

## Dağıtık Sistemler Ders Notları

**Kaynak:** Andrew S. Tanenbaum and M. V Steen,  
*Distributed Systems: Principles and Paradigms, 2<sup>nd</sup> Ed.*,  
Pearson, 2007

**İkincil Kaynak:** George Coulouris, Jean Dollimore and  
Tim Kindberg, *Distributed Systems: Concepts and  
Design, 4th Ed.*, Pearson, 2005

# Giriş

**İletişim Ağı:** Birbirleriyle iletişim halindeki birimlerin kümesi  
Her yer bilgisayar ağı!

- Mobil telefon ağları, Kurumsal ağlar, Fabrika ağları, Kampus ağları, Ev ağları, Araç-içi ağlar, Uçak ve trenlerdeki yerleşik ağlar

## Dersin Amacı:

- dağıtık sistemler/uygulamalarda temel kavramlar,
- bu sistemlerin belirleyici özellikleri, avantajları
- Mimari yapılar, client(istemci)-server(sunucu)
- görevler arası iletişim, socket programlama ile DS
- Ortakatman ve dağıtık nesnelere (Sun RPC, Java RMI),
- İsimlendirme ve ağ hizmetleri,
- Senkronizasyon ve Mantıksal Saat
- Tutarlılık ve Veri Tekrarı
- dağıtık dosya sistemleri tasarımı ve gerçekleştirimi

# Tanımlar

- A distributed system is a collection of independent computers that appears to its users as a single coherent system.
- *Dağıtık bir sistem, kullanıcılara tek bir uyumlu sistem olarak görünen, bağımsız bilgisayarlar topluluğudur. [Tanenbaum]*
- *Ağdaki bilgisayarlarda bulunan sistem bileşenlerinin yalnızca mesaj göndererek haberleştikleri ve hareketlerini koordine ettikleri bir sistem. [Coulouris]*

## Tanımda öne çıkanlar:

- DS, otonom bileşenlerden (bilgisayarlar gibi) oluşuyor.
- Kullanıcılar(program/insan), tek bir sistemle muhatap. O halde bileşenler eşgüdüm içinde olmalı.

**Örnek:** Bir bölüm ağındaki birkaç workstation + işlemciler. Eğer sistem bir bütün olarak, tek işlemcili timesharing sistem gibi duruyor ve öyle de davranıyorsa, bu distributed system olarak değerlendirilir.

# Tanım (devam)

1)Kullanıcılardan **gizlenmiş** durumda:

- Çeşitli bilgisayarlar arası farklılıklar
- Bunların nasıl haberleştiği (communicate)
- DS'nin iç mimarisi

2) Tektip tip görünüş (Uniform view)

- Zaman ve mekandan bağımsız olarak, kullanıcılar (insan/uygulama) DS ile tutarlı/tek tip (consistent/uniform) bir etkileşim içinde

