

# Pengantar Sistem Terdistribusi

---

**Budi Susanto**  
[budsus@ukdw.ac.id](mailto:budsus@ukdw.ac.id),  
[budsus@yahoo.com](mailto:budsus@yahoo.com)

# Definisi Sistem Terdistribusi

- "A system in which hardware or software components located at *networked computers* communicate and coordinate their actions only by *message passing*." [Coulouris]
- "A system that consists of a collection of two or more *independent* computers which coordinate their processing through the exchange of synchronous or asynchronous *message passing*."
- "A distributed system is a collection of *independent* computers that *appear to the users* of the system as a single computer." [Tanenbaum]
- "A distributed system is a collection of *autonomous* computers linked by a network with software designed to produce an *integrated computing facility*."

# Jaringan Komputer vs Sistem Terdistribusi

- **Jaringan komputer** : komputer otonom yang secara eksplisit terlihat (secara eksplisit teralamatasi)
- **Sistem terdistribusi**: keberadaan beberapa komputer otonom bersifat transparan
- Secara normal, setiap sistem terdistribusi mengandalkan layanan yang disediakan oleh jaringan komputer
- Beberapa layanan pada jaringan komputer (seperti, name service) juga merupakan sistem terdistribusi
- Sistem terdistribusi lebih banyak masalah yang dihadapi

# Alasan untuk sistem terdistribusi

- **Distribusi fungsi** : komputer memiliki kemampuan fungsi yang berbeda-beda
  - client/server
  - Host/terminal
  - Data gathering / data processing
- **Distribusi beban/keseimbangan** : pemberian tugas ke prosesor secukupnya sehingga unjuk kerja seluruh sistem teroptimasi.
- Sifat terdistribusi mencegah terjadinya *application domain*, e.g.
  - cash register dan sistem persediaan untuk supermarket,
  - Komputer pendukung *collaborative work*

# Alasan untuk sistem terdistribusi

- **Replikasi kekuatan pemrosesan** : *independent processors* bekerja untuk pekerjaan yang sama
  - Sistem terdistribusi terdiri dari kumpulan mikrokomputer yang memiliki kekuatan pemrosesan yang tidak dapat dicapai oleh superkomputer
    - 10000 CPU, masing-masing berjalan pada 50 MIPS, mencapai 500000 MIPS,
      - Maka satu perintah dijalankan dalam waktu 0.002 nsec
- **Pemisahan fisik** : sistem yang menggantungkan pada fakta bahwa komputer secara fisik terpisah (e.g., untuk mencapai kehandalan).
- **Ekonomis** : kumpulan mikroprosesor menawarkan harga/unjuk kerja yang lebih baik dari pada mainframe

# Mengapa Sistem Terdistribusi ?

---

- Butuh berbagi data dan *resource* di antara pemakai
- Mendukung komunikasi *person-to-person*
- **Fleksibilitas** : komputer yang berbeda dengan kemampuan yang berbeda dapat di share antar user

# Masalah dengan sistem terdistribusi

---

- Software - bagaimana merancang dan mengatur software dalam DS
- Ketergantungan pada infrastruktur jaringan (world wide wait....)
- Kemudahan akses ke data yang di share, memunculkan masalah keamanan

# Konsekuensi...

## (Karakteristik Sistem Terdistribusi)

- **Sistem terdistribusi adalah sistem *concurrent* (serentak)**
  - Setiap komponen hardware/software bersifat otonom (kita akan menyebut komponen otonom adalah "proses")
  - Komponen menjalankan tugas bersamaan
    - *Contoh* : A dan B adalah concurrent jika A dapat terjadi sebelum B, dan B dapat terjadi sebelum A
  - Sinkronisasi dan koordinasi dengan message passing
  - Sharing resources
  - Masalah umum dalam sistem concurrent
    - Deadlock
    - Lifeclock
    - Komunikasi yang tidak handal

