WEB SERVİSLERİ
TEMEL KAVRAMLAR

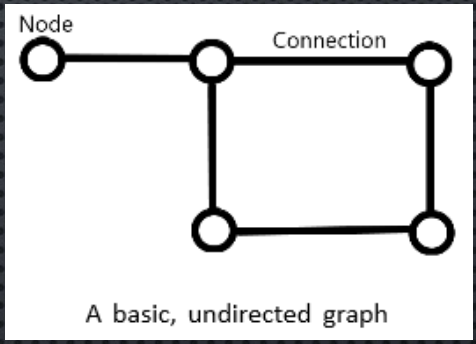
Dr. Öğr. Üyesi Ömer Faruk ACAR

BİR PROGRAMDA NETWORK

- Neden tercih edilir?
  - Uygulamanın ölçeklenebilirliğini arttırmak
  - Performansını ve verimini arttırmak
  - Uzaktan haberleşme ve veri paylaşımı sağlamak
  - İşlem/ hesap gücünü arttırabilmek için

- Tutarlı ve verimli programlama için temel programlama yapılarını kavramak, network hakkında bilgiye sahip olmak gerekmektedir.

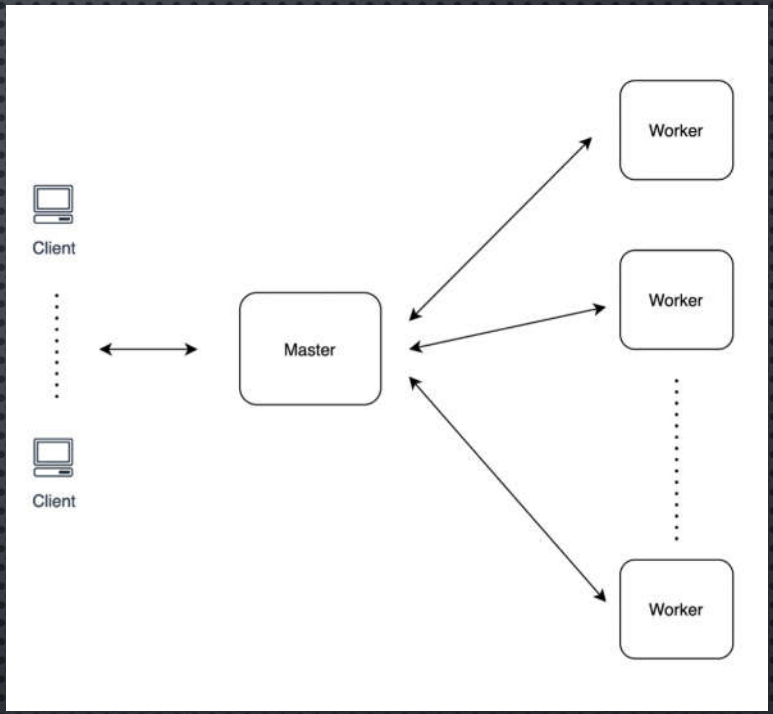
BİLGİSAYAR AĞI

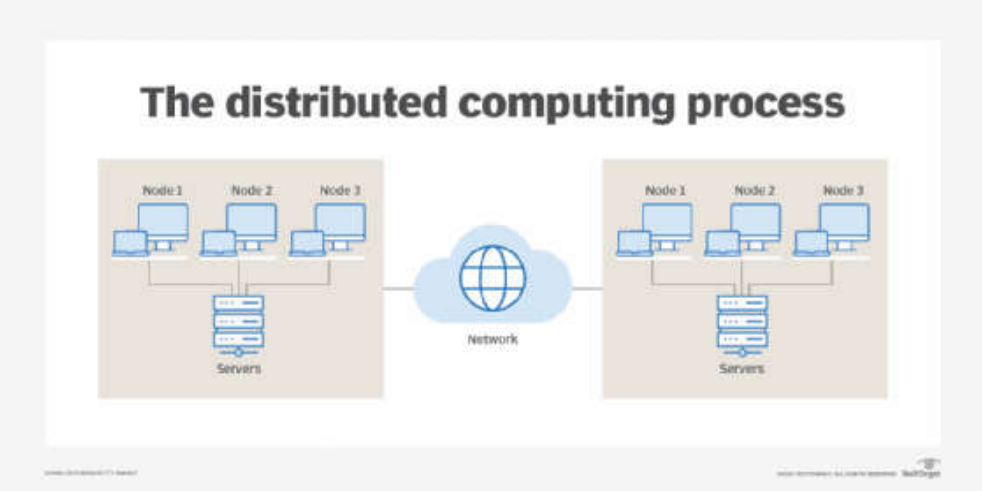


- BİLGİSAYAR AĞI YAN ŞEKİLDE GÖSTERİLEN GRAF İLE BENZERDİR. BİRBİRİNE BAĞLI NOKTALAR.
- DÜĞÜMLER BİLGİSAYAR YADA CİHAZLARI, ARASINDAKİ BAĞLANTILAR KABLOLARI TEMSİL EDEBİLİR.
- PROGRAMCI BAKIŞ AÇISINDAN BİLGİSAYAR AĞI: BİR AMACI YADA AMAÇLARI GERÇEKLEŞTİRMEK İÇİN İLETİŞİM KANALLARI İLE BAĞLANMIŞ BİLGİSAYARLAR ARASINDA HESAPLANACAK KAYNAKLARIN GÜVENLİ BİR ŞEKİLDE GÖNDERİLDİĞİ, ALINDIĞI, İLETİLDİĞİ CİHAZLAR KÜMESİNİN OLUŞTURDUĞU YAPI.

- Bir networkde 2 cihazda olabilir, yüzlerce hatta milyonlarca cihaz da olabilir.
- Cihazlar arasında iletişim protokoller ile gerçekleştirilir.
- Programcılar genellikle; ağ yapısının nasıl oluşturulduğu, hangi yapılandırmaların kullanıldığı, bunların nasıl birbirine bağlandığına odaklanmazlar.
- Networkcüler genellikle; programın nasıl yazıldığı, verilerin nasıl oluşturulduğu, hesaplamaların nasıl yapıldığına odaklanmazlar.

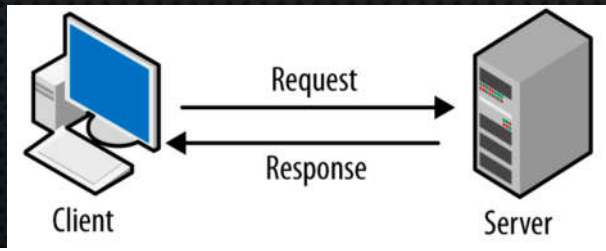
- Dağıtık sistem: Bir işi gerçekleştirmek için birden fazla cihazın bir arada çalışması olarak tanımlayabiliriz.
- Network dağıtık bir sistem olsada her dağıtık sistem bir network oluşturmaz.
- Programsal olarak bir networkden bahsedebilmek için açık iletişimi sağlıyabiliyor olması gerekmektedir.





- Device agnosticism: Yerel yazılımlarda kaynakları kontrol edebilir ve erişimimiz olup olmadığını görebilir. İşler ağ üzerine geldiğinde zorlaşmaya başlar.
- Ağa bağlı cihazlar kendisine neyin bağlı olduğunu tam olarak bilemezler. Ağa bağlı cihazlar kendini ve kaynaklarını tanıtmaları gerekmektedir. Yalnızca bağlanırken değil bağlantı devam ettiği sürece kaynağın erişilebilir olup olmadığının kontrol edilmesi gerekmektedir.

- Ağda farklı cihazlar farklı kaynaklara sahip olabilir. Bu kaynaklar paylaşılabilir yada kullandırılabilir.
- Kaynağa sahip olmayan cihaz; erişim yada kullanım için istekte bulunabilir. İstekte bulunan cihaz **istemci (client)** olarak isimlendirilmektedir.
- Aynı şekilde sahip olduğu kaynağı diğer cihazlara erişime yada paylaşıma açan cihaz **sunucu (server)** olarak isimlendirilmektedir.
- P2P: Per to per kısaltması olan p2p, noktadan noktaya kurulan haberleşmedir. Bu durumda istemci olan cihaz aynı zamanda kendi kaynaklarını paylaşabilir ve sunucu gibi davranabilir.



- **Senkron haberleşme**: Gönderilen cevap, son isteğin cevabı ise buna senkron haberleşme, bu şekilde çalışan sistemleri senkron sistemler olarak isimlendirebilir.
- Ağ üzerinde mesafeye, yoğunluğa, istekte bulunan cihaza bağlı olarak her zaman istenilen bilgi yada cevap anında gelmeyebilir.
- **Asenkron haberleşme**: Gelen istek işlenip diğer isteklerden bağımsız olarak herhangi bir zamanda cevap dönmesi, bu şekilde çalışan sistemleri asenkron sistemler olarak isimlendirebiliriz.

