SAKARYA ÜNİVERSİTESİ
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BSM 451
NESNELERİN İNTERNETİ VE UYGULAMALARI

(Internet of Things (IoT) and Applications)

NESNELERİN İNTERNETİ UYGULAMA/HABERLEŞME PROTOKOLLERİ
REST, SOAP, XMPP

Doç. Dr. Cüneyt BAYILMIŞ

Nesnelerin İnterneti Uygulama Protokolleri

- Temsili Durum Aktarımı (REpresentational State Transfer, REST)
- Basit Nesne Erişim Protokolü (Simple Object Access Protocol, SOAP)
- Genişletilebilir Mesajlaşma ve Durum Protokolü (Extensible Messaging and Presence Protocol, XMPP)
- Sınırlanmış Uygulama Protokolü (Constrained Application Protocol, CoAP)
- Mesaj Kuyruk Telemetri Ulaştırma (Messega Queue Telemetry Transport, MQTT)
- İleri Mesaj Kuyruklama Protokolü (Advanced Message Queuing Protocol, AMQP)
- Veri Dağıtım Servisi (Data Distribution Service, DDS)

IoT Kullanılan Protokoller

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

Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP REST
Service Discovery		mDNS				DNS-SD		
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN					IPv4/IPv6	
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal	IEEE 802.15.4			Z-Wave	
Influential Protocols		IEEE 1888.3, IPSec					IEEE 1905.1	

IoT Uygulama/Haberleşme Protokollerine Genel Bakış

- Ağ üzerinden birlikte çalışabilen makineden makine etkileşimi/haberleşmeyi desteklemek için tasarlanmış yazılım sistemleri 'Web Servisleri' olarak adlandırılmaktadır.

- Web Servisleri kavramı, IoT konusunda uygulama/haberleşme protokolleri içerisinde ele alınmaktadır.

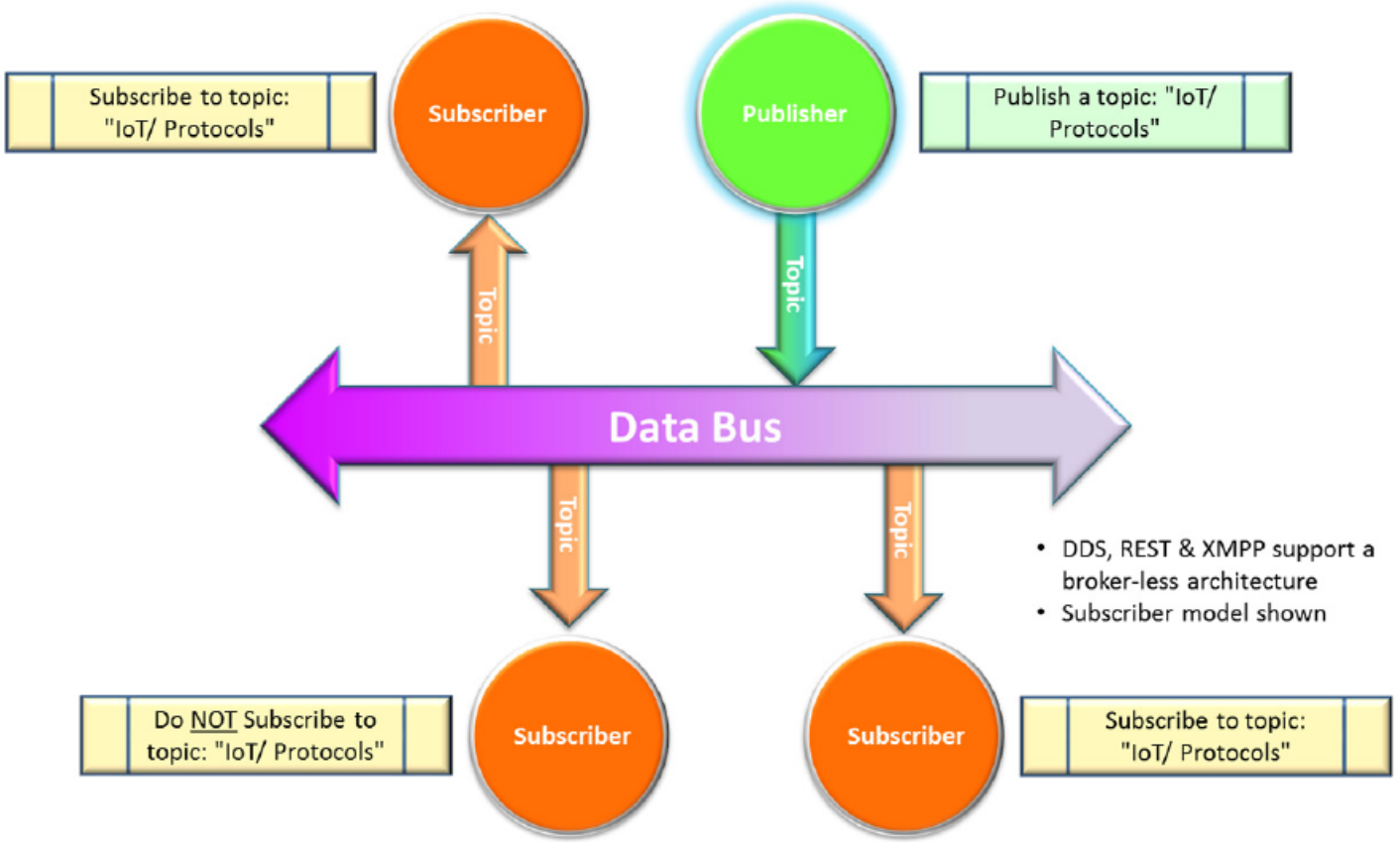
- Uygulama/Haberleşme Protokolü seçiminde kısıtlı donanıma sahip IoT cihazlarının kaynak kullanımı CPU, bellek vb. kaynaklarını efektif kullanabilmesi kriteri gözönünde tutulmalıdır. Bu nedenle IoT uygulamalarının hafif (lightweight) protokoller ile haberleşme ihtiyacı vardır.