# Konsekuensi...(lanjut)

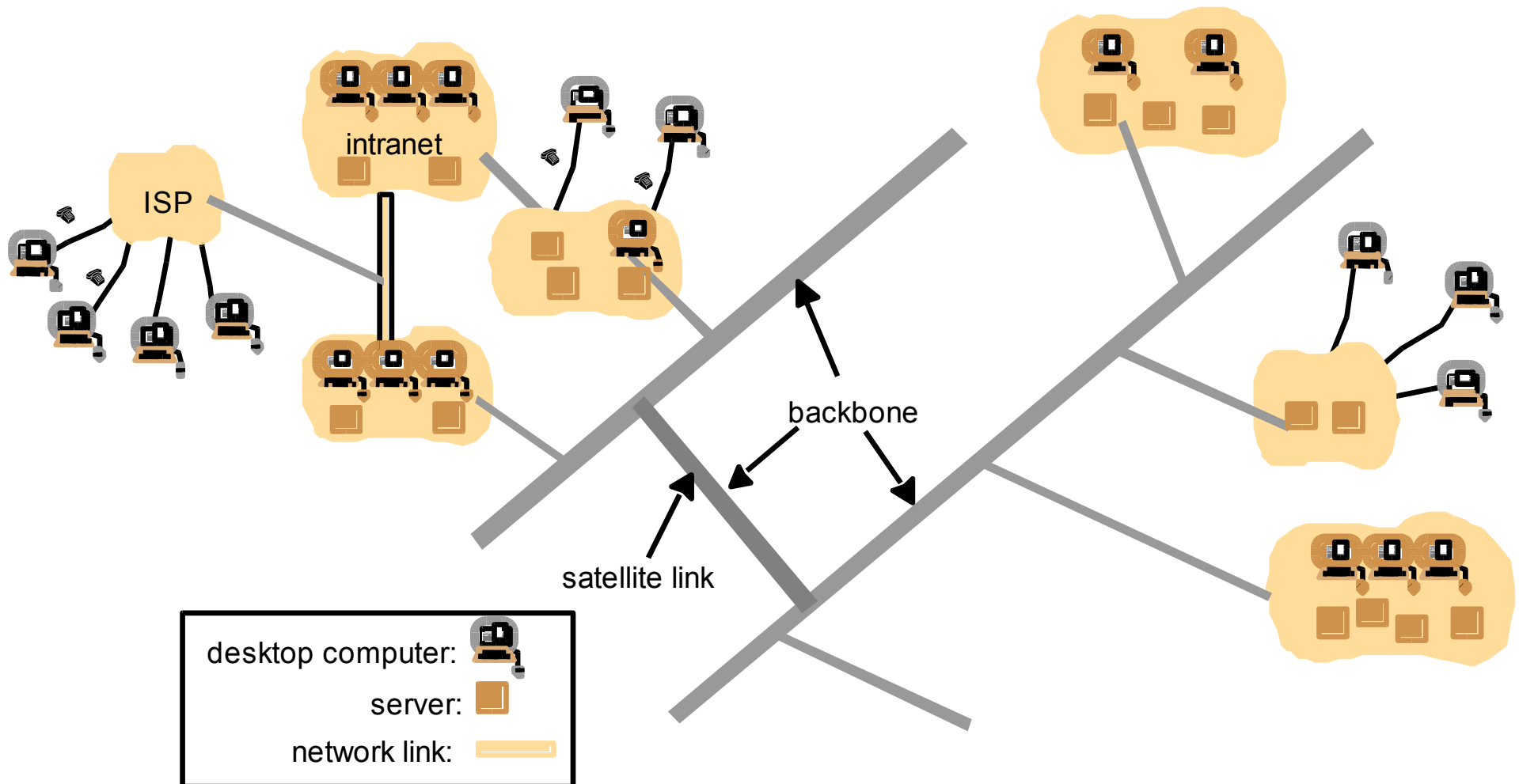
## (Karakteristik Sistem Terdistribusi)

