

# Sistem Terdistribusi 1

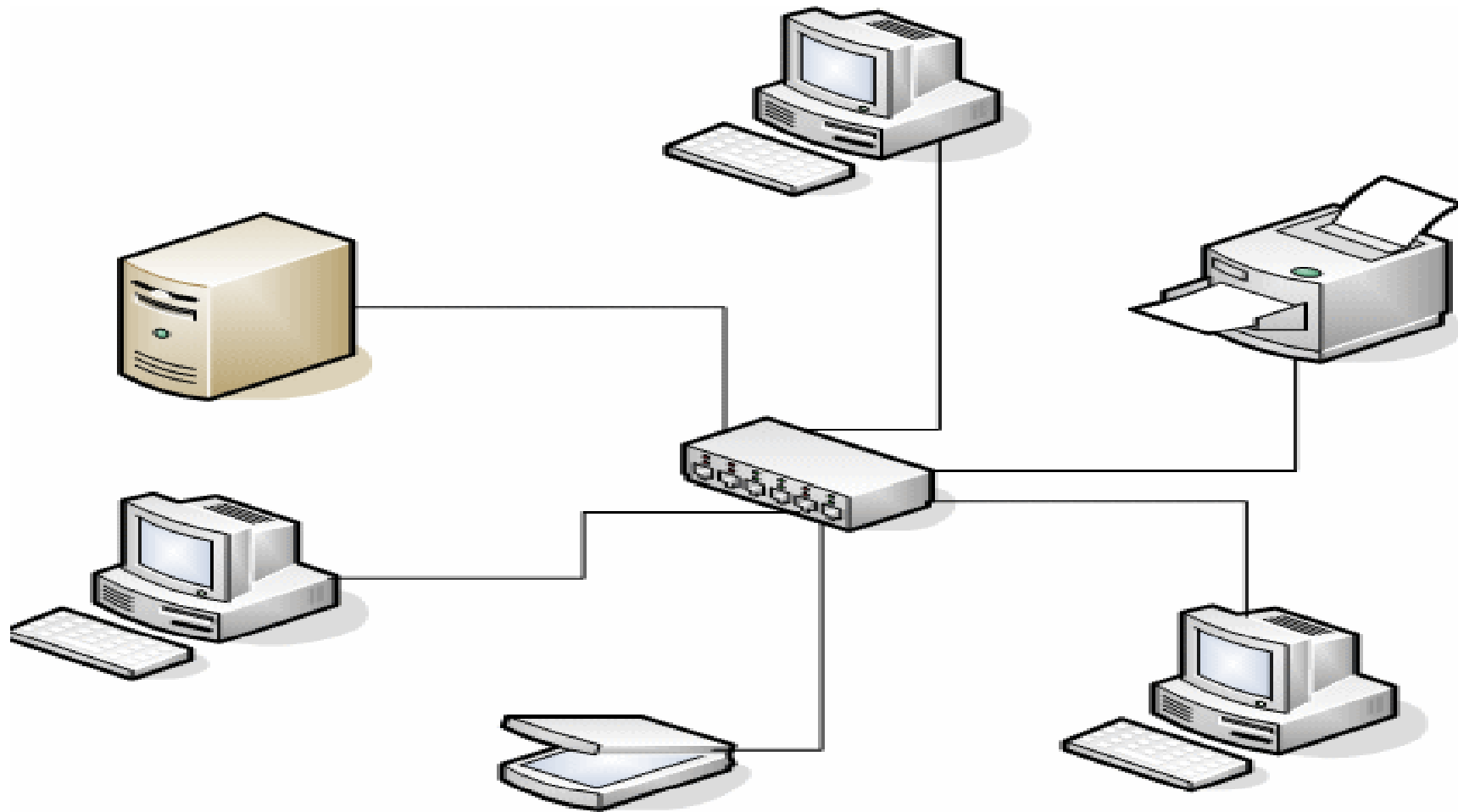
## Introduction

Pengantar Sistem Terdistribusi &  
Overview

# JarKom vs SisTer

- **Jaringan komputer** : komputer otonom yang secara eksplisit terlihat (secara eksplisit teralamatasi)
  - Dengan IP address masing-masing komputer
- **Sistem terdistribusi**: keberadaan beberapa komputer otonom bersifat transparan, sebagai satu kesatuan. (tanenbaum)
- Secara normal, setiap sistem terdistribusi mengandalkan layanan yang disediakan oleh jaringan komputer
  - Berbasis TCP/IP