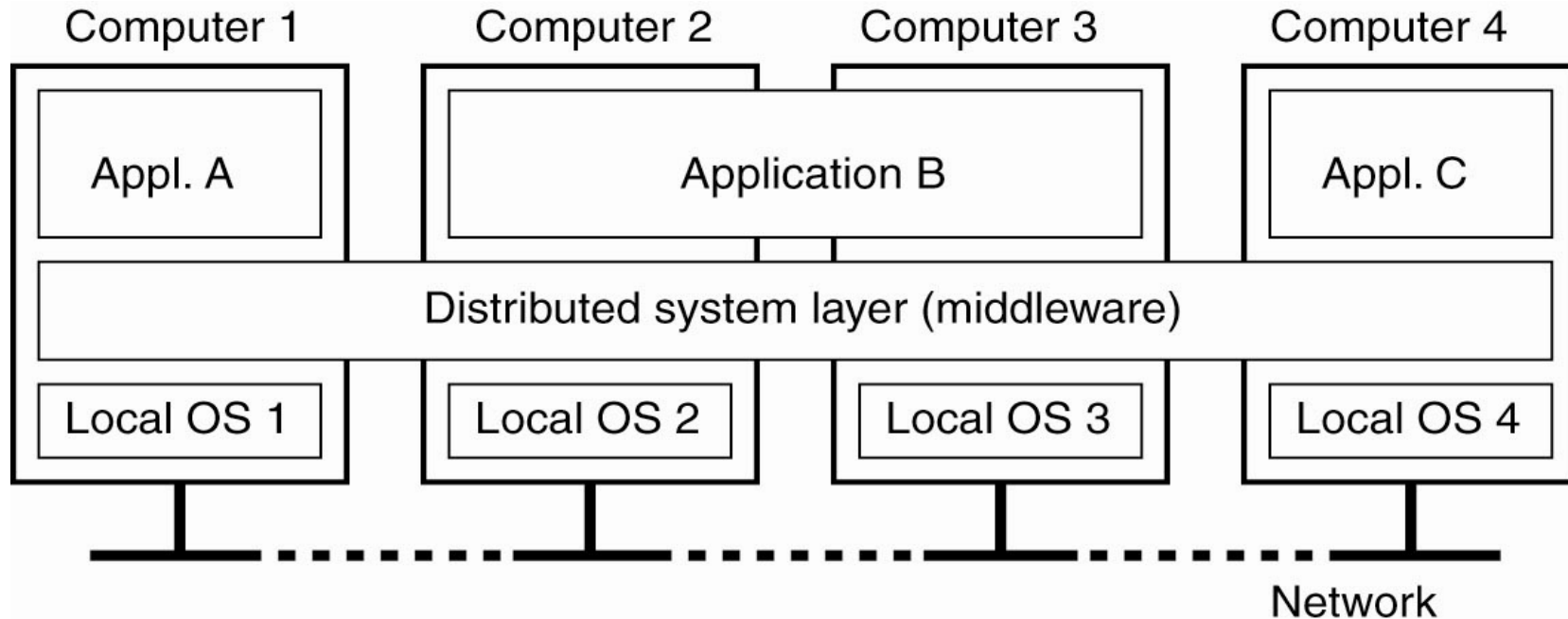
3) Ölçeklenebilir (Scaleable)

- Genişleme/ölçekleme kolaylığı (bağımsız bilgisayarlar ve bunların DS'ye katılımının gizlenmiş olmasının neticesi olarak)

4) Süreklilik (Availability)

- DS sürekli servis sağlar (bazı parçaları/bileşenleri çalışmasa da)

# DS: Yazılım (S/W) katmanı olarak



- DS, iki katman arasında. Bunlar, bir üst katman (users/apps) ile bir alt katman (OS and basic comm facilities)
- Böyle bir DS, **middleware**(orta katman) olarak da anılmakta
- Orta katman çok sayıda makine üzerine yayılmış durumda,
- Her uygulamaya aynı arayüzü (same interface) sunmakta
- Uygulamalardan donanım(H/W)/OS farklılıklarını gizlemekte.

## DS'yi olabilir eden 3 Gelişme (3 Advances)

- Güçlü mikro-işlemci (micro-processor):
  - 8-bit, 16-bit, 32-bit, 64-bit
  - x86 family, 68k family, Alpha chip
  - Clock rate: 4.77MHz up to 4.0 GHz
- Bilgisayar ağı (Computer network)
  - Local Area Network (LAN), Wide Area Network (WAN), MAN, Wireless Network (Wi-Fi), Mobile Network (3G/UMTS)
  - Network type: Ethernet, Token-bus, Token-ring, FDDI, ATM, Fast-Ethernet, Gigabit Ethernet, Fibre Channel
  - Transfer rate: 64 kbps up to 1Gbps
- İkincil Bellek (Secondary storage, Hard Disk)
  - 5-10Mb(85), 100-250Mb(90), 1Gb(93), 4-6Gb(97), 10-20Gb(00), 80-120+Gb.

# DS: Amaçlar

## 4 Önemli Amaç:

### 1. Kaynaklara kolay erişim - Make resources easily accessible

- Users/apps to easy access remote resources: printers, computers, storage facilities, data, files, web pages, networks
- Why? Economics. 1 printer per user vs 1 printer shared by n users
- Share costly resources such as supercomputers, high-performance storage-systems, imagesetters, other expensive peripherals
- Collaborate and exchange information (Internet): files, mail, audio, video
- Virtual organization. Groupware: s/w for collaborative editing, teleconference
- Electronic commerce: buy/sell all kinds of goods

### 2. Kaynakların ağda dağıtık halde bulunduğu gerçeğini sakla

- Hide the fact that DS processes and resources are physically distributed
- A DS that is able to present itself to users / apps as if it were only a single computer system is said to be *transparent*.