---

- **Keterbatasan dalam *global clock***
  - Terdapat batasan pada ketepatan proses sinkronisasi clock pada sistem terdistribusi, oleh karena *asynchronous message passing*
  - Pada sistem terdistribusi, tidak ada satu proses tunggal yang mengetahui *global state* sistem saat ini (disebabkan oleh *concurrency* dan *message passing*)
- **Independent failure**
  - Kemungkinan adanya kegagalan proses tunggal yang tidak diketahui
  - Proses tunggal mungkin tidak peduli pada kegagalan sistem keseluruhan

# Contoh DS : Internet

- Jaringan komputer dan aplikasi yang heterogen
  - Mengimplementasikan protokol Internet

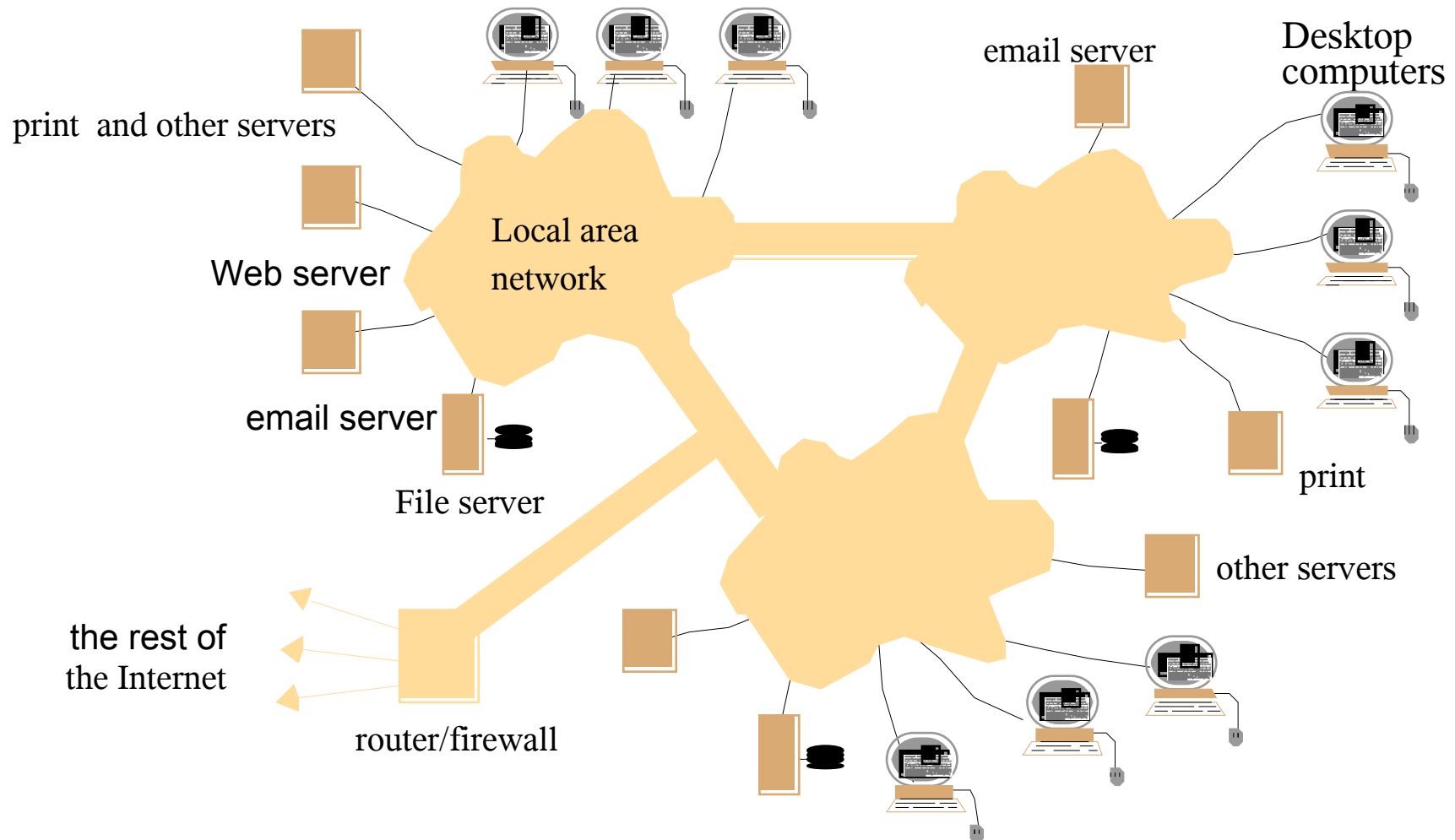


# Contoh DS : Sistem Multimedia Terdistribusi

- Biasanya digunakan pada infrastruktur internet
- Karakteristik
  - Sumber data yang heterogen dan memerlukan sinkronisasi secara real time
    - Video, audio, text
  - Multicast
- Contoh:
  - Teleteaching tools (mbone-based, etc.)
  - Video-conferencing
  - Video and audio on demand

