

# IoT Vize

## 1- kablosuz ve mobil ağların neden önemli ve temel kavramlar

kablosuz (mobil) telefon abone sayısı, kablolu telefon abone sayısını geçti

Kablosuz internete bağlı cihazların sayısı, kablolu internete bağlı cihazların sayısına eşit hale geldi. Yani, telefonlar, tabletler gibi kablosuz cihazlar artık masaüstü bilgisayarlar kadar yaygın.

her yerden internete erişim imkanı

### #Mobil (Mobile):

Mobil terimi, hareket halindeyken bile ağ bağlantısını sürdürebilen cihazları ve teknolojileri ifade eder. Genellikle **kablosuz (wireless) ve hücreli (cellular) ağlar** üzerinden bağlantı kurarlar.

### Non-Mobil (Non-Mobile):

Non-mobil terimi, sabit bir konumda bulunan ve genellikle **kablolu (wired) veya yerel kablosuz ağ (Wi-Fi) üzerinden bağlanan** cihazları ifade eder. Bu cihazlar hareket etmez veya mobil şebekeleri kullanmaz.

### #Base Station (Baz İstasyonu)

- **Ne:** Genellikle kablolu ağa bağlı olan bir aracı. wireless link sağlar. bu da -> **Multiple Access Protocol:** Birden fazla cihazın aynı bağlantıyı kullanmasını sağlar (örneğin Wi-Fi'de herkes aynı ağa bağlanır).
- **Görev:** Kablolu ağ ile kablosuz cihazlar arasında veri paketlerini iletmek.
- **Örnekler:**
  - Hücreli ağlardaki baz istasyonları (cep telefonu kuleleri).
  - Wi-Fi erişim noktaları (access points).

### #Infrastructure Mode (Altyapı Modu)

- **Ne:** Baz istasyonunun mobil cihazları kablolu ağa bağladığı bir sistem.
- **Nasıl Çalışır:**
  - Baz istasyonu, mobil cihazları (örneğin telefon, laptop) kablolu bir ağa (genellikle internete) bağlar.
  - **Handoff (El Değiştirme):** Mobil bir cihaz, hareket ettiğinde bağlı olduğu baz istasyonunu değiştirir. Örneğin, yolda giderken telefonun bir baz istasyonundan diğerine geçmesi.

### #Ad Hoc Mode (Ad Hoc Modu)

- **Ne:** Baz istasyonu olmadan çalışan bir sistem.
- **Nasıl Çalışır:**
  - Cihazlar (düğümler) doğrudan birbirine bağlanır.
  - Sadece bağlantı mesafesi içindeki cihazlarla iletişim kurabilirler.
  - Cihazlar kendi kendilerine bir ağ oluşturur ve birbirleri arasında veri yönlendirmesi (routing) yapar.

## 2- kablosuz ve mobil ağlarla ilgili zorluklar neler?

Kablosuz ve mobil ağlarla ilgili iki büyük zorluk var:

1. **Wireless:** Kablosuz bağlantı üzerinden iletişim kurmak. kablolardan kurtulmamızı sağlıyor ama aynı zamanda teknik zorluklar getiriyor.
2. **Mobility:** Mobil bir kullanıcının ağa bağlanma noktasını değiştirdiğinde (örneğin bir yerden başka bir yere hareket ettiğinde) bu durumu yönetmek.
3. altyapı ve taşınabilirlik sağlama
4. hız ve güvelik

## 3- Hand off ve hand over nedir? kaç türe ayrılır?

"Handoff" (veya bazen "handover" olarak da geçer), cihazın bir ağın kapsamından başka bir ağın kapsamına geçmesi.

KAPSAMA ALANI: sinyalin alınabildiği yer

### #Handoff (El Değiştirme) Nedir?

Handoff, bir mobil cihazın (telefon, IoT sensörü, vb.) hareket halindeyken bir baz istasyonundan (veya erişim noktasından) diğerine geçiş yapması ve bağlantısını kesintisiz sürdürmesidir.

- **Amaç:** Kullanıcının veya cihazın iletişimini koparmadan ağı bağlı kalmasını sağlamak.
- **Örnek:** Arabayla giderken telefonun bir baz istasyonundan diğerine geçmesi ve aramanın kesilmemesi.

Kablosuz ağlarda her baz istasyonunun bir kapsama alanı (coverage area) vardır. Cihaz bu alanın dışına çıktığında, bağlantıyı sürdürebilmek için başka bir baz istasyonuna "el değiştirmesi" gerekir. IoT'da mesela bir drone veya araçtaki sensör hareket ederken bu olay sıkça yaşanır.

- 1- **Sinyal Gücü Azalır:** Cihaz, mevcut baz istasyonundan uzaklaştıkça sinyal zayıflar.
- 2- **Yeni Baz İstasyonu Tespit Edilir:** Cihaz veya ağ, yakındaki daha güçlü bir baz istasyonunu algılar.
- 3- **Bağlantı Aktarılır:** Cihazın trafiği (veri, ses, vb.) eski baz istasyonundan yenisine aktarılır.
- 4- **Kesintisiz Devam:** Kullanıcı (veya IoT cihazı) bu geçişi genellikle fark etmez bile.

### HAND OFF TÜRLERİ

1. **Hard Handoff (Sert El Değiştirme):** iot'de nadiren ağ değiştiren cihazlarda kullanılabilir

- **Ne:** Cihaz, eski baz istasyonu ile bağlantısını tamamen keser ve ancak ondan sonra yeni baz istasyonuna bağlanır.
- **Özellik:** "Break before make" (önce kop, sonra bağlan) prensibiyle çalışır.
- **Avantaj:** Daha basit bir sistem, daha az kaynak gerektirir.
- **Dezavantaj:** Kısa bir kesinti olabilir (örneğin sesli aramada bir anlık boşluk).
- **Örnek:** Eski GSM ağlarında sıkça kullanılırdı.

#### 2. Soft Handoff (Yumuşak El Değiştirme)

- **Ne:** Cihaz, yeni baz istasyonuna bağlanmadan önce eski bağlantısını korur; bir süre iki baz istasyonuna aynı anda bağlıdır.
- **Özellik:** "Make before break" (önce bağlan, sonra kop) prensibiyle çalışır.
- **Avantaj:** Kesintisiz geçiş sağlar, veri kaybı veya kesinti olmaz.
- **Dezavantaj:** Daha fazla ağ kaynağı gerektirir (çünkü iki bağlantı aynı anda aktif).
- **Örnek:** CDMA tabanlı 3G ağlarında kullanılır.

#### 3. Horizontal Handoff (Yatay El Değiştirme)

- **Ne:** Aynı tür ağlar arasında geçiş. Yani, aynı teknolojiyi kullanan iki baz istasyonu arasında handoff yapılır.
- **Örnek:** Wi-Fi ağında bir erişim noktasından diğerine geçmek veya 4G'de bir baz istasyonundan diğerine geçiş.
- **Açıklama:** Teknoloji değişmediği için daha basit bir süreçtir.

#### 4. Vertical Handoff (Dikey El Değiştirme)

- **Ne:** Farklı tür ağlar arasında geçiş. Örneğin, Wi-Fi'den hücresel ağı (4G) geçmek.
- **Örnek:** Evde Wi-Fi'den çıkıp dışarıda 4G'ye geçen bir telefon.
- **Zorluk:** Farklı protokoller ve teknolojiler arasında uyum sağlamak gerekir.
- **Avantaj:** Esneklik sağlar, cihaz her zaman en uygun ağı kullanabilir.

handoff iot'de

- Bağlantılarını sürdürmesini,
- Veri kaybı yaşamamasını,
- Enerji verimli çalışmasını sağlar.

### 4- mobil ve kablosuz haberleşmenin zorlukları nelerdir?

**Hidden Terminal Problemi**, kablosuz ağlarda bazı cihazların birbirlerini doğrudan algılayamaması nedeniyle oluşan veri çakışması problemidir. Örneğin, A ve C cihazları aynı erişim noktasına (B) bağlı olabilir, ancak birbirlerini doğrudan "göremedikleri" için aynı anda veri gönderdiklerinde çakışma yaşanır ve veriler bozulur. Bu

sorunu önlemek için **RTS/CTS (Request to Send / Clear to Send) protokolü** ve **CSMA/CA (Collision Avoidance) mekanizması** gibi yöntemler kullanılır.

**Signal Attenuation (Sinyal Zayıflaması)**, kablosuz sinyallerin mesafe arttıkça veya fiziksel engeller nedeniyle güç kaybetmesi durumudur. Duvarlar, atmosfer koşulları ve yüksek frekanslı sinyallerin doğası gereği zayıflama etkisi artabilir. Bunu önlemek için **güçlü antenler, düşük frekans bantları (örneğin 2.4 GHz Wi-Fi) ve erişim noktalarının stratejik yerleştirilmesi** gibi çözümler uygulanabilir.