# DS: Amaçlar (devam)

## 3. Açıklık (Openness)

- Open system offer services according to standard rules that describe those services' syntax and semantics
- In computer networks, standard rules govern the format, contents, and meaning of msgs sent and received. Such rules are formalized in **protocols**.
- In DS, services are specified through interfaces (described in IDL).
- IDL: captures syntax of services. NOT what services do (semantics)

## 4. Ölçeklenebilirlik (Scalability)

- More users or resources : with respect to size (/geography/admin)
- Services, data and algorithms centralized? Can become a bottleneck!
  - Centralized services: a single server for all users
  - Centralized data: a single on-line telephone book
  - Centralized algorithms: routing based on complete info
- Decentralized alg:partial state, decide on local info, one failure:ok, no g.clock



# Distributed Systems VS. Centralized Systems

## Avantajlar

- Ekonomik: Microprocessors offer a better price / performance than mainframes
- Hız: A dist. system may have more total computing power than a mainframe.
- Uygulama ile uyum: Some apps (banking) involve spatially separated machines
- Güvenilirlik (Reliability): If 5% of the machines are down, the system as a whole can still survive with a 5% degradation of performance.
- Adım adım büyüme: Computing power can be added in small increments
- Veri Paylaşımı (Data sharing): Allow many users access to a common database
- Aygıt (Device) Paylaşımı: Allow many users to share expensive peripherals.
- İletişim: Make human-to-human communication easier-- Email, ICQ.
- Esneklik (Flexibility): İş yükünü var olan makinelere dağıtma kolaylığı

## Dezavantajlar

- Yazılım-Software: Little software exists at present for distributed systems
- Ağ problemi: The network can saturate or cause other problems
- Güvenlik-Security: Easy access also applies to secret data

# Dağıtık Sistemlerin Nitelikleri

## **Paralel işlemler**

- Birlikte çalışan bağımsız bileşenlerden oluşur

## **Mesaj gönderme ile iletişim**

- Ortak bellek yok (No shared memory)

## **Kaynak paylaşımı**

- Veri tabanı, yazıcılar, işleme gücü, veri, diğer servisler

## **Genel sistem durumu belirsizliği (No global state)**

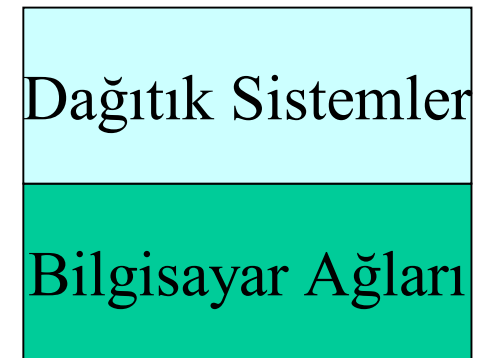
- Hiçbir işlem, sistemin genel durumuna dair tam bilgiye sahip değil

## **Ortak saat yok**

- İşlemler için sadece kısıtlı derecede saat eşitleme mümkün olabilir

# Ağlar VS. Dağıtık Sistemler

- Ağ: Yerel veya geniş alandaki bilgisayarları bağlamak ve protokoller arası mesaj alışverişi yapmak için kullanılan ortam. Ağ birimleri görülebilirdir ve açıkça adreslenmiştir (IP adresleriyle).
- Dağıtık sistem: Birçok bağımsız bilgisayarın varlığı belirgin değildir
- Açıklık, güvenilirlik gibi birçok problem ortaktır, ancak farklı katmanlardadır.
  - Ağlar, paket yönlendirme vs üzerine odaklanırken, dağıtık sistemler uygulamalara odaklanır.
  - Bir dağıtık sistem bilgisayar ağı tarafından sağlanan servisler üzerine bina edilir.



## Network OS, Distributed OS, and Multiprocessor OS

- Network OS:
  - Config I: All stand-alone workstation each with its own CPU, memory, hard disk, and OS, and connected by a LAN.
  - Make use of the following commands: rlogin, rcp, rsh, telnet, ftp.
  - Config II: System like the above but with some dedicated servers -- file servers, application servers, ftp server, mail server, name server, PPP server.
- True Distributed System: Single-system image or virtual uniprocessor concept
  - A single, global interprocess communication mechanism -- do not have to deal with different mechanisms on different machines and deal with local versus remote machines.
  - A global protection scheme -- no mixing of access control list, protection bits and capabilities.
  - Process management is the same everywhere -- how processes are created, destroyed, started, and stopped must not vary from machine to machine.
  - File system must look the same everywhere -- having different filename construct is a nightmare.
  - Same system call everywhere.
  - Kernel will take care of scheduling, memory management, and local resources management -- no need to centralize these.