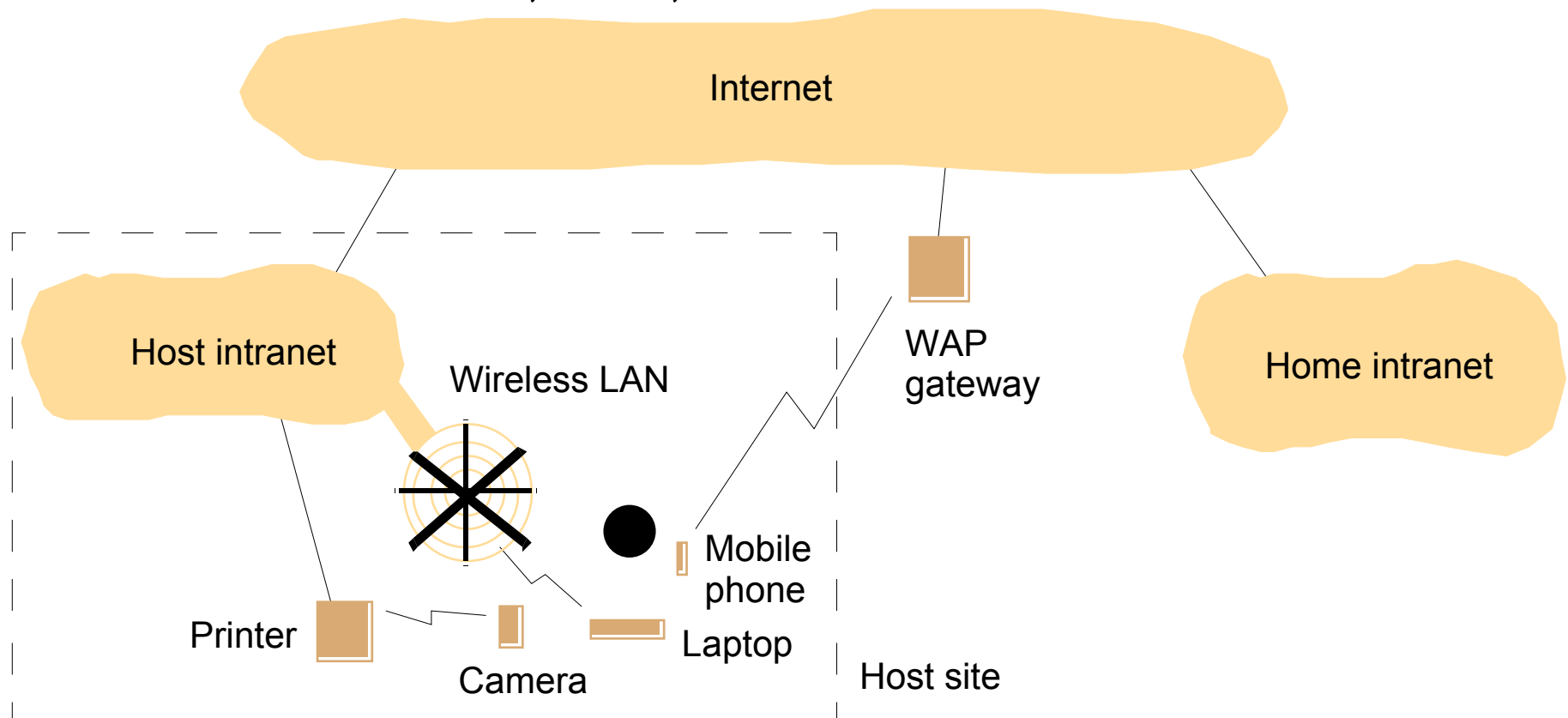
# Contoh DS: Intranet

- Jaringan yang teradministrasi secara lokal
- Biasanya proprietary
- Terhubung ke internet (melalui firewall)
- Menyediakan layanan internal dan eksternal