### 1. Gürültü (Noise)

Kablosuz haberleşmede **gürültü**, sinyalin taşınması sırasında istem dışı eklenen parazit veya istenmeyen sinyallerdir.

- ◆ **Kaynakları:** Elektronik cihazlar, güneş radyasyonu, atmosferik olaylar.
- ◆ **Etkisi:** Sinyal gücünü düşürerek veri kaybına ve hatalı iletme neden olur.

◆ **Örnek:** Wi-Fi sinyalinin çevredeki mikrodalga fırın, Bluetooth veya diğer elektronik cihazlar nedeniyle bozulması.

### 2. Sinyalin Silinmesi (Signal Attenuation)

Sinyalin yayılırken mesafe arttıkça gücünün azalmasıdır. **Özellikle yüksek frekanslı sinyaller daha hızlı zayıflar.**

- ◆ **Kaynakları:** Uzun mesafeler, binalar, atmosferdeki gaz molekülleri.
- ◆ **Etkisi:** Kapsama alanı dışında sinyalin alınamamasına neden olur.

◆ **Örnek:** Cep telefonu ile baz istasyonundan uzaklaştıkça sinyal seviyesinin azalması.

### 3. Fading (Sinyal Kaybı)

Fading, **sinyalin zamanla, frekansla veya uzamsal olarak değişkenlik göstermesi** anlamına gelir.

- ◆ **Kaynakları:** Çevresel faktörler, hareketlilik, engeller.
- ◆ **Etkisi:** Veri kaybı, kesinti veya düşük sinyal gücü.

◆ **Örnek:** Bir araba ile hareket halindeyken telefonun çekim gücünün sürekli değişmesi.

### 4. Çok Yollu Yayılım (Multipath Propagation)

Sinyalin farklı yollar üzerinden alıcıya ulaşmasıdır. Bu, genellikle **binalar, duvarlar ve engellerden dolayı sinyalin yansıması, kırılması veya saçılmasıyla oluşur.**

- ◆ **Kaynakları:** Binalar, dağlar, su yüzeyleri, araçlar.
- ◆ **Etkisi:** Gecikme, girişim (interference) ve sinyal bozulması.

◆ **Örnek:** Kapalı alanlarda Wi-Fi sinyalinin duvarlardan sekerek yön değiştirmesi ve bazen "ölü noktalar" oluşması.

### 5. Çevresel Engeller (Environmental Obstacles)

Sinyalin önüne çıkan fiziksel engeller, sinyalin gücünü azaltabilir veya yönünü değiştirebilir.

- ◆ **Kaynakları:** Binalar, ağaçlar, dağlar, kötü hava koşulları (yağmur, sis).
- ◆ **Etkisi:** Sinyalin zayıflaması veya tamamen kesilmesi.

◆ **Örnek:** Dağlık bir bölgede cep telefonu sinyalinin kesilmesi.

### 6. Girişim (Interference)

Başka sinyallerin, mevcut sinyalin iletimine zarar vermesi durumudur.

- ◆ **Kaynakları:** Diğer kablosuz cihazlar, baz istasyonları, radyo dalgaları, elektrik hatları.
- ◆ **Etkisi:** Veri kaybı, sinyal kalitesinde düşüş, bağlantı kopmaları.

◆ **Örnek:** Kalabalık bir kafede herkesin aynı Wi-Fi kanalını kullanması sonucu internet hızının düşmesi.

### 7. Aynı Kanal Girişim (Co-Channel Interference - CCI)

Aynı frekansta çalışan birden fazla cihazın veya baz istasyonunun birbirine parazit yapmasıdır.

♦ **Kaynakları:** Aynı frekansta çalışan baz istasyonları, Wi-Fi router'lar.

♦ **Etkisi:** Bağlantı hızının düşmesi, paket kaybı.

✦ **Örnek:** Bir apartmanda çok fazla Wi-Fi modem olunca sinyallerin çakışması.

## 8. Doppler Etkisi (Doppler Effect)

Hareket halindeki bir verici veya alıcı nedeniyle sinyal frekansının değişmesidir.

♦ **Kaynakları:** Hızlı hareket eden araçlar (tren, uçak, araba).

♦ **Etkisi:** Frekans kayması, veri kaybı, ses ve görüntü bozulması.

✦ **Örnek:** Hızlı giden bir arabada FM radyosunun netliğinin sürekli değişmesi.

## 5- IoT nedir? önemi nedir? ekosistemi için ne gerekli?

HER ŞEYİN HER YERDE HER ZAMAN ERİŞİMDE OLACAĞI BİR YAPIDIR. nesnelerin içerik bilgilerini paylaştırılması ve birbirine dönüştürülmesi konseptine dayanır.

"Internet of Things" İsmi: Bu terim iki kelimeden oluşuyor:

1. **Internet:** Ağa yönelik bir vizyonu öne çıkarıyor (network-oriented).
2. **Things:** Genel nesneleri ve bunların bir çerçeveye entegre edilmesini vurguluyor (object-oriented).

**Tanım:** IoT, modern kablosuz telekomünikasyon dünyasında hızla yayılan yeni bir paradigma (yaklaşım). Temel fikir, çevremizde bulunan çeşitli nesnelerin (RFID etiketleri, sensörler, aktüatörler, cep telefonları gibi) birbiriyle etkileşime girebilmesi ve ortak hedefler için iş birliği yapabilmesi.

- **Nasıl Çalışır:** Bu nesneler, benzersiz adresleme sistemleri (unique addressing schemes) sayesinde birbirine bağlanır.
- **Örnek:** Bir buzdolabının içindeki yiyeceklerin azaldığını sensörle tespit edip sana mesaj göndermesi veya bir arabanın trafik durumuna göre rotasını değiştirmesi.
- **Akıllı ev sistemleri** (Google Nest, Amazon Alexa, akıllı prizler)
- **Akıllı sağlık cihazları** (giyilebilir sağlık izleyicileri)
- **Akıllı fabrikalar** (Endüstri 4.0, otomatik üretim sistemleri)
- **Akıllı şehir uygulamaları** (trafik yönetimi, çevre sensörleri)

GEREKİLİKLER:

## 1 Cihazlar ve Sensörler

- IoT'nin temel bileşeni, çevresel verileri ölçen **sensörlerdir**.
- **Örnek:** Isı sensörleri, hareket dedektörleri, nabız ölçerler.

## 2 Ağ ve Bağlantı

- IoT cihazları, **Wi-Fi, Bluetooth, 4G/5G, LPWAN, LoRaWAN gibi ağ protokolleri** ile internete bağlanır.
- **Örnek:** Akıllı saatlerin telefona Bluetooth ile bağlanması veya trafik kameralarının 5G ile veri göndermesi.

## 3 Veri İşleme ve Bulut Bilişim

- IoT cihazlarından gelen büyük miktarda veri **bulut platformlarında veya edge computing (uç bilişim) sistemlerinde** işlenir.
- **Örnek:** Bir fabrikada IoT sensörlerinin ürettiği verilerin anlık analiz edilmesi.

## 4 Yapay Zeka (AI) ve Veri Analitiği

- Toplanan büyük verinin anlamlı hale getirilmesi için **AI ve ML (Makine Öğrenimi) algoritmaları** kullanılır.
- **Örnek:** Akıllı termostatların kullanıcının alışkanlıklarını öğrenerek otomatik sıcaklık ayarı yapması.

## 5 Güvenlik ve Veri Gizliliği

- IoT cihazları **siber saldırılara** açık olabilir.
- **Şifreleme, kimlik doğrulama ve güvenli ağ protokolleri** kullanılmalıdır.
- **Örnek:** Akıllı kameraların hackerlar tarafından ele geçirilmesini önlemek için veri şifreleme.

### 6- IoT zorlukları nelerdir ve neler yapmalıyız?

MAKALE:

- **Teknolojik Sorunlar:**
  - Cihazların tam birbiriyle uyumlu (interoperable) çalışması.
  - Daha akıllı hale gelmeleri (adaptation, autonomous behavior).
  - Güven, gizlilik ve güvenlik sağlanması.
- **Ağ Sorunları:**
  - IoT cihazları genelde düşük işlem ve enerji kapasitesine sahip.
  - Çözümlerin kaynak verimliliği ve ölçeklenebilirlik (scalability) sağlaması gerek.
  - **Ana Zorluklar**
  - **Benzersiz Adresleme (Unique Addressing):** Her nesneye eşsiz bir kimlik vermek.
  - **Bilgi Temsili ve Depolama:** Nesneler arasında değiş tokuş edilen verilerin nasıl saklanacağı ve işleneceği.
  - **Üçüncü Vizyon:** Bu zorluklar, IoT'a "Semantic-Oriented" (Anlamsal Yönelimli) bir bakış açısını getiriyor.