IoT Uygulama/Haberleşme Protokollerine Genel Bakış

- IoT uygulama/haberleşme protokollerini iki temel mimariye göre sınıflandırmak mümkündür.
  - Veriyolu Temelli (Bus-based)
    - ✓ İstemciler, belirli bir konu için abonelere doğrudan ulaşan mesajları yayımlar.
    - ✓ Merkezileştirilmiş sunucu ya da sunucu temelli servis yoktur.
    - ✓ DDS, REST, XMPP
  - Sunucu Temelli (Broker-based)
    - ✓ Sunucu bilginin/mesajın dağıtımını kontrol eder.
    - ✓ Sunucu bilgiyi/mesajı depolar, iletir, filtreler, öncelikleri belirler, yayımcıdan (Publisher) istekleri abonelere yayımlar (subscriber) yayımlar.
    - ✓ İstemciler, uygulamanın amaçlarına bağlı olarak yayımcı ve abone rolleri arasında anahtarlar.
    - ✓ AMPQ, CoAP, MQTT, Java Message Service API (JMS)
- IoT uygulama/haberleşme protokolleri mesaj açısından da sınıflandırılabilir.
  - Mesaj Merkezli (message-centric)
    - ✓ Mesajın/verinin içeriğine bakılmaksızın (payload) alıcılarına ulaştırılmasına odaklanır.
    - ✓ AMQP, MQTT, JMS, REST
  - Veri Merkezli (data-centric)
    - ✓ Mesajın/verinin alıcı tarafından anlaşıldığını varsayar ve verinin ulaştırılmasına odaklanır
    - ✓ DDS, CoAP, XMPP

