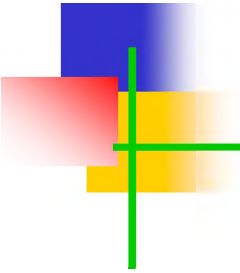


# Dağıtık Hesaplama ve Dağıtık Sistemler

---



# Konular

- Giriş
- Dağıtık Hesaplama Nedir?
- Dağıtık Sistem Nedir?
- Dağıtık Sistemlerin Nitelikleri Nelerdir?
- Örnek Dağıtık Sistemler
- Dağıtık Sistemlerin Zorlukları Nelerdir?
- Özet

# Giriş

- Bilgisayar ağları her alana yayılmıştır.
  - Mobil telefon ağları
  - Kurumsal ağlar
  - Fabrika ağları
  - Kampus ağları
  - Ev ağları
  - Araç-içi ağlar
  - Uçak ve trenlerdeki yerleşik ağlar

# Giriş

- **Dağıtık hesaplama** bilişim biliminde dağıtık sistemleri inceleyen bir bilim dalıdır. **Dağıtık sistem** birden fazla otomatik bilgisayarın bir ağ üzerindeki iletişimidir. Ağdaki bilgisayarlar belirli bir hedefe ulaşmak için birbirleriyle etkileşim içerisindedirler. Dağıtık sistemi çalıştıran bilgisayar programına **dağıtık program** denir. Bu tür programları yazma işlemine **dağıtık programlama** adı verilir.
- Dağıtık hesaplama aynı zamanda dağıtık sistemleri kullanarak hesaplama yöntemiyle problemleri çözmeye denir. Dağıtık hesaplamalarda bir problem birden fazlaya parçaya bölünür ve her bir parça farklı bir bilgisayar tarafından çözülür.

# Dağıtık Hesaplama (Distributed Computing)

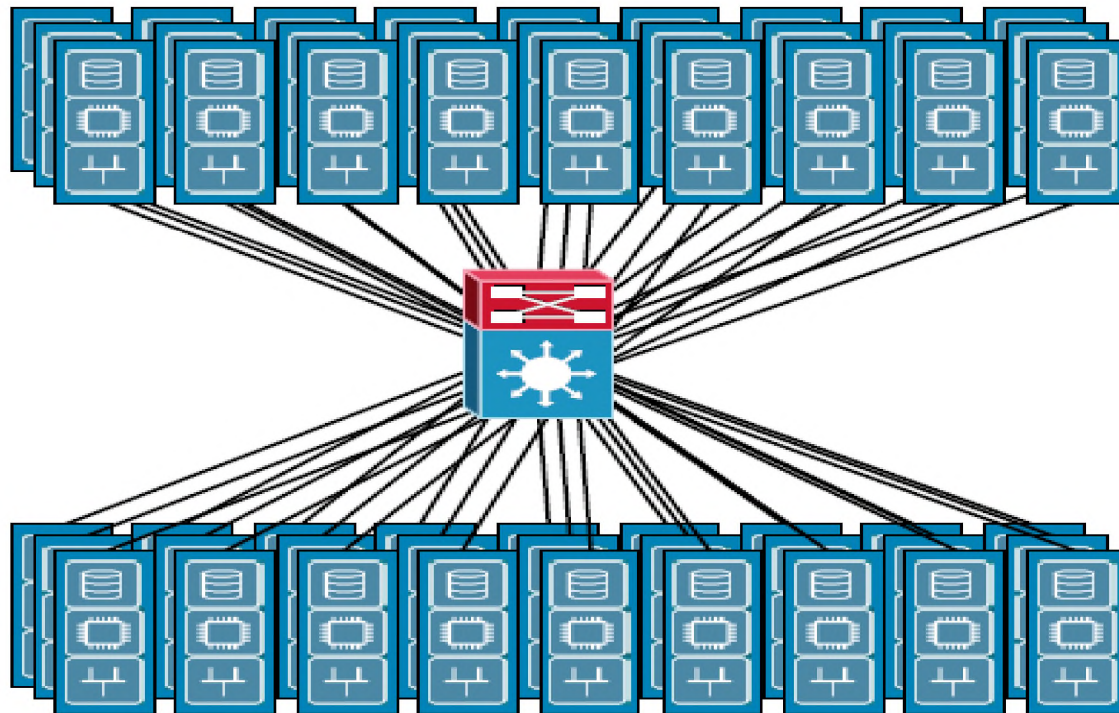
- Dağıtık hesaplama yönteminde işlem parçaları, ağ (network) üzerinde birbirine bağlı işleme ünitelerine dağıtılır. Bağlantı bu şekilde ağ üzerinde olduğundan son derece ölçeklenebilir bir sistemdir. Dağıtık hesaplama sistemleri de kendi içinde üç ana kısma ayrılır:
  - Cluster Computing (Bilgisayar Kümeleri)
  - Massive Parallel Processing (Yüksek Düzeyde Paralel İşleme)
  - Grid Computing (Izgara Hesaplama)

# Cluster Computing

- Bu yöntemde bilgisayarlar ağ üzerinde birbirine bağlanarak tek bir bilgisayar gibi çalıştırılırlar. Bu yöntemle birleştirilen bilgisayarların her biri bu ağdan bağımsız da çalışabilen (standalone) bilgisayarlardır. Bilgisayar kümelerinin en çok kullanılan tipi Beowulf kümeleridir. Beowulf kümelerinde günlük hayatta alışık olduğumuz bilgisayar kasaları, ethernet kullanılarak TCP/IP üzerinden yerel alan ağı ile birbirine bağlanır. Dünyanın en hızlı ilk 500 bilgisayarı içinde en çok kullanılan yöntem budur. Hesaplama küme bilgisayar kullanımı 1994 senesinde NASA'da Beowulf projesi ile başlamıştır. 16 Intel 486 DX4 işlemci ethernet ile bağlanmıştır.

# Cluster Computing

- Yüksek performanslı hesaplama, artık küme bilgisayarlarla hesaplama halini almıştır. Küme bilgisayar, birlikte çalışmak üzere bağlanmış birden fazla sunucudan oluşur. En önemli dezavantajı kullanıcıya tek sistem arayüzü sunamamasıdır.



# Cluster Computing

- Küme hesaplamanın tanımından da belirtildiği gibi kümeli sistem kullanıcıya tek bir sistem olarak görünmektedir; fakat burada dikkat edilmesi gereken sistemdeki tüm bilgisayarların kullanıcıların kullanımına açık olduğudur. Ayrıca küme hesaplama da tüm bilgisayarlar dış dünyaya açıktır. Bu tür sistemler için bir yerel ağa bağlı NT iş istasyonları kümesi örnek olarak verilebilir.