Hocanın yazdırdıkları:

- 1- standartizasyona ihtiyaç var. nesnelerin taşınması ve kontrolü kapsamında.
- 2- hareketlilik desteğine ihtiyaç var, hand off başarılı yönetilmeli
- 3- nesnelerin haberleşirken IP yerine NAMING olarak haberleşmesi, isim bilgisinin eklenmesi
- 4- TCP UDP protokolleri transportta elverişsiz.
- 5- tıkanıklık ve hatalı okuma gibi problemler
- 6- big data yönetmesi zor

CHAT GPT:

## IoT'nin Karşılaştığı Zorluklar

- ♦ **1 Güvenlik ve Gizlilik Sorunları** – Veri sızıntısı ve siber saldırılara karşı savunmasızlık.
- ♦ **2 Veri Yönetimi ve Büyük Veri** – IoT cihazlarının ürettiği büyük veri yükünün işlenmesi.
- ♦ **3 Enerji ve Pil Ömrü** – Kablosuz cihazların düşük güç tüketimi ihtiyacı.
- ♦ **4 Standartların Eksikliği ve Uyumluluk** – Farklı üreticilerin cihazları arasında uyumsuzluk.
- ♦ **5 Ağ Kesintileri ve Güvenilirlik Sorunları** – Sürekli bağlantı ihtiyacı ve ağ kesintileri riski.

### 7- mobil IP ve tünelleme desteği nedir? hareketlilik desteği ve agent meselesini anlatın.

**Mobil IP (Mobile IP)**, bir cihazın farklı ağlara bağlandığında **IP adresini değiştirmeden** iletişimi sürdürebilmesini sağlayan bir protokoldür. Normalde, bir cihaz farklı bir ağa geçtiğinde IP adresi değişir ve bağlantılar kesilir. Ancak **Mobil IP**, kullanıcının IP adresini koruyarak kesintisiz bağlantı sunar.

#### 🔴 Nasıl Çalışır?

Mobil IP, cihazın sürekli erişilebilir olmasını sağlamak için **tünelleme (tunneling)** kullanır. Tünelleme, cihazın gerçek IP adresini korumak için paketleri özel kapsülleme (encapsulation) yöntemleriyle yönlendirir.

Mobil IP, üç temel bileşenle çalışır:

1. **Home Agent (HA - Ana Ajan):** Cihazın kayıtlı olduğu orijinal ağdaki sunucudur. Gelen paketleri izler ve cihaz başka bir ağa geçtiğinde paketleri yönlendirir.
2. **Foreign Agent (FA - Yabancı Ajan):** Cihazın ziyaret ettiği yeni ağdaki sunucudur. Cihazın burada olduğunu Home Agent'a bildirir ve verileri alıp cihaza iletir.
3. **Mobile Node (MN - Mobil Düğüm):** Ağlar arasında hareket eden mobil cihazdır (örneğin bir akıllı telefon veya dizüstü bilgisayar).

### ✦ Örnek Senaryo:

- Bir telefon normalde **Home Agent'a bağlıdır**.
- Farklı bir Wi-Fi ağına bağlandığında **Foreign Agent üzerinden iletişim kurar**.
- Home Agent, gelen paketleri Foreign Agent'a yönlendirir ve o da telefonu yeni IP ile ilişkilendirerek veriyi taşır.

### 8- IoT ve altyapı(infrastructure) imcelemesi

IoT, mevcut internet altyapısını (örneğin Wi-Fi, hücresel ağlar) kullanır ama tam anlamıyla IoT'ye özel bir altyapı her yerde hazır değil. IoT'un altyapısı şunlara dayanır:

- **Ağ Altyapısı:** Baz istasyonları, router'lar, gateway'ler.
- **Bulut Sistemleri:** Verilerin depolanması ve işlenmesi için (örneğin AWS IoT, Google Cloud IoT).
- **Protokoller:** MQTT, CoAP gibi IoT'ye özgü hafif protokoller.

#### Altyapı Problemleri:

- **Kapsama Alanı:** Kırsal bölgelerde veya gelişmemiş yerlerde ağ altyapısı zayıf. Mesela bir IoT tarım sensörü internete bağlanamazsa işe yaramaz.
- **Ölçeklenebilirlik:** Milyarlarca cihaz bağlandığında mevcut altyapı (örneğin IPv4 adresleri) yetmeyebilir. IPv6 geçişi bir çözüm ama yavaş ilerliyor.
- **Enerji:** IoT cihazlarının çoğu pil ile çalışıyor, sürekli enerji sağlayacak altyapı eksik.
- **Güvenlik:** Mevcut ağlar IoT cihazlarının hack'lenmesine karşı yeterince korunaklı değil.
- **Maliyet:** Altyapıyı genişletmek (örneğin 5G kurmak) pahalı ve zaman alıyor.

IoT'da genellikle **mesh** ve **yıldız** topolojileri yaygın çünkü IoT cihazları hem yerel (local) hem de geniş alan (wide area) ağlarında çalışır. Mesela bir IoT sensörü evde yıldız topolojisiyle router'a bağlanırken, tarımda mesh topolojisiyle diğer sensörlerle iletişim kurabilir.

#### Problem:

- Topoloji seçimi, cihazların enerji tüketimi, mesafe ve ölçeklenebilirlik gibi faktörlere bağlı. Yanlış topoloji seçilirse ağ verimsiz olur.

#### IoT'da Bağlantı Problemi:

- **Heterojenlik:** Farklı cihazlar farklı protokoller kullanır (Wi-Fi, ZigBee, Bluetooth). Bunların birbiriyle uyumlu çalışması zor.
- **Mesafe:** Bazı cihazlar kısa mesafede çalışır (Bluetooth), bazıları uzun mesafede (LoRa). Doğru teknolojiyi seçmek kritik.
- **Bant Genişliği:** Çok fazla cihaz bağlanırsa ağ tıkanabilir.
- **Güvenlik:** Bağlantılar zayıfsa hack'lenme riski artar.

### 9- IoT senkronizasyon problemleri ve çözüm yöntemleri

#### IoT'daki Senkronizasyon Sorunları:

- **Zamanlama Uyumsuzluğu:**
  - Farklı cihazlar farklı hızlarda veri gönderirse (örneğin bir sensör saniyede 1 veri, diğeri dakikada 1 veri gönderiyorsa), sistem karışır.
  - **Örnek:** Bir trafik IoT sisteminde, bir sensör kaza bildirirken diğeri eski veri gönderirse yanlış karar alınır.
- **Enerji Kısıtları:**
  - Düşük güçlü IoT cihazları sürekli açık kalamaz, uyku moduna geçer. Bu, senkronizasyonu zorlaştırır.
  - **Örnek:** Bir tarım sensörü pili korumak için uyuyorsa, diğeri cihazlarla veri paylaşımı aksar.
- **Ağ Gecikmeleri (Latency):**
  - Veriler ağda gecikirse (özellikle geniş alan ağlarında), cihazlar arasında uyum bozulur.
  - **Örnek:** Bir IoT güvenlik kamerası anlık görüntü gönderemezse alarm gecikir.

## Çözüm Yaklaşımları:

- **Protokoller:** Zaman senkronizasyonu için NTP (Network Time Protocol) veya PTP (Precision Time Protocol) kullanılabilir.
- **Gateway'ler:** Ara cihazlar, farklı hızdaki verileri senkronize edebilir.
- **Düşük Güç Tasarımı:** Cihazlar uyku modunda bile belirli aralıklarla uyanıp senkronize olabilir.

## 10 - IoT de context, context sharing, service ve hareketlilik kavramları ne demektir?

IoT (Nesnelerin İnterneti) ortamında "context" kavramı, bir cihazın, kullanıcının veya çevrenin mevcut durumunu, özelliklerini ve davranışlarını tanımlayan bilgileri ifade eder. Bu bilgiler; konum, zaman, sıcaklık, hareket, kullanıcı tercihleri gibi verileri içerebilir. Böylece sistemler, mevcut bağlamı anlayıp, o anki duruma uygun kararlar alarak, daha akıllı ve kişiselleştirilmiş hizmetler sunabilirler.

"Context sharing" ise, bu bağlam bilgilerini farklı IoT cihazları veya sistemleri arasında paylaşma sürecidir. Farklı uygulamalar ve cihazlar, ortak bir anlayışa sahip olmak ve uyumlu çalışabilmek için kendi topladıkları bağlam verilerini diğer sistemlerle paylaşırlar. Bu, hem veri tekrarı ve işlem yükünü azaltır hem de heterojen cihazların daha uyumlu çalışmasını sağlar.

