SAKARYA ÜNİVERSİTESİ
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BSM 313
NESNELERİN İNTERNETİ VE UYGULAMALARI

(Internet of Things (IoT) and Applications)

NESNELERİN İNTERNETİ UYGULAMA/HABERLEŞME PROTOKOLLERİ
CoAP, MQTT, AMQP, DDS

Doç. Dr. Cüneyt BAYILMIŞ

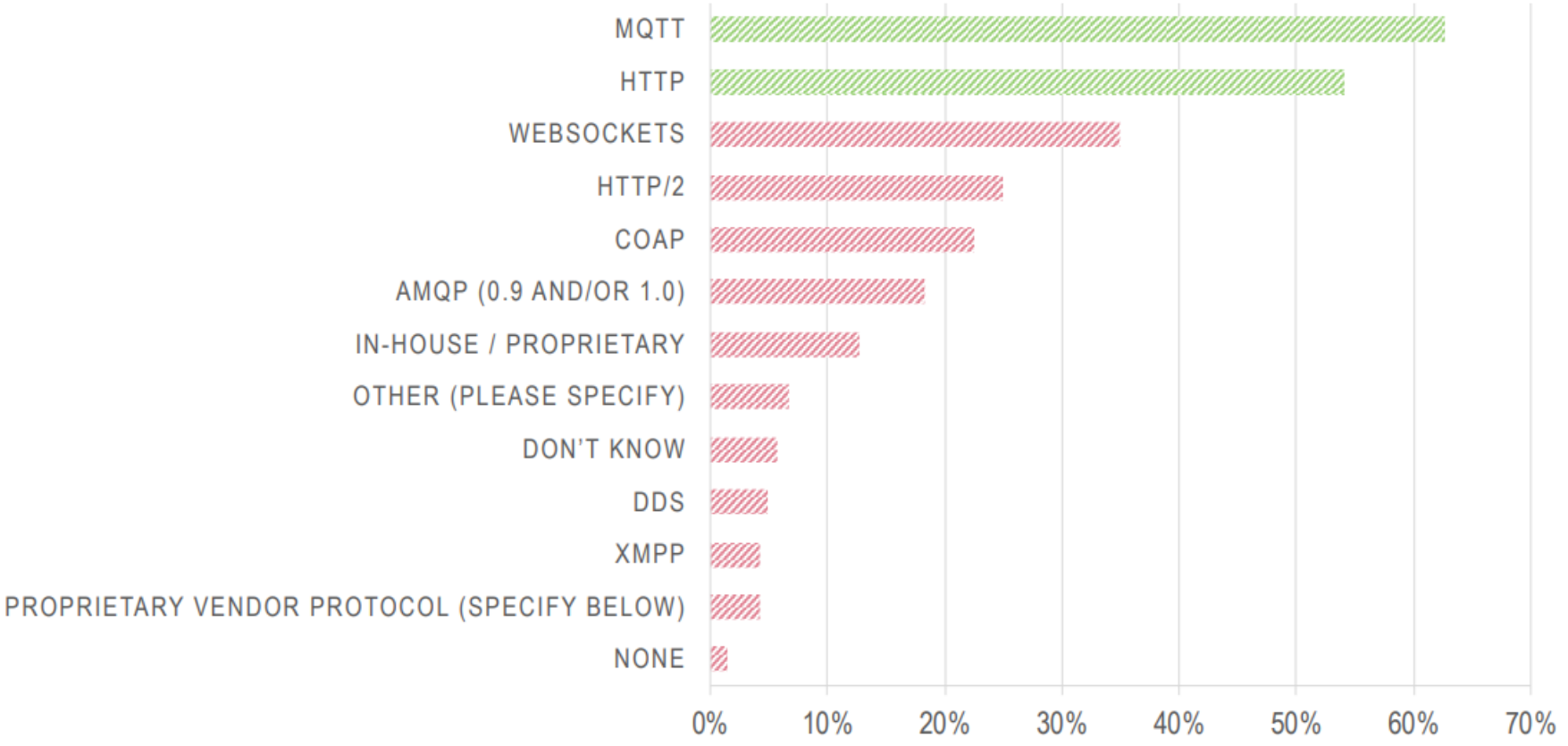
Nesnelerin İnterneti Uygulama Protokolleri

- 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)
- WebSocket

Nesnelerin İnterneti Uygulama Protokolleri

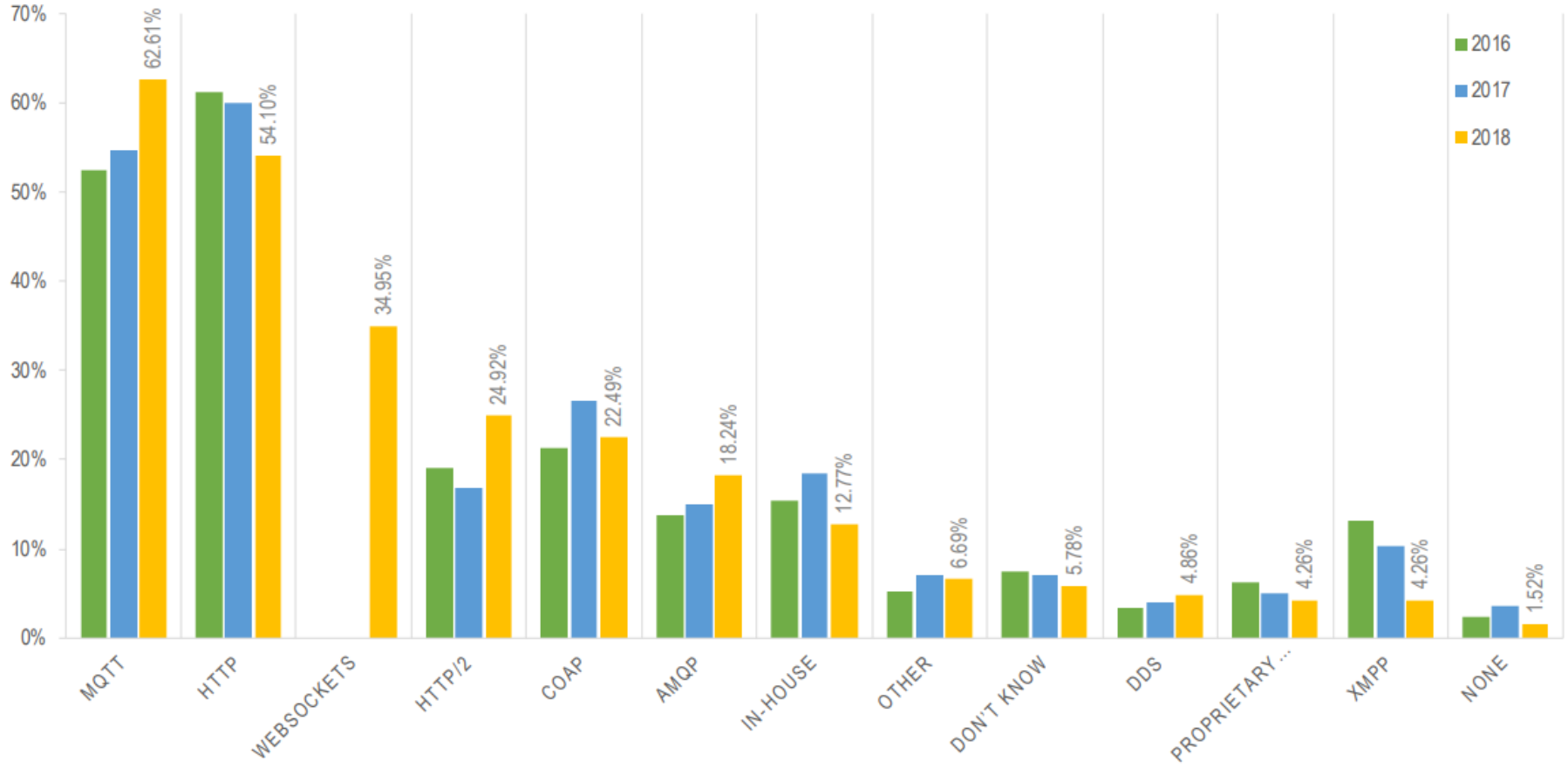
MESSAGING STANDARDS

What messaging protocol(s) do you use for your IoT solution?



Nesnelerin İnterneti Uygulama Protokolleri

MESSAGING STANDARDS - TRENDS



Sınırlanmış Uygulama Protokolü

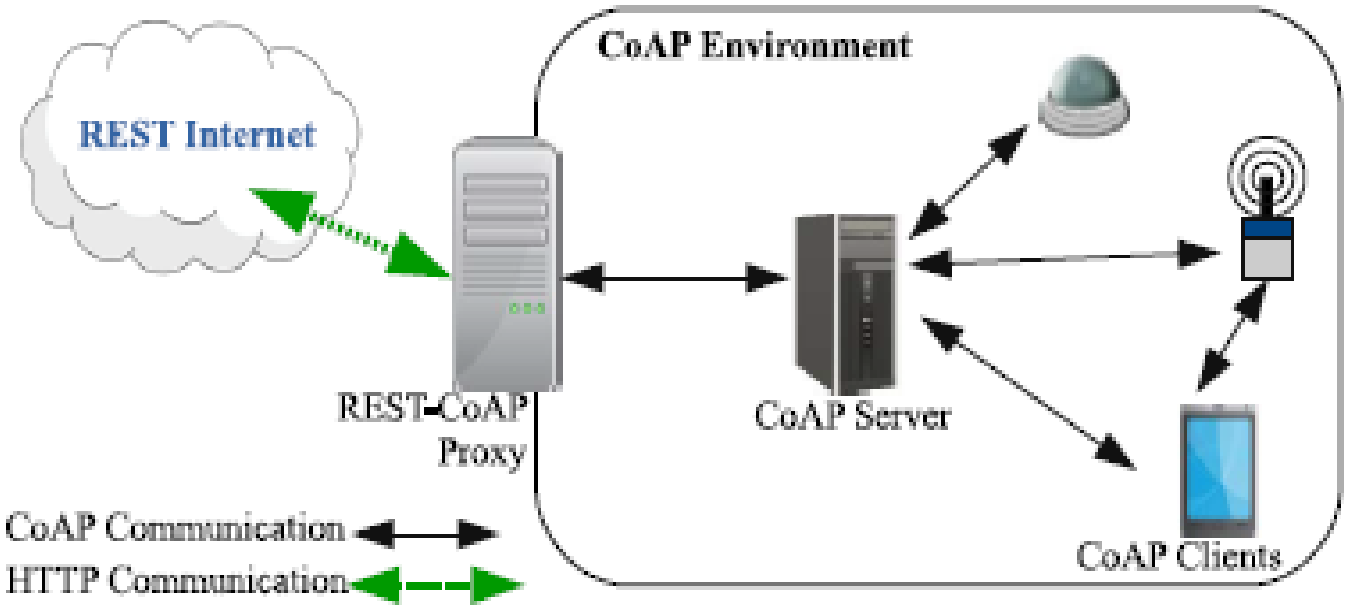
(Constrained Application Protocol, CoAP)