# Cluster Computing Avantajları

- Birden çok bilgisayarın kaynakları kullanıldığından bir bilgisayar sisteminin kullanılmasından elde edilebilecek çok daha yüksek seviyede başarımlar ve işlem gücü elde edilmesi küme hesaplamanın sistemlerde kullanılmasını cazip kılan en önemli özelliktir.
- Küme hesaplama çökmelere karşı etkin bir koruma sunmaktadır. Bir bilgisayarın çökmesi durumunda o bilgisayara verilmiş olan iş hemen başka bir bilgisayara yönlendirilerek yapılmakta olan işin aksaması önlenmektedir.
- Kümeli sisteme bir bilgisayar dahil edilmek istendiği zaman bu bilgisayarın işletim sistemi sistemdeki diğer bilgisayarlardan kopyalanarak kurulabilmekte ve ufak ayar değişiklikleriyle bu bilgisayar sisteme kolayca dahil edilmektedir.

# Cluster Computing Dezavantajları

- ❑ Küme hesaplamaya tabi tutulacak sistemler doğaları gereği paralel işleme uygun olmalı ve bu tür uygulamaların sayısı oldukça sınırlıdır.
- ❑ Küme hesaplama bilgisayar ağlarının gelişmesi sonucu ortaya çıkmış bir hesaplama türüdür. Ancak küme hesaplamadaki en büyük sorunlar ağ iletişiminden kaynaklanmaktadır.

# Massive Parallel Processing (Yüksek Düzeyde Paralel İşleme)

- Yüksek düzeyde paralel işlem (MPP), aynı programın iki veya daha fazla işlemci tarafından ortaklaşa işlenmesi biçimidir. Her işlemci programın farklı iş parçacıklarını işler ve her işlemcinin kendisi kendi işletim sistemine ve özel belleğe sahiptir. MPP'de yer alan farklı işlemcilerin iş parçacıklarını işlemeyi düzenlemelerine izin vermek için bir mesajlaşma arabirimi gereklidir. Bazen, bir uygulama, uygulama üzerinde işbirliği içinde çalışan binlerce işlemci tarafından ele alınabilir.

# Massive Parallel Processing (Yüksek Düzeyde Paralel İşleme)

- MPP, ilgili tüm işlemciler arasında belirli bir veritabanı fonksiyonlarının paylaşılmasını gerektiren karmaşık bir süreçtir. MPP sırasında veri yollarının bir araya getirilmesi yoluyla işlemciler arasında mesajlar değiş tokuş edilir. MPP genellikle karar destek sistemleri ve veri ambarları gibi uygulamalarda bulunur. Süper bilgisayarlar da MPP mimarisinin örnekleridir.

# Grid Computing

- Distributed computing olarak adlandırılan dağıtık bilgi işleme yönteminin sanallaştırılmasını sağlayan çözüm mimarisine kısaca Grid Computing denilmektedir.
- Buradaki temel amaç dağıtık bilgi işleme ve veri kaynaklarının kullanmakta olduğu işlemci güçleri, ağ kapasiteleri ve depolama kapasiteleri ile tek büyük bir sistem yaratmaktır. Yaratılan bu sistem tamamen birbirinden bağımsız çalışmakta olan ve birbirine benzemeyen sistemlerin bir araya gelerek oluşturduğu sanal bir işleme gücüdür. Bilgi sistemleri kapasitelerine sonsuz bir kullanım imkanı sunmaktadır.

# Grid Computing

- Grid Computing, ana yapısı açık standartlardan ve protokollerden oluşmaktadır. Open Grid Services Architecture (OGSA) olarak adlandırılan bu açık mimari heterojen ve coğrafi dağınık çevrelerin birbirleri arasında haberleşmelerine imkan tanır.
- Grid Computing ile organizasyon içindeki bilgi işleme gücü ve veri kaynakları optimize olarak kullanılır, bunun için büyük kapasiteler iş yüklerine ayrılarak (paylaştırılması amacıyla) bütün kaynaklar tarafından işlenir, böylece en optimum kullanım sağlanmış olur.

# Grid Computing

- Grid computing dağıtık ve sanal bir yapıdır. Merkezi yoktur.
- Grid Computing açık standart ve protokoller ile kurulur.
- Grid Computing genel iş hedeflerine göre belirlenmiş bir servis kalitesi çerçevesinde çalışmalıdır.
- Grid Computing, bilgi teknolojileri dünyasında bir devrim değildir, aslında halen kullanılmakta olan dağıtık bilgi işleme, peer-to-peer, Web ve sanallaştırma teknolojileri gibi bir çok kavramın evrim geçirmesidir.

# Dağıtık Sistem Nedir?

- “*Ağ üzerindeki bilgisayarlarda bulunan donanım veya yazılım bileşenlerinin yalnız mesaj göndererek haberleştikleri sistem.*” [Coulouris]
- “*Dağıtık bir sistem, kullanıcılara tek bir sistem olarak görünen, bağımsız bilgisayarlar bütünüdür.*” [Tanenbaum]
- Örnek: WWW, Intranet (organization), P2P sistemler (Napster gibi)
  - Bankalar (Bankamatikler)
  - Bilet rezervasyonu
- Dağıtık işletim sisteminde birden fazla CPU (işlemci) olmasına rağmen kullanıcı bunu tek işlemci olarak görür.



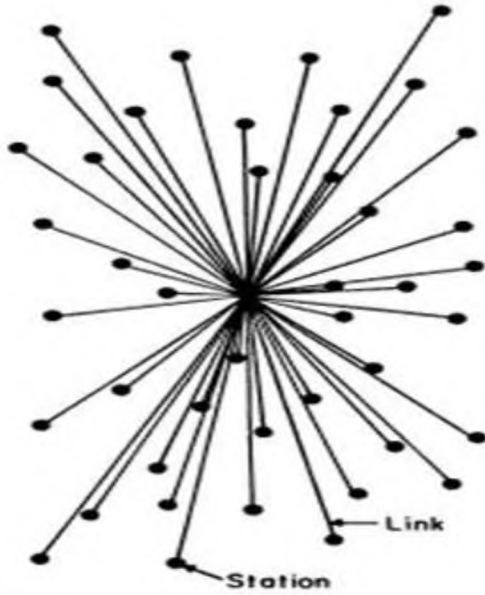
# Leslie Lamport'un Tanımı

- *“Dağıtık bir sistem, hiç haberim olmayan bir makine çöktüğü için, üzerinde hiçbir iş yapamadığım bir sistemdir.”*
  - Leslie Lamport; dağıtık sistemlerde
    - Timing (zamanlama)
    - Message ordering (mesaj sıralama) ve
    - Clock synchronizationüzerinde çalışan tanınmış bir araştırmacıdır.
  - Kullanıcı çalıştırdığı programları nerede çalıştırdığını ve dosyaların nerede yerleşmiş olduğunu bilmez. Bu tür işlemlerin hepsi sistem tarafından otomatik yapılır.

