

Sistem Terdistribusi 2

Model arsitektur Terdistribusi

Masalah Membangun SisTer

- Dimana kita meletakkan bagian hardware?
- Dimana kita meletakkan bagian software?
- Bagaimana antara hardware & software berkomunikasi?

Kesulitan-kesulitan

- Mode pemakaian
 - Variasi yang beragam terhadap karakteristik pemakaian sistem
 - Berapa kali suatu halaman dikunjungi?
 - Seberapa sibuk suatu server?
- Masalah Internal
 - Masalah konkurensi
- Masalah lingkungan sistem
 - Masalah heterogenitas: hardware, sistem operasi dan jaringan
- Masalah ancaman eksternal
 - Masalah keamanan data

Model Arsitektur SisTer

- **Model**

Menyediakan sebuah gambaran abstrak aspek yang relevan dengan sistem

- **Tujuan**

Menyediakan sebuah kerangka kerja untuk memahami permasalahan dan pemecahannya

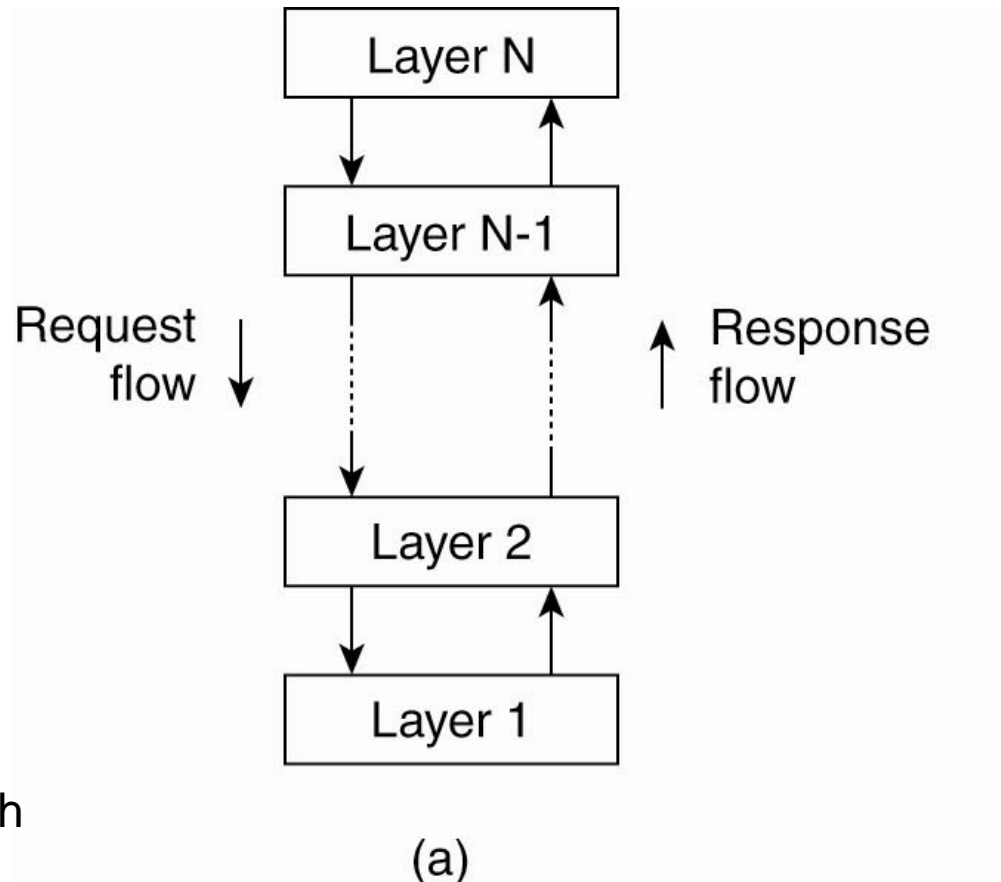
- **Model Arsitektur**

Hubungan dan interkoneksi seperti apa antara komponen komponen dari sistem terdistribusi

Model Arsitektur SisTer

- **Software architecture**
 - Organisasi logika dari komponen-komponen software
 - Ada 4 jenis:
 - Layered architectures
 - Object-based architectures
 - Data-centered architectures
 - Event-based architectures
- **System Architecture**
 - Placement of machines
 - Placement of software on machines

