerindedir. Ölçeklenebilir oluşu ve dinamik uygulama topolojisine imkân verebilmesi sebebiyle KAA'lara uygulanabilirler. Temel Yayınla/Katıl modelinde bulunan üç ana öge olan yayıncı, katılımcı ve bildirim servisinin gerçekleşmesi için beş farklı yazılım geliştirilmiştir. Bu beş yazılım; üç farklı çalışma ortamında, iki farklı programlama dili kullanılarak farklı davranış biçimlerini gerçekleştirmesi için geliştirilmiştir. Önerilen sistemin temel amacı, duyurga düğümlerinin ürettiği ölçüm değerlerini, merkez düğüm üzerinden katılımcıların çeşitli platformlardan erişebileceği bulut ortamına aktararak daha etkin kullanılmasını sağlamaktır.

Bulut bilişim, kuruluşlara herhangi bir altyapı yatırımı yapmadan servis kapasitelerini etkili ve hızlı bir şekilde arttırmalarını sağlamaktadır. Bulut bilişim, kullanıcıların ihtiyaç duyduğu yazılım, donanım, depolama ve hesaplama hizmetleri gibi temel ihtiyaçlarını veri merkezleri üzerinden sunan mimaridir. Üretilen ölçüm verilerinin saklanması ve katılımcılara sunulmasında, son on yılda yükselmekte olan bu teknoloji tercih edilmiştir.

Geleneksel KAA sistemleri, gerçek dünyadan veri toplamak için tasarlanmıştır, fakat verileri toplayan organizasyonun artık o veriye ihtiyaç duymaması durumunda, o veriye ne yapılacağı veya ne şekilde saklanacağına yanıtı bulunmamaktadır. Çok sıklıkla yapılan ortam karakteristiklerine ait ölçümlerin, ilişkisel veritabanları üzerinde depolanması kullanım zorlukları doğurmaktadır. Toplanan verinin büyük ölçekte oluşu sebebiyle, belirli periyotlarda veritabanlarından geçmiş veriler temizlenir. Bu durum geçmişe dönük raporların alınmasını veya ileriye dönük kestirim hesaplarının tutarsız

olmasına sebep olmaktadır. Geliştirilen sistem ile tüm ölçüm verisi Microsoft SQL Azure bulut veritabanı üzerinde kapasite sorunu olmadan depolanabilmektedir.

Bulut bilişimin sunduğu sınırsız hesaplama ve depolama yetisi sayesinde, geleneksel KAA'daki bazı sorunların çözümü mümkün olabilmektedir. Araştırmanın ikinci kısmı, ölçüm verilerinin bulut ortamı üzerinde nasıl etkin kullanılacağı üzerinedir. Bu kapsamda üç temel problemin çözümü üzerinde çalışmalar gerçekleştirilmiştir. Bunlar; KAA'lardaki düğüm dağıtımının iyileştirilmesi, hatalı ölçüm yapan sensör düğümlerin tespit edilmesi ve ölçüm yapılamayan bölgeler için tahmini ölçüm değerinin hesaplanmasıdır.

Plansız yapılan düğüm dağıtımlarından oluşan kapsama alanı boşluklarının tespit edilmesi ve giderilmesi için bulut ortamında, bir servis geliştirilmiştir. Bu servis genetik algoritma kullanarak, optimum değişiklik ile dağıtımın iyileştirilmesini sağlamaktadır.

KAA'ların ürettiği ölçüm değerlerinin doğruluğu, servis kalitesinin belirlenmesindeki en önemli ölçütlerden biridir. Fiziksel hasar veya yazılımsal sebeplerden hatalı ölçüm yapmaya başlayan düğümlerin tespit edilerek, üretmekte oldukları hatalı verilerin katılımcılara aktarılması engellenmelidir. Bulut ortamı üzerinde geliştirilmiş olan ikinci servis; aykırı gözlem teşhisidir. Dinamik Bayes sınıflandırma yöntemi kullanılarak hatalı ölçüm yapan düğümler tespiti gerçekleştirilmiş ve üretilen hatalı verinin sisteme dahil olması engellenmiştir.

Geliştirilen model, KAA sistemleri ve bulut mimarisi arasında bir bütünleşme çatısı önermektedir. Önerilen entegrasyon çatısının görevi, KAA sisteminden üretilen olayların daha yaygın kitlelere sunulmak üzere bulut ortamına aktarılmasıdır. Bu sayede bulut bilişimin sunduğu imkanlar ile bahsi geçen servisler gerçekleştirilebilmektedir.

## **SUMMARY**

### **EXTENDING WIRELESS SENSOR NETWORKS INTO THE INTERNET USING CLOUD COMPUTING**

Wireless Sensor Networks (WSN) is a concept that has been the subject of the researches for many years. WSN systems are used in a wide range such as transport, business, health, military, industrial automation, environmental imaging with its incoming and winning solutions. Unit costs of sensor nodes of these networks which consist of so many sensor nodes and can establish wireless communication and have the ability of measuring environmental characteristics and have been positioned in the area desired to be observed are low and they have low set-up costs. On the other hand, power supply is limited, calculation performance is very low and they are deprived of high level communication (IP) and these are the basic structural limitations.

Transferring traditional WSN data to a wider area in our day is an important research subject that is studied on. Since communication protocol IEEE 802.15.4 that sensor motes use complies with IP, academically and sectoral studies focus on two aspects. First approach is to develop a low consumed IP service for WSN communication; international project defined as 6LoWPAN continues its studies within this subject. Other approach is to present an accommodation structure that enables communication between heterogeneous networks. In the first part of thesis research, a Publish/Subscribe infrastructure has been developed to provide WSN measurements to be accessed through internet. The systems that have messaging in Publish/Subscribe model are the best-known examples of data-centric communication. They can be applied to WSNs because of that they are scalable and enable dynamic application topology. Five different software have been developed so that publisher, attendee and notification services, three main elements in main Publish/Subscribe model can take place. These five software have been developed to perform different course of actions by using two different programming languages in three different work environments. In brief, the basic purpose of the proposed model is to enable measurement values that sensor nodes produce to be transferred into cloud environment that the participants access through central node from various platforms and provide them to be used more efficient.

Cloud computing provides companies to increase their service capacities in an efficient and quick way without making any infrastructure investment. Cloud computing is the architecture that presents the basic needs of the users such as software, hardware, storage and calculations services through data centers. This technology which has been on the increase for the last ten years has been preferred to keep measurements values produced and present to the participants.

Traditional WSN systems has been designed to collect data from the real world but on condition that the organization which collects the data doesn't need that data any longer, there will not any response what to do that data and how it will be kept. The storage of measurements frequently made regarding environment characteristics on relational database causes difficult of use. Because of that the data collected is very big, previous data is deleted from databases in certain periods. This situation causes forward-looking estimates to be in consistent and reports looking back to be received. With the system I

developed, all measuring data can be stored on Microsoft SQL Azure cloud database without having a capacity problem.

By the virtue of unrestricted estimating and storing ability that Cloud computing presents, the solution of some problems in traditional WSN is possible. The second part of my thesis research is aimed how to use measuring data efficiently on cloud environment. In this scope, the studies have been made for the solution of three basic problems. These are; improvement of nodal distribution in WSNs, determination of sensor nodes that make wrong measuring and calculation of estimated measuring value for the areas that can't be measured.

A service has been developed to determine and overcome network coverage spaces arising from unplanned nodal distributions in cloud environment. This service provides improvement of distribution with optimum change by using genetic algorithm.

The accuracy of measuring values that WSNs produce is one of the most important metrics in the determination of service quality. Nodes that start making wrong measurements due to various reasons should be determined and transferring wrong data they produce to the participants should be prevented. The second service developed in cloud environment is detection of outlier observation. By using dynamic Bayes classification method, the nodes that make wrong measurements are determined.

The developed model proposes integration between WSN systems and cloud architecture. The role of proposed integration frame is to transfer the data WSN system produces to cloud environment to transfer bigger masses and aforementioned services have been occurred within the possibility cloud computing presents.

## 1. GİRİŞ

Günümüzün ilerleyen teknolojisi ile dağıtık sistemlerin kullanımını artmaktadır. Pek çok farklı hizmet sahasında çok çeşitli amaçlarda çalışan bu sistemler üstün bir iletişim kabiliyetini beraberinde getirmektedir. Kablosuz Algılayıcı (duyurga) Ağlar (KAA) ortam karakteristiklerinin tespit ve düzenli takibinde sıklıkla kullanılmaktadırlar. KAA'ların en önemli avantajlarından biri tasarısız (ad-hoc) bir şekilde kendi kendine organize olabilme yeteneğine sahip olmalarıdır. Ölçüm yapan düğümlerin birim maliyetlerinin düşük oluşu ve kablosuz iletişim kabiliyetleri ile pek çok alanda kullanılmaktadırlar.

Sensör düğümlerin hesaplama kabiliyetlerinin çok kısıtlı olması ve dış ortamlara (ağlara) veri aktaramamaları sınırlı sayıdaki dezavantajlarındanır. Tez konusu dâhilinde bu dezavantajların giderilmesine yönelik araştırmalar ve geliştirmeler yapılmıştır.

Öncelikli olarak, geleneksel KAA bilgilerinin daha geniş ortamlara taşınması için altyapı planı yapılmıştır. Tasarlanan altyapının sahip olması gereken karakteristikleri belirlenerek, bu doğrultuda uygulamalar gerçekleştirilmiştir.

Çözüm üretirken, son on yılın güncel servis mimari bulut bilişim kullanılmıştır. Bulut mimarisinin sunmuş olduğu yüksek hesaplama ve depolama yetisi ile geleneksel KAA sorunlarına çözümler aranmıştır.

### 1.1. BİLİMSEL KATKILAR

Bu bölümde, bu tez çalışmasının tamamlanması sonucunda ortaya çıkartılmış olan katkılar anlatılmaktadır. Yapılan araştırmalar ve geliştirmeler KAA'lar ve Bulut Bilişim konularına odaklıdır. Çalışmanın sonucu olarak KAA'ların bulut mimarisine entegrasyonun imkan kıldığı servisler aşağıda maddeler halinde sunulmaktadır.

**Katkı 1** KAA'lar kapalı devre ağ özelliğinde olmalarından ötürü, ağ içerisinde üretilen veri sadece yönetim bilgisayarında tutulmaktadır. Ağ içerideki veriyi daha geniş ortamlara aktarabilmek için IP ağ protokolü seviyesine çıkmak gerekmektedir. Radyo frekansı üzerinden yapılan iletişimin daha üst katmanlara iletimini sağlamak için bir ara

katman hizmeti sunulmalıdır. Tez çalışmasının bel kemiği olarak geliştirilen yayımla/katıl mimarisi ölçüm yapan düğümlerin üretmiş olduğu veriyi, ölçüm sahasından bulut üzerindeki depolama alanına kadar aktarabilmek üzere tasarlanmıştır. Mimarinin ana aktörleri olan yayımlayıcı uygulama *WatchMan*, mantıksal bölümlere ayrılmış olan gözlem alanındaki her bir yetkili düğüm üzerinde çalışan aracı uygulama *Mediator* ve son olarak tüm verilerin bulut ortamına aktarımını sağlayan *DataAggregator* uygulamalarının tasarımı ve tatbiki yapılmıştır. Geliştirilen uygulamalar *WatchMan* ve *Mediator* Sun Spot sensör kiti üzerinde çalıştırılmış ve yapılan sıcaklık ölçüm bilgisi *DataAggregator* uygulaması ile bulut ortamındaki veritabanı hizmetine aktarılmıştır. Bu iletişim altyapısı 3. bölümde detaylandırılmaktadır.

Katkı 1'in KAA verilerine erişimi etkin kılması ile geleneksel KAA sistemlerinde bulunmayan bazı özelliklerin tanımlanması mümkün olmaktadır. Yapılan diğer geliştirmeler bu başlık altında değerlendirilebilir.

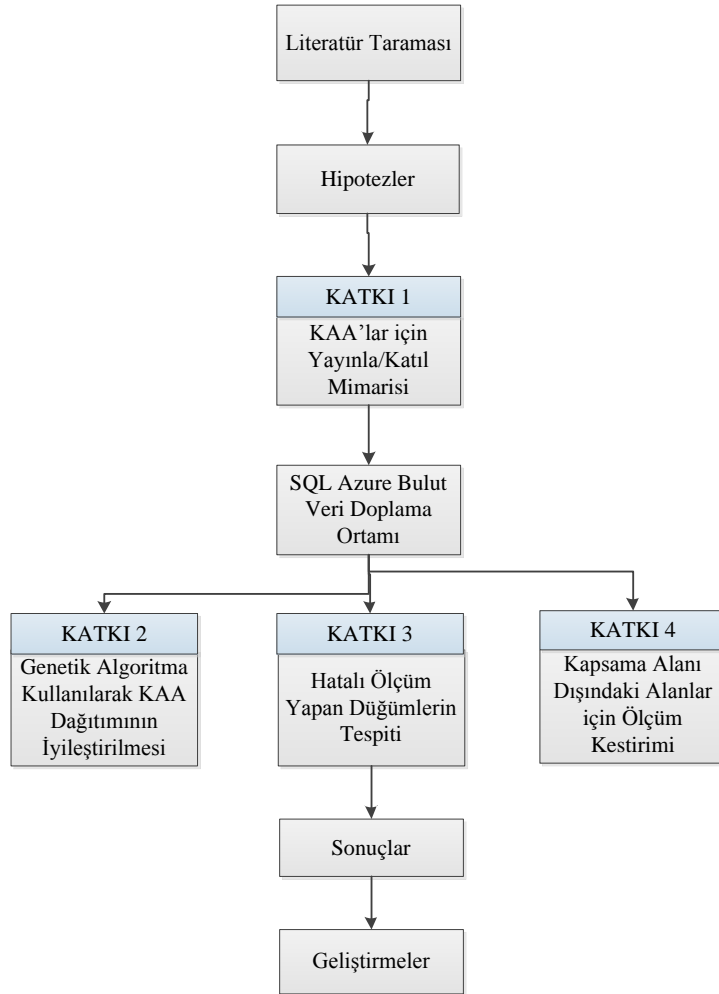
**Katkı 2** KAA'lardaki düğüm dağıtımının iyileştirilmesine yönelik servis geliştirilmiştir. Bir bulut uygulaması olan bu servis; sistem üretmiş olduğu veriyi kullanarak incelenmekte olan KAA bölgesinin kapsama alanı boşluklarını tespit etmektedir. Aynı şekilde bu kapsama alanı boşluklarına yerleştirilmek üzere etkin kullanılmayan sensör düğümler belirlenerek, genetik algoritma vasıtası ile boşluklara atanması sağlanmaktadır. Dağıtımın iyileştirilmesi ile gözlem alanının yüzdesel artışı sağlanmakta ve bu tarz sistemlerin en önemli servis kalitesi metriklerinden biri geliştirilmektedir. KAA dağıtımının iyileştirilmesini sağlayan bu servis 4.3 bölümünde anlatılmaktadır.

**Katkı 3** KAA sistemlerinin, kullanıcılarına aktardığı bilgilerin doğru olması ve ortam karakteristiklerinin gerçek değerlerini yansıtmaları gerekmektedir. Özellikle çevre takibi amacıyla kullanılan sistemlerdeki bilgilerin doğruluğu çok kritiktir. Bilginin içerisinde yer alan gürültü, doğruluğu azaltan bir etmendir. Bu gürültünün süzülmesi ile temiz veri oluşturulur, bu da kullanım doğruluğu yaratır. Yüksekten düşmeye bağlı olarak fiziksel hasar alan sensör düğümler hatalı ölçümler gerçekleştirebilirler. Benzer şekilde ölçüm yapan düğümün etrafında bulunan çevresel bir engel, ölçüm doğruluğunu bozabilmektedir.



Bu düğümlerin ürettiği ölçümlerin kullanılmaması için aykırı gözlem teşhisi (outlier detection) uygulanmalıdır. Geleneksel KAA sistemlerinde bulunmayan fakat hatalı ölçüm yapan sensör düğümlerin tespit ederek ölçüm değerlerinin daha anlamlı olmasını sağlayan bu servis 4.4 bölümünde anlatılmaktadır. Hatalı ölçüm yaptığı tespit edilen düğümler tüm sistem içerisinde kara listeye alınarak sadece veri aktarım görevini sürdürürler.

**Katkı 4** Fiziksel hasar alma veya pil tükenmesi durumlarından ölçüm yapamaz hale gelen düğümlerin bölgelerindeki ölçüm hizmeti sonlanmaktadır. Bu bölgeler kapsama boşluğu oluşturduğu için KAA sistemlerinde istenmeyen en önemli durumdur. Ölçüm yapılamayan bölgeye yeni düğüm konuşlandırılana kadar, son kullanıcıya aksamının yansıtılmaması gerekir. Bu saydamlık (transparency) ilkesine bağlı olarak geçmiş verilerin yorumlanarak tahmini ölçüm değerinin sunulduğu servis 4.5 bölümünde anlatılmaktadır.



Şekil 1.1 : Araştırma/Geliştirme Süreci

Şekil 1.1 tezimin geliştirilmesi sürecinde izlediğim süreçleri özetlemektedir. Metodolojik bir açıdan bakıldığında; kapsamlı bir literatür taraması sonucunda hipotezlerim oluşmuştur. Hipotezlerimi gerçeklerken kullandığım öncelik sırası Katkı 1'in sonlaması ile paralel bir çalışmaya dönmüştür. Katkı 2, 3 ve 4, bulut ortamının kullanılabilirliği ile mümkün olmuştur. Bulut üzerinde sunulan bu 3 servisin başarımları ve gelecekte yapılabilecek iyileştirmeler süreci sonlandırmaktadır.

## **1.2. AMAÇ**

Araştırmanın amacı umumi KAA sistemlerinin erişilebilirliğini arttırarak, yöneticilerin ihtiyaç duydukları özel servisleri geliştirmektir. Araştırma kapsamında kurumlara/şahıslara ait olan özel KAA sistemlerinin birleştirilerek devasa bir ölçüm ağı oluşturulması ve bu ağların yaptığı ölçüm verilerine internet üzerinden kullanıcıların abonelik sistemi ile erişebilmesi hedeflenmektedir. İkinci bölümde araştırma konusu olan KAA teknolojileri ve Bulut Bilişim anlatılmaktadır. Üçüncü bölümde, geleneksel KAA sistemlerinde bulunmayan servisler olan; katkı 2, 3 ve 4'te özetlemiş hizmetlerin kullanılabilir olması için yapılan çalışmalar detaylandırılmıştır. Bölüm dört, geliştirme modelini ve başarımları anlatmaktadır. Araştırmanın genel sonuçları ve çıkarımları beşinci bölümde sunulmuştur.

## **2. GENEL KISIMLAR**

### **2.1. GİRİŞ**

Yapılan çalışma geleneksel bir Kablosuz Algılayıcı Ağ (KAA) sisteminin ürettiği verileri internete genişletmek üzerine gerekli altyapı ve servislerin tasarlanmasıdır. Bunun için öncelikle, geleneksel bir KAA sisteminin sahip olduğu karakteristikler incelenmelidir. KAA içerisindeki veriyi dışarıya taşıyacak iletişim modelinin belirlenmesinde günümüze kadar yapılan çalışmalar kıyaslanmalıdır. Hangi iletişim modeli tercih edilirse edilsin, bilgilerin muhafaza edileceği ve katkı 2, 3 ve 4'teki servislerin barındırılacağı bir bulut ortamına ihtiyaç duyulmaktadır. Hali hazırda sunulan bulut bilişim modellerinin kıyaslaması yapılarak tercih edilecek bulut bilişim belirlenmelidir. Bu bölüm KAA'lar, KAA ara katmanları ve bulut bilişim hakkındaki literatür taramasını sunmaktadır.

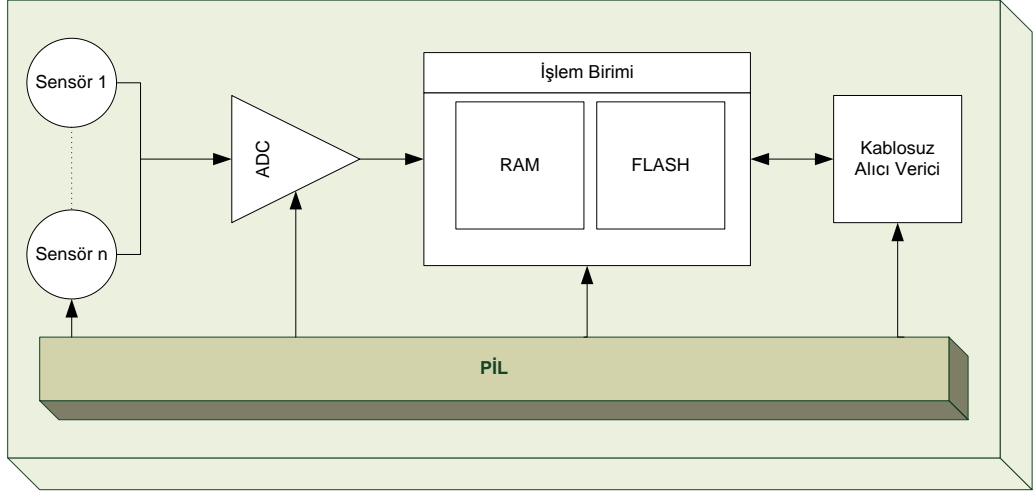
### **2.2. KABLOSUZ ALGILAYICI AĞLAR**

KAA'lar günümüzde hızla kullanım alanı genişlemekte ve sınır güvenliği gibi kritik görev durumlarında kullanılabilen, büyük potansiyele sahip bir teknoloji olarak karşımıza çıkmaktadır. Bir bölge üzerine dağıtılmış ölçüm kabiliyetine sahip, kablosuz iletişim yapabilen çok sayıda düğümünden oluşmakta olan bu ağlar [1,2,3,4]; güncel hayatımızda trafik analizi, depo yönetimi ve sıcaklık, nem, ışık gibi ortam karakteristiklerini gözlem gibi uygulama alanlarında kullanılmaktadır. Bu ağ içerisindeki temel yapı taşı olan her bir düğüm; bir mikroişlemciye, bir radyo-vericisine, sensör(ler)e ve hesap yapabilme yeteneğine sahiptir. Güç kaynağının sınırlı olması ve hesaplama gücünün çok az oluşu en temel yapısal kısıtlamalarıdır. Öte yandan; sensör düğümlerin birim maliyetlerinin az oluşu, kablosuz iletişim kabiliyetleri sayesinde düşük kurulum maliyetlerinin bulunması, bu teknolojinin yaygınlaşmasındaki en önemli karakteristikleridir.

#### **2.2.1. Sensör Düğüm (Mote)**

KAA içerisindeki tüm düğümler ağ için röle vazife görür ve ölçüm yaparak veri üretiminden sorumludur. Bir sensör düğüm; hesaplama, iletişim ve ölçüm yeteneklerinin birleştirmektedir. Yerleşik (on-board) mikroşlemcilere sahip

olduklarından, çoğunlukla karmaşık görevleri yerine getirmek yerine sadece gözlemlediklerini iletmek için programlanmışlardır. Amaç, tek bir yonga çözümüne bahsi geçen bu ölçüm ve iletişim özelliklerini sığdırabilmektir. Şekil 2.1 tipik bir sensör düğümünün yapısını göstermektedir.



Şekil 2.1 : Sensör Düğüm Yapısı

Alıcı-verici ünite, gözlem yapılan fenomenlere ait değerlerin iletimi için kablosuz bir bağlantı altyapısı sunar. KAA'da genel olarak kullanılan standart iletişim teknolojileri IEEE802.11, IEEE802.15.1/Bluetooth, WiBree, IEEE802.15.4, ZigBee, Wireless HART, ISA-SP100.11, Z-Wave ve 6LoWPAN'dır [5]. Sensör düğümlerde genellikle sabit ve sınırlı kapasiteli piller kullanılmaktadır. Uygulama alanına göre Zink-Karbon (Z-C), Alkalin, Nikel-Kadmiyum (Ni-Cd), Lityum ve Lityum-Iyon veya Nikel-Hidrojen (Ni-H) piller kullanılabilir [6]. Çoğunlukla bu pil türleri, kullanım esnasında çevresel kısıtlardan ötürü tekrar şarj edilememekte veya değiştirilememektedir. Fakat güneş panelleri, akustik veya piezo-elektrik dönüştürücüler gibi enerji kazanımlı mekanizmaların KAA'a entegrasyonu ile pil kullanım süreleri uzatılabilir [7,8,9]. Sensör düğümlerin yerleri kullanım esnasında çoğunlukla değiş(tirile)mediğinden, güç yönetimi için KAA'daki ağ topolojisi dinamik olarak değiştirilmektedir. Buna ek olarak, düğümler belirli şartlarda alıcı-verici ünitelerini kapatarak enerji tasarrufu sağlarlar. Fakat bu şartlarda bahsi geçen düğümün ağdan bağlantısı kesilmiş olur. Sensör düğümlerin yaşan durumlarının %95'i uyku halinde geçmektedir. Bu da KAA'daki en önemli sorun olan; en az enerji harcayarak en fazla iletişimin sağlanması durumunu doğurur. Tipik bir KAA'da, ölçüm yapan tüm düğümler bilgiyi iş birliği yaparak daha güçlü bir düğüme; merkez düğüme (sink) iletir. Merkez düğüm ise bu bilgiyi kapalı ağ'ın dışına çıkartarak kullanılabilir hale getirir.

KAA'daki anahtar bileşen olan düğümler, ilk testlerin yapıldığı zamandan günümüze kadarki olan süreçte büyük gelişim göstermiştir. Araştırmalarda ve ticari uygulamalarda çok fazla marka ve model sensör düğümler kullanılmaktadır. MicaZ, Cricket, Iris, Telos, SunSPOT, ve Imote2 en bilindik modellerdir [10]. Ancak genel olarak mevcut sensör düğümleri, özellikleri ve yeteneklerine göre yalın ve üst düzey olmak üzere iki başlık altında sıralanabilir. Tablo 2.1'de popüler platformların işlemci, hız, bellek miktarı, iletişim sistemi gibi temel karakteristikleri sunulmaktadır.

Tablo 2.1 : Sensör Düzüm Özellikleri

Düzüm tipi	İşlemci hızı (MHz)	Prog. Mem. (kB)	RAM (kB)	Radyo fre. (MHz)	Tx. Oran (kbps)
Berkeley					
WeC	8	8	0.5	916	10
rene	8	8	0.5	916	10
rene2	8	16	1	916	10
dot	8	16	1	916	10
mica	6	128	4	868	10/40
mica2	16	128	4	433/868/916	38.4 kbaud
micaz	16	128	4	2.4 GHz	250
Cricket	16	128	4	433	38.4 kbaud
EyesIFX	8	60	2	868	115
TelosB/Tmode	16	48	10	2.4 GHz	250
SHIMMER	8	48	10	BT/2.4 GHz <sup>a</sup>	250
Sun SPOT	16-60	2 MB	256	2.4 GHz	250
BTnode	8	128	64	BT/433-915 <sup>a</sup>	Varies
IRIS	16	128	8	2.4 GHz	250
V-Link	N/A	N/A	N/A	2.4 GHz	250
TEHU-1121	N/A	N/A	N/A	0.9/2.4 GHz	N/A
NI WSN-3202	N/A	N/A	N/A	2.4 GHz	250
Imote	12	512	64	2.4 GHz(BT)	100
Imote2	13-416	32 MB	256	2.4 GHz	250
Stargate	400	32 MB	64MB	2.4 GHz	Variables <sup>b</sup>
Netbridge NB-100	266	8 MB	32 MB	Variables <sup>b</sup>	Variables <sup>b</sup>

### 2.2.2. Yalın Platformlar (Low-End Platforms)

Yalın (düşük-uç) platformlar, işlemci, bellek ve iletişim açısından sınırlı yetenekleri ile bilinmektedirler. Bu platformlar çoğunlukla çok sayıda düğümün ölçüm yapması gerektiği senaryolarda kullanılmaktadırlar. Yalın platformların işlem yeteneğinin sınırlı olmasından sebep çoğunlukla iletişim kabiliyetleri üzerine çalışmalar yapılmaktadır. En bilindik yalın platformlar Mica ailesi sensör düğümler, Telos/Tmote ve EYES'dır. Bu düğümler genellikle maliyet ve enerji tüketimini azaltmak için düşük güç tüketimli mikro denetleyiciler ve alıcı-vericiler ile donatılmıştır. Yoğunlukla kullanıldıkları alanlar; endüstriyel, tıbbi ve bilimsel çalışmalardır. Daha özel olarak, en son platformlar 2.4GHz bandında çalışır ve IEEE 802.15.4 standardı ile uyumlu olan CC2420 alıcı-verici üniteye sahiptir. Bu da pek çok ülkede lisans ücreti ödenmeden kullanılabilen iletişim bantlarının tahsis edilebilmesine imkân sağlamaktadır.