IoT Uygulama/Haberleşme Protokollerine Genel Bakış

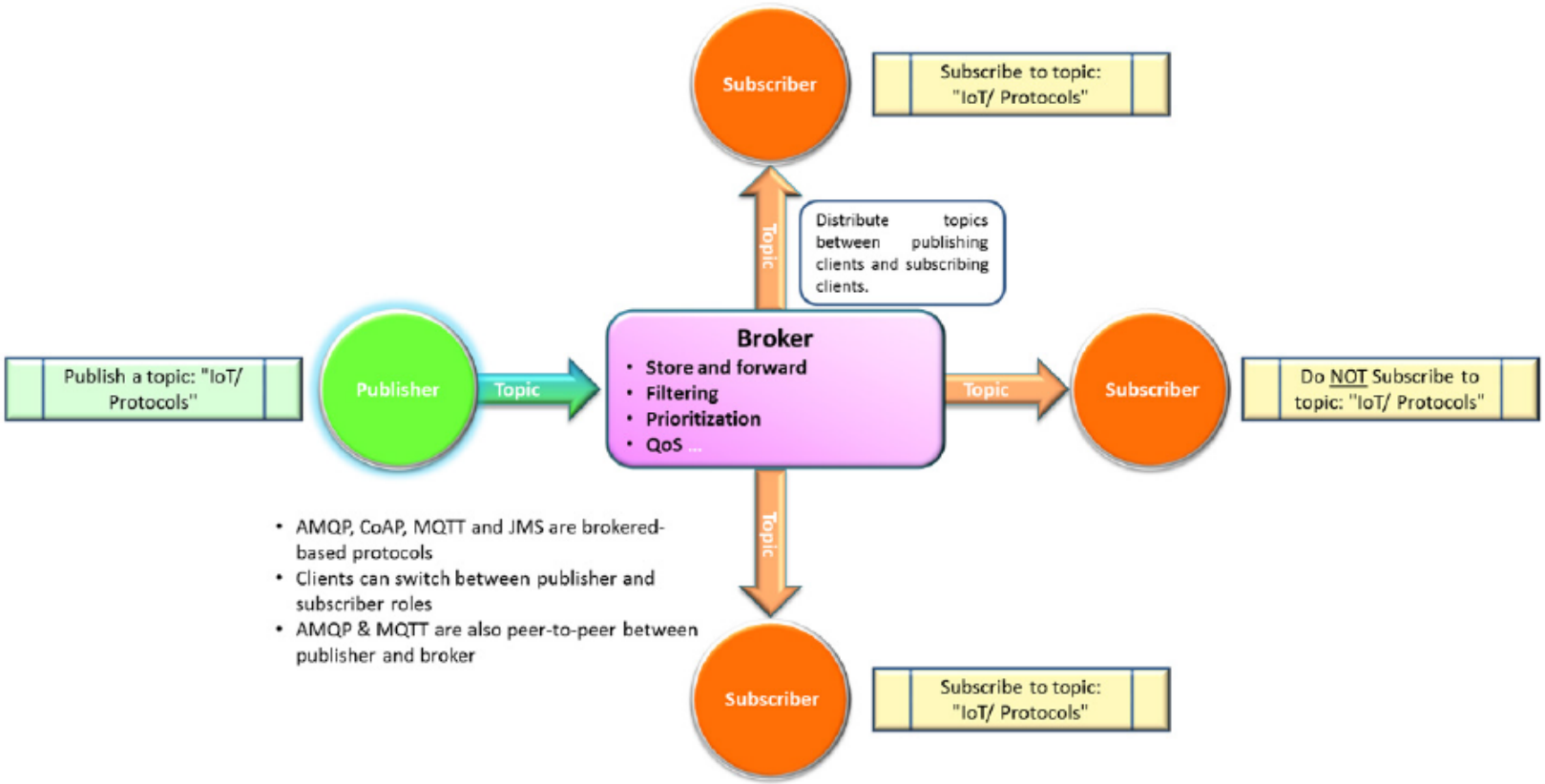
- Veriyolu Temelli (Bus-based)



Kaynak: O. Vermesan, P. Friess (Editors), ''Internet of Things – From Research and Innovation to Market Deployment'', River Publishers, 2014.

IoT Uygulama/Haberleşme Protokollerine Genel Bakış

- Sunucu Temelli (Broker-based)



Kaynak: O. Vermesan, P. Friess (Editors), ''Internet of Things – From Research and Innovation to Market Deployment'', River Publishers, 2014.

Bilgi Kaynakları (XML ve JSON)

- XML yapısında veri tanımlamalarında başlık etiketleri kullanılır.
- JSON yapısında ise veri tanımlamaları noktalama işaretleri ile yapılır.
- JSON veri boyutu olarak XML'den daha azdır.