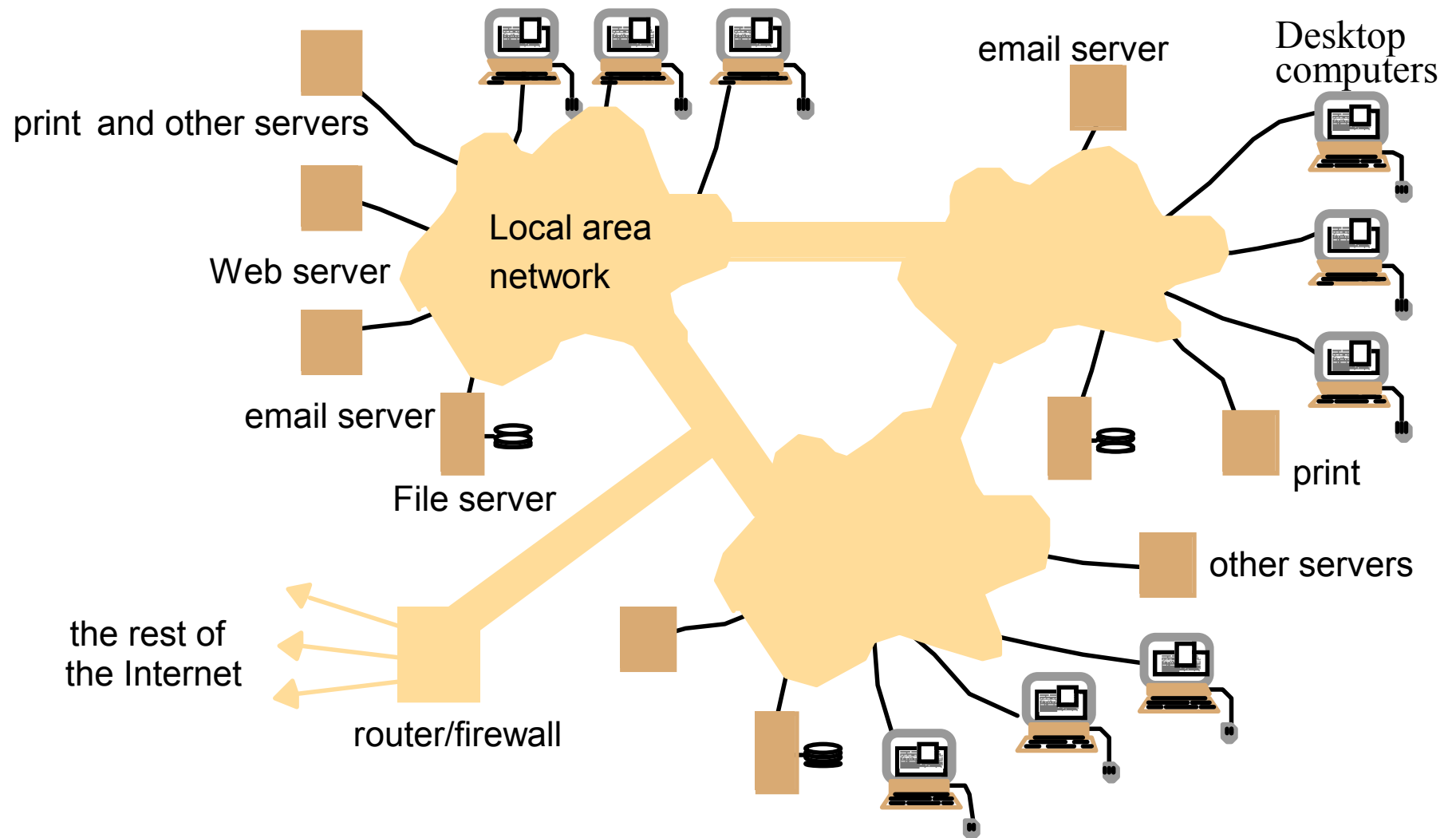
## Network OS, Distributed OS, and Multiprocessor OS (continue)

Item	Network OS	Distributed OS	Multiprocessor OS
Does it look like a virtual uniprocessor?	No	Yes	Yes
Do all have to run the same OS?	No	Yes	Yes
How many copies of OS are there?	N	N	1
How is communication achieved?	Shared files	Messages	Shared memory
Are agreed upon network protocol required?	Yes	Yes	No
Is there a single run queue?	No	No	Yes
Does file sharing have well-defined semantics?	Usually no	Yes	Yes