Sınırlanmış Uygulama Protokolü
(Constrained Application Protocol, CoAP)

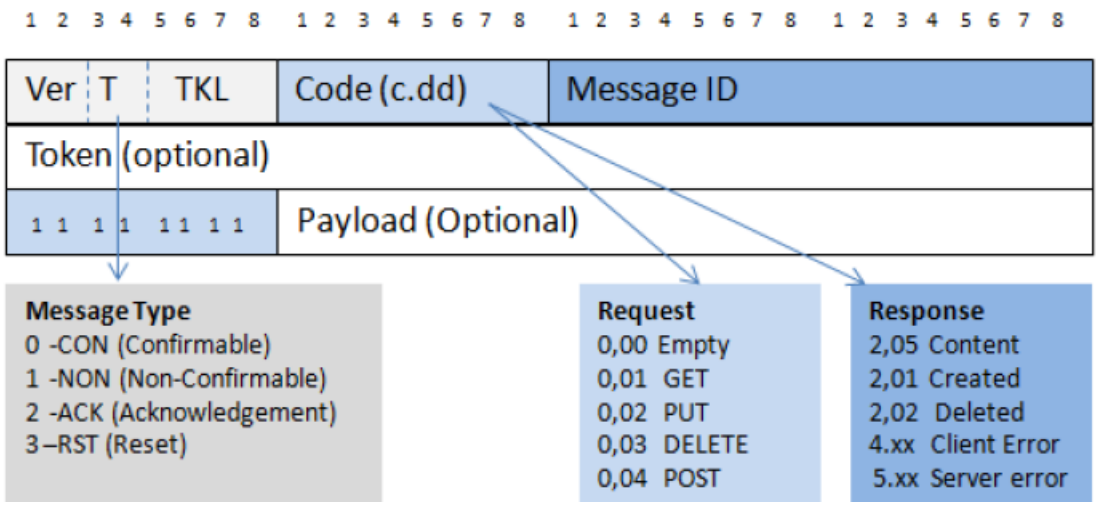
- CoAP, İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force, IETF) ve ARM tarafından birlikte gerçekleştirilmiştir.
- Haziran 2014'te RFC7252 olarak standartlaştırılmıştır.
- CoAP, nesnelerin interneti (makineler arası haberleşme) uygulamaları için tasarlanmış, güç, işlem kapasitesi, bellek 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.
  - HTTP'deki GET, PUT, POST, DELETE metotları kullanılır.
- CoAP, UDP üzerinden kullanılan düşük ağırlıklı (lightweight) bir protokoldür.
  - CoAP paketleri, HTTP TCP akışlarından çok daha küçüktür. UDP paket başlığı olarak sadece 8 bayt kullanır. 4 bayt, uygulama başlığı olarak CoAP tarafından eklenir.
  - İstemci ve sunucular bağlantısız datagramlar aracılığıyla haberleşir. Datagram temelli bir protokol olduğundan diğer paket temelli haberleşme protokollerinde kullanılabilir.
  - CoAP, adresleme için UDP yayın (broadcast) ve çoklu yayın (multicast) mekanizmalarına izin verir.

Sınırlanmış Uygulama Protokolü
(Constrained Application Protocol, CoAP)

- CoAP, çoklu yayın haberleşme desteği nedeniyle yayımcı/abone (Publisher/subscriber) mimarisini destekler. Asenkron mesaj değişimlerine sahip CoAP, IoT için özelleştirilmiştir.
- CoAP güvenli iletişimi Datagram Transport Layer Security (DTLS) veya Transport Layer Security (TLS) benzeri datagram kullanımı ile sağlar.
- CoAP mimarisi

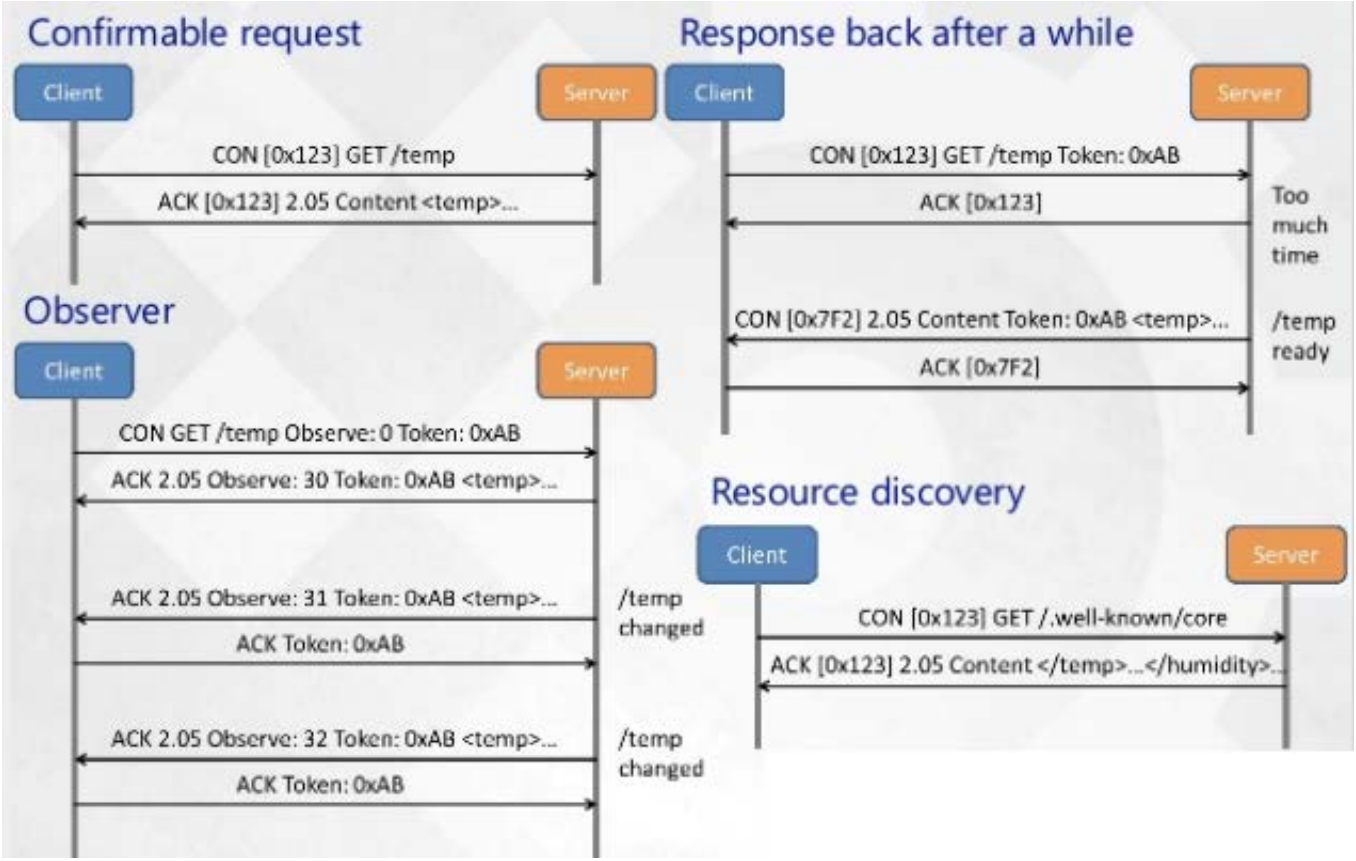


CoAP Mesaj Yapısı



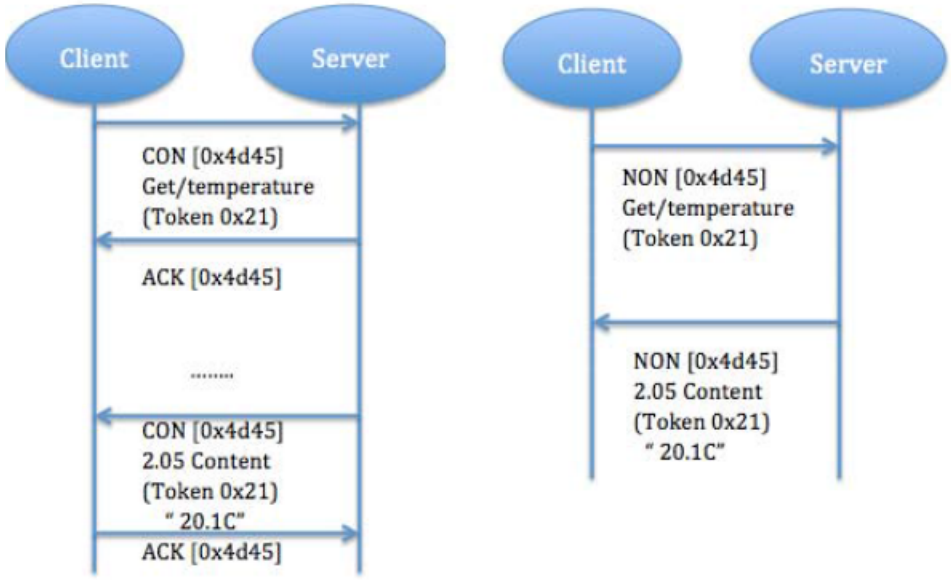
- Ver: CoAP protokol versiyon (2 bit),
- T: İşlem Tipi (Transaction Type, 2 bit)
- TKL (Token Length): 0 ile 8 bayt arasında Jeton'un boyutunu belirtir.
- Code, istemci ve sunucu arasındaki iletişimin sonuçlarını gösterir..
- Message ID: Güvenli haberleşmeyi sağlamak ve mesaj kopyalamalarını tespit için mesaj tanıtıcı olarak kullanılır.
- Token: İstek ve Yanıtlar, token-id ile birbirlerini adresler (eşleştirir).
- Payload: Yük

CoAP İstek/Yanıt Protokol Örneği



- Onaylı (Confirmable, CON) haberleşme seçildiğinde, işlemlerin tamamlandığına (mesajların alındığına dair) ACK paketi alınır/gönderilir.
- Onaysız (Non-Confirmable, NON) haberleşme seçildiğinde ise ACK kullanılmaz.
- İstek ve Yanıt arasındaki eşleştirmeler Mesaj ID ve Token-ID ile yapılmaktadır.

