

Kablosuz Algılayıcı Ağlar (WSN)

Algılayıcı Düğüm

- Ortamdaki fiziksel büyüklükleri algılayabilen , Düşük maliyetli tüm devredir.
- Rasgele yerleştirilebilme
- Kendi kendine organize olabilme
- Ortak çalışma
- Yerel hesaplama yapma

KAA Türleri

- Yeraltı Kablosuz Algılayıcı Ağlar
- Su altı Kablosuz Algılayıcı Ağlar
- Kablosuz Vücut Algılayıcı Ağlar
- Araç Alan Ağları
- Kablosuz Çoklu Ortam Algılayıcı Ağlar

KAA Mantıksal Mimari (MICAz)

- **İşleme** → uygulamaların çalıştırılması , Kaynakların Yönetimi, Çevre Birimlerin Kontrolü
- **Giriş / Çıkış** → Algılama / programlama boardları arasında arabirim
- **Saklama** → Algılama değerlerini saklar
- **Güç Yönetimi** → Sistem besleme gerilimini düzenler
- **Algılama** → Ortamdaki fiziksel büyüklükleri algılamak
- **RF İletişimi** → Kablosuz olarak veri gönderme ve alma , Düğümün Fiziksel katmanıdır

Kablosuz Algılayıcı Ağların Kısıtlamaları

- Üretim maliyeti
- Donanımsal Sınırlamalar
- Güç tüketimi
- Çevresel Koşullar
- Hata toleransı
- İletim ortamı
- Ölçeklenebilirlik

Z-Wave → Ev ve ofis ortamlarında elektrik/elektronik cihazlar ile haberleşmek için geliştirilmiş

ANT/ANT+ → ANT, sensör ağlar, sağlık, spor gibi benzer uygulamalar için geliştirilmiş

Mobil Teknolojiker

Hücresel ağlar

- Mobil sistemlerde, haberleşme alanı hücre (cell) adı verilen küçük coğrafik alanlara ayrılır.
- Tek bir yüksek güçlü verici kullanımı yerine çok sayıda düşük güçlü verici kullanımı
- Coğrafi olarak ayrı alanlarda aynı frekansların tekrar kullanım
- Bir hücredeki baz istasyonunun bozulması durumunda tüm sistemin değil yalnızca ilgili hücre alanının etkilenmesi

Genel Paket Radyo Servisi (GPRS)

- Hücresel ağlar üzerinden iletişim için kullanılan bir standarttır.
- Genellikle 2,5G teknoloji olarak adlandırılır.

3G'nin getirmiş olduğu yenilikler

- Mesajlaşma, internet erişimi ve yüksek hızda çoklu ortam haberleşme desteği
- Konumlandırma hizmetlerinin sağlanması
- ØKonumlandırma hizmetlerinin sağlanması
- Mevcut şebekelere geriye doğru uyum sağlayabilme, düşük kurulum maliyeti

4G → IPv6 tabanlı bir iletişim teknolojisidir , 4G tamamıyla IP ile tümleştirilmiş bir sistemdir

WiMAX

- WiMAX teknolojisi sabit, taşınabilir ve mobil erişimleri destekleyen bir genişbant kablosuz erişim teknolojisidir.
- Uygulama alanları:
 - Savunma
 - Enerji
 - Kamu Güvenliği
 - Kurumsal
 - Kamu & Belediye
 - Transport

Uzun Süreli Evrim (LTE)

- LTE'nin en önemli artışı yüksek kapasite ve veri aktarım hızı özellikleridir
- Verizon **LTE** 700 MHz'de çalışıyor ve teorik olarak 100 Mbit/s hızı destekliyor
- Sprint **WiMAX** ağı ise 2.5 GHz frekansında teorik olarak 128 Mbit/s hızı desteği sunuyor

- ❖ **GNSS**, uydularla konum belirleme (seyrüsefer) sistemleri için kullanılan genel bir addır
- ❖ Küresel Konumlama Sistem (**GPS**)
 - Dünya çevresini yaklaşık 12 saatte bir tur atan 31 adet uydudan oluşan bir sistemdir.
 - GPS sistemi, uydular (uzay bölümü), kontrol (yer istasyonları) ve kullanıcı (GPS alıcılar) olmak üzere 3 ana kısımdan oluşmaktadır

Düşük Güç Geniş Alan Ağı Teknolojiler (LPWAN)