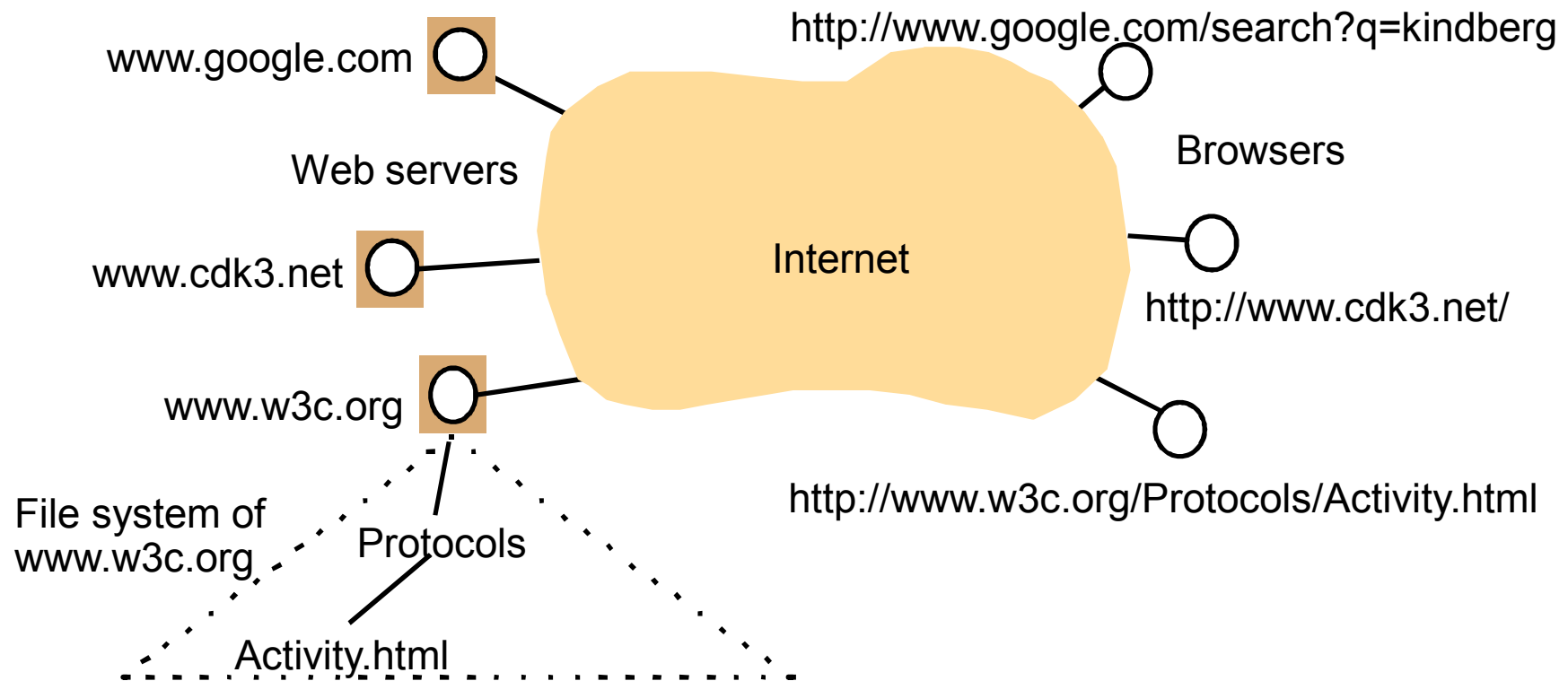
# Örnek Dağıtık Sistemler

- **Internet:**
  - Çeşitli servisler barındıran birçok farklı bilgisayar türlerinden oluşan büyük bir ağ topluluğu
  - Uygulamalar iletişimi mesajlarla sağlar
  - Servisler: www, email, dosya gönderme gibi
- **Intranet: Yerel güvenlik politikası var.**
  - Internetin bir alt parçası,
  - Ayrı bir yönetimi var (organizasyon),
  - Kaynakların ağ içerisinde paylaşımını destekler (dosya/depolama sistemleri ve yazıcılar)
- **Kablosuz ağlar**

# Genel bir intranet



# Kaynak paylaşımı ve Web: açık protokoller, ölçeklenebilir sunucular, takılabilir tarayıcılar





# World Wide Web kaynakları

- Web'de **kaynaklar** ve diğer servisler URL'ler ile isimlendirilir
- Web Sayfaları **paylaşılan kaynaklara** örnektir. İstemci URL ile isteğini belirtir.
- Bu kaynaklar **Web sunucuları (web server, process)** tarafından yönetilir
- **Web, istemci-sunucu** mimarisini kullanır.
  - **Web tarayıcısı:** istemci programı (client process),
  - **Web sunucusu:** istenen web sayfalarını içeren yerel dosyalara (kaynaklara) erişir ve istemciye iletir (server process).
- **URL - Uniform Resource Locator**