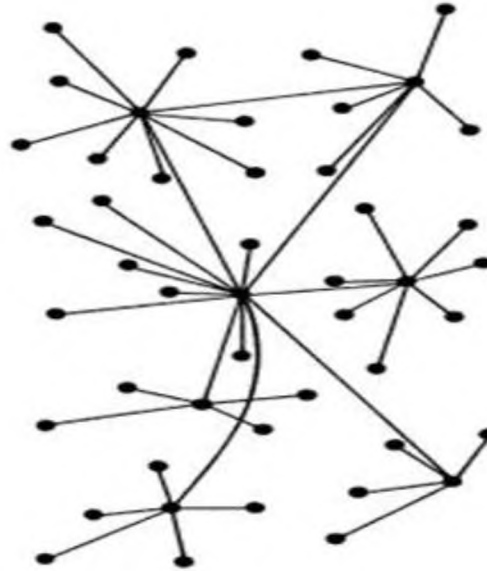
# Dağıtık Sistem Nedir?

- Dağıtık sistem, birden fazla bilgisayar ve/veya sunucunun birbirleri arasında iletişim kurması ve birbirleri ile replika şekilde bir ağ bütünü olarak çalışmasına denir.
- Dağıtık sistem, bir ağ üzerine kuruludur. Yazılımla, ağdaki bileşenler arası uyumluluk ve kullanıcı açısından saydamlık sağlanır. Dolayısıyla, dağıtık sistemin diğer ağlardan farkı donanım değil, yazılım katmanından işletim sisteminden kaynaklanmaktadır.
- Bu büyük ağda bulunan donanımlar kullanıcıya tek bir bilgisayar gibi davranır ve en iyi performansı sağlamayı amaçlar.
- Dağıtık sistemlerin önemli bir amacı ise, kaynakları paylaşmaya duyulan gerekliliktir. Bu kaynaklar donanımsal bileşenler (hdd, yazıcı) olabileceği gibi, dosyalar, veri tabanı, gibi servislerdir.

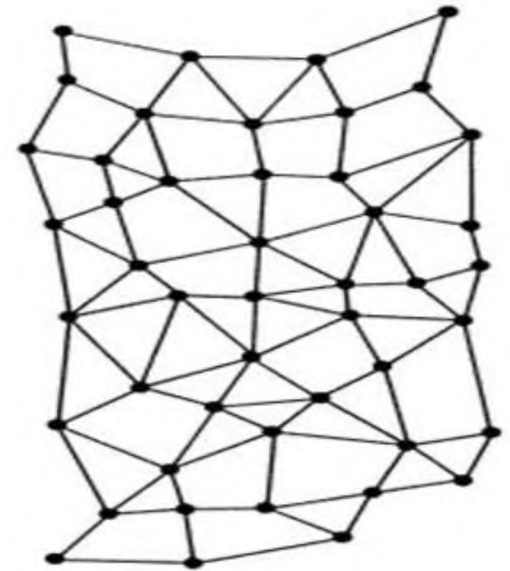
# Merkezi ve Dağıtık Sistem Yapısı



MERKEZİ



ÇOK MERKEZLİ



DAĞITIK

# Dağıtık Sistemin Karakteristik Özellikleri

- Dağıtık sistemler ortaya çıktıkça belli özellikleri sağlamak ile yükümlü olurlar. Bu karakteristik özellikleri sağlıyor olmak, dağıtık sistemlerin en büyük ve en zor problemidir.
- **Yüksek Erişebilirlik:** Dağıtık sistemlerin, erişebilirliğinin olabildiğince yüksek olması beklenir. Başarısız bir işlem olsa bile, işlemi tetikleyen kaynağın mutlaka sisteme bir şekilde erişimi olması beklenir.
- **Güvenlik:** Dağıtık sistemlerde, işlemlere ve veriye erişim yetkilendirilir.
- **Performans:** Dağıtık sistemlerin, kendi performanslarının farkında olması beklenir. Belli sistem yoğunluklarında alacağı aksiyonların belirli olması sistem tasarımında önemlidir.

# Dağıtık Sistemin Karakteristik Özellikleri

- **Açıklık:** Dağıtık sistemler genişletilmeye açık olmalıdır. Sisteme kolaylıkla alt sistem bileşenleri eklenebilmelidir.
- **Ölçeklenebilirlik:** Dağıtık sistemler kaynak kullanımı konusunda kolayca ölçeklenebilmelidir. Kaynak kullanımına göre sistemin verimli çalışmaya devam etmesi bir şekilde sağlanmalıdır.
- **Tutarlılık:** Sistemin tutarlı olması beklenir. Aynı anda, aynı veri kaynağına erişen sistemlerin uyumlu ve tutarlı çalışması gerekmektedir.

# Ağlar ve Dağıtık Sistemlerin Karşılaştırılması

- **Ağ:** Yerel veya geniş alandaki bilgisayarları bağlamak ve protokoller arası mesaj alışverişi yapmak için kullanılan ortamdır. Ağ birimleri görülebilirdir ve açıkça IP adresleriyle adreslenmiştir.
- **Dağıtık Sistem:** Birçok bağımsız bilgisayarın varlığı belirgin değildir. Ancak,
  - Açıklık, güvenilirlik, gibi birçok problem ortaktır, ancak farklı katmanlardadır.
  - Ağlar, paket yönlendirme vs üzerine odaklanırken, dağıtık sistemler uygulamalara odaklanır.
  - Her dağıtık sistem bir bilgisayar ağı tarafından sağlanan servislere dayanır.

# Dağıtık Sistemlerin Kullanım Amaçları

- Kullanıcıya yönelik (neden dağıtık sistem?)
  - Kullanıcı ve kaynakların kolayca bağlanması
  - Şeffaflığın sağlanması
- Teknik hedefler (nasıl başarılır?)
  - Açık olarak
  - Ölçeklenebilir olarak
- Bu hedeflere bakmak şu soruyu cevaplamaya yardımcı olur:
  - “Dağıtık bir sistem kurmak, uğraşmaya değer mi?”