KAA'ların işletildiği çalışmaların çoğunluğu yalın platformlar kullanılarak gerçekleştirilmektedir.

### 2.2.3. Üst Düzey Platformlar ( High-End Platforms )

Bazı senaryolarda yalın platformların ölçüm, iletişim ve lokal işlem yeteneklerine ek özellikler ihtiyaç duyulmaktadır. Yalın platformların yetersiz kaldığı bu durumlarda yüksek işlemci ve bellek kapasitesi sunabilen ve yazılımsal veya donanımsal geliştirmeye imkân veren üst düzey ( yüksek-uç ) platformlar tercih edilmektedir. Stargate ve Imote en bilindik üst düzey sensör platformlardır.

### 2.2.4. KAA'ların Kullanım Alanları

KAA'nın ortaya çıkması birçok alanda çeşitli araştırmaları tetikledi. Bu bölüm, mevcut KAA'nın akademik ve ticari olarak geliştirilip kullanılmasına genel bir bakış sunmaktadır.

KAA'lar; sismik, manyetik, termal, görsel, kızılötesi, akustik, ve radar sensörleri gibi bir çok farklı sensor tipini içerir [1, 11] ve sıcaklık, nem, basınç, hız, yön, hareket, ışık, toprak bileşimi, titreşim, gürültü seviyesi, belirli tür nesnelere varlığı ya da yokluğu, nesnelere üzerindeki mekanik stres seviyeleri gibi çeşitli fiziksel yada çevresel ortam

koşullarını izlemek için kullanılabilirler. Bu bağlamda hayatı kolaylaştırma adına birçok uygulama geliştirilebilir; güvenlik, istihbarat toplama, savunma, savaş, hava durumu tahminleri, sıcaklık ölçümü gibi birçok alanda geliştirmeler yapılabilir. Artık hayatımızın içine girmiş ve gün geçtikçe de artan uygulamaları beş ana başlık altında toplarsak bunlar; askeri, çevresel, sağlık, ev ve sanayi uygulamaları olarak karşımıza çıkar.

### **2.2.5. Askeri Uygulamalar**

Askeri alandaki ihtiyaçlar göz önüne alındığında KAA'nın kullanımı hem düşük maliyetli hem de geleneksel yöntemlere göre çok daha güvenlidir. Günümüzde askeri komuta zincirinin ayrılmaz bir parçası olan KAA'lar; kontrol, iletişim, izleme, keşif, istihbarat, hasar değerlendirme, biyolojik ve kimyasal saldırı tespiti, hesaplama ve hedef tespit sistemlerinde yoğun olarak kullanılmaktadırlar [12, 13, 14].

### **2.2.6. Çevresel Uygulamalar**

Belirli bir alana yayılan sensörler otonom özelliklerinden dolayı çevre kontrolü veya izleme alanlarında kullanılabilirler. En yaygın kullanılan çevresel sensör çalışmalarına bakarsak, doğada hayvanların hareketlerinin izlenmesi, tarım ve hayvancılığı etkileyen çevre koşullarının izlenmesi, toprak, deniz, atmosferik bağlamlarda çevresel izleme, orman yangınlarının tespiti, meteorolojik veya jeofizik araştırmaları, sel tespiti, çevrenin biyolojik karmaşıklığını haritalama ve çevresel kirlilik gibi birçok başlık sayılabilir [15, 16, 17, 18, 19, 20].

### **2.2.7. Sağlık Uygulamaları**

Biyolojik sistemlerle kombine edilebilen sensörler sağlık alanında önemli bir kullanım alanına sahiptir. Hastalıkların tanı, tedavi ve izlenmesinde etkin olarak kullanılan sensörler; entegre hasta izleme, teşhis, hastanelerde ilaç yönetimi, insan fizyolojik verilerinin izlenmesi, engelliler için ara yüzlerin sağlanması, hastane içinde hasta ve doktorları takip ve izlenmesi gibi faaliyetlerde kullanılabilirler [21, 22, 23, 24, 25].

### **2.2.8. Ev Uygulamaları**

Evimizde bulunan büyüklü küçüklü tüm cihazlara gömülebilen sönörlerle birlikte artık akıllı ev kavramı hayatımıza iyiden iyiye girmiş bulunuyor. Aklımıza gelebilecek hemen her cihaza bütünleşmiş sönörlerin birbirleri ve dış dünya ile internet veya uydu üzerinden iletişime geçilebilmesi ile uzaktan veya lokalden yönetilmesine olanak sağlanmaktadır. Ayrıca KAA'lar uzaktan ölçüm ile gaz, elektrik, oda sıcaklığı gibi verileri istenilen noktaya ileterek evdeki güvenliği de sağlamış olmaktadırlar.

### **2.2.9. Endüstriyel Uygulamalar**

KAA'ların yoğun kullanımı ile birlikte ticari uygulamalarda da örnekleri görülmüştür. Kablolulu sönör ağlar endüstride uzun süredir kullanılmakta olan bir teknolojiydi fakat kurulum maliyetlerinin yüksek oluşu ve yer değişikliklerine açık olmayışlarından sebep bazı kullanımsal kısıtları bulunuyordu. Bu kısıtların uygulamalardaki etkilerini ortadan kaldırmak için kablosuz iletişim yapabilen sönör cihazları kullanılmaya başlanmıştır. Bu anlamda KAA'lar; envanter yönetimi, malzeme yoğunluklarının izlenmesi, ürün kalitesinin izlenmesi, çevre kontrolü, interaktif oyuncaklar, etkileşimli müzeler, robot kontrolü, fabrika otomasyon izleme ve kontrol, araç izleme gibi bir çok endüstriyel uygulamalarda kullanılmıştır [26, 27, 28, 29, 30, 31].

## **2.3. KABLOSUZ ALGILAYICI AĞLARDAKİ VERİYE ULAŞMAK**

Kablosuz algılayıcı ağların verimli çalışabilmesi için ihtiyaçlara uygun konfigürasyonlar yapılması gerekmektedir. KAA'lar doğru tercih edilecek kurulum parametreleri ile günümüzde çok büyük ihtiyaçlara yanıt verebilmektedir. Bu sayede yapılan ölçüm ve gözlemler; endüstriyel, askeri, tıbbi, çevresel ve araştırma alanlarında artan sıklıkla kullanılmaktadır. KAA'lardaki sönör düğümlerin yapıları gereği hesaplama, depolama ve uzun mesafeli iletişim yeteneklerinden yoksun olmaları, ölçüm verilerine ulaşmayı zorlaştırmaktadır. Günümüzde geleneksel KAA verilerin daha geniş alana taşınabilmesi, üzerinde çalışılmakta olan önemli bir araştırma konusudur. Ölçüm verilerinin anlık değerlerinin sınırlı sayıda kullanıcıya gösterilmesinden ziyade, geçmişe dönük sorguların ve büyük ölçekli analizlerin yapılabildiği bir sistem üzerinden geniş bir kullanıcı kitlesine sunulabilmesi sağlanmalıdır. Hem araştırmacılar, hem de sanayiciler bu konu ile ilgili pek çok çalışma gerçekleştirmişlerdir. Bu makalede



günümüze kadar yapılmış olan çalışmalar ve bu çalışmalarda kullanılan teknolojiler anlatılmaktadır.

### 2.3.1. KAA'lar İçin Tasarlanmış Ara Katmanlar (Middleware)

DCOM, CORBA, PVM ve MPI gibi geleneksel dağıtık ara katmanlar bellek ve hesaplama işlemlerinde kaynak tüketimlerinin fazla olmalarından dolayı KAA'lar için uygun değildirler. KAA'larda ara katmanın amacı kurulum, ölçüm, yürütme ve bakım faaliyetlerini desteklemektir. Ölçüm yapan düğümlerin koordinasyonuna ve hatalarının tolere edilmesine imkân vermektedir. Ayrıca türdeş olmayan düğümlerin sistem içinde sorunsuz ve tam uyumlu olarak çalışmasından da ara katman sorumludur. Araştırmacılar günümüze kadar pek çok farklı ara katman tasarımları hazırlamışlardır. Mevcut çalışmaları beş ana başlık altında gruplayabiliriz.

- Veri tabanından Esinlenilmiş Çözümler (Database-Inspired Solutions)
- Yayınla / Katıl Sistemler (Publish / Subscribe Systems)
- Servis Keşfi Tabanlı Yaklaşımlar (Service Discovery Based Approaches)
- Demet Alanı Çözümleri (Tuple Space Solutions)
- Sanal Makine Yaklaşımları (Virtual Machine Approaches)

### 2.3.2. Veritabanından Esinlenilmiş Çözümler

Geleneksel veritabanı mantığından bakıldığında, algılayıcı ağlar bir dağıtık veritabanı sistemi olarak değerlendirilebilir. Uygulama seviyesinde algılayıcı düğümleri SQL benzeri bir dil kullanır gibi sorgulama yapılmasına imkan veren ara katmanlar bulunmaktadır. Bu tarz çalışan ara katmanlardan en bilindik olanları COUGAR [32] ve TinyDB [33,34]'dir. Her iki sistem de ölçüm yapan düğümlerin verilerini sorgulamaya imkân vermektedir. TinyDB, TinyOS işletim sistemi kullanan düğümler üzerinde programlama yapmadan sorgular oluşturmanıza izin vermektedir. Aşağıdaki kod bloğundaki örnekte olduğu gibi, sorgulama SQL benzeri bir dil ile yapılmaktadır.

```
SELECT nodeid, light, temp
FROM sensors
SAMPLE INTERVAL 1s FOR 10s
```

Şekil 2.2 : TinyDB Sorgu Örneği

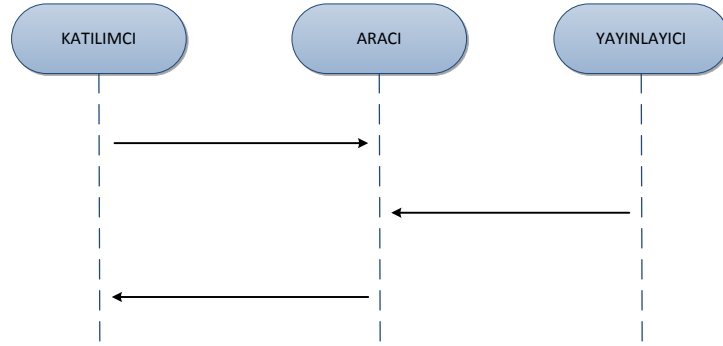
COUGAR ise Cornell Üniversitesi tarafından geliştirilmiş olan bir sistem olup TinyDB gibi sensörlerin sorgulanmasına imkan vermektedir. Sensör verileri bir zaman serisi (time series) olarak değerlendirilir ve bir zaman damgası (timestamp) ile eşleştirilmiştir. Sensör düğümler üzerinde soyut veri tipi (abstract data type) oluşturularak, içerilen (encapsulated) verinin biri dizi fonksiyon ile sorgulanması mümkün olmaktadır.

Ara katmanların veritabanı gibi çalıştığı bir başka sistem ise SINA [35]'dir. SINA, COGAR ve TinyDB'nin aksine bir veritabanı sistemi değildir. Kullanıcılara bir veri modeli sunmaktadır. Ölçüm verileri ilişkili tablolarda (associative spreadsheet) saklanmaktadır. Ayrıca sorgulama dili olan SQLT (Sensor Query and Tasking Language) ile sensör donanımına ve iletişimine direkt olarak erişim sağlanabilmektedir.

### 2.3.3. Yayınla / Katıl Sistemler

KAA'ların biriktirdikleri ölçüm verilerinin daha verimli kullanılabilmesi, başka ağlara ve başka uygulamalara aktarılabilmesi ile mümkün olmaktadır. Dolayısıyla veri aktarımı sağlayabilmek için KAA'lar geleneksel ağlar ile bütünleşmiş edilmelidir. KAA ile geleneksel ağ arasında aktarım fiziksel olarak geçit (gateway) cihazlar üzerinden yapılabilmektedir. Geleneksel ağ üzerindeki cihazların adresleri bilinirken çoğu zaman KAA tarafındaki düğümlerin adresleri bilinmemektedir. Sensör düğümler isteyerek veya istemeyerek bazen adreslerini değiştirebilmekte ve pillerinin bitmesi durumunda ise ulaşılamaz olmaktadır. Bu durumda aynı ölçüm verisini sağlayabilecek bir düğümün tespiti yapılmalıdır. Bu tarz sorunların çözümü için veri-merkezli (data-centric) iletişim yöntemleri kullanılmalıdır. Yayınla/Katıl (pub/sub) mesajlaşma yapan sistemler, veri-merkezli iletişimin en bilindik örneklerinde biridir. Ölçeklenebilir oluşu ve dinamik uygulama topolojisine imkân verdiği için kurumsal ağlarda sıkça kullanılmaktadırlar.

Yayınla/Katıl iletişim modelinin temel çalışma prensibi, objelerin ilgi duydukları veriyi sunan kaynaklara kayıt yaptırmalarıdır. Bu kayıt işlemine katılma (subscription) ve katılmayı gerçekleştirene de katılımcı (subscriber) denilir. İlgi duyulan konuya ait verileri ölçümleyip duyuran objelere ise yayıncı (publisher) denilmektedir. Yayıncı ve katılımcı arasındaki ilişki düzenleyen ve iletişimin gerçekleşmesine imkân veren objeye de aracı (broker) denilmektedir. Bu üç bileşen arasındaki iletişim Şekil 2.3'te sunulmuştur.



Şekil 2.3 : Yayınla Katıl İletişim Modeli

Mires [36] KAA için yayınla/katıl mimarisidir. Çalışma şekli, algılayıcı düğüm yaptığı ölçümü katılımcı kullanıcıya iletmesidir. Yayınlama mesajları biriktirilerek, katılımcılara toplu olarak da sunulabilmektedir.

The Global Sensor Network (GSN) [37] bir omurgaya üzerinden birbirlerine bağlı olan KAA'ların bağımsız olarak sorgulanmasına Yayınla/Katıl iletişim modeli üzerinden imkân vermektedir.

Bir başka bilindik yayınla/katıl sistemi olan TinySIP [38] Oturum Başlatıcı Protokol'ün (Session Initiator Protocol (SIP)) [39] fonksiyonalitesini KAA'lara taşımaktadır. Diğer bir önemli özelliği ise çoklu geçit kullanımına izin vermesidir.

Messo and Preso [40] birbirini tamamlayıcı olarak çalışan iki ayrı yayınla/katıl protokolüdür. Messo düğümlerden veri toplanmasına imkân vermektedir. Preso ise toplanan verinin ilgili nesnelere iletimini sağlamaktadır.

MQTT-S [41] IBM KAA sına ortamı [42] üzerinde geliştirilmiş bir başka yayınla/katıl yaklaşımdır. Sistem içerisindeki istemci C, geçit uygulamaması ise Java programlama dili ile geliştirilmiştir. Geçit istemcinin ilgilendiği verilere ait oturumu muhafaza eder ve aracıya iletir.

### 2.3.4. Servis Keşfi Tabanlı Yaklaşımlar

Daha önceki bölümlerde anlatılan ara katman yaklaşımlarına farklı olarak, MILAN (Middleware Linking Applications and Networks) [43] isteğe bağlı (arbitrary) protokolleri bir eklenti mekanizması ile sisteme dâhil etmektedir. Uygulamalar ölçüm yapabilmeleri için gerekli olan ihtiyaçlarını ara katmana bir API vasıtasıyla iletmektedirler. Ara katman uygulamalardan gelen bilgiler doğrultusunda servis kalitesi (QoS), enerji verimliliği, ölçüm sıklığı gibi parametrelere karar vererek, sistemin etkinliğini arttırmaya çalışır.

### 2.3.5. Demet Alanı Çözümleri

Bir veritabanı terimi olan demet (tuple), bir veriye ait bilgileri taşıyan kayıttır. KAA'lar için düşünüldüğünde ise sensör düğümün yaptığı bir ölçüme ait bilgiler topluluğudur. Örneğin sıcaklık ölçümü yapan bir düğümün oluşturacağı demet <düğümNo,düğümYeri,zamanDamgası,sıcaklıkÖlçümü> şeklinde olacaktır. Demetin biçimi uygulamalar arasında farklılık göstermektedir. Demetler ölçümler yapıldıkça sanal tablolar üzerinde saklanır ve bu tablolar üzerinde sorgulanması sağlanır.

TinyLIME [44] bir demet alanı çözümdür ve LIME [45] ara katmanının Gezici Amaca Yönelik Ağ'lar (MANET) için düzenlenmiş halidir. Mobil hesaplama yapan cihazlar ile sensörler arasında iletişime imkân vermektedir. Sensör düğümlerin kendi aralarında iletişim yapmayacağı öngörülerek, tek atlamalı (single-hop) bir iletişime izin verilmektedir.

### 2.3.6. Sanal Makine Yaklaşımları

Bu yaklaşımda, sistem sanal makineler (Virtual Machines), yorumlayıcılar (interpreters) ve mobil etmenler (mobile agents) barındırabilmektedir. Sanal makine kullanımının en önemli sebebi türdeş olmayan donanımlar için uygulamaların değiştirilme ihtiyacı olmamasıdır. Sistem bu sayede fiziksel soyutlama yetisi kazanmış olmaktadır.

Mate [46] sanal makineler kullanılarak çok uzun programların 100byte'lık küçük modüllere ayrıştırılarak çalıştırılması sağlamıştır.

SensorWare [47]'de sanal makinelerin tanımlanmasına, kullanılmasına ve desteklenmesine imkân vermektedir. Her bir modül veya servis bir sanal makine olarak temsil edilmektedir. Sistemdeki bütün cihazlar için sabit bir ara yüz bulunmaktadır. Bu ara yüzde iletişim için kullanılan dört özel komut barındırılmaktadır; sorgu (query), eylem (act), ilgi (interest) ve ortadan kaldırma (dispose).

SwissQM [48] sensör düğümlerde kullanılan nesC gibi alt-seviye programlama dillerinin zorluğuna alternatif sunmaktadır. Sistem üst-seviye program bloklarını sanal makine üzerinde çalıştırabilmektedir.

DAIS [49] (Declarative Applications in Immersive Sensor networks) sistemi ise uygulama geliştiricilerin algılayıcı cihazlar ile direkt olarak çalışabilen kodlar yazmasına imkân vermektedir.

### **2.3.7. Genel Ara Katman Değerlendirmesi**

Günümüzde KAA'ların sunduğu imkânları daha geniş alanlara yaymak için çok çeşitli ara katmanlar kullanılmaktadır. Bu çalışmada KAA'larda kullanılan ara katman teknolojilerinin en son gelişmeleri beş ana başlık altında incelenmiştir. Veritabanından Esinlenilmiş Çözümler, Yayınla / Katıl Sistemler, Servis Keşfi Tabanlı Yaklaşımlar, Demet Alanı Çözümleri ve Sanal Makine Yaklaşımları'na ait çalışmalar özetlenmiştir.

Bütün bu çalışmalar göstermektedir ki, geleceğe yönelik araştırmalar ağırlıklı olarak melez sistemler üzerine kurulacaktır. Sanal makine yaklaşımının giderek yaygınlaşacağı ve Yayınla / Katıl Sistemler kullanılarak KAA verilerinin Bulut Bilişim ile entegrasyonu sağlanacaktır.

## **2.4. BULUT BİLİŞİM**

Bulut Bilişim terimi Servis Odaklı Mimari (SOA) içerisinde kullanılmakta olan eski kavramlara yeni bir boyut getirmiştir. Web Servisler, Grid Computing ve Utility Computing temelleri üzerine kurulmuş olan ve sanallaştırma (Virtualization) desteği ile sunulan bu teknoloji günümüzün en büyük sorunu olan kaynak yönetimine çözüm sunmaktadır. Bu bağlamda bulut bilişim ağı üzerinden farklı soyutlama seviyelerinde kullanıcılara ihtiyacı olan hizmetlerin verilmesini sağlayan yeni bir bilgisayar mimarisi olarak kabul edilebilir. Bulut bilişim için literatürde yapılmış pek çok tanım

bulunmaktadır. En sade tanımlardan bir tanesi; genellikle internet gibi bir ağ üzerinden talep anında kaynak ve servislerin sağlanması olarak ifade edilebilir [50]. Bulut bilişim altında sunulan servis ve kaynakları 3 farklı model altında kategorize edebiliriz [51, 52]; Yazılımın Servis olarak Sunulması, Platformun Servis olarak Sunulması ve Altyapının Servis olarak Sunulması. Fakat bulut bilişimin üzerine kurulduğu teknolojiler ile farkı tamamen bilinmemekte ve çoğunlukla karıştırılmaktadır. Her bir modelin mevkidaş SOA teknolojisi ile kıyaslamaları alt bölümler içinde açıklanmıştır.

#### **2.4.1. Yazılımın Servis Olarak Sunulması (Saas)**

Yazılımın servis olarak sunulması (SaaS) modelinde, kullanıcıların ihtiyacı olan yazılımı internet tarayıcıları (browser) ile kullanması sağlanmaktadır. Çoğunlukla metin editörü, hesap tabloları, ajanda gibi ofis uygulamalarında ve Müşteri İlişkileri Yönetimi (CRM) uygulamalarında kullanılmaktadır. En önemli avantajı kullanıcılara platform bağımsız olarak uygulamayı tarayıcı olan herhangi bir cihazdan çalıştırabilme özgürlüğü vermesidir. Bu anlamda istemciler bulut'da sunulan bu uygulamalara sadece masaüstü bilgisayarları ile değil; akıllı telefonlar, tablet bilgisayarlar, el terminalleri gibi pek çok mobil cihaz ile erişebilmektedirler. SaaS modelinin hareketliliğe sunduğu bu katkının yanı sıra diğer bir önemli özelliği ise sürüm yönetimidir. Uygulamalar lokal sistemlerde çalışmadıklarından, yeni sürüm geçişleri kullanıcılara hissettirilmeden ve hizmeti aksatmadan sağlanabilmektedir. SaaS uygulamalarından en bilindik örnekler Google uygulamaları, Salesforce.com, Yahoo eposta hizmeti, WebEx ve Microsoft Office Live'dır. Bütün bu olumlu özelliklerin aksine, SaaS modelinde güvenlik, ölçeklenebilirlik, istemcilerin senkronize edilmesi, ücretlendirme yönetimi ve son kullanıcı desteği gibi konular halen geliştirilmeye müsait araştırma konuları olarak bilinmektedir.

SaaS modeli çoğunluk SOA'nın en bilindik şekli olan Web Servisleri ile kıyaslanmaktadır. Temel fark SaaS'in web bağımlı olması ve fonksiyonitesinin tarayıcı tabanlı yazılımlar aracılığıyla sunulmasıdır. Web servisleri [53] ise tarayıcı tabanlı olmayıp SOAP protokolü üzerinde çalışmaktadırlar

#### **2.4.2. Platformun Servis Olarak Sunulması (Paas)**

Platformun servis olarak sunulması (PaaS) modeli, bulut barındırma hizmeti veren servis sağlayıcılardaki işletim sistemlerinin, veri tabanlarının, ara katmanların

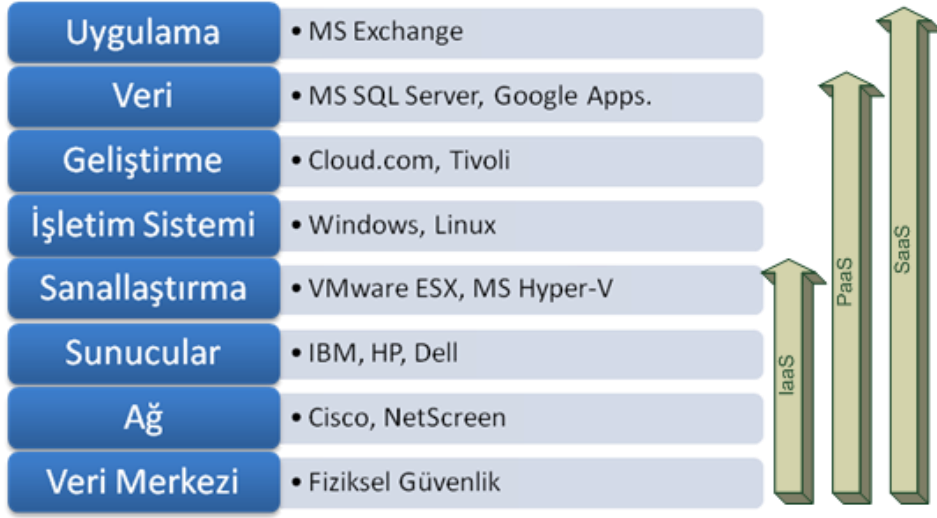
(middleware) ve web sunucuların uzaktan istemcilere kullanılması ve tahsis edilmesi hizmetidir. Sanallaştırma teknolojisinin [54] en çok kullanıldığı bulut bilişim modelidir. Sanallaştırma ile sunulan bu ortamlar tek bir fare tıklaması ile küçültülüp genişletilebilmektedir. Bu modelin servis sağlayıcılar için en önemli özelliği keskin doğrulukta ücretlendirme yeteneğidir. Bulut üzerinde çalışan sanal işletim sistemlerinin kullandığı kaynaklar (bant genişliği, disk kapasitesi, vb.) kesin olarak ölçülebilmektedir. Kullanıcı tarafında ise; orta ve büyük ölçekteki firmaların, ihtiyaç duydukları sunucuyu kurum bünyesinde kurmak ve barındırmak yerine bulut üzerinde çalıştırmasının hem ulaşılabilirlik hem de servis kalitesi anlamında artıları bulunmaktadır. Ayrıca müşterilerin, standart faturalandırma şekli olarak senelik veya ömür boyu lisanslar satın almalarına gerek kalmamaktadır. İhtiyaçları olan platformların lisans ücretlerinin tamamının finanse edilmesi yerine aylık, hatta günlük kullanım bedellerinin ödenmesi mümkün olmaktadır. Google'ın Uygulama Motoru, Microsoft Azure, Amazon Web servisleri ve Force.com bu model için market lideri firmalar olarak bilinmektedirler.

#### **2.4.3. Altyapının Servis Olarak Sunulması (IaaS)**

Altyapının servis olarak sunulduğu modelde (IaaS) müşteriler ödediği kadar kullan mantığında işlemci, bellek, depolama alanı ve bant genişliği gibi bilgisayar kaynaklarını bulut üzerinden temin edebilmektedir. Bu şekilde sunulan kaynakların yönetimi için özel bulut mimarileri kullanılmaktadır. Bu modelin en dikkat çekici sağlayıcıları Amazon'un EC2'si, GoGrid'in Bulut Sunucuları ve Joyent'dir. IaaS'ye ilişkin en bilindik problem uygulamaların taşınabilir olmamasıdır.

Her üç model incelendiğinde görülmektedir ki; SaaS modeli bulut bilişimdeki temel başlangıç adımıdır. Firmaların kullandıkları veya ihtiyaç duydukları yazılımları, kurulum ve işletme maliyetleri olmadan tarayıcılar üzerinden cihaz bağımsız bir şekilde kullanabilme imkanı sağlamaktadır. Öte yandan PaaS modelinde, BT altyapısını genişletmek isteyen firmalara büyük bir erişim imkânı sunmaktadır. Bulut üzerinde kullanıma alınacak sunucuların fiziksel ve yazılımsal yönetimleri, yerel olarak çalışmalarından çok daha pratik olacaktır. IaaS modeli ise işlemci ve disk alanı gibi kaynaklara ihtiyaç duyan, fakat BT altyapısını büyültmek istemeyen firmalar için önemli bir çözüm olmaktadır.