- ❖ **LPWAN teknolojileri, IoT uygulamalarının birçok temel gereksinimi karşılayan kablosuz (RF) iletişim teknolojileridir**
 - Geniş coğrafik alanda (km'lerce mesafede, 2-1000 km) uzun mesafeli iletişim desteği
 - Geniş coğrafik alanda (km'lerce mesafede, 2-1000 km) uzun mesafeli iletişim desteği
 - Düşük enerji tüketimi
 - Düşük maliyet
 - Küçük boyutlu veri iletimi, (10-1000 bayt)
- ❖ **Birçok LPWAN temelli teknoloji bulunmaktadır:**
 - **Sigfox** → Sigfox, çift yönlü iletişim sunan ultra dar bant teknolojidir.
 - **LoRaWAN** → düşük güçlü alan ağının sistem mimarisi ve iletişim protokolünü tanımlar
 - **NB-IoT** → Mobil operatörler tarafından sağlanan NB-IoT, 3GPP standartlarına dayalı lisanslı frekans üzerinden haberleşen bir teknolojidir
 - **LTE – M** → Diğer LPWAN teknolojilerine göre daha büyük bir bant genişliği sunar.
 - **eLTE-IOT** → Huawei şirketinin IoT ağların haberleşme ihtiyacı için geliştirilmiş lisanssız frekans bandı kullanan kablosuz haberleşme teknolojisidir.

IoT Protocols

Web Servisleri

- Ağ üzerinden birlikte çalışabilen makineden makine etkileşimi desteklemek için tasarlanmış yazılım sistemleri
- IoT konusunda uygulama / haberleşme protokollerin içerisinde

IoT uygulama protokolleri iki temel mimariye sahiptir

1. **Veriyolu Temelli** (Bus-based)
 - İstemciler, belirli bir konu için abonelere doğrudan ulaşan mesajları yayımlar
 - Merkezileştirilmiş sunucu ya da sunucu temelli servis yoktur.
2. **Sunucu Temelli** (Broker-based)
 - ✓ Sunucu bilginin/mesajın dağıtımını kontrol eder
 - ✓ İstemciler, uygulamanın amaçlarına bağlı olarak yayımcı ve abone rolleri arasında anahtarlar

IoT uygulama protokolleri mesaj açısında da sınıflandırabiliriz

1. **Mesaj Merkezli** (Broker-based)
 - Mesajın/verinin içeriğine bakılmaksızın (payload) alıcılarına ulaştırılmasına odaklanır
2. **Veri Merkezli** (data-centric)
 - Mesajın/verinin alıcı tarafından anlaşıldığını varsayar ve verinin ulaştırılmasına odaklanır

Temsili Durum Transferi Protokolü (REST)

- ✓ REST, makineler arasında bağlantı kurmak için basit HTTP kullanan ağ uygulamalarının tasarımı için işletim sistemi ve dil bağımsız bir mimaridir
- ✓ REST bir **standart değildir**, içerisinde HTTP, URL, XML vb. standartları barındıran bir mimari yaklaşımdır.

- ✓ **İstemci** (client) / **Sunucu** (server) ve istek (request) / yanıt (response) modeline dayalı eşe-eş (**P2P**) haberleşme yapısına sahiptir
- ✓ REST mimarisini kullanan servislere genel olarak **RESTful servis** denir
- ✓ HTTP üzerinden POST, GET, DELETE metotları ile iletişimi gerçekleştirir
- ✓ Ulaşım katmanı olarak **TCP** kullanır
- ✓ Güvenlik https
- ✓ Mesaj formatında başlık ASCII yapısında, yük/data (payload) ise XML, JSON, HTML yapısındadır
- ✓ Diğer protokollere göre daha fazla güç, kaynak ve veri kullanır

REST Web Servisinin Özellikleri ve Kısıtlamaları (Tasarım İlkeleri)

- ❖ **İstemci – Sunucu (Client – Server)**
 - İstemci (kullanıcı arayüzleri) ve sunucu (veri saklama birimleri) yapısının birbirinden ayrı olması anlamına gelir.
 - İstemcinin veri kaynağı hakkında hiçbir şey bilmemesi ve sunucunun doğru istekler geldiği sürece yanıt vermesidir.
- ❖ **Durumsuzluk (Stateless)**
 - Durumsuzluk, istemci-sunucu sisteminin iletişim sağlandığında uygulamanın herhangi bir durumda olmamasıdır
- ❖ **Ön Bellekleme (Cache)**
 - İstek ve cevap arasındaki veri sunucu tarafından ön belleğe alınabilir (cacheble) olarak işaretlenirse bu veri istemci (kullanıcı) tarafından ön belleklenir ve daha sonra aynı istekler için tekrar kullanılabilir.
- ❖ **Tek Biçimlilik (Uniform Interface)**
 - Sunucu ve istemci için tek biçimli (ortak) arayüz, iletişim yöntemini basitleştirmektedir. Böylece veri iletişiminin takibi kolaylaşır
- ❖ **Katmanlı Sistem (Layered System)**
 - Katmanlı sistem ile hiyerarşik bir mimari sağlanır ve her katman yalnızca komşu (alt ya da üst) katmanı bilmektedir.
- ❖ **İstek Durumunda Kod (Code-On-Demand)**
 - İhtiyaç durumunda istemci bazı kaynaklara ulaşabilir ancak bu kaynakları nasıl kullanacağını bilmez.
 - İsteğe bağlı bir kısıtlamadır