- **Yönlendiriciler:** Bu cihazlar veri oluşturmaz, kendi kaynaklarını kullanmazlar. Ağdaki diğer cihazların verilerini taşır yada onlara yol gösterirler.
- **İletişim Kanalları**: cihazların haberleşmesinde fiziksel yada mantıksal arabirimler, yollar.

  Standartlar ve Protokoller: Yazılan program başka bir kaynaktan veri alıyorsa, kaynağın gönderdiği şekilde veriyi almamız yada belirlediğimiz şekilde veriyi onaylamamız gerekir. Aynı şekilde yazdığımız programın başka programlara kaynak olmasını istiyorsak piyasada kullanılan ve kabul görmüş standartlarda (HTML, Json, TCP, UDP) veriyi göndermemiz gerekiyor.

İLETİŞİM YÜKÜ

- Ağa dayalı programlarda açık iletişim beraberinde fazla konuşmayı da getiriyor. Bu konuşma ile baş edebilmemiz, haberleşmede standartları ve protokolleri iyi seçmemiz gerekmektedir.
- Konuşmanın ne zaman başlayacağı, nasıl konuşulacağı, neler yapılacağının kodlanması gerekmektedir.

DAYANIKLILIK

- Ağa dayalı program geliştirirken programın her iki uçunun açık olması gerekmektedir.
- Uçlardan bir tanesi açık olmadığında, ağdan düştüğünde, olası bir durum olduğunda ne yapılması gerektiğinin, nasıl bir cevap vermesi gerektiğinin kodlanması gerekmektedir.
- Bununla birlikte başka sistemler ile haberleşme yapıyorsak; bu sistemlere bağımlılıklarımızın hesaba katılması gerekmektedir. Bağımlı olduğumuz sistemin çökebileceği, değişebileceği yada güncellenebileceği göz önünde bulundurulmalıdır.

GÜVENLİK

- AĞ ÜZERİNDEN HABERLEŞME GERÇEKLEŞTİRİLDİĞİNDE KAYNAKLARIN VE VERİLERİN GÜVENLİ BİR ŞEKİLDE PAYLAŞILMASI GEREKMEKTEDİR.
- AĞ DIŞINDAKİ BİR CİHAZIN KAYNAKLARA ERİŞİMİNİN OLUP OLMADIĞI HESABA KATILDIĞI GİBİ, AĞ İÇİNDEKİ CİHAZLARIN ARASINDAKİ İLETİŞİM, İŞLEYİŞDE GÜVENLİ OLARAK SAĞLANMASI GEREKMEKTEDİR.
- GÜVENLİK YALNIZCA SUNUCU TARAFI İÇİN DEĞİL, İSTEMCİ TARAFI İÇİNDE DÜŞÜNÜLMELİDİR.

ÖZETLE

Network üzerinden yazılan programlarda göz önünde bulundurulması gereken 2 önemli husus vardır.

- Programın dağıtımı ve yönetilmesi
- Dağıtımlar arasında haberleşmenin sağlanması

Yazılan program;

- yalnızca istemcide çalışabilir,
- yalnızca sunucuda çalışabilir,
- hem istemci hem sunucuda çalışabilir yada
- Bir kısmı istemcide, bir kısmı sunucuda

çalışabilir.

DAĞITIMLAR ARASI HABERLEŞMENİN SAĞLANMASI

- DAĞITIK PROGRAMLARIN HABERLEŞMESİ İÇİN ÖNCELİKLE HABERLEŞECEĞİ YERİ BİLMESİ GEREKMEKTEDİR. BUNU BİR MEKTUP GÖNDERMEYE BENZETEBİLİRİZ.
  - MEKTUBUN İÇERİĞİ GÖNDERİLECEK DATA
  - ALICI AD/SOYAD= ALICI PROGRAM
  - ADRES = PROGRAMIN ÇALIŞTIĞI CİHAZ
- GÖNDEREN CİHAZIN HEDEF CİHAZIN ADRESİNİ BİLMESİ GEREKİYOR (KISMEN YADA TAM OLARAK)

DAĞITIMLAR ARASI HABERLEŞMENİN SAĞLANMASI

- EN TEMEL EN İLKEL ADRESLEME OLARAK TELEFON ÖRNEĞİ VERİLEBİLİR.
- ÜLKEMİZDE KULLANILAN 11 HANELİ SABİT TELEFON NUMARALARININ İLK HANESİ 0 İLE BAŞLAYIP SONRAKİ 3 HANESİ ALAN KODUNU OLUŞTURMAKTADIR. SONRAKİ 7 HANE ABONE NUMARASINI OLUŞTURMAKTADIR.
- TELEFON NUMARALARI EZBERLENMESİ ZOR OLDUĞUNDAN ARAMAK İSTENEN KİŞİYİ BULABİLMEK, HIZLI BİR ŞEKİLDE ULAŞABİLMEK İÇİN TELEFON REHBERLERİ TUTULURDU.
- TELEFON REHBERLERİNE BENZER ŞEKİLDE NETWORKDE CİHAZLARI KOLAY HATIRLAYABİLMEK VE BULABİLMEK İÇİN MODERN REHBERLER: DNS

DOMAİN NAME SYSTEM

- ALAN ADI DÖNÜŞÜM SİSTEMİ OLARAK DA BİLİNEN DNS, ALAN ADLARINI IP ADRESLERİNE VE PROTOKOLLERE ÇEVİRİRLER.
- DNS LERE BAŞVURU URL İLE GERÇEKLEŞTİRİLİR.

https://www.google.com.tr

https://www.karabuk.edu.tr

https://bbf.karabuk.edu.tr/

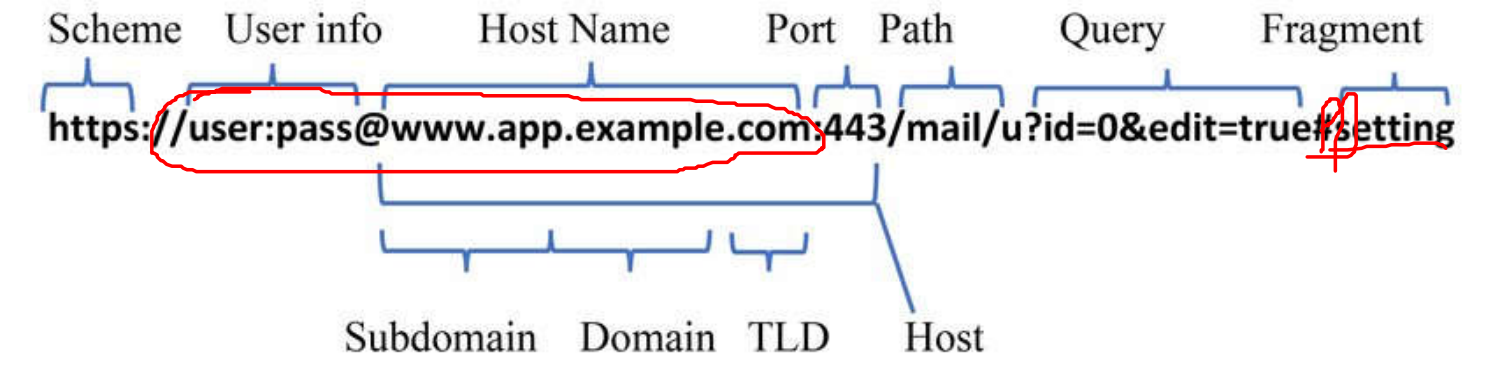
https://bbf.karabuk.edu.tr/yazilim

DOMAİN NAME SYSTEM DNS

Type	Type id. (decimal)	Defining RFC	Description	Function
A	1	RFC 1035[1]	Address record	Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc.
AAAA	28	RFC 3596[2]	IPv6 address record	Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
AFSDB	18	RFC 1183	AFS database record	Location of database servers of an AFS cell. This record is commonly used by AFS clients to contact AFS cells outside their local domain. A subtype of this record is used by the obsolete DCE/DFS file system.
APL	42	RFC 3123	Address Prefix List	Specify lists of address ranges, e.g. in CIDR format, for various address families. Experimental.
CAA	257	RFC 6844	Certification Authority Authorization	DNS Certification Authority Authorization, constraining acceptable CAs for a host/domain
CDNSKEY	60	RFC 7344		Child copy of DNSKEY record, for transfer to parent
CDS	59	RFC 7344	Child DS	Child copy of DS record, for transfer to parent
CERT	37	RFC 4398	Certificate record	Stores PKIX, SPKI, PGP, etc.
CNAME	5	RFC 1035[1]	Canonical name record	Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.