Bu üç modele ilişkin; müşteri ile servis sağlayıcısı arasındaki katmanlara göre sorumluluk paylaşımını Şekil 2.4’de gösterilmektedir.



Şekil 2.4 : Bulut Bilişim Modelleri

#### 2.4.4. Bulut Bilişim Ortamlarının Değerlendirilmesi

Verilerin saklanacağı ve hesaplama işlemlerinin gerçekleştirileceği Bulut Bilişim Ortamı tercih edilirken 4 farklı alternatif değerlendirilmiştir; Windows Azure [55], Google App Engine [56] (GAE), Amazon Web Services [57] ve AppHarbor [58]. Bu bölümde bu teknolojilerin çalışma metodolojileri detaylandırılmıştır.

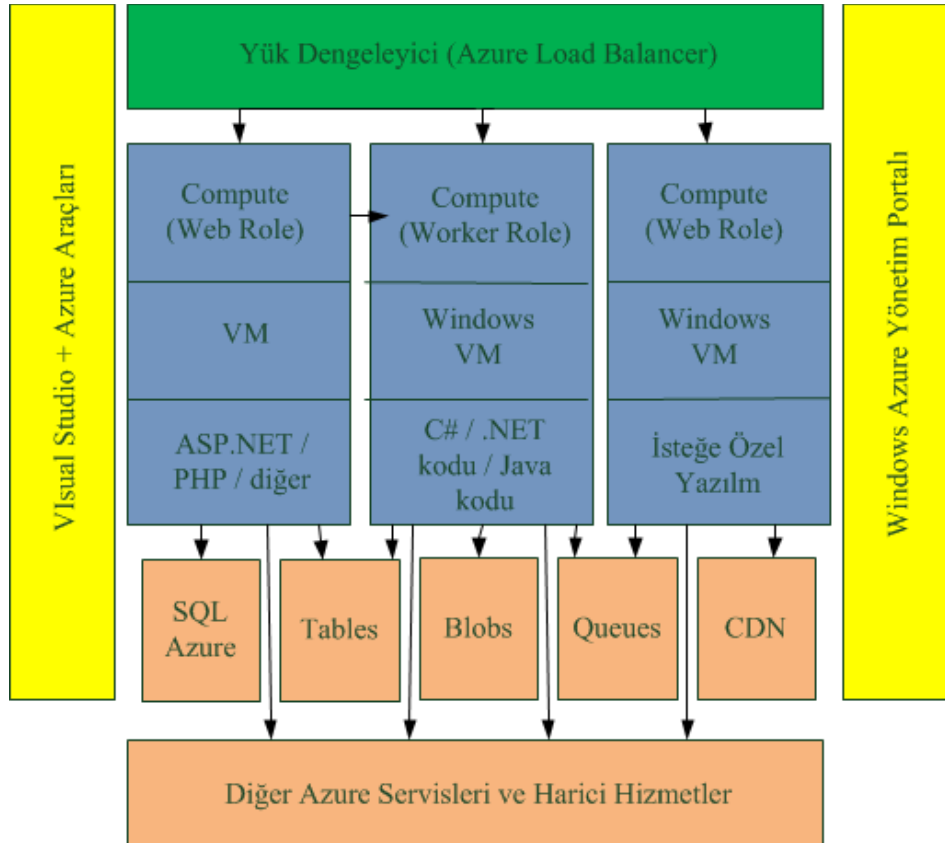
#### 2.4.5. Microsoft Azure

Windows Azure Microsoft tarafından sunulan ve dünyada en hızlı gelişim gösteren Bulut Bilişim ortamı olarak karşımıza çıkmaktadır. Çok geniş PaaS platformu sunmaktadır ve .NET geliştiricileri için web ve Windows tabanlı kütüphane kullanılabilmektedir. Şu ana kadar geliştirilmiş olan tüm .NET teknolojileri (ASP.NET MVC, WCF, WWF, ADO.NET vb.) ile uyumlu çalışması en önemli avantajıdır. Fakat programlama desteği sadece .NET platformu ile sınırlı değildir, Java, PHP ve Node.js API'lerini de kullanmaktadır. Ayrıca 2012 yılı itibarıyla geliştiriciler için 3 aylık deneme süresi tanınmaktadır. Maalesef Türkiye deneme süresi katılımları yapabilmeyen ülkeler arasında değildir.

Azure mimarisi Şekil 3’de gösterildiği gibi istenilen hizmete ulaşmak için ilk olarak karşılaşılan bir yük dengeleyicisi ile başlar. Hizmetin tipine bağlı olarak Web Role,



Worker Role veya VM Role tipinde projeler geliştirilebilmektedir. Web role Internet Information Services (IIS) üzerinde çalışabilen web tabanlı uygulamalar ve WCF servisleridir. Bu bağlamda Azure platformunun SaaS'i olarak değerlendirilebilir. Worker Role uzun süreli hesaplamalarda kullanılan ve çoğunlukla otonom çalışması istenen servislerde tercih edilmektedir. VM role ise; istenilen servisin sanal ve kalıcı olmayan bir yapıda sunulduğu halidir. Hizmetlerin iletişim halinde çalıştığı veri saklamak için tasarlanmış olan mekanizmalar SQL Azure, Tables, Blobs, Queues ve CDN'lerdir. Azure Table Storage'lar yüksek derecede ölçeklenebilir dağıtık veritabanlarıdır ve varlıkları özellikleri ile beraber muhafaza edebilmektedir. Azure Queue Storage; mesajların kuyruk yapısında saklanmasına imkan vermektedir. Azure bileşenleri (Blob, Drive vb.) NTFS biçiminde veri saklayabilmektedir. Fakat SQL Azure, SQL Server 2012'in Bulut Bilişim teknolojisine uyarlanmış ölçeklenebilir ilişkisel veri tabanı olarak en çok tercih edilen bilgi saklama servisedir.

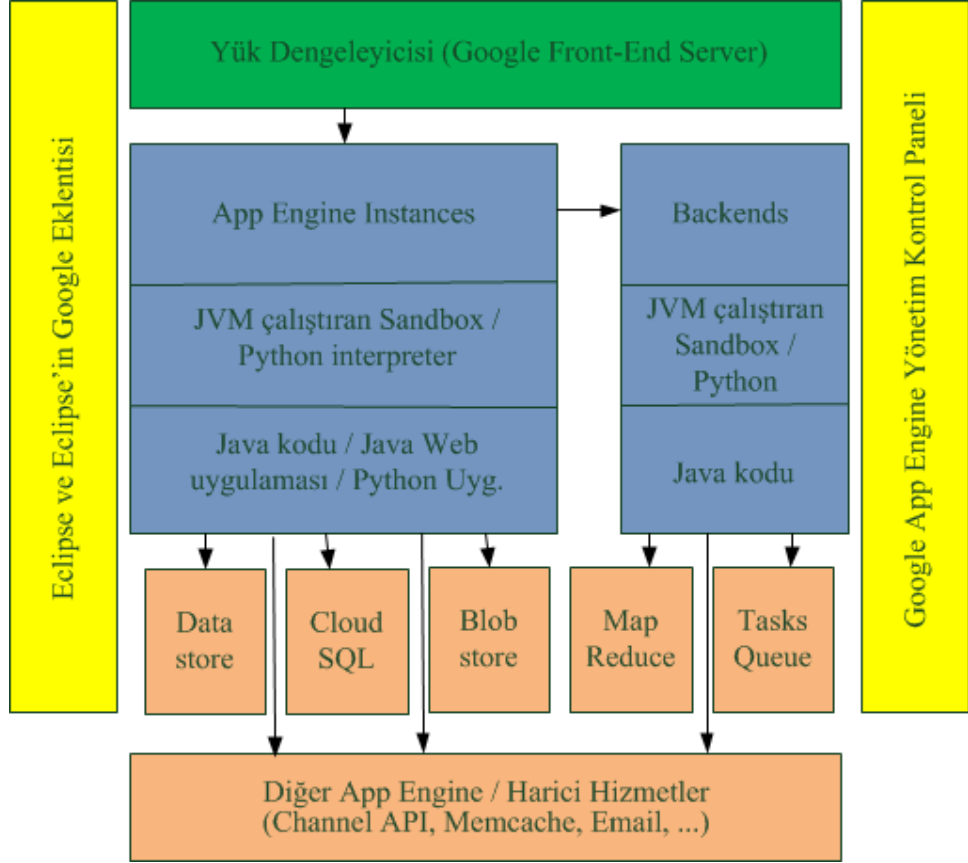


Şekil 2.5 : Azure Mimarisi

#### **2.4.6. Google Uygulama Motoru (GAE)**

GAE Google tarafından sunulan Bulut hizmetidir. Java ve Python kodları ile geliştiricilere alt yapı sunmaktadır. Altyapısı Gmail ve Google Docs'ta kullanılanlara çok benzerlik göstermektedir. GAE'nin tamamen ücretsiz olan bir versiyonu bulunmaktadır. Bu versiyona kayıt yapılması durumunda aylık 5 Milyon kere sayfa bakımı kullandırabilen CPU / Bant Geniřlięi ve Saklama Alanı hizmetinden faydalanılabilmektedir.

GAE mimarisi Google air bir ön-uç sunucusuna ulařılmasıyla bařlar. Bu sunucu bir yük dengeleyicisi gibi davranarak istemcilerden gelen talepleri daęıtmaktadır. Sunulan hizmetler eęer java tabalı ise sandbox vasıtasıyla aksi halde Python kodlarının çalıştırılması için Python yorumlayıcı (interpreter) çalışır. Ayrıca Google'a ait arka uç sunucularda Java ile MapReduce kullanılabilmektedir. MapReduce; büyük ölçekli hesaplamalar gerektiren görevlerin paralel programlama ile tamamlandıęı bir API'dir. Tüm bu hizmetlerin verilerini barındırmak için çeřitli mekanizmalar geliştirilmiřtir. En önemli veri depolama mekanizması Cloud SQL olarak karřımıza çıkmaktadır. Bu mekanizma Bulut üzerinde GAE tarafından sunulan daęıtık bir MySQL sunucusu türevidir. GAE'nin servis mimarisi ařaęıdaki Őekil 2.6'da sunulmaktadır.



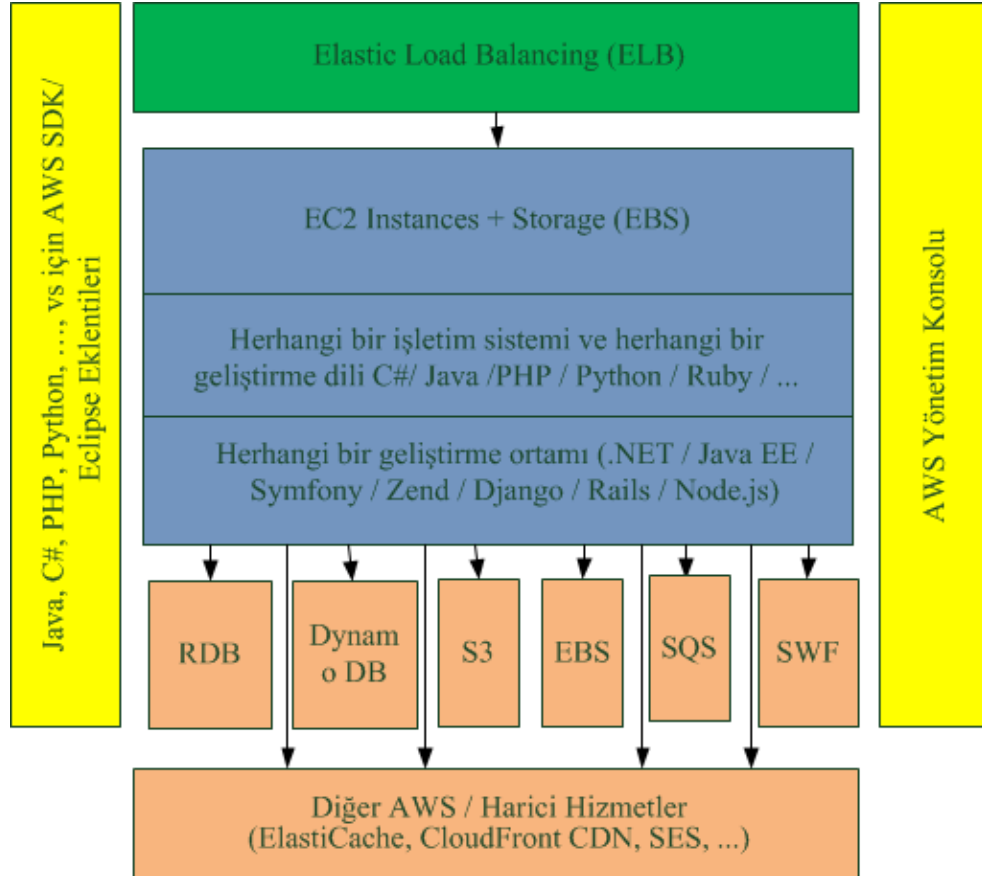
Şekil 2.6 : GAE Mimarisi

#### 2.4.7. Amazon Web Servisleri (AWS)

Amazon 2002 yılından bu yana Bulut Bilişim ile direkt olarak ilgilenmekte ve hizmetlerinde geliştirdiği teknolojileri kullanmaktadır. Umumi (public) Bulut hizmetlerinin öncüsü olarak nitelendirilebilirler ve talebe göre ölçeklenebilir IaaS ve PaaS sunmaktadırlar. Amazon Elastic Compute Cloud (EC2) coğrafi olarak Amerika, Avrupa, Japonya, Brezilya gibi çeşitlik noktalara dağılmış bir şekilde platform bağımsız bir şekilde ( Linux, Windows vb.) sanal işletim sistemi hizmeti sunmaktadır.

AWS mimarisinin detayları Şekil 2.7 'te sunulmaktadır. Elastic Load Balancing (ELB) hizmeti istemcilerden gelen talepleri yük dengelemekte kullanılmaktadır. İstemciler bu sayede EC2 örneklerine (instance) ve sanal sabit disk hizmeti veren Amazon Elastic Block Store (EBS) hizmetlerine erişebilmektedirler. Sanal makineler platform bağımsız oluşturulabildiklerinden, seçilen konfigürasyona göre geliştirme ortamı çeşitlenebilmektedir.

Amazon'un veri depolama için sunduğu çözüm Amazon Simple Storage Service (S3)'dür. S3, doküman, resim ve video formatındaki objelerin web üzerinden erişilmesine imkân vermektedir. Ayrıca veritabanı hizmeti olarak 2 farklı seçenek sunmaktadır. Amazon SimpleDB geliştirme sürecinde ilk olarak sunulan NoSQL veritabanı hizmetidir. Şuan yerini DynamoDB'ye bırakmıştır. DynamoDB, SimpleDB'ye oranla daha hızlı sorgu süreleri döndürmektedir. Bulut üzerinde ilişkisel veritabanı kullanmak için tercih edilecek hizmetin adı, Amazon Relational Database Services (RDS)'dir. Bulut üzerindeki MySQL ve Oracle veritabanlarının yönetilebildiği bu servis, replikasyon, yedek alma ve ölçekleme özelliklerine de sahiptir.

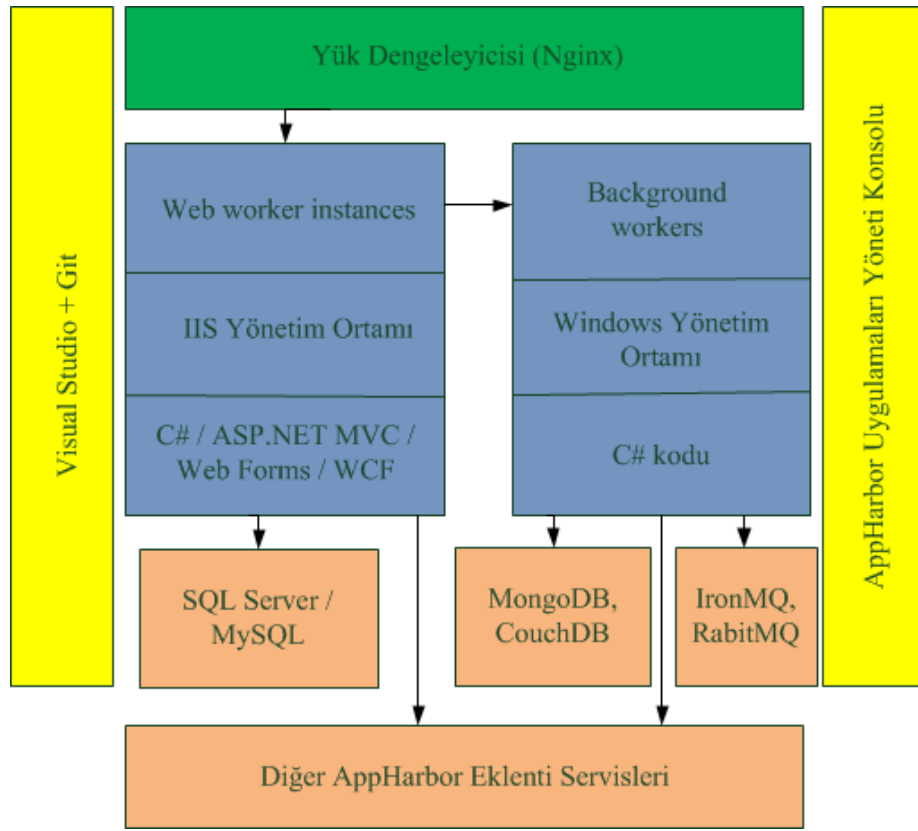


Şekil 2.7 : AWS Mimarisi

Diğer Amazon Hizmetleri; mesaj kuyruklarının kullanıldığı Amazon SQS, içerik dağıtım ağı (CDN) olarak kullanılan Amazon CloudFront, ön bellekleme hizmeti Amazon ElastiCache, DNS kayıt yöneticisi Amazon Route 53, eposta servisi Amazon SES olarak bilinmektedir. Tüm bu hizmetlerden 1 senelik deneme süresine kayıt yaptırarak faydalanılabilmektedir.

### 2.4.8. Appharbor

AppHarbor .NET uygulamaları için bir Bulut Bilişim platformudur. Geleneksel .NET geliştirme ortamında türetilerek oluşturulmuştur ve Amazon Web Services üzerine kuruludur. Geliştirilen uygulamalarda kurulum Git, SVN ve TFS vasıtasıyla yapılmaktadır. Kurulumlar otomatik oluşturma (build) işlemini barındırmaktadır ve birim sınamaya (unit test) imkânı bulunmaktadır. AppHarbor'un çalışmam mimarisi Şekil 2.8'de sunulmuştur.



Şekil 2.8 : AppHarbor Mimarisi

AppHarbor'un en çok öne çıkan özelliği oldukça geniş bir eklenti paketlerine sahip olmasıdır. Eklenti paketlerinden başlıcalar; hata loglama için Airbrake, performans gözlemi için Blitz, e-posta yönetimi için Mailgun ve veritabanı hizmetlerinde kullanılabilen Cloudant, JustOneDB, MongoHQ, Raven HQ, Shared SQL Server olarak sıralanabilir.

#### **2.4.9. Seçilen Bulut Bilişim Platformu: Microsoft Azure**

Detaylı olarak incelediğim Bulut Bilişim Platformları arasında en çok öne çıkanlar Microsoft Azure ve Amazon Web Services olmuştur. Google App Engine'in paralel programlamadan yoksun oluşu, ilişkisel bir veritabanı hizmeti sunamayışı ve web uygulamalarının geliştirme süreci zahmetli oluşu rakipleri arasında geri bırakmaktadır. Benzer şekilde AppHarbor'ın melez yapısının hali hazırda oturmuş olmayışı ve geliştirme sürecinin devam ediyor oluşu bir eksi durum oluşturmaktadır.

Azure ve AWS arasında tercih yapılması gerektiğinde; dikkat edilmesi gereken konu istenilen hizmetin ölçeğidir. Azure daha çok PaaS modelinde hizmetler sunarken, AWS IaaS modeli üzerine yoğunlaşmaktadır. Tez konum olan projenin ölçeğinin IaaS hizmetlerini kullanmayacak oluşu ve Azure platformunun. NET teknolojilerini uygulamaya imkân vermesinden sebep; tez konusu olan çalışmayı gerçeklerken kullanacağım Bulut Bilişim platformunun Azure olmasına karar verilmiştir.

#### **2.4.10. Windows Azure üzerinde uygulama geliştirme**

Windows Azure hizmet olarak Microsoft yazılım firmasının altyapısıdır ve bu sebeple geliştirme yapabilmek için MS Visual Studio yazılımına ihtiyaç duymaktadır. Visual Studio üzerine kurulacak olan Windows Azure SDK'sı ile yazılım geliştirme yapılabilir. Windows Azure SDK'nın kurularak uygulama geliştirmeye başlanabilmesi aşağıdaki minimum gereksinimlerin sağlanması gerekmektedir.

- .NET Framework 3.5 SP1
- Windows Vista SP1
- Microsoft SQL Server Express 2005/2008/2010
- IIS 7.0 (ASP.NET ve WCF HTTP Aktivasyon)
- Windows PowerShell (Opsiyonel)

Gereksinimler sağlandıktan sonra hesap alma işlemine geçilebilir. 90 günlük deneme sürümü veya kullandığın kadar öde seçeneklerinden biri seçilerek hesap alma işleminin tamamlanmasının ardından Windows Azure Service Platformuna erişim sağlanmış olmaktadır.

Visual Studio geliştirme ortamına Azure eklentileri kurulduktan sonra proje şablonlarından Windows Azure Cloud Service seçilerek Web Role veya Worker Role tipinde bulut uygulamaları geliştirilebilmektedir. Geliştirilen uygulamanın tanımlanmış hesaba ait bulut üzerindeki ulana yüklenmesi de Visual Studio yazılımı üzerinden yapılabilmektedir.

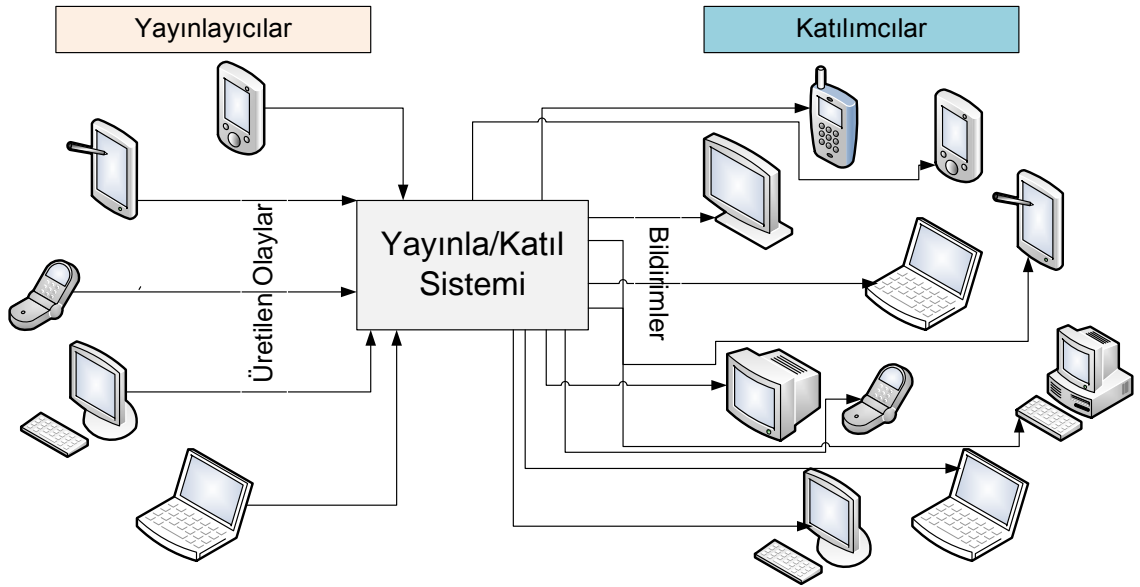
### 3. MALZEME VE YÖNTEM

#### 3.1. KAA'LAR İÇİN KULLANILABİLİR BİR YAYINLA/KATIL MİMARİSİ

##### 3.1.1. Yayınla/Katıl İletişim Modeli

Yayınla / katıl sistemi, yayıncılardan, katılımcılardan ve üretilen olayları aktaran altyapıdan oluşan bir dağıtık hesaplama paradigmasıdır. Şekil 3.1’de görüldüğü gibi yayıncıların ürettiği olaylar yayınla katıl sistemi tarafından katılımcılara bildirilmektedir. Bu yapının en temel kullanımı, Anadolu Ajansı’nın ürettiği haberlerin yazılı ve sözlü yayın organlarına iletilmesi şeklinde örneklendirilebilir.

Bu mimari içerisindeki altyapı, üretilen olayları ilgili katılımcılara aktararak bildirimlerin sevkiyatından sorumludur. Katılımcılar ise bu bildirimleri tüketerek, yaşam döngüsünü tamamlarlar.



Şekil 3.1 : Yayınla / Katıl Sisteminin Genel Yapısı

Katılımcıların üretilen olaylara olan ilgisi başlık tabanlı (topic-based) ve içerik tabanlı (content-based) olmak üzere iki kategori altında değerlendirilir.

Başlık tabanlı yayınla/katıl sistemlerinde katılımcılar belirli başlık(lar)a ilgi duyduklarını beyan ederler. Örneğin spor haberleri yapan bir derginin Galatasaray kulübünden sorumlu yazarının “Galatasaray”, “Fatih Terim”, “Ünal Aysal” gibi



başlıklara sahip haberlere abone olmasıdır. Bu yapı aynı zamanda bilgilendirme (feed), grup ve kanal yapısı olarak da bilindiğinden, katılımcıların yayıncılara bir başlık üzerinden bağlanması olarak ifade edilir.

İçerik tabanlı yayıncı/katıl sistemlerinde katılımcılar bütün bilgilere erişebilmektedir. Fakat bu yapının etkin kullanılabilmesi bildirimlerin filtrelenmesi ile sağlanmaktadır. Katılımcıların ilgisi matematiksel modelde bir filtre olarak uygulanır ve bildirim sevkiyatı gerçekleştirilir.

İki yaklaşımı birbirleri ile kıyaslanması durumunda; başlık tabanlı yayıncı/katıl sistemler, içerik tabanlı yayıncı/katıl sistemlere göre daha sade ve durağan yapıdadır. Bu sadelik uygulamaların daha süratli, etkin ve kolay kullanılmasını sağlamaktadır. İçerik tabanlı yayıncı/katıl sistemlerinin çalışması için çok yönlü iletişim protokollerinin tanımlanması gerekmektedir. Bildirimlerin ilgi duyan katılımcılara ulaştırılması için tanımlanacak filtrelerin, sistem içerisindeki tüm nesnelere tarafından uygulanabilir olmalıdır. Bu çalışma yöntemi başlık tabanlı yayıncı/katıl sistemlere göre daha az süratli fakat daha etkin bir yapıdadır.

### **3.1.2. Mevcut Yayıncı/Katıl Mimarileri**

Bu bölümde akademik ve sektörel amaçlar için geliştirilmiş olan yayıncı / katıl mimarileri anlatılmaktadır. Her sisteme ait temel özellikler ve önemli karakteristikler vurgulanacaktır.

TIB/Rendezvous [59] TIBCO tarafından geliştirilmiş olan ticari bir yayıncı/katıl altyapısıdır. TIB/Rendezvous konu tabanlı bir yayıncı mekanizmasına sahiptir. Olaylar özel olarak biçimlendirilmiş veri kümeleri üzerinden iletilmektedir. Sistem içerisinde üç katmalı bir mimari bulunmaktadır. Ağ içerisindeki her bir düğüm olayları filtrelemek ile yükümlü istemci yazılımını çalıştırmaktadır. Bu istemci yazılımlar aynı alt ağ içerisinde (subnet) tüm ortama yayın yaparak (broadcast) iletişim kurarlar. Bu üretilen olaylar daha üst katmanlara farklı yazılımlar ile yönlendirilirler.