XML örneği

```
<?xml version="1.0" Encoding=="UTF-8">
<sensor>
   <_class>cihaz.Model </_class>
   <cihazID> 4e22c934e7</cihazID>
   <cihazAdi>Termometre </cihazAdi>
   <sicaklik> 23.00</sicaklik>
</sensor>
```

JSON örneği

```
{
      "_class" : "cihaz.Model"
       "cihazID" : " 4e22c934e7 "
       "cihazAdi : " Termometre "
       "sicaklik" : " 23.00 "
}
```

Temsili Durum Transferi Protokolü

(REpresentational State Transfer, REST)

Temsili Durum Transferi
(REpresentational State Transfer, REST)

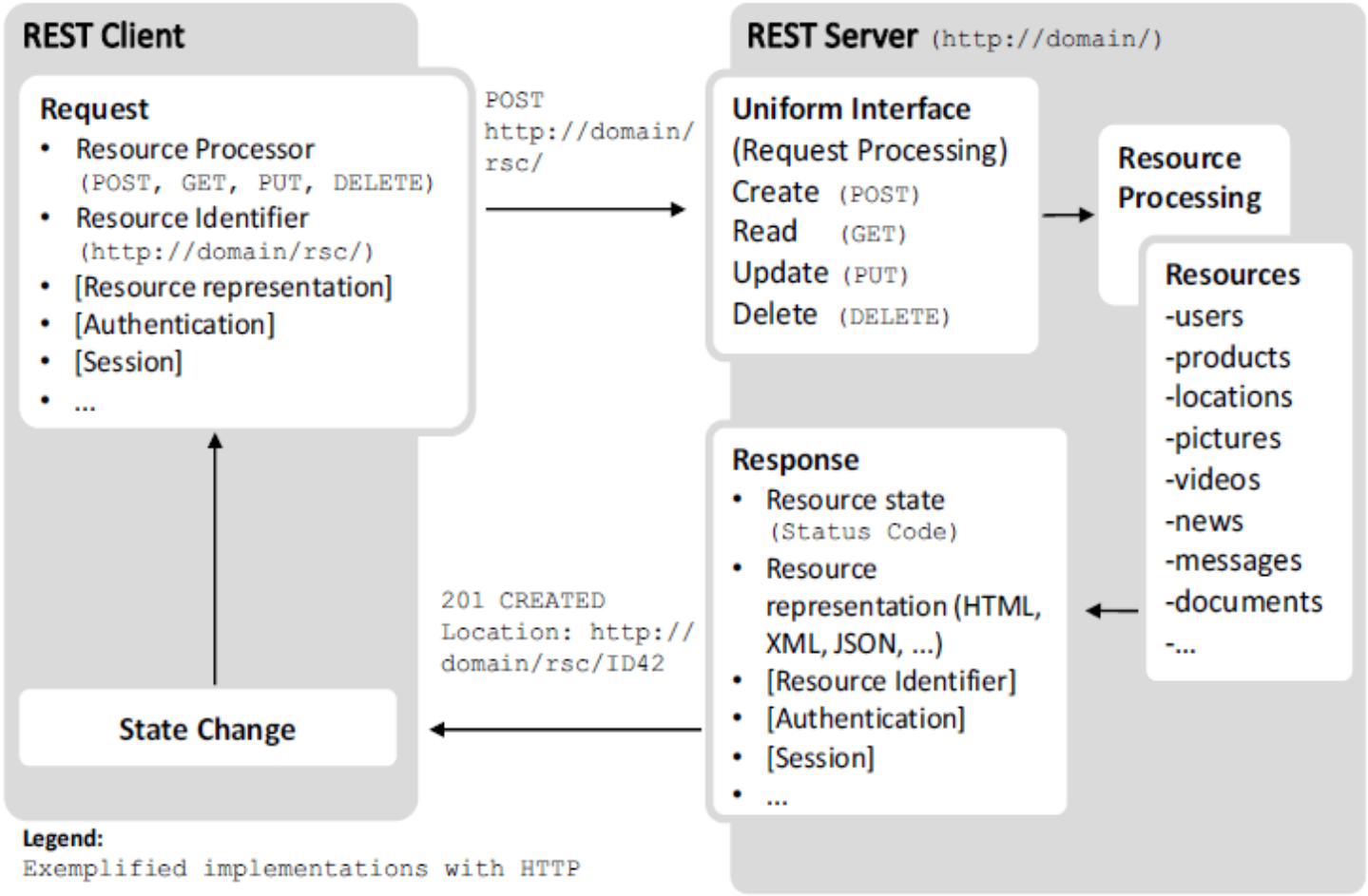
- REST, makineler arasında bağlantı kurmak için basit HTTP kullanan ağ uygulamalarının tasarımı için işletim sistemi bağımsız bir mimari ve dildir.
- R.T. Fielding'in doktora çalışmalarında ağ sistemlerine yönelik bir mimari stil tanımlamak için kullanılmış bir deyimdir.
- REST bir standart değildir, 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.
- Basit, esnek ve kolay genişletilebilir olduğundan dolayı web tabanlı uygulamalarda yaygın olarak kullanılır.
- Diğer web servisler ile yapılabilecek her şey RESTful servislerle yapılabilir.