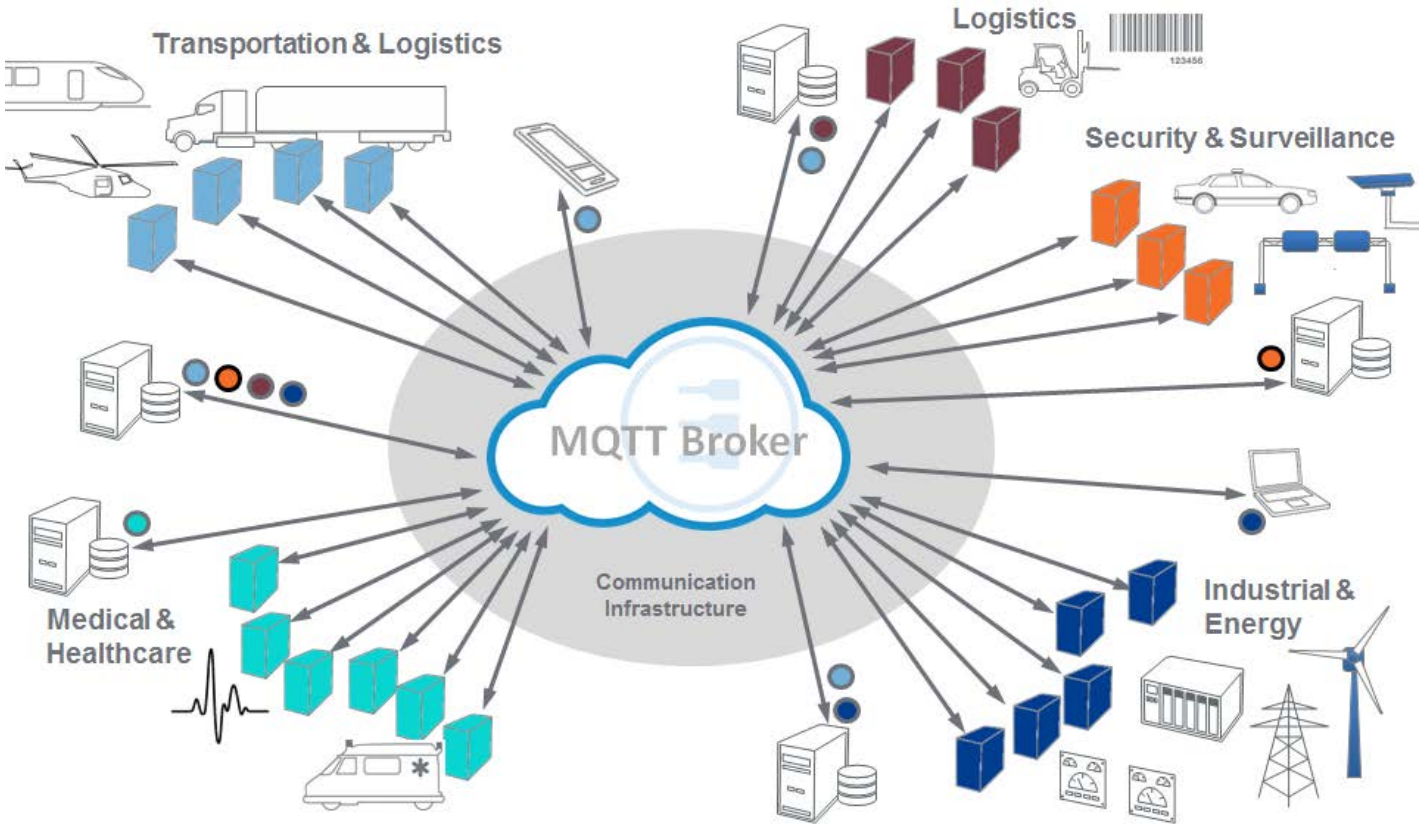
CoAP İstek/Yanıt Protokol Örneği



- Onaylı (Confirmable, CON) haberleşme seçildiğinde, işlemlerin tamamlandığına (mesajların alındığına dair) ACK paketi alınır/gönderilir.
- Onaysız (Non-Confirmable, NON) haberleşme seçildiğinde ise ACK kullanılmaz.
- İstek ve Yanıt arasındaki eşleştirmeler Mesaj ID (0x4d45) ve Token-ID (0x21) ile yapılmaktadır.

Mesaj Kuyruk Telemetri Ulaştırma

(Messega Queue Telemetry Transport, MQTT)

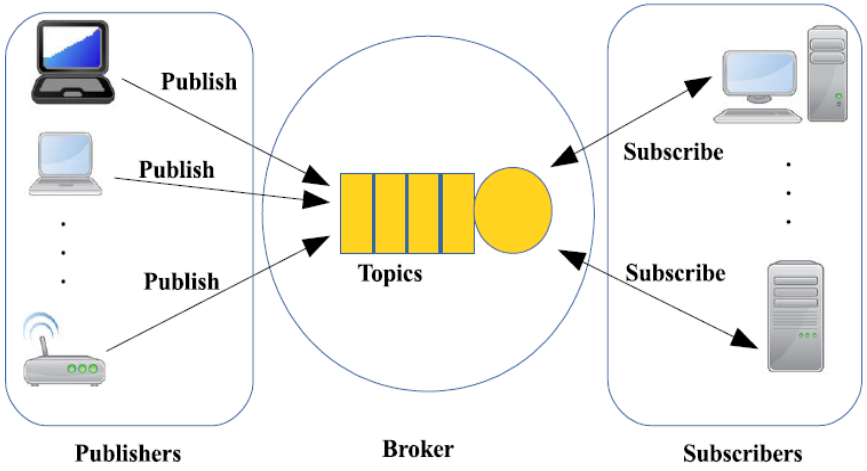


Messega Queue Telemetry Transport (MQTT)

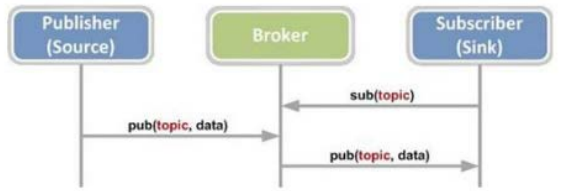
- MQTT, nesnelerin interneti ya da makineler arası haberleşmede kullanılan mesajlaşma protokolüdür.
- 1999 yılında IBM ve Arcom (bugün Eurotech) tarafından tanıtılmış ancak 2013'te OASIS standartlaştırmıştır.
- MQTT, gömülü sistemler ve network (bulut) bağlantısını sağlamayı amaçlamaktadır. Diğer bir değişle küçük cihazlardan oluşan ağları bulut üzerinden kontrol etmeyi ve izlemeyi sağlar.
- MQTT, yayınlama (publisher) ve abone (subscriber) olma mantığına dayanan telemetri mesajlaşma protokolüdür.
- MQTT protokolünün öne çıkan özellikleri arasında mesajları saklayabilme ve bağlantı kurulduğunda tekrar gönderebilme yeteneğidir.
- MQTT, TCP protokolünün üzerine kurulmuştur.
- MQTT mesajları 3 farklı servis kalitesi seviyesine (QoS) göre iletilmektedir. Best Effort, At Least Once, Exactly Once
- MQTT'in iki türü vardır:
  - MQTT v3.1
  - MQTT-SN (MQTT for Sensor Network)

Messega Queue Telemetry Transport (MQTT)

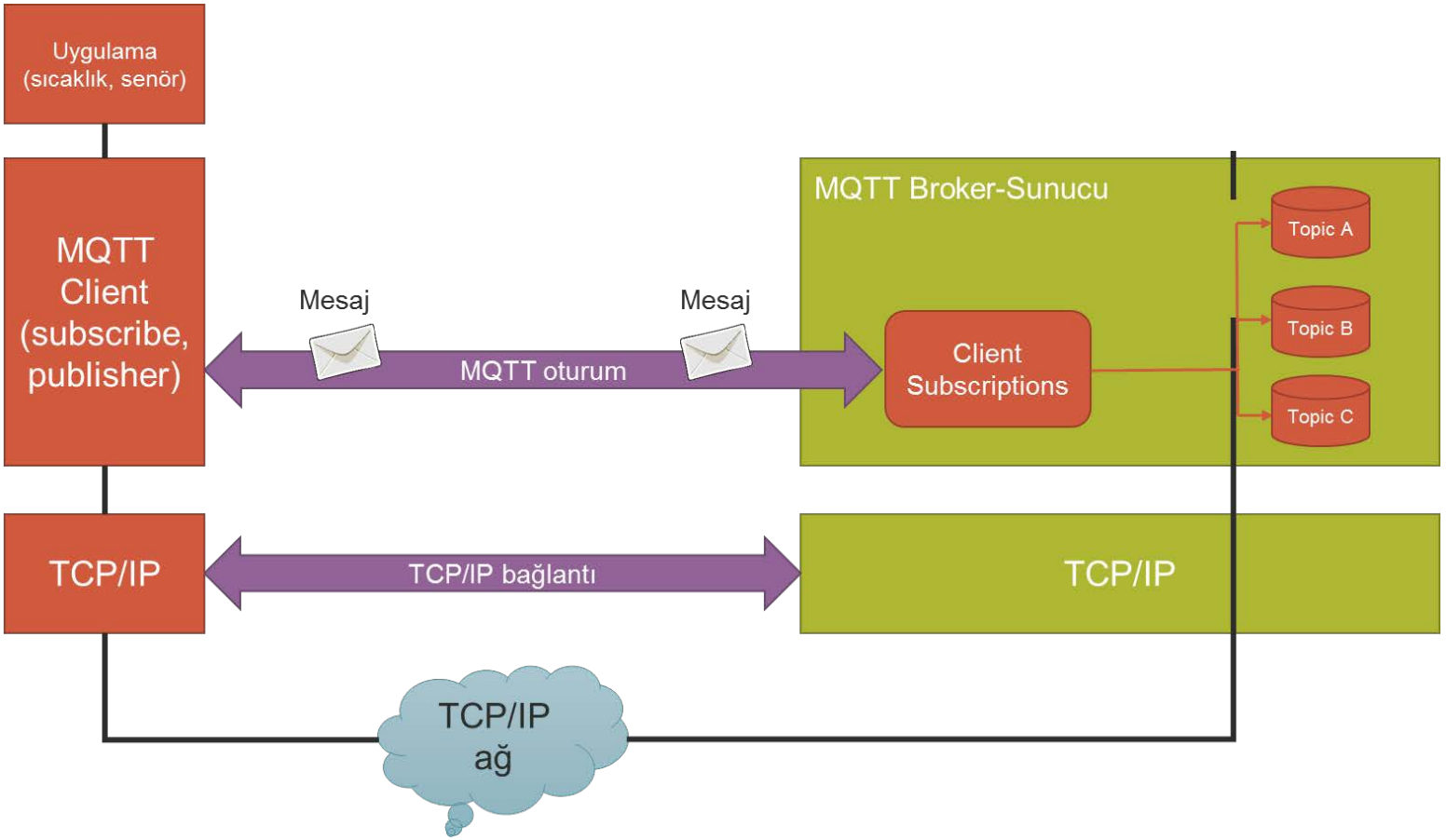
- MQTT, abone (subscriber), yayımcı (Publisher) ve sunucudan (broker) oluşur.
- MQTT mimarisinde cihazlar, bir MQTT broker kullanarak mesaj yayımlayabilir ya da mesaj dinlemek için kayıt olabilir.
- Mesaj, konu (topic) ve bilgi/yük (payload) iki ana kısımdan oluşur.
- Publisher belirli bir konuda mesaj yayımlar, mesajı sadece abone olanlar alır.
- Örneğin bir sensör belirli aralıklarla ölçüm değerlerini yayımlar, bu sensöre abone olanlar bu mesajı alır.