UNİFORM RESOURCE LOCATOR

- TEKTİP KAYNAK KONUMLANDIRICI OLARAK ÇEVİREBİLDİĞİMİZ URL LER
- WEB SAYFALARINA, YADA AĞ ÜZERİNDEN ÇALIŞAN PROGRAMLARA ULAŞIRKEN KULLANILAN URL LER, ERİŞİLECEK OLAN KAYNAĞIN ADRESİ/ROTASI/TÜRÜ HAKKINDA BİLGİLER İÇERMEKTEDİR.



UNİFORM RESOURCE LOCATOR

- Şema/Protokol: Kaynağa erişim için kullanılacak protokolü belirtir.
- User Info: Varsa kullanıcı ID ve şifre bilgilerinin girildiği alandır. Default olarak başvuruda bulunulacaksa @ işareti konulmadan yazılır.
- Host Name Alanı: Kaynağın barındırıldığı cihaza erişim için belirtilen rota. Mevcut kaynağa direk erişim mümkün ise IP adreside yazılabilir. Her alt alan ‘’ . ’’ işareti ile aytılır.
- Port: Hedef cihaz farklı bağlantı noktalarından farklı isteklere cevap veriyor olabilir. Kaynağın dinlendiği port numarası yazılır.

UNİFORM RESOURCE LOCATOR

- PATH/YOL:KAYNAĞA ULAŞMAK İÇİN İSTEKLERİN GEÇMESİ GEREKEN YOLU BELİRTİR. HER BİLEŞEN ‘‘/’’ İŞARETİ İLE AYRILIR.
- QUERY: ‘‘?’’ İLE BAŞLAYAN SORGU ALANI İLE KAYNAĞA PARAMETRE GÖNDEREBİLİR. HER PARAMETRE KEY VE VALUE ALANI İÇERİR.
- ANAHTAR BELİRTİLDİKTEN SONRA ‘‘=’’ İŞARETİ İLE DEĞERİ GÖNDERİLİR.
- EĞER BİR URL DE BİRDEN FAZLA SORGU İÇERİYORSA BU SORGULAR ‘‘;’’ YADA ‘‘&’’ İŞARETLERİ İLE AYRILIR.
- FRAGMENT/PARÇA/KISIM: KAYNAKTAKİ BELİRLİ BİR PARÇAYA ULAŞILMAK İSTENİLDİĞİNDE KULLANILAN KISIM. BİR WEB SAYFASINDA BELİRLİ BİR BÖLÜME GİTMEK İÇİN YADA GELEN VERİLERİN YALNIZCA BELİRLİ BİR ALANINI ALMAK İSTENİLDİĞİNDE KULLANILABİLİR.

PORT

- Hangi verinin hangi kanala ulaştırılacağı port sayesinde belli olur.
- Port: Hedef ve kaynak arasında hizmet sağlayan servis adresi
- İşletim sistemi tarafından yazılımsal olarak kontrol edilir.
- Hem TCP hemde UDP protokollerinde 16 bit olarak temsil edilir.
- 2^16=65536 adet port adresi vardır.

PORT KAVRAMI

- WELL-KNOW: 0 İLE 1023 ARASI NUMARALARDIR. SUNUCU TARAFINDA KULLANILIR. PROTOKOLLERLE BİREBİR İLİŞKİLİDİR.
- REGİSTERED: 1024 İLE 49151 ARASI NUMARALARDIR. HEM SUNUCU HEM İSTEMCİ UYGULAMALARI TARAFINDAN KULLANILIR.
- DYNAMİC: 49152 İLE 65535 ARASI NUMARALARDIR. İSTEMCİ TARAFINDAN KULLANILIR. GENELLİKLE AYNI PORTA İSTEK YAPAN UYGULAMALARI AYIRT ETMEK İÇİN KULLANILIR.

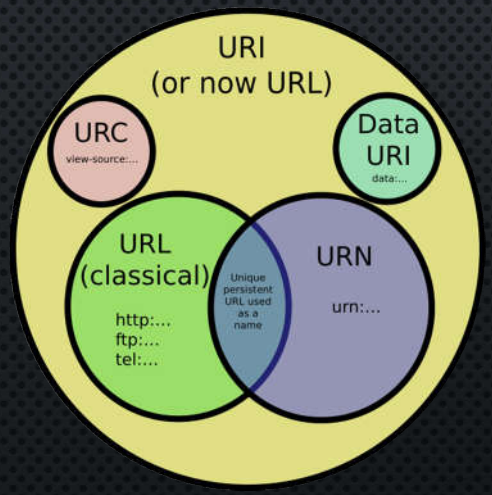
- LIST OF TCP AND UDP PORT NUMBERS - WIKIPEDIA

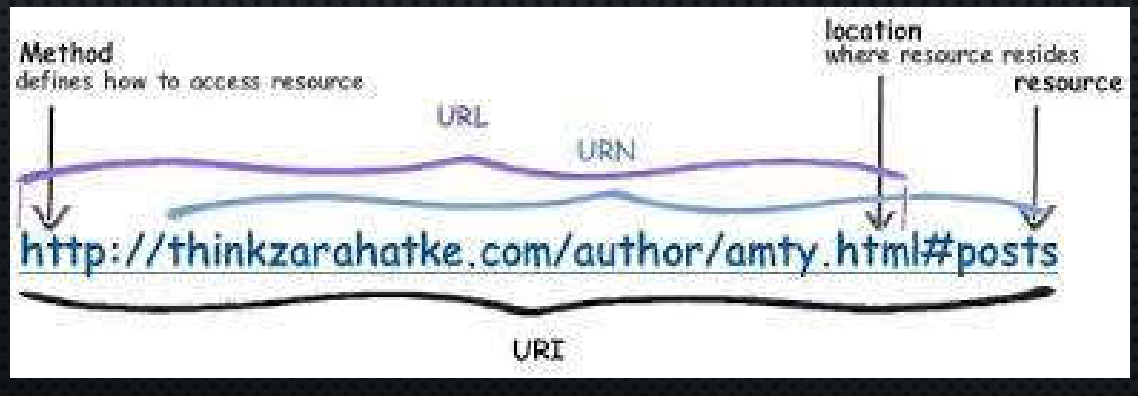
URI,URN,URC NEDİR?

UNİFORM RESOURCE IDENTİFİER

- KAYNAK TANIMLAYICI İLE KAYNAK KONUMLANDIRICI BENZER ŞEKİLDE ÇALIŞMAKTADIR. ARALARINDAKİ UFAK FARKI:
  - URL, URI NİN STRİNGE ÇEVİRİLMİŞ ÖZELLEŞTİRİLMİŞ HALİ OLARAK DÜŞÜNEBİLİRİZ.
- URI LER URL TANIMLARININ DAHA SPESİFİK VE KOLAY YAPILMASINI SAĞLARLAR.

Venn diagram of URIs
as defined by the W3C





OSI MODELİ

- Open System Interconnection – Açık Sistemler Arabağlantısı
- 1984 yılında Uluslararası Standart Kuruluşu (ISO) tarafından ortaya konulmuştur.
- Haberleşen uç birimlerinin, dünyaca kabul edilmiş standartlara göre ve üreticiden bağımsız olarak haberleşebildiği bütünsel bir iletişim ortamının yaratılması için kullanılan 7 katmanlı bir referans modelidir.

Layer 7	Application	Uygulama
Layer 6	Presentation	Sunum
Layer 5	Session	Oturum
Layer 4	Transport	Taşıma
Layer 3	Network	Ağ
Layer 2	Data Link	Veri Bağlantısı
Layer 1	Physical	Fiziksel