"Service" kavramı, IoT ortamında sensörlerden, cihazlardan veya bulut platformlarından elde edilen veriler doğrultusunda sunulan işlevsel hizmetleri veya uygulamaları ifade eder. Bu hizmetler; veri analizine dayalı öneriler, otomasyon çözümleri veya kullanıcıya özel bildirimler gibi pek çok farklı biçimde ortaya çıkabilir. Servisler, IoT ekosisteminin temel değer önerisini oluşturur çünkü fiziksel dünyanın dijitalleşmesini ve etkileşimli hale gelmesini sağlarlar.

"Hareketlilik" (mobility) kavramı ise IoT cihazlarının fiziksel olarak yer değiştirme yeteneğini ve bu durumlarda da kesintisiz bağlantı ve hizmet devamlılığını ifade eder. Mobil cihazlar, farklı ağlara bağlandıklarında IP adresleri veya bağlantı parametreleri değişebilir; ancak bu değişim sırasında hizmetlerin kesintiye uğramaması için hareketlilik yönetimi önem kazanır. Böylece, cihazlar hareket halinde bile çevresel bağlamlarına uygun ve sürekli hizmet sunabilirler.

- **Context:** IoT ortamında, bir varlığın (cihaz, kullanıcı, çevre) mevcut durumunu belirten, konum, zaman, sensör verileri gibi bilgilerin tümü.
- **Context Sharing:** Bu bağlam bilgilerini farklı cihazlar veya sistemler arasında paylaşarak, ortak ve uyumlu bir anlayış oluşturulmasını sağlayan süreç.
- **Service:** IoT cihazları ve platformlarından elde edilen verilerle oluşturulan, kullanıcıya veya sisteme değer katan işlevsel uygulamalar veya hizmetler.
- **Hareketlilik (Mobility):** IoT cihazlarının yer değiştirme yeteneği ve hareket halindeyken bile kesintisiz bağlantı ve hizmet sunabilme özelliği.

## 11- IoT içinde ne tür servisler olur? nasıl hizmetler gerçekleşir?

Servis, bir IoT cihazının "ne yaptığını" ve bu yaptığının kime, nasıl fayda sağladığını ifade eder. Örneğin, bir akıllı termostatin evi ısıtması bir servistir.

### 1. Veri Toplama Servisleri (Data Collection Services)

- **Tanım:** IoT cihazlarının çevreden veri toplamasını sağlayan servislerdir. Bu, IoT'un temel yapı taşıdır çünkü diğer servisler bu verilere dayanır.
- **Nasıl Çalışır:** Sensörler aracılığıyla fiziksel dünyadan bilgi alınır (sıcaklık, nem, hareket vb.).
- **Örnek:**
  - Bir hava kalitesi sensörünün CO2 seviyesini ölçmesi.
  - Bir tarım sensörünün toprak nemini izlemesi.
- **Kullanılan Teknoloji:** Sensörler, düşük güç protokolleri (örneğin LoRaWAN).
- **Amaç:** Ham veri sağlayarak diğer servislerin çalışmasını destekler.

### 2. Kontrol Servisleri (Control Services)

- **Tanım:** IoT cihazlarının fiziksel dünyayı etkilemesini sağlayan servislerdir. Aktüatörler (hareket ettiren cihazlar) kullanılır.
- **Nasıl Çalışır:** Kullanıcı komutları veya otomatik kararlarla cihazlar kontrol edilir.
- **Örnek:**
  - Akıllı bir lambanın açılıp kapanması.
  - Bir sulama sisteminin vanayı açması.
- **Kullanılan Teknoloji:** Aktüatörler, MQTT gibi protokoller.
- **Amaç:** Kullanıcıya veya sisteme otomasyon ve kontrol imkanı sunar.

### 3. İzleme Servisleri (Monitoring Services)

- **Tanım:** Gerçek zamanlı veya periyodik olarak durumu takip eden servislerdir.
- **Nasıl Çalışır:** Sensörlerden gelen veriler sürekli izlenir ve raporlanır.
- **Örnek:**
  - Bir hasta monitörünün nabızı takip etmesi.
  - Bir araç filosunun konumunu izlemesi.
- **Kullanılan Teknoloji:** Bulut sistemleri, dashboard arayüzleri.
- **Amaç:** Durum farkındalığı sağlar, anormallikleri tespit eder.

### 4. Öngörü Servisleri (Predictive Services)

- **Tanım:** Verileri analiz ederek gelecekteki olayları tahmin eden servislerdir.
- **Nasıl Çalışır:** Makine öğrenimi ve veri analitiği ile modeller oluşturulur.
- **Örnek:**
  - Bir fabrikanın makine arızasını önceden tahmin etmesi (predictive maintenance).
  - Bir enerji sisteminin talep artışını öngörmesi.
- **Kullanılan Teknoloji:** AI, bulut analitiği (örneğin AWS IoT Analytics).
- **Amaç:** Proaktif çözümler sunar, maliyeti ve riski azaltır.

### 5. Otomasyon Servisleri (Automation Services)

- **Tanım:** İnsan müdahalesi olmadan otomatik kararlar alan ve uygulayan servislerdir.
- **Nasıl Çalışır:** Bağlam (context) ve kurallara göre cihazlar kendi kendine çalışır.
- **Örnek:**
  - Evde kimse yoksa ışıkları kapatan bir sistem.
  - Bir depoda stok azaldığında sipariş veren IoT cihazı.
- **Kullanılan Teknoloji:** Kenar bilişim (edge computing), gömülü yazılımlar.
- **Amaç:** Verimliliği artırır, manuel işleri ortadan kaldırır.

### 6. Entegrasyon Servisleri (Integration Services)

- **Tanım:** Farklı IoT cihazlarını veya sistemleri birleştiren servislerdir.
- **Nasıl Çalışır:** Birden fazla cihazın verisi bir araya getirilir ve ortak bir hizmet sunulur.
- **Örnek:**
  - Akıllı ev hub'ının ışık, kapı ve termostatu tek bir sistemde birleştirmesi.
  - Bir fabrikada sensörlerin ve robotların entegre çalışması.
- **Kullanılan Teknoloji:** API'ler, middleware (örneğin Node-RED).
- **Amaç:** Karmaşık sistemlerin uyumlu çalışmasını sağlar.

### 7. Bildirim Servisleri (Notification Services)

- **Tanım:** Kullanıcıya bilgi veya uyarı gönderen servislerdir.
- **Nasıl Çalışır:** Belirli bir olay gerçekleştiğinde (örneğin eşik aşımı) bildirim gönderilir.
- **Örnek:**



- Bir güvenlik kamerasının hareket algıladığında telefonuna mesaj göndermesi.
- Bir su sensörünün sızıntı tespitinde uyarı vermesi.
- **Kullanılan Teknoloji:** Push bildirimleri, SMS, e-posta entegrasyonu.
- **Amaç:** Kullanıcıyı bilgilendirir, acil durumlarda tepki süresini kısaltır.

## 8. Veri Analitiği Servisleri (Data Analytics Services)

- **Tanım:** Toplanan verilerden anlam çıkaran ve raporlayan servislerdir.
- **Nasıl Çalışır:** Büyük veri analiziyle trendler ve içgörüler elde edilir.
- **Örnek:**
  - Bir fitness bilekliğinin haftalık aktivite raporu sunması.
  - Bir perakende mağazasının müşteri hareketlerini analiz etmesi.
- **Kullanılan Teknoloji:** Bulut platformları (Google Cloud IoT), veri görselleştirme araçları.
- **Amaç:** Stratejik karar alma ve kullanıcı deneyimini geliştirme.

## IoT'da Servislerin İşleyiş Mekanizması

Bu servisler, şu şekilde hayata geçer:

1. **Cihaz Katmanı:** Sensörler veri toplar, aktüatörler eyleme geçer.
  2. **Bağlantı Katmanı:** Veriler Wi-Fi, ZigBee gibi teknolojilerle iletilir.
  3. **İşleme Katmanı:** Bulut veya kenar cihazları veriyi işler (örneğin analiz yapar, karar verir).
  4. **Sunum Katmanı:** Kullanıcıya bir arayüz (app, web) veya doğrudan eylem (lambda yanması) ile sunulur.
- Örnek Akış:**

- **İzleme + Kontrol:** Bir bebek monitörü (izleme) ses algıladığında ninniye çalar (kontrol).
- **Öngörü + Bildirim:** Bir makine sensörü arıza öngörür (öngörü) ve teknisyene uyarı gönderir (bildirim).