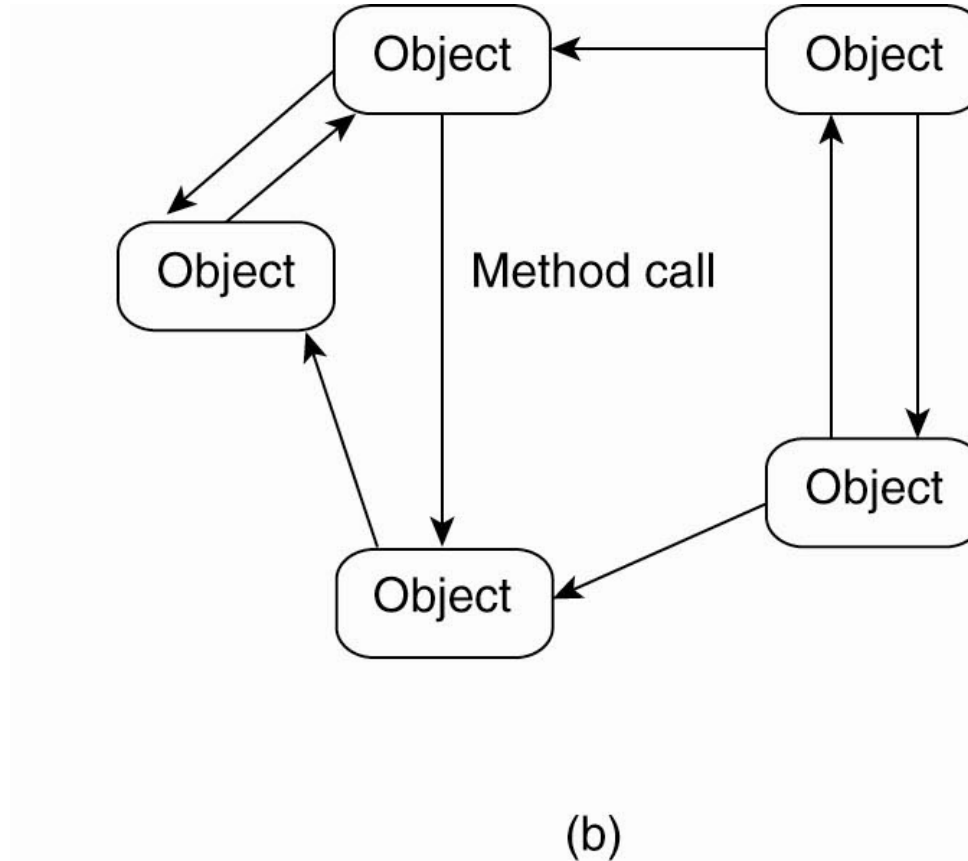
Layered Architecture



Contoh

- Windows
- Linux
- Macintosh
- Sun OS

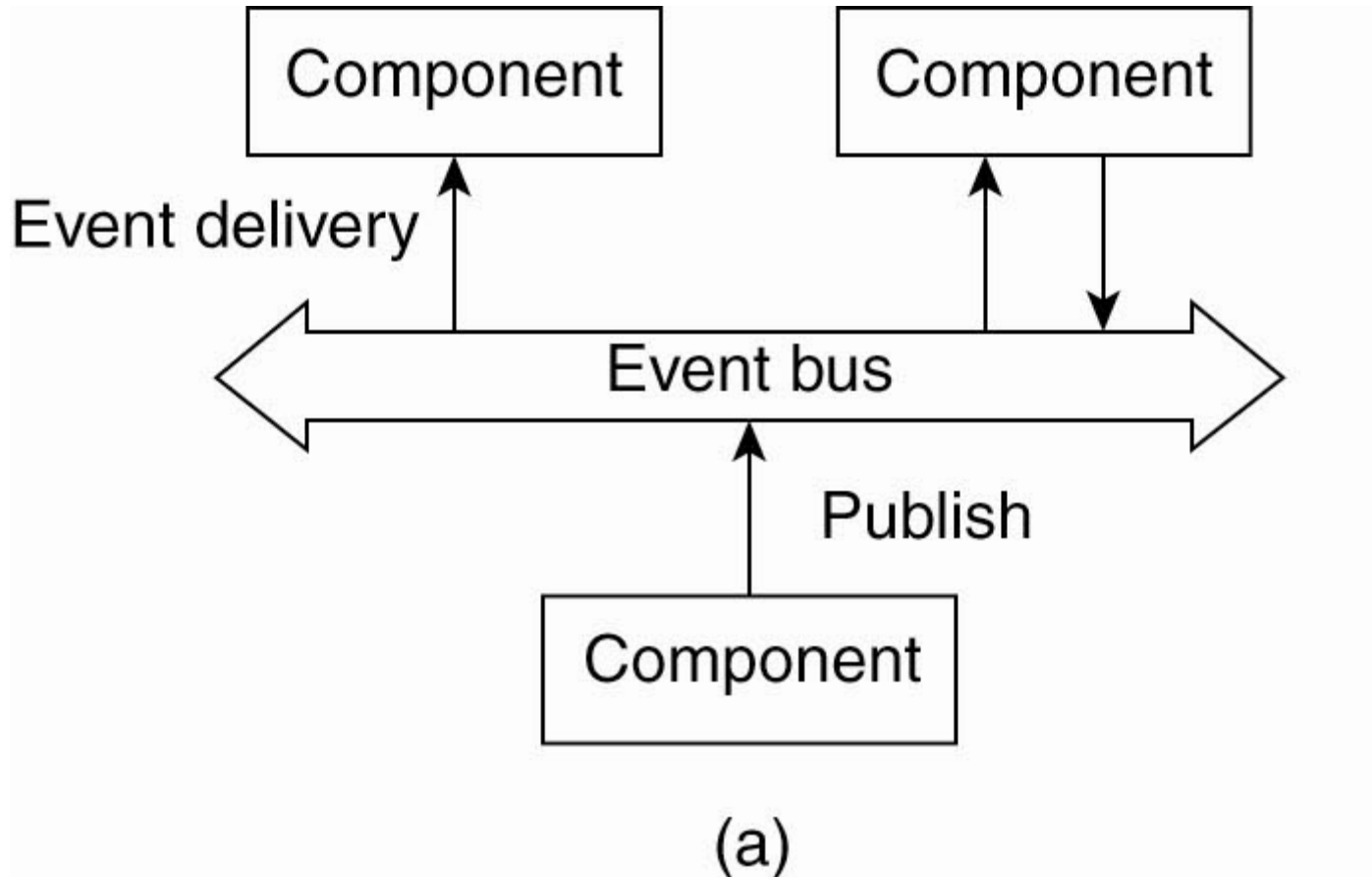
Object based Architecture



Bagus, menggunakan mekanisme **Remote Procedure Call** antar obyek

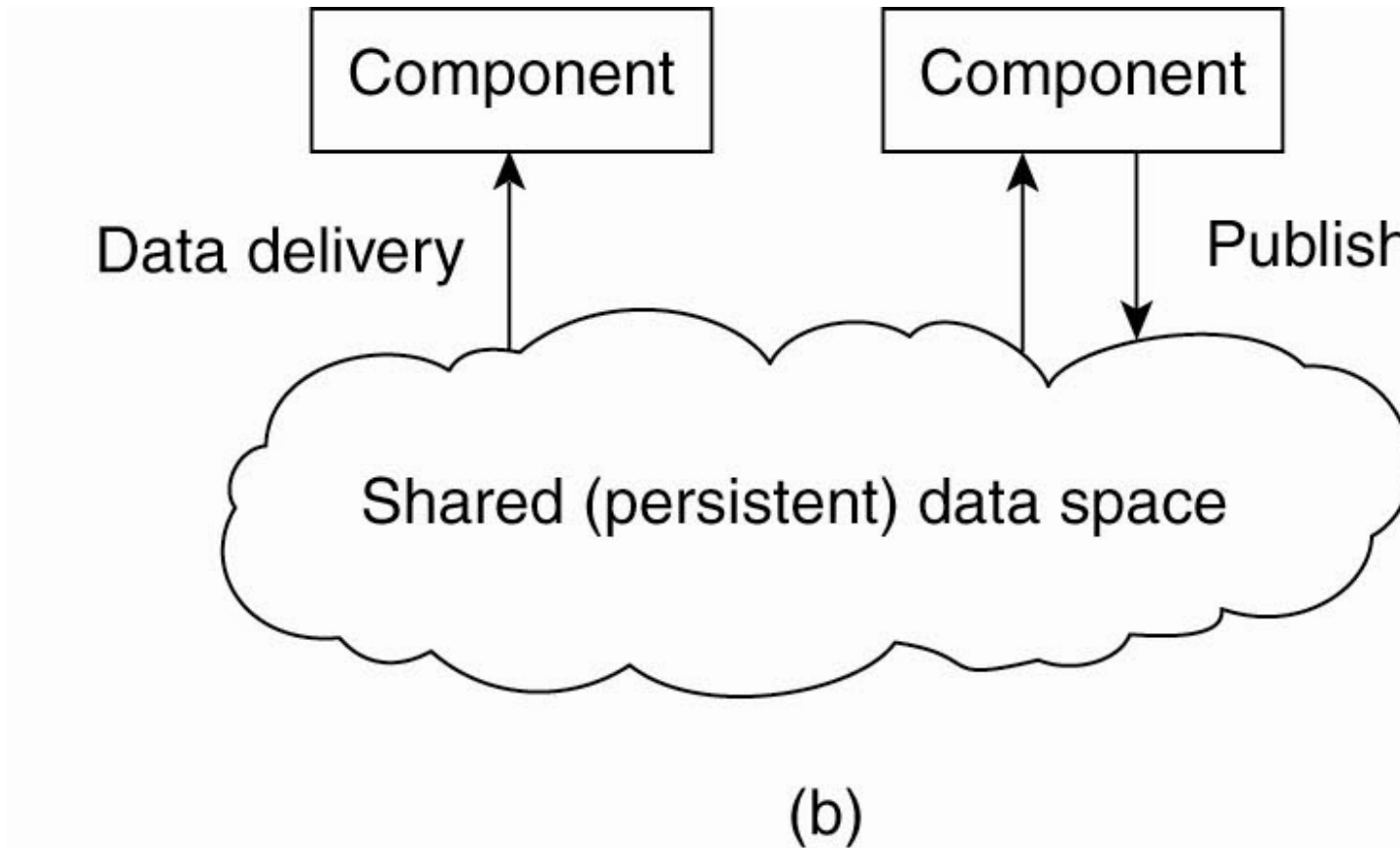
Event-based Architecture

Dikenal sbg: **Publish/Subscribe System**



Proses Berkomunikasi berdasarkan event yg terjadi

Data Centered Architecture



Proses-proses berkomunikasi melalui repository data (database terdistribusi)

Model Arsitektur SisTer

- **Software architecture**

- Logical organization of software components

- Ada 4 jenis:

- Layered architectures

- Object-based architectures

- Data-centered architectures

- Event-based architectures

- **System Architecture**

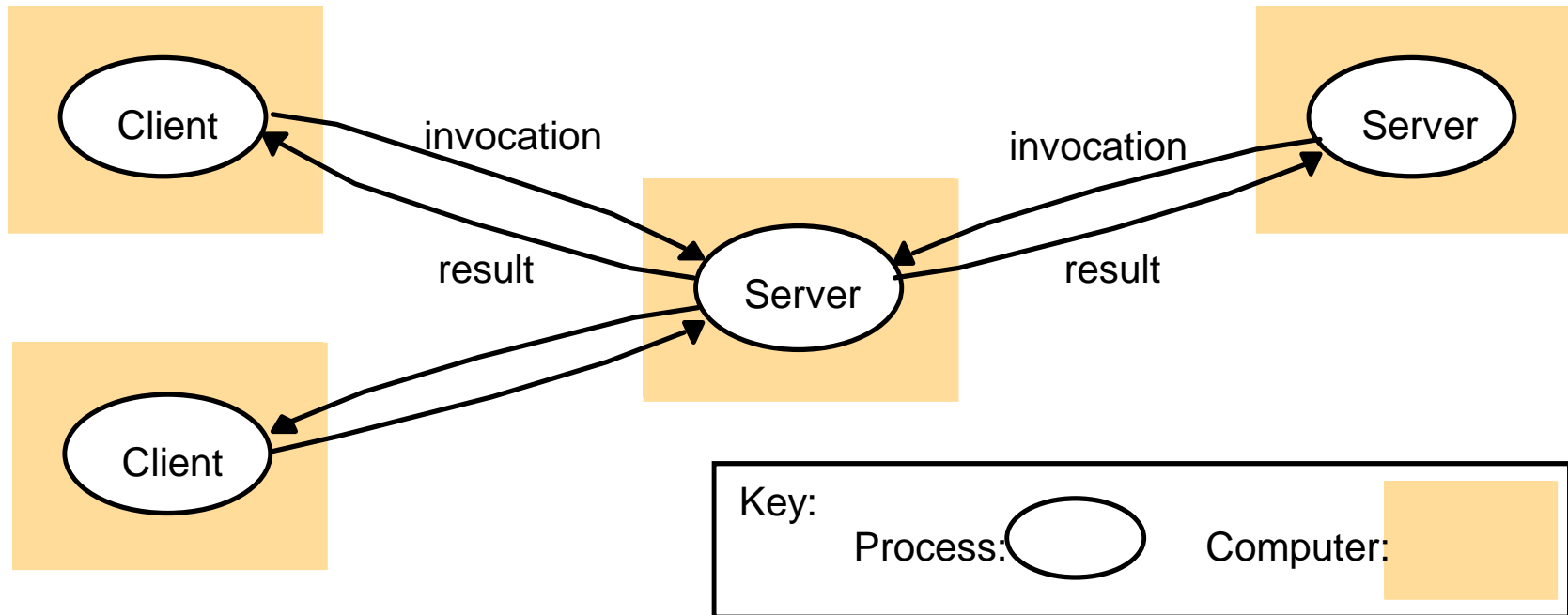
- Placement of machines

- Placement of software on machines

Klasifikasi Proses

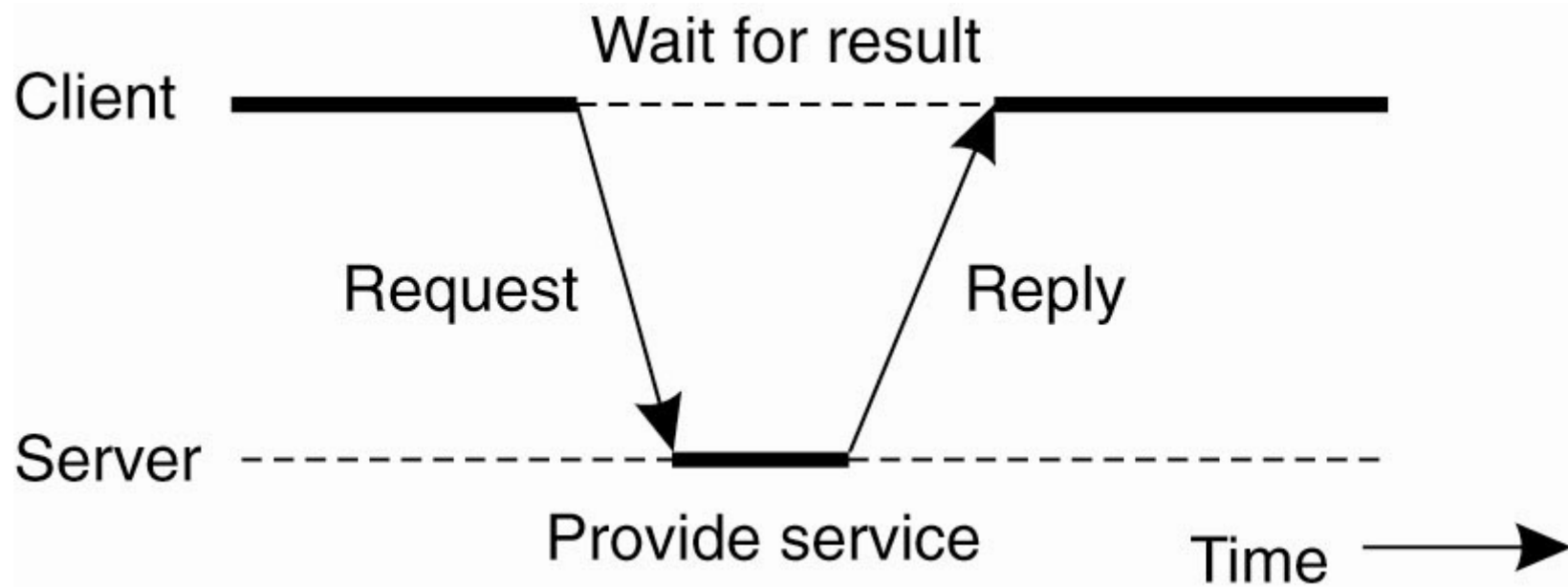
- Proses Server
 - Menyediakan layanan dan menangani request
- Proses Client
 - Proses membuat melakukan request
- Proses Peer
 - Proses yang saling bekerja sama dan berkomunikasi
- Middleware
 - Menyediakan transparansi terhadap keanekaragaman platform proses dan objek pada sekumpulan mesin yang menerapkan protokol untuk aplikasi terdistribusi
 - Cth: CORBA, Java RMI, DCOM

Clients - Servers

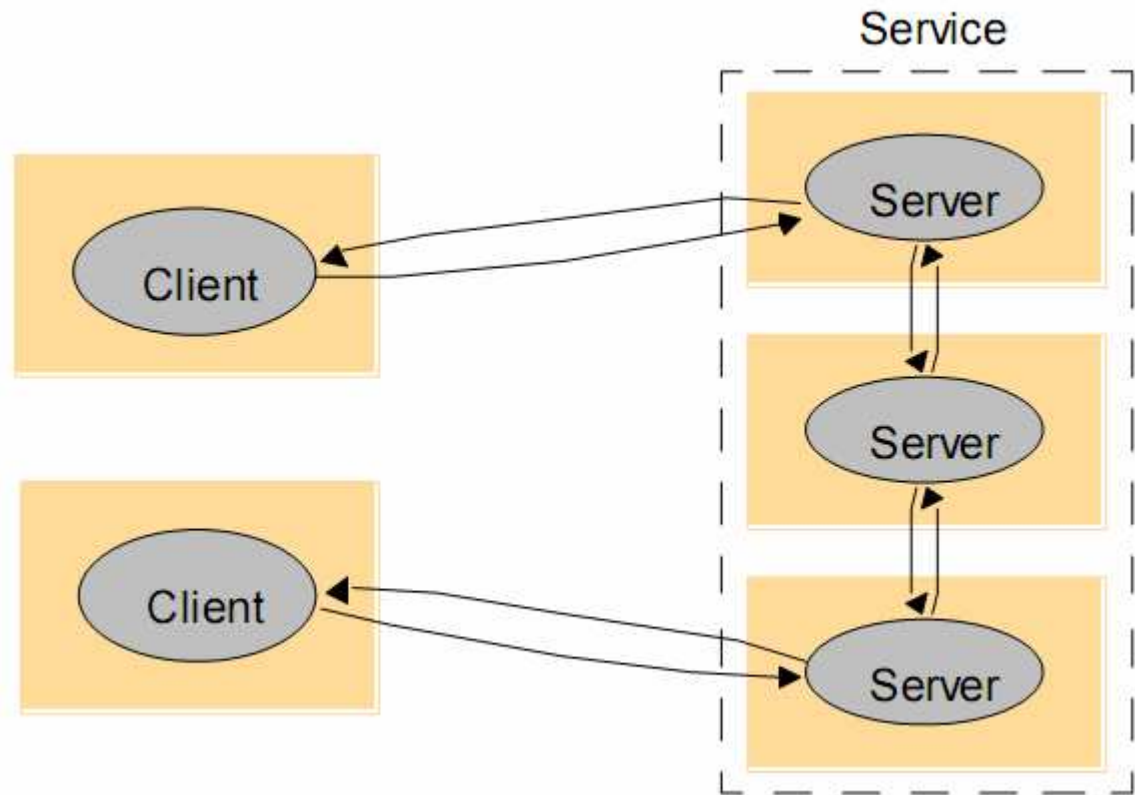


- Bersifat: **centralized architecture**
- Terjadi invocation dari client ke server
- Contoh: client (browser) meminta HTML ke HTTP Server