OSI MODELİ

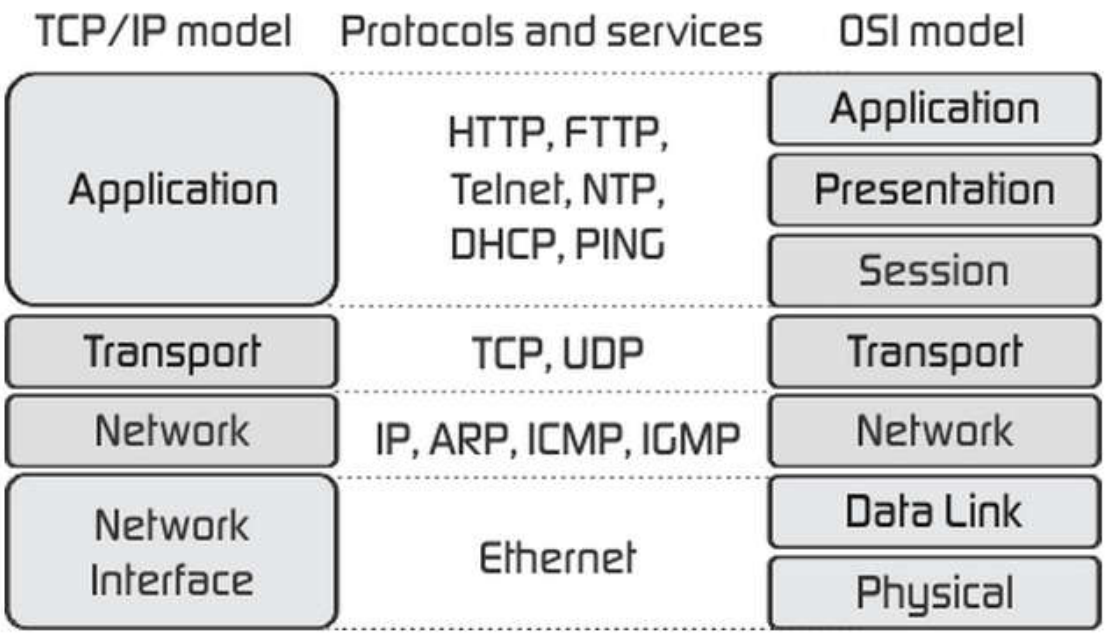
- Layer 1 (Fiziksel Katman): Topolojide yer alan medya (kablolu/kablosuz) kullanılarak verilerin 1 ve 0 olarak iletilmesinden ve alınmasından sorumlu olan katmandır.
- Layer 2 (Data Link): Veri çerçevelerini cihazlar arası aktarımdan sorumludur. Çerçeveleri oluşturma, fiziksel adresleme, hata düzeltimi gibi görevleri üstlenir.
  - MAC adresleri kullanılır.

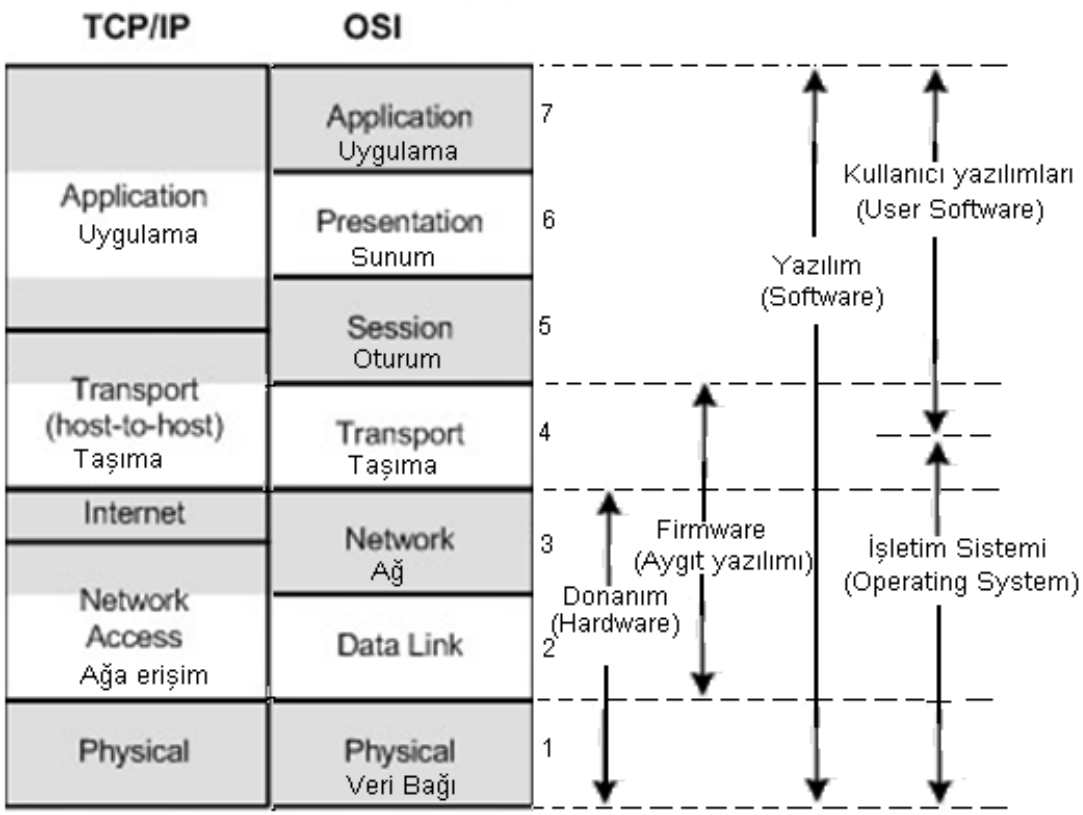
OSI MODELİ

- Layer 3 (Network): Ağlar arası verilerin iletilmesinden sorumludur.
  - IP adresi
  - Yönlendirme ve yol seçimi yapar
- Layer 4 (Transport): Gönderilen ve alınan verileri parçalara (segment/datagram) ayırarak aktarılmasından ve tekrar birleştirilmesinden sorumludur.
  - TCP / UDP

OSI MODELİ

- Layer 5 (Session): Gönderen ve alan cihaz arasında oturumların kurulmasından ve yönetilmesinden sorumludur.
- Layer 6 (Presentation): Verilerin sıkıştırılmasından, şifrelenmesinden, formatlandırılmasından, kodlanmasından sorumludur
- Layer 7 (Application): Verilerin ham halinin oluşturulduğui, kullanıcının ve kodların etkileşim içinde olduğu katmandır.


TCP/IP model
Protocols and services
OSI model
Application
HTTP, FTTP, Telnet, NTP, DHCP, PING
Application
Presentation
Session
Transport
TCP, UDP
Transport
Network
IP, ARP, ICMP, IGMP
Network
Network Interface
Ethernet
Data Link
Physical


TCP/IP
OSI
Application Uygulama
Transport (host-to-host) Taşıma
Internet
Network Access Ağa erişim
Physical
Application Uygulama
Presentation Sunum
Session Oturum
Transport Taşıma
Network Ağ
Data Link
Physical Veri Bağı
7
6
5
4
3
2
1
Kullanıcı yazılımları (User Software)
Yazılım (Software)
Firmware (Aygıt yazılımı)
Donanım (Hardware)
İşletim Sistemi (Operating System)

OSI Katmanları Arasında Veri Aktarımı

Terminal A

İşlem Gönderimi

Veri

Terminal B

İşlem Alımı

UB

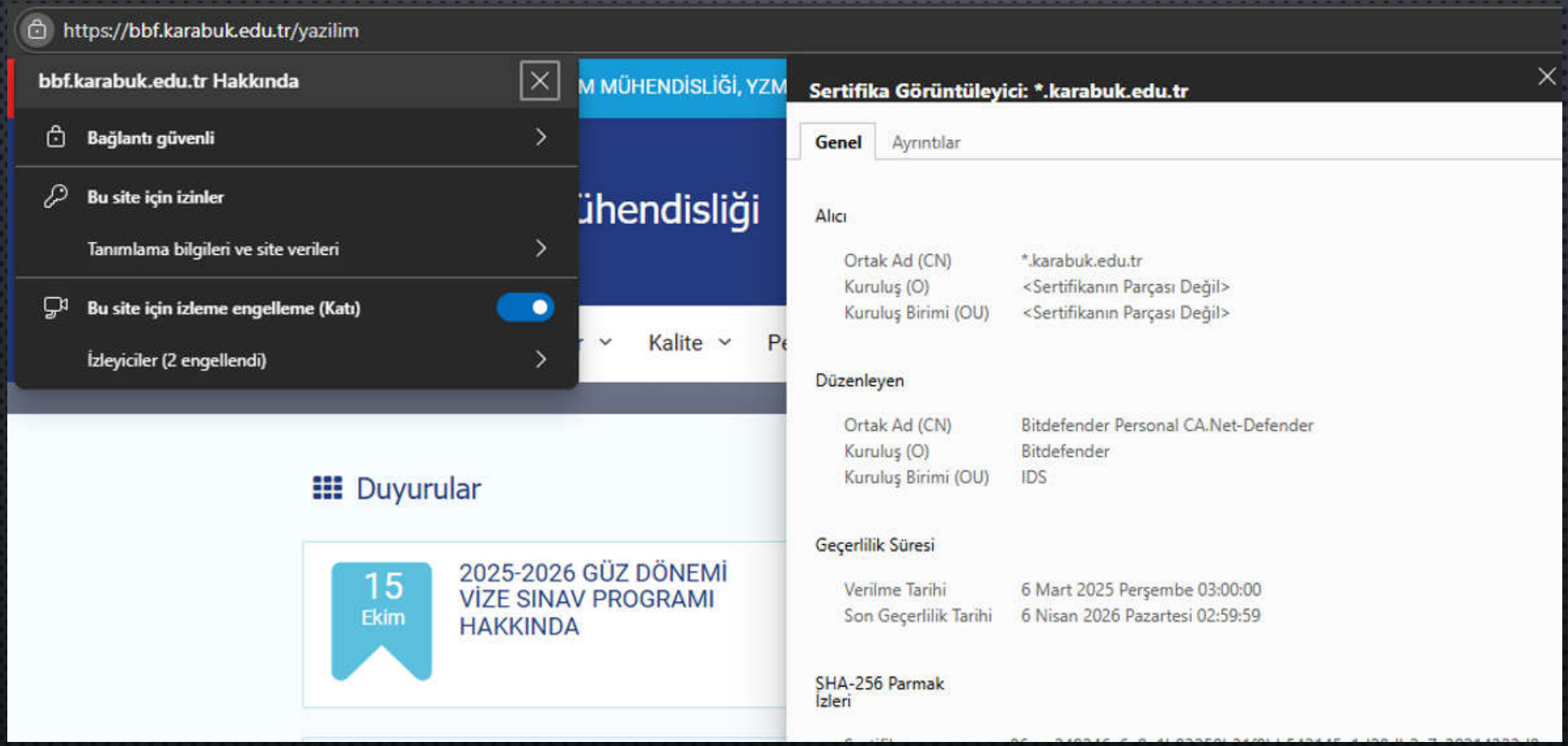
7	**Uygulama**	Veri	UB					7 **Uygulama**
6	**Sunum**	Veri	UB	SB				6 **Sunum**
5	**Oturum**	Veri	UB	SB	OB			5 **Oturum**
4	**Taşıma**	Veri	UB	SB	OB	TB		4 **Taşıma**
3	**Ağ**	Veri	UB	SB	OB	TB	AB	3 **Ağ**
2	**Veri iletim**	VK	Veri	UB	SB	OB	TB	AB VB 2 **Veri iletim**
1	**Fiziksel**	0111001111011110011110011101111 01						1 **Fiziksel**