# Dağıtık Sistemlerin Gerekliliği

- İşlevsel ayırım:
  - Kapasite ve amaç bakımından farklı bilgisayarların varlığı:
    - İstemciler ve Sunucular
    - Veri toplama ve veri işleme
- Doğal dağıtıklık:
  - Bilgi:
    - Farklı bilgiler farklı kişiler tarafından oluşturulup kontrol edilir (Web sayfaları gibi)
  - İnsan:
    - Bilgisayar destekli işbirliği (sanal ekipler, mühendislik, sanal ameliyat)
  - Süpermarket zincirleri için perakende ve stok sistemleri



# Dağıtık Sistemlerin Gerekliliği

- Güç dengesizliği ve yük çeşitliliği:
  - İşlem yükünün farklı bilgisayarlar arasında dağıtılması.
- Güvenilirlik:
  - Farklı yerlerde uzun süreli koruma ve veri yedekleme (çoğaltma).
- Ekonomi:
  - Birçok kullanıcı arasında yazıcı paylaşımı ile sahip olma masrafının azaltılması.
  - Bir bilgisayar ağından süper-bilgisayar elde etme.

# DS Tanımının Getirdikleri

- Dağıtık sistemlerdeki bilgisayarlar ayrı kıtalar üzerinde, aynı bina veya aynı oda içerisinde bulunabilir. DS'in getirdikleri:
  - Birlikte ve birbirinden bağımsız çalışan sistemler
    - İşlerini birbirinden bağımsız yaparlar
    - *Aynı zamanda program çalıştırır, bütün bir işleme kaynağı gibi görünür, birlikte çalışırlar*
    - İşlemler mesaj alışverişiyle anlaşılırlar.
  - Heterojen (çeşitlilik, farklılık): networks, hw, os, PLa
  - Ortak saat yok: Bilgisayarlar saatlerini kısıtlı derecede senkronize edebilir
  - Bağımsız bozulma: biri bozulsa da diğerleri çalışmaya devam eder

# Dağıtık Sistemlerin Altında Yatan Temel Prensipler

- Dağıtık sistemlerin altında yatan temel prensipler :
  - Processes (İşlemler)
  - Communication (İletişim)
  - Naming (İsimlendirme)
  - Synchronization (Senkronizasyon)
  - Consistency (Tutarlılık)
  - Fault Tolerance (Hata Payı)
  - Security (Güvenlik)
- Dağıtık sistemlerdeki bazı temel kavramlar:
  - Client - Server (İstemci-Sunucu)
  - Object - Based Systems (Nesne-Tabanlı Sistemler)
  - File Systems (Dosya Sistemleri)

# Dağıtık Sistemlerin Nitelikleri

- Paralele işlemler
  - Birlikte çalışan bağımsız bileşenler
- Mesaj gönderme ile iletişim
  - Ortak bellek yok (No shared memory)
- Kaynak paylaşımı
  - Yazıcı, veri tabanı, diğer servisler
- Genel sistem durumu belirli değil (No global state)
  - Hiçbir işlem, sistemin genel durumuna dair bir bilgi edinemez
- Ortak saat yok
  - İşlemler için sadece kısıtlı derecede saat eşitleme mümkün olabilir

# Dağıtık Sistemlerin Amaçları

1. **Kullanıcılarla kaynakları birleştirmek:** Erişim servisleri uygulama çalıştırır.
  - Belirli kaynaklar:
    - Yazıcılar, bilgisayarlar, işleme gücü, veri
  - Neden paylaşım?
    - Ekonomi
    - İşbirliği, Bilgi değiş-tokuşu (grup çalışması)
  - Paylaşım problemleri
    - Güvenlik
    - İstenmeyen işbirlikleri
2. **Şeffaflık:** Kaynaklar ve işlemlerin ağ üzerinde dağıtık olduğu açık değildir.
3. **Açıklık:** Servislerin sentaks ve semantikleri ile ilgili standart kurallar
4. **Ölçeklenebilirlik:** Daha fazla kullanıcı/kaynak, coğrafya, yönetim
5. **Uzun süre bozulmadan çalışabilirlik.**

# Paralel Sistemlerden Farkı

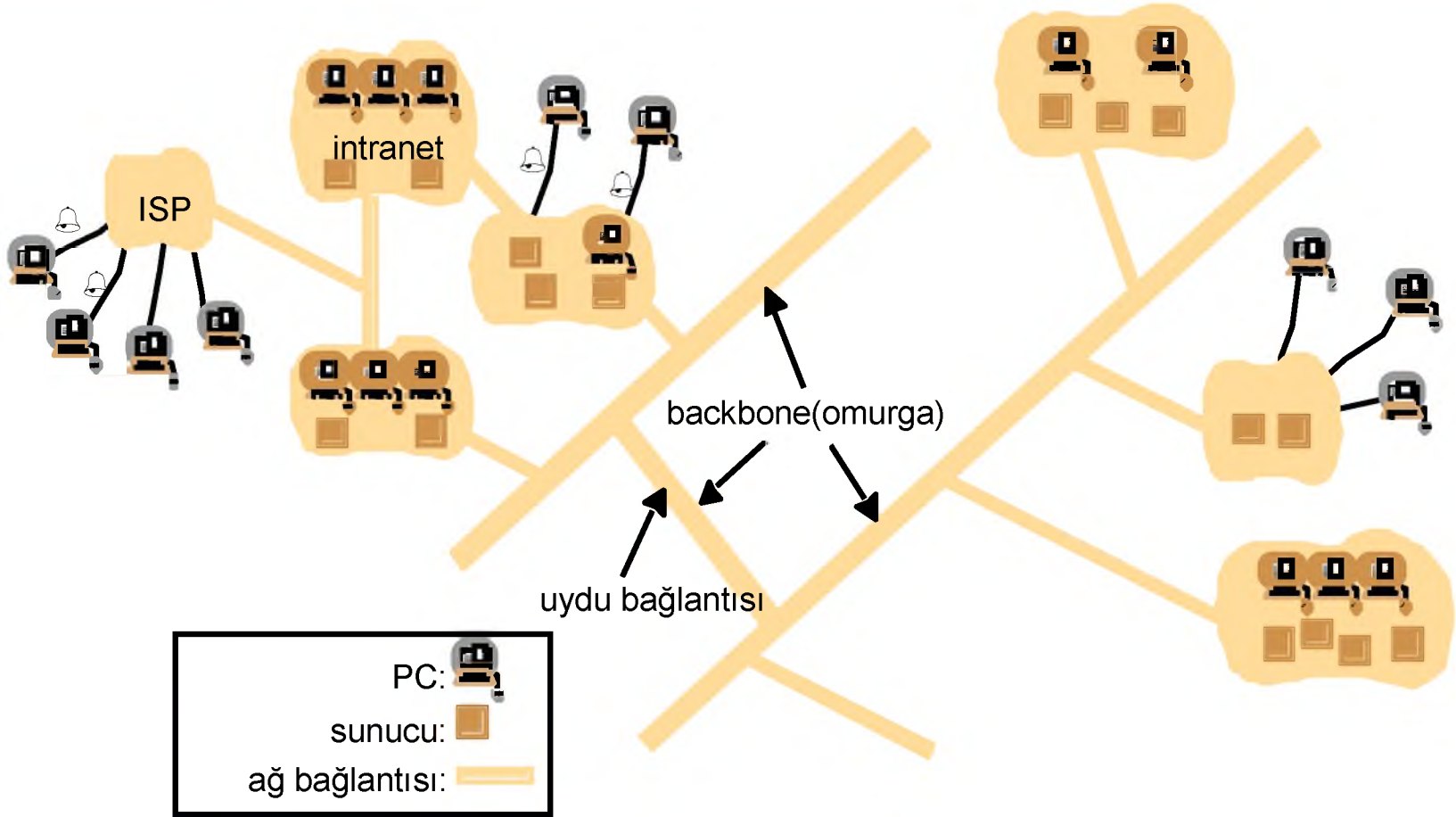
- Multiprocessor (çok işlemcili) sistemler: DS değildir.
  - Shared memory (ortak bellek)
  - Bus-based interconnection network (kablolu bağlantı)
  - **Örnek:** İki veya daha fazla CPU'lu SMP'ler (symmetric multiprocessor)
- Çoklu bilgisayar sistemleri: DS değildir.
  - No shared memory (ortak bellek yok)
  - Homojen donanım ve yazılım
    - Massively Parallel Processors (MPP)
      - Tightly coupled high-speed network
    - PC/Workstation clusters
      - High-speed networks/switches based connection.