# URL Örneği

- URL: **http://www.dcs.qmw.ac.uk/research/distrib/book.html**
- **İlk kısım:** Kullanılacak protokol. (:’dan önceki kısım)
  - protokol http’dir ("HyperText Transport Protocol").
- **İkinci kısım:** Web sunucusunun domain adı. (// ile / arası).
  - domain adı (hostname) : www.dcs.qmw.ac.uk.
- **Kalan kısım:** (Tek ‘/’tan sonraki kısım). O domain üzerindeki kaynağı (dosyayı) gösterir
- Web sunucusunun kullandığı üst dizinden itibaren adlandırılır (pathname). **research/distrib/book.html**

## **Design Issues: a) Transparency(şeffaflık) in DS**

- 1. Access(erişim):** hide differences in data representation and how a resource is accessed
- 2. Location(konum) transparency:** The users cannot tell where resources are located
- 3. Migration(taşınma):** Resources can move at will without changing their names.
- 4. Replication transparency:** The users cannot tell how many copies exist.
- 5. Relocation transparency:** Resources can move to another location while in use.
- 6. Concurrency(Eşzamanlılık):** Multiple users can share resources automatically.
- 7. Parallelism transparency:** Activities can happen in parallel without users knowing
- 8. Failure transparency:** hide the failure and recovery of a resource.

## Design Issues (continue)

- b) Reliability
  - Availability
  - Fault tolerance
- c) Performance
  - Fine-grained parallelism
  - Coarse-grained parallelism
- d) Scalability
  - Potential bottle-necks in very large distributed systems
    - Centralized components: A single mail server for all users
    - Centralized tables: A single on-line telephone book
    - Centralized algorithms: Doing routing based on complete information
  - Use decentralized algorithms. Their characteristics are:
    - No machine has complete information about the system state
    - Machines make decisions based only on local information
    - Failure of one machine does not ruin the algorithm
    - There is no implicit assumption that a global clock exists