Interaksi client-server

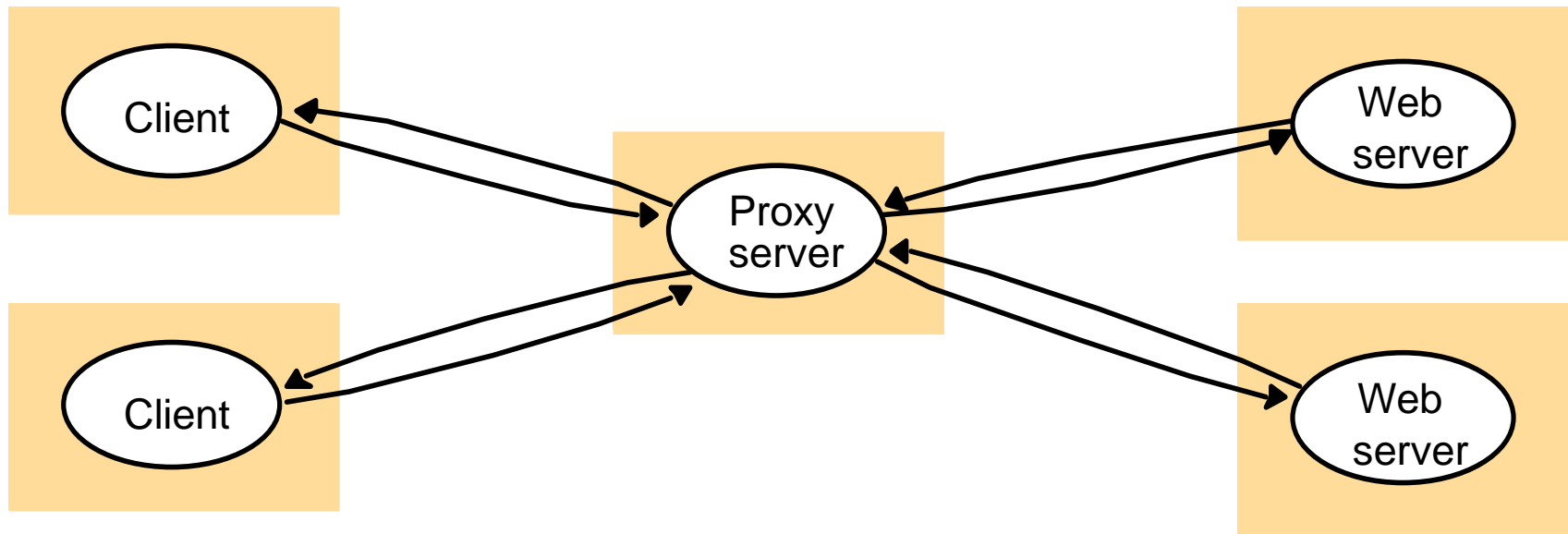


Multiple server



- Service disediakan oleh beberapa server
- Contoh: sebagian besar layanan web komersial diterapkan melalui server fisik yang berbeda
- Untuk: kehandalan, unjuk gigi
- Server menggunakan replikasi atau database terdistribusi

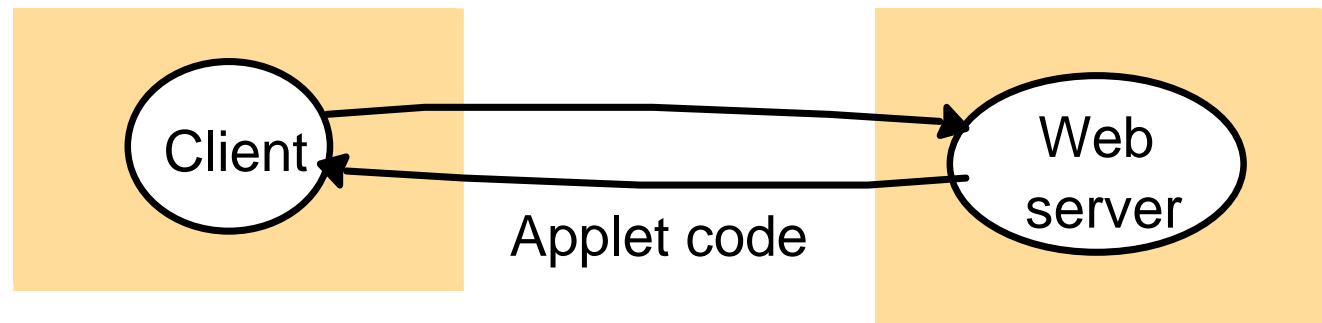
Web proxy server



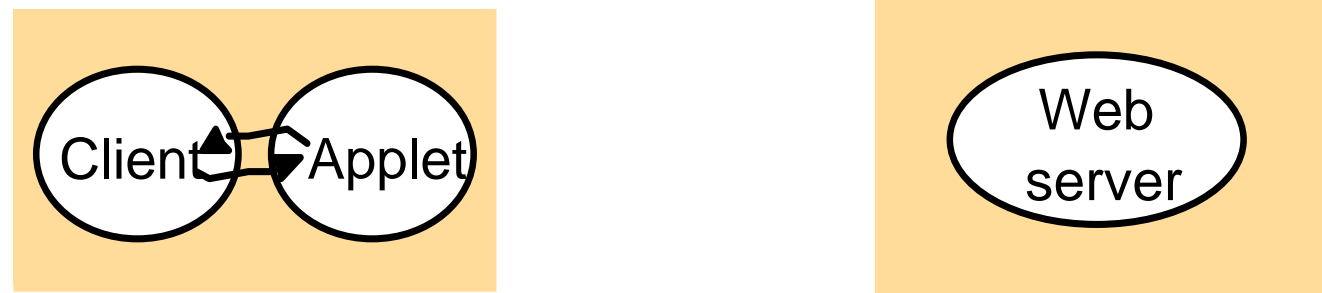
- Server duplikasi informasi untuk bertindak sebagai proxy
- Cache:
 - menyimpan informasi lokal yg sering digunakan
 - Mengurangi beban kerja server
 - Meningkatkan kinerja
 - Wajib digunakan pada search engine