Mosquitto
HiveMQ
Mosca
Paho

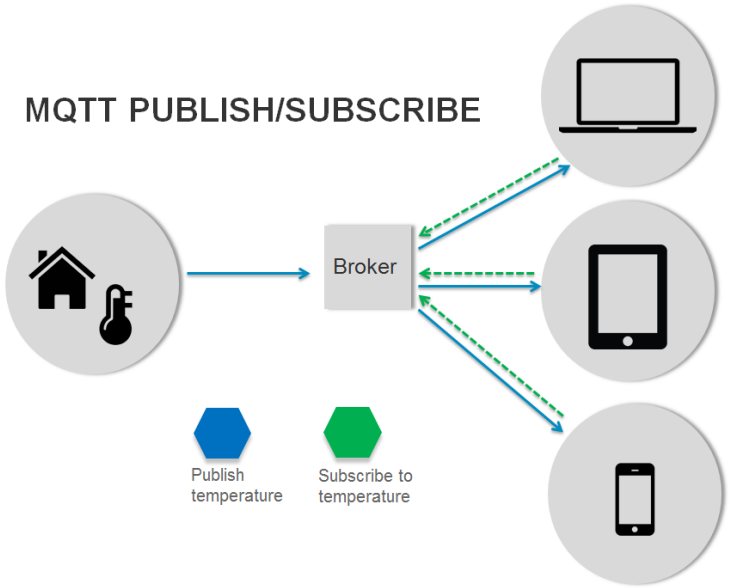


MQTT Katmanlı Mimarisi



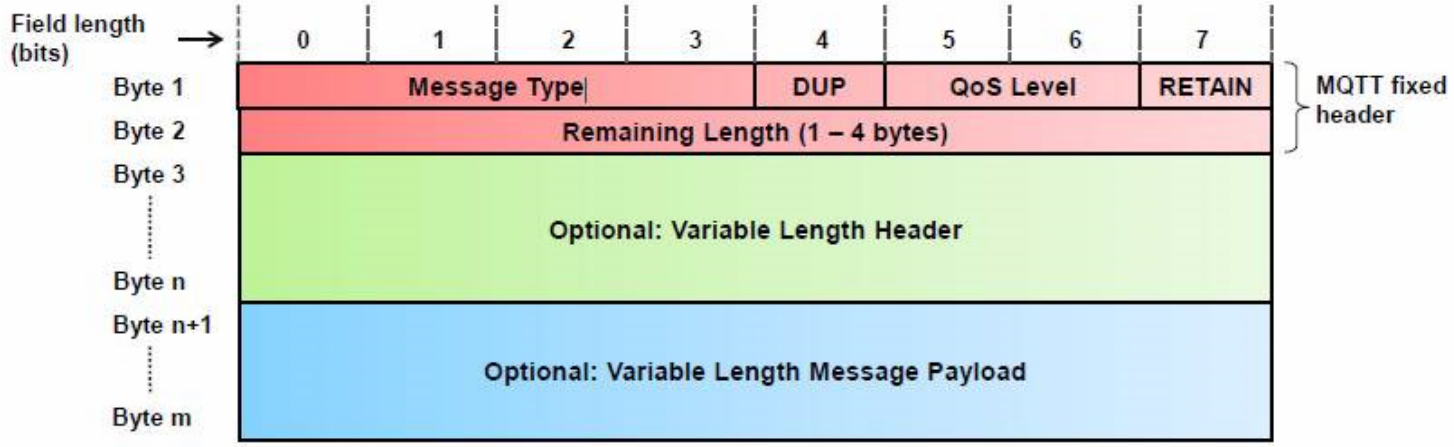
MQTT Broker

- M2M cihazlar, MQTT broker ile mesaj yayımlayabilir ya da mesaj dinlemek için abone olabilir.
- Broker, TCP sunucu görevini yerine getiren ve dinamik veritabanına sahip bir programdır.
- Alınan paketler konulara (topics) göre indekslenmiş olarak veritabanında tutulur.
- Bir paket için abone olan yoksa o paket veritabanından atılır. Eğer abone iletişimde değilse paket belirtilen süre sonra silinir.
- Topic özelliği ile MQTT protokolünde birden bire, birden çoğa, çoktan çoğa bağlantılar kurulabilmektedir.
- MQTT Broker varsayılan portu 1883'tür.



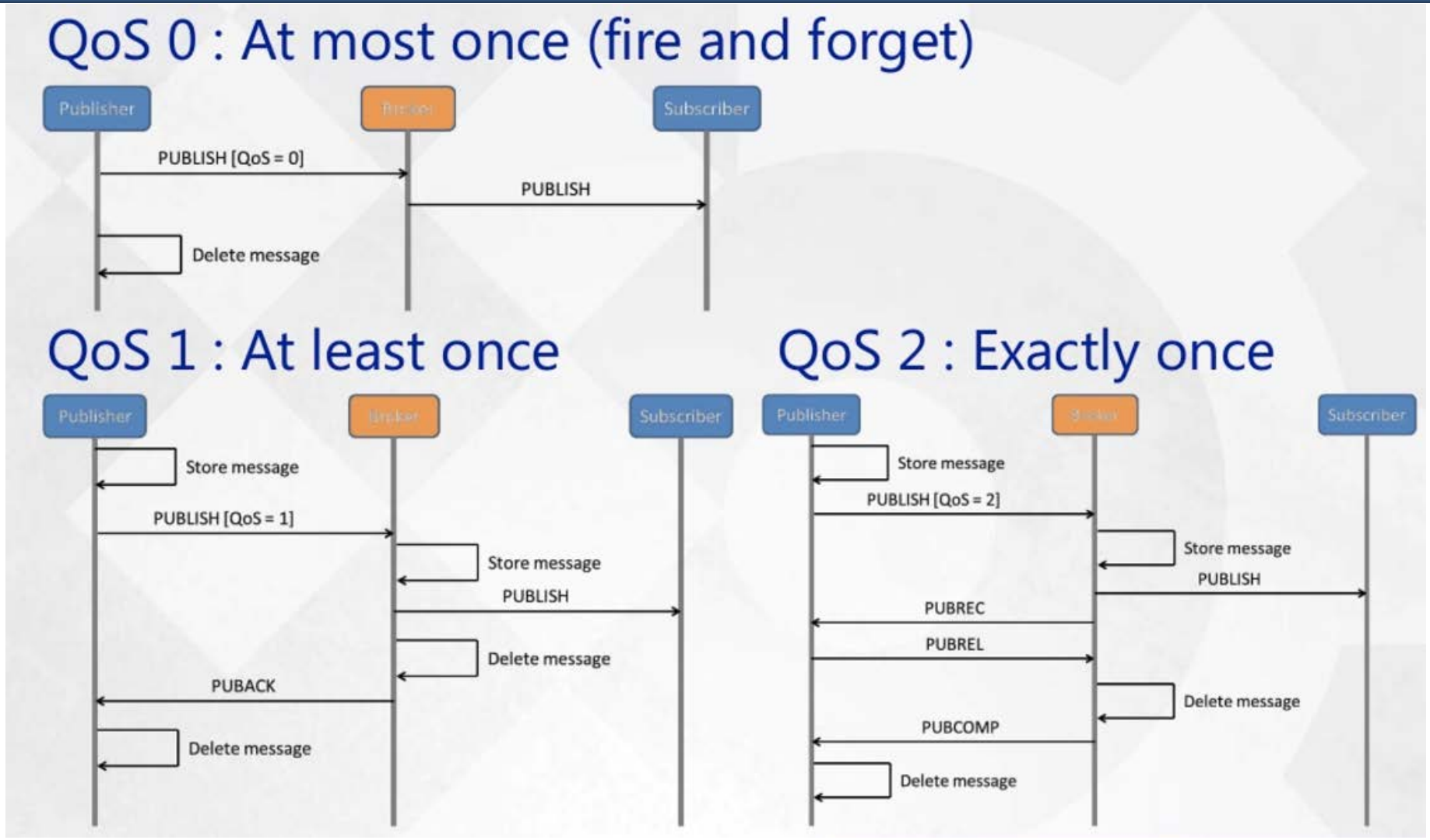
- Mosquitto: En yaygın MQTT brokerlarından biridir. C dilinde geliştirilmiştir ve MQTT protokol standartlarına göre M2M iletişimi gerçekleştiren platformdur.

MQTT Mesaj Yapısı



- MQTT minimum mesaj boyutu 2 bayttır. (C# M2Mqtt kütüphane 30 KB)
- Message Type alanı, CONNECT, CONNACK, PUBLISH, SUBSCRIBE mesaj türünü gösterir.
- DUP, mesajın kopyalandığını (duplicate) gösteren bayraktır.
- QoS Level, yayımlanan mesajın hangi QoS seviyesinde iletileceğini gösterir. QoS 0, mesaj iletilmeyebilir, minimum trafik. QoS 1, gönderilen mesaj kendi veritabanında saklanır ve tekrar tekrar gönderilebilir. Mesaj kesin iletilir, birden fazla iletilebilir. QoS 2, QoS 1 ile benzer ancak mesaj iletimi gerçekleştiğinde bir cevap bilgisi alınır. Mesaj tek seferde ve kesin iletilir. Maksimum trafik.
- Retain, son alınan publish mesajı sunucuya bilgilendirir ve yeni aboneye ilk mesaj olarak iletilir.
- Remaining Length, mesajın kalan boyutunu gösterir.
- Header kısmı, protokol versiyonu vb. bilgileri içerir.
- Payload kısmı, topic, mesaj, kullanıcı, şifre gibi bilgileri içerir.