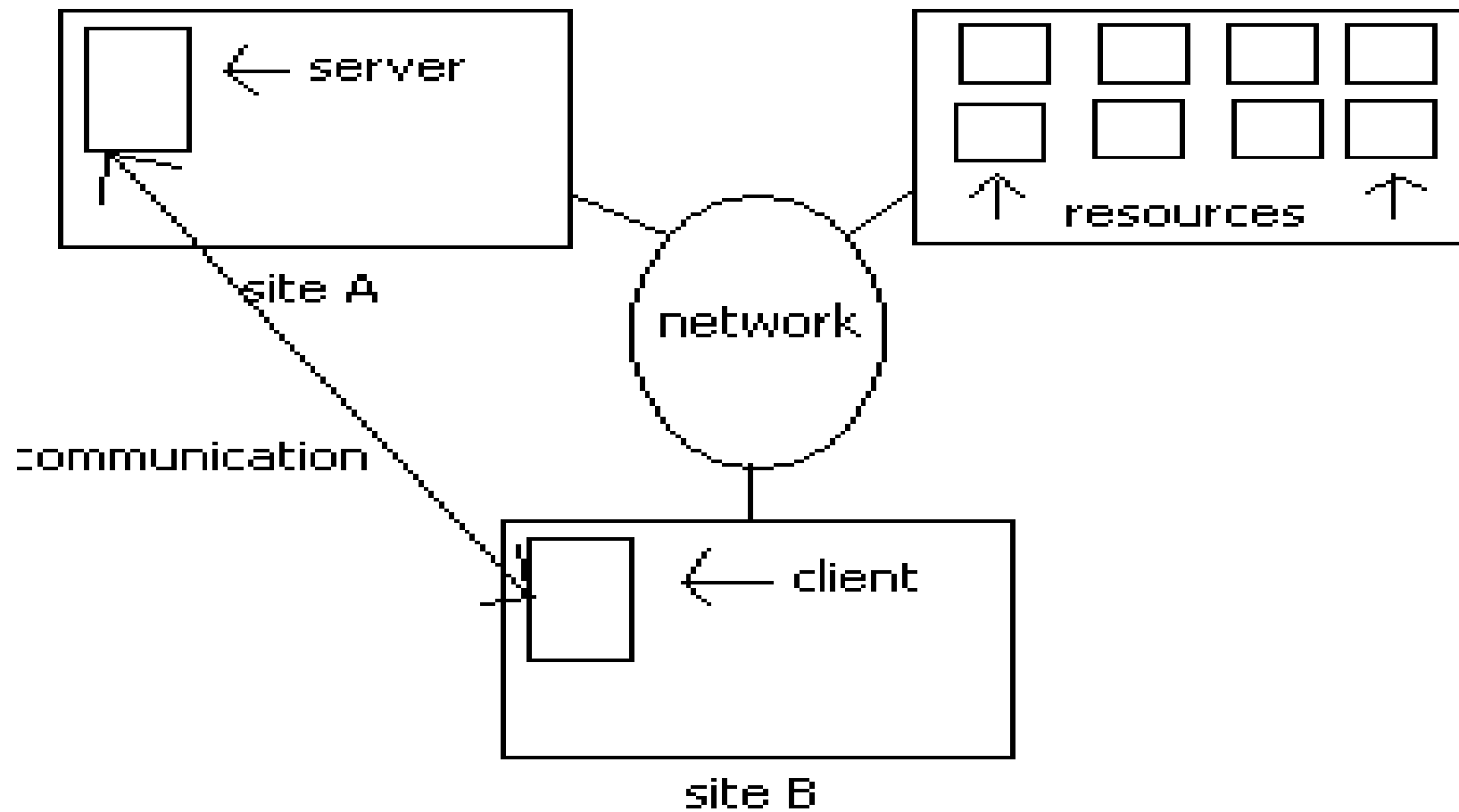
# Jaringan komputer



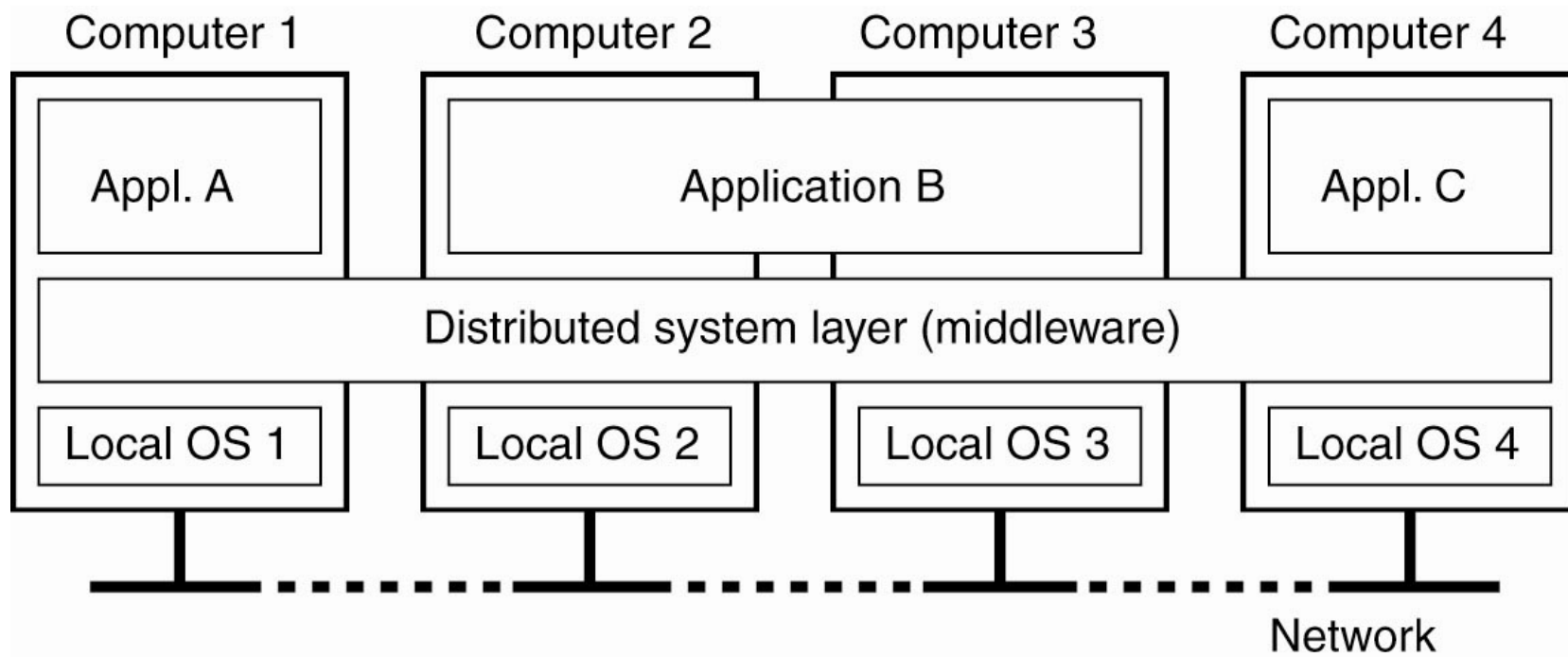
# Sistem Terdistribusi

- Satu sistem dimana **beberapa komputer** pada **jaringan** saling berkomunikasi, berkoordinasi, dan bekerja sama dengan cara saling **bertukar pesan** (messages)
  - Komputer-komputer saling independen
    - Memiliki memori dan prosesor sendiri
  - Dihubungkan dalam jaringan komputer
    - LAN / WAN
  - Terlihat sebagai satu kesatuan
  - Komputasi terintegrasi
  - Dapat diterapkan pada middleware (tanenbaum)

# Ilustrasi



# DS in middleware



A distributed system organized as middleware. The middleware layer extends over multiple machines, and offers each application the same interface.