## IoT'de Hangi Tür Hizmetler Var?

IoT, yani Nesnelerin İnterneti, cihazların internete bağlanarak sunduğu çeşitli hizmetleri içerir. Bu hizmetler, evden endüstriye kadar geniş bir yelpazede bulunur. Örneğin:

- **Akıllı Ev Hizmetleri:** Işıkların uzaktan kontrolü, enerji yönetimi.
  - **Sağlık Hizmetleri:** Nabız takibi, ilaç hatırlatıcıları.
  - **Endüstriyel Hizmetler:** Makine arızalarını önceden tahmin etme, lojistik takibi.
  - **Akıllı Şehir Hizmetleri:** Trafik akışını optimize etme, güvenlik sistemleri.
- Bu hizmetler, cihazların çevreden veri toplaması ve bu verileri analiz ederek kullanıcıya fayda sağlamasıyla çalışır. Örneğin, bir akıllı termostat evin sıcaklığını ayarlar; bu, hem konfor sağlar hem de enerji tasarrufu yapar.

## Hizmetler Nasıl Gerçekleşir?

Bir IoT hizmeti genellikle şu adımları izler:

1. **Veri Toplama:** Sensörler (örneğin sıcaklık, hareket) çevreden veri toplar.
2. **Veri İletimi:** Bu veriler Wi-Fi, Bluetooth gibi bağlantılarla buluta veya bir sunucuya gönderilir.
3. **Veri İşleme:** Bulut, verileri analiz eder ve karar verir (örneğin, ışığı aç).
4. **Kullanıcı Etkileşimi:** Kullanıcı, bir mobil uygulama üzerinden hizmeti kontrol eder veya sonuçları görür. Bu süreç, cihazların birbiriyle ve kullanıcıyla nasıl iletişim kurduğunu gösterir. Örneğin, bir fitness bilekliği

adımlarınızı sayar, veriyi buluta gönderir ve siz uygulamadan günlük raporu alırsınız.

### IoT Hizmet Türlerinin Sınıflandırılması

IoT hizmetleri, sundukları işlevsellik ve hedefledikleri sektörler göre çeşitli kategorilere ayrılabilir. Aşağıdaki tablo, farklı sektörlerdeki yaygın IoT hizmetlerini özetlemektedir:

Sektör	Hizmet Türü	Örnek
Akıllı Ev	Ev Otomasyonu, Enerji Yönetimi	Işıkların uzaktan kontrolü, enerji tasarrufu
Sağlık ve Fitness	Uzaktan Hasta Takibi, İlaç Hatırlatıcıları	Nabız ölçümü, ilaç alma hatırlatıcıları
Endüstriyel	Öngörülü Bakım, Tedarik Zinciri Yönetimi	Makine arızası tahmini, RFID ile kargo takibi
Akıllı Şehirler	Trafik Yönetimi, Kamu Güvenliği	Trafik ışık optimizasyonu, güvenlik kameraları
Tarım	Hassas Tarım, Otomatik Sulama	Toprak nemi takibi, otomatik sulama sistemleri
Enerji	Akıllı Sayaçlar, Talep Yanıtı	Gerçek zamanlı enerji tüketimi izleme
Perakende	Akıllı Alışveriş Arabaları, Beacon Pazarlama	Kişiselleştirilmiş öneriler, mağaza içi reklam

Bu hizmetler, IoT'nin geniş uygulama alanlarını yansıtır. Örneğin, akıllı ev hizmetleri, kullanıcı konforunu artırırken enerji yönetimi hizmetleri sürdürülebilirlik sağlar. Endüstriyel hizmetler ise işletmelerin verimliliğini artırarak maliyetleri düşürür.

### Hizmetlerin Gerçekleşmesinde Kullanılan Teknolojiler

IoT hizmetlerinin çalışması, çeşitli teknolojilere dayanır. Aşağıdaki tablo, bu teknolojilerin bir özetini sunar:

Teknoloji	Kullanım Alanı	Örnek
Wi-Fi	Yüksek hız, ev içi IoT	Akıllı ampuller, termostatlar
Bluetooth	Kısa mesafeli, düşük enerji	Akıllı saatler, fitness bileklikleri
ZigBee	Düşük güç, mesh ağlar	Akıllı ev sistemleri
LoRaWAN	Uzun mesafeli, düşük güç	Tarım sensörleri, şehir IoT
Hücrel (4G/5G)	Geniş alan kapsama	Araç takip sistemleri, filo yönetimi
MQTT, CoAP	Hafif protokoller, veri iletimi	Bulut ile cihaz iletişimi
Bulut Platformları	Veri işleme, depolama	AWS IoT, Google Cloud IoT

Bu teknolojiler, hizmetlerin verimli ve güvenilir olmasını sağlar. Örneğin, MQTT protokolü, düşük bant genişliği gereksinimleriyle enerji verimliliği sağlar, bu da pil ile çalışan IoT cihazları için idealdir.

## 12- IoT de kullanılan mimari çeşitleri nelerdir? 3, 4, 5li mimari katmalarını açıklayın ve karşılaştırın

Nesnelerin İnterneti (IoT) sistemlerinin mimarisi, farklı ihtiyaçlar ve uygulama alanlarına göre çeşitli katmanlardan oluşabilir. Genellikle 3, 4 ve 5 katmanlı mimari modeller kullanılmaktadır.

Nesnelerin İnterneti (IoT), cihazların internete bağlanarak iletişim kurduğu ve hizmet sunduğu bir teknoloji olarak, farklı mimari modellerle uygulanır. Bu rapor, IoT'daki 3, 4 ve 5 katmanlı mimari çeşitlerini detaylı bir şekilde ele alır

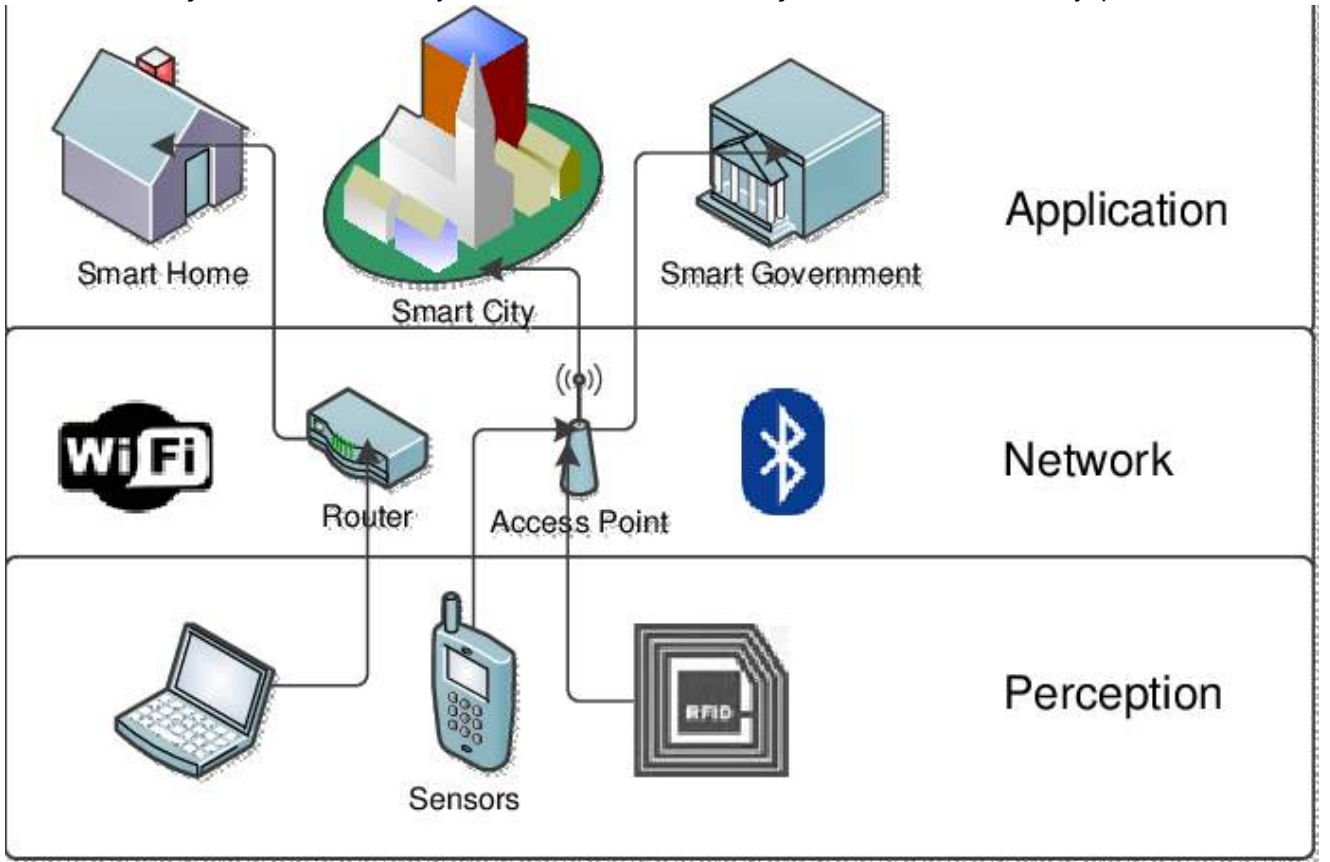
ve bu modelleri karşılaştırır. Araştırma, mimari seçiminin projenin karmaşıklığına ve ihtiyaçlarına bağlı olduğunu göstermektedir. Aşağıda, her mimari katmanı ve karşılaştırma tablosu ile detaylı bir analiz sunulmuştur.