Basit Nesne Erişim Protokolü (SOAP)

- SOAP, merkezeleştirilmemiş, dağıtık ortamlar tarafından yapısal bilgi değişimi için tasarlanmıştır
- Yayın (broadcast) mesajları için kullanılabilir
- İnternet Komitesi W3C (World Wide Web Consortium) standardıdır
- Uzak İşlev Çağırma (Remote Procedure Call, RPC) modelini kullanan İstemci – Sunucu mimarisine dayalı bir protokoldür

SOAP 4 temel kısımdan oluşur

✚ SOAP Zarfı (Envelope)

- Mesaj içeriğini ve nasıl işleneceğini açıklayan bir zarf (SOAP mesajın tüm iskeletini oluşturur)

✚ SOAP Kodlama Kuralları (Encoding Rules)

- Uygulamalar tarafından tanımlanabilen veri türleri için kodlama kurallarını içerir

✚ SOAP RPC Sunum (RPC Representation)

- Uzaktan prosedür çağırımı ve geri dönüşlerini tanımlar

✚ Protokol Bağlantısı (Protocol Binding)

- SOAP'ın HTTP içerisinde nasıl kullanılacağını tanımlar

Restful VS SOAP

- **RESTful** daha düşük veri boyutları kullanmaktadır
- **RESTful**, güncel internet standartlarını kullanmakta iken, **SOAP** tabanlı uygulamalar/yaklaşımlar kendi standartlarını içermektedirler
- **SOAP** servisleri, sistemler arası uyumluluk uygulamaları için tercih edilirler
- **SOAP** servislerin , geliştirilmesi ve tasarımı daha komplekstir

Genişletilebilir Mesajlaşma ve Durum Protokolü (XMPP)

- **XMPP**, genişletilebilir işaretleme dili (XML) tabanında, gerçek zamanlı sayılabilecek, mesajlaşma, durum yönetimi ve istek-yanıt hizmetleri için geliştirilmiş bir uygulama protokolüdür
- **XMPP**, istemciler arasında veri iletimi için, XML paket formatını kullanan **gerçek zamanlı iletişim** protokolüdür
- **XMPP**, internet üzerinden **anlık mesaj gönderimi** ile kullanıcıların birbirleri ile haberleşmesine izin verir

XMPP Adresleme

- XMPP ağı üzerindeki her varlık JID (jabber kimliği) olarak adlandırılan tek bir adrese sahiptir
- Bir JID kimliği, user@domain/resource (kullanıcı@alanadı.com/kaynak) yapısında, yerel ad, alan adı ve kaynak bildirimlerine sahiptir
- Temel olarak bare jid XMPP federasyonundaki o kullanıcıyı ifade ederken, full jid anlık oturumu ifade etmektedir

XMPP Sinyalleşme

- Genel olarak bir XMPP sinyalleşme süreci yapısal olarak 6 katmandan oluşur

XMPP Paket Yapıları

- XMPP sinyalleşme gerçekleştirildikten sonra bağlantı üzerinden XMPP'de tanımlı XML paket yapıları üzerinden iletişim akışı sürdürülür
- XML paket yapıları XML Stanza olarak isimlendirilir
- XMPP paket yapıları, alıcılar tarafından paketin nasıl işleneceğini belirleyen öznitelik:değer tanımlamalarından oluşur
- XMPP'de XML paket formatı 3 temel tanımlama alanına sahiptir → **mesaj , durum , bilgi/sorgu**
- XMPP paket yapıları 5 farklı özniteliğe sahiptir → **kime , kimden , kimlik , Tip, XML dili**

Sınırlanmış Uygulama Protokolü (CoAP)