SIENA [60] Milano ve Colorado üniversiteleri ile ortaklaşa olarak geliştirilen bir olay bildirim hizmetidir. Başlıca amacı, ölçeklenebilirlikten ödün vermeden abonelik mekanizmasının daha anlamlı olmasını sağlamaktır. SIENA içerisindeki olay bildirimleri; tip, isim ve değer şeklinde biçimlendirilmiş özellik kümeleri ile ifade edilmektedir. Abonelik dili içerik tabanlıdır ve duyuruları konusuna göre filtreleyen bir tanım içermektedir. Üretilen olayların sevkiyatı dağıtık mimari üzerinde hiyerarşik veya abonelik iletimi yöntemiyle gerçekleştirilmektedir.

XSiena [61,62], en güncel içerik tabanlı yayınla/katıl hizmetlerinin dâhil edildiği SIENA evrimidir. Geleneksel modelden farklı olarak, tıkanıklık kontrolü, takı aktarımı, Bloom filtresi tabanlı yönlendirme ve hataya dayanıklı işlem kabiliyetine sahiptir.

Le Subscribe [63] INRIA tarafından geliştirilmiş olan içerik tabanlı bir yayınla / katıl sistemidir. Projenin amacı çok sayıda dağıtık bileşene hizmet verebilen duyarlı uygulamalar sunmaktır. Projenin odak noktası çok sayıda üretilen olayların yüksek doğrulukta filtrelenmesi sağlamaktır.

JEDI [64] (Java Event-based Distributed Infrastructure) Milano Üniversitesi tarafından geliştirilmiş olan olay tabanlı altyapıdır. Nesne tabanlı olayların iletiminde demet alanı bildirim sistemi kullanmaktadır. Olaylar; birden fazla karakter dizisinin birleşimden oluşmaktadır. Üretilen olayların sevkiyatı merkezi ve dağıtık yapı üzerinden hiyerarşik iletilme stratejisi ile sağlanır.

Hermes [65, 66] bileşenler arası (p2p) yönlendirme trafiğini azaltmak için tasarlanmış, ayarlanabilir ve ölçeklenebilir yayınla/katıl ara katmanıdır. İçerik tabanlı iletişim kullanmaktadır. İletilmesi istenen olaylar, aktarımı sağlayacak olan ana düğümlerde buluşturularak, ağaç yapısında gönderim sağlanır.

Gryphon [67], IBM tarafından geliştirilmekte olan araştırma projesinin bir ürünüdür. Paralel çalışan arama ağaçları ile dağıtık bir filtreleme servisi ile mesajların yönlendirilmesi sağlanmaktadır. Sistem dayanıklılığının replikasyon ile desteklendiği bu modelde, olay sevkiyatı ağaç topolojisi üzerinden dağıtık olarak gerçekleştirilmektedir.

XNet [68] çok fazla içeriği XML biçiminde kullanırmayı sağlayan, ölçeklenebilir bir içerik-tabanlı ağ altyapısıdır. Eurocom tarafından geliştirilen bu sistem, yönlendirme tablolarının kapasitesini, veri iletiminde oluşan trafiğe göre düzenleyen bir iletim sistemine sahiptir.

Rebeca [69] farklı yönlendirme algoritmalar kullanan bir bildirim servsidir. Ağ yapısı kullandığı iletim modelinde kiralama mantığı ile abonelikleri yönetmektedir. Abonelikler belirli periyotlarda düşürülerek kullanılmayan bağlantılar elimine edilmektedir. Kullanımı devam edenlerde ise kira uzatılarak servis sürdürülür. Bu yaklaşımın amacı tüketici etmenlerin sadece ilgi duydukları etmenlere ulaşmasını sağlayarak, iletim trafiğinin azaltılmasıdır.

Elvin [70] Distributed Systems Technology Centre'de geliştirilmiş olan yayınla/katıl bildirim servsidir. İçerik tabanlı abone dili, iletilerin filtrelenmesinde çeşitli veri tipleri kullanmaktadır. Çok sayıda aracı etmenin iletişim halinde olması kullanılarak geniş alanlarda hizmet verecek şekilde tasarlanmıştır. Sistemin yoğunlaştığı özellik; ölçeklenebilirliğidir.

ToPSS [71] içerik tabanlı eşleştirme yapan, merkezi aracı etmenlerin başrolde kullanıldığı bir mimaridir. Günümüze kadar farklı yaklaşımlara tabi tutulmuştur. Örneğin A-ToPSS sistemi [72,73] klasik Boole komutlarından tahmini eşleştirmeler yaparak çalışmaktadır. Bir başka yaklaşım; aboneler ve yayıncılar arasındaki semantik ilişkileri ele alarak eşleştirme yapan S-ToPSS sistemidir [74]. L-ToPSS sistemimde [75] ise olayların, filtreleri ve bileşenleri yerlerini baz alan eşleştirme algoritmaları çalışmaktadır.

PADRES [76], ToPSS temel alınarak geliştirilen ve dağıtık aracı mimariyi destekleyecek şekilde uyarlanan bir sistemdir. Geçmişe dönük veri erişimi, kaynak keşfi, yüksek hizmet kapasitesi ve ölçeklenebilir servisleri ile ön plana çıkmaktadır.

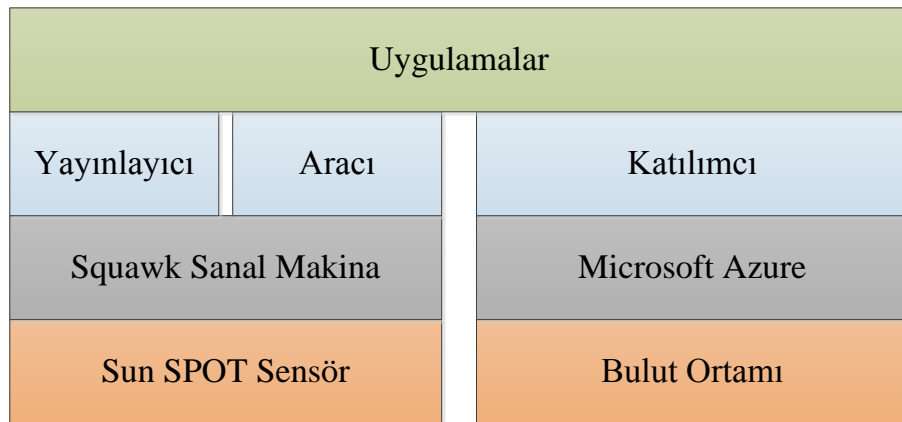
### **3.1.3. Geliştirilen Yayınla/Katıl Modeli**

Geliştirilen sistem geleneksel bir yayınla/katıl mimarisindeki temel öğeler olan yayıncıları, katılımcıları ve bildirim altyapısını barındırmaktadır. Yayıncılar Sun

SPOT kit içerisindeki ölçüm yapan düğümlerden oluşmaktadır. Yayıncıların ürettiği ölçüm değerleri bildirim olarak katılımcılara aktarılmaktadır. Bu aktarım işlemi sürecinde, aracı olarak nitelendirilen ölçüm yapmayan düğümler kullanılmaktadır. Ölçüm yapan yayıncı düğümlerin tek iletişim sebebi, ürettikleri olayların bulunduğu bölgedeki aracıya iletilmesidir. Toplanan tüm ölçüm bilgilerini iletimi, aracı düğümler vasıtasıyla gerçekleştirilir. Aracı düğümlerin tek görevi diğer aracı düğümler ile iletişim omurgasını oluşturmaktır. Yönlendirme işlemi sonucunda toplanan tüm olaylar veritabanında saklanmak üzere kaydedilirler. Katılımcılar, çoğu zaman ilgi duydukları bilgilere olayların üretimlerine eş zamanlı olmayan şekilde (kapalı devre) ulaşırlar. Bilgilere ulaşma yöntemi bulut ortamındaki web uygulamasının kullanılması ile mümkün olabilmektedir. Fakat bildirim sistemi, üretilen olayların içeriğine bağlı önceden tanımlanmış filtrelere takılan durumları anlık olarak da iletmektedir. Bu iletiler çoğunlukla acil durumları ifade etmektedir.

### 3.1.3.1. Altyapı

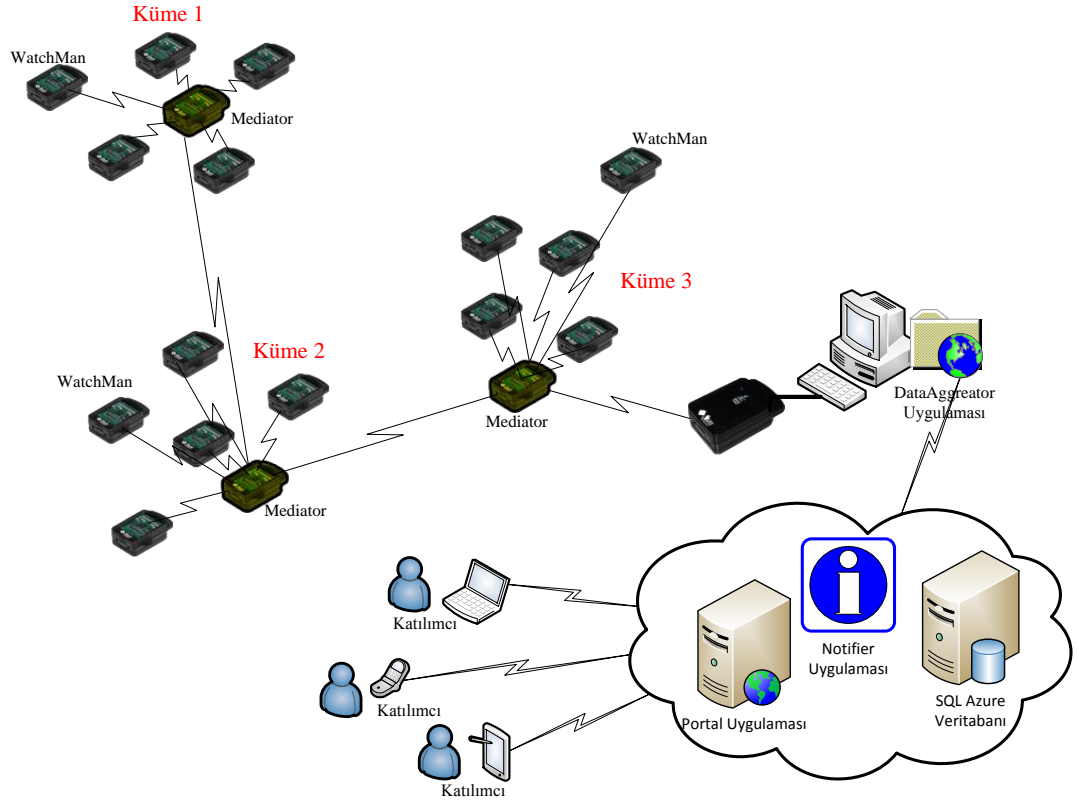
Geliştirilen model, içerik tabanlı basit yönlendirme yapan bir yayıncı/katıl sistemidir. Ölçüm yapan sensörler üzerinde çalışan uygulama tercih edilen davranış biçimine göre yayıncı veya aracı olarak hareket etmektedir. Yayıncıların ürettiği ölçüm değerleri, ilgili aracı düğümler üzerinden geçit uygulamasının çalıştığı merkez düğüme iletilir. Merkez düğüm toplu haldeki ölçümleri bulut ortamında saklanmak üzere aktarır. Katılımcılar bu bilgileri bulut üzerinde çalışan servisler vasıtasıyla tüketirler. Geliştirilen Yayıncı/Katıl modelinin iskeleti aşağıdaki Şekil 3.2'de gösterilmektedir.



Şekil 3.2 : Geliştirilen Yayıncı/Katıl Modelinin İskeleti

### 3.1.3.2. Bileşenler

Geliştirilen model içerisinde farklı görevlerde çalışan 5 adet uygulama bulunmaktadır. Bu uygulamalar *WatchMan*, *Meadiator*, *DataAggregator*, *Notifier* ve *Portal* olarak isimlendirilmiş ve çalıştıkları platformlara göre farklı yazılım geliştirme ortamlarında gerçekleştirilmişlerdir. Tablo 3.1'de gösterildiği üzere; *WatchMan* uygulaması sensör düğümlerin ürettiği ölçüm bilgilerini toplamak için çalışmaktadır. Sun SPOT sensör cihazı üzerinde çalışacak şekilde geliştirilmiş diğer uygulama olan *Mediator*; üretilen ölçümleri bölgesel birikimleri, sahip olduğu yönlendirme tablosunu kullanarak merkez düğüme sevkıyatı gerçekleştirir. Merkez düğümün USB üzerinden bağlı olduğu bilgisayarda *DataAggregator* uygulaması ölçüm verilerini bulut ortamındaki depolama alanına kaydeder. Bu iletişim sonucunda ölçüm yapan sensör düğümden çıkan bulut ortamındaki veritabanına ulaşmış olur. En son yazılım olan *Notifier*'in görevi bulut ortamına aktarılan verilerin tanımlanmış filtrelere uyması durumunda anlık bilgilendirme epostası göndermektir. Filtreye takılmayan sıradan bilgilerin erişimi için katılımcılar *Portal* uygulamasını kullanmaktadırlar. Bütün bileşenlerin bir-birleri ile olan etkileşimi Şekil 3.3'de sunulmaktadır. Amaç kapalı devre bir ağ içinde olan bilgilerin bulut ortamı üzerine aktarılmasına ve sonrasında ilgili etmenlere duyurulmasına imkân veren modeli detaylandırmaktır.



Şekil 3.3 : Geliştirilen Yayınla/Katıl Modelinin Fiziksel Yapısı

Sistem içerisindeki beş bileşenine ait geliştirme ve çalışma ortamının bilgileri aşağıdaki Tablo 3.1'de ifade edilmiştir. Her yazılım üzerinde çalışmakta olduğu işletim sistemine ve donanıma uygun geliştirme ortamlarında gerçekleştirilmiştir.

Tablo 3.1 : Yayınla/Katıl Modelinin Bileşenleri

Uygulama İsmi	Geliştirme Ortamı	Çalışma Platformu
<i>WatchMan</i>	Java	Squawk
<i>Meadiator</i>	Java	Squawk
<i>DataAggregator</i>	C#.NET	Microsoft Windows
<i>Notifier</i>	C#.NET	Microsoft Azure
<i>Portal</i>	C#.NET	Microsoft Azure

### 3.1.3.3. İletişim modeli

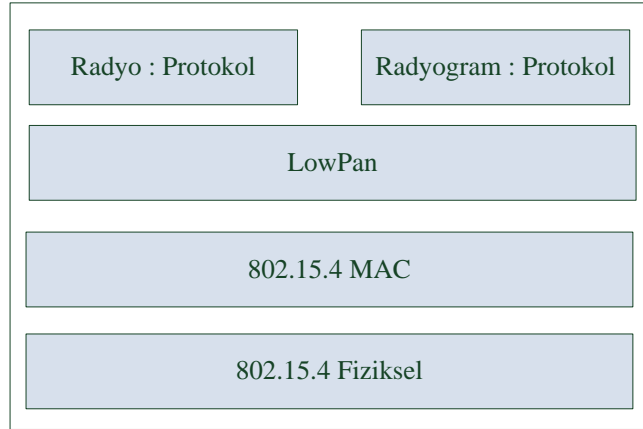
Sun SPOT sensör cihazlar üzerinde çalışmakta olan J2ME, genel özellikleri bulunan bir bağlantı altyapısı (Generic Connection Framework (GCF)) kullanarak iletişim oluşturur.

GCF ile cihazlar arası radyo iletişimi ve çoklu atlama yönlendirmesi olmak üzere iki farklı iletişim yöntemi kullanmak mümkündür.

*Radiostream* protokolü ile iki cihaz arasında güvenilir, tamponlanmış ve akış tabanlı (stream based) iletişim kurulmaktadır.

*Radyogram* protokolü ise veri iletimi tabanlı (datagram based) iletişimi kullanır. Bu iletişimin karşı tarafa ulaşmasını garantisi bulunmamaktadır. Birden fazla atlama ile gönderilen veri iletilerinin ağ içerisinde kaybolması olası bir durumdur. Tezat şekilde, aynı veri iletilerinin birden fazla iletim durumu da yaşanabilmektedir.

Bu iki protokol, Şekil 3.4'de gösterildiği gibi 802.15.4 uygulamasının MAC katmanında gerçekleştirilmektedirler.



Şekil 3.4 : Sun SPOT cihazı radyo iletişim yığı (stack)

Ölçüm bölgesindeki düğümlerin iletişimi esnasında cihazların isimlendirmesi için 64 bitlik IEEE adresleri kullanılmaktadır. Bu adresler dört adet onaltılık (hexadecimal) (*nnnn.nnnn.nnnn.nnnn*) bloktan oluşur ve ilk sekiz hane her zaman "0014.4F01" ile başlar. *Radiostream* protokolü, soket iletişimi benzeri eşler arası (peer-peer (p2p)) bir iletişimi öngörmektedir. İletişim başlangıcı için hedefin 64 bitlik IEEE adresi ve belirlenmiş bir bağlantı noktası (port) tanımlanmalıdır. Bağlantı noktası değeri 0 ile 255 değerleri arasında tercih edilir. Bağlantı açılabilmesi için yapılması gereken tanım şu şekilde olmalıdır;

*RadioStreamConnection Baglanti =*

*(RadioStreamConnection)Connector.open("radiostream:<HedefAdresi>:<BaglantiNoktasi>");*

Bağlantının sağlanabilmesi için her iki düğümünde aynı bağlantı noktasını açması birbirlerinin adreslerini bilmesi gereklidir. Bağlantı bir kere açıldı mı; her iki düğüm de gönderim ve alım işlemlerini uygulayabilirler.

*DataInputStream GelenIleti = Baglanti.openDataInputStream();*

*DataOutputStream GidenIleti = Baglanti.openDataOutputStream();*

Gönderilen iletinin havada kaybolmaması için tampon (buffer) doluluk miktarı takip edilmeli, ihtiyaç duyulan durumlarda da *flush()* methodu ile yer açılmalıdır. Kullanılan iletişim hedefe ulaşmaması durumunda *NoRouteException* hatası fırlatılır. İletişimin bir sonraki durumu için her zaman MAC katmanı için olan bilgilendirme (*acknowledgement*) beklenir.

*RadioStream* protokolünün kullanılması veri iletimini garanti etmektedir fakat bazı durumlarda birden fazla nesne ile iletişime girilmesi gerekebilir. Bu noktada, *Radiogram* protokolünün tüme gönderim (broadcast) özelliği kullanılmaktadır. Tüme gönderim bağlantısı kurmak için bir bağlantı noktası tespit edilmelidir.

*DatagramConnection Baglanti =*

*(DatagramConnection)Connector.open("radiogram://broadcast:< BaglantiNoktasi >");*

Bağlantı noktasını dinleyen tüm düğümler gönderilen veri iletimlerini alabilirler. Bu sayede iletişim menzili içerisindeki tüm düğümlere duyuru niteliği taşıyan bilgi iletilebilir.

Geliştirilen Uygulamaların detayına bulgular kısmından ulaşılabilir



### 3.1.4. Kullanılan Fiziksel KAA Platformu: Sun SPOT

KAA platformu tercih edilirken ilk aranan özellik programlanabilme yetisi olmuştur. Tercih edilen platform Sun SPOT [77], Java ile güçlü bir geliştirme imkanı sunmaktadır. Sun firması tarafından geliştirilen bu sistem küçük programlanabilir obje teknolojisi (Small Programmable Object Technology) olarak lanse edilmektedir. Temel kit içerisinde bulunan 2 adet sensör düğüm cihazı ve 1 adet istasyon cihaz Şekil 3.5'de gösterilmektedir.



Şekil 3.5 : Deney için Tercih Edilen KAA Platformu: Sun SPOT

Sensör düğüm fiziksel olarak 3 temel katmandan oluşmaktadır. Alt kısmından yukarıya doğru pil, işlemci kartı ve sensör kartı bulunmaktadır. Sensör kartın üzerindeki koruyucu plaka ile beraber 41 x 23 x 70 mm ebatlarında ve 56gr ağırlığındadır. 180 Mhz 32-bit ARM920T çekirdek işlemci 512K RAM/4M Flash bellek ile düşük ölçekli hesaplama işlemlerini gerçeklemek için yeterlidir. Sun SPOT, 3.7V'da şarj edilebilen 750mAh prizmatik lityum iyon bataryası ile uzun süreli kullanımlara imkân vermektedir.

Kullanım kapasitesi ölçüm sıklığına bağlı olarak değişmekle birlikte; derin uykuda 32uA tüketimi bulunmaktadır.

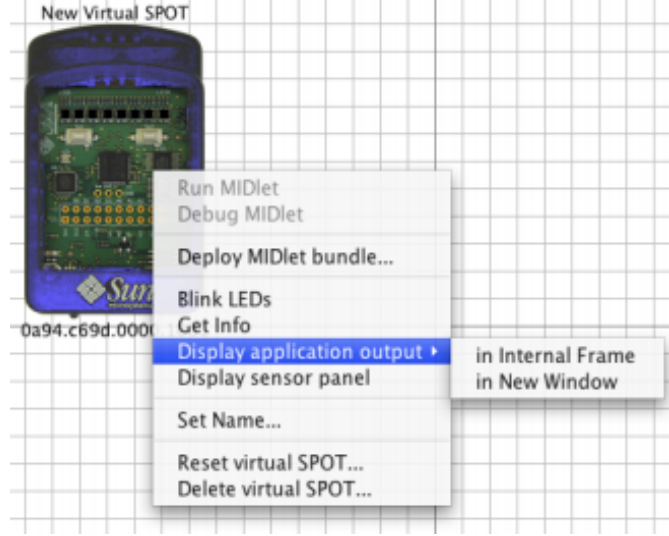
Cihazda üzerinde sıcaklık, ışık ve 3 eksen ivmeölçer (accelerometer) 3 adet sensör bulunmaktadır. Ayrıca harici sensörlerin bağlanmasına imkân veren pinler, farklı ortam karakteristiklerinin incelenmesine olanak sağlamaktadır. Düğümler arası iletişim IEEE 802.15.4 üzerinden yapılmaktadır. Baz istasyon ise USB üzerinden bilgisayara bağlanarak daha üst katmanlara iletişimi taşıyabilmektedir.

Cihazlar üzerinde doğrudan çalışan bir işletim sistemi bulunmamaktadır. Sanal makine yaklaşımı olan Squawk [78] J2ME CLDC 1.1 Java VM ile uyumlu olarak cihaz yönetimini sağlamaktadır. Tüm cihaz sürücüleri JAVA ile geliştirilmiştir.

2008 yılından sonra tüm platform açık kaynak kodlu hale gelmiştir. Java'nın sunduğu yüksek programlama kabiliyeti ve mevcut kütüphanelere doğrudan erişme imkânı ile Sun SPOT pek çok araştırma için KAA platformu olmaktadır.

#### **3.1.4.1. Sun Spot Uygulama Geliştirme Ortamı**

Sun SPOT cihazlar üzerinde çalışmak üzere geliştirilen yazılımlar JAVA programlama dili tabanlıdır. Geliştirme ortamı olarak NetBeans 6.1 kullanılmıştır. Sun SPOT Kit ile birlikte gelen Solarium isimindeki emülator yazılım, Sun SPOT cihazların masaüstünde çalışmasına imkan sağlamaktadır. Solarium içerisinde sanal düğümler oluşturarak (Şekil 3.6), gerçek ortamdaki gibi ortam karakteristiklerinin analizi yapılabilmektedir.



Şekil 3.6 : Solarium emülatöründeki sanal düğümlerin görüntüsü

Deney ortamındaki sanal düğümlerin sensörlerine erişim imkanı bulunmaktadır. Işık, sıcaklık ve ivme ölçer değerlerine istenilen girdiler atanabilir ve paylaşımlı merkez istasyon tanımlanması ile emülatörde çalışan düğümler ile gerçek düğümler arasında iletişim kurulabilmektedir.

Geliştirme sürecinde Squawk üzerinde çalışacak olan yazılımlar Watchmen ve Mediator'un MIDlet'leri sanal düğümler üzerine konulmuştur. Her iki yazılımının genel fonksiyonlarını başarılı bir şekilde sergiledikleri ve düğümler arasındaki otonom iletişimin sağlanmış olduğu beta sürümlerine kadar gerçek cihazlara kurulum yapılmamıştır.

Sonların emülatöründeki sanal düğümlerin, gerçek düğümler için hazırlanmış MIDlet'leri çalıştırabilme özelliği sayesinde Watchmen ve Mediator uygulamalarının geliştirme süreci daha kısalmıştır. Ayrıca gerçek ortam testlerine sanal düğümleri entegre ederek kablosuz iletişimde yaşanan sorunların tespiti daha kolay olmuştur.

## 3.2. BULUT ORTAMI VE GELİŞTİRİLEN SERVİSLER

### 3.2.1. Bulut Ortamında Veri Saklama

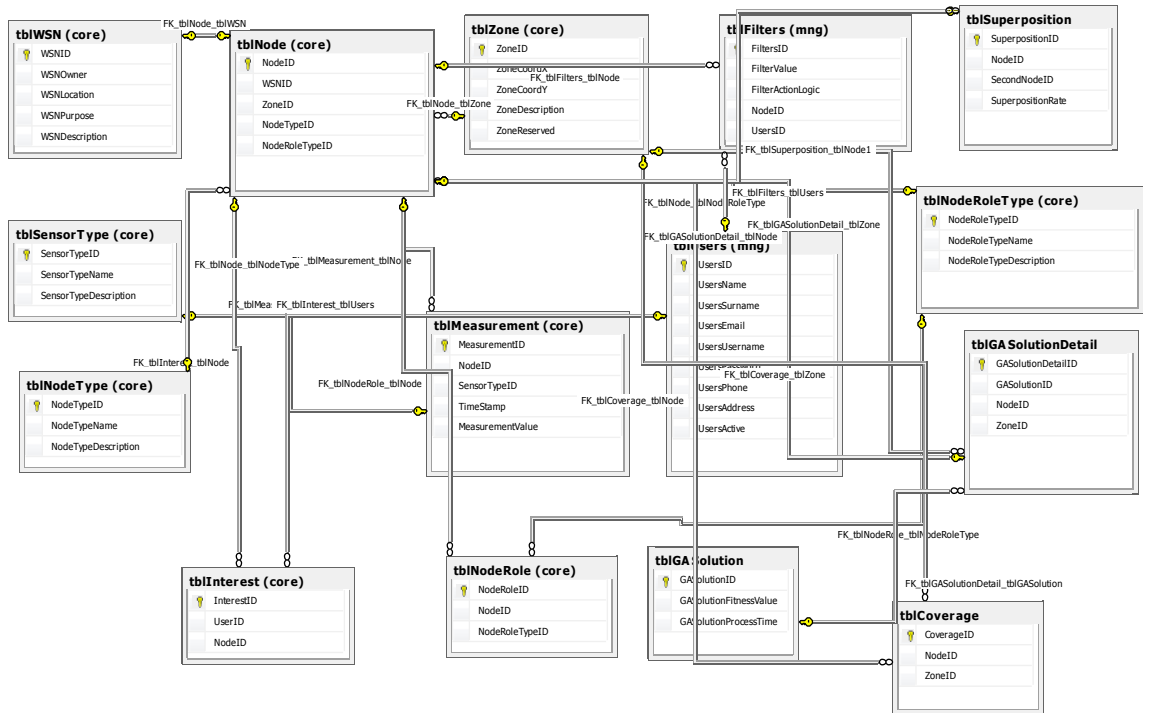
Bulut üzerindeki veri depolama hizmeti için kullanılan teknoloji SQL Azure [79] olmuştur. Windows Azure bulut verilerini yönetmenize imkân veren birden çok hizmet sunmaktadır. Microsoft'un geliştirmiş olduğu ilişkisel veri tabanı yönetim sistemi SQL Server 2012 [80] temel alınarak; bulut içine veritabanı uygulamalarını oluşturabildiğimiz ve genişletebildiğimiz altyapıdır. Kısaca Microsoft tarafından bulut içerisinde sunulan yüksek kullanılabilirliği olan, ölçeklenebilir ve çok kiracılı veritabanı hizmeti olarak adlandırabiliriz. SQL Azure aslında geleneksel veritabanı hizmetinin bulut üzerine aktarılmasından fazlasıdır [81]. Örneğin veri federasyonu (federation) teriminin bulut üzerinde kullanılabilmesi SQL Azure ile mümkün olabilmektedir. Veri paylaşırma (partitioning) hizmeti yerel veri tabanlarındaki ilgili tabloya ait kolonların birden fazla tabloya parçalanması ile sağlanmaktadır. Bu işlem yatay paylaşırma (horizontal partitioning) olarak adlandırılır ve yüksek miktardaki veri saklanması senaryolarında; sorgu sürelerini kısaltabilmektedir. Fakat daha büyük etki; dikey paylaşırma (vertical partitioning) ile sağlanabilmektedir. Çok yüksek miktarda veri depolanan tabloları belirli bir kayıt sayısından sonra bölmek gerekir. KAA'ların yaptığı ölçümlerin depolanması buna en güzel örneklerden biridir. Saat başı yapılan çok sayıda ortam karakteristiğinin saklanması, bir tablo için büyük bir yükür. Bir milyon üzerisi kayıtları paylaşırarak başka eşdeğer başka tablolara bölmek sorgu sürelerini önemli ölçüde kısaltmaktadır. Yapılan çalışmada bu yöntemler kullanılarak veri depolaması sağlanmıştır.