### 3-Katmanlı IoT Mimarisinin Detaylı İncelemesi

3 katmanlı mimari, IoT'nin en temel ve yaygın kullanılan modelidir. Bu model, özellikle basit ve küçük ölçekli projeler için uygundur. Katmanlar şunlardır:

- **Algılama Katmanı (Perception Layer):** Bu katman, çevreden veri toplayan sensörler ve cihazlardan oluşur. Örneğin, bir sıcaklık sensörü odanın sıcaklığını ölçer. Bu katman, IoT sisteminin fiziksel dünyayla etkileşimini sağlar.
- **Ağ Katmanı (Network Layer):** Verilerin cihazlardan merkezi sisteme iletilmesini sağlar. Wi-Fi, Bluetooth, ZigBee gibi iletişim protokolleri bu katmanda kullanılır. Bu katman, veri akışını ve cihaz bağlantısını yönetir.
- **Uygulama Katmanı (Application Layer):** Toplanan verilerin işlenip kullanıcıya sunulduğu katmandır. Mobil uygulamalar, web arayüzleri ve veri analitiği bu katmanda gerçekleşir. Örneğin, bir kullanıcı akıllı ev uygulamasından ışıkları kontrol edebilir.

Bu model, veri işleme ve kullanıcı arayüzünü tek bir katmanda birleştirir, bu da daha basit bir yapı sunar.

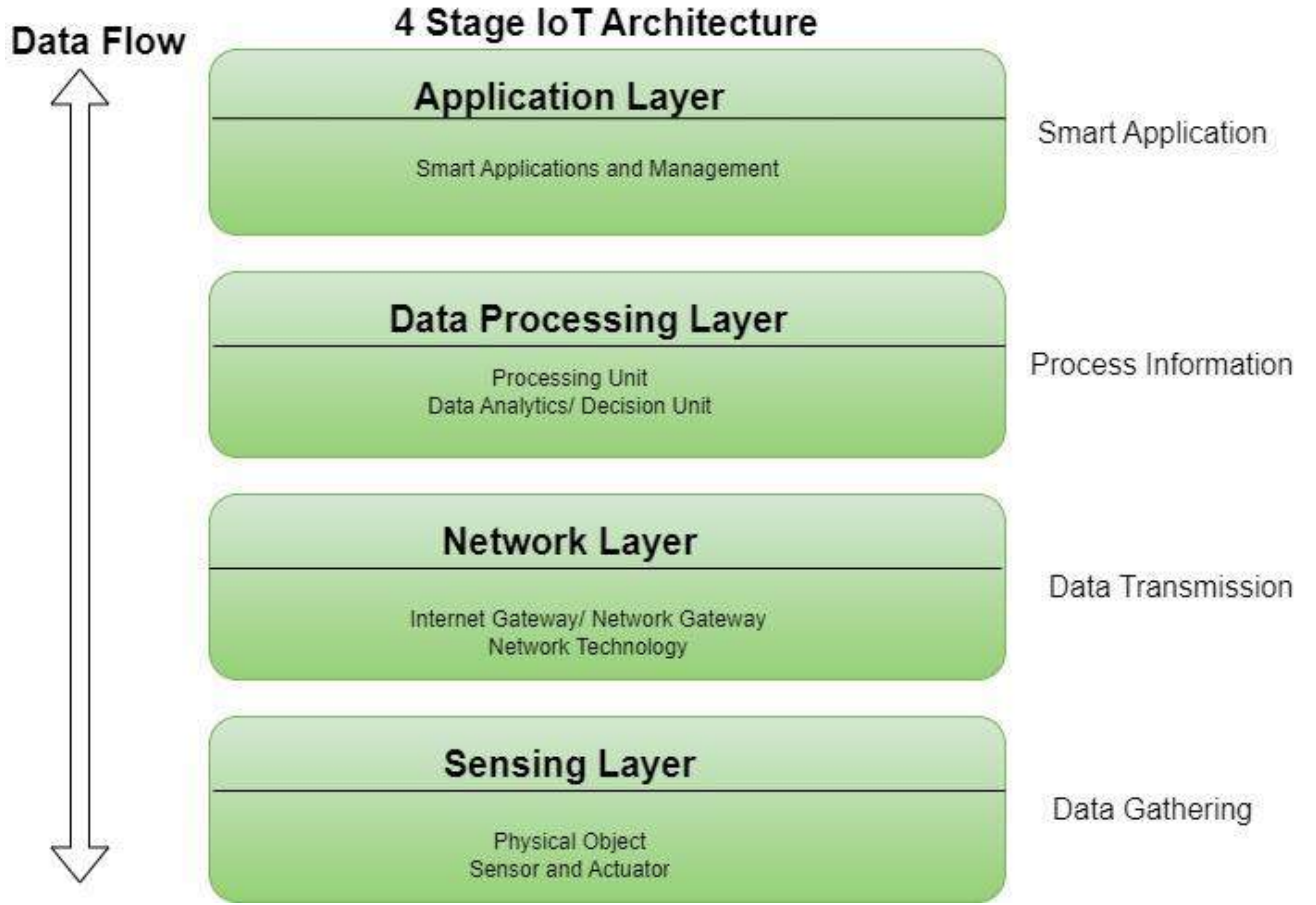


### 4-Katmanlı IoT Mimarisinin Detaylı İncelemesi

4 katmanlı mimari, 3 katmanlı modele ek olarak veri işleme katmanını ayırır, böylece daha karmaşık sistemler için uygundur. Katmanlar şunlardır:

- **Algılama Katmanı (Sensing Layer):** Çevreden veri toplayan sensörler ve cihazlar. Örneğin, bir hareket sensörü güvenlik için hareket algılar.
- **Bağlantı Katmanı (Connectivity Layer):** Cihazlar arası iletişim ve veri iletimi için sorumludur. Gateway'ler ve iletişim protokolleri (ör. 5G, LoRaWAN) bu katmanda yer alır.
- **Veri İşleme Katmanı (Data Processing Layer):** Toplanan verilerin analiz edildiği ve depolandığı katmandır. Bulut platformları veya kenar bilişim burada veri madenciliği yapabilir.
- **Kullanıcı Arayüzü Katmanı (User Interface Layer):** Kullanıcıların sistemle etkileşime geçtiği katmandır. Mobil uygulamalar, dashboard'lar ve raporlama araçları bu katmanda sunulur.

Bu model, veri işleme ve kullanıcı arayüzünü ayırarak daha modüler bir yapı sunar, özellikle veri yoğun uygulamalar için uygundur.