- ✚ **CoAP**, nesnelerin interneti (makineler arası haberleşme) uygulamaları için tasarlanmış, güç, işlem kapasitesi, bellek gibi kısıtlı/sınırlanmış düğümler ve ağlar için özelleştirilmiş bir web transfer protokolüdür.
- ✚ **CoAP**, HTTP fonksiyonlarının üstündeki **REST** (Representational State Transfer)'e dayalı bir web transfer protokolüdür.
- ✚ **CoAP**, istek/yanıt haberleşmeyi kullanarak **istemci/sunucu modeline** dayanır
- ✚ **CoAP**, UDP üzerinden kullanılan **düşük ağırlıklı** (lightweight) bir protokoldür
- ✚ yayımcı/abone (Publisher/subscriber) mimarisini destekler

Mesaj Kuyruk Telemetri Ulaştırma (MQTT)

- ◆ **MQTT**, nesnelerin interneti ya da makineler arası haberleşmede kullanılan **mesajlaşma** protokolüdür
- ◆ **MQTT**, **TCP** protokolünün üzerine kurulmuştur
- ◆ **MQTT** mesajları 3 farklı servis kalitesi seviyesine (QoS) göre iletilmektedir. Best Effort, At Least Once, Exactly Once
- ◆ küçük cihazlardan oluşan ağları bulut üzerinden kontrol etmeyi ve izlemeyi sağlar
- ◆ **MQTT'in iki türü vardır**: MQTT v3.1 ve MQTT-SN (MQTT for Sensor Network)
- ◆ **MQTT**, **abone** (subscriber), **yayımcı** (Publisher) ve **sunucudan** (broker) oluşur
- ◆ **MQTT** Broker varsayılan portu **1883'tür**
- ◆ **Mosquitto**: En yaygın MQTT brokerlarından biridir
- ◆ **Güvenlik** olarak SSL/TLS destekler
- ◆ TCP/IP bağlantı türünü kullanır
- ◆ Bire bir, birden çoğa, çoktan çoğa bağlantı sağlar
- ◆ **Broker temelli** haberleşme mekanizmasına sahiptir
- ◆ **MQTT-SN** : Kablosuz algılayıcı ağlar için MQTT protokolü versiyonudur
- ◆ **Enerji tüketimi** HTTP protokolüne göre daha **düşüktür**

İleri Mesaj Kuyruklama Protokolü (AMQP)

- AMQP, **mesaj yönlendirme** odaklanan IOT için açık standart uygulama protokolüdür
- büyük veri yapıları ile haberleşme sağlayan bir protokoldür
- 3 farklı seviyedeki (at-most-once/best effort, at-least-once, exactly once delivery) mesaj ulaştırma garantisi ile güvenilir haberleşme sunar
- AMQP, ulaşım katmanı olarak TCP kullanır
- MQTT gibi **asenchron** bir protokoldür
- AMQP iki tür mesaj kullanır
 - **Bare mesaj**: Gönderici kaynaklı mesajlar
 - **Annotated mesaj**: Alıcıda görülen mesaj

Veri Dağıtım Servisi (DDS)

- ✚ gerçek zamanlı M2M haberleşmesi için geliştirilmiştir
- ✚ DDS, daha az katman, daha az iş yükü gibi avantajlar sunar
- ✚ DDS, publish-subscribe protokolüdür
- ✚ Ulaşım katmanı olarak UDP ve TCP'nin her ikisini de destekler
- ✚ Güvenlik olarak SSL/TLS ile Datagram TLS (DTLS) destekler
- ✚ DDS, Tokyo Metropolitan Otoban araç takibinde kullanılmaktadır
- ✚ DDS iki katmanlı mimariye sahiptir
 - Data-Centric Publish-Subscribe (**DCPS**): Abonelere bilginin ulaştırılmasından sorumludur
 - Data-Local Reconstruction Layer (**DLRL**): Opsiyonel bir katmandır ve DCPS fonksiyonlarının yerine getirilmesini sağlar. Dağıtık nesneler arasında dağıtık verinin paylaşılmasını kolaylaştırır

WebSocket

- ❖ Sunucu ve istemciler arasında TCP kanal üzerinden gerçek zamanlı ve çift yönlü bağlantı sunar
- ❖ WebSocket protokol, El Sıkışma (Handshake) ve veri transferi olmak üzere iki temel aşamadan oluşur
- ❖ Protokol bağımsız TCP tabanlı protokoldür
- ❖ HTTP'deki gibi şifresiz ya da TLS kullanılarak şifreli bağlantı sağlanabilir