Verinin saklandığı veri tabanı modeli *core*, *ga* ve *mng* isimindeki 3 şemadan (schema) oluşmaktadır. Her şema farklı içeriğe göre gruplanmış olan verilerin tablolarını barındırmaktadır. Veri modelini oluştururken çoklu şema yapısının tercih edilmesinin sebebi; veritabanına erişen kullanıcıların haklarının daha güvenli bir şekilde yönetilmesini sağlamaktır. İlgili kişiler sadece kendilerine ait şemalara erişebilirken, erişilmemesi gereken tablolar ise farklı şemalar altına alınarak güvende tutulabilmektedir. Okunabilirlik (readability) ve yönetilebilirlik (manageability) özelliklerinden ötürü çoklu şema kullanımı önemlidir.

Veri modelindeki *core* şeması temel bilgileri barındıran tabloları saklamaktadır. Sisteme kayıtlı olan KAA'alar, içerisindeki düğüm çeşitleri, düğümlere ait olan bireysel bilgiler ve daha fazlası *tblWSN*, *tblNode*, *tblZone*, *tblNodeType*, *tblNodeRole*, *tblNodeRoleType*, *tblInterest*, *tblMeasurement*, *tblSensorType* tablolarında tutulmaktadır.

Diğer şema *ga* ise 3.bölümde anlatılacak olan genetik algoritma ile KAA dağıtımının iyileştirilmesinde kullanılan/üretilen verileri barındıracak tablolara ev sahipliği yapmaktadır. *tblCoverage*, *tblSuperPosition*, *tblGASolution*, *tblGASolutionDetail* ve *tblGASolutionCoordinates* bu şemaya ait tablolardır.

En son şema *mng* ise geliştirilen yayınlı/katıl mimarisindeki katılımcıların veriye ulaşabilmesi için gerekli olan kayıtları tutan tablolar olan *tblFilter* ve *tblUsers*'dan oluşmaktadır. Tüm şemalara ait tablolar ve aralarındaki ilişkiler Şekil 3.7'de gösterilmektedir.



Şekil 3.7 : SQL Azure Şema Görünümü

Veritabanındaki tüm tablolar ilgili veri modeline özgül bilgiyi tutmaktadırlar. Bilgilerin SQL Azure üzerinde doğru veri tipi ile saklanması veri bütünlüğünü koruduğu (data integrity) gibi, yapılacak olan sorgulamaların işlem süresini de azaltmaktadır.

### 3.2.2. Genetik Algoritma Kullanılarak Dağıtımın İyileştirilmesi

Kapsama alanı boşlukları (Coverage holes) kablosuz algılayıcı ağlarda (Wireless Sensor Networks) efektif olmayan dağıtım (deployment) yapılmasından, hatalı düğümlerin veya engelleyici cisimlerden bulunmasından dolayı oluşan ölçüm yapılamayan alanları ifade etmektedir. Bu alanlardaki aktivite takip ve kontrol edilemez. Ayrıca bu alanların sıkça bulunduğu ortamlarda, ölçüm yapan düğümlerin merkez düğüme olan iletişimde engellemeler veya zorlaştırmalar oluşur. Bu açıdan bakıldığında kapsama alanı boşluklarının mevcudiyeti, servis kalitesinin belirlenmesindeki en önemli metriklerden biri olmaktadır. Tez çalışmasının bu bölümden KAA'ların herhangi bir t anındaki kapsama alanı kabiliyetini ölçümleyen ve ihtiyaca göre genetik algoritma (genetic algorithm) kullanarak düğümlerin olması gereken ideal konumlarını tespit eden bir sistem tasarımı anlatılmaktadır. Konumları bilinen sensör düğümlerin oluşturduğu ölçüm verileri bir yayınla-katıl (publish-subscribe) mimarisi ile toplanarak bulut (cloud) ortamına aktarılmakta ve işlenmektedir. Sistemin bulut üzerinde sakladığı veriler kapsama alanı başarısı tespiti, ideal ölçüm yapan düğüm sayısı gibi hayati parametrelerin tespitinde kullanılmaktadır. Önerilen sistemdeki genetik algoritmanın çalışma konfigürasyonu, simülasyon testleri gözlemlenerek belirlenmiştir. İdeal parametrelerin bulut üzerindeki uygulamada çalıştırılması ile, dağıtımdan kaynaklanan kapsama alanı boşlukları ortadan kaldırılmıştır.

KAA'lar [5] [82] özel kullanım alanlarına sahip ad-hoc ağlardır. İlk olarak askeri uygulamalarda, düşman tespiti amaçlı geliştirilmiştir fakat artık günümüzde pek çok diğer alanlarda da kullanılmaktadır. Çalışma metodolojisi olarak KAA'ları iki ana başlık altında toplayabiliriz. Olay güdümlü (event-driven) uygulamalarda; gözlem altına alınan ortamın karakteristiklerindeki değişiklikler takip edilmektedir. Sınırdaki düşman hareketliliği veya ormanda oluşan sıcaklık artışı gibi önceden belirlenmiş takip parametrelerine bağlı olarak gözlem yapılmaktadır. Talep güdümlü (demand-driven) uygulamalarda ise algılayıcı düğümler istek gelmediği müddetçe beklemede kalırlar ve ortam karakteristiklerini ölçümlemezler. Ticari depo takip sistemleri bu uygulama tipinin en çok kullanıldığı örneklerden biridir. Şirket yöneticisi istediği an depo envanterini canlı olarak çıkartabilmektedir. KAA'lar hangi uygulama tipinde kullanılırsa kullanılınsınlar, çokça düğümün ortama dağıtımının yapılması gerekmektedir.

Dağıtımın başarısı kapsama alanının belirlenmesindeki en önemli parametredir. Zaman içerisinde kullanıma ve ortam şartlarına bağlı olarak bazı düğümlerin ölçüm yapamaz hale gelmesi, kapsama alanını küçültmektedir.

Çevre takibi gibi KAA'ların yoğunlukla kullanıldığı senaryolarda verinin kesin ve doğru oluşu çok büyük önem arz etmektedir. En doğru bilgiyi tespit edebilmek için ortam karakteristiklerine uygun donanımı tercih etmek ve uygun topolojiyi belirlemek gerekmektedir. Tüm parametrelerin en ideal şekilde tercih edildiği ortamlarda bile düğüm dağıtımları, ölçüm sonuçlarını direkt etkileyen bir işlemdir. En yüksek kapsama alanı elde edebilmek için gözlemlenecek olan ortamın önceden araştırılarak, düğümlerin yerleştirileceği koordinatlar tespit edilmelidir. Bu tespit işlemi için çoğunlukla eşkenar dikdörtgenler veya altıgen yapıda şablonlar tercih kullanılmaktadır [83]. Yerleşim koordinatlarının tespitinden sonra, tüm düğümler elle ilgili noktalara yerleştirilirler. Manüel olarak yapılan bu dağıtımın kurulum maliyeti yüksektir. Ayrıca zorlu doğa şartlarının bulunduğu ve düşman bölgesi gibi senaryolarda ise tercih edilmemektedir.

Öte yandan, çok sayıda düğümün büyük bir alana dağıtımında öncelikli olarak tercih edilecek yöntem uçaktan fırlatmaktır. Uçaktan atılan düğümlerin konuşlanması pek çok çevresel sebebe bağlı olarak rastlantısal bir olaydır. Rastgele dağıtımın, kurulum maliyetlerinin düşük olması en önemli tercih sebebidir, fakat ölçüm yapacak düğümlerin düştüğü koordinatların kestirilememesi nedeniyle gözlem yapılacak alanın tamamı kapsama altına alınamayabilmektedir. Kusursuz yapılan bir dağıtımda bile, ölçüm yapan düğümlerin düştüğü bölgelerde engelleyici nesnelere veya yere sert çarpan bir düğümün ölçüm yapamaz hale gelmesi kapsama alanı boşlukları oluşabilmektedir. Bu sebeple dağıtım sonrası en yüksek ölçüm alanını elde edebilmek için sistemin kalibrasyonu gerekmektedir. Kalibrasyon işlemi; kapsama alanı anlamında efektif çalışmayan düğümlerin tespit edilerek en ideal ölçüm noktasına taşınmasını amaçlamaktadır. Bu sayede rastlantısal dağıtım yapılmış olan KAA'ların ölçüm kapasiteleri iyileştirilmiş olacaktır.

### 3.2.2.1. Kapsama Alanı Boşluklarının Tespiti

KAA'da kapsama alanı boşluklarının tespiti, aslında Sanat Galerisi Problemi'nin (Art Gallery Problem) [84, 85] bu araştırma sahasına uyarlanmasıdır. Problemin amacı; bir sanat galerisindeki her noktayı görebilen izleyicilerin sayısının ve duruş noktalarının belirlenmesidir. Bu yaklaşıma benzer şekilde; problemin çözümü için günümüzde pek çok farklı teknikler kullanılmaktadır. Günümüze kadar yapılmış olan çalışmaları 3 ana başlık altına toplayabiliriz. Bunlar; geometrik, istatistiksel ve bağlantı tabanlı yaklaşımlardır.

Wang ve arkadaşları yaptıkları çalışmada [86], şebeke tabanlı (grid-based) bir delik tespit yaklaşımı geliştirmişlerdir. Ölçüm yapılan bölge mantıksal bölümlere ayrılarak sanal bir şebeke oluşturulur. Her mantıksal bölüm içinden bir adet baş düğüm (head node) tespit edilir. Bu düğümün görevi bölgesindeki diğer düğümlerin iletişim bilgilerini toplamak ve komşu baş düğümler ile bilgi paylaşımı yapmaktır. Toplanan bilgiler doğrultusunda iki boyutlu bir düzlem üzerinde düğümler konumlandırılarak hesaplamalar yapılmaktadır.

Bir diğer çalışmada [87] düzlemsel olmayan (non-planar) sensör ağlardaki kapsama alanı boşluklarının tespiti amaçlanmıştır. Yazarlar ortamdaki düğümlere ait iletişim grafiği (communication graph) üzerinden cebirsel topolojileri (algebraic topology) kullanan 2 farklı yöntem önermektedirler. Yapılan simülasyon testlerinde %99'luk tespit başarısı ile deliklerin koordinatları belirlenmiştir.

Alzoubi ve arkadaşları [88, 89] NP-hard olarak bilinen "Minimal Connected Dominating Set" (CDS) problemini kullanarak ilk dağıtık yerleşimi sağlamışlardır. CDS iletişim omurgasını sağlamakla beraber, kapsamanın kuvvetli ve zayıf olduğu noktaların tespitinde de kullanılabilir. Ayrıca gereksinim duyulduğu anlarda düğümlerin buldukları yerlerin değiştirilerek bu omurganın bakımı sağlanabilmektedir.

Tian ve Georganas çalışmasında [90] tüm ölçüm yapan düğümler kendi komşularının bilgisini takip etmekle yükümlüdürler. Her bir düğüm belirli uyanma periyotlarında ölçüm yaptıktan sonra komşularına durumu hakkında bilgi mesajı gönderir. Bu bilgi



mesajlarının ynetime aktarılması ile sistemdeki lm yapan dğmlerin kontrol saėlanmaktadır.

Problemin zmnde alternatif olarak; daėıtık ve yerel Voronoi tabanlı algoritma kullanılmıřtır [91]. Bu yaklařımda yakın komřuluk grafiėi (relative neighbor graph (RNG)) geliřtirilerek aė yařam sresinin uzatılması ve kapsama alanı takibi saėlanmıřtır. Voronio tabanlı algoritma belirli aralıklarda dğmlerin uyku durumuna ve uyanma durumuna gemesini kontrol etmektedir.

Kapsama kmelerini (cover set) kullanan yaklařımda [92] uygun sensrn belirlenmesinde kalan pil kapasitesi, lm alan kabiliyeti gibi parametreleri kullanan bir maliyet fonksiyonu (cost function) tercih edilmiřtir. Sistem bařarısı yapılmıř olan simlasyonlarda grlmektedir ki; literatrde yazılmıř olan benzer sezgisel algoritmalar (heuristic algorithms) ile yakın bařarı elde etmektedir.

Kapsama alanı ile ilgi karřılařılabilecek tek sorun deliklerin mevcudiyeti deėildir. Aynı blge ierisinde lm yapan ok sayıda dğmn bulunması da istenemeyen bir durumdur. Bu tarz bir senaryoda birbirine yakın olan dğmlerin yakın zamanlarda yaptıkları lmler ok anlamlı olmadığı gibi g kaybı demektir. Bu sorunun zm ile ilgili geliřtirilmiř protokol PEAS'a [93] gre bir dğm kendisine yakın aktif bir dğm olduėu mddete uyku durumunda bekletilmelidir. Uyku durumundaki pil tketimi ok dřk olduėundan lm yapan dğmler daha efektif kullanılmıř olurlar. PEAS tm alan iin kapsama garantisi vermediėi ve olasılı (probabilistic) olmasından dolayı zme olan katkısı sınırlıdır.

Bir diėer alıřmada [94] trdeř nitelikteki dğmlerin lm yapabilme srelerini uzatma amalanmıřtır. alıřmada kapsama alanı (area coverage), talep daėıtımı (request spreading) ve bilgi toplama (data aggregation) zelliklerine ynelik geliřtirme yapılmıřtır. lm yapan dğmlerden birbirlerine baėlı etki kmesi (connected dominating set) tespit edilerek kapsama, iletiřim mesafesi ve komřuluk parametreleri deėerlendirilmiřtir. Yapılan alıřma trdeř olmayan KAA'da kullanılmak zere geliřtirilmektedir.

Geliştirdiğimiz sistemin genetik algoritma kullanılarak dağıtımı iyileştirdiği bölümü başka yazarlar tarafından da ele alınmıştır. Birbirleri ile bağlantılı olan iki çalışma [95] ve [96]'de düğümlerin ölçüm yapılacak olan alana dağıtımı için GA kullanılmıştır. Bölgelere yapılacak düğüm atamalarında, ölçüm yapılacak yerlerin öncelik sırasına göre sistem çalışmaktadır.

Yine [97] veri toplama için tarama ağacı (spanning tree) yapısı önermektedir. En ideal düğümlerden oluşan tarama ağacının kullanılması, enerji tüketimini en az seviyeye düşürmektedir. Ayrıca kapsama alanı kontrolünde yine bu tarama ağacı yapısı kullanılmaktadır. Bu tarama ağaçlarının oluşturulmasında yazarlar genetik algoritma kullanmışlardır.

### 3.2.2.2. Yöntem

#### 3.2.2.2.1. Fiziksel Mimari

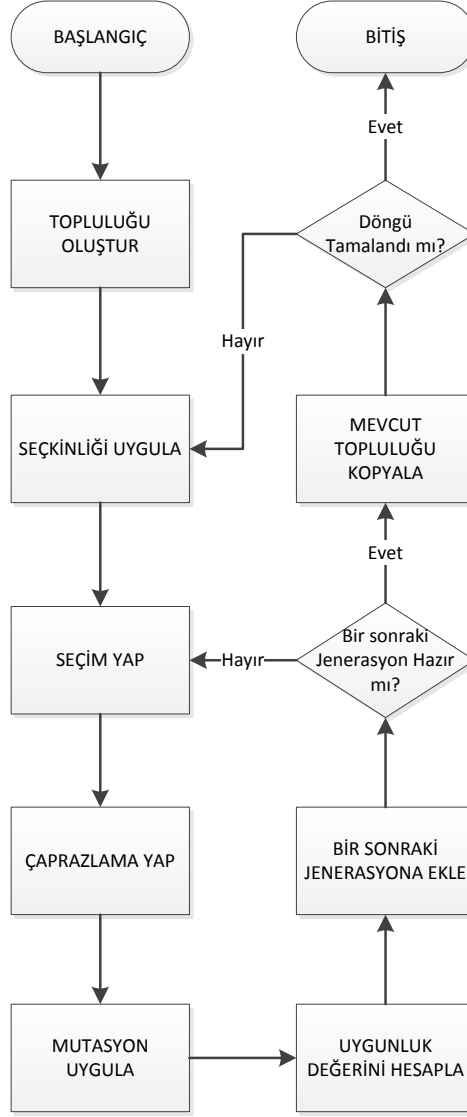
KAA'lardaki sensör düğümlerin yapıları gereği hesaplama, depolama ve uzun mesafeli iletişim yeteneklerinden yoksun olmaları, ölçüm verilerine ulaşmayı zorlaştırmaktadır. Günümüzde geleneksel KAA verilerin daha geniş alana taşınabilmesi, üzerinde çalışılmakta olan önemli bir araştırma konusudur [98]. Geliştirdiğimiz sistemde bir yayıncı-katıl (publish-subscribe) mimarisi kullanarak yerel bilgilerin ilgili enstrümanlar sayesinde bulut üzerindeki depolama alanına taşınması sağlanmıştır. Standart yayıncı/katıl iletişim modelinin temel çalışma prensibi, objelerin ilgi duydukları veriyi sunan kaynaklara kayıt yaptırmaları olduğundan; bu kayıt işlemine katılma (subscription) ve katılmayı gerçekleştirene de katılımcı (subscriber) denilir. İlgi duyulan konuya ait verileri ölçümleyip duyuran nesnelere ise yayıncı (publisher) denilmektedir. Yayıncı ve katılımcı arasındaki ilişki düzenleyen ve iletişimin gerçekleşmesine imkân veren objeye de aracı (broker) denilmektedir. Önerdiğimiz bu sistemdeki bu üç bileşen Java ve C# programlama dilleri kullanılarak geliştirilmiştir. Ölçüm yapan düğümlerin üzerinde veriyi iletmek için geliştirilen yayıncı uygulama *WatchMan*, mantıksal bölümlere ayrılmış olan gözlem alanındaki her bir yetkili düğüm üzerinde çalışan aracı (broker) uygulama *Mediator* ve son olarak tüm verilerin bulut ortamına aktarımını sağlayan *DataAggregator* bulunmaktadır. İlişkisel bir veritabanı hizmeti sunması ve web tabanlı uygulama geliştirmedeki kolaylığı sebebiyle bulut ortamı olarak Microsoft Azure [99] tercih edilmiştir.

### 3.2.2.2. Genetik Algoritmanın Uygulanması

Dağıtımı yapılmış olan bir WSN’de, ölçüm yapan düğümlerin yerlerinin düzenlenmesi bir arama problemidir. Bu çalışmada çözüm uzayının genetik algoritma ile tespit edilmesi yolu tercih edilmiştir. İlk olarak Goldberg tarafından düşünülen genetik algoritma yapısı doğal seleksiyon kavramından türetilmiş ve günümüzde pek çok arama problemlerinin çözümünde alternatif olarak kullanılmıştır. Biyolojide doğal seleksiyon [100] iklim şartları, besin durumu, tür içerisindeki rekabet gibi zorlu koşullara uyabilen canlıların yaşamlarını sürdürebilmesi olarak ifade edilmektedir. Canlıların ihtiyaç duydukları özelliklere sahip olması onları hayatta tutan yegâne nedendir. Kalıtsal üreyen canlılar bu olumlu özellikleri barındıran genotipleri diferansiyel üretim ile sonraki nesillere aktarırlar. Diğer bir deyiş ile doğal seleksiyon mekanizması bir canlı topluluğunun en ideal forma ulaşmasına imkan veren yapıdır. En ideal formu tespitinde her biri diğerlerinden farklı bir çözümü temsil eden topluluk (population) kullanılmaktadır. Topluluklar üzerinde seçim (selection), çaprazlama (cross-over) ve mutasyon (mutation) işlemleri uygulayabildiğimiz bir dizi kromozomu barındırmaktadır. KAA’ içerisinde kullanılan düğümlerin ölçüm yapılacak herhangi bir (x,y,z) koordinatına konulması bir kromozom ile temsil edilir. Her kromozomun sahip olduğu özelliklere göre bir uygunluk değeri (fitness value) bulunmaktadır. Uygunluk değeri, sunulan çözümün bizim ihtiyaçlarımıza olan uygunluğunun sayısal bir ifadesidir. *ProduceRandomChromosome()* metodu ile rastgele olarak üretilen kromozomların, *CheckForHardConstraints()* metodu ile bazı istenmeyen özelliklere sahip olup olmadığı kontrol edilir. Bazı zor kısıtlar (hard constraints) çözüm için kabul edilemeyeceğinden, bu özelliklere sahip olan kromozomlar göz ardı edilerek yeniden üretilir.

Bir sonraki süreçte uygulanacak işlem seçim (selection)’dir. Seçim; topluluktan bir sonraki jenerasyona mutasyon ve çaprazlama yapılarak aktarılacak kromozomların belirlenmesidir. Testler sürecinde iki farklı seçim algoritması kullanılmıştır; Rulet Seçimi (Roulette Wheel Selection) ve Turnuva Seçimi (Tournament Selection). Bütün jenerasyonlara ait toplam uygunluk değerleri (fitness value) hesaplanarak sıralamaya konulur. En yüksek değere sahip olanlar en çok tercih edilmesi gereken çözümler olarak kullanıcıya sunulur. Testler süresince mutasyon olasılığı %1 ve çaprazlama

olasılığı %80 olarak kullanılmıştır. Genetik algoritmanın çalışma şeması aşağıda gösterilen Şekil 3.8’de detaylandırılmıştır.



Şekil 3.8 : Genetik Algoritmanın Akış Diyagramı (Flow Diagram)

### 3.2.2.2.3. Yer değişikliği yapılacak düğümlerin belirlenmesi

Algoritma öncelikle ölçüm yapan düğümlerin komşularıyla ne oranda çakıştıklarını keşfeder. Türdeş olmayan bir KAA yapısı düşünüldüğü için, her düğümün teknik özelliklerine göre ölçüm menzili dinamik olarak değişmektedir. KAA içerisindeki tüm düğümlerin komşuları ile olan menzil çakışmaları bulut üzerindeki veritabanının *tblSuperPosition* tablosuna kaydedilir. Düğümlerin çakışma oranı coğrafi bir teknik ile hesaplanır. Bütün düğümlerin konumları bilindiği için ölçüm yapabildikleri tüm

noktalar *tblCoverage* tablosuna kaydedilir. Bu kayıtlar ölçüm yapılan dairesel alan altında kalan x,y koordinatlarının tespit edilmesidir. Koordinatların tespitinde merkez koordinatları bilinen dairenin içinde kalan noktaları bulmak için Bresenham çember algoritması (Bresenham's Circle Algorithm) [101] [102] kullanılmıştır. Algoritmanın döngüsünün her bir iterasyonunda tabloya {x,y,nodeID} bilgileri eklenerek tablo satır kayıtları oluşturulmaktadır. Bu kayıtlar gözlem bölgesinin tüm koordinatlarının hangi sensörler tarafından gözlemlendiğini analiz etmemizde kullanılmaktadır. Bir başka deyişle; bu kayıtların bulunması bize tüm kapsama alanı kapasitesini (1) ve düğümler arasındaki ölçüm alanı çakışma oranlarını (2) hesaplayabilmemize imkân vermektedir. Gerçeklemede, bu formülleri hesaplarken SQL Cursor yapısı kullanılmıştır.

$$C = \frac{\sum \text{MarkedCoordinate}(x,y)}{\sum \text{Coordinate}(x,y)} \quad (3.1)$$

$$R = \frac{n(\text{Node}_i(x,y) \cap \text{Node}_j(x,y))}{n(\text{Node}_i(x,y))} \quad (3.2)$$

Yerleri değiştirilmesi gereken düğümlerin tespiti, senaryo bazlı tercih edilen bir alfa parametresi ile belirlenir. Alfa oranı ölçüm senaryosuna bağlı olarak değişen bir yüzdeyi ifade etmektedir. Alfa oranı ve altındaki çakışmalar tüm dağıtımın yeniden yapılmaması için göz ardı edilir. Örneğin askeri uygulamalarda ölçüm alanının tamamen kaplanması öncelikli amaç olduğundan bu yüzde yüksek tercih edilmelidir. Bir sınır gözlem sisteminde alfa oranı %50 tercih edilebilirken, öte yandan bölgesel sıcaklıkların takip edildiği bir sistem için alfa oranı %20 civarında belirlenebilmektedir. Alfa oranı ve üzeri çakışması olan düğümler tespit edilerek genetik algoritmaya sokulmak üzere işaretlenir.

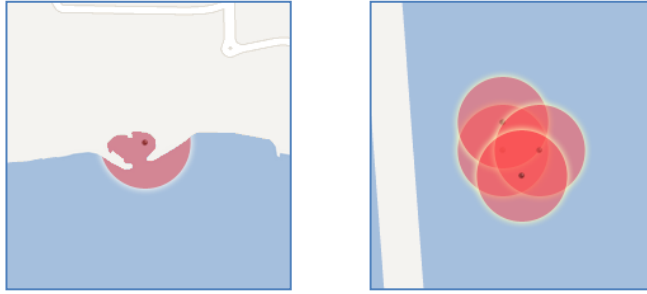
#### 3.2.2.2.4. Genetik algoritma ile mevcut dağıtımın iyileştirilmesi/güncellenmesi

Çakışma oranı yüksek düğümlerin yeni yerinin belirlenmesinde genetik algoritma kullanılır. Genetik algoritma ile efektif kullanılmayan düğümlere ölçüm sahası üzerinde uygun bir lokasyon aranır. Uygun lokasyonun tespiti için arama kriterlerinde kabul edilebilir (soft) & zor (hard) kısıtlar (constraints) uygulanmaktadır.

Zor kısıtlar kromozonların oluşturulması esnasında kesinlikle kabul edilmeyen durumları ifade etmektedir. Bu durumlar şu şekildedir;

- Düğümün ölçüm kapasitesinin yarısından fazlasının boşa gittiği veya kullanılmadığı (Şekil 3.9'da gösterilen) yerleşimler,
- Bir veya birden fazla düğümün ölçüm alanının %70'inin çakıştığı (Şekil 3.10'da gösterilen) yerleşimler ve
- Gözlem yapılacak alanın tamamının ölçüm alanı altına alınmasından sonra yapılacak ihtiyaç harici düğüm yerleştirmeleri

hard constraintler olarak değerlendirilmektedir. Bu tarz durumları barındıran dağılımlar çözüm olarak kabul edilmezler.



Şekil 3.9 : Konumundan dolayı ölçüm kapasitesi efektif olarak kullanılmayan düğümler

Şekil 3.10 : Birbirine çok yakın konumlandırılmış düğümlerin çakışması durumu

Öte yandan tercih edilmeyen fakat Kabul edilebilen istenmeyen durumları soft constraints olarak adlandırmaktayız. Bu durumların barındığı çözümler daha az tercih edilmesi gereken seçeneklerdir.