Fiziksel veri aktarımı; Kablolar vb...

HYPER TEXT TRASFER PROTOKOL

- HTTP: GÖNDERİLEN TÜM VERİLER HERKESİN OKUYABİLECEĞİ ŞEKİLDE AÇIK METİN OLARAK İLETİLİR

- HTTPS: HTTP Secure kısalması olan HTTPS veriler SSL yada benzer protokol ve yapılarla şifreli olarak iletilir.



HTTP

- HTTP protokolü uygulamaların ağ üzerinden etkileşim kurabilmesi için yaygın kullanılan iletişim protokollerinden biridir.
- İstek/yanıt temelinde çalışır. İstemci sunucuya bilgi gönderir, sunucu istemciye yanıt döner.
- HTTP iletişiM karşı bilgisayar oturumu devam ettiği sürece iki cihaz arasında bilgi akışı devam eder. Sunucu yaptığı iş dışında istemciden veri talep etmez/etmemelidir.
- HTTP isteğini URL de belirtilen portlara yapılabildiği gibi, belirtilmeden 80,8080,443 gibi default noktalara da iletilebilir.
- HTTP genellikle TCP kullanır (HTTP 1.0,HTTP 1.1,HTTP/2).

HTTP BİLEŞENLERİ

- OPTIONS: Sunucunun desteklediği metodların listesini döner.
- CONNECT: Bağlantı isteği, sunucu ile istemci arasında TCP/IP tüneli yani bağlantı oluştulmasını sağlar.
- GET: Kaynaktan gönderilen sorguya karşılık cevap beklenen isteklerdir. İçeriğinde parametreler gönderilir. Aynı parametreler URL kısmında da gönderilebilir lakin URL açık olduğundan URL içindeki sorgular; cihaz arasındaki tüm istasyonlarda okunabilir.
  - Önbelleğe alınabilir
  - Tarayıcı geçmişinde kalır
  - Hassas veriler kullanılmamalıdır.
  - Uzunluk kısıtlaması vardır (2048 karakter)

HTTP BİLEŞENLERİ

- HEAD: Get ile benzer şekilde cevap beklenen isteklerdir. Genellikle kaynağın var olup olmadığını anlamak için kullanılır. Cevaplarda gövde bulunmaz.
  - Bir diğer kullanımı; cevabın dönüş tipi öğrenilebilir.
- POST: Kaynağa veri gönderme ve yazma işleminde kullanılır.
- PUT: POST ile benzerdir. Varolan içeriğin güncellemesinde kullanılır.
- DELETE: Kaynaktaki verinin silinmesi işleminde kullanılır.
- PATCH: Kaynaktaki verinin küçük çaplı yada belli bir kısmının güncellenmesi işleminde kullanılır.

HTTP BİLEŞENLERİ

- GET,PUT,POST ve DELETE yöntemleri sunucu tarafinda tanimlanmadığı sürece çalıştırılmaz yada yorumlanmazlar
- HTTP yanıtları ile birlikte içerisinde durum kodları da geri gönderilir. Yapılan işlemin başarı olup olmadığı, olası bir durumun oluşup oluşmadığı durum kodlarına bakılarak anlaşılabilir.

HTPP STATUS KODLARI

- Durum kodları 5 kategoride toplam uzunluğu 3 hane olacak şekilde belirlenmiştir.
- Yazılımcı tarafından belirli durumlarda uygun durum kodlarının gönderilmesi gerekmektedir.
- 1. Kategori 1 ile başlayan kodlar
  - Bilgi vermek amaçlı bu kodlar kullanılmalıdır.
  - Örneğin 100 Continue, 101 Switching Protocols
- 2. Kategori 2 ile başlayan kodlar
  - Yapılan işlemin başarılı olduğu ile ilgili kullanılan kodlar.
  - Örneğin 200 OK, 201 Created, 202 Accepted

HTPP STATUS KODLARI

- 3. Kategori 3 ile başlayan kodlar
  - Yönelendirme ile ilgili kodlar.
  - Örneğin 301 Moved Permanently, 307 Temporary Redirect, 308
- 4. Kategori 4 ile başlayan kodlar
  - İstemci/istem hataları ile ilgili kodlar.
  - Örneğin 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 NotFound
- 5. Kategori 5 ile başlayan kodlar
  - Sunucu/hizmet hataları ile ilgili kodlar.
  - Örneğin 500 internal server error, 502 bad gateway, 503 service unavailable

HTPP MESAJI

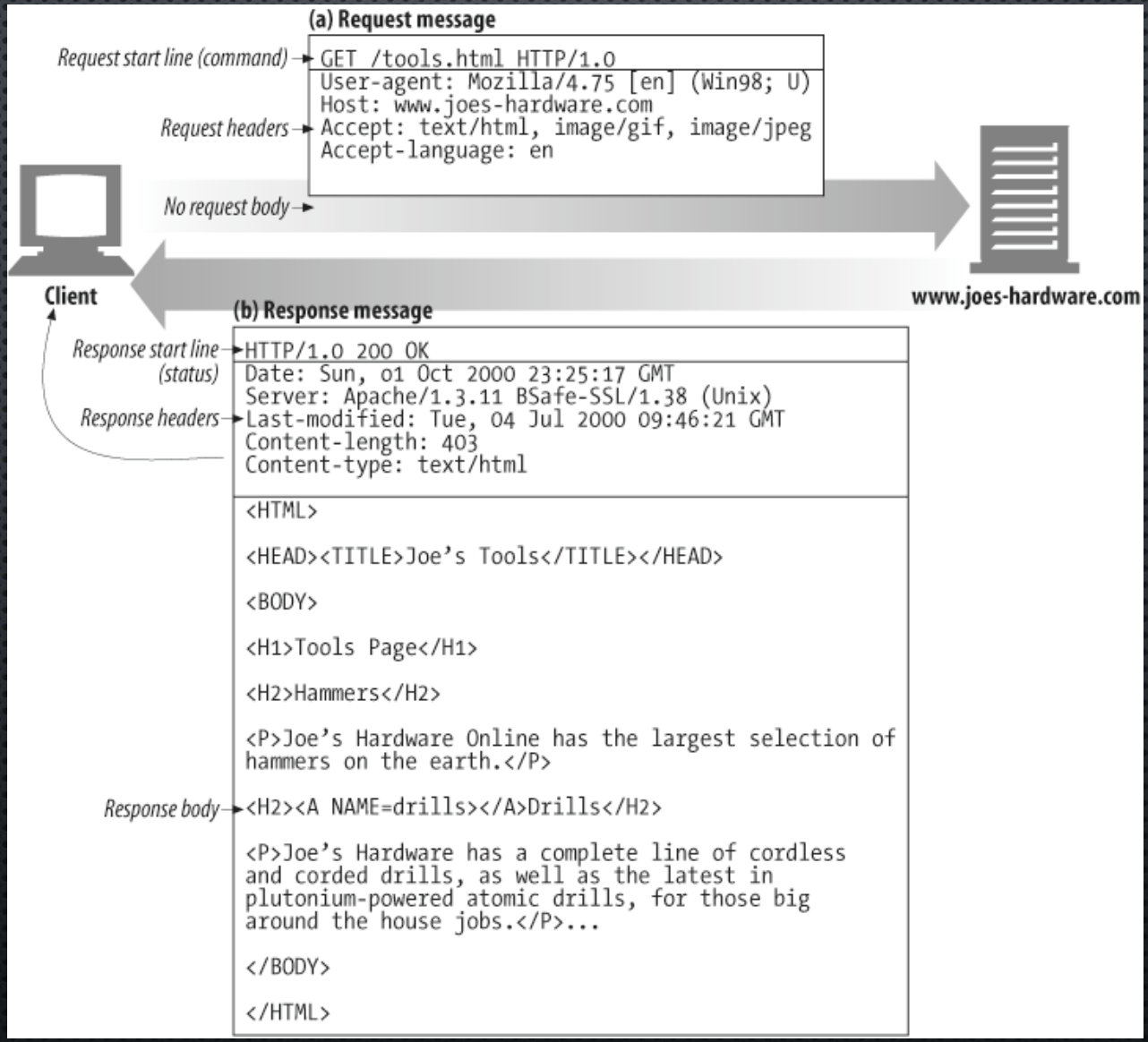
```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 10 Aug 2016 13:17:18 GMT
Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"
Keep-Alive: timeout=5, max=999
Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT
Server: Apache
Set-Cookie: csrftoken=……
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding
X-Frame-Options: DENY

(body)
```