IoT Uygulama Protokollerinin Karşılaştırılması

- ❑ Nesnelerin internetinde IEEE, ETSI gibi organizasyonların en sık tanımladığı protokoller,

Application Protocol	RESTful	Transport	Publish/Subscribe	Request/Response	Security	QoS	Header Size (Byte)
COAP	✓	UDP	✓	✓	DTLS	✓	4
MQTT	✗	TCP	✓	✗	SSL	✓	2
MQTT-SN	✗	TCP	✓	✗	SSL	✓	2
XMPP	✗	TCP	✓	✓	SSL	✗	-
AMQP	✗	TCP	✓	✗	SSL	✓	8
DDS	✗	TCP UDP	✓	✗	SSL DTLS	✓	-
HTTP	✓	TCP	✗	✓	SSL	✗	-

Büyük Veri ve Nesnelerin İnterneti (Big Data and IoT)



Özellik	Zorluk	Teknik (Çözüm Yöntemi)
Hacim (Volume)	Depolama / Ölçek	Dağıtık Dosya Sistemleri
Hız (Velocity)	Hızlı İşleme	Paralel Programlama
Çeşitlilik (Variety)	Heterojenlik	NoSQL Veritabanları
Değer (Value)	Bilgi Keşfi (Knowledge Discovery) Anlambilim (Semantics)	Veri Madenciliği Algoritmaları
Doğruluk (Veracity)	Analitik (Analytics)	

- ✓ Birbirine bağlı çok sayıda fiziksel cihazın veya sensörün internet üzerinden ürettiği veri büyük veri (**big data**) olarak adlandırılır
- ✓ Big Data kabul edilebilir bir sürede işlenebilir olmalıdır
- ✓ Büyük Veri Artma Nedenleri:
 - Sosyal ağ (facebook, twitter, linkedin vb.) kullanımının artması
 - Akıllı sensörler sayesinde lokasyon duyarlı cihazların artması
 - Fiziksel dünya hakkında bilgi yakalayan ve ileten akıllı sensörlerin sayısındaki artış

Mobil Büyük Veri Kaynakları:

- ◆ Mobil veri, gezgin kullanıcılar (akıllı telefon) ya da mobil ağlardan (gateway, baz istasyonu) elde edilir
- ◆ Mobil veri iki sınıfa ayrılır:
 - **Uygulama Verisi:** Akıllı telefon ya da gömülü sistem sensörlerinden (GPS, ivmeölçer vb.) toplanan veriler

- **Ağ Verisi:** Ağ operatörleri tarafından toplanan veriler (Kullanıcı ID, konum, cihaz tipi, servis tipi vb.)

IoT Büyük Veri Araçları



Hadoop Ekosistemi

- Hadoop, açık kaynak kodlu Java için geliştirilmiş büyük veri işleme aracıdır
- En yaygın kullanım örneği Facebook
- Hadoop'un sunduğu özellikler:
 - İşlem gücü, hata toleransı, esneklik, düşük maliyet, ölçeklenebilirlik



MapReduce

- Büyük veriyi, büyük kümelerle işlemeyi amaçlayan bir dağıtık programlama modelidir
- Google, web sayfalarını indekslemek için MapReduce kullanır
- MapReduce iki kısımdan oluşur:
 - **Map:** Bir yığının tüm üyelerini sahip olduğu fonksiyon ile işleyerek bir sonuç listesi döndürür.
 - **Reduce:** Paralel bir şekilde çalışan iki ya da daha fazla map fonksiyonundan dönen sonuçları harmanlar ve işler/çözer.



NoSQL

- NOSQL kullanan sistemlere örnek olarak Google ve Amazon tarafından kullanılan Big Table, DynamoDB verilebilir
- Veri için bir kullanım **ömrü belirlenebilir** (bir anahtara bağlanabilir), çok uzun tutulmayı gerektirmeyen veriler NoSQL'de tutulabilir
- Veriler doküman gibi saklanabilir. (MongoDB)
- veriler sütun halinde tutulabilir

MongoDB

- Doküman odaklı bir veri modeli kullanan açık kaynaklı NoSQL veritabanıdır
- Master, slave çoğaltmasını destekler
- Boyut farketmeksizin dosyaları kolayca depolar

Apache Spark

- genel amaçlı, hızlı, yüksek seviye API desteği ve zengin büyük veri araçları sunan küme hesaplama platformudur
- **Desteklediği Birimler/Teknolojiler**, **Spark SQL** (veri analizi için sorgu işlemleri), **Spark Streaming** (akış ile alınan veri işlemleri), **MLlib** (Makine öğrenmesi), GraphX (Grafik algoritmaları)

Apache Storm

- Yüksek hacimli verilerin işlenmesi için dağıtılmış bir gerçek zamanlı hesaplama sistemidir
- **Özellikleri**
 - Yüksek hızlı çalışma, Kolay kullanım, Güvenilir, Ölçeklenebilirlik, Hata toleransı

Apache Kafka

- dağıtık veri akış platformudur
- Sistemler ya da uygulamalar arasında güvenilir şekilde gerçek zamanlı veri akış hatları oluşturur
- Üretici, Tüketici, Akış ve Bağlayıcı olmak üzere **4 temel API desteği** sunar.