Temsili Durum Transferi
(REpresentational State Transfer, REST)

- HTTP üzerinden POST, GET, DELETE metotları ile iletişimi gerçekleştirir.
- Platform bağımsızdır (Windows, Linux).
- Programlama dili bağımsızdır (C#, JAVA, vb.).
- Ulaşım katmanı olarak TCP kullanır.
- Servis kalitesi desteği yoktur.
- 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.

Temsili Durum Transferi
(REpresentational State Transfer, REST)

- REST Web Servis Mimarisi



Kaynak: P. L. Gorski, L. L. Iacono, H. V. Nguyen, D. B. Torkian, "Service Security Revisited", IEEE International Conference on Services Computing, 464-471, 2014

Temsili Durum Transferi
(REpresentational State Transfer, REST)

REST Web Servisinin Özellikleri ve Kısıtlamaları (Tasarım İlkeleri)

- İstemci – Sunucu (Client – Server)
  - Temel tasarım (ilk kısıtlama) istemci – sunucu mimarisidir.
  - İstemci (kullanıcı arayüzleri) ve sunucu (veri saklama birimleri) yapısının birbirinden ayrı olması anlamına gelir.
  - İstemci ve sunucunun ayrılığı, platformdan bağımsız çalışma (kullanıcı arayüzünün taşınabilirliğini) ve sunucu elemanlarının basitleştirilmesiyle ölçeklendirilebilmenin (kapsamın) artmasını sağlar.
  - 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.
  - Sistem herhangi bir durumda olmadığından istemcilerden herhangi bir zamanda gelen istekler sunucu tarafından uygun yanıt içerikleri ile dönecektir.
  - Sunucuda istemci ile ilgili bir bilgi ya da oturum (session) tutulmaz. İstemci tarafından yapılan her istek sunucu için gerekli bilgiyi taşır.

Temsili Durum Transferi
(REpresentational State Transfer, REST)

REST Web Servisinin Özellikleri ve Kısıtlamaları (Tasarım İlkeleri)

- Ön Bellekleme (Cache)
  - Ağ performansını arttırır. Kullanıcıya daha hızlı cevap verilmesini sağlar.
  - 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.
  - Uygulamaların birbirinden bağımsız geliştirilmesine imkan sağlar.
  - Ortak arayüz kullanmak uygulamaların özel gereksinimlerini dikkate almak yerine belirli bir standart sağladığından verimliliği düşürebilir ancak birlikte çalışabilir uygulama geliştirmeyi teşvik eder.
  - REST arayüzü, geniş bant iletişim için tasarlanmıştır.

Temsili Durum Transferi
(REpresentational State Transfer, REST)

REST Web Servisinin Özellikleri ve Kısıtlamaları (Tasarım İlkeleri)

- Katmanlı Sistem (Layered System)
  - İstemci her durumda doğrudan son sunucuya bağlanmayabilir, arada başka sunucular olabilir.
  - Katmanlı sistem ile hiyerarşik bir mimari sağlanır ve her katman yalnızca komşu (alt ya da üst) katmanı bilmektedir.
  - Katmanlı sistemin dezavantajı ek gecikmelere neden olmasıdır.

- İstek Durumunda Kod (Code-On-Demand)
  - İsteğe bağlı bir kısıtlamadır.
  - İhtiyaç durumunda istemci bazı kaynaklara ulaşabilir ancak bu kaynakları nasıl kullanacağını bilmez.
  - Sunucu belirli durumlarda istemci tarafına çalıştırılabilir scriptler ve uygulamalar gönderebilir.

Temsili Durum Transferi
(REpresentational State Transfer, REST)

RESTful Web Servislerinde Kullanılan HTTP Metotları

- PUT
  - Tek biçimlilikteki (uniform interface) oluşturma (create) eylemini, SQL veritabanı uygulamalarında Insert işlemini gerçekleştirir.
  - Yeni bir kaynak oluşturmak için kullanılır.
  - Web Servisi, başarı ya da hatalı yanıtı verir.

- GET
  - Tek biçimlilikteki okuma (read) eylemini, SQL veritabanı uygulamalarında Select işlemini gerçekleştirir
  - Bir kaynağa sorgu yapmak ya da bir kaynaktan veri almak amaçlı kullanılır.
  - Kaynaktan kullanıcıya geri dönen veri, istenilen kaynağın temsilidir.