Response headers

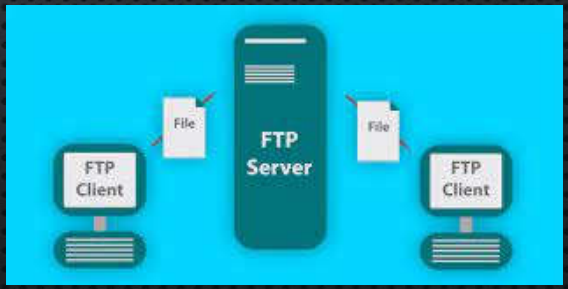
Representation headers

General headers


(a) Request message
Request start line (command)
GET /tools.html HTTP/1.0
Request headers
User-agent: Mozilla/4.75 [en] (Win98; U)
Host: www.joes-hardware.com
Accept: text/html, image/gif, image/jpeg
Accept-language: en
No request body
Client
www.joes-hardware.com
(b) Response message
Response start line (status)
HTTP/1.0 200 OK
Response headers
Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.11 BSafe-SSL/1.38 (Unix)
Last-modified: Tue, 04 Jul 2000 09:46:21 GMT
Content-length: 403
Content-type: text/html
Response body
<HTML>
<HEAD><TITLE>Joe's Tools</TITLE></HEAD>
<BODY>
<H1>Tools Page</H1>
<H2>Hammers</H2>
<P>Joe's Hardware Online has the largest selection of hammers on the earth.</P>
<H2><A NAME=drills></A>Drills</H2>
<P>Joe's Hardware has a complete line of cordless and corded drills, as well as the latest in plutonium-powered atomic drills, for those big around the house jobs.</P>...
</BODY>
</HTML>

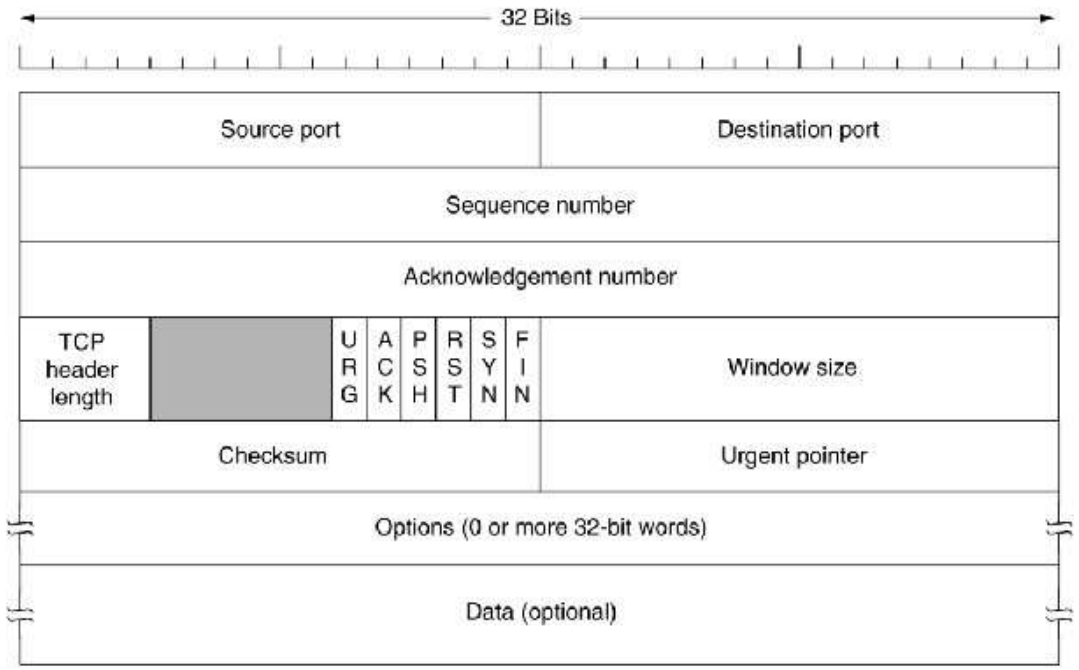
FİLE TRANSFER PROTOCOL

- DOSYA AKTARIM PROTOKOLÜ
- SERVER VE HOST ARASINDA DOSYA TRANSFERİNİ SAĞLAR.
- BİR ÇOK İŞLETİM SİSTEMİNDE DEFAULT DESTEKLİ OLARAK GELMEKTEDİR.
- FTP:// ETİKETİ KULLANILIR.
- GET / PUT İSTEKLERİ KULLANILIR



TRANSMİSSİON CONTRO

- GEÇİŞ/AKTARIM KONTROL PROTOKOLÜ
- TCP PROTOKOLÜ 20 BYTA DAN OLUŞAN, GÜVENİLİR VERİ AKTARIMI SAĞLAYAN İLETİŞİM PROTOKOLÜDÜR.
- BAĞLANTI TEMELLİ ÇALIŞIR.
- VERİLERİN KARŞI TARAFA ULAŞMASINI GARANTİ EDER



TCP header length in 32 bit words,
URG-urgent, ACK- ack number is valid, PSH-push, RST-reset connection,
SYN-used to establish connection, FIN-used to release connection

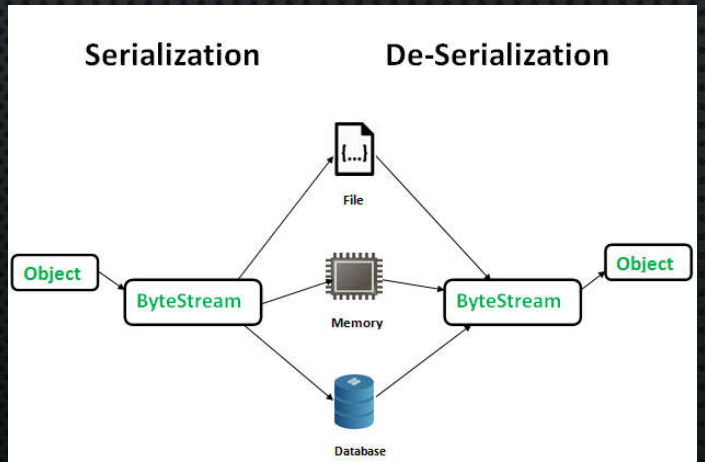
UDP

- Daha hızlı veri aktarım sağlamak üzere UDP protokolü geliştirilmiştir.
- TCP ye göre güvenilir olmayan aktarım protokolüdür.
- Ses, görüntü, VOIP, streaming gibi uygulamalarda tercih edilir.
- Paketlerin gidip gitmediği kontrol edilmez.
- TCP e göre daha kısa başlık dosyası içerir.
- DNS,SNMP,DHCP,RIP,TFTP protokolleri UDP kullanan protokoller
- Parçalanmış UDP veri parçalarına datagram ismi verilir.