# Örnek Dağıtık Sistemler

- **Internet:** birbirine bağlı çeşitli bilgisayar ağları.
  - Uygulamalar iletişimi mesajlarla sağlar.
  - Dağıtık sistem kullanıcıları www, email, dosya gönderme gibi servislerden faydalanır.
- **Intranet:** bir organizasyon tarafından yönetilen alt ağ. Yerel güvenlik politikasıyla sınırlıdır.
  - Uydu bağlantısı, fiber-optik kablolar gibi hızlı iletim kapasitesi olan ağ altyapısı ile bağlıdırlar.
- **Kablosuz ağlar**

# Genel internet uygulamaları ve servisleri:

Müzik, radyo, TV kanalları, video konferans erişimi için multimedia servisleri, ve çoklu kullanıcı desteği.

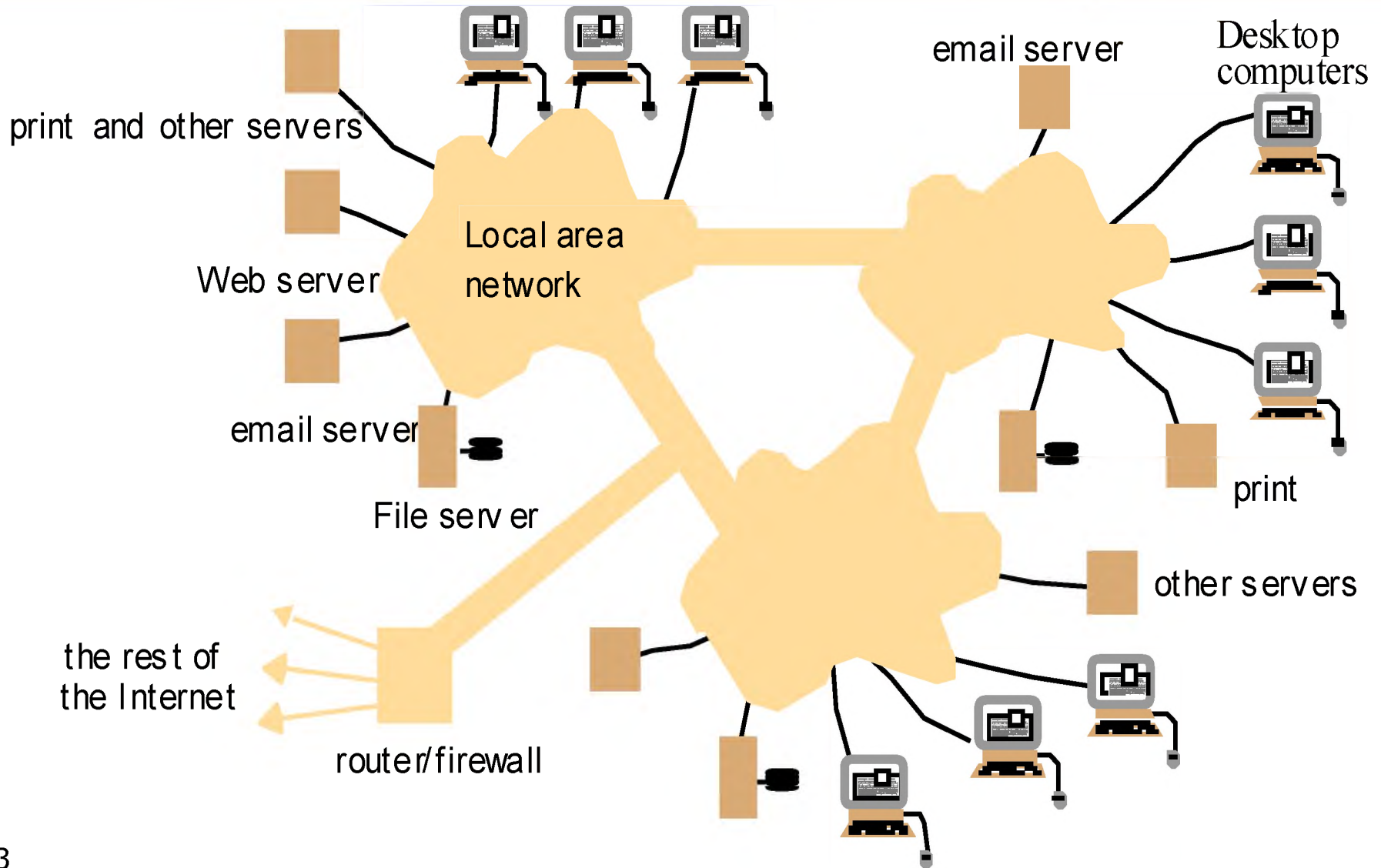


- Internet, çeşitli servisler barındıran birçok farklı bilgisayar türlerinden oluşan büyük bir ağ topluluğudur.