# Scalability problems

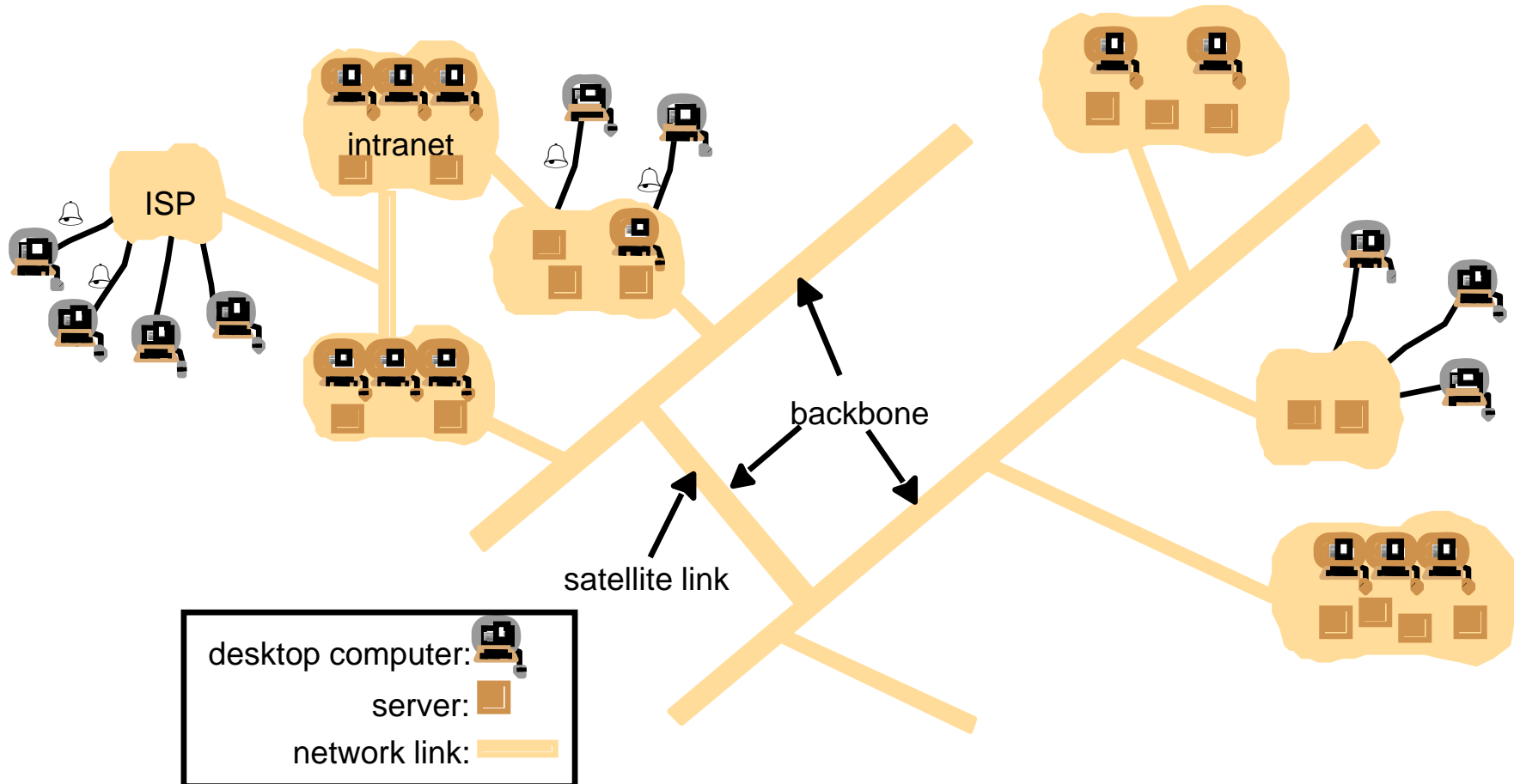
- Centralized services: single service for all requests
- Centralized data: single data point for all services
- Centralized algorithms: single computation for all requests

# Contoh SisTer

- Internet = Interconnection Network
- Intranet Corporation
- Mobile Computing
- Automated banking systems
- Tracking roaming cellular phones
- Global positioning systems
- Retail point-of-sale terminals
- Air-traffic control