# Scaling Techniques: a) Hide comm. delay

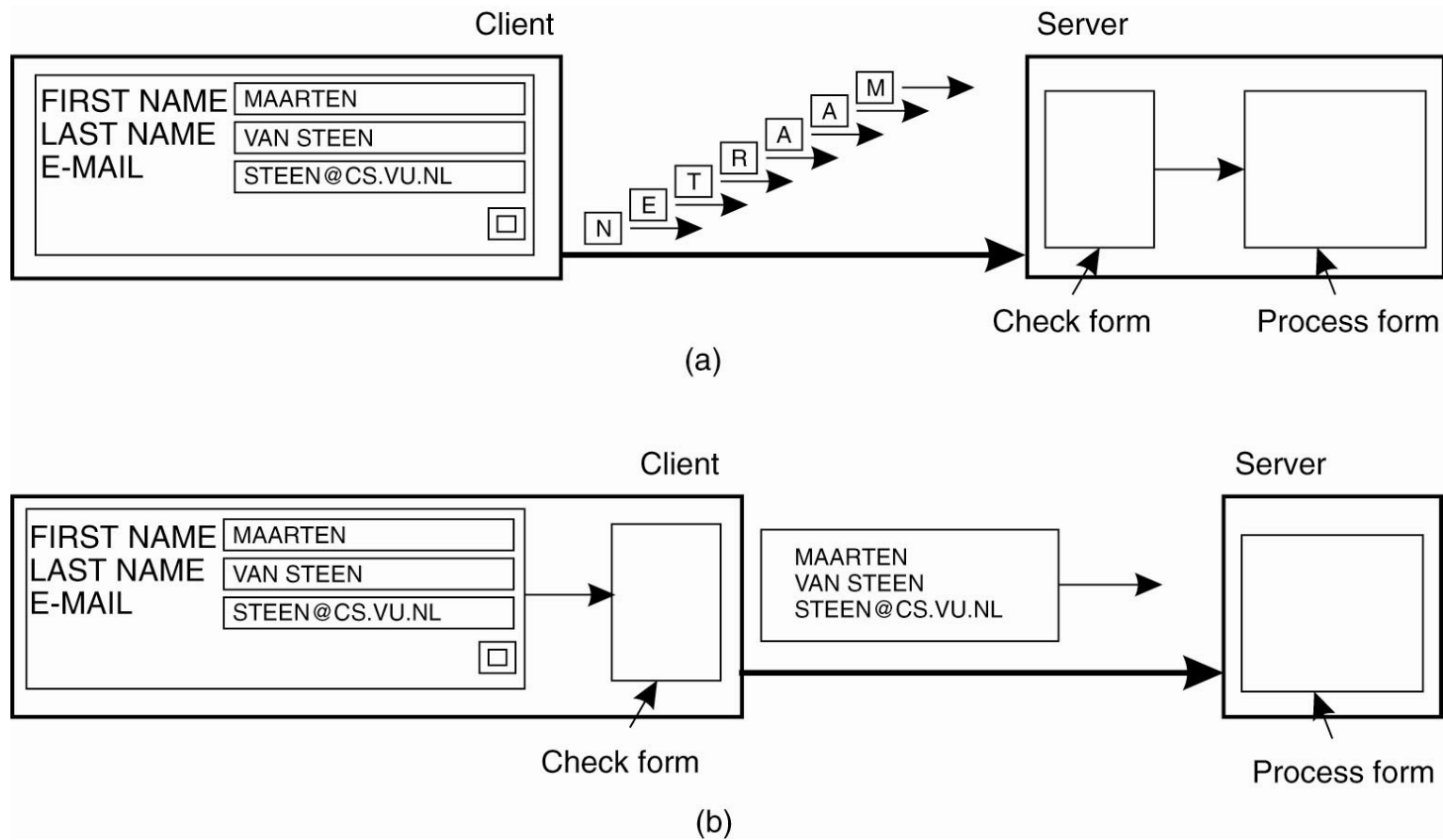
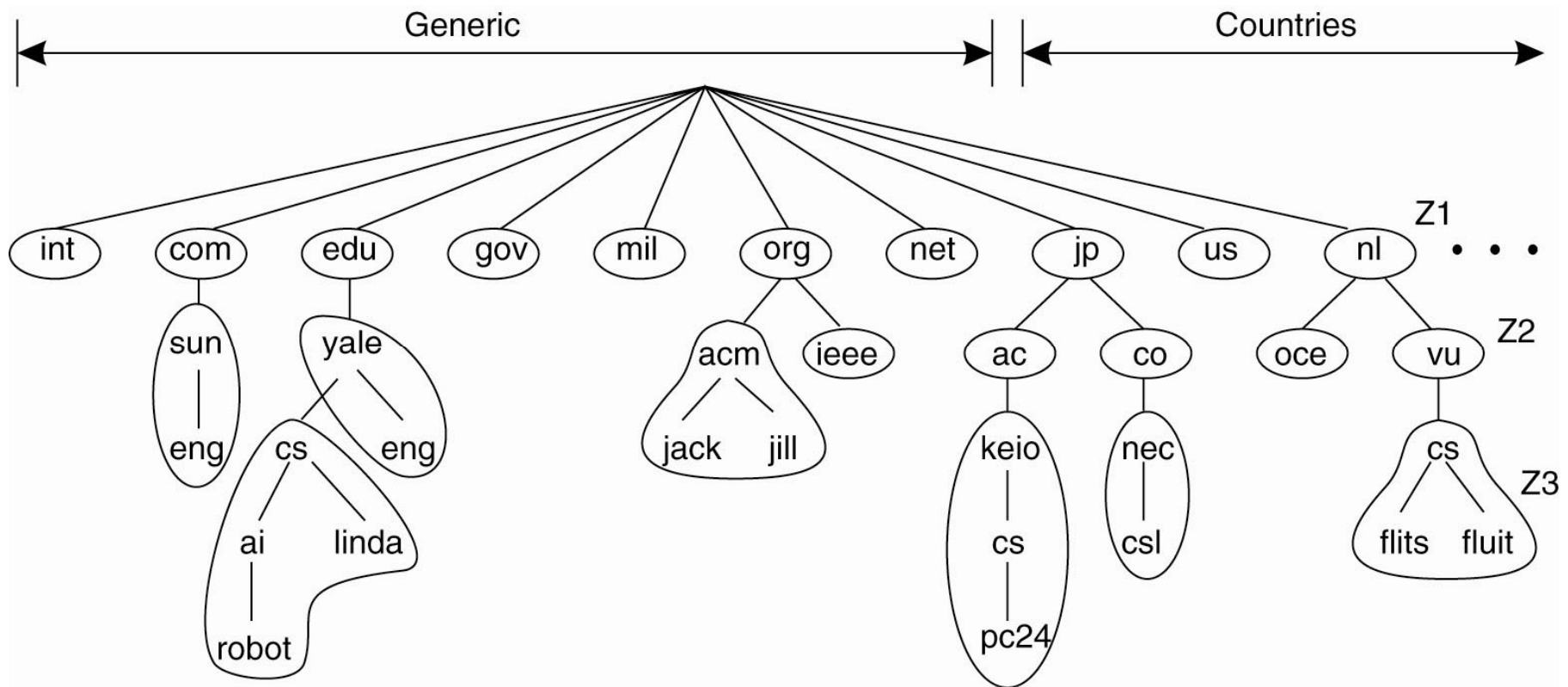