Elasticsearch

- Geniş hacimli verilere karşı hızlı ve duyarlı arama
- Belgelerin depoya indekslenmesi
- Geniş dağıtılabirlik ve yüksek ölçeklenebilirlik

Bulut Bilişim (Cloud Computing)

- ◆ **Bulut Bilişim**, kullanıcıların bilgi ve kaynaklara (depolama alanları, cihazlar vb.) internet üzerinden erişimi, paylaşımı ve bilgi üzerinde ortak işlem yapmalarını sağlayan bir teknolojidir
- ◆ **Bulut Bilişim Kullanımının Faydaları:**
 - Esneklik ve Verimlilik: Kapasite arttırma ve azaltma işlemlerinin kolaylığı
 - İnternet erişimi olan her yerden kullanım imkanı
 - Güvenlik , Yazılım güncelleme , Maliyet kazancı
- ◆ **Avantajları:**
 - Kullanıcının kullanılan teknolojiler ile ilişkisinin olmaması
 - Teknik destek kolaylığı (On-line)
 - İnternet üzerinden her yerden kullanılabilmesi
- ◆ **Dezavantajları:**
 - IaaS ve PaaS ile uyum problemleri
 - Üründe sınırlı değişiklik yapabilme

Nesnelerin İnterneti ve Bulut Bilişim

- ◆ **Karşılaşılan Zorluklar**
 - **Senkronizasyon** : Bulut Sistem tarafında cihazların verileri eş zamanlı alma zorluğu
 - **Standardizasyon**: IOT tarafında farklı üreticilerin cihazları arasındaki iletişim zorluğu
 - **Dengeleme** : Farklı altyapıların bulut bilişimde dengelenmiş servis sunabilmesi
 - **Güvenlik** : Bulut ve cihazlar arasındaki güvenlik
 - **Yönetim** : IoT cihazlarının, Bulut Bilişimin kaynaklarının ve bileşenlerinin ayrı olması
 - **Güvenilirlik** : Kullanıcıların istediği sonuçların doğrulanması

Nesnelerin İnterneti için Sis Bilişim

- ◆ Sis Bilişim, Bulut Bilişim ve IOT cihazları arasında yer alır. Köprü görevi görür
- ◆ Sis Bilişimin Nesnelerin İnternetine Sağladığı **Avantajlar**:
 - **Konum**: Sis Bilişim IOT Cihazlara daha yakındır.
 - **Dağıtık**: Küçük parçalar ayrılıp mikro merkezler oluşturulabiliyor.
 - **Ölçeklenebilirlik , Cihaz Yoğunluğu , Mobil Destek**
 - **Gerçek Zamanlı İletişim**: Bulut sisteme göre cevap zamanı daha hızlıdır
 - **Ön Analiz**: Veri boyutu anlamlandırılarak azaltılabilir

NESNELERİN İNTERNETİ İŞ MODELLERİ

✚ **İş prosesi** belirli bir amaca/sonuca ulaşmak için bir şirket içerisindeki belirli bir mantıksal sırada birbiriyle ilişkili proseslerden/aktivitelerden/olaylardan oluşur

✚ **İş modeli**, bir şirketin “değeri” nasıl oluşturup, nasıl dağıttığını ve finansal sürekliliğini kazandığını tanımlayan mantık modelidir