# Contoh DS : Mobile dan Sistem Komputasi Ubiquitous

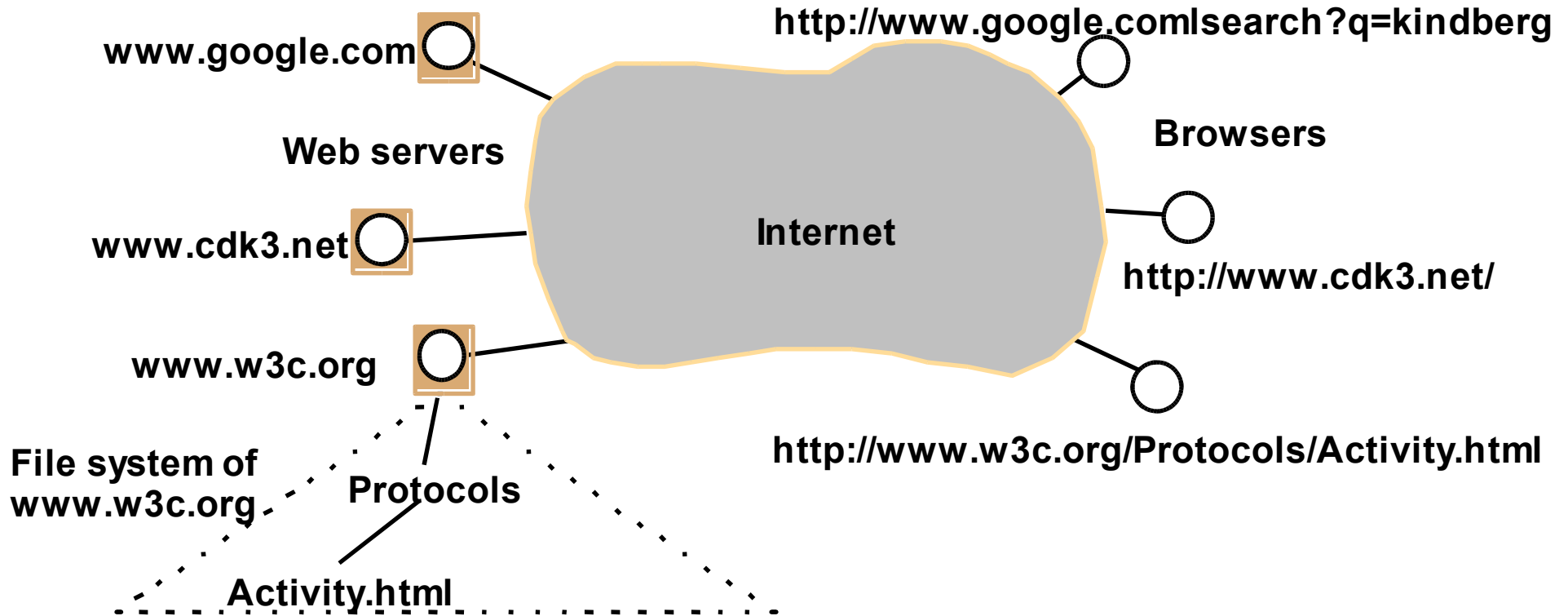
- Sistem telepon Cellular (e.g., GSM)
  - Resources dishare : frekuensi radio, waktu transmisi dalam satu frekuensi, bergerak
- Komputer laptop, ubiquitous computing
- Handheld devices, PDA, etc