MQTT Servis Kalitesi Mekanizması (QoS)



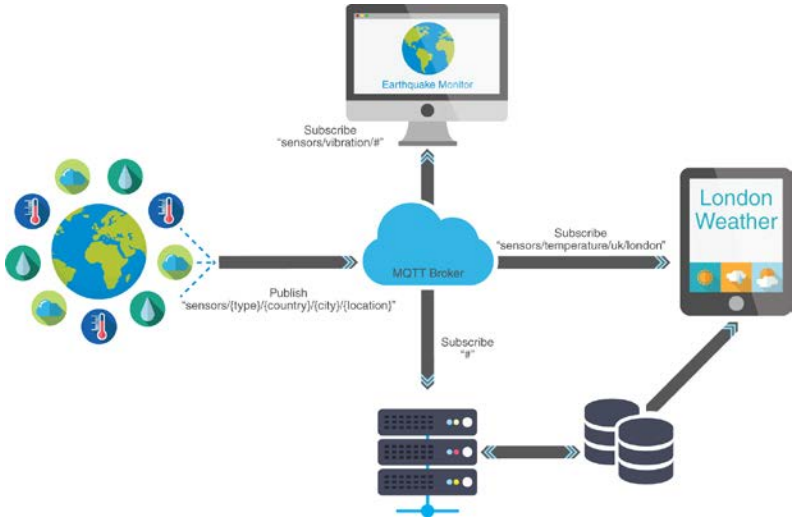
MQTT Protokolünde Güvenlik

- TCP protokolü üzerine kurulu olan MQTT, güvenlik için SSL/TLS kullanır.
- Bağlantı (CONNECT) mesajında kullanıcı adı/şifre kullanır.
- MQTT mesaj yapısında yük şifreli iletilir.

Messega Queue Telemetry Transport (MQTT)

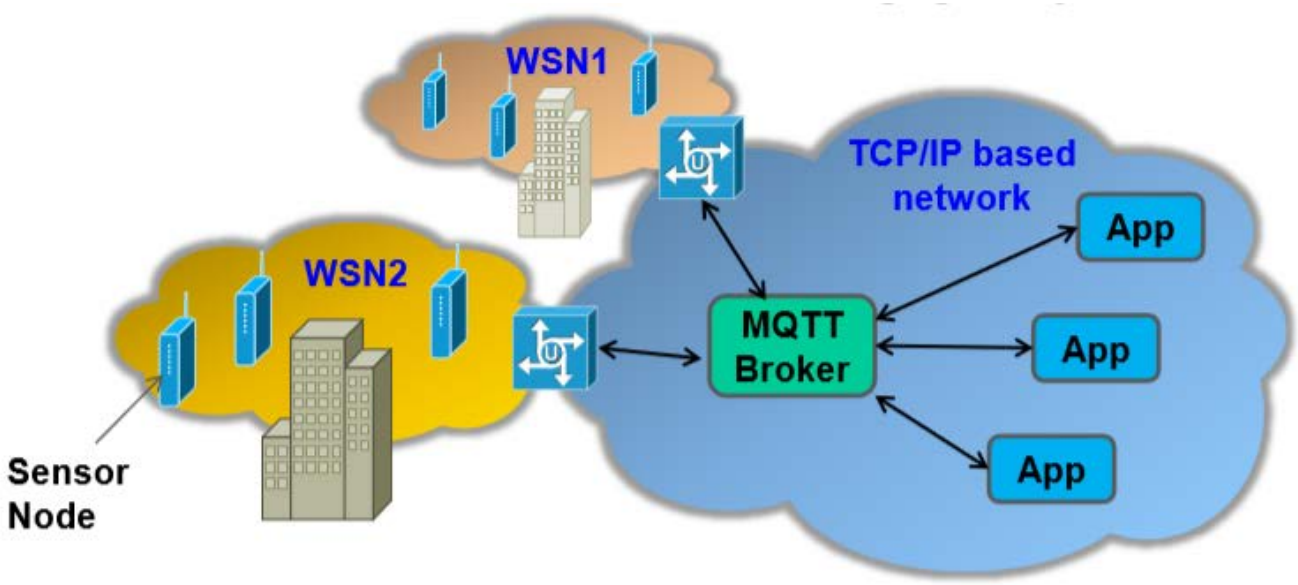
- Sağlık (health care), enerji metre, izleme (monitoring), Facebook bildirim gibi çok sayıda uygulama MQTT kullanır.

- MQTT, genel karakteristikleri
  - Asenkron protokoldür,
  - Bire bir, birden çoğa, çoktan çoğa bağlantı sağlar.
  - Broker temelli haberleşme mekanizmasına sahiptir.
  - Topic'e dayalı adresleme vardır.
  - Güvenlik olarak SSL/TLS destekler.
  - En az kaynak kullanımını hedefler.
  - Milisaniyeler seviyesinde bir haberleşme sunar.
  - TCP/IP bağlantı türünü kullanır.
  - Varsayılan port 1883'tür.
  - TCP/IP protokolünün yazılabildiği Linux, Windows, Android, iOS, MacOS işletim sistemlerinde çalışır.



MQTT-SN

- Kablosuz algılayıcı ağlar için MQTT protokolü versiyonudur.
- MQTT-S, KAA taşıma katmanında TCP/IP protokolünü kullanmaz. Kendine özel protokol yığınları vardır. Bu nedenle doğrudan TCP/IP üzerinde çalışmaz.
- KAA geleneksel TCP/IP ağına ağ geçit cihazları (gateway) yardımıyla bağlanmaktadır.
- MQTT-S'de yaşanan bazı değişiklikler arasında Topic katarının ID ile değiştirilmesi, önceden tanımlanan Topic ID'leri için kayıt zorunluluğunun olmaması örnek olarak verilebilir.



Neden MQTT

- Enerji tüketimi HTTP protokolüne göre daha düşüktür.
- Bağlantıyı sürdürme maliyetinin karşılaştırması

	% Battery / Hour			
	3G		Wifi	
Keep Alive (Seconds)	HTTPS	MQTT	HTTPS	MQTT
60	1.11553	**0.72465**	0.15839	**0.01055**
120	0.48697	**0.32041**	0.08774	**0.00478**
240	0.33277	**0.16027**	0.02897	**0.00230**
480	0.08263	**0.07991**	0.00824	**0.00112**

- Alım (1024 mesaj/ 1 bayt)

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	**16.13%**	**3.45%**	4.23%
Messages / Hour	1708	**160278**	3628	**263314**
% Battery / Message *	0.01709	**0.00010**	0.00095	**0.00002**
Messages Received	240 / 1024	**1024 / 1024**	524 / 1024	**1024 / 1024**

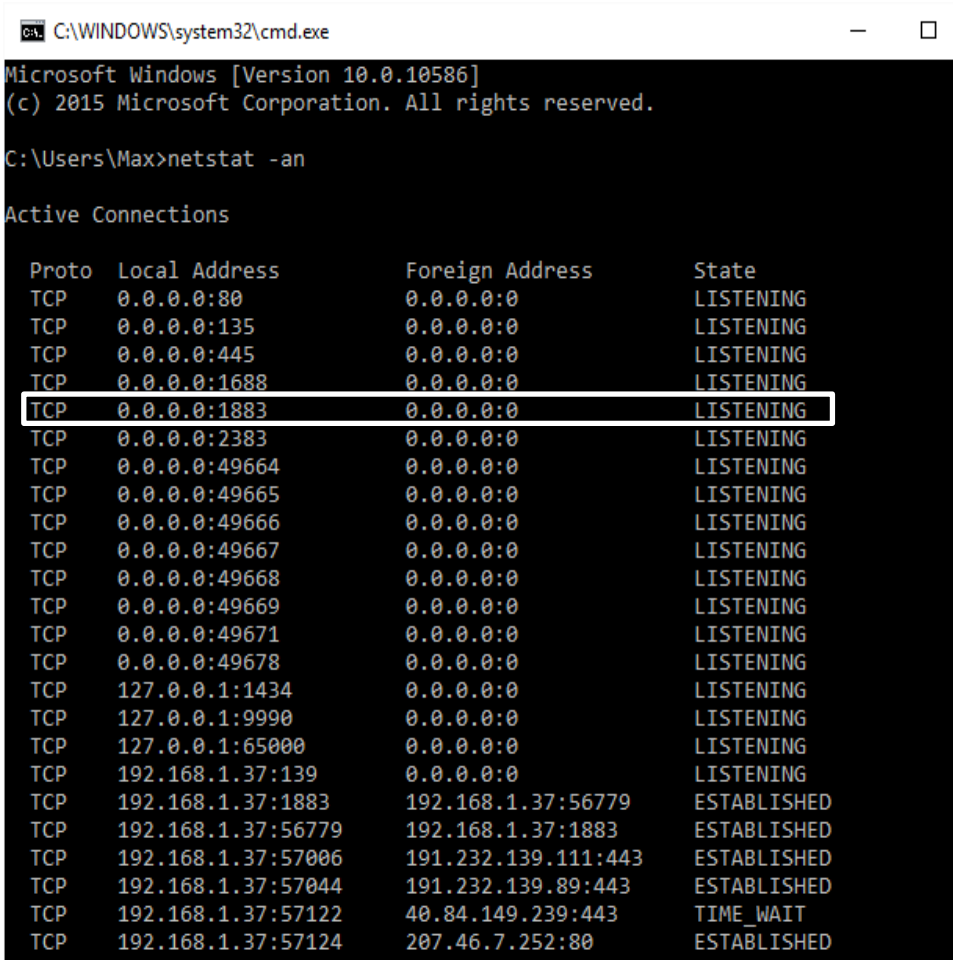
- Gönderim (1024 mesaj/ 1 bayt)

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.79%	**17.80%**	5.44%	**3.66%**
Messages / Hour	1926	**21685**	5229	**23184**
% Battery / Message *	0.00975	**0.00082**	0.00104	**0.00016**