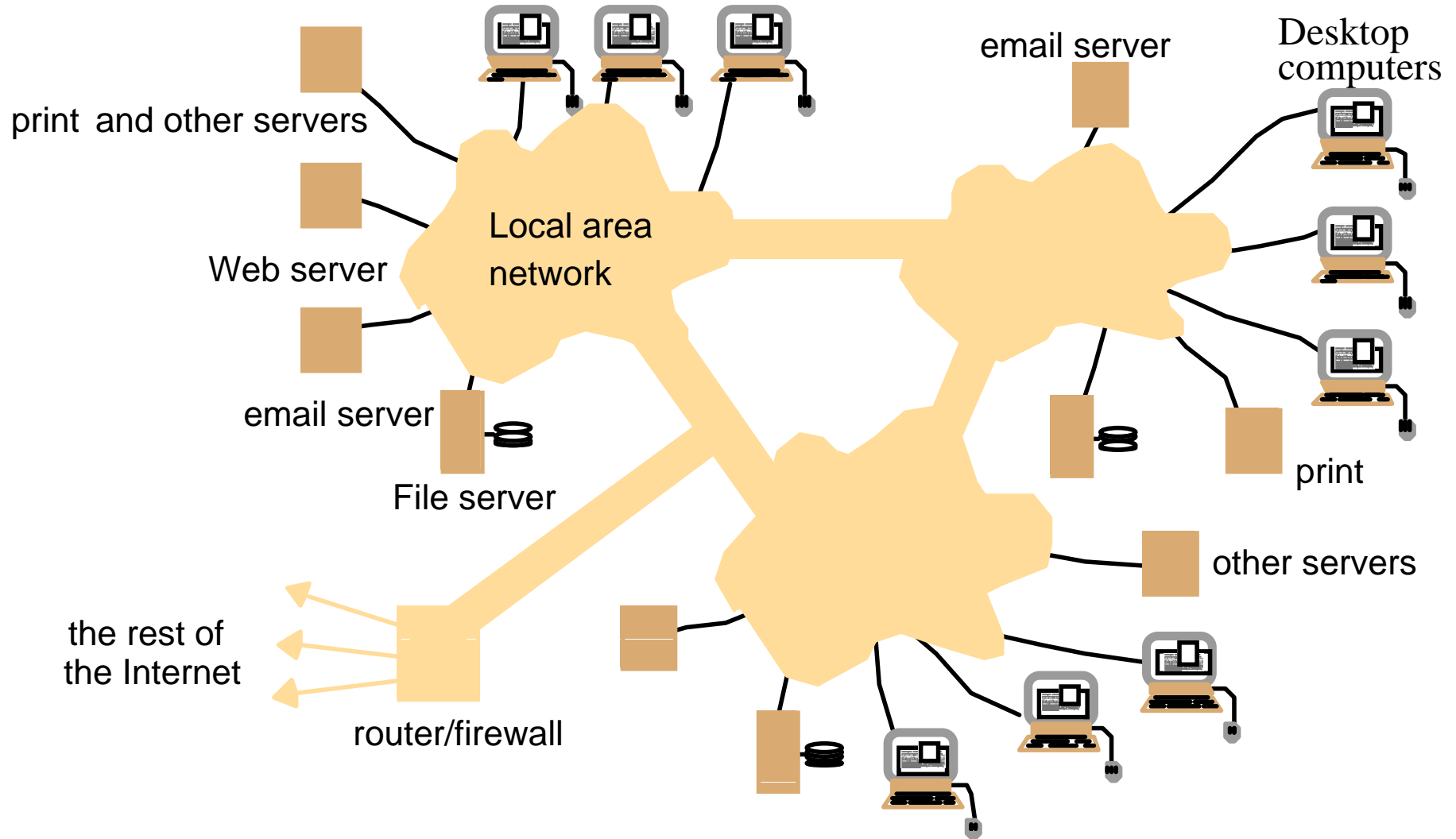
# Internet



# Intranet

- **Intranet** adalah sebuah jaringan privat yang menggunakan protokol-protokol Internet (TCP/IP), untuk membagi informasi rahasia **perusahaan** atau operasi dalam perusahaan tersebut kepada karyawannya.
- Bersifat internal (cth: web internal)
- Untuk membangun sebuah intranet, maka sebuah jaringan haruslah memiliki beberapa komponen yang membangun Internet, yakni protokol Internet (Protokol TCP/IP, alamat IP, dan protokol lainnya), klien dan juga server.
- Biasanya proprietary
- Terhubung ke internet (melalui firewall)