# Contoh DS lainnya

- Sistem telepon
  - ISDN, PSTN
- Manajemen jaringan
  - Administrasi sumber jaringan
- Network File System (NFS)
  - Arsitektur untuk mengakses sistem file melalui jaringan
- WWW
  - Arsitektur client/server terbuka yang diterapkan di atas infrastruktur internet
  - Shared resources (melalui URL)

# World Wide Web



# Tantangan perancangan Sistem Terdistribusi

---

- ***Heterogen terhadap***
  - Infrastruktur jaringan
  - Hardware dan software (sistem operasi, perbedaan UNIX socket dan Winsock)
  - Bahasa pemrograman
- Beberapa pendekatan :
  - Middleware (contoh : CORBA)
  - Kode program Mobile (contoh : JAVA)

# Tantangan perancangan Sistem Terdistribusi (lanjut)

---

- **Openness**
  - Memastikan sistem dapat diperluas dan mudah dalam pemeliharaan
    - Mengikuti standard antarmuka
- **Security**
  - Privacy
  - Authentication
  - Availability

# Tantangan perancangan Sistem Terdistribusi (lanjut)

- **Scalability**

- Apakah sistem masih efektif dan handal dalam perkiraan pertumbuhannya?

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>	<i>Percentage</i>
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6
1999, July	56,218,000	6,598,697	12

# Tantangan perancangan Sistem Terdistribusi (lanjut)

- **Penanganan Kegagalan**
  - Pendeteksian
  - Masking
    - Transmisi ulang
    - Redudansi penyimpanan data
  - Toleransi
    - Exception handling
  - Redudancy
    - Redudan rute jaringan
    - Replikasi data pada beberapa mesin
- **Concurrency**
  - Penjadwalan yang konsisten terhadap concurrent thread
  - Menghindari masalah deadlock dan lifelock

# Tantangan perancangan Sistem Terdistribusi (lanjut)

- **Transparency**

- Menyembunyikan keanekaragaman (*heterogeneous*) dan tersebarnya sistem sehingga tampak sebagai satu sistem bagi user.
- Kategori Transparency (ISO's Reference Model for ODP)
  - **Access**
    - menyembunyikan penggunaan komunikasi untuk mengakses remote resource sehingga user beranggapan bahwa semua resource adalah lokal.
    - Contoh : pemetaan drive menggunakan Samba Server, NFS
  - **Location**
    - user tidak perlu mengetahui lokasi dari remote resources

# Tantangan perancangan Sistem Terdistribusi (lanjut)

- **Kategori Transparency (*lanjut*)**
  - ***Concurrency***
    - user tidak peduli keberadaan paralel akses ke remote resource dan inkonsistensi dihindari dengan menggunakan mekanisme concurrency control.
  - ***Replication***
    - menyembunyikan perbedaan-perbedaan antara layanan replicated dan non-replicated.
  - ***Failure***
    - menyembunyikan pengaruh partial fail. Hal ini dicapai dengan cara replikasi resources dan menerapkan mekanisme recovery
  - ***Migration/Mobility***
    - Kemampuan untuk melakukan relocate resource secara dinamik tanpa kekuatiran rekonfigurasi ulang dari user.

# Tantangan perancangan Sistem Terdistribusi (lanjut)

- **Kategori Transparency (*lanjut*)**
  - ***performance***
    - meminimalkan tambahan performance dalam menggunakan remote resource, sehingga response time dan throughput sebanding dengan ketika mengakses resource secara lokal.
  - ***scaling***
    - Menyembunyikan variasi dalam kelakuan sistem ketika dilakukan perubahan scope atau skala sistem. Scaling seharusnya tidak memerlukan perubahan besar untuk struktur sistem dan operasi untuk mengakomodasi perubahan skala tersebut. Skala diukur dalam hal kecepatan (slow to fast), size (small to large), geographical scope (local or remote).