- POST
  - Tek biçimlilikte ve SQL veri tabanı uygulamalarında güncelle (update) eylemini gerçekleştirir.
  - Mevcut bir kaynağın ya da verinin güncellenmesini sağlar.
  - Ayrıca yeni bir kaynak oluşturmak ve mevcut bir kaynağı silmek için de kullanılabilir.

- DELETE
  - Tek biçimlilikte ve SQL veri tabanı uygulamalarında silme (delete) eylemini gerçekleştirir
  - Mevcut bir kaynağın ya da verinin silinmesini sağlar.

Basit Nesne Erişim Protokolü

(Simple Object Access Protocol, SOAP)

Basit Nesne Erişim Protokolü
(Simple Object Access Protocol, SOAP)

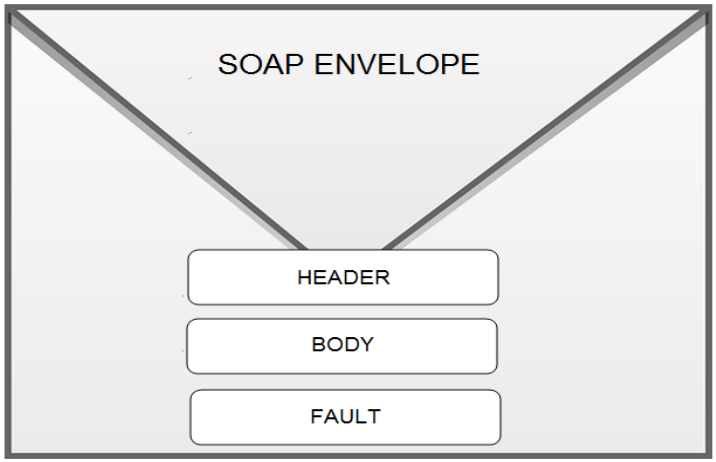
- REST mimarisi ile benzerliklere sahiptir.
- SOAP, merkezileştirilmemiş, dağıtık ortamlar tarafından yapısal bilgi değişimi için tasarlanmıştır.
- Platform bağımsız genişletilebilir biçim sağlamak için bilgi gönderimi ve tanımlanmasında XML kullanır. XML mesajlaşması için HTTP protokolünü genişletebilir.
- Yayın (broadcast) mesajları için kullanılabilir.
- Uzak İşlev Çağırma (Remote Procedure Call, RPC) modelini kullanan İstemci – Sunucu mimarisine dayalı bir protokoldür.
- İnternet Komitesi W3C (World Wide Web Consortium) standardıdır.
- Web Servisleri için veri iletişimi sağlar.

Basit Nesne Erişim Protokolü
(Simple Object Access Protocol, SOAP)

- 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 kuralarını içerir.
- SOAP RPC Sunum (RPC Representation)
  - Uzaktan prosedür çağırımı ve geri dönüşlerini tanımlar.
- Protokol Bağlantısı (Protocol Binding)
  - SOAP'ın HTTP içerisinde nasıl kullanılacağını tanımlar.
  - Bir diğer değişle mesaj değişimi için bir bağlanma konvasiyonu

Basit Nesne Erişim Protokolü
(Simple Object Access Protocol, SOAP)

- SOAP zarfı olarak adlandırılan mesaj yapısı, Başlık (Header), Gövde (Body) ve Hata (Fault) alanlarından oluşur.
  - Zarf, SOAP mesajın bütününü oluşturur.
  - Başlık, mesaj sağlayıcılarının kullanacağı isteğe bağlı özeliklerinin ayarlandığı verilerden oluşur.
  - Gövde, asıl mesajı oluşturur ve XML veriyi içerir.
  - Hata, opsiyonel olup fonksiyonda oluşan hatayı içerir.




```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
  <m:Trans xmlns:m="http://www.example.com/transaction/">100 </m:Trans>
</soap:Header>

<soap:Body>
  <m:GetName xmlns:m="http://www.example.com/customers">    <m:Item>1234</m:Item>
    </m:GetName>
  </soap:Body>

</soap:Envelope>
```