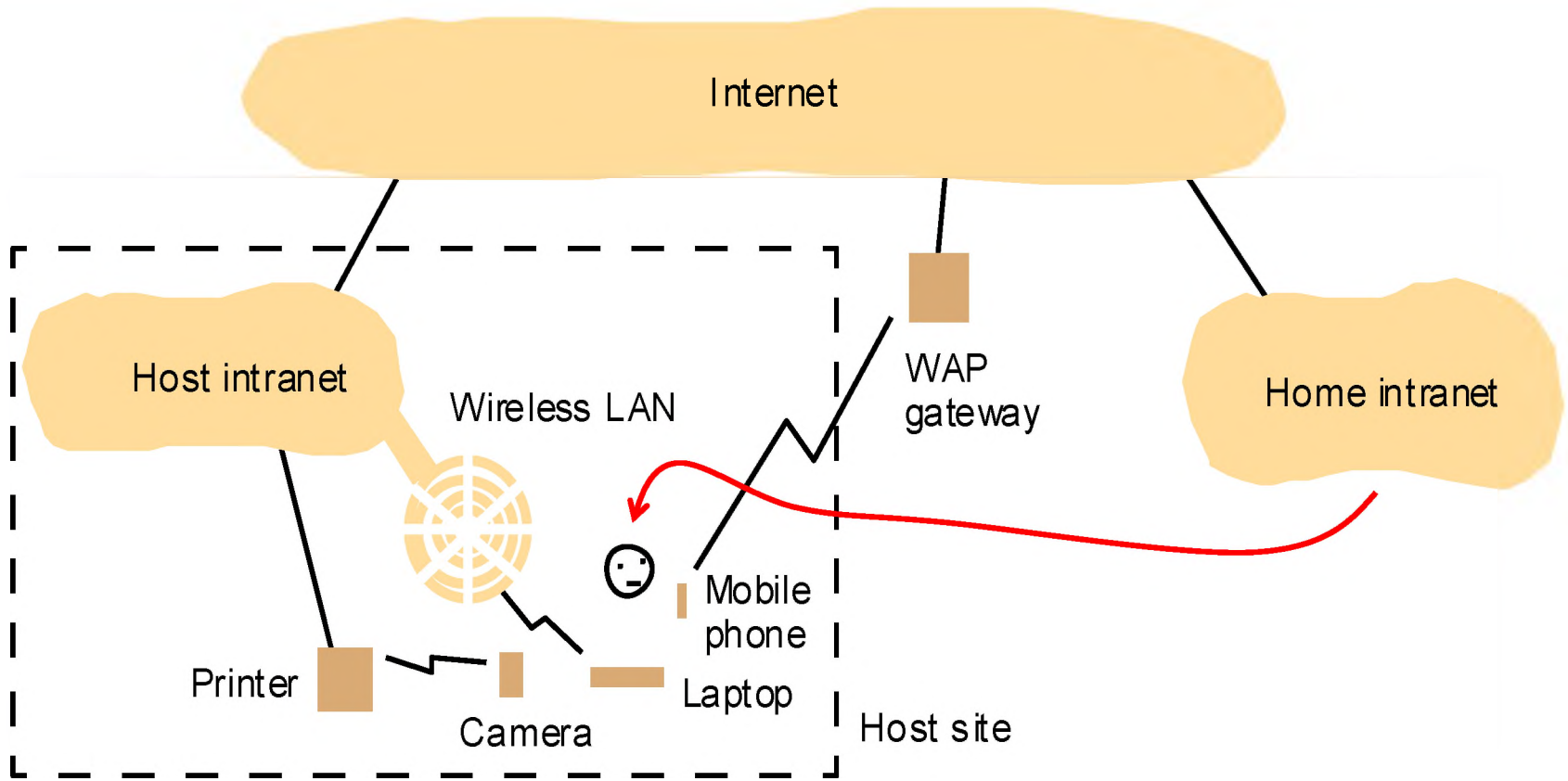


# Genel Bir İtranet:

İnternetin bir alt parçası olup, ayrıca yönetilir ve kaynakların ağ içerisinde paylaşımını destekler (dosya/depolama sistemleri ve yazıcılar)