- Düğümün ilk dağıtımda bulunduğu bölgenin haricinde bir bölgeye yerleştirilmesi,
- Düğümün ölçüm kapasitesinin %25'inin boşa gittiği veya kullanılmadığı yerleşimler,
- Bir veya birden fazla düğümün ölçüm alanının en fazla %69'unun çakıştığı yerleşimler genetik algoritmanın soft constraintleridir.

Bu durumların her birinin oluşması halinde uygulanacak bir katsayı bulunmaktadır. Bu katsayılar çözümlerin tercih edilmesinde kullanılacak olan uygunluk değerlerini düşürmektedir.

Ayrıca sistem yöneticisi bazı bölgeleri önemli olarak kaydedebilir. Örneğin yangın gözlem sistemi kurulu bir ormanda, piknik için ayrılmış alanda yangın çıkma ihtimali fazla olduğundan bu tarz bölgeler daha yoğun bir şekilde takip edilmelidir. Sistemde önemli olarak işaretlenen bölgelere daha fazla ölçüm imkânı verilmekte ve ölçüm yapan düğümlerin çakışmaları bu gölgelere denk getirilmektedir. Bu sayede önemli bölgeler birden fazla düğüm ile kontrol edilmiş olurlar.

### 3.2.3. Hatalı Ölçüm Yapan Düğümlerin Kestirimi

KAA ölçümleri gibi gerçek verilerle yapılan çalışmalarda bazı gözlemlerin, dağılımdaki diğer elemanlara göre büyük, küçük veya verinin çoğundan uzak olduğu durumlar ile karşılaşılabilir. Bu gözlenen değerlerin verinin büyük çoğunluğundan farklı olması beklenir. Verinin çoğunun sahip olduğu dağılımdan farklı olduğu için bu karakteristiğe sahip bilgiye aykırı gözlem adı verilmiştir. Bir başka deyişle aykırı gözlem(ler), veri setinin ortalamasının çok uzağına düşen gözlem(ler) olarak tanımlanabilir.

Bu aykırı gözlemler verinin standart sapmasını arttırmanın yanında, dağılımı değiştirerek karar ve bilgi alma sürecinde yanılsamalara neden olarak, sistemin hatalı sonuç üretmesini sağlarlar. Aykırı gözlem tespitinde en çok kullanılan yaklaşımlar zaman serileri ve veri setinden çıkartılan özellikler yardımı ile uzaklık hesabıdır.

Verinin zaman serisi olarak elde edilebilmesi uygulamayı oldukça genişletmektedir. Ele alınan veri belirli aralıklarla (sayı, sıklık, süre, vs.) zamanda dizilebilir ise aykırılık tespitinin kestirimi daha rahat yapılabilir. Eğer zaman serileri yerine özellikler seçilir ise kümeleme algoritmalarını daha doğrudur.

Rousseeuw ve Zomeren'in [103] araştırmalarına göre bu aykırı gözlemler, toplam gözlem sayısının yarısından daha az olmak zorundadır ve aynı zamanda elde edilmek istenilen bilgiye engel olacak sonuçlar doğurarak sistemde yanılgılara ve hatalara sebep olurlar.

Standart bir Bayes ağı değişkenler arasındaki zamansal bağımlılıkları doğrudan temsil edemez. Zaman serilerine ait veriyi modellerken şimdiki ve önceki davranışlar geleceği tahmin etmek için kullanılabilir.

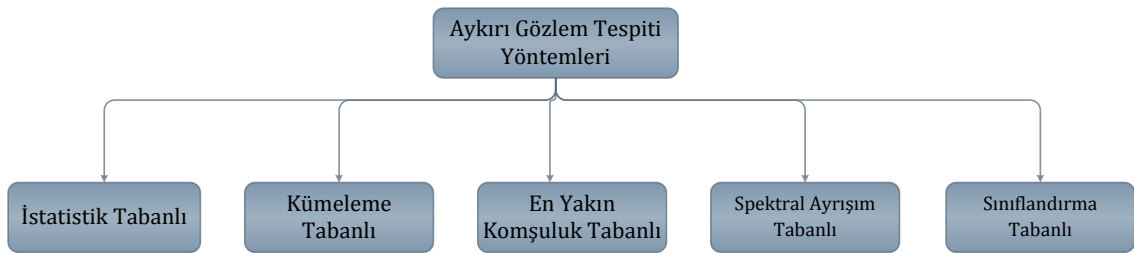
Valdes ve arkadaşları trafik esnasında saldırıları tespit eden Naive Bayes ağı kullanan anormallik tespit sistemi geliştirdiler [104]. EMERALD'ın bir parçası olan bu sistemleri potansiyel olarak alarm durumuna geçmek için yeterince şüpheli olmayan kişisel ataklarda ortaya çıkan dağıtık saldırıları tespit etme kabiliyetine sahipti.

### 3.2.3.1. Yöntem Seçimi

Aykırı gözlemlerin tespiti işleminde, verinin yapısı ve boyutu bize nasıl bir yöntem ile karar vermemiz gerektiğini belirlemektedir. Buna göre sensör verisinden beklentiler iki ana başlık altında toplanabilir. Her sensör düğüm sadece kendisinden sorumlu ve aktarması gereken özellik veya özellikleri iletmekle yükümlü olacaktır. Yâda, düğümler birbirinden bağımsız olmayarak, veri iletiminin yanında komşuluk bilgilerini de tutarak tüm ağ ile bağlantısı olacaktır.

Her kablosuz sensör ağında olası üç tip aykırılık tespiti yapılabilir, bunlar; hata, olay ve saldırı tespitleridir. Olası senaryoda hangi veya hangilerinin önemli olduğu bilgisi belirlenerek buna uygun yöntem seçilir. Bizim durumumuzda sadece sensörlerin hata tespitleri yapılacaktır. Bu da yine kendi içerisinde 3 ana başlıkta ele alınacaktır; bunlardan ilki sensör düğümünün hatalı veri hesaplaması, ikincisi sabit veri gönderimi üçüncüsü de hem sabit hem de hatalı veri gönderimidir. Bunun tespit edilme işlemi diğer düğümlerden gelen verinin ilgili düğüm verisi ile kıyaslaması sonucu elde edilecektir.

Aykırı gözlem tespit teknikleri beş kategori altında ifade edilmektedir. Şekil 3.11'da görüleceği üzere bunlar, istatistik tabanlı, en yakın komşuluk tabanlı, kümeleme tabanlı, spektral ayrışım tabanlı ve sınıflandırma tabanlı yaklaşımlardır [105].



Şekil 3.11 : Aykırı gözlem tespit teknikleri



İstatistiksel modeller olasılık dağılımına göre parametrik ve parametrik olmayan olarak ele alınırlar. Verinin istatistiksel bir dağılıma uyduğu varsayımında bulunulur. Gauss tabanlılar ve Gauss tabanlı olmayanlar parametrik, kernel ve histogram tabanlılar ise parametrik olmayan grup içine girerler. Modelde dağılıma göre bir eşik değeri belirlenir ve bu eşik değeri üstünde kalan sensörler aykırı olarak tanımlanır.

Yakın komşuluk tabanlı yaklaşımlar veri madenciliğinde ve yapay öğrenmede en sık kullanılan yaklaşımlardır. Burada iki veri örneği arasındaki mesafe kullanılır. En yakın komşuluk tabanlı yaklaşımlar veri dağılımı hakkında herhangi bir varsayımda bulunmaz ve bu sebeple uygun parametreleri belirlemek oldukça zordur [106]. Bununla birlikte çok değişkenli sistemlerde hesaplama maliyeti oldukça yüksek olmasından sensör düğümlerinde pil ömürlerinin kısılmasına sebep olmaktadır. Değişken yoğunluğu bulunan veri gruplarında doğru sonuç üretemeyen bir modeldir.

Kümeleme [107], benzer davranışlar içeren gruplardaki aykırılığı tespit etmek için oldukça popüler bir yaklaşımdır. Metodda sınıflar arası bir bağ mevcuttur, benzer özellikte olan verilerle yeni gruplar oluşturmak asıl hedefdir. Kümeleme temelli yaklaşım veri dağılımı hakkında bilgi sahibi olmayı gerektirmez, bu sayede yeni veriyi sisteme tanıtır sistemi eğitmek ve buna göre aykırılığı tespit etmek kolaylaşır. Fakat küme boyutunun belirlenmesi oldukça sıkıntılı bir süreçtir. Ayrıca değişkenler arası mesafenin hesaplanması da oldukça maliyetli bir işlemdir.

Spektral ayrışma temelli yaklaşımlar aykırı tespiti için temel bileşen analizi kullanırlar. Sınıflandırma tabanlı yaklaşımda genellikle karşımıza Bayes Ağı ve Destek Vektör Makineleri modelleri karşımıza çıkar. Bayes ağ tabanlı yaklaşımlar olasılıksal grafik modeli değişkenleri ve bunların olasılıksal bağımsızlıkları ile temsil edilir. Grafikteki düğümler değişkenlere kenarlar koşullu bağıntılara karşılık gelir. İki düğüm arasındaki bağıntıya koşul kenar denir. Koşul kenar iki düğümü sebep sonuç ilişkisi ile birbirine bağlar. Önceden bilinen ya da deneyimler sonucu elde edilmiş bilgiler ışığında koşullu olasılık değerleri belirlenir. Değişkenler arasındaki olasılıksal bağımsızlıkların derecesine göre üç ana gruba ayrılırlar; Naïve Bayes, Belief Bayes ve Dinamik Bayes Ağlarıdır [105].

### 3.2.3.2. Bayes Ağ Tabanlı Yaklaşımlara Ait Gerçekleşmiş Çalışmalar

Elnahrawy ve Nath [108] yerel aykırı gözlemleri ve hatalı düğümleri teşhis etmek için Naive Bayes Ağ tabanlı bir model sunmuşlardır. Sensör düğümler arasındaki uzay zaman ilişkisini yakalamak için oluşturdukları tekniğe göre gruplara bölünmüş veri kümesindeki her bir düğüm yerel olasılıkları hesaplar. Eğer algılanan veri diğer sınıflardaki veriden farklı bulunursa bunun aykırı gözlem olduğu varsayılır. Ayrıca eksik veri okumalarını saptamak için kullanılır ve bunun için herhangi bir eşik değerine ihtiyaç duyulmaz. Fakat düğümler birbirinden bağımsız ve birbiri hakkında veri tutmadığı için ağa eklenen yeni düğümler ile nasıl komşuluk kuracağını bilemez.

Janakiram ve arkadaşlarının [109] sensör veri akışındaki yerel aykırı gözlemleri tespit etmek için ortaya sunduğu teknik Bayes Belief Ağ modeline dayanmaktadır. Bu teknik sensör düğümlerinin gözlemleri arasındaki ilişkiyi irdelemek için kullanılır. Her düğüm kendi okuma davranışlarının yanı sıra komşusunun verilerinin aykırılığını da algılamak için ağ eğitir. Bayes İnanç Ağları verinin kısmen ulaşılamaz olduğu durumları modellemekte daha etkilidir. Naive Bayes Ağlar ile karşılaştırıldığında koşullu bağımlılıkların tespitinde daha doğru sonuçlar vermektedir. Fakat bu teknikte dinamik ağ topoloji değişikliğine ayak uyduramaz.

Hill ve arkadaşları [110] tarafından çevresel sensör düğümlerdeki yerel aykırı gözlem verilerini tespit etmek için Dinamik Bayes Ağ tabanlı iki teknik sunulmuştur. Tekniklerden biri çok değişkenli verilerde sadece bir ölçüm durumu olduğunu ve mevcut durumun da sadece onun tarihsel geçmişine bağlı olduğunu varsayar. Bu teknik ile en son veri kullanılarak bir sonraki olasılık hesaplanır. Eğer hesaplanan veri beklenen aralıkta çıkmaz ise aykırı gözlem olarak tespit edilir. Diğer teknik biraz daha karmaşıktır; birde çok durumun ölçümü aynı anda yapılmaya çalışılır. Dinamik ağ topolojisindeki gibi gelişime açıktır, sisteme yeni dâhil olan nesnelere de kapsayan güncel durum üzerinden işlem yapmak için kullanılır.

Sensör düğümlerindeki bir veya daha fazla niteliğin belirli bir zaman aralığında, eğilimindeki değişim ve sapma durumlarını incelemek, belirlenen zaman aralığında ölçüm, tahmin değerlerini karşılaştırmalı olarak inceleyerek sapmayı tespit etmek için

zaman serilerini kullanan Bayes ağı yaklaşımı tercih edilmiştir. Sınıflandırma tabanlı yaklaşımların en önemli dezavantajı kendi hesaplama karmaşıklıkları ve uygun çekirdek fonksiyonunun seçimidir. Sensör ağı ne kadar büyük olursa eğitimin doğruluğunu sağlamakta o oranda zorlaşır.

### 3.2.3.3. Dinamik Bayes Ağı İle Hatalı Sensör Tespiti

Bayes ağı değişkenler arasındaki olasılıksal ilişkileri modelleyen bir grafiksel modeldir [111]. Değişkenler arasındaki ilişkiyi modellediği için ilişkinin bozulması durumunu belirlemek için en ideal yöntemlerden biridir. İkinci en önemli avantajı ise nedensel ilişkileri temsil etme kabiliyetlerinden gelmektedir ve sonuç tahmini için oldukça elverişlidirler [112]. Sonuç olarak hem olasılıksal hem de nedensel temeller üzerine kurulu bir yöntem olmasından dolayı veriye ait bilgileri birleştirerek problemleri modellemek için kullanılabilirler [111]. Bu bizim durumumuzda aykırı gözlemleri tespit etmedir.

Zamansal analizde kullanılan Dinamik Bayes Ağı (DBA) bir dizi değişkenin stokastik evriminin modellenmesinde kullanılır [105]. DBA'da akış her zaman ileri doğrudur bu sebeple gelecekteki olaylar ile şu an arasında nedensellik bağı bulunmaktadır. Bu sebeple DBA'nın grafiksel gösterimi birinci dereceden bir Markov model şeklini alır. Yani her değişken doğrudan doğruya önceki değişkenden etkilenir.

DBA, BN'nin zaman dilimindeki uzantısı [113] olarak tanımlanabilir bu sebeple öncelikle düğümleri, düğümler arası ilişkiyi, hatalı düğümü modellemek için EM algoritması kullanarak sabit bir BN oluşturulmuştur. Buna göre her bir düğümün iki durumu bulunmaktadır 0 (normal) veya 1 (hatalı). DBA da zamanlar arası dolaşım için geçiş özellikleri kullanılmıştır. Sensör düğümlerden oluşan sistemimizde her sensörün görevi her an değişebileceği için dinamik bir yapı söz konusudur. Modelini daha detaylıca incelemek için hata modeli bölümüne bakınız.

### 3.2.3.4. Hata Modeli

Hata tespit modelini kurarken belirli varsayımlarda bulunulmuştur. Bunlardan ilki KAA'da bulunan tüm düğümlerin aynı iletim menziline olduğudur. Bir diğeri önceden belirlenmiş olan bölgeye tüm düğümlerin rastgele dağıtılacağıdır. Burada modeli

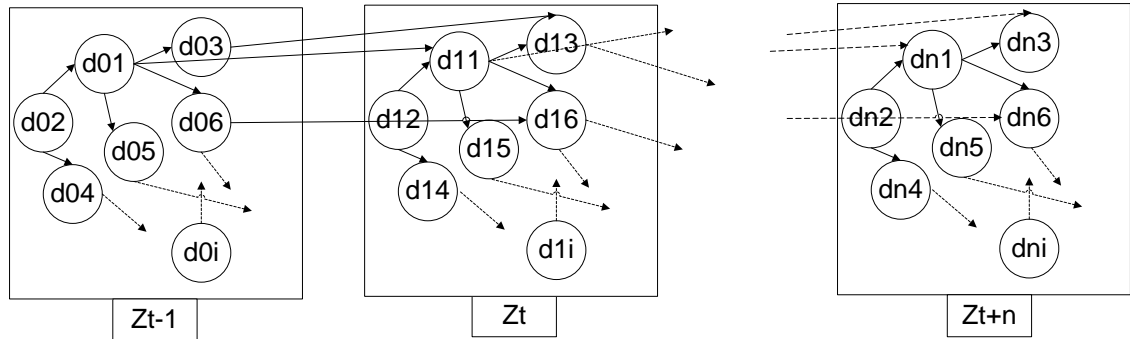
oluştururken amacımız veri iletiminde sorun yaşayan ve hatalı sonuç üreten düğümleri tespit ederek bunları devre dışı bırakmaktır. Sadece zaman iletimde sorun yaşayan veya ara sıra hesaplamalarda hata yapan bu süre haricinde hala bilgi yönlendirme yeteneğine sahip olan düğümler kullanılabilir olarak değerlendirilmiştir. Sadece kalıcı bir hatası olan veya pil ömrü tükenmiş olan düğümler arızalı olarak belirlenmişlerdir.

Sistem modellenirken bağımsız küçük kümeler halinde çevrimsiz çizge ile tanımlanmış ve her kümeye bir adet haberleşmeden sorumlu düğüm konulmuştur. Böylelikle haberleşme halindeki düğüm sayısı azaltılarak sistem karmaşıklığının önemli ölçüde önüne geçilmesi hedeflenmiştir.

$$D_m = \{ d_1, d_2, \dots, d_m \} \quad (3.3)$$

Her özellik  $D_m$  ağı içerisinde bir boğum olarak gösterilmiştir. Her boğum iki düğüm arasındaki özelliğin olasılıksal bağımlılıklarını göstermektedir, boğumlar arası bağ bulunmaması düğümlerin birbirlerinden bağımsız olduğunu göstermektedir. Her boğum kendisinin üstündeki  $\prod(D_m)$  boğumunda kendisi için saklanan olasılık değerini almaktadır. Üst boğumu bulunmayan düğümler sadece altlarındaki  $D_m$  boğumları için olasılık dağılıma sahiptirler. Bayes ağının içindeki tüm boğumlar bir vektör içinde toplanmıştır.

Dinamik Bayes ağı ve zaman algısı Şekil 3.12’de ki gibi ifade edilebilir.



Şekil 3.12 : Düğümlerin zamandaki değişimini gösteren kesitler

Temsili gösterime baktığımızda DBA’da  $(d_{01}, d_{11} \rightarrow)$  düğüm çifti olarak tanımlanabilir.  $d_{01}$  bir önceki zaman dilimini ifade eder bu da yönlendirilmiş çizge grafiğinde

$P(Z_t | Z_{t-1})$  olarak gösterilir. Burada amaç düğümün bir sonraki olasılığını hesaplamaktır  $P(d_{11}|d_{01},d_{12})$ , burada  $d_{12}$  gözlem değeridir.

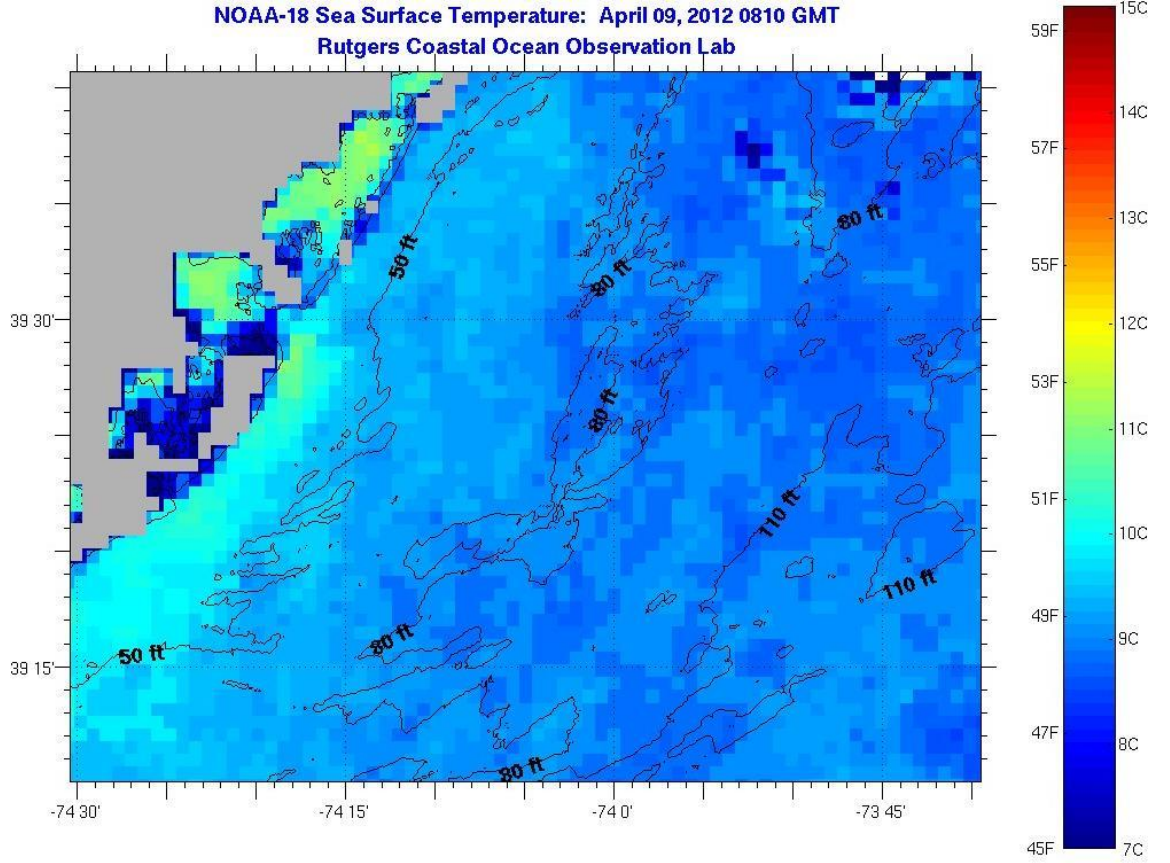
$$P(Z_t | Z_{t-1}) = \prod_{i=1}^N P(Z_t^i | \pi(Z_t^i)) \quad (3.4)$$

$Z(i,t)$  t zaman dilimindeki gizli bir düğüm, gözlem düğümü veya kontrol düğümünü ifade eder.  $P(Z(i,t))$ ,  $Z(i,t)$  'nin haberleşmeden sorumlu düğümüdür ve t veya t-1 zaman diliminde olabilir. İlk zaman dilimindeki düğümlerde ilişkili bir parametre bulunmamaktadır ama ikinci zaman diliminde koşullu olasılık dağılımı bilgileri içeren ilişkili düğümler bulunmaktadır.

Burada amaç geçmiş verilerden yararlanarak koşullu olasılık değerleri belirlemek ve hatalı düğümleri tespit etmektir. Değer belirlenirken hangi değişkenlerin ilgili değişkeni etkilediği ve hangilerinden etkilendiği bilgilerine dikkat edilir. Arızası bulunmayan sensörlerin komşuluklarında da herhangi bir sorun olmadığı varsayımında bulunulmuştur.  $d_m$  ve  $d_n$  birbirine komşu düğümler ve  $x_m$  de  $d_m$  düğümünde algılanan veri olsun.  $|x_m - x_n| \leq P(e)$  durumunda veri iletiminde herhangi bir problem yoktur.  $P(e)$  uygulamadan uygulamaya değişebilmektedir örneğin sensör düğümler ve bunların komşuları aynı sıcaklığı ölçüyor ise  $P(e)$ 'nin küçük olması beklenir. Yerel ikililerden oluşan düğümlerde algılanan veri sorunsuz ve doğru şekilde iletiliyor ise  $P(e)$  sıfırdır.

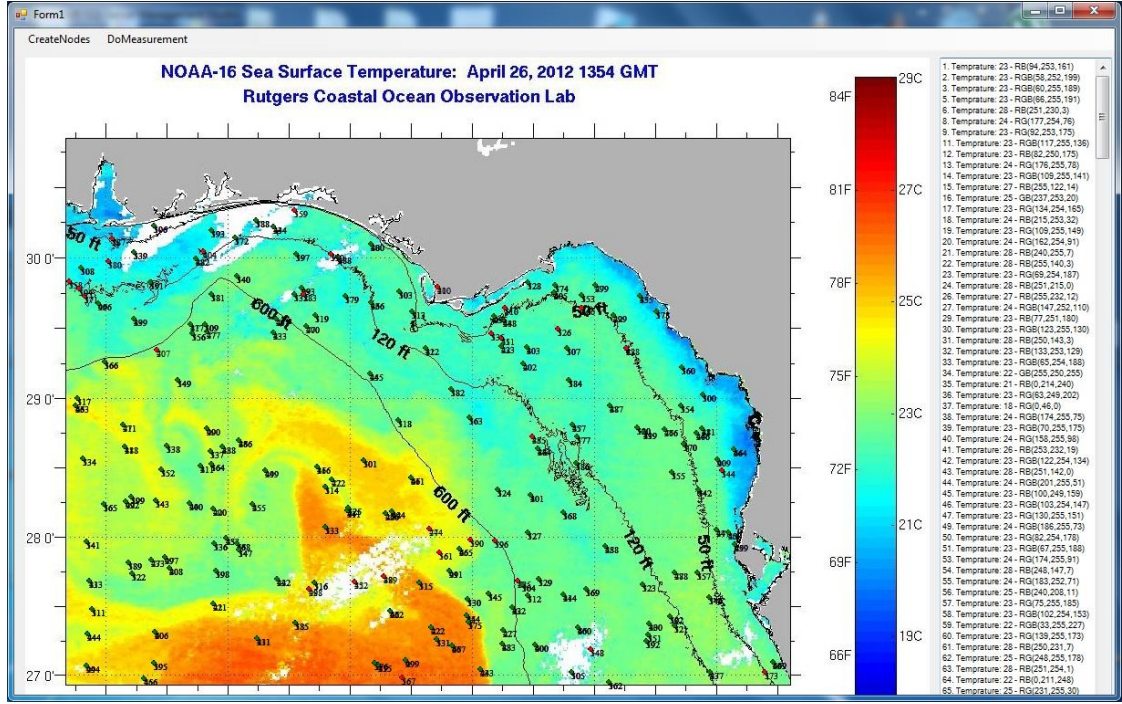
### 3.2.3.5. Kullanılan Veri Kümesinin Oluşturulması

Hatalı ölçüm yapan düğümlerin tespitinde (aykırı gözlem teşhisi) kullanılmak üzere sentetik veri üretilmiştir. Rutgers Üniversitesi'ndeki K1y1 Okyanus Gözlem Laboratuvarının (Coastal Ocean Observation Lab) [114] çalışmalarından deniz suyu sıcaklıkları alınmıştır. Laboratuvarın NOAA-12, 14, 15, 17, ve 18 uydularından aldığı fotoğrafların günün 3 farklı zaman dilimi kullanılmış ve 4 aylık bir zaman serisi oluşturulmuştur. Deniz suyu sıcaklıklarının gösterimi Şekil 3.13'de sunulmaktadır.



4 aylık süre için günde 3 farklı zaman diliminden toplanan 362 uydu görüntüsü kullanılarak deney simülasyonu gerçekleştirilmiştir. Her simülasyonda farklı sayıda ölçüm yapan sensör düğümler, uydu görüntülerinin deniz kısımlarına yerleştirilmiştir. Düğümler yerleştirilirken görüntülerin RGB renk uzayı kullanılmıştır. RGB renk uzayı 'Red' 'Green' 'Blue' yani 'Kırmızı' 'Yeşil' 'Mavi' ana renklerinden oluşan ve en sık kullanılan renk uzayıdır. Tüm sıcaklık değerleri bu renklerin karşılımları ile ifade edilmişlerdir yani her bir piksel 3 renk kodundan oluşmaktadır (0 dan 255'e kadar). Uydu görüntülerindeki renk gösterimlerinden, verilen sıcaklık paletindeki (R,G,B) renk kodları tespit edilerek bunlara karşılık gelen santigrat değerleri saptanmıştır. Saptanan bu aralıklara göre sanal olarak yerleştirilmiş düğümler aynı koordinatlarda olacak şekilde tüm 362 uydu görüntüsüne uygulanarak istenilen sayıda zaman serisi ölçümleri üretilmiştir.