RESTful ile SOAP Karşılaştırılması

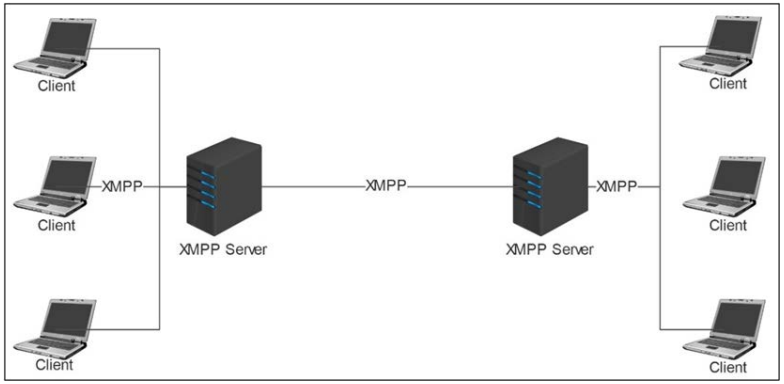
- RESTful, ön bellekleme ile gecikmeleri azaltmakta ve hızı arttırmaktadır. Ayrıca SOAP zarf yapısını kullanmadığından RESTful daha düşük veri boyutları kullanmaktadır.
- RESTful sistemler, SOAP'a göre sunucu ölçeklendirilmesinde daha gelişmişlerdir. Bu RESTful'un oturum durumlarına erişmeye daha az ihtiyaç duymasından kaynaklanır.
- RESTful'da, tarayıcı programlar ile uygulama ve kaynaklara ulaşılabildiğinden, diğer servisler ile karşılaştırıldığından istemci kısmında daha az yazılım çalıştırılmasına imkan vermektedir.
- Yazılımcı desteği olarak SOAP daha eski bir servis olduğundan kurumsal olarak daha geniş alanlarda kullanılmaktadır. RESTful'un kullanımı ise gün geçtikçe artmaktadır.
- RESTful, güncel internet standartlarını kullanmakta iken, SOAP tabanlı uygulamalar/yaklaşımlar kendi standartlarını içermektedirler.
- SOAP servisleri, sistemler arası uyumluluk uygulamaları için tercih edilirler.
- SOAP servislerin, güvenlik, kayıt ekleme ve senkron, asenkron, istek/geri çağırma, uyarı gibi bazı iletişim modelleri sunduğundan geliştirilmesi ve tasarımı daha komplekstir.

Genişletilebilir Mesajlaşma ve Durum Protokolü

(Extensible Messaging and Presence Protocol, XMPP)

Genişletilebilir Mesajlaşma ve Durum Protokolü
(Extensible Messaging and Presence Protocol, XMPP)

- XMPP, genişletilebilir işaretleme dili (XML) tabanında, gerçek zamanlı sayılabilecek, mesajlaşma, durum yönetimi ve istek-yanıt hizmetleri için geliştirilmiş bir uygulama protokolüdür.
- XMPP, Internet Engineering Task Force (IETF) tarafından yayımlanan RFC 3920 ve RFC 3921 ile tanımlanmıştır.
- Jeremie Miller tarafından 1999 yılında anında mesajlaşma hizmeti için ortaya konulmuştur. Başlangıçta Jabber olarak bilinir.
- XMPP, gerek iki sunucu, gerekse iki istemci arasında, P2P (Peer-to-Peer) haberleşme olanağı sağlayabilen ve birbirlerine bağlanabilen servisler arasında otonom ağlar oluşturabilmektedir.
- 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.
- Her XMPP kimliğinde bulunması gereken alan adı alanı, tüm XMPP ağı içerisinde o varlığa ait alt ağı tanımlamak için kullanılır. Kullanıcı adı ve alan adı alanının bileşkesiyle (kullanıcı@alanadı.com) elde edilen jid yapısı, çıplak (bare) jid olarak isimlendirilir.
- Kaynak bildirimi ise, bir kullanıcının farklı istemciler üzerinden bağlantı kurabildiği durumlarda hangi istemcinin hangi kaynağı kullandığını tanımlamak için kullanılmaktadır.
- Bir jabber kimliğini oluşturan bu üç bileşenin yer aldığı adres yapısı tam (full) jid olarak isimlendirilir. 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 şu şekildedir:

1. Sunucu ile uzun ömürlü TCP bağlantının kurulması,
2. XML akışlarının başlatılması,
3. Çeşitli XML şemalarının paylaşımı,
4. Sunucu üzerinde istemci kimliğinin doğrulanması,
5. Kaynak seçiminin sağlanması,
6. Anında mesajlaşma süreci öncesi kullanıcı listelerinin alınması ve ilk durumun sunucuya bildirimi şeklinde gerçekleşir.