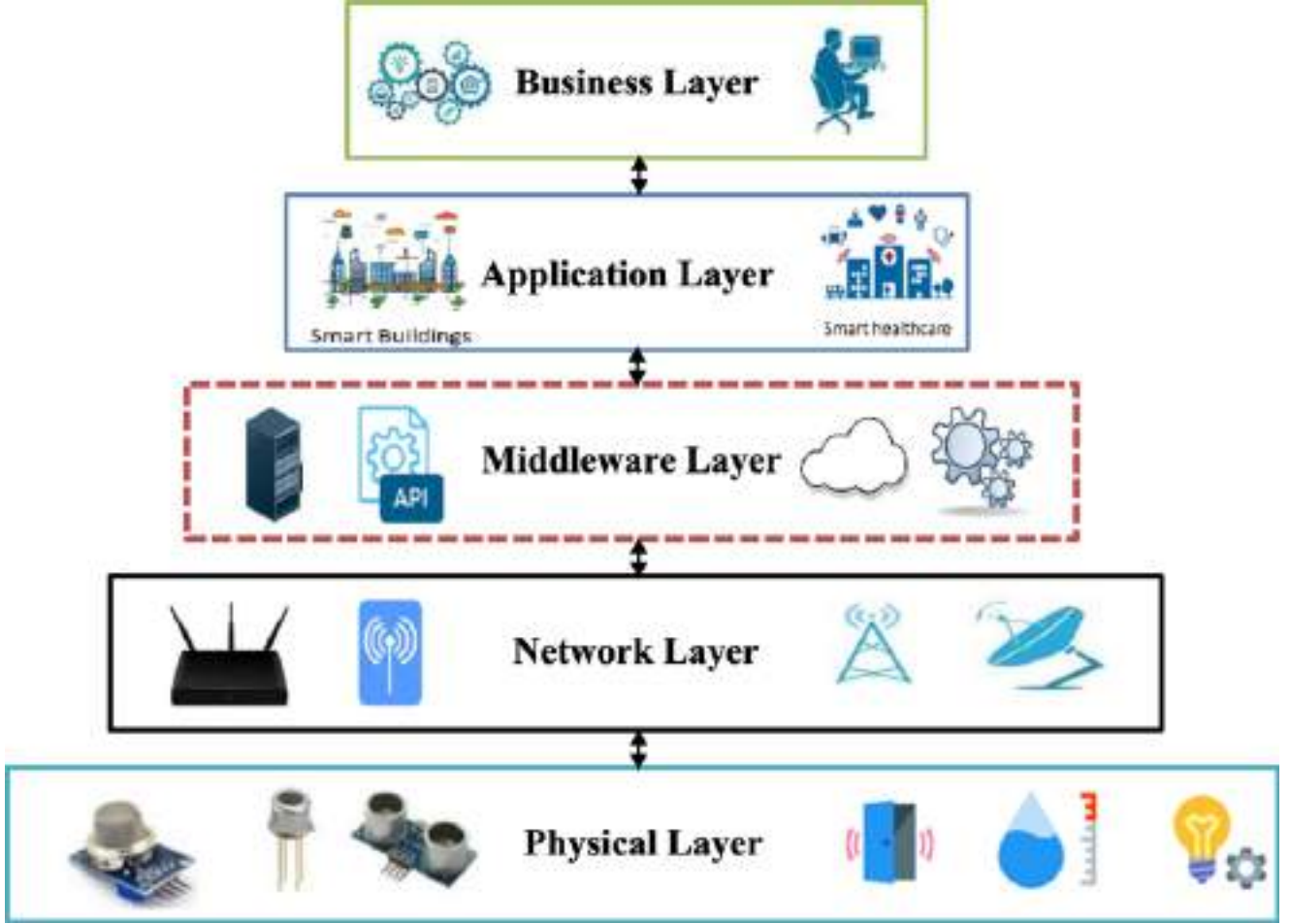


## 5-Katmanlı IoT Mimarisinin Detaylı İncelemesi

5 katmanlı mimari, 4 katmanlı modele ek olarak bir İş Katmanı ekler ve büyük ölçekli, işletme düzeyinde sistemler için tasarlanmıştır. Katmanlar şunlardır:

- **Algılama Katmanı (Sensing Layer):** Çevreden veri toplar (ör. sensörler).
- **Ağ Katmanı (Network Layer):** Verilerin iletilmesini sağlar (ör. Wi-Fi, ZigBee).
- **Veri İşleme Katmanı (Data Processing Layer):** Verileri analiz eder ve depolar.
- **Uygulama Katmanı (Application Layer):** Kullanıcıya hizmet ve arayüz sunar.
- **İş Katmanı (Business Layer):** İş süreçlerini yönetir, kararlar alır ve mevcut iş sistemleriyle entegre olur. Örneğin, bir fabrika, makine verilerini analiz ederek üretim planlamasını optimize edebilir.

Bu model, işletme düzeyinde karar alma ve entegrasyon gerektiren projeler için idealdir.



#### KARŞILAŞTIRMA:

##### Karşılaştırma Tablosu

Aşağıdaki tablo, 3, 4 ve 5 katmanlı mimarilerin karşılaştırmasını sunar:

Özellik	3-Katmanlı Model	4-Katmanlı Model	5-Katmanlı Model
Katmanlar	Algılama, Ağ, Uygulama	Algılama, Bağlantı, Veri İşleme, Arayüz	Algılama, Ağ, Veri İşleme, Uygulama, İş
Karmaşıklık	Düşük, basit projeler için uygun	Orta, veri yoğun projeler için ideal	Yüksek, işletme düzeyinde sistemler için
Veri İşleme	Uygulama katmanında birleştirilmiş	Ayrı bir katman, daha detaylı analiz	Ayrı katman, iş entegrasyonu ile genişletilmiş
Uygulama Alanı	Küçük ölçekli, ev otomasyonu	Orta ölçekli, sağlık izleme	Büyük ölçekli, akıllı şehirler
Avantaj	Kolay uygulanabilir, düşük maliyet	Daha modüler, veri analizi güçlü	İş süreçleriyle entegre, kapsamlı
Dezavantaj	Veri işleme sınırlı, ölçeklenebilirlik zor	Daha karmaşık, kaynak gereksinimi artar	Yüksek maliyet, karmaşıklık artar

#### 13- MQTT ve CoAP protokolleri arasındaki çalışma farklarını anlatınız.

MQTT, yayıncı abone modeline dayanırken CoAP istek cevap modeline dayanır. MQTT, TCP/IP üzerinde çalışırken Coap UDP tabanlıdır. MQTT, sensör gibi veri toplama alanında kullanılırken CoAP düşük güç tüketen



basit IoT cihazlarında kullanılır.

### MQTT(message queuing telemetry transport)

açık kaynaklı, TCP/IP üzerine kurulmuş ve tcp/ip'nin kullandığı windows/linux/ios o.s'lerde çalışan hafif bir mesajlaşma protokolüdür.

M2M iletişimde etkin çözüm

güvenlik için **TLS/SSL** desteği sunar.

nesnelerin internete bağlanabilmesi için gerekli server'ı oluşturur. cihazlar verileri direkt olarak değil bir

broker(aracı)(baz istasyonu gibi) üzerinden bir topic ile birbirlerine gönderir. cihazdan çıkan veri bir topic ile broker üzerinden publish edilir ve broker'a subscribe olan diğer cihazlar verileri görüntüleyebilir. işlemler şifreli veya şifresiz olabilir.

tcp/ip bağlantısı paketlerin sırayla gönderilmesini sağlar. clientler broker'a bağlanır ve sürekli veri bekler. her dakika mesaj yollanarak tcp yoklaması alınır. sinyal alınamazsa ping gönderilerek keepalive periyodu yapılır.

### 24- CoAPP

Mikrodenetleyiciler gibi sınırlı cihazlar ve ağlar için düşük maliyetli ve basit bir yapıya sahiptir. **UDP** üzerinden asenkron mesaj alışverişini yönetir.

M2M veri alışverişinde kullanılır.

güvenlik için **DTLS (Datagram Transport Layer Security)** kullanır. AES, RSA algoritmaları destekler.

mesaj ve request/response olmak üzere 2 katmandan oluşur.

http ile benzer çalışır fakat daha hafif.

**UDP Üzerinden İletim:** CoAP, HTTP'den farklı olarak TCP yerine UDP üzerinde çalışır. UDP, TCP gibi bağlantı tabanlı değildir, bu yüzden paketler daha hızlı iletilir ve veri trafiği daha düşüktür. Ancak, UDP'de TCP gibi bir hata kontrol mekanizması olmadığı için CoAP kendi hata kontrol yöntemlerine sahiptir.

COAP, tek bir isteği birden fazla cihaza gönderebilmek için multicast kullanabilir.

### COAP vs MQTT:

Özellik	COAP	MQTT
Protokol Türü	UDP tabanlı	TCP tabanlı
Mesajlaşma Modeli	Request-Response (RESTful)	Publish-Subscribe (Mesaj Kuyruğu)
Güvenlik	DTLS ile güvenlik	TLS/SSL ile güvenlik
Enerji Tüketimi	Düşük enerji tüketimi	Daha fazla enerji tüketebilir
Gecikme	Daha düşük gecikme	Daha yüksek gecikme
Kullanım Alanı	Düşük bant genişliği, IoT cihazları	IoT cihazları ve gerçek zamanlı uygulamalar
Multicast	Destekler	Desteklemez
Veri Bütünlüğü	Daha az güvenilir (UDP tabanlı)	Daha güvenilir (TCP tabanlı)

### BİTTİ EKSTRA

## 1. Context (Bağlam) Nedir?

#### Tanım:

IoT'da "context" (bağlam), bir cihazın veya sistemin çevresindeki durumu, koşulları ve ilgili bilgileri ifade eder. Yani, bir IoT cihazının neyi, nerede, ne zaman ve nasıl yaptığını anlamak için kullanılan verilerdir.