SERİALİZATİON / DESERİALİZATİON

SERİALİZATİON / DESERİALİZATİON

- NETWROK ÜZERİNDEN GÖNDERİLEN VERİLERİN KARŞI TARAFA İLETİLMESİ VE KARŞI TARAFTA BU VERİ VE/VEYA VERİ YAPISININ TEKRAR OLUŞTURULMASI GEREKMEKTEDİR.
- DEĞİŞKEN, NESNE VB YAPILARIN BYTE'A DÖNÜŞTÜRÜLMESİNE SEİRALİZATİON OLARAK İSİMLENDİRİLMEKTEDİR.
- BYTE'LARIN DEĞİŞKEN, NESNE VB. YAPILARA DÖNÜŞTÜRÜLMESİNE DESERİALİZATİON OLARAK İSİMLENDİRİLMEKTEDİR.



SERİALİZATİON / DESERİALİZATİON

- BU İŞLEM 2 FARKLI TÜRDE YAPILABİLİR.
  - BİNARY SERİALİZATİON: VERİ YAPISI BELİRLİ BİR FRAMEWORK TARAFINDAN OKUNABİLECEK ŞEKİLDE 1 VE 0 LARA ÇEVRİLİR VE BYTE OLARAK SAKLANIR.
  - TEXT SERİALİZATİON: VERİ YAPISI FRAMEWORKLERİN VE İNSANLARIN OKUYABİLECEĞİ ŞEKİLDE KARAKTERLERE ÇEVRİLİR VE BYTE OLARAK SAKLANIR.

```
0x00: 00 01 00 00 00 FF FF FF FF 01 00 00 00 00 00 00  .....ÿÿÿÿ.......
0x10: 00 0C 02 00 00 00 42 5F 57 6F 72 6B 53 70 61 63  ......B_WorkSpac
0x20: 65 5F 2C 20 56 65 72 73 69 6F 6E 3D 31 2E 30 2E  e_, Version=1.0.
0x30: 30 2E 30 2C 20 43 75 6C 74 75 72 65 3D 6E 65 75  0.0, Culture=neu
0x40: 74 72 61 6C 2C 20 50 75 62 6C 69 63 4B 65 79 54  tral, PublicKeyT
0x50: 6F 6B 65 6E 3D 6E 75 6C 6C 05 01 00 00 00 0F 53  oken=null......S
0x60: 74 61 63 6B 4F 76 65 72 46 6C 6F 77 2E 41 02 00  tackOverFlow.A..
0x70: 00 00 1B 3C 53 6F 6D 65 53 74 72 69 6E 67 3E 6B  ...<SomeString>k
0x80: 5F 5F 42 61 63 6B 69 6E 67 46 69 65 6C 64 1A 3C  __BackingField.<
0x90: 53 6F 6D 65 56 61 6C 75 65 3E 6B 5F 5F 42 61 63  SomeValue>k__Bac
0xA0: 6B 69 6E 67 46 69 65 6C 64 01 00 08 02 00 00 00  kingField.......
0xB0: 06 03 00 00 00 03 61 62 63 7B 00 00 00 0B        ......abc{....
```

```
user
  -L9K-8AF9fxZfe6bwTjk
    Done: false
    ID: "323642ac-89e2-4f14-b9b0-e9b7306aee
    Name: "ufhd"
    Notes: "nvnv"
  -L9K07oNjtvPQYbdyItU
    Done: false
    ID: "c7a53ad2-533f-454c-9ac0-5b21c3d266
    Name: "abc"
    Notes: "cde"
  -L9K1n1r7Q7ZIgA3KFaD
    Done: false
    ID: "938bd62f-f117-4733-a1e4-a4ed22233c
    Name: "ffef"
    Notes: "dfg"
```

XML

<tag> value </tag>

- Extensible Markup Language XML
- Doküman, belge ve verileri temsil edebilir.
- Html benzer şekilde yazılır.
- Açılan her etiket mutlaka kapatılmalıdır.
- Verilerin özellikleri temsil edilebilir.

```
<ogrenciler>
    <ogrenci sinif="1">
        <ogrNo type="int">220000101</ogrNo>
        <adSoyad>Ahmet Mehmet</adSoyad>
    </ogrenci>
    <ogrenci sinif="1">
        <ogrNo type="int">220000102</ogrNo>
        <adSoyad>Ayşe Fatma</adSoyad>
    </ogrenci>
    <ogrenci sinif="2">
        <ogrNo type="int">220000103</ogrNo>
        <adSoyad>Ömer Burak</adSoyad>
    </ogrenci>
</ogrenciler>
```

JSON

"anahtar" : value

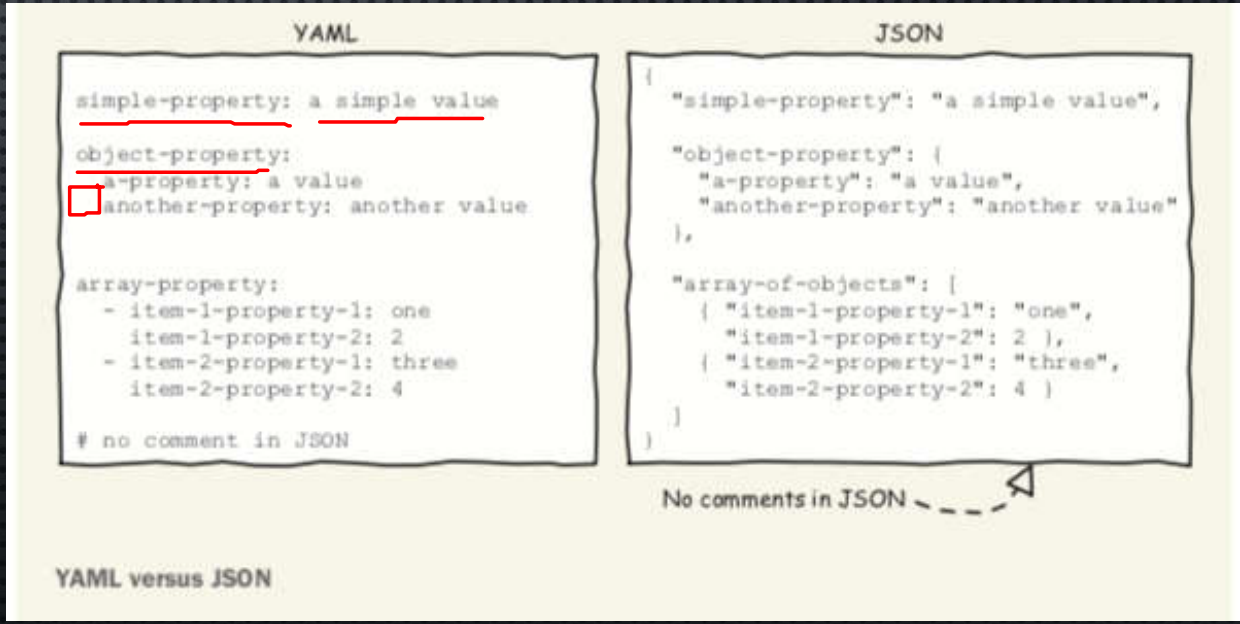
ogrNo,

- JavaScript Object Notation (JSON)
- XML ile yakın zamanlarda çıkmıştır.
- Boyutu XML e göre daha küçük olabilir.
- Tip yada tür dönüşümlerine bağımlı değildir.

```
[
    {
        "ogrNo":220000101,
        "adSoyad":"Ahmet Mehmet"
    },
    {
        "ogrNo":220000102,
        "adSoyad":"Ayşe Fatma"
    },
    {
        "ogrNo":220000103,
        "adSoyad":"Ömer Burak"
    }
]
```

YAML

- XML ve JSON dan sonra çıkmıştır.
- Alt özellik ve nitelikleri girinti olarak ifade edilir.
- Yorum satırı eklenebilir.
- XML ve JSON kullanılan hemen hemen her yerde kullanılabilir.
- Daha çok config dosyalarının temsilinde başvurulur.