Simülasyonda ölçüm yapan düğümlerin konumlandırılmasında rastlantısal dağılım kullanılmıştır. Ölçümlerin üretildiği simülasyonun ekran görüntüsü aşağıdaki Şekil 3.14'de gösterilmektedir.



Şekil 3.14 : Veri Kümesini Üreten Simülasyon Yazılımının Ekran Görüntüsü

Her bir düğümün bulunduğu noktaya göre aldığı renk değerleri ve buna karşılık gelen sıcaklık değeri simülasyon ortamından izlenebilmektedir. Simülasyon sonucu üretilen veriler uydu görüntüsündeki tarih ve zaman etiketi kullanılarak bulut üzerindeki depolama alanın kaydedilmiştir. Böylelikle bir sonraki adım aykırı gözlem (outlier) yapacak olan hatalı sensör düğümlerin oluşturulmasıdır. Her simülasyonda parametrik olarak değişen yüzdesel oranda hatalı düğümler seçilmiştir. Gerçek hayatta hatalı çalışan sensör düğümler iki farklı davranış şekli sergiler. Ya tamamen sabit bir değere takılı kalıp yanlış ölçüm gönderir, ya da çeşitli sebeplere bağlı olarak belirli bir aralıkta değişen hatalı değerleri iletir. Bu sebepler çoğunlukla cihazın fiziksel hasar görmesinden, nem gibi elektronik cihazları uzun vadede etkileyen ortam karakteristiklerine maruz kalmasından veya çok yakınında ölçüm değerinin değişmesine neden olan bir mani bulunmasından kaynaklanmaktadır. Bu nedenlerden ötürü seçilen bu düğümlere ait ölçümler üç farklı yöntem ile deforme edilmiştir.

İlk deforme yönteminde ölçüm değerleri sabit bir değer ile değiştirilerek tüm zaman serisi boyunca hatalı bilgi gönderimi oluşturulmuştur. İkinci yöntemde ise işaretli düğümlerin ölçümleri { +10, -10 } aralığında rastlantısal olarak üretilen bozulma değerleri eklenerek, sabit olmayan hatalı ölçümler oluşturulmuştur. Üçüncü yöntem ilk iki yöntemin beraber kullanılması olarak hem sabit hem de değişken bozulmuş verileri meydana getirmiştir. Aykırı gözlem teşhisi yapacak servis bu üç farklı yöntem ile bozulmuş ölçüm değerler kullanarak sonuç üretmesi istenmiştir.



## 4. BULGULAR

### 4.1. YAYINLA/KATIL MİMARİSİ İLE İLGİLİ YAPILAN ÇALIŞMALAR VE EDİLEN BULGULAR

#### 4.1.1. Yayınlayıcı Uygulama: WatchMan

Geleneksel Yayınla/Katıl sistemlerinin en önemli ögesi olan yayınlayıcının davranış şeklini sergileyen uygulama *WatchMan*'dir. Bu uygulama KAA konfigürasyonunda tanımlanmış olan zaman aralığında sıcaklık ölçümü gerçekleştirir ve sistemdeki olayları üretir. Cihaz üzerindeki ADT7411 duyarga modülü ile -40 ile +125 santigrat derecelik aralıkta ortam sıcaklığını ölçümleyebilmektedir. Sıcaklık ölçümü yapılırken kullanılan kütüphaneler ve methodlar şu şekildedir;

```
Import com.sun.spot.sensorboard.EDemoBoard;
```

```
Import com.sun.spot.sensorboard.peripheral.ITemperatureInput;
```

```
// Isı sensörünün örneğini oluştur.
```

```
ITemperatureInput SicaklikSensoru =
```

```
EdemoBoard.getInstance().getADCTemperature();
```

```
// Tanımlanan örnekten ilgili methodu kullanarak değeri değişkene ata
```

```
double Santigrat = SicaklikSensoru.getCelsius();
```

Yapılan ölçümün iletiminde Düğüm No ve ölçüm değeri arasında: işareti konularak veri iletimi sağlanır (Örneğin 298:20,54).

*WatchMan* ölçüm aralıklarını takip ederken, Sun SPOT sensörün bünyesinde bulunan T91 zamanlayıcı (timer) sayacı kullanılmaktadır. Uygulama, yaptığı ölçümü bölgedeki aracı düğümüne iletilmesinin ardından tüketimi azaltmak için uyku durumuna geçiş yapar ve bir sonraki planlı ölçüm için beklemede kalır. Uyku durumuna geçerken statik değişkene true ataması yapılır ve ilgili method çağrılır.

```
// Yarım saat süre ile ölçüm yapan düğümün uyku konumuna getirilmesi
```

```
ISleepManager UykuYonetimi = Spot.getInstance().getSleepManager();
```

*UykuYönetimi.ensureDeepSleep(360000);*

Bir sonraki uyanmada öncelikli olarak yapılan işlem bölgedeki aktif aracı düğümün işlevsel olup olmadığının kontrolüdür. Eğer bölgeden sorumlu olan aracı düğüm pil bitmesi veya başka bir sebeple yanıt vermiyorsa, bölgeye yani bir yetkili aracı düğüm atanması gerekir.

#### **4.1.2. Aracı Uygulama: Mediator**

Aracı uygulama davranış biçimi olan *Mediator*, ölçüm bölgesindeki üretilen olayların daha üst seviyeye aktarılması için çalışır. Toplanan düğümsel ölçümlerin merkez istasyona iletilmeden önce birleştirilmesi gerekir. Aynı küme içerisindeki düğümlerin tüm iletileri zaman aşımı beklenerek, birleştirilir. Gönderilecek bireysel ölçümler, araya // konularak birbirlerinden ayrılabilir şekilde bir iletiye dönüştürülür (298:20,54//275:21:67//304:21,20//299:20,79).

Düğümlerin çakar bellekleri (flash memory) üzerinde, bölgesel toplanmış olan ölçümlerin hangi yönlendirme rotası üzerinden iletileceğinin bilgisini tutan yönlendirme tablosu bulunmaktadır. Bu yönlendirme tablosu bir XML doyası olup, merkez düğüme (istasyon) olan veri sevkiyatının adımlarını tutmaktadır. Aracı uygulamaların aktarımı yapabileceği ikisi alternatif, biri ana yol olmak üzere toplam üç adres bulunmaktadır. Aracı uygulama ana yol adresine veri iletimi için gönderilen eşleme paketine yanıt alamazsa, alternatif rota üzerinden bilgiyi ulaştırmak için diğer adresleri dener.

Aracı uygulamanın, üretilen verileri merkez düğüme iletmenin haricinde ikinci bir görevi bulunmaktadır. Bu görev bulut hizmeti üzerinde çalışmakta olan hatalı düğümlerin tespit edilmesi için geliştirilmiş olan aykırı gözlem teşhisi destek vermektir. Yayınla/Katıl modelinde tanımlı olan kontrol parametresine bağlı olarak, belirli periyotlarda topladığı ölçüm değerlerinin uyumluluğunu analiz eder. Basit bir aykırı gözlem teşhisi yöntemi olan bu fonksiyon ile üretilen veriler arasında hatalı ölçüm aranır ve hatalı olduğu tespit edilen ölçüm değeri işaretlenerek merkez düğüme iletilir. Bu sayede bulut ortamındaki servisin gözetmeli (supervised) olarak çalışması sağlanmış olur.

Aykırı gözlem tespiti yapılabilmesi için verilerin normal dağılımlı ve varyanslarının homojen olması gerekmektedir. Bunun için aşağıdaki adımlar izlenilerek sensör düğümleri içinde basit bir aykırılık tespiti yapılmıştır.

Adım 1: Tüm sensör düğümlerinin ölçtüğü sıcaklık verileri  $x_i$  kullanılarak ortalama sıcaklık  $\bar{x}$  bulunmuştur.

$$\bar{x} = \frac{1}{N} \sum_{i=0}^N x_i \quad (4.1)$$

Adım 2: Tüm veri için standart sapma  $\sigma$  hesaplanmıştır.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}} \quad (4.2)$$

Adım 3: Düğümlerden ölçüden verinin normal verilere dönüştürülerek karşılaştırma yapılabilmesi için her bir veri için z skoru hesaplanmıştır.

$$z = \frac{x - \bar{x}}{\sigma} \quad (4.3)$$

Farklılığı 0.05'in altında kalanlar aykırı gözlem olarak belirlenmiş ve bulut ortamına bu bilgide eklenerek aktarılmıştır.

### 4.1.3. Geçit Uygulaması: DataAggregator

Merkez düğümün USB ile bağlı olduğu bilgisayar üzerinde çalışan ve üretilen tüm verileri bulut ortamındaki veritabanı hizmetine kaydeden uygulamadır. Aracı düğümlerin yönlendirdiği *RadioStream* iletilerini dinler ve // karakterini taban olarak ölçümleri ayrıştırır. Her ayrıştırılmış ölçümü bulut üzerindeki SQL Azure veritabanına zaman damgası ile beraber kaydeder.

#### 4.1.4. Bilgilendirme Uygulaması: Notifier

KAA gözlem alanı içerisinde üretilen ölçüm bilgilerinin veritabanına kaydedilmesinin ardından katılımcıların tanımlamış oldukları ölçüm filtreleri taban alınarak içerik kontrolü gerçekleştirilir.

Katılımcıların ilgi duyduğu yayıncı topluluğuna ait üretilen olaylara, veritabanında saklanan filtre değeri kıyaslanır. Örnek vermek gerekirse ağaçlık alan içerisindeki düğümlerin yaptığı ölçümlerin 30 santigrat dereceyi geçmesi durumunda, bulut üzerinde çalışmakta olan bilgilendirme servisi katılımcıyı mail yolu ile uyaracaktır. Bulut üzerinde çalışmakta olan *Worker Role* uygulaması olan Notifier, sisteme dâhil olan tüm yeni kayıtları anlık olarak kontrol eder ve kirik zamanlı bilgileri katılımcılara ulaştırır.

#### 4.1.5. Yönetim Uygulaması: Portal

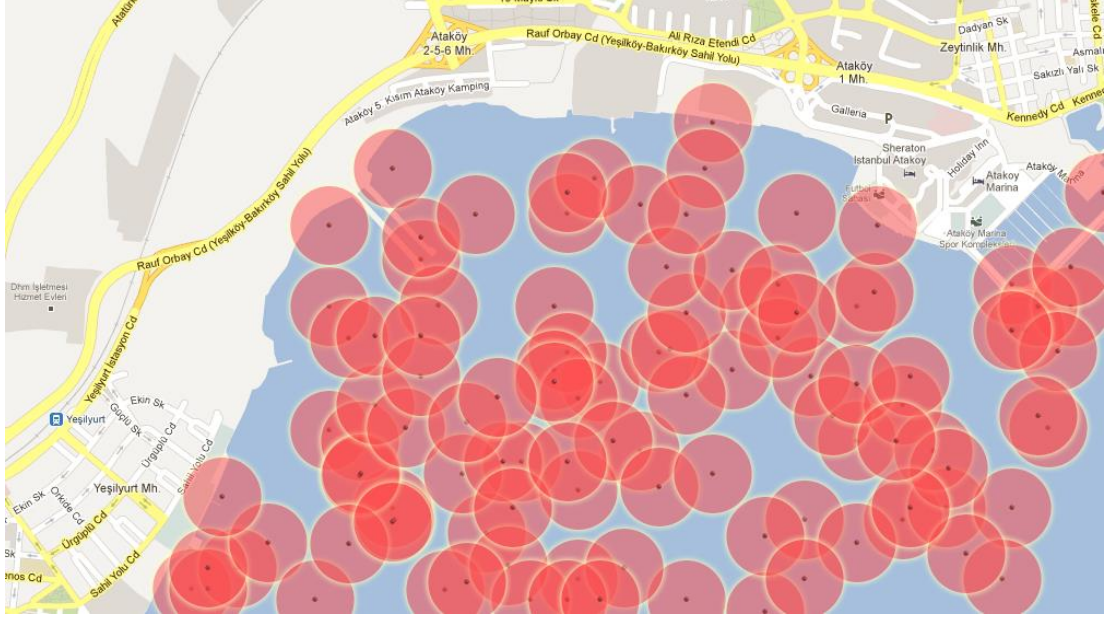
Portal uygulaması katılımcıların ilgi duydukları yayıncılar ile eşleşmelerine imkân veren bir *Web Role* uygulamasıdır. Bulut üzerinde barındırılan bu uygulama katılımcıların anlık yük yoğunluğunu dengelenebilmektedir. Ana ekranı güvenli giriş sonrası, abone olunmuş yayıncıların zaman serisi şeklindeki ölçüm değerlerini göstermesidir. *Portal* uygulaması üzerinde mevcut bulunana sorgu filtreler ile ilgili zaman aralıklarındaki ölçüm değerlerini tarayıcı bulunan herhangi bir platform üzerinden göstermektedir. Ayrıca aboneliğin yönetimine de imkân veren ara yüzleri barındırmaktadır.

ZAMAN / DÜĞÜM	278	285	287	300	309	341
19.12.2012, 19.01	21,07	22,93	22,26	28,87	21,03	22,55
19.12.2012, 20.00	19,07	16,25	20,89	19,82	24,09	23,31
19.12.2012, 21.01	22,18	17,29	23,41	17,81	21,46	21,71
19.12.2012, 22.01	22,95	22,00	18,34	24,21	18,88	20,66
19.12.2012, 23.01	18,16	26,32	17,66	20,68	25,44	21,21
20.12.2012, 00.00	23,78	17,00	20,11	19,50	20,29	19,95
20.12.2012, 01.01	11,91	21,89	19,10	23,04	20,78	21,91
20.12.2012, 02.00	19,63	24,79	24,10	22,60	23,99	23,40
20.12.2012, 03.01	16,64	22,74	17,39	20,07	27,67	16,32
20.12.2012, 04.00	27,09	21,84	16,59	26,17	25,24	15,84
20.12.2012, 05.01	22,69	13,86	21,06	23,72	21,38	19,93
20.12.2012, 06.01	24,21	17,63	18,45	26,11	19,54	17,45
20.12.2012, 07.01	23,31	21,02	19,83	26,10	17,53	28,63
20.12.2012, 08.01	23,83	21,67	22,88	22,23	21,08	24,64
20.12.2012, 09.01	20,40	20,91	20,94	19,07	19,88	24,00
20.12.2012, 10.01	15,38	21,09	22,63	16,41	23,48	18,91
20.12.2012, 11.01	24,00	22,93	20,25	15,90	18,88	19,65

Şekil 4.1 : Portal Uygulaması Ekran Görüntüsü

#### 4.2. GENETİK ALGORİTMA KULLANILARAK YAPILAN DAĞITIMIN İYİLEŞTİRİLMESİ İLE İLGİLİ SİMULASYON ÇALIŞMALARI VE ELDE EDİLEN BULGULAR

En ideal dağıtımın sağlanabilmesi için farklı parametrelerde çeşitli testler yapılmıştır. Genetik algoritmanın ideal halinin tespiti için Google Maps'in [115] sunduğu harita hizmeti temel alınarak simülasyon için İstanbul Türkiye sahil kesiminin su sıcaklıklarını kontrol eden bir senaryo tercih edilmiştir. Simülasyonda mavi ile ifade edilen deniz kesimlerine su sıcaklıklarını ölçen türdeş olmayan sensörler rastgele olarak dağıtılmıştır. Dağıtımın ölçüm yapılacak tüm alanı kapsayacak ideal forma çevrilmesi için iyileştirme testleri uygulanmıştır. Şekil 4.2'de rastgele yerleştirilen düğümleri ve kapsama alanları gösterilmektedir. Simülasyonlarda türdeş olmayan düğümler kullanıldıklarından ölçüm kapasiteleri farklılıklar göstermektedir. Şekil 4.3 ise; iyileştirme yapılmış olan dağıtımı göstermektedir. Yer değişikliği yapılmış olan düğümler yeşil ile gösterilmektedir.



Şekil 4.2 : Simülasyon dağıtım görüntüsü



Şekil 4.3 : Simülasyon dağıtımının iyileştirilmiş halinin görüntüsü

Topluluk (population), iterasyon sayısı, mutasyon olasılığı, çaprazlama olasılığı, en iyi kromozom oranı ve kromozom seçim algoritması parametrelerine farklı değerler verilerek 10 farklı test gerçekleştirilmiştir. Testlerin gerçekleştirildiği simülasyon ortamı Intel® Core™ i7-3770 işlemcili ve 8GB bellekli bir masaüstü bilgisayardır. Yapılan testlere ait uygunluk değerlerini ve işlem sürelerini gösteren Tablo 4.1 aşağıda sunulmaktadır.

Tablo 4.1 : Test Sonuçları

Sonuçlar / Değişkenler	Topluluk Boyutu	Tekrar Sayısı	Mutasyon Olasılığı	En İyi Kromozom Oranı	Seçim Yöntemi
İşlem Süresi = 35:41 Uygunluk Değeri = 65630	500	50	0,01	0,3	Roulette
İşlem Süresi = 3:21:17 Uygunluk Değeri = 69080	3000	50	0,01	0,3	Roulette
İşlem Süresi = 52:48 Uygunluk Değeri = 66570	1000	100	0,01	0,3	Roulette
İşlem Süresi = 19:18:37 Uygunluk Değeri = 82235	5000	200	0,01	0,3	Roulette
İşlem Süresi = 1:03:56 Uygunluk Değeri = 615052	1000	50	0,001	0,3	Roulette
İşlem Süresi = 23:38 Uygunluk Değeri = 62330	1000	50	0,2	0,3	Roulette
İşlem Süresi = 25:05 Uygunluk Değeri = 67290	1000	50	0,01	0,7	Roulette
İşlem Süresi = 59:45 Uygunluk Değeri = 67190	1000	50	0,01	0,3	Tournament

Görülmektedir ki; işlem süresine etki eden en önemli faktörler topluluğun boyutu ve iterasyon sayısı parametreleridir. Topluluğun boyutunu büyütürük yüksek uygunluk değerleri elde edilmesine karşın, genetik algoritmanın çalışma süresi logaritmik oranda artmaktadır. Yapılan testlerde elde edilen bir diğer gözlem mutasyon olasılığının yüksek tutulması işlem süresinde kısalmaya yol açtığıdır. Fakat işlem süresinde kazanılan bu iyileşme, uygunluk değerlerindeki düşüşü beraberinde getirmektedir. En ideal mutasyon olasılığı test tecrübelerinden %1 olarak tespit edilmiştir. Benzer şekilde; en iyi kromozom oranının kullanımı, mutasyon olasılığı ile aynı etkiyi göstermektedir. Rulet seçimin kullanıldığı testler Turnuva seçiminin kullanıldığı testlere göre daha çabuk sonuçlandığından, rulet seçimi varsayılan seçim algoritması olarak belirlenmiştir.

Lokal bir bilgisayarda yapılan ilk testlerden, en ideal parametreler belirlenerek genetik algoritma konfigürasyonu oluşturulmuştur. Uzun işlem sürelerinin bulunması, çalışma ortamının bulut mimarisine taşınacak olmasından ötürü problem teşkil etmemektedir.

Bulut ortamının [116] ölçeklenebilir (scalable) oluşu sayesinde, kullanılan işlemci gücünü anlık olarak değiştirilmesi mümkün olabilmektedir. Bu özellik kullanılarak yüksek hesaplama gücü gerektiren işlemlerin kısa sürede sonuçlandırılması mümkün olmaktadır. Tablo 4.1’de gösterilmiş olan beşinci testi ait parametreler ile 64 core işlemcili 32GB bellek tahsisli bir bulut ortamındaki hesaplanma süresi 16dk 23sn olarak ölçülmüştür. Bu süre lokal masaüstü bilgisayarın genetik algoritmayı tamamlama süresinin 4’te birinden bile daha kısadır. Bulut ortamında ihtiyaca göre işlemci ve bellek gücünü değiştirebilme kabiliyetinin sunulması, bu tarz projelerde kullanılabilirliği arttıran en önemli karakteristiklerden biri olmuştur.

Yapılan çalışma ile internete kapalı olan KAA verilerinin bulut ortamına taşınarak mevcut problemlerin çözülmesi hedeflenmiştir. Geliştirilen yazılımlar *WatchMan*, *Mediator* ve *DataAggregator* ile bir yayınla katıl mimarisi oluşturularak, veri merkezli (data-centric) bir iletişim sağlanmıştır. Bu sayede veri iletimindeki kayıplar en az indirilmektedir.

Bulut ortamı olarak tercih edilen Microsoft Azure, verilen depolanmasına ve dağıtımın iyileştirilmesi için çalışan genetik algoritmanın bulunduğu web uygulamasına ev sahipliği yapmaktadır. Genetik Algoritmanın en ideal şartlarda çalışmasını sağlayabilmek için simülasyon ortamında çeşitli parametreler ile testler yapılmıştır. Bu testlerin sonuçlarının yorumlanması ile bulut ortamında kullanılması düşünülen gerçek parametrelere ulaşılmıştır.

Genetik algoritma ile yer iyileştirmesi yapılacak düğümlerin tespit edilmesi için senaryo bazlı kurallar uygulanmıştır. Konumundan dolayı ölçüm kapasitesi efektif olarak kullanılmayan düğümler ve birbirlerine çok yakın konumlandırılmış çakışan düğümler tespit edilmiştir. Bu düğümlerin tüm sistemin ölçüm kapasitesini arttıracak ideal (x,y) koordinatlarına yerleştirilebilmesi için bulut üzerindeki tüm veriler genetik algoritma tarafından kullanılmıştır. Yapılan testler göstermektedir ki ilk dağıtım sonrası iyileştirilen sistem %99’a kadar verimli hale getirilmektedir. Ayrıca genetik algoritma ile önemli bölgelerin daha yoğun gözlem altına alındığı dağıtım iyileştirmeleri de sağlanabilmiştir.



### 4.3. HATALI ÖLÇÜM YAPAN DÜĞÜMLERİN TESPİTİ İLE İLGİLİ SİMULASYON ÇALIŞMALARI VE ELDE EDİLEN BULGULAR

Günde 8 saat aralıklarla alınan sensör düğüm ölçüm verilerinin bulut ortamına aktarılması ile her günün sonunda veritabanındaki tüm veriler aykırı gözlem tespiti algoritmasına gönderilir. Algoritma hata tespitini yaparken geleceğe yönelik çıkarım yapmak için iki ayrı zaman dilimi kullanmaktadır. Bunlar; şimdiki zamandan iki dilim sonrası ( $t=2$ ) ve şimdiki zamandan beş dilim sonrası ( $t=5$ ) olarak belirlenmiştir. Geçiş olasılıklarını aşağıdaki gibi tanımlanmıştır.

$$P(Z_t | Z_{t-1}) = \begin{cases} 0.9 \cdots eger \dots Z_t = Z_{t-1} \\ 0.1 \cdots eger \dots Z_t \neq Z_{t-1} \end{cases} \quad (4.4)$$

Her hatalı düğüm grubu için testler 100'er defa tekrarlanmıştır. İlk zaman dilimi için hata olasılıkları geçmişteki veriler incelenerek tahmini olarak belirlenmiştir. Sonraki geçişler ise  $P(Z_t | Z_{t-1})$  yardımı ile üretilmiştir.

Toplam düğüm sayısı 100 iken hatalı sonuç üreten düğümün kestirimi  $t=1$  anında ortalama %71 doğrulukta,  $t=2$  anında ortalama %73 doğrulukta,  $t=3$  anında ortalama %71,6 doğrulukta ve  $t=4$  anında ortalama %71 doğrulukta saptanmıştır. Aynı deneyin düğüm sayısı arttırıldığı senaryolarında daha yüksek kestirim yüzdesinin elde edildiği gözlemlenmiştir. Bütün kesitirim deneylerine ait sonuçlar aşağıdaki Tablo 4.2'de gösterilmektedir.

Tablo 4.2 : Hatalı Düğüm Kestirim Deney Sonuçları

Düğüm Sayısı	Zaman Dilimi	1			2			3			4		
	Hatalı Düğüm Oranı	0%	10%	30%	0%	10%	30%	0%	10%	30%	0%	10%	30%
100	Yanlış Tespit	18	28	41	17	25	39	15	31	39	25	42	55
	Tespit Oranı	82%	72%	59%	83%	75%	61%	85%	69%	61%	75%	58%	45%
250	Yanlış Tespit	27	45	68	31	43	73	25	41	69	36	42	81
	Tespit Oranı	90%	82%	73%	88%	83%	71%	90%	84%	73%	86%	84%	68%
500	Yanlış Tespit	47	61	92	42	65	89	43	63	84	59	78	102
	Tespit Oranı	91%	88%	82%	92%	87%	83%	92%	88%	84%	89%	85%	80%
1000	Yanlış Tespit	61	96	134	72	93	128	69	92	128	85	86	131
	Tespit Oranı	94%	91%	87%	93%	91%	88%	94%	91%	88%	92%	92%	87%

Deneilerin tamamı yorumlandığı vakit; hatalı ölçüm yapan düğümlerin tamamı keşfedilebilmektedir. Fakat bu kadar büyük doğrulukdaki kestirim işlemi, hatalı ölçüm yapmadığı halde, öylemi gibi sınıflandırılan (false positive) seçimleri de beraberinde getirmektedir.

## 5. TARTIŞMA VE SONUÇ

Tez araştırmamın temel hedefi olan KAA bilgilerinin internete genişletilerek daha çok kullanıcı tarafından erişilebilir olması için yapmış olduğum çalışmalarda; bu hizmeti verebilmek için ihtiyaç duyulan içerik tabanlı bir Yayınla/Katıl modeli geliştirilmiştir. Geleneksel Yayınla/Katıl sistemlerin içerisinde bulunan ana aktörler; yayıncılar, katılımcılar ve bildirim mekanizması değişik platformlar üzerinde çalışacak şekilde gerçekleştirilmiştir. Günümüze kadar geliştirilmiş prototiplerin aksine bildirim servisi, bulut ortamında çalışacak bir servis olarak tasarlanmıştır. Bunun temel sebebi; bulut bilişimin bize sunmakta olduğu ölçeklenebilir, yüksek hesaplama ve depolama kabiliyeti ile servisin daha kapsamlı ve daha erişilebilir şekilde verilmesini sağlamaktır.

Bulut bilişim ortamında istek anında (on-demand) sunulabilen iş uygulamalarının kullanılması, küçük ve orta ölçekli firmalara düşük maliyet avantajını yanı sıra yüksek erişilebilirlik sağlamaktadır. Üretilen ölçüm verilerinin geleneksel KAA sisteminin bağlı olduğu yerel masaüstü bilgisayarın haricinde de bulut üzerinden erişilebilir olması pek çok avantajı beraberinde getirmiştir.

Tüm toplanan ölçüm verilerine uzun süreli depolama hizmeti verilebilmektedir. Geleneksel KAA sistemlerindeki gibi, ölçümlenen verinin uzun zaman sonunda veritabanının şişirmesi (oversized) gibi problemler yaşanmamaktadır. Başlangıçtan itibaren oluşturulan tüm kayıtların saklanabilmesi, bize geçmişe dönük sorgular yapabilme ve kestirim hesaplarının daha keskin doğrulukta çıkmasına imkân vermektedir.

Geliştirilen modelin birden fazla KAA sisteminde kullanılması durumunda; tüm veriler aynı merkezde toplandığı için katılımcıların KAA sisteminden bağımsız izleme yapmaları mümkün olmaktadır. Bu noktada abonelik; KAA sistemi seviyesinden ölçüm yapan düğüm seviyesine indirgenmiş olmaktadır. Katılımcılar bilgi aldıkları düğümlerin

hangi sistem(ler) altında çalıştıklarını bilmelerine gerek kalmadan, farklı kaynaklardan toplanmış bilgileri yalın olarak kullanabilmektedirler.