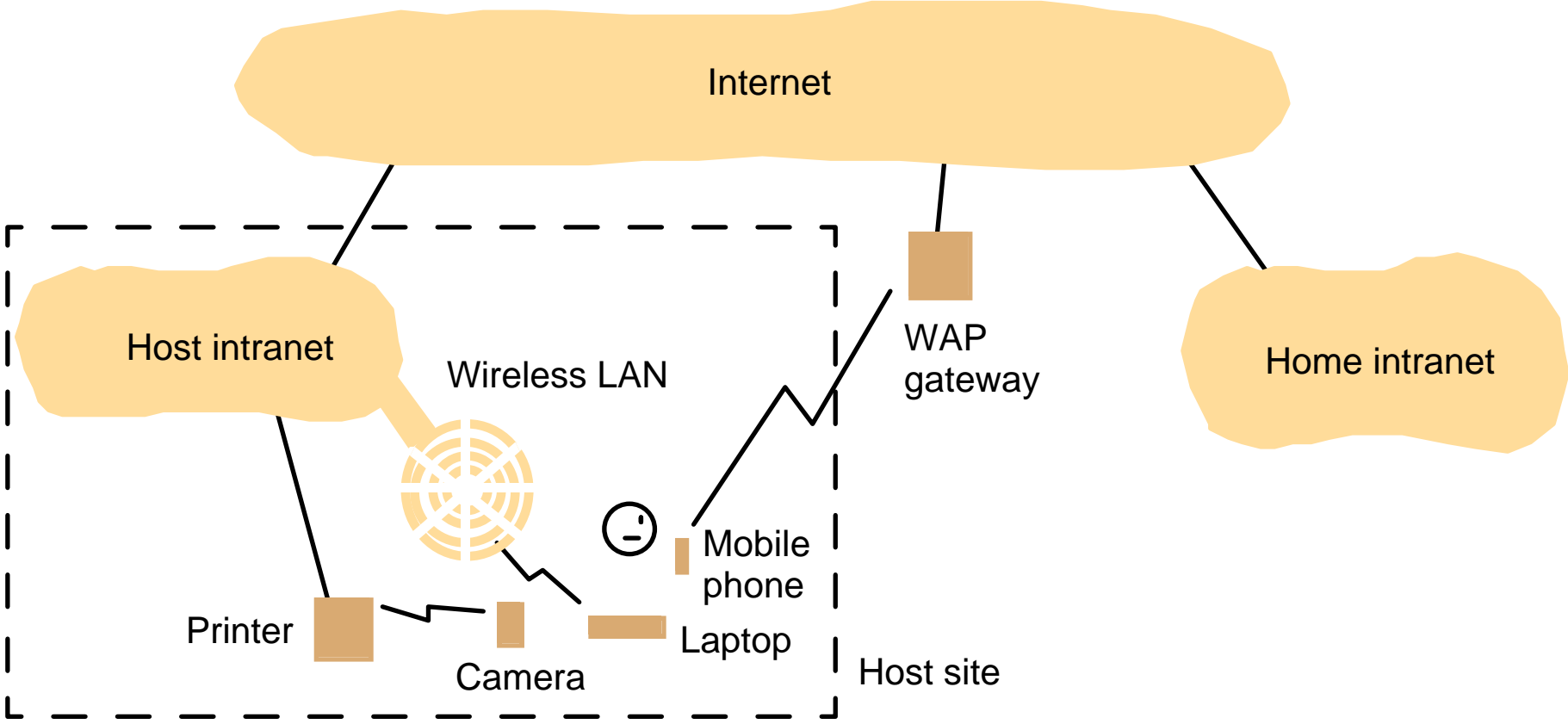
# Intranet



# Sistem terdistribusi multimedia

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

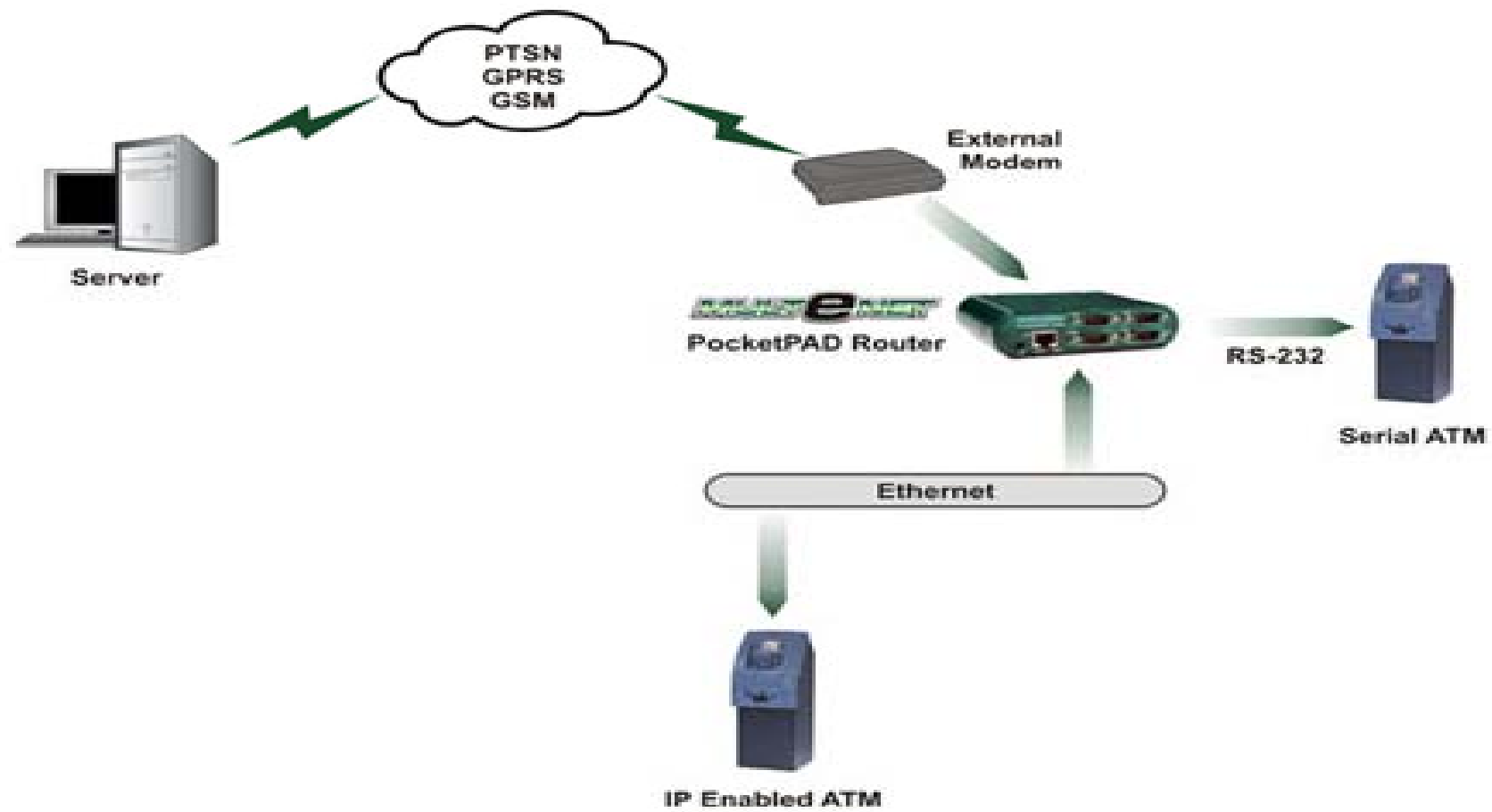
# Mobile computing



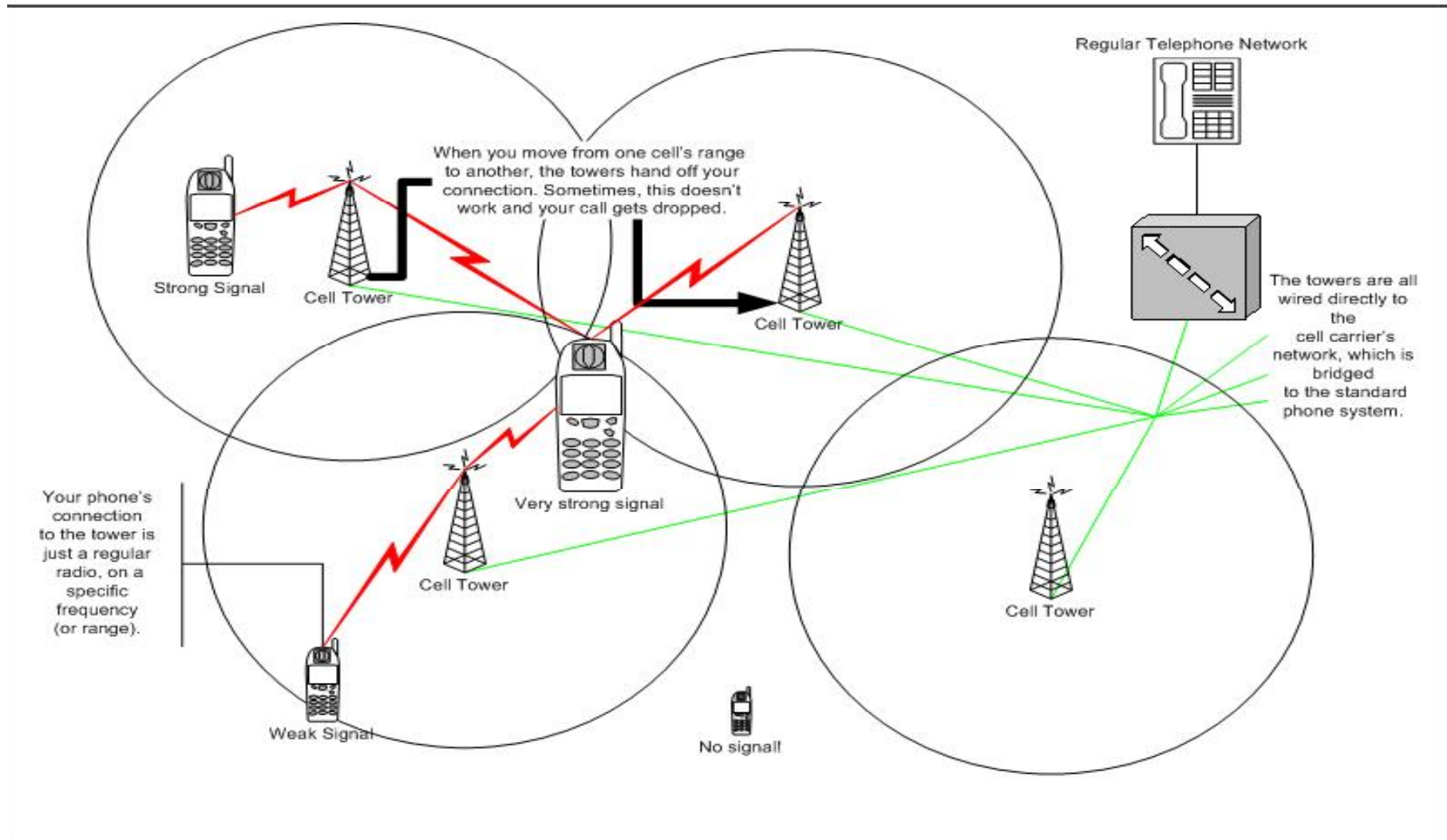
# ATM

- Mesin ATM ada di cabang-cabang bank
- Klien dapat mengakses pada saat yg **simultan bersamaan**
  - Mekanisme deadlock & sinkronisasi
- Sistem ATM akan menggunakan central-central office terdekatnya
  - Relay mode
- Setiap central office akan menjadi backup bagi yang lainnya
  - Replication
- Bagaimana handle transaksi? Keamanan? Network failure?
  - Security