#### Örnekler:

- Bir akıllı termostatin odanın sıcaklığını, nemini ve kullanıcıların evde olup olmadığını bilmesi.
- Bir trafik sensörünün araç sayısını, hava durumunu ve zamanı algılaması.

#### IoT'da Bağlamın Rolü:

- **Karar Verme:** Cihazlar, bağlama göre akıllıca hareket eder. Mesela termostat, evde kimse yoksa ısıtmayı kapatır.
- **Kişiselleştirme:** Kullanıcıya özel çözümler sunar (örneğin, sabah kalktığında kahve makinesi otomatik çalışır).
- **Verimlilik:** Gereksiz işlemleri önler (örneğin, ışıklar sadece biri odadayken yanar).

#### Bağlam Türleri:

- **Fiziksel Bağlam:** Sıcaklık, konum, ışık seviyesi gibi çevresel veriler.
- **Kullanıcı Bağlamı:** Kullanıcının kim olduğu, alışkanlıkları, tercihleri.
- **Zaman Bağlamı:** Günün saati, tarih gibi zamanla ilgili bilgiler.
- **Sistem Bağlamı:** Cihazın durumu (pil seviyesi, bağlantı durumu).

#### Problem:

Bağlamı doğru algılamak için sensörlerin hassas ve güvenilir olması lazım. Yanlış veri, yanlış kararlara yol açar.

## 2. Context Sharing (Bağlam Paylaşımı) Nedir?

#### Tanım:

Bağlam paylaşımı, IoT cihazlarının topladığı bağlam bilgilerini diğer cihazlarla, sistemlerle veya uygulamalarla paylaşmasıdır. Bu, cihazların birbiriyle iş birliği yapmasını ve daha büyük bir sistemin parçası olmasını sağlar.

#### Örnekler:

- Bir akıllı evde, kapı sensörü "kapı açıldı" bilgisini ışığa gönderir ve ışık otomatik yanar.
- Bir akıllı şehirde, trafik sensörleri verilerini diğer sensörlerle paylaşarak trafik ışıklarını optimize eder.

#### IoT'da Bağlam Paylaşımının Önemi:

- **İşbirliği:** Cihazlar tek başına değil, bir ekip gibi çalışır.
- **Verimlilik:** Aynı veriyi tekrar toplamak yerine paylaşım yapılır (enerji tasarrufu).
- **Akıllı Sistemler:** Daha karmaşık senaryolar mümkün olur (örneğin, bir hastanede IoT cihazları hasta durumunu paylaşıp doktoru uyarır).

#### Zorluklar:

- **Gizlilik:** Bağlam verileri (örneğin konum) hassas olabilir, paylaşırken güvenlik önemli.
- **Uyumluluk:** Farklı cihazlar farklı formatlarda veri ürettiği için standartlaşma gerekir.
- **Gecikme:** Verinin hızlı paylaşılması lazım, yoksa sistem geç tepki verir.

#### Teknolojiler:

- **MQTT, CoAP:** Hafif protokoller, bağlam paylaşımını kolaylaştırır.
- **Bulut/Edge Computing:** Veriler merkezi bir yerde veya yerel olarak paylaşılır.

## 3. Service (Hizmet) Nedir?

**Tanım:**

IoT'da "service" (hizmet), IoT cihazlarının veya sistemlerinin kullanıcılara, diğer cihazlara veya uygulamalara sunduğu işlevselliktir. IoT'un amacı, bu hizmetlerle hayatı kolaylaştırmaktır.

**Örnekler:**

- Bir akıllı sulama sisteminin tarlayı otomatik sulaması (hizmet: sulama).
- Bir fitness bilekliğinin adım sayısını gösterip uyarı vermesi (hizmet: sağlık takibi).
- Bir akıllı ev sisteminin kapıyı kilitlemesi (hizmet: güvenlik).

**IoT'da Hizmet Türleri:**

- **Temel Hizmetler:** Tek bir cihazın sunduğu işlev (örneğin, bir sensörün sıcaklık ölçmesi).
- **Karma Hizmetler:** Birden fazla cihazın iş birliğiyle sunduğu hizmet (örneğin, kapı açılınca ışığın yanması).
- **Bulut Tabanlı Hizmetler:** Verilerin bulutta işlenip kullanıcıya sunulması (örneğin, hava durumu tahmini).

**IoT'da Hizmetin Rolü:**

- **Kullanıcı Odaklılık:** İnsanların hayatını iyileştirir.
- **Otomasyon:** Manuel işleri azaltır (örneğin, çöp dolunca belediyeye haber veren bir IoT çöp kutusu).
- **Değer Yaratma:** İşletmeler için yeni fırsatlar sunar (örneğin, lojistikte takip hizmetleri).

**Problem:**

- **Güvenilirlik:** Hizmet kesintiye uğrarsa (örneğin internet giderse), sistem çalışmaz.
- **Kaynak Kısıtları:** Düşük güçlü cihazlar karmaşık hizmetleri sunmakta zorlanabilir.

---

## 4. Hareketlilik (Mobility) Nedir?

**Tanım:**

IoT'da "mobility" (hareketlilik), cihazların fiziksel olarak yer değiştirebilmesi ve bu sırada ağ bağlantısını sürdürebilmesidir. Hareketli IoT cihazları, sabit olanlardan farklı zorluklar getirir.

**Örnekler:**

- Bir araçtaki IoT sensörünün yolda giderken veri göndermesi.
- Bir drone'un tarım arazisinde uçarken durumu izlemesi.
- Bir akıllı saatin sen yürürken kalp atışını ölçmesi.

**IoT'da Hareketliliğin Türleri:**

- **Düşük Hareketlilik:** Cihaz sabit bir erişim noktasına bağlı kalır (örneğin, evde Wi-Fi'ye bağlı bir robot süpürge).
- **Yüksek Hareketlilik:** Cihaz sürekli baz istasyonları arasında geçiş yapar (örneğin, bir arabanın 5G bağlantısı).
- **Bağlantı/Kopma Hareketliliği:** Cihaz ara sıra ağa bağlanır (örneğin, bir kargo takip cihazı).

**Hareketliliğin IoT'daki Rolü:**

- **Esneklik:** Cihazlar her yerde çalışabilir (örneğin, bir IoT sağlık cihazı hastayla hastaneye gider).
- **Gerçek Zamanlı Veri:** Hareket halindeyken bile veri toplanır (örneğin, trafik izleme).

**Hareketlilikle İlgili Problemler:**



- **Handoff (El Değiştirme):** Cihaz bir baz istasyonundan diğerine geçerken bağlantı kopabilir (daha önce detaylı anlattım).
- **Enerji:** Hareketli cihazların pili çabuk bitebilir.
- **Bağlantı Sürekliliği:** Kapsama alanı dışına çıkarsa iletişim kesilir.
- **Gecikme:** Ağ değişimi veri akışını yavaşlatabilir.

#### Çözüm Yaklaşımları:

- **Mobile IP:** Cihaz hareket ederken IP adresini korur.
- **5G:** Daha hızlı ve kesintisiz bağlantı sağlar.
- **Düşük Güç Protokolleri:** LoRaWAN gibi teknolojiler hareketli cihazlar için uygundur.

---

## IoT'daki Bu Kavramların Birbirleriyle İlişkisi

- **Bağlam:** Cihazın çevresini anlaması, hizmet sunabilmesi için temel.
- **Bağlam Paylaşımı:** Cihazların bağlamı paylaşması, karma hizmetlerin çalışmasını sağlar.
- **Hizmet:** Bağlam ve paylaşımın sonucunda kullanıcıya sunulan fayda.
- **Hareketlilik:** Tüm bu süreçlerin cihaz hareket halindeyken de devam etmesi.

#### Örnek Senaryo:

Bir IoT trafik sistemi düşün:

- **Bağlam:** Sensör, trafik yoğunluğunu ve hava durumunu algılar.
- **Bağlam Paylaşımı:** Bu veri trafik ışıklarına ve diğer sensörlere gönderilir.
- **Hizmet:** Trafik ışıkları optimize edilerek akış sağlanır.
- **Hareketlilik:** Araçtaki IoT cihazı hareket ederken bu verileri kullanır.

---

## Vize İçin Özet

- **Context:** Cihazın çevresel bilgisi (sıcaklık, konum vb.).
- **Context Sharing:** Bu bilginin diğer cihazlarla paylaşılması.
- **Service:** Kullanıcıya sunulan işlev (sulama, güvenlik vb.).
- **Mobility:** Cihazın hareket ederken bağlantıyı sürdürmesi.