Web applets / Mobile Agent

a) client request results in the downloading of applet code



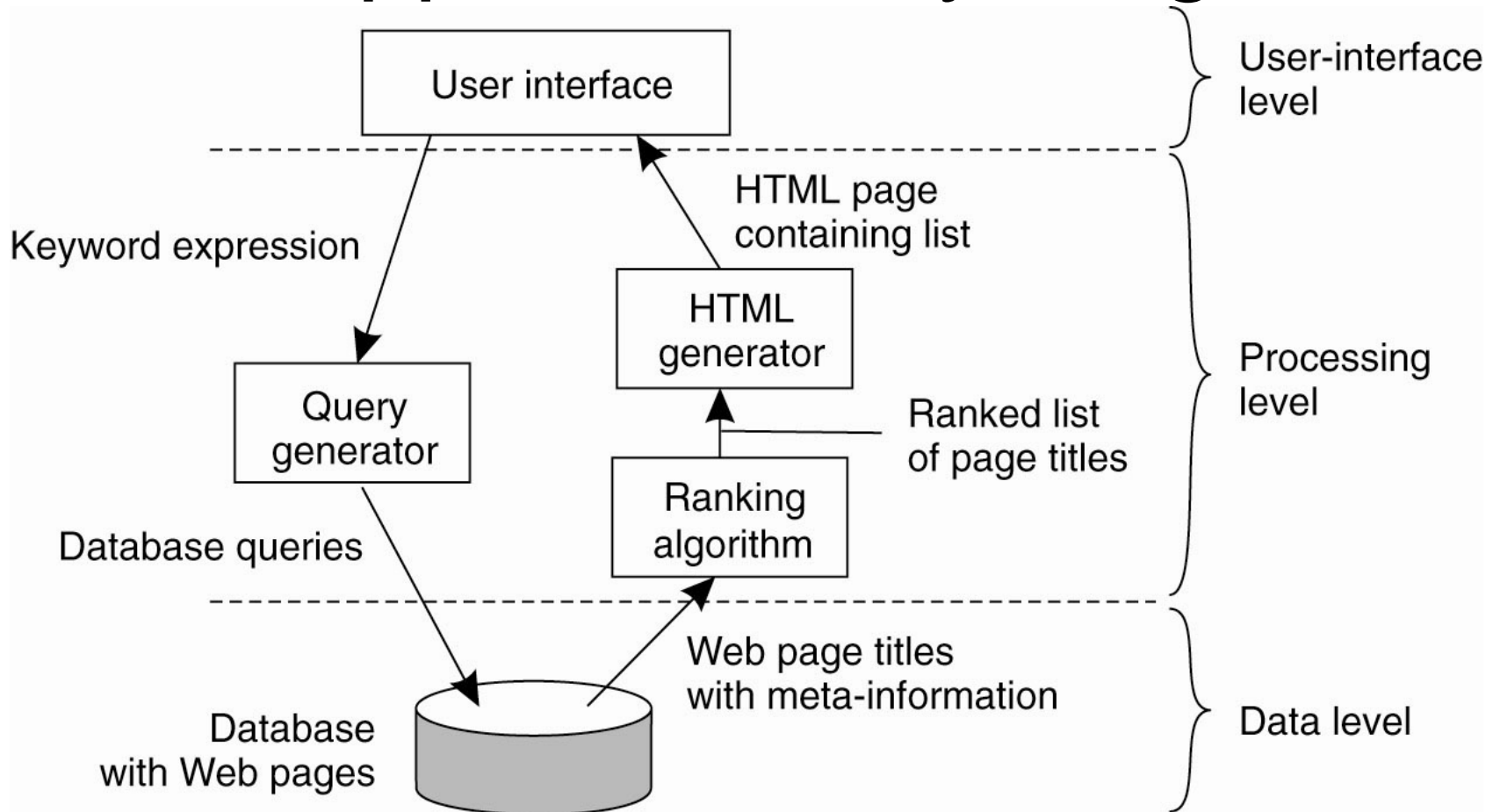
b) client interacts with the applet



Application Layering

- **Data**, menyediakan struktur dan fungsi untuk manipulasi informasi, yang terdiri dari banyak objek data.
 - Sebagai contoh : relational Database, file grafik, file suara atau multimedia data stream
- **Processing**, terfokus pada pemrosesan objek data, yang biasanya disebut sebagai middleware.
- **Presentation**, terfokus untuk membuat data tersedia untuk user dan menangani interaksi dengan user. Ada dua level:
 - user interface
 - presentation management, menyediakan operasi dasar untuk membangun dan mengontrol user interface di bawah kontrol aplikasi. Meliputi : layanan tampilan, kontrol dialog dan API. Contoh : X-Windows, MS Windows

Application Layering

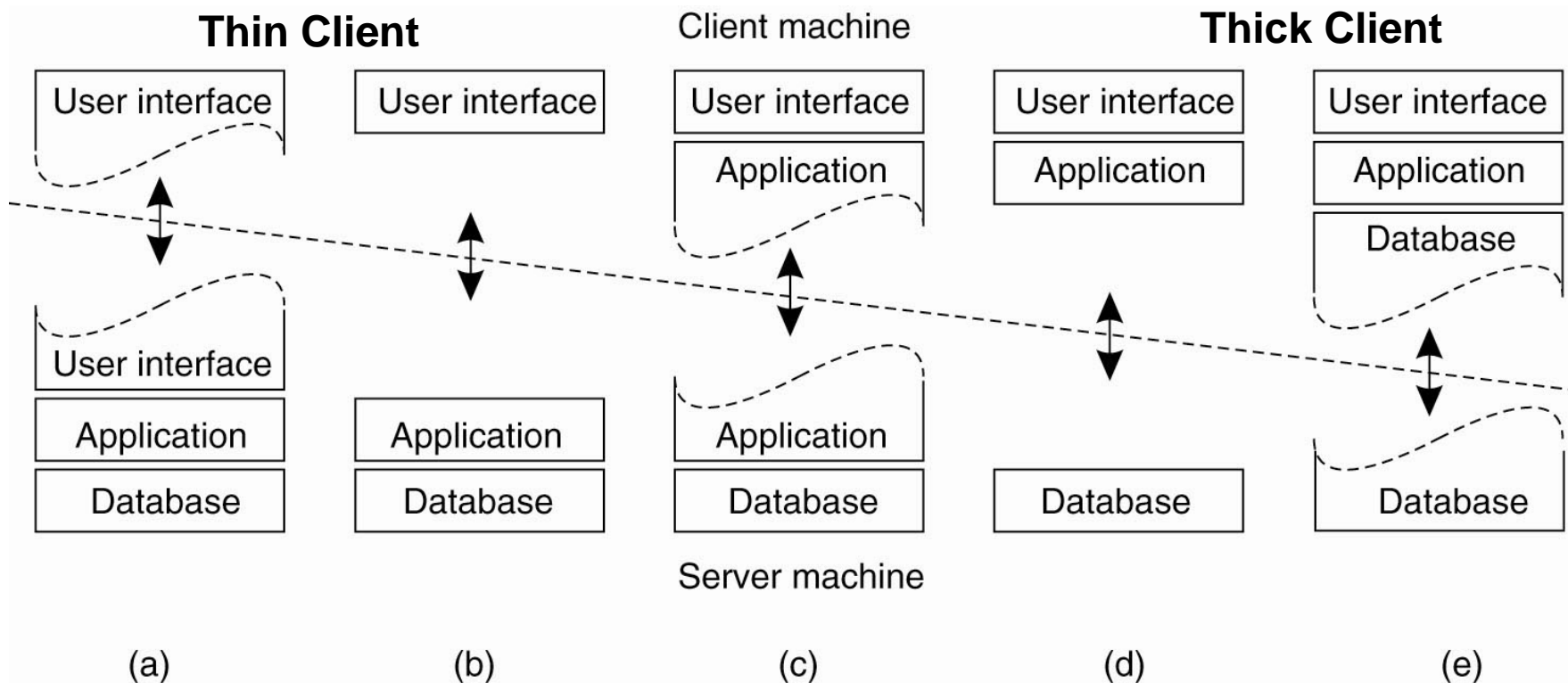


- The simplified organization of an Internet search engine into three different layers.