Figure 1-4. The difference between letting (a) a server or (b) a client check forms as they are being filled.

# Scaling Techniques: b) distribution



Split a component into smaller parts, spread parts across the system

Ex: divide DNS name space into a tree of domains, then zones (names in a zone are handled by a name server). Consider name [nl.vu.cs.flits](#)

## Scaling Techniques: c) replication

Replicate components across distributed system

- Make a copy of resource
- Put it near of the client that access resource

Result

- Increase availability
- Balance the load between components
- Better performance
  
- Caching: a form of replication
  - Decided by client, not owner of resource
  - Happens on demand, not planned in advance
- **Both lead to consistency problem**

# DS Uygulama Örneği

- Online kitapçı (World Wide Web üzerinde)
  - Müşteriler kendi bilgisayarlarından sizin bilgisayarınıza (web sunucunuza) bağlanabilir:
    - Stoktaki ürünlere bakabilir
    - Sipariş verebilir
    - ...



# DS Uygulama Örneği – zorluklar I

## ■ Ya ... ?

- Müşteriniz çok farklı bir donanım kullanıyorsa? (PC, MAC,...)
- ... farklı bir işletim sistemi? (Windows, Unix,...)
- ... farklı bir veri gösterim biçimi? (ASCII, EBCDIC,...)
- **Çeşitlilik (networks, h/w, os, prog lan)**
- **Erişim şeffaflığı**

## ■ Veya ... ?

- İşinizi ve bilgisayarlarınızı güneye taşımak isterseniz (hava şartlarından dolayı)?
- Müşterileriniz güneye taşınırsa (daha muhtemel)?
- **Konum / Taşınma şeffaflığı**

# DS Uygulama Örneği – zorluklar II

## ■ Ya . . . ?

- İki müşteri aynı ürünü aynı anda sipariş verirse?
- **Eşzamanlılık/Parallelizm şeffaflığı**

## ■ Veya . . . ?

- Stok bilginizi tutan veri tabanı çökerse?
- Sipariş esnasında müşterinizin bilgisayarını çökerse?
- **Failure transparency (Fault tolerance)**

# DS Uygulama Örneği – zorluklar III

- Ya ... ?
  - Birileri veri çalmak için sisteminize girmeye çalışırsa?
  - ... bilgi çekerse?
  - ... müşteriniz sipariş verir de, sonradan vermedim diyerek ödemeyi reddederse?
  - **Security (Güvenlik)**
- Veya ... ?
  - Öyle başarılı olursunuz da, milyonlarca insan aynı anda online mağazanızı aynı anda ziyaret ederse?
  - **Scalability (Ölçeklenebilirlik)**

# DS Uygulama Örneği – zorluklar IV

- **Sistem kurulurken ...**
  - Bütün yazılımı tek başınıza mı yazmak istersiniz (network, database,...)?
  - Güncellemeler, yeni teknoloji takibi?
  - **Reuse (tekrar kullanılabilirlik) ve Openness (açıklık) (Standartlar)**

# Heterogeneity (Çeşitlilik)

- Çeşitli bileşenler birbiriyle uyumlu şekilde çalışabilmeli
  - İşletim sistemleri
  - Donanım mimarileri
  - İletişim mimarileri
  - Programlama dilleri
  - Yazılım arayüzleri
  - Güvenlik ölçüleri
  - Bilgi gösterimleri

# DS Geliştirirken Yanlış Varsayımlar

False assumptions made by first time developer:

- The network is reliable.
- The network is secure.
- The network is homogeneous.
- The topology does not change.
- Latency is zero.
- Bandwidth is infinite.
- Transport cost is zero.
- There is one administrator.