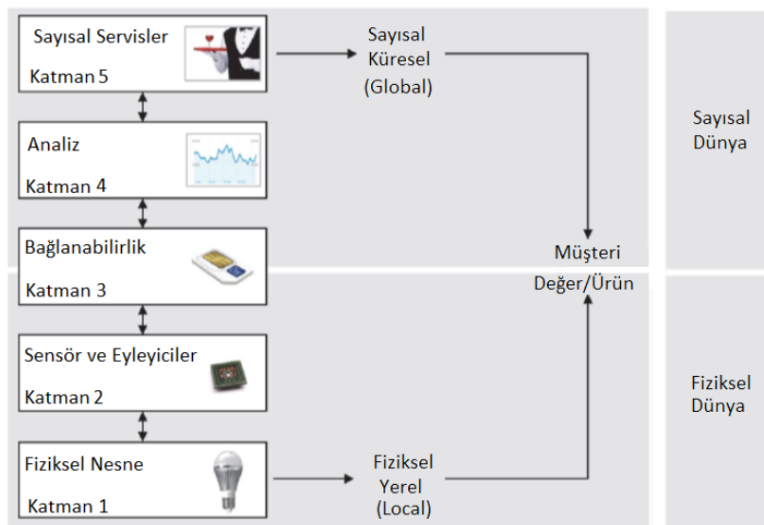
- **İş modeli 4 temel öğeye sahiptir:**
 - Müşteri, Öneri (Satılan ürün/şey), Altyapı ve Finansal Süreklilik
- **İş modeli, 3 temel olay için kullanılır:**
 - Teknoloji ve inovasyon yönetimi
 - Stratejik konular
 - Elektronik işletme (e-business)

İş Modeli Temel Bileşenleri:

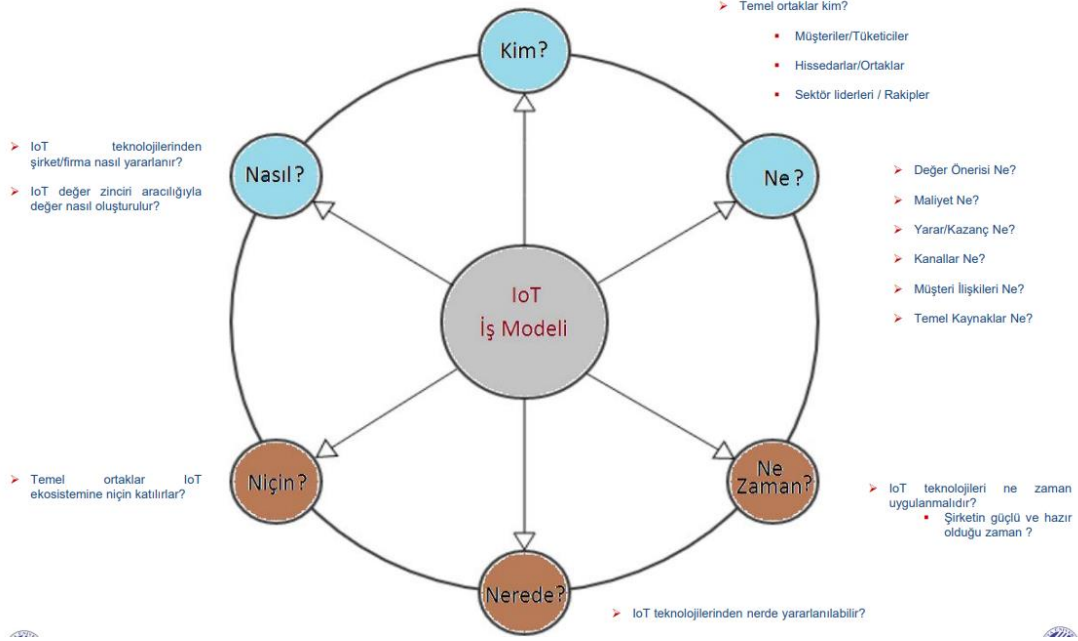
- Müşteri Segmentleri (Customer Segments) , Değer/Ürün Önerisi (Value Propositions)
- Kanallar (Channels) , Müşteri İlişkileri (Customer Relationships) , Gelir Akışı (Revenue Streams)
- Temel/Kilit Kaynaklar (Key Resources) , Temel/Kilit Aktiviteler (Key Activities)
- Temel/Kilit Aktiviteler (Key Activities) , Maliyet Yapısı (Cost Structure)

IoT Çözümlerinde Değer Oluşturma Katmanları

- **Sayısal Servisler:** Uygulama katmanı olarak adlandırılabilir. Kullanıcıya sunulacak uygulamayı (mobil, web vb.) içerir.
- **Analiz:** Kullanıcılardan elde edilen verilerin analizi böylelikle kullanıcı profillerinin çıkarılması vb.
- **Bağlanabilirlik:** IoT donanımlara çevrimiçi (online) erişim. Örneğin bisikletin yerini ve hareket durumunu uzaktan izleme.
- **Sensör ve Eyleyiciler:** Bu katman mikroilemcili sistem, sensör ve eyleyiciler içerir. Yerel olarak çalışan bir sistemi oluşturur. İvmeölçer ile mesafe, GPS ile konum, batarya durumu vb. bilgiler
- **Fiziksel Nesne:** Kullanıcıya yarar sağlayan fiziksel nesne. Örneğin bisiklet, ulaşım, sağlık aktiviteleri gibi faydalar sağlar.