# ATM

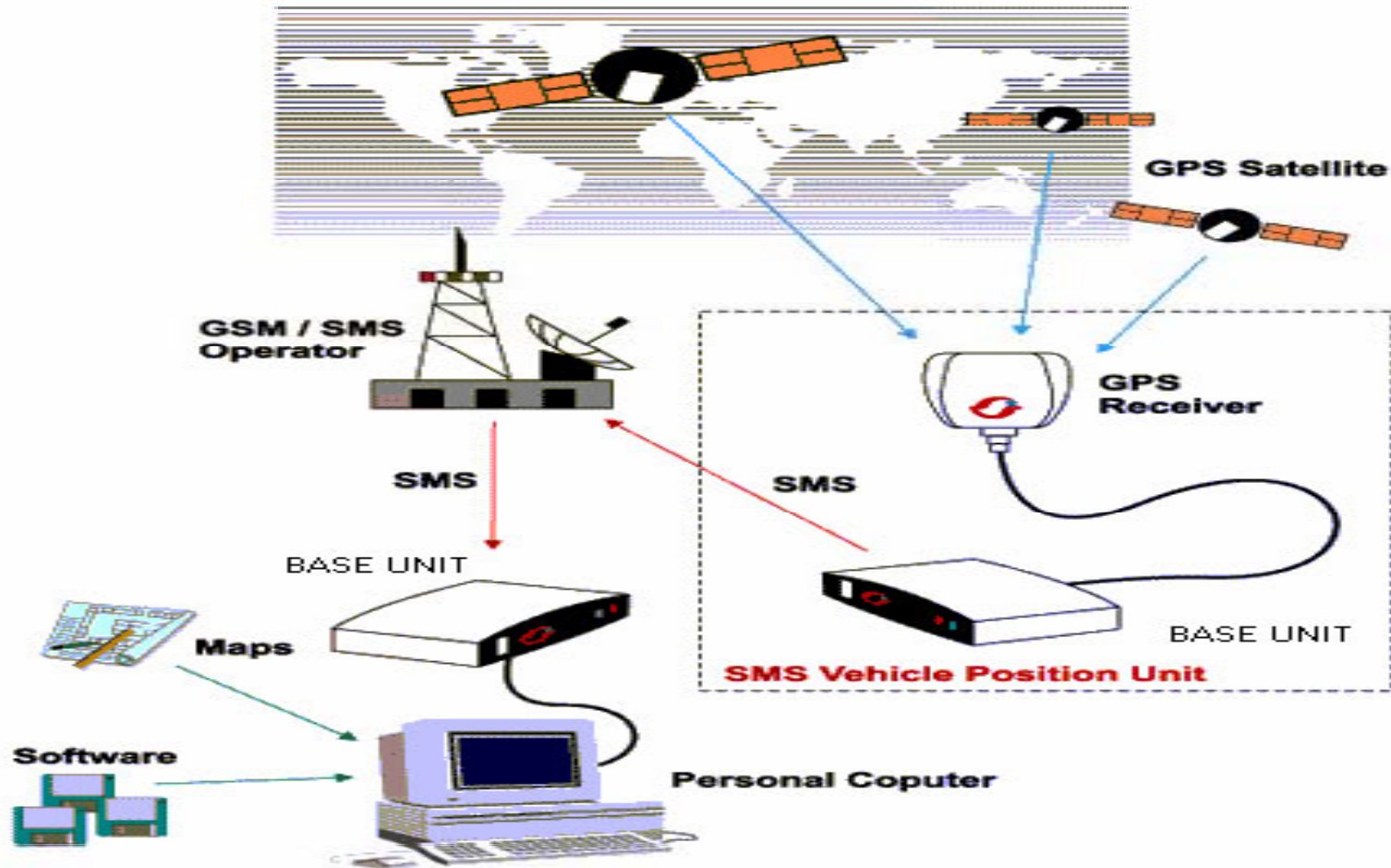


# Tracking Cellular Phone





# GPS



# Contoh SisTer yg lain

- Sistem telepon
  - ISDN, PSTN
- Manajemen jaringan
  - Administrasi resource 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)

# Alasan SisTer

- **Resources sharing:** sumber daya dapat digunakan secara bersama / bergantian
- **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.

# Alasan SisTer

- **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
  - Mis: 10000 CPU, masing-masing berjalan pada 50 MIPS, mencapai 500000 MIPS,
  - Maka satu perintah dijalankan dalam waktu 0.002 nsec
- **Reliability** : dalam sistem terdistribusi, apabila sebuah situs mengalami kegagalan, maka situs yang tersisa dapat melanjutkan operasi yang sedang berjalan. Hal ini menyebabkan reliabilitas sistem menjadi lebih baik.

# Alasan SisTer

- **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
- **Fleksibilitas** : komputer yang berbeda dengan kemampuan yang berbeda dapat di share antar user

# Kesulitan

- 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

# Pitfalls when Developing Distributed Systems

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.
- There is one administrator.

# Karakteristik SisTer

- **Concurrency:** Beberapa komputer dapat berjalan sekaligus dengan tugas yang berbeda
  - Sinkronisasi dan koordinasi dengan message passing
  - Sharing resources
    - Contoh: WEB diakses oleh bnyk orang
  - Masalah umum dalam sistem concurrent
    - Deadlock
    - Komunikasi yang tidak handal



# Karakteristik SisTer

- **No global clock:** Pada sistem terdistribusi, tidak ada satu proses tunggal yang mengetahui global state sistem saat ini (disebabkan oleh concurrency)
  - Hal ini menyebabkan kesulitan dalam mensinkronkan waktu seluruh komputer/perangkat yang terlibat
- **Independent failure:** kegagalan komputer/jaringan bisa terjadi kapan saja
  - Setiap komponen/perangkat dapat mengalami kegagalan namun komponen/perangkat lain tetap berjalan dengan baik.

# Tantangan SisTer

- **Heterogeneity:**
  - Infrastruktur jaringan
  - Hardware dan software (sistem operasi, perbedaan UNIX socket dan Winsock)
  - Bahasa pemrograman
  - Solusi: Perlu ada protokol yang standar, Middleware (contoh : CORBA), Kode program universal (contoh : JAVA)
- **Scalability:** Sistem tetap efektif meskipun terdapat peningkatan resource dan pengguna secara signifikan

# Tantangan

- **Openness**

- Memastikan sistem dapat diperluas dan mudah dalam pemeliharaan

- Mengikuti standard antarmuka
- Solusi: Adanya publikasi dari spesifikasi (RFC)

- **Security**

- Confidentiality (pencegahan terhadap hak akses oleh orang yang tidak berhak)

- Integrity (pencegahan terhadap perubahan data)

- Availability (pencegahan terhadap masalah ketersediaan, misalnya mencegah DDOS)

# Tantangan

- **Menghandle Kegagalan:** Kesalahan/Kegagalan bisa ditemukan/diperbaiki A.S.A.P dan mampu melakukan proses recovery
  - Pendeteksian, Toleransi dan Redudancy
  - Solusi: Replikasi, Load Balancing, Backup
- **Konkurensi:** Banyak client yang mengakses banyak data dalam waktu yang bersamaan, sedangkan data harus tetap konsisten!
  - Menghindari masalah deadlock

# Transparansi

- **Transparency:** Sistem terlihat sebagai satu kesatuan, bukan gabungan dari bbrp komponen

# Transparansi

- ***Access transparency***: memungkinkan resource lokal / remote untuk diakses menggunakan operasi yg sama (tidak berbeda-beda)
- ***Location transparency***: memungkinkan resources untuk diakses tanpa pengetahuan ttg jaringan fisik/lokasi (lokasi dan IP address).
- ***Concurrency transparency***: memungkinkan beberapa proses untuk beroperasi secara konkuren menggunakan shared resources tanpa “menggangu” mereka.
- ***Replication transparency***: memungkinkan multiple instances dari resources untuk digunakan menaikan reliability dan performance tanpa pengetahuan pemrograman replikasi.