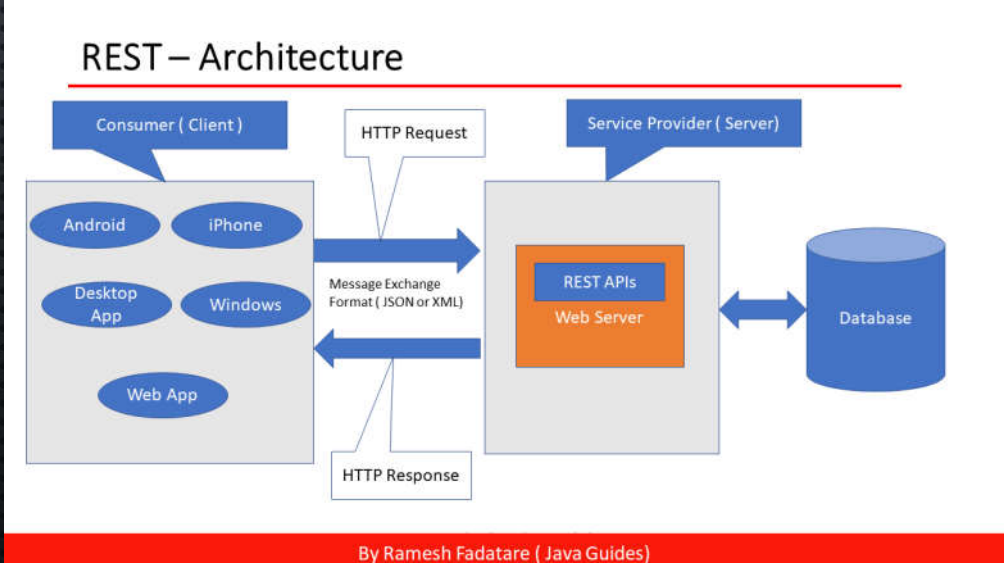
YAML versus JSON

REST/WEB API

- Ortak olarak kullanılan belirli işleyişlerin diğer porgram yada kullanıcılar ile paylaşılması gerekebilir.
- Aynı işi yapan kodların her projede tekrar tekrar yazılmaması için proje içinde yada dışında ayrı bir alana taşınabilir.
- Dışarıya hizmet verilmesi gerekebilir.
- Bu durumlarda Application Program Interface kullanımı tercih edilir.

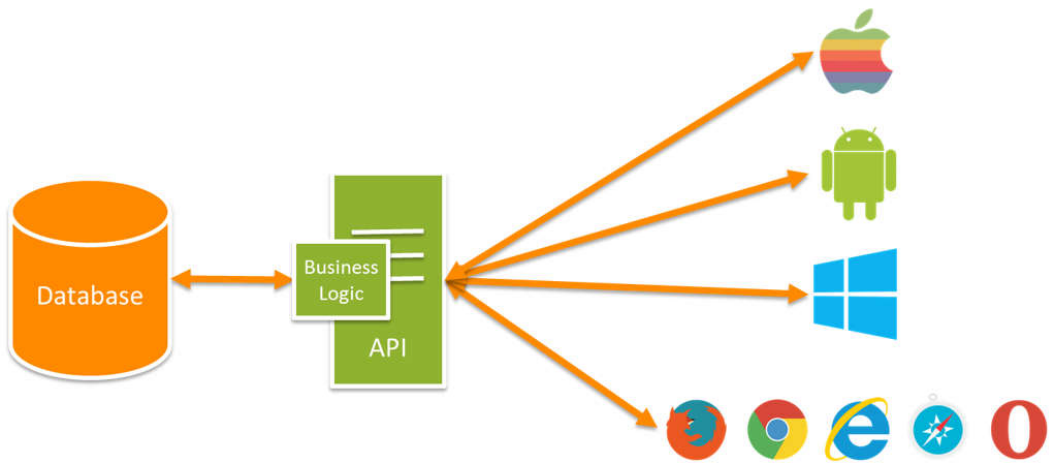
REST API

- REST GENELLİKLE DURUMA DAYALI İŞLEMLERE ODAKLANIR.
- XML/HTML/JSON ÇALIŞABİLİR.
- KATMANLI MİMARİ KULLANIR, İSTEKLERİ KISITLAYABİLİR.
- GENELLİKLE HİZMET SAĞLAMAK İÇİN YAZILIR.
- ENTEGRASYONU HIZLI VE KOLAYDIR.

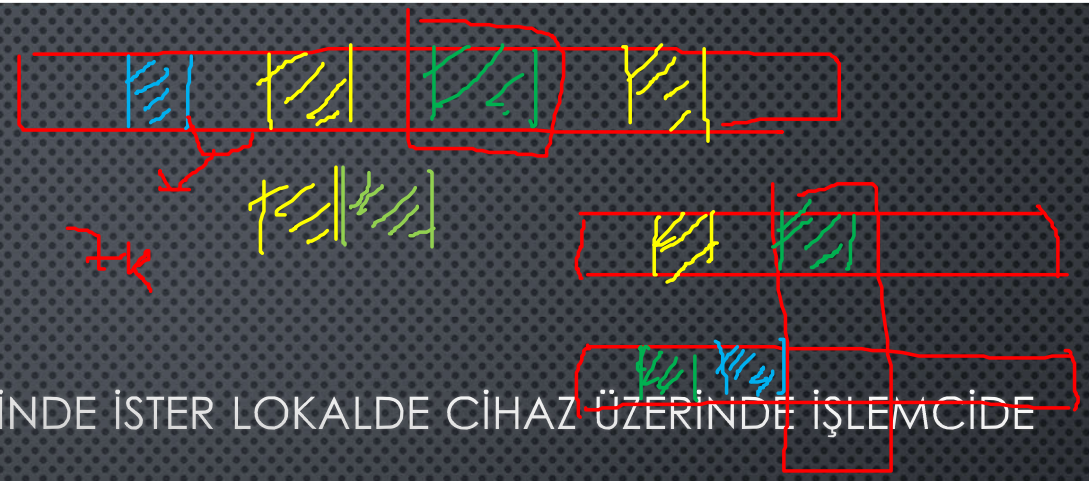


WEB API

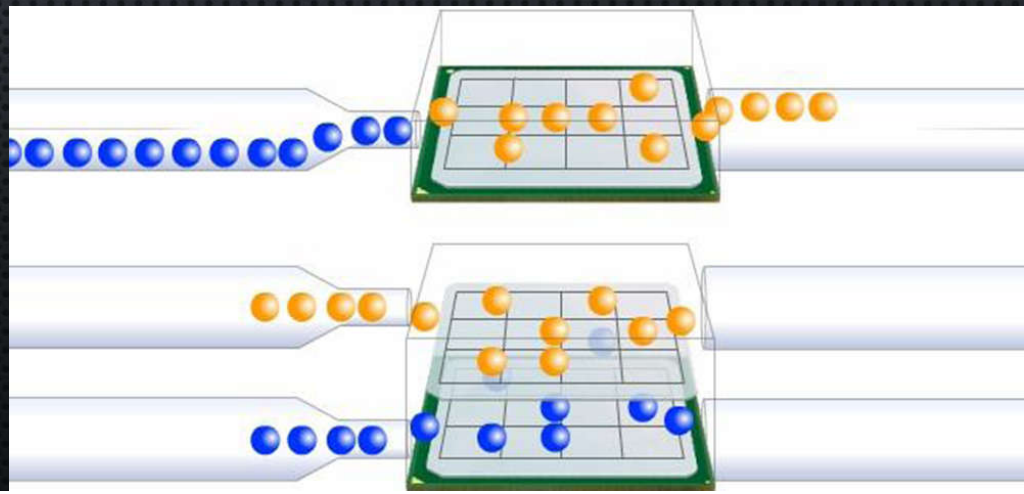
- ORTAK VERİ TEMSİLİNE DAYALI YAPISI VARDIR.
- ESNETİLEBİLİR, GENİŞLETİLEBİLİR.
- GENELLİKLE JSON KULLANIR. (JSON/HTML/XML)
- YAPISI BASİT OLDUĞUNDAN BİRÇOK CİHAZDA KULLANILABİLİR.
- GENELLİKLE VERİ PAYLAŞIMI İÇİN YAZILIR.
- HTTP/S VE URL ÜZERİNDEN ÇALIŞIR.



THREAD/THREADİNG



- PROGRAMLAR İSTER NETWORK ÜZERİNDE İSTER LOKALDE CİHAZ ÜZERİNDE İŞLEMCİDE ÇALIŞTIRILIRLAR.
- İŞLEMCİ ŞAYET TEK ÇEKİRDEKLİ İSE AYNI ANDA YALNIZCA BİR İŞLEM GERÇEKLEŞTİREBİLİR.
- BİLGİSAYARDA BİRDEN FAZLA PROGRAMIN EŞ ZAMANLI OLARAK ÇALIŞTIRILABİLMESİ İÇİN THREADİNG YAPISI KULLANILMAKTADIR.



THREAD/THREADİNG

- Çalıştırılan her program ayrı bir thread olarak çalıştırılır.
- Dış kaynaklardan veri okurken yada yazarken, bu verileri işlerken, kaynak hazır değilse/ yada kullanımda ise programda bekleme/gecikme/donmalar yaşanabilir.
- Şayet program birden fazla uç noktaya hizmet sağlıyorsa bu bekleme/gecikme/donmalar diğer hizmetlerin yada işleyişlerin aksamasına neden olabilir.
- Bunu engellemek için özellikle programlada threadİng yapısı sıklıkla kullanılmaktadır.
- Hatta bazen bunun için kanallar bile tasarlanıp optimize edilebilir.