Bulut üzerine ölçüm verilerinin aktarılması ile, geleneksel KAA sistemlerinde bulunmayan bazı özellik ve servisler sunulabilir olmuştur. Tez kapsamında geliştirilmiş olan ilk servis, KAA dağıtımının iyileştirilmesi üzerinedir. Genetik Algoritma kullanan bu servis ölçüm yapılmak istenen tüm bölgenin en etkin bir şekilde kapsama alanı içerisine alınmasını öneren ideal dağıtımı göstermektedir. Hesaplamaların bulut ortamının yüksek işlemci kapasitesi ile yapılması ile sonuçların ortaya çıkması hızlanmıştır.

Toplanan verilerin sadece anlık olarak değerlendirilmemesi ve geleceği dönük tahminlerin yapılabilmesi kullanıcıların ihtiyaç duyduğu bir hizmettir. Bu doğrultuda, hazırlanmış olan diğer iki servisten ilki; sistem içerisinde aykırı gözlem yapan düğümlerin tespit edilmesine imkân vermektedir. Geçmişe dönük olarak toplanan veriler, geliştirilmiş melez aykırı gözlem teşhisi algoritması ile incelenerek, katılımcılara hatalı bilgi veren düğümler tespit edilebilmektedir. Bu sayede katılımcıların ilgi duydukları veriler daha doğru olarak sunulabilmektedir. Son olarak, çeşitli sebeplerle ölçüm yapamaz hale gelen düğümlerin bölgelerindeki takip edilmek istenen ortam karakteristiği değeri, geçmiş verilerin analizi kullanılarak tahmin edilebilmektedir.

Geliştirilen Yayınla/Katıl mimarisinin uygulanması ölçüm verilerini bulut ortamına taşımıştır. Bulut ortamına aktarılan veriler ile geleneksel KAA sistemlerinde bulunmayan pek çok hizmet sağlanabilmiştir.

Genel tez çıkarımım; KAA sistemleri gibi dağıtık mimaride çok sayıda kablosuz iletişim yapabilen sistemlerin etkin kullanılabilmesi için sorunsuz çalışan bir IPv6 ihtiyaç vardır. Bu yapı Nesnelerin İnterneti'ne (IoT) daha çabuk ulaşmamızı sağlayacak ve sistemlerin daha etkin kullanılmasına imkan verecektir. Bu zaman kadar

arařtırmamın konusu olan geit modelleri tırdeř olmayan ađların etkin iletiřimine imkân verecek yegâne özřmler olacaktır.

Üretilen ölçüm verileri, KAA sisteminin dıřına ne řekilde ıkartılırsa ıkartılsın, verinin iřlenmesi ve depolanması için Bulut Biliřim kullanılmalıdır. Bu sayede, servislerin geleneksel yaklařımlara göre daha performanslı ve daha eriřilebilir bir atıdan sunulması mümkün olacaktır.

Tez alıřmasının mevcut geliřtirmelerinde daha ileri seviye olarak alıřılabilecek konular; önerilen iletiřim modelinin tırdeř olmayan düđümler ile gereklenecek iletiřime göre uyarlanması, tercih edilen bulut biliřim mimarisi olan Microsoft Azure'un desteklemediđi fakat Amazon tarafından sunulan MapReduce tekniđinin uygulanması, kapsama alanının hesaplanmasında sıka kullanılan fakat uzun iřlem süreleri bulunan Bresenham ember algoritmasının yerine daha performanslı yöntemlerin aranması olacaktır.

Önerilen yapıda; beraber alıřan ok sayıda KAA sisteminin verileri uzun vadede deđerlendirilerek, mevcut sistemlerin genel performans analizi ve birbirleriyle olan kıyaslamaları mümkün olabilecektir.

## KAYNAKLAR

1. AKYILDIZ, I.F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E., 2002, A survey on sensor networks, *IEEE Communications Magazine* , 40 102–114.
2. HAC, A., 2003, *Wireless Sensor Network Designs*, John Wiley & Sons Ltd: Etobicoke, Ontario, Canada.
3. TUBAISHAT, M., MADRIA, S., 2003, Sensor networks: an overview, *IEEE Potentials*, 22, 20–30.
4. RAGHAVENDRA, C., SIVALINGAM, K., ZNATI, T., 2004, *Wireless Sensor Networks*, Springer: New York.
5. AKYILDIZ, I.F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E., 2002, Wireless sensor networks: a survey, *Computer Networks*, 38(4):393–422.
6. TILIUT, D.E., 2007, Battery Management in Wireless Sensor Networks, *Electronics And Electrical Engineering - Microelectronics*, Nr. 4(76).
7. ALIPPI, C., GALPERTI, C., 2008, An Adaptive System for Optimal Solar Energy Harvesting in Wireless Sensor Network Nodes, *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.55, no.6, pp.1742-1750.
8. OTTMAN, G.K, HOFMANN, H.F., BHATT, A.C., LESIEUTRE, G.A., 2002, Adaptive Piezoelectric Energy Harvesting Circuit for Wireless Remote Power Supply, *IEEE Transactions on Power Electronics*, Vol. 17, No. 5, pp. 669 - 776.
9. JOSEPH, A.D., 2005, Energy Harvesting Projects, *IEEE Pervasive Computing*, pp. 69-71.
10. JOHNSON, M., HEALY, M., VAN DE VEN, P., HAYES, M.J., NELSON, J. NEWE, T., LEWIS, E., 2009, A comparative review of wireless sensor network mote technologies, *IEEE Sensor*, pp 1439 – 1442.
11. ESTRIN, D., GOVINDAN, R., HEIDEMANN, J., KUMAR, S., 1999, Next century challenges: scalable coordination in sensor networks, *In Proceedings of ACM/IEEE MobiCom '99*, pp. 263–270.
12. Boomerang shooter detection system. <http://bbn.com/boomeran>, [Ziyaret Tarihi: 8 Ağustos 2012].

13. Dust Networks Inc. <http://www.dust-inc.com> , [Ziyaret Tarihi: 20 Kasım 2012].
14. LÉDECZI, Á., NÁDAS, A., VÖLGYESI, P., BALOGH, G., KUSY, B., SALLAI, J., PAP, G., DÓRA, S., MOLNÁR, K., MARÓTI, M., SIMON, G., 2005, Countersniper system for urban warfare, *ACM Transactions on Sensor Networks*, 1(2):153–177.
15. Center for Coastal Margin Observation and Prediction, <http://www.ccalmr.ogi.edu/corie>, [Ziyaret Tarihi: 3 Ekim 2011].
16. ENSCO Inc. <http://www.ensco.com>, [Ziyaret Tarihi: 7 Mayıs 2012].
17. BASHA, E. A., RAVELA, S., RUS, D., 2008, Model-based monitoring for early warning flood detection, *In Proceedings of ACM SenSys '08*, pp. 295–308.
18. BONNET, P., GEHRKE, J., SESHADRI, P., 2000, Querying the physical world. *IEEE Personal Communications*, 7(5):10– 15.
19. CERPA, A., ELSON, J., HAMILTON, M., ZHAO J., 2001, Habitat monitoring: application driver for wireless communications technology, *In Proceedings of ACM SIGCOMM'01*, pp. 20–41.
20. JUANG, P., OKI, H., WANG, Y., MARTONOSI, M., PEH, L. S., RUBENSTEIN, D., 2002, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, *ACM SIGOPS Operating Systems Review*, 36(5):96–107.
21. ARTIFICIAL RETINA PROJECT, <http://artificialretina.energy.gov>, [Ziyaret Tarihi: 15 Nisan 2011].
22. GAO, T., PESTO, C., SELAVO, L., CHEN, Y., KO, J. G., LIM, J.H., TERZIS, A., WATT, A., JENG, J., CHEN, B. R., LORINCZ, K., WELSH, M., 2008, Wireless medical sensor networks in emergency response: implementation and pilot results. *In Proceedings of IEEE International Conference on Technologies for Homeland Security*, pp. 187–192, Waltham, MA, USA.
23. KAHN, J. M., KATZ, R. H., PISTER, K. S. J., 1999, Next century challenges: mobile networking for smart dust, *In Proceedings of ACM MobiCom '99*, pp. 271–278, Seattle, WA, USA.

24. MALAN, D., FULFORD-JONES, T., WELSH, M., MOULTON, S., 2004, CodeBlue: an Ad Hoc sensor network infrastructure for emergency medical care, *In Proceedings of Workshop on Applications of Mobile Embedded Systems*, Boston, MA, USA.
25. WOOD, A., VIRONE, G., DOAN, T., CAO, Q., SELAVO, L., WU, Y., FANG, L., HE, Z., LIN, S., STANKOVIC, J., 2006, ALARMNET: wireless sensor networks for assisted-living and residential monitoring, *Technical report CS-2006-11*, Department of Computer Science, University of Virginia.
26. <http://www.chipcon.com>, [Ziyaret Tarihi: 7 Temmuz 2012].
27. <http://www.soflinx.com>, [Ziyaret Tarihi: 7 Temmuz 2012].
28. SENSICAST SYSTEMS, <http://www.sensicast.com>, [Ziyaret Tarihi: 7 Temmuz 2012].
29. SYS TECHNOLOGIES, <http://www.systechnologies.com>, [Ziyaret Tarihi: 7 Temmuz 2012].
30. POTTIE, G. J., KAISER W. J., 2000, Wireless integrated network sensors, *Communications of the ACM*, 43:51–58.
31. PRIYANTHA, N., CHAKRABORTY, A., BALAKRISHNAN, H., 2000, The cricket location-support system, *In Proceedings of ACM MobiCom '00*, Boston, MA, USA, pp. 32–43.
32. Cougar Project, [www.cs.cornell.edu/database/cougar/](http://www.cs.cornell.edu/database/cougar/), [Ziyaret Tarihi: 16 Aralık 2012].
33. MADDEN, S.R., FRANKLIN, M.J., HELLERSTEIN, J.M., 2005, TinyDB: An Acquisitional Query Processing System for Sensor Networks, *ACM Trans. Database Systems*, vol. 30, no. 1, pp. 122-173.
34. TinyDB: A Declarative Database for Sensor Networks, <http://telegraph.cs.berkeley.edu/tinydb/>, [Ziyaret Tarihi: 16 Aralık 2012].
35. SHEN, C.C., SRISATHAPORNPHAT, C., JAIKAEAO, C., 2001, Sensor information networking architecture and applications, *IEEE Personal Communications*, 8(4):52–59.



36. SOUTO, E., GUIMARAES, G., VASCONCELOS, G., VIEIRA, M., ROSA, N., FERRAZ, C., KELNER, J., 2006, Mires: a publish/subscribe middleware for sensor networks, *Personal and Ubiquitous Computing*, 10(1):37–44.
37. ABERER, K., HAUSWIRTH, M., SALEHI, A., 2006, The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks, *Swiss Federal Institute of Technology, Lausanne (EPFL)*, Tech. Rep.
38. KRISHNAMURTHY, S., 2006, TinySIP: Providing Seamless Access to Sensorbased Services, *1st International Workshop on Advances in Sensor Networks*.
39. ROSENBERG, J. SCHULZRINNE, H. CAMARILLO, G. JOHNSTON, A. PETERSON, J. SPARKS, R. HANDLEY, M., SCHOOLER, E., 2002, SIP: Session Initiation Protocol, *The Internet Society*.
40. ROONEY, S., GARCES-ERICE, L., 2007, Messo & Preso practical sensor network messaging protocols, *the Fourth European Conference on Universal Multiservice Networks (ECUMN'07)* , pp. 364–376.
41. HUNKELER, U., TRUONG, H.L., Stanford Clark, A., 2008, MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks, *3rd International Conference on Communication Systems Software and Middleware and Workshops, COMSWARE*, pp 791 – 798.
42. FURRER, S., SCHOTT, W., TRUONG, H. L., WEISS, B., 2006, The IBM wireless sensor networking testbed, *2nd International Conference on Testbeds & Research Infrastructures for the DEvelopment of NeTworks & COMmunities (TRIDENTCOM'06)*.
43. MURPHY, A., HEINZELMAN, W., 2002, MiLAN: Middleware Linking Applications and Networks, TR-795, *University of Rochester, Computer Science*.
44. CURINO, C., GIANI, M., GIORGETTA, M., GIUSTI, A., MURPHY, A. L., PICCO, G. P., 2005, TinyLIME: Bridging mobile and sensor networks through middleware, *3rd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp 61–72.
45. PICCO, G. P., MURPHY, A. L., ROMAN, G.-C., 1999, LIME: Linda meets mobility, *Proc. of the 21st Int. Conf. on Software Engineering (ICSE)*, pp. 368–377.

46. LEVIS, P., CULLER, D., 2002, MAT<sup>E</sup>: A tiny virtual machine for sensor networks, *10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*.
47. BOULISA, A., HANB, C.C., SHEAB, R., SRIVASTAVA, M.B., 2007, SensorWare: Programming sensor networks beyond code update and querying, *Elsevier Science Publishers Pervasive and Mobile Computing Journal*, Volume 3 Issue 4.
48. MÜLLER, R., ALONSO, G., KOSSMANN, D., 2007, A Virtual Machine For Sensor Networks, *ACM SIGOPS Operating Systems Review - EuroSys'07 Conference Proceedings*, Volume 41 Issue 3.
49. KABADAYI S., JULIEN, C., 2006, DAIS: Enabling Declarative Applications in Immersive Sensor Networks, *Technical Report TR-UTEDGE-2006-000 Center for Excellence in Distributed Global Environments*, The University of Texas at Austin.
50. GROSSMAN, R., 2009, The case for cloud computing, *IT Professional*, 11(2), pp. 23–27.
51. ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A., 2009, Above the clouds: A Berkeley view of cloud computing, *University of California, Berkeley, Tech. Rep.*
52. SULTAN, N.A., 2011, Reaching for the “cloud”: How SMEs can manage, *International Journal of Information Management*, Volume 31, Issue 3, Pages 272–278.
53. JAMIL, E., 2008, What really is SOA. A comparison with Cloud Computing, Web 2.0, SaaS, WOA, Web Services, PaaS and others, *White Paper, soalib.com*
54. SAHOO, J., MOHAPATRA, S., LATH, R., 2010, Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues, *Second International Conference on Computer and Network Technology (ICCNT)*, pp(s): 222 – 226.
55. Windows Azure, [www.windowsazure.com](http://www.windowsazure.com/), [Ziyaret Tarihi: 08 Ocak 2013].
56. Google Engine, [https://appengine.google.com](https://appengine.google.com/), [Ziyaret Tarihi: 08 Ocak 2013].
57. Amazon AWS, <http://aws.amazon.com/>, [Ziyaret Tarihi: 08 Ocak 2013].
58. Application Harbor, <https://appharbor.com/>, [Ziyaret Tarihi: 08 Ocak 2013].

59. TIBCO Rendezvous, [www.tibco.com](http://www.tibco.com) , [Ziyaret Tarihi: 08 Ocak 2013].
60. CARZANIGA, A., ROSENBLUM, D.S., WOLF, A.L., 2001, Design and evaluation of a wide-area event notification service, *ACM Trans. on Computer Systems*, 19(3):332–383.
61. JERZAK, Z., 2009, *XSiena: The Content-Based Publish/Subscribe System*, Thesis (PhD), Technische Universität Dresden, Germany.
62. JERZAK, Z., FETZER, C., WOJCICKI, R., 2009, Soft State in the XSiena Publish/Subscribe System, *DEBS. ACM*.
63. FABRET, F., JACOBSEN, H.A., LLIRBAT, F., PEREIRA, J., ROSS, K.A., SHASHA, D., 2001, Filtering algorithms and implementation for very fast publish/subscribe systems, *ACM SIGMOD Record*, 30(2):115–126.
64. CUGOLA, G., NITTO, E.D., FUGGETTA, A., 2001, The JEDI event-based infrastructure and its application to the development of the OPSS WFMS, *IEEE Trans. on Software Engineering*, 27(9):827–850.
65. PIETZUCH, P.R., BACON, J. M., 2002, Hermes: A distributed event-based middleware architecture. *In Proc. of the 1st Int. Workshop on Distributed Event-Based Systems (DEBS) IEEE Computer Society Press*, Vienna, Austria.
66. PIETZUCH, P.R., BACON, J.M., 2003, Peer-to-peer overlay broker networks in an event-based middleware, *In Proc. of the 2nd Int. Workshop on Distributed Event-Based Systems (DEBS) ACM Press*, San Diego, CA, USA.
67. STROM, R., BANAVAR, G., CHANDRA, T., KAPLAN, M., MILLER, K., MUKHERJEE, B., STURMAN, D., WARD, M., 1998, Gryphon: An information flow based approach to message brokering, *In Int. Symp. on Software Reliability Engineering IEEE Computer Society Press*, Paderborn, Germany.
68. CHAND, R., FELBER., P.A., 2004, XNet: a reliable content based publish subscribe system, *In Proc. of the 23rd Symp. on Reliable Distributed Systems, Florianópolis IEEE Computer Society Press*, Brazil.
69. FIEGE, L., MÜHL, G., GARTNER, F.C., 2002, Modular event-based systems, *Knowledge Engineering Review*, 17(4):359–388.

70. SEGALL, B., ARNOLD, D., BOOT, J., HENDERSON, M., PHELPS, T., 2000, Content Based Routing with Elvin4, *In Proc. of AUUG2K*, Canberra, Australia.
71. JACOBSEN, H.A., 2005, ToPSS - The Toronto Publish/Subscribe System Family, *Proceedings of the 35th Annual Meeting* (Volume 1), volume 67 of Lecture Notes in Informatics, pp 424–428.
72. LIU, H., JACOBSEN, H.A., 2002, A-ToPSS - A Publish/ Subscribe System Supporting Approximate Matching, *In VLDB Morgan Kaufmann*, pp 1107–1110..
73. LIU H., JACOBSEN, H.A., 2004, A-ToPSS: A Publish/ Subscribe System Supporting Imperfect Information Processing, *In VLDB Morgan Kaufmann*, pp 1281–1284.
74. PETROVIC, M., BURCEA, I., JACOBSEN, H.A., 2003, S-ToPSS: Semantic Toronto Publish/Subscribe System. *In VLDB*, pp 1101–1104.
75. BURCEA, I., JACOBSEN, H.A., 2003, L-ToPSS - Push-Oriented Location-Based Services, *TES, volume 2819 of Lecture Notes in Computer Science, Springer*, pp 131–142.
76. FIDLER, E., JACOBSEN, H.A., LI, G., MANKOVSKI, S., 2005, The Padres Distributed Publish/Subscribe System, *FIW IOS Press*, pp 12–30.
77. Sun Spot, <http://www.sunspotworld.com/>, [Ziyaret Tarihi: 03 Ocak 2013].
78. SIMON, D., CIFUENTES, C., CLEAL, D., DANIELS, J., WHITE D., 2006, Sensor Devices—The Squawk virtual machine, *In VEE'06 ACM Press*, Ottawa, Ontario, Canada, 1-59593-332-6/06/0006.
79. SQL Azure Tanıtım Sayfası, <http://www.windowsazure.com/en-us/home/features/data-management/>, [Ziyaret Tarihi: 10 Ocak 2013].
80. SQL Server Tanıtım Sayfası, <http://www.microsoft.com/sqlserver/en/us/default.aspx>, [Ziyaret Tarihi: 10 Ocak 2013].
81. AGRAWAL, D., ABBADI, A.E., DAS, S., ELMORE, A.J., 2011, Database scalability, elasticity, and autonomy in the cloud, *DASFAA'11 Proceedings of the 16th international conference on Database systems for advanced applications*, Volume Part I, pp 2-15

82. AKYILDIZ, I.F., KASIMOGLU, I.H., 2004, Wireless sensor and actor networks: research challenges, *Ad Hoc Networks*, 2 (4) , pp. 351–367
83. BAI, X., KUMAR, S., XUAN, D., YUN, Z., LAI, T. H., 2006, Deploying wireless sensors to achieve both coverage and connectivity, *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 131–142.
84. MARENGONI, M., DRAPER, B. A., HANSON, A., SITARAMAN, R. A., 2000, A system to place observers on a polyhedral terrain in polynomial time, *IMAVIS: Image and Vision Computing*, p18.
85. O'ROURKE, J., 1987, *Art Gallery Theorems and Algorithms*, Oxford University Press.
86. WANG, Y.H., HUANG, K.F., LIN, S.T., 2011, A Grid-based Hole Detection Scheme in WSNs, *14th International Conference on Network-Based Information Systems (NBIS)*, pp(s): 202-207.
87. KANNO, J., BUCHART, J.G., SELMIC, R.R., PHOHA, V., 2009, Detecting Coverage Holes in Wireless Sensor Networks, *17th Mediterranean Conference on Control & Automation*, Makedonia Palace, Thessaloniki, Greece.
88. WAN, P., ALZOUBI, K. ,FRIEDER, O., 2001, Distributed construction of connected dominating set in wireless ad hoc networks, *In Proceedings of the IEEE INFOCOM*, volume 3.
89. KHALED, M., ALZOUBI, WAN, P.J., FRIEDER, O., 2002, Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks, *MOBIHOC'02*, June 9-11, EPFL Lausanne, Switzerland.
90. TIAN, D., GEORGANAS, N.D., 2002, A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks, *Proc. 1st ACM Workshop Wireless Sensor Networks and Applications*, ACM Press, pp. 32-41
91. ZHOU, Z., DAS, S., GUPTA, H., 2005, Fault tolerant connected sensor cover with variable sensing and transmission ranges. *In SECON '05*, IL, USA.
92. ZORBAS, D., GLYNOS, D., KOTZANIKOLAOU, P., DOULIGERIS, C., 2010, Solving Coverage Problems in Wireless Sensor Networks Using Cover Sets, *Ad Hoc Networks*, 8 no. 4, 400–15.

93. YE, F., ZHONG, G., CHENG, J., LU, S., ZHANG, L., 2003, PEAS: a robust energy conserving protocol for long-lived sensor networks, *23th IEEE International Conference on Distributed Computing Systems, (ICDCS '03)*, pp. 28–37.
94. CARLE, J., SIMPLOT-RYL D., 2004, Energy-Efficient Area Monitoring for Sensor Networks, *IEEE Computer Journal*, Volume: 37 , Issue: 2 pp(s): 40 - 46
95. YILDIRIM, K. S., KALAYCI, T. E., UGUR, A., 2008, Optimizing Coverage in a K-Covered and Connected Sensor Network Using Genetic Algorithms, *9th WSEAS International Conference on Evolutionary Computing (EC'08)*, Stevens Point, Wisconsin, USA, 21-26.
96. KALAYCI, T. E., UĞUR, A., 2011, Genetic Algorithm-Based Sensor Deployment with Area Priority, *Cybernetics and Systems: An International Journal*, 42(8), 605-620.
97. HUSSAIN, S., ISLAM, O., 2008, *Genetic algorithm for energy efficient trees in wireless sensor networks” Advanced intelligent environments*, Chapter 1, Springer
98. AKBULUT, A., AYDIN, M.A., ATMACA, T., ZAIM, A.H., 2012, Accessing to Data in WSNs, *International Conference on Networking and Future Internet ICNFI*, pp: 70 -73, Istanbul – Turkey.
99. Windows Azure: Microsoft's Cloud Platform, [www.windowsazure.com](http://www.windowsazure.com), [Ziyaret Tarihi: 7 Ocak 2013].
100. ENDLER, J. A., 1986, *Natural Selection in the Wild*, Princeton: Princeton University Press.
101. PITTEWAY, M.L.V., 1967, Algorithm for Drawing Ellipses or Hyperbolae with a Digital Plotter, *Computer J.*, 10(3), pp 282-289
102. AKEN, J.R.V., 1984, An Efficient Ellipse Drawing Algorithm, *CG&A*, 4(9), pp 24-35
103. ROUSSEUW, P. J., VAN ZOMEREN, B. C., 1990, Unmasking Multivariate Outliers and Leverage Points, *Journal of the American Statistical Association*, 85, 633 – 639.

104. VALDES, A., SKINNER, K., 2000, Adaptive, modelbased monitoring for cyber attack detection, *3rd International Workshop on Recent Advances in Intrusion Detection*. Springer-Verlag, 80 - 92.
105. ZHANG, Y., MERATNIA, N., HAVINGA, P., 2010, Outlier Detection Techniques For Wireless Sensor Networks: A Survey, *Journal IEEE Communications Surveys & Tutorials*, pp 159-170, Volume 12 Issue 2.
106. CHEN, H, CHANG, K. C., 2008, K-Nearest Neighbor Particle Filters for Dynamic Hybrid Bayesian Networks, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, No. 3.
107. ELAHI, M., LI, K., NISAR, W., LV, X., WANG, H., 2008, Efficient Clustering-Based Outlier Detection Algorithm for Dynamic Data Stream, *Fifth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD '08*, Volume: 5, pp(s): 298-304.
108. ELNAHRAWY, E., NATH, B., 2004, Context-Aware Sensors, *The First European Conference on Wireless Sensor Networks (EWSN 2004)*, Volume 2920 of Lecture Notes in Computer Science (LNCS), pp 77-93, Springer-Verlag Berlin Heidelberg.
109. JANAKIRAM, D., REDDY, V.A.M., KUMAR, A.P., 2006, Outlier detection in wireless sensor networks using Bayesian belief networks, *First International Conference on Communication System Software and Middleware*, pp: 1-6
110. HILL, D. J., MINSKER, B. S., AMIR, E., 2007, Real-time Bayesian anomaly detection for environmental sensor data, *32nd Congress of the International Association of Hydraulic Engineering and Research*.
111. JINLIN, Z., ZHENGDAO, Z., 2012, Fault Prognosis for Data Incomplete Systems: A Dynamic Bayesian Network Approach, *IEEE 24th Control and Decision Conference (CCDC)*, Chinese.
112. XU, J., SHELTON, C.R., 2010, Intrusion Detection using Continuous Time Bayesian Networks, *Journal of Artificial Intelligence Research* 39 745–774.
113. CHIN JR.G., CHOUDHURY, S., KANGAS, L., MCFARLANE,S., MARQUEZ, A., 2010, Fault Detection in Distributed Climate Sensor Networks

using Dynamic Bayesian Networks, *Sixth IEEE International Conference on e-Science*.

114. Coastal Ocean Observation Lab, <http://rucool.marine.rutgers.edu/>, [Ziyaret Tarihi: 15 Nisan 2012].
115. Google Maps, <https://maps.google.com/>, [Ziyaret Tarihi: 15 Ocak 2013].
116. PATLAR, F., AKBULUT, A., ZAIM, A.H., Cloud Computing: The Evolution of Service Architectures, *Advanced Science Letters Journal* ,ISSN: 1936-6612, (in publishing)



## ÖZGEÇMİŞ

Akhan Akbulut 1979 yılında, avukat baba ve devlet memuru annenin oğlu olarak İstanbul'da doğdu. İlköğretimini Kültür İlkokulu'nda tamamlamasının ardından orta ve lise öğrenimi için Vefa Anadolu Lisesi'ne başladı. Lise yıllarında çok ilgi duyduğu bilgisayar ve bilişim sistemlerinde kendini geliştirebilmek amacıyla tercih ettiği İstanbul Kültür Üniversitesi, Bilgisayar Mühendisliği bölümünden 2001 yılında mezun olarak bilişim sektörüne katıldı. 2005 Yılında başladığı İstanbul Kültür Üniversitesi, Bilgisayar Mühendisliği araştırma görevliliği pozisyonundan önce pek çok orta ve büyük ölçekli sektörel projede; uzman yazılım geliştirici, analist, takım lideri ve proje yöneticisi sıfatıyla görevler aldı. 2008 yılında yine İstanbul Kültür Üniversitesi, Bilgisayar Mühendisliği, Yüksek Lisans programından "*An Architectural Model for Content Management in e-Commerce Applications Using Intelligent Agents*" başlıklı tezini vererek mezun oldu. 2009 Yılında İstanbul Üniversitesi, Bilgisayar Mühendisliği Bölümü'nde doktora eğitimine başladı. Kendisi gibi doktora eğitimini sürdüren Fatma Patlar Akbulut ile evli olan Akhan Akbulut'un araştırmalarından 2 SCI indeksinde taranan uluslararası dergi makalesi ve 13 uluslararası konferans bildirisi bulunmaktadır.