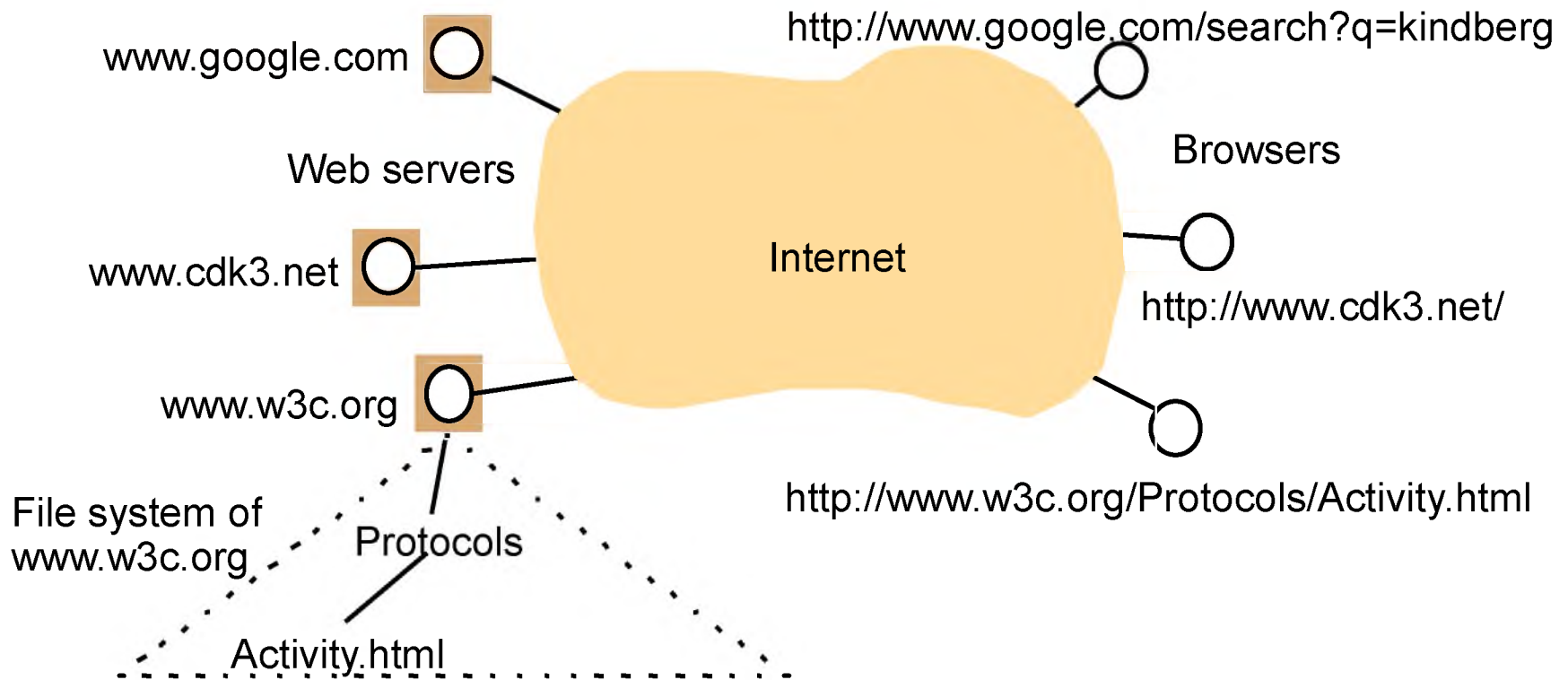


# Gezgin ve Dağıtık Bilgi İşleme: Dağıtık Bir Sistemdeki Taşınabilir ve El Aletleri



- Uygun konumlara yerleştirilmiş (yazıcı gibi) kaynakları değerlendirmek için kablosuz veya kablolu ağ aracılığı ile Home intranet kaynaklarına sürekli erişim desteği

# Kaynak Paylaşımı ve Web: Açık Protokoller, Ölçeklenebilir Sunucular, Takılabilir Tarayıcılar



# World Wide Web kaynakları

- World Wide Web **kaynakları** ve diğer servisler URL'ler ile isimlendirilir.
- Web Sayfaları paylaşılan kaynaklara örnektir
- Bu kaynaklar **Web sunucuları** tarafından yönetilir
- **Web, istemci-sunucu** yapısı üzerine kuruludur.
  - **Web tarayıcısı:** İstemci programı,
  - **Web sunucusu:** İstenen web sayfalarını içeren yerel dosyalara (kaynaklara) erişir ve istemci tarayıcısına iletir.
- **URL - Uniform Resource Locator**

# URL

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

# İş Örnekleri ve Zorluklar

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

# İş Örnekleri ve Zorluklar

## ■ Ya . . . ?

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

## ■ Veya . . . ?

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

# İş Örnekleri ve Zorluklar

- Ya . . . ?
  - İki müşteri aynı ürünü aynı anda sipariş verirse?
  - **Concurrency (Aynı anda kullanım)**
- Veya . . . ?
  - Stok bilginizi tutan veri tabanı çökerse?
  - Sipariş esnasında müşterinizin bilgisayarını çökerse?
  - **Fault tolerance (Hata payı)**



# İş Örnekleri ve Zorluklar

## ■ Ya . . . ?

- Birileri veri çalmak için sisteminize girmeye çalışırsa?
- ... bilgi çekerse?
- ... müşteriniz sipariş verir de, sonradan vermedim diyerek ürün teslimini reddederse?
- **Security (Güvenlik)**

## ■ Veya . . . ?

- Öyle başarılı olursunuz da, milyonlarca insan aynı anda online mağazanızı aynı anda ziyaret ederse?
- **Scalability (Ölçeklenebilirlik)**

# İş Örnekleri ve Zorluklar

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

# Kısaca Zorluklar

- Heterogeneity (Çeşitlilik)

- Çeşitli bileşenler birbiriyle uyumlu şekilde çalışabilmelidir.

- Distribution Transparency (Dağınıklık Şeffaflığı)

- Dağınıklığın varlığı mümkün oldukça kullanıcıdan saklanmalıdır.

- Fault Tolerance (Hata Payı)

- Bir bileşenin bozulması (kısmi bozukluk) tüm sistemin bozulmasına sebep olmamalıdır.

- Scalability (Ölçeklenebilirlik)

- Sistem, artan kullanıcı sayısına rağmen verimli çalışmaya devam edebilmelidir.
- Sisteme yeni kaynaklar eklenerek performans artışı sağlanabilmelidir.

# Heterogeneity (Çeşitlilik)

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

# Dağıtık Bir Sistemde Şeffaflık

- **Şeffaf DS:** Kullanıcılara, tek bir bilgisayar sistemi gibi görünür, bağımsız bilgisayarlar topluluğu olarak değil.
- Şeffaflık çeşitleri
  - **Erişim:** Veri gösterimindeki farklar, ve kaynaklara nasıl erişildiği gizlenir. Yerel ve uzak kaynaklara erişim aynı işlemlerle sağlanır. Örn., Network File Systems (Ağ Dosya Sistemleri)
  - **Konum:** Bir kaynağın bulunduğu yer gizlenir. Kaynaklara, fiziksel konumları bilinmeden erişilir. Alan adının (domain name) makine adresinden ayrımı gibi.
  - **Migration (Taşınma):** Kaynağın yer değiştirme durumu gizlenir.

# Dağıtık Bir Sistemde Şeffaflık

- **Relocation (Yeniden konumlandırma):** Kaynağın, kullanım esnasında, yer değiştirme durumu gizlenir.
- Migration / relocation şeffaflığı, bir sistemdeki kullanıcıların veya uygulamaların işlemlerini etkilemeden taşınabilmelerine olanak sağlar.
  - Çalışma esnasında (runtime) bir isim sunucusundan (name server) bir başkasına geçiş yapma
  - Bir vekil veya işlemin (agent/process) bir düğümden (node) diğerine taşınması gibi

# Şeffaflık Düzeyi

- Şeffaflık
  - **Her zaman istenmez:** Farklı kıtalarda bulunan kullanıcılar (context-aware), time-zone (zaman-dilimi), hız
  - **Her zaman mümkün olmaz:** Hataların gizlenmesi (bir bilgisayar yavaş mı, bozuk mu)
- Yüksek şeffaflık ile performans arasındaki denge

# Şeffaflık Çeşitleri

- **Replication (Kopyalama):** Bir kaynağın birden fazla yerde yedeğinin (kopyasının) tutulduğu gizlenir. Kopyalanmış kaynaklara, sadece bir kopya varmış gibi erişilir. Güvenilirlik ve performans kopyalarla artırılır, ama kullanıcıların veya uygulama geliştiricilerinin kopyalardan haberi olmaz.
- **Concurrency (Birlikte çalışma):** Bir kaynağın birden fazla kullanıcı tarafından paylaşıldığı gizlenir. Bir işlem, diğer bazı işlemlerin de aynı kaynaklara erişmekte olduğunu farkında olmamalıdır
- **Failure (Bozulma):** Bir kaynağın bozulma veya düzelme durumu gizlenir. Bozulmalar olsa da, görevler tamamlanabilir. Mesaj iletim tekrarı, bir ağ sunucusu düğümünün bozukluğu, web sitesini çökertmemeli.