Kaynak: http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https

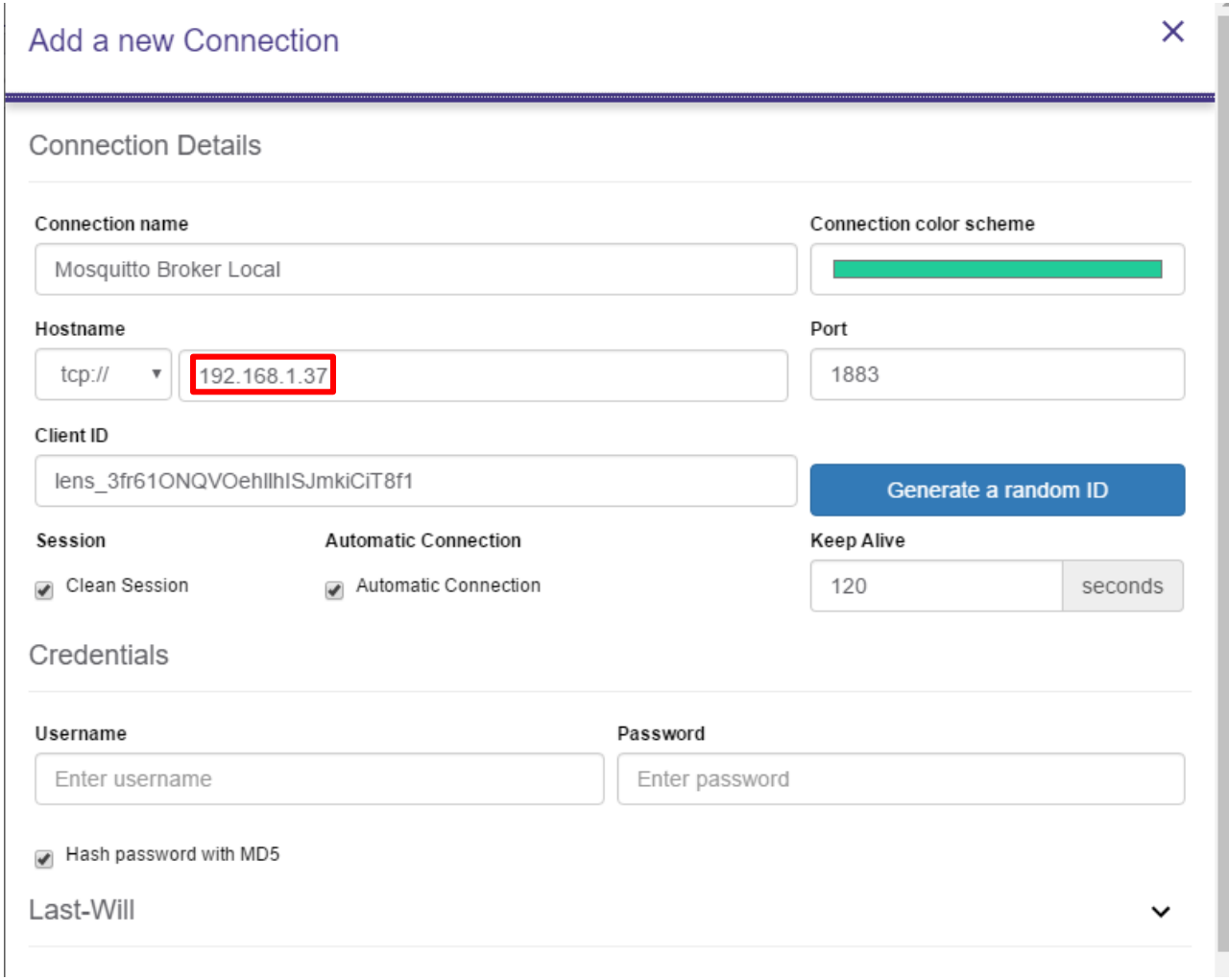
Windows Ortamında Mosquitto MQTT Broker Kurulumu

1. http://mosquitto.org/js/mosquitto-1.1.js resmi sitesinden mosquitto'yu indirip kurunuz
2. Win32 OpenSSL kuruyoruz
3. libeary32.dll, ssleay32.dll, pthreadVC2.dll dosyalarını C:\Windows\System32 klasörüne yapıştırın.
4. Mosquitto'nun çalıştığını kontrol etmek için komut satırına netstat –an yazıyoruz ve 1883 portunun açıldığını kontrol edin.

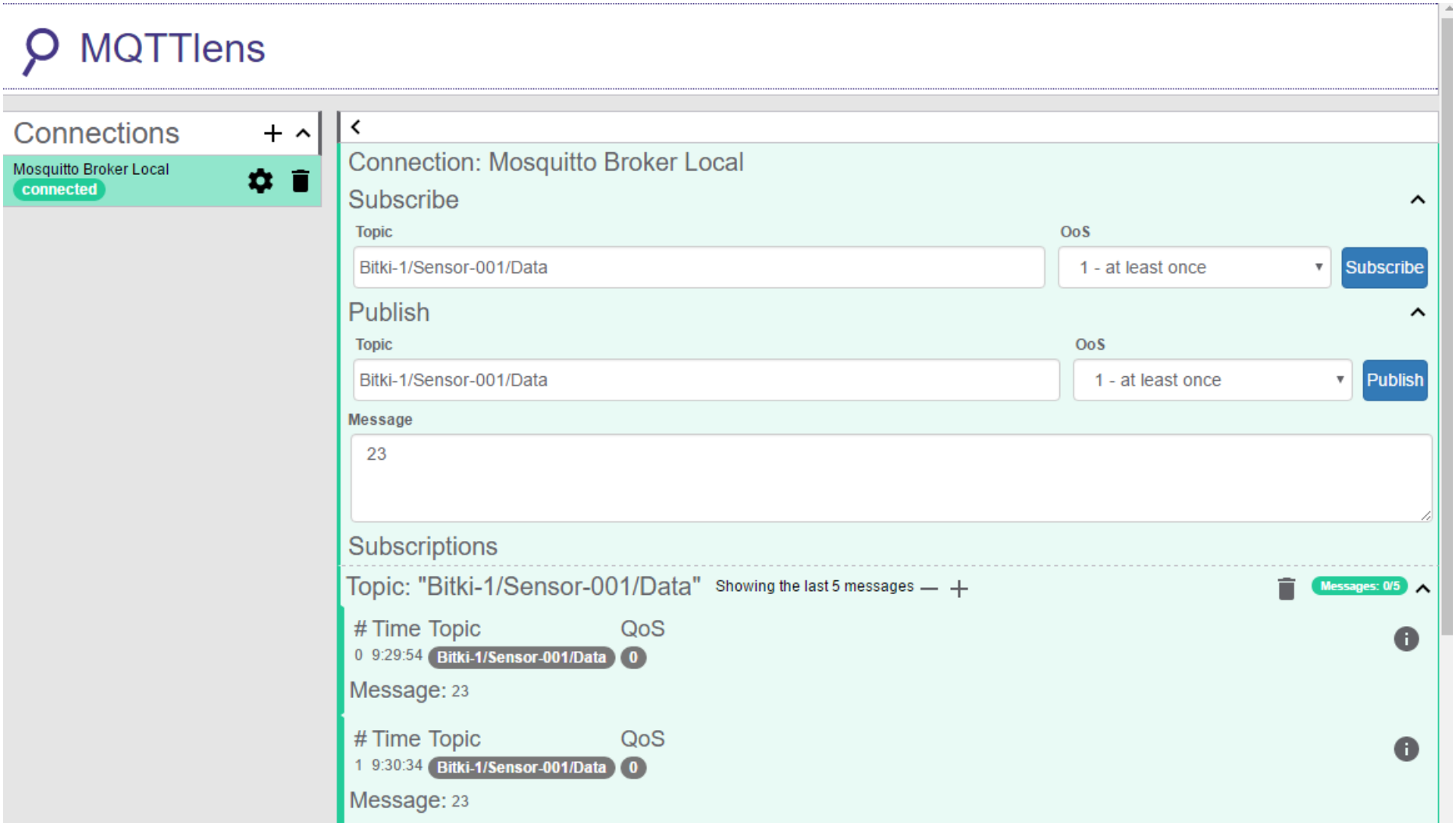


MQTT Broker Yerel Bağlantı (MQTT lens)

- MQTT lens uygulaması MQTT Broker ile iletişimi sağlayan Chrome tarayıcı eklentisidir.
- Bu eklenti ile MQTT mesajlarının Publisher ve subscriber arasında iletişimi test edilebilir.
- Hostname kutusuna, yerel ip adresini girerek connection butonuna basınız.



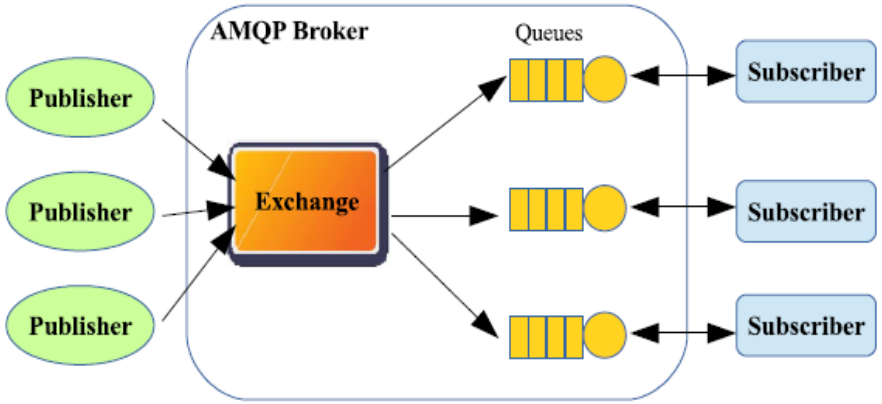
MQTT Broker Yerel Bağlantı (MQTT lens)



İleri Mesaj Kuyruklama Protokolü

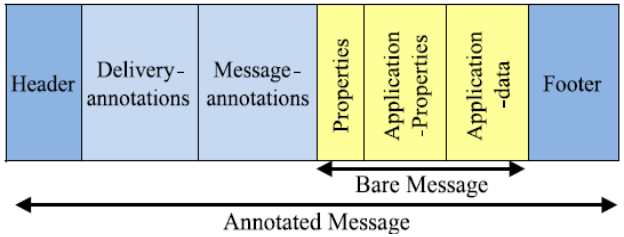
(Advanced Message Queuing Protocol, AMQP)

- AMQP, mesaj yönlendirme odaklanan IOT için açık standart uygulama protokolüdür.
- Kuyruk yapılarına sahip sunucular (broker) ile yayımcı (Publisher) ve abonelere (subscriber) hizmet sunan, 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'de haberleşme exchanges ve mesaj kuyrukları ile sağlanır. Exchanges mesajları uygun kuyruğa yönlendirir. Bu yönlendirme önceden tanımlanmış kurallar ve koşullara (binding) göre gerçekleştirilir.
- AMQP, noktadan noktaya haberleşmenin yanısıra publish/subcribe haberleşme modelini de destekler.
- AMQP mimarisi



İleri Mesaj Kuyruklama Protokolü
(Advanced Message Queuing Protocol, AMQP)

- AMQP, ulaşım katmanı olarak TCP kullanır. Bu katmanın üzerine mesajlaşma katmanını tanımlar.
- AMQP iki tür mesaj kullanır.
  - Bare mesaj: Gönderici kaynaklı mesajlar.
  - Annotated mesaj: Alıcıda görülen mesaj.
- AMQP haberleşme milisaniyeler süresinde gerçekleşir.
- Minimum paket boyutu 60 bayttır.
- MQTT gibi asenkron bir protokoldür.