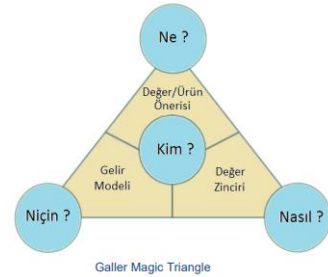
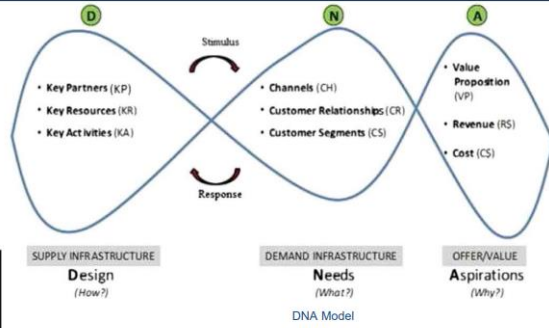
IoT İş Modeli Araştırma/Geliştirme Kriterleri



IoT için Önerilen İş Modelleri

IoT için önerilen iş modelleri

- Business Model Canvas,
- DNA Model,
- Galler Magic Triangle
- Value Net Model,
- MOP Model,



NESNELERİN İNTERNET'İNDE GÜVENLİK

IoT Güvenliğindeki Temel Zorluklar

- IoT nesnelerinin sınırlı kaynaklara sahip olması
- Standart ve yönetmeliklerin eksikliği
- Tasarım süreçlerinde güvenliğin en öncelikli olmaması
- IoT güvenliği alanında uzman eksikliği
- IoT cihazlarına güvenlik güncellemelerinin uzaktan yüklenme zorluğu
- Sorumlulukların/yükümlülüklerinin açıkça belirtilmemesi

OWASP IoT Güvenlik Riskleri

- **OWASP** (Open Web Application Security Project) dünya üzerindeki kurulu olan birçok platformda bulunan zafiyetlerin güncel sürümlerini yayınlamaktadır
- IoT cihazların potansiyel **güvenlik riskleri**
 - Zayıf, tahmin edilebilir veya gömülü kodlanmış şifreler
 - Güvensiz ağ servisler
 - Güvensiz ekosistem arayüzleri
 - Güvenli güncelleme mekanizması eksikliği
 - Cihaz yönetimi eksikliği
 - Güvensiz varsayılan ayarlar
 - Fiziksel güçlendirme eksikliği

IoT uç cihaz ve ağ geçidi seviyesinde saldırı türleri ve riskler

- ◆ **Veri sızıntısı** (data leakage)
- ◆ **Düğüm yayımı bozma** (jamming) , Fiziksel hasar (efficiency)
- ◆ **Kötü niyetli düğüm yazılımı** (malicious node adware)
- ◆ **Hizmet engelleme saldırıları** (DoS)

Bağlantı (connectivity) seviyesinde güvenlik riskleri ve saldırılar (haberleşme ve internet)

- ◆ *Hizmet engelleme saldırıları* (DoS)
- ◆ *Ortakdaki adam saldırısı* (man in the middle attacks)
- ◆ *İletişim güvenliği*
- ◆ *İzinsiz erişim* (unauthorized access)
- ◆ *Sahte ağ mesaj*
- ◆ *Yönlendirme atakları*

IoT bulut seviyesinde güvenlik riskleri ve saldırılar

- ◆ Savunmasız web uygulamaları ve API'ler

IoT uygulaması geliştirirken dikkat edilmesi gereken **birincil güvenlik ilkeleri**

- ◆ Mesaj gizliliği (**confidentiality**)
- ◆ Veri bütünlüğü (**data integrity**)
- ◆ Veri tazeliği/güncelliği (**data freshness**)
- ◆ Verimlilik (**efficiency**)
- ◆ Kendi kendini idare etme (**autonomy**)
- ◆ Kimlik doğrulama (**authentication**)

IoT uygulamalarında güvenliğin dört temel bileşeni

- ◆ IoT nesnesi kimlik doğrulama,
- ◆ Güvenli iletişim bağlantıları,
- ◆ Güvenli yazılımlar
- ◆ Güvenli depolama