# Şeffaflık Çeşitleri

- **Persistence (Süreklilik):** Bir yazılım kaynağının bellekte veya disk üzerinde olma durumu gizlenir.
- **Performans:** Yük değişimine göre performansı artırmak için sistemin tekrar yapılandırılması sağlanır. Örn., bileşenlerin dinamik olarak eklenip kaldırılması. Kullanıcı sayısı artınca, doğrusal yapılardan basamaklı (hierarchical) yapılara geçilmesi.
- **Scaling (Ölçekleme):** Sistemin ve uygulamaların, sistem yapısında veya uygulama algoritmalarında değişikliğe gerek olmadan genişletilmesine olanak sağlar.
- **Ağ şeffaflığı:** erişim + konum şeffaflıkları

# Bozulma Şeffaflığı: Hata Payı

- **Bozulma:** Sunulan bir servisin artık tarifine uygun çalışmaması
- **Hata:** Bozulma sebebi (bileşen bozulması gibi)
- **Hata Payı:** Hatalara rağmen bozulmama
  - Örn: e-mail mesajlarının sonradan iletilmesi

# Hata Payı Mekanizmaları

- Hata Algılama
  - Checksums (sağlama), heartbeat (kalp atışı), ...
- Hata Maskeleye
  - Bozuk mesajların tekrar gönderimi, fazlalık, ...
- Hataya Müsamaha
  - Exception handling (istisna işleme), timeouts (zaman aşımaları),...
- Hatadan Kurtulma
  - Rollback (geri sarma) mekanizmaları,...

# Ölçeklenebilirlik

- Sistem, küçük bir Intranet'ten Internet'e kadar uzanan birçok farklı ölçekte verimli olarak çalışabilmelidir.
- Kaynak ve kullanıcı sayısında belirgin bir artış olsa da etkili olarak çalışmaya devam edebilmelidir.
- Zamanla her şey çoğalır, kullanıcılar artar, bilgisayar sayısı büyür, veri miktarı fazlalaşır, ...
- Sistemler nasıl ölçeklenebilir olur? Sadece donanım eklemekle mümkün değildir. Bir makinenin 20 kullanıcıya hizmet vermesi, iki makinenin 40 kullanıcıya hizmet vereceği anlamına gelmez.
- Ölçeklenebilirlik, bir sistemin artan iş yüküne rağmen işlevselliğini düzgün bir şekilde devam ettirebilmesidir.

# Concurrency (Birlikte Çalışma)

- Resource (Kaynak):
  - Yalnız bir kullanıcı → kısıtlı verim
  - Çoklu istemci talepleri: Birlikte erişim
- Paylaşılan kaynaklara çoklu erişimin desteklenip yönetilmesi:
- Birlikte çalışılan ortamdaki bir nesnenin güvende olması için;
  - İşlemleri, verileri tutarlı kalacak şekilde, senkronize olmalı (eşitlenmeli) (örn. Banka hesabı).
  - Bu, dağıtık olmayan sistemlerde semafor kullanımından daha zordur.

# Openness and Interoperability (Açıklık ve Birlikte işlerlik)

- **Açık sistem:** "... düzenli geliştirilmiş uygulama yazılımlarının, küçük değişikliklerle, çok çeşitli sistemlere taşınabilmesini ve yerel ve uzak sistemlerdeki diğer uygulamalarla birlikte işleyebilmesini sağlamak ve kullanıcılarla, taşınabilirliği kolaylaştıran bir tarz ile etkileşmek için, arayüzler, servisler ve desteklenen formatlara yeterli **açık tarif** veren bir sistemdir." (POSIX Open Systems Environment rehberi).
- Açık tarif/standart geliştiriciler - topluluklar:
  - ANSI, IETF, W3C, ISO, IEEE, OMG, Ticari ortaklıklar,...
- Yeni servisler, var olan bileşenler bozulmadan eklenebilir.

# Güvenlik

- Kaynaklar yetkili kullanıcılarca erişilebilir ve istekleri doğrultusunda kullanılabilir.
- Confidentiality (Gizlilik)
  - Yetkisiz bireylerce görülmeye karşı koruma.
  - Örn. Bilgiye yetkili erişim sağlamak için kullanılan ACL'ler (Erişim kontrol listeleri).
- Integrity (Bütünlük)
  - Değişime veya bozulmaya karşı koruma.
  - Örn. Bir para isteğinde hesap numarası veya miktar değerinin değiştirilmesi

# Güvenlik

- Availability (Ulaşılabilirlik)
  - Kaynak erişim yollarındaki çakışmalara karşı koruma.
  - Örn: Servis saldırılarının reddi
- Reddedilememe
  - Bir bilginin gönderilme / alınma delili
  - Örn: Dijital imza



# Güvenlik mekanizmaları

- Encryption (Şifreleme)
  - Örn: Blowfish, RSA
- Authentication (Doğrulama)
  - Örn: Password (şifre), açık anahtarla yetkilendirme
- Authorization (Yetkilendirme)
  - Örn: Erişim kontrol listeleri

# Özet

- Dağıtık sistemler her yerde bulunur.
- İnternet, dünyanın her bir yanındaki kullanıcıların her bir yandaki servislere erişimlerini sağlar.
- Kaynak paylaşımı dağıtık sistem kurmaya teşvik eden etmenlerin başta gelenidir.
- DS kurulumu birçok zorluğu beraberinde getirir:
  - Çeşitlilik, Açıklık, Güvenlik, Ölçeklenebilirlik, Hata denetimi, Birlikte çalışma, Şeffaflık.
- Dağıtık sistemler küreselleşmeyi sağlar:
  - Topluluk (Sanal takımlar, kuruluşlar, sosyal ağlar)
  - Science (e-Science) (Bilim)
  - Business (e-Bussiness) (İş)

# Kaynaklar