Veri Dağıtım Servisi
(Data Distribution Service, DDS)

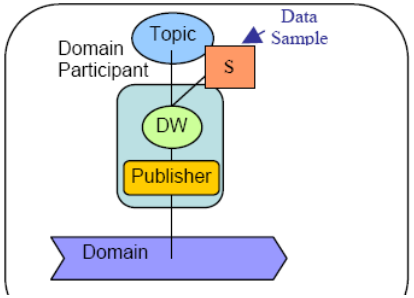
- DDS, Object Management Group (OMG) tarafından gerçek zamanlı M2M haberleşmesi için geliştirilmiştir.
- DDS, daha az katman, daha az iş yükü gibi avantajlar sunar.
- DDS, publish-subcribe protokolüdür.
- Mikrosaniyeler seviyesinde haberleşme ile gerçek zamanlı iletişimi destekler.
- Publish-Subscribe yapısındaki MQTT, ve AMQP protokolünden farklıları daha çok QoS desteği sunar. Güvenli, acil/ivedi, öncelikli, güvenilir gibi haberleşme kriterlerini dikkate alarak 23 QoS seviyesi/politikası sağlar.
- Ulaşım katmanı olarak UDP ve TCP'nin her ikisini de destekler.
- 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.
- Güvenlik olarak SSL/TLS ile Datagram TLS (DTLS) destekler.
- İçerik farkında yönlendirme özelliği vardır (Payload inceleme).

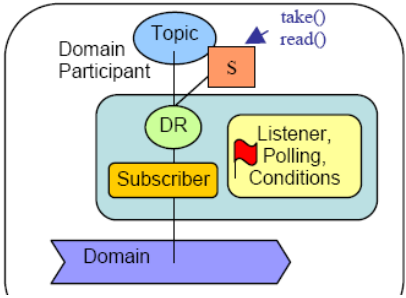
Veri Dağıtım Servisi
(Data Distribution Service, DDS)

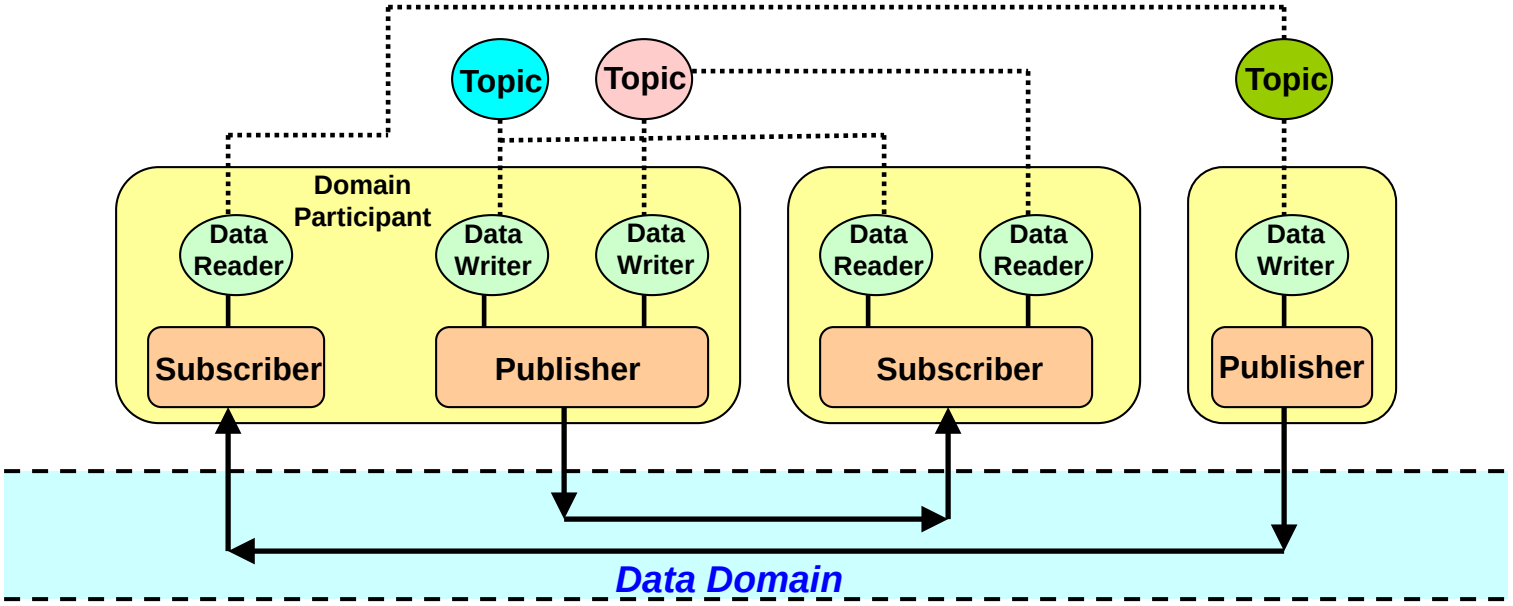
- Topics: Veri dağıtım konuları
- Data Writers: Veri üreticileridir.

  Publisher ile birlikte göndericiyi oluşturur.
- Data Readers: Veri tüketiciler/alıcılardır.

  Subscriber ile birlikte alıcıyı oluşturur.
- Topics, Data Readers/Writers QoS politikaları ile konfigüre edilir.







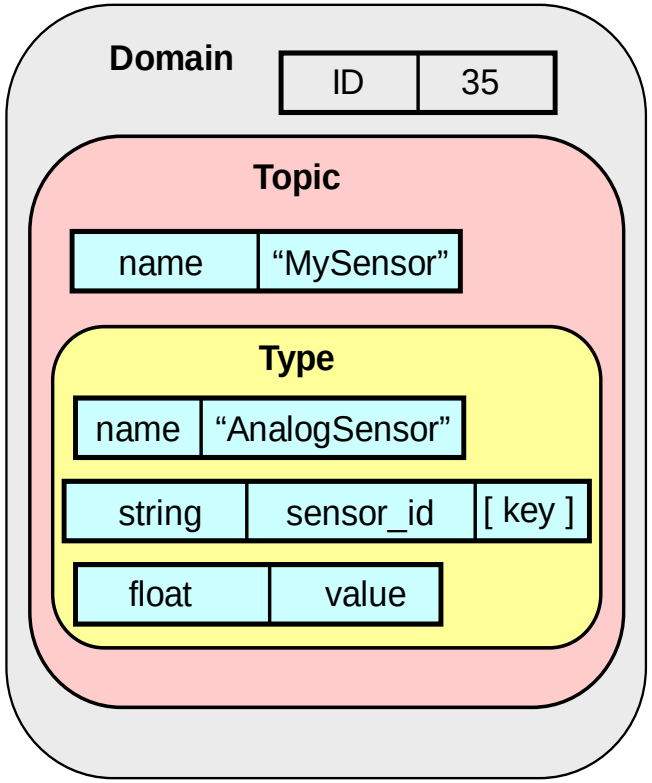
Veri Dağıtım Servisi
(Data Distribution Service, DDS)

- Topics, Publisher ile subscriber arasındaki bağlantıdır.

- Topic, Name ve Type alanlarından oluşur.
  - Name, alan (domain) içerisindeki tektir (unique).
  - Birden fazla topic, aynı tipe sahip olabilir.

- İçerik farkında abonelikler için multitopics,

ve content-filtered topics gibi yapılar sağlar.

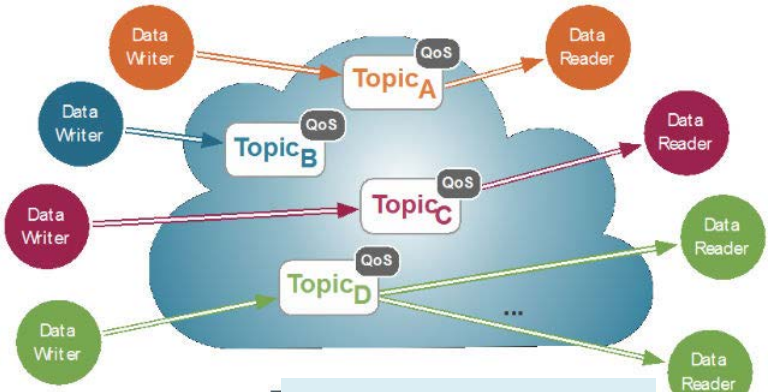


Veri Dağıtım Servisi
(Data Distribution Service, DDS)

- DDS QoS Protokolleri
  - Deadline
  - Destination Order
  - Durability
  - Entity Factory
  - Group Data
  - History
  - Latency Budget
  - Lifespan
  - Liveliness
  - Ownership
  - Ownership Strength
  - Partition
  - Presentation
  - Reader Data Lifecycle
  - Reliability
  - Resource Limits
  - Time-Based Filter
  - Topic Data
  - Transport Priority
  - User Data
  - Writer Data Lifecycle

Veri Dağıtım Servisi
(Data Distribution Service, DDS)

- DDS desteklenen servis kalitesi (QoS) politikalarının bir kısmı
  - Latency_Budget: Uçtan uca kabul edilebilir gecikmeleri tanımlar.
  - Time_Based_Filter: Yavaş tüketiciler ile hızlı üreticiler arasında değişime aracılık eder.
  - Resource_Limits: Servis tarafından kullanılan kaynakları kontrol eder.
  - Reliability: Verinin gerçek zamanlı ulaştırılmasını sağlar.
  - History: Ulaştırılan verinin içeriğini (değerlerini) kontrol eder.