# Transparansi

- ***Failure transparency***: memungkinkan penyembunyian kegagalan, memperbolehkan users dan program aplikasi untuk menyelesaikan tugas mereka walaupun ada kegagalan komponen hardware / software.
- ***Mobility transparency***: memungkinkan perubahan resources dan clients didalam sistem tanpa berefek pada operasi user dan program.
- ***Performance transparency***: memungkinkan sistem untuk dikonfigurasi ulang untuk meningkatkan performa yang berubah secara cepat.
- ***Scaling transparency***: memperbolehkan sistem dan aplikasi untuk diperluas tanpa mengubah struktur sistem atau algoritma aplikasi.

# Pengembangan Lebih lanjut

- Distributed Database
  - A logically interrelated collection of shared data (and a description of this data), physically distributed over a computer network
  - Penyimpanan data bisa dilakukan secara terdistribusi (tidak lagi tersentralisasi)
  - Menggunakan Replikasi dan Fragmentasi
- Distributed Processing
  - Menggunakan RMI, RPC, atau .NET Remoting
- Distributed Transactions

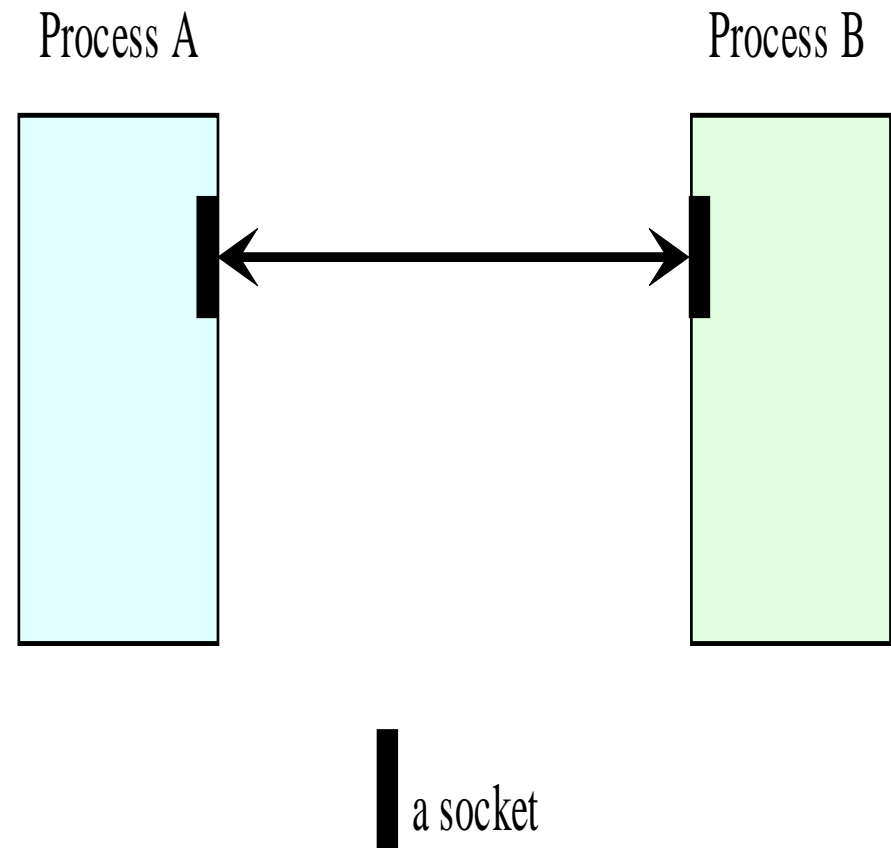


# Prinsip berkomunikasi

- Source
  - generates data to be transmitted
- Transmitter
  - Converts data into transmittable signals
- Transmission System
  - Carries data
- Receiver
  - Converts received signal into data
- Destination
  - Takes incoming data

# Socket

- Socket adalah sebuah abstraksi perangkat lunak yang digunakan sebagai suatu "terminal" dari suatu hubungan antara dua mesin atau proses yang saling berinterkoneksi.
- **End to end communication**



# Distributed Objects

- Located **separately** on each host
- Must communicate with others
  - Interprocess communication
    - RPC (Remote Procedural Call)
    - RMI (Remote Method Invocation)
    - CORBA (Common Object Request Broker Architecture)
    - XML RPC & Web Service
- Transparency
  - Location
- Receive events notification from other objects

# Distributed programming paradigms

- Client/server model
- Remote procedure calls
- Distributed File Systems
- Group communication and multicasts
- Distributed transactions
- Distributed object-based systems
- Publish-subscribe model
- Peer-to-peer model
- The Web

# DOS dan DFS

- Pada perkembangannya sistem operasi juga dibuat terdistribusi
- Menciptakan juga konsep DFS
  - Distributed file system (**DFS**) adalah sebuah sistem di mana banyak pengguna dapat berbagi berkas dan sumber daya penyimpanan.
  - Muncul masalah mengenai penamaan resource terdistribusi:
    - Human oriented vs machine
    - URI

# Operating Systems

- Flexibility
  - Monolithic kernel vs. Microkernel
  - Microkernel -- keep it small, use **user-level** servers for system services
    - An interprocess communication mechanism
    - A small amount of low-level process management and scheduling
    - Low-level input / output
  - Monolithic kernel
    - provide the file system and directory system
    - full process management, and much system call handling

# NEXT

- Model sistem terdistribusi