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'de XML paket formatı 3 temel tanımlama alanına sahiptir.
  - mesaj (<message>),

    kullanıcılar (istemci) arasında her türlü bilgiyi aktarmak için kullanılabilir.
  - durum (<presence>),

    bir kullanıcının XMPP ağında varlığını kontrol eder ve raporlar. Kısaca XMPP ağındaki bir kullanıcının çevrimiçi, meşgul, çevrimdışı gibi durumlarını yönetmek için kullanılan paket yapısıdır. Böylece kullanıcıların birbirlerinin erişilebilirliğini kontrol edebilmeleri sağlanır.
  - bilgi/sorgu (<iq>)

    istemci-sunucu arasında iletilmek istenen bir verinin, mesaj ya da durum paket yapılarına uymadığı durumlarda bilgi/sorgu paketi olarak kullanılması için tasarlanmıştır. Bilgi/sorgu paketleri genel olarak istek/cevap paketi olarak konumlandırılmıştır. Bu paket yapısı özelleştirilerek, protokolün genişletilebilirliği sağlanır. Bu paket yapısı özellikle HTTP mimarisinde kullanılan GET, POST, PUT vb. yöntemlerinde olduğu gibi istek-yanıt etkileşimini sağlamaya yönelik iş akışları için bir yapı sağlamaktadır.
- Bu paket yapılarının her biri sunucu tarafından farklı yönlendirildiği gibi istemciler tarafından da farklı işlenir.

XMPP Paket Yapıları

- XMPP paket yapıları, alıcılar tarafından paketin nasıl işleneceğini belirleyen öznitelik:değer tanımlamalarından oluşur.
- XMPP paket yapıları 5 farklı özniteliğe sahiptir.
  - Kime (to) özniteliği, ilgili paketinin alıcısını tanımlayan JID'dir.
  - Kimden (from) özniteliği, ilgili paketin kim tarafından gönderildiğini tanımlayan JID'dir. Bir istemci kimden özniteliği olmayan bir paket alırsa bu paketin istemcinin bağlı olduğu sunucudan geldiğini kabul eder. Ancak sunucuya gelen bir XMPP paketinde kimden alanı geçersiz yada tanımlanmamışsa bu durumda sunucu istemciye "geçersiz kimden" (<invalid-from/>) cevabı dönecektir.
  - Kimlik (id) özniteliği, XMPP paketlerinde oluşan cevap paketlerinin eşleştirilmesinde/kimliklendirilmesinde yardımcı olmak için bilgi/sorgu (iq) paketi hariç seçimli olarak kullanılır.
  - Tip (type) özniteliği, mesaj (<message>), durum (<presence>) ve bilgi/sorgu (<iq>) paketlerinin içeriği yada amacı hakkında bilgiler ifade etmektedir. Üç tip paket yapısı için ortak olan tip özniteliği hata (error) değeridir. Tip (type) özniteliğinin alabileceği değerler bu üç paket yapısı için ayrı ayrı özelleştirilmiş anlamlar ifade etmektedir
  - XML dil (xml:lang) özniteliği, XML karakterlerinin varsayılan dilini ifade eder. XML dil tanımı, alt etiketler tarafından değiştirilebilir. Bu özniteliğin hiç tanımlanmadığı durumlarda sunucunun varsayılan dil tanımı kullanılır.

KAYNAKLAR

- A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "*Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*", IEEE Communication Survey&Tutorials, vol. 17 (4), 2347-2376 ,2015.
- L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A Survey", Computer Networks, vol. 54, 2787-2805, 2010.
- O. Vermesan, P. Friess (Editors), "Internet of Things – From Research and Innovation to Market Deployment", River Publishers, 2014.
- R.T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures". PhD Thesis, University of California; 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
- P. L. Gorski, L. L. Iacono, H. V. Nguyen, D. B. Torkian, "Service Security Revisited", IEEE International Conference on Services Computing, 464-471, 2014.
- A. N. Çiçek, "RESTful Web Servisleri ile E-Sağlık Sistemleri Gerçekleştirimi", Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enst. 2009.
- Halil Arslan, Doktora Tezi, SAÜ Fen Bilimleri Enstitüsü, 2016