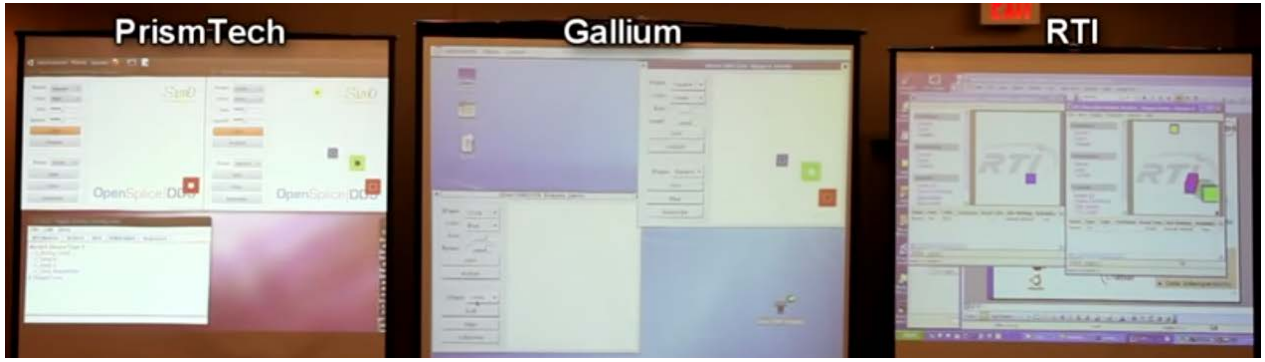


DDS Kavramsal Model

Veri Dağıtım Servisi
(Data Distribution Service, DDS)

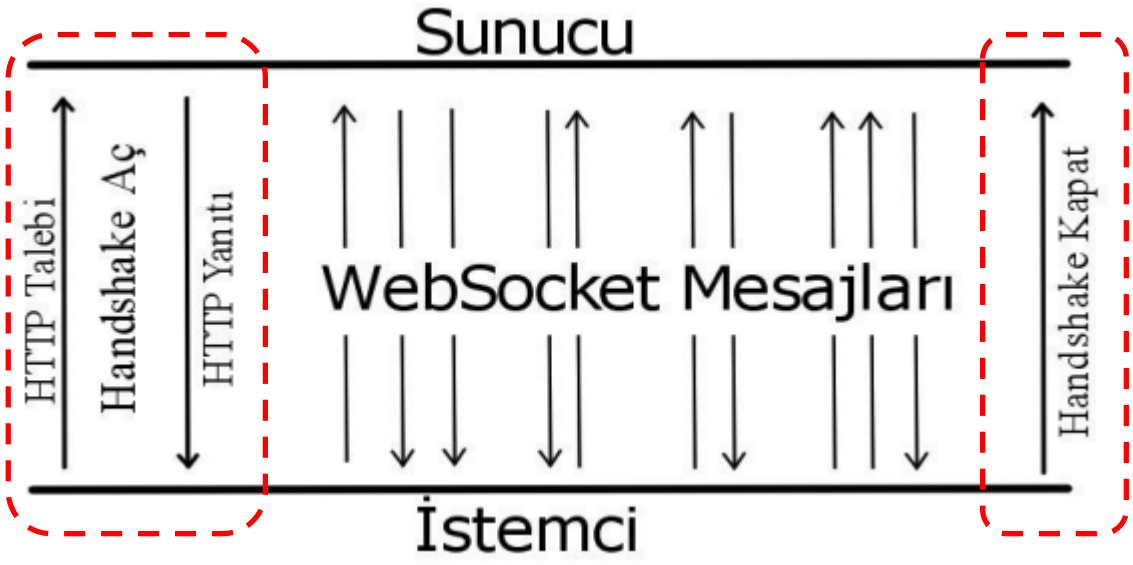
- DDS, ilk olarak Amerikan ordusunda düşman/araç/silah yer tespiti bilgisinin uçaklara iletilerek hedefi yok etmek için kullanılmıştır.
- DDS, Tokyo Metropolitan Otoban araç takibinde kullanılmaktadır.
- Kullanım Alanları
  - Akıllı Şehir gibi Smart Grid yönetimlerinde,
  - Akıllı telefonların işletim sistemlerinde,
  - Araç ve Taşıma Sistemlerinde,
  - İnternet radyolarında,
  - Sağlık hizmetinde

- DDS Platformlar
  - RTI,
  - PrismTech,
  - Gallium,



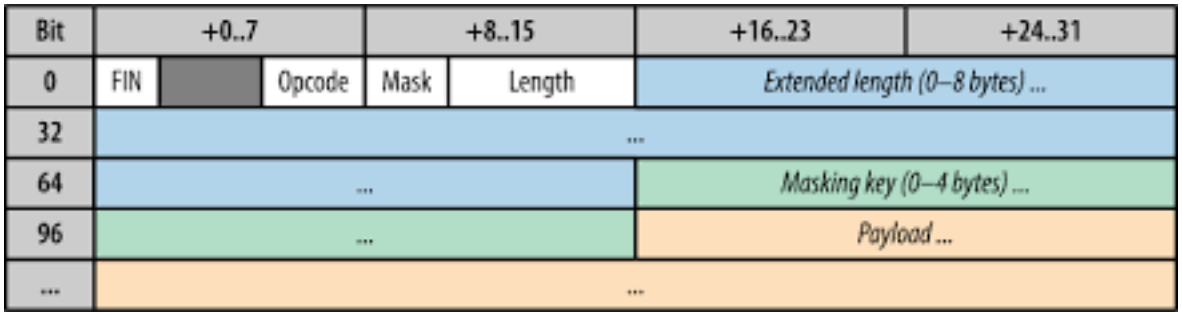
WebSocket

- Sunucu ve istemciler arasında TCP kanal üzerinden gerçek zamanlı ve çift yönlü bağlantı sunar.
- Çift yönlü mesajlaşma desteği nedeniyle taraflar (istemci-sunucu) istediğine mesaj gönderebilir.
- WebSocket protokol, El Sıkışma (Handshake) ve veri transferi olmak üzere iki temel aşamadan oluşur.
- El Sıkışma, HTTP talepleri üzerinden başlatılarak sağlanır. Ardından mesajlar TCP üzerinden iletilir.
- Cihazlar çalıştığı sürece bağlantı korunur.
- WebSocket IETF tarafından kabul edilen bir protokoldür. RFC 6455, WebSocket protokolünün özelliklerini belirtmektedir.



WebSocket

- WebSocket protokolünün genel özellikleri şunlardır:
- Küçük başlık boyutu (2-6 bayt arası),
- Bağlantı bir kez kurulduğundan yeniden başlık ihtiyacı olmaması yeni bir gecikme süreside gerektirmemektedir.
- Protokol bağımsız TCP tabanlı protokoldür.
- HTTP'deki gibi şifresiz ya da TLS kullanılarak şifreli bağlantı sağlanabilir.
- HTTP ve HTTPS ile aynı portları (wss:// 443, ws:// 80) portları kullandığından güvenlik duvarları ve proxy'ler tarafından tanınır. Böylelikle var olan HTTP ayarlarında herhangi bir değişiklik olmadan çalışabilir.



WebSocket Paket Yapısı

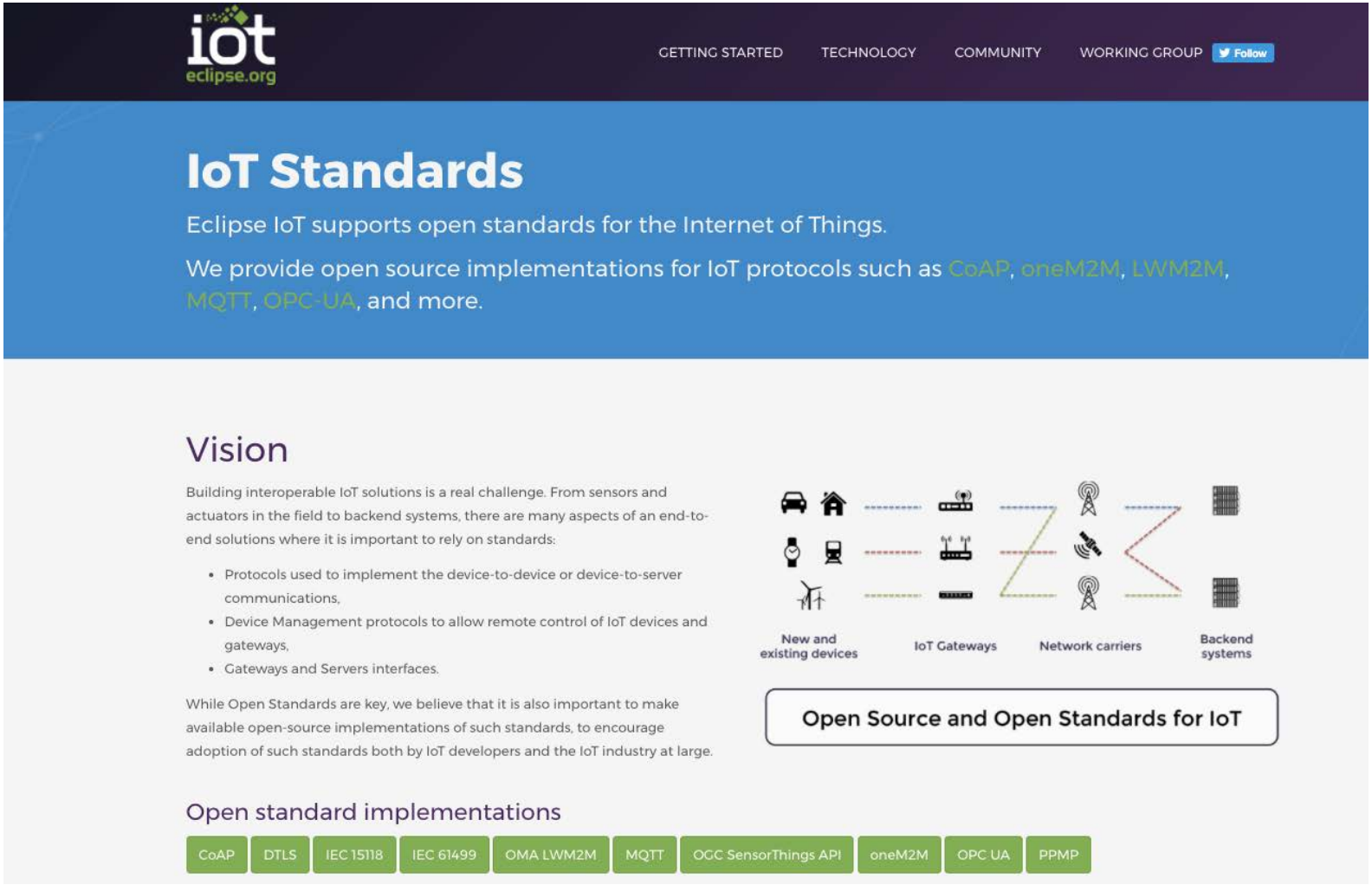
IoT Uygulama Protokollerinin Karşılaştırılması

- 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	×	-

IoT Haberleşme Protokollerinin Gerçeklenmesi

- iot.eclipse.org adresinden birçok IoT Haberleşme Protokollerinin kodlarına erişilebilir.



KAYNAKLAR

❖ Temel Kaynaklar

- Doç. Dr. Cüneyt BAYILMIŞ ve Doç. Dr. Kerem KÜÇÜK, **"Nesnelerin İnternet'i: Teori ve Uygulamaları"**, Papatya Yayınevi, 2019.

❖ Diğer 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.**
- **M. H. Amaran, N. A. M. Noh, M. S. Rohmad, H. Hashim, "A Comparison of Lightwight Communication Protocols in Robotic Applications", IEEE International Symposium on Robotics and Intelligent Sensors (IRIS2015), 400 – 405, 2015.**
- **P. Penial, M. Franekova, 'Model of Integration of Embedded Systems via CoAP Protocol of Internet of Things', IEEE International Conference on Applied Electronics, 2016.**
- **www.mqtt.org**
- **https://www.oasis-open.org/committees/tc_cat.php?cat=iot**
- **DDS Uygulama Örneği: https://www.rti.com/downloads/shapes-demo.html**
- **www.omg.org**
- **iot.eclipse.org**