Multitiered Architectures

- Disebut sebagai: **Decentralized architecture**
- The simplest organization is to have only **two types** of machines:
 - **A client machine** containing only the programs implementing (part of) the user-interface level
 - **A server machine** containing the rest,
 - the programs implementing the processing and data level

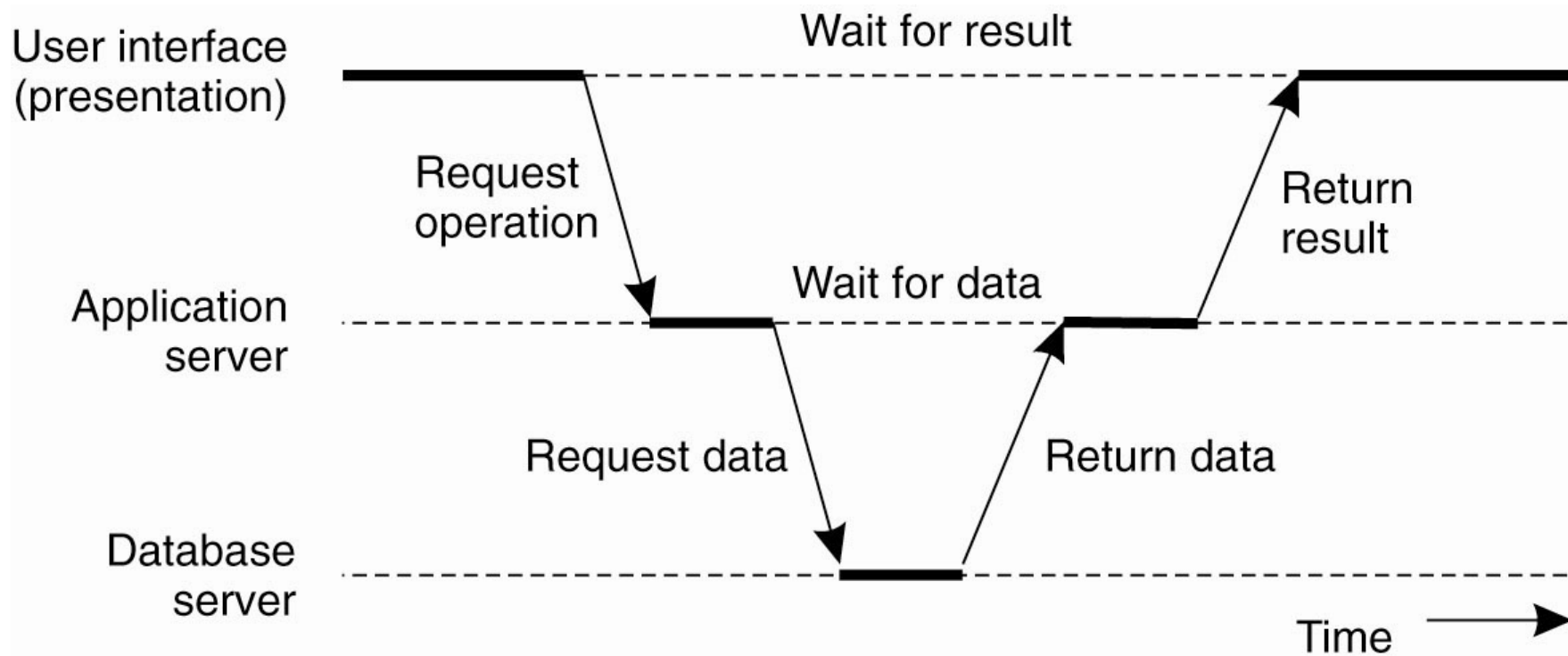
Multitiered Architectures (2)



Contoh Thin Client:

- X11 Window
- VNC

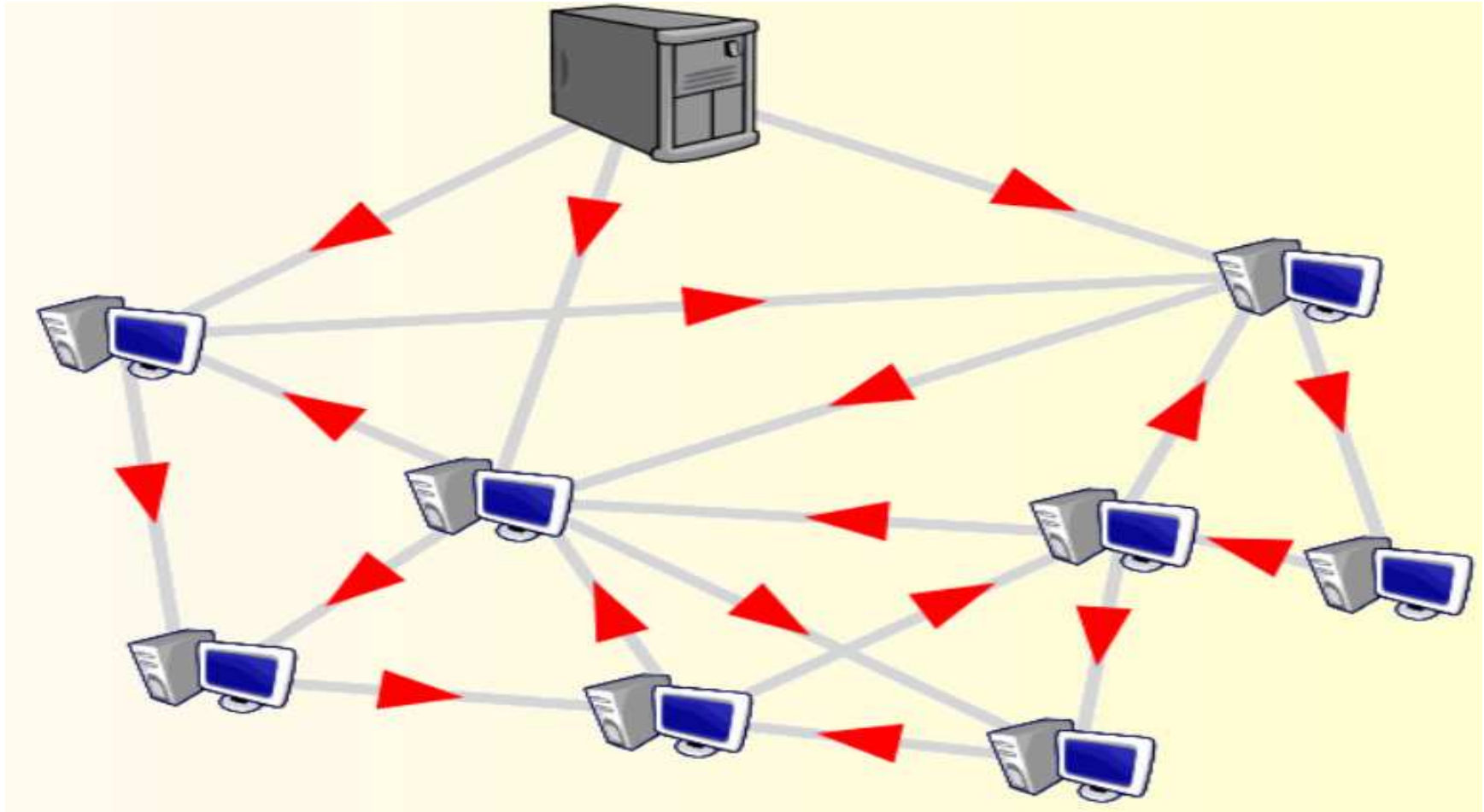
Multitiered Architectures Interaction



Middleware

- Sebuah komponen untuk
 - Menyamarkan heterogeneity
 - Melakukan message passing
- Direpresentasikan dalam bentuk object
- Contoh: Sun RPC, CORBA, RMI, DCOM
- Keuntungan: OS + Hardware independen

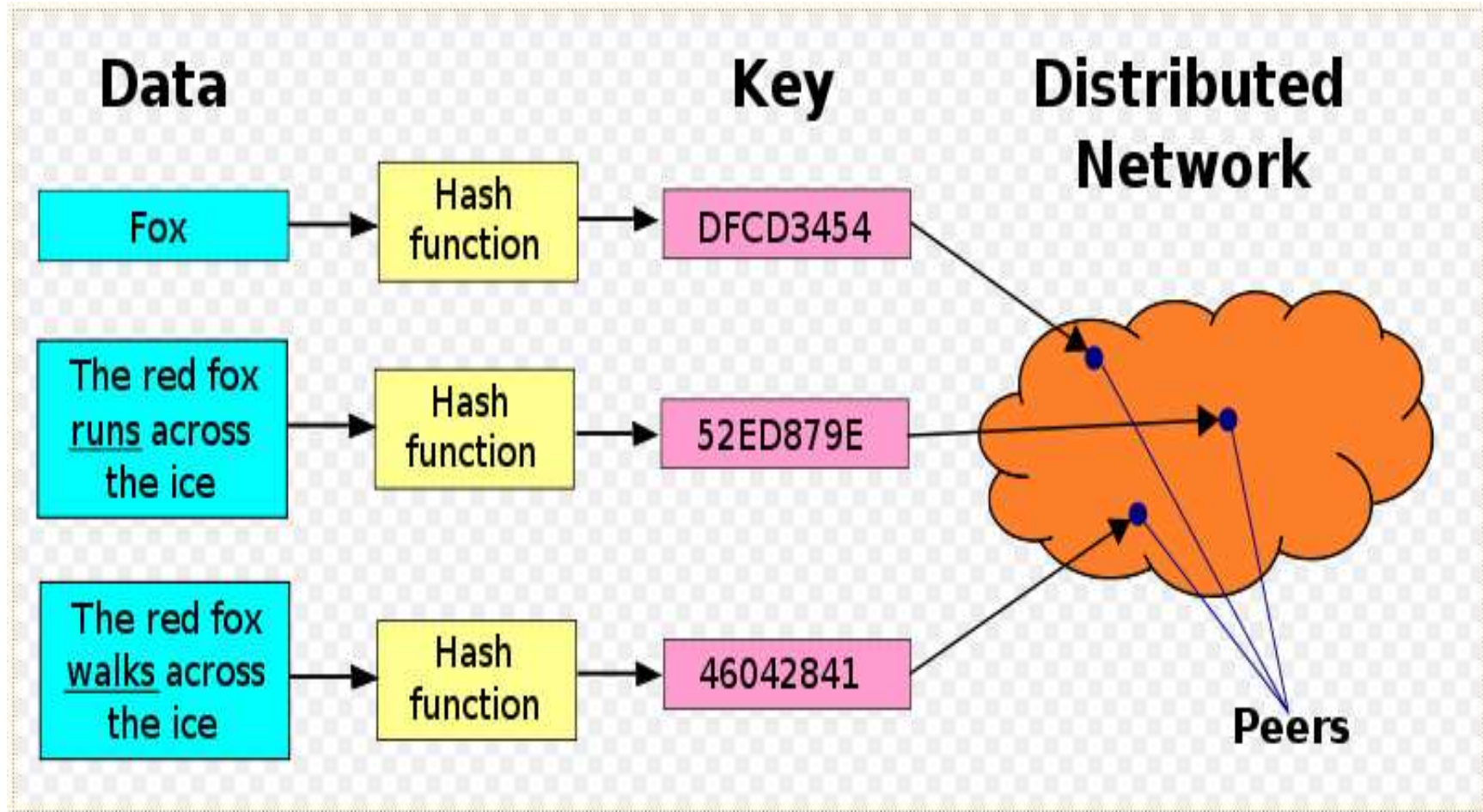
Peer-to-Peer



Peer-to-peer

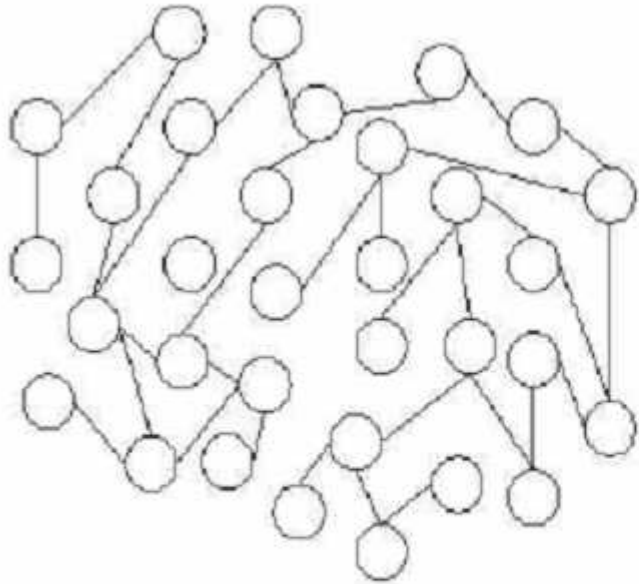
- Static structure
 - LAN
 - Example: Borgchat
- Dynamic structure
 - Structured
 - Distributed **hash table**
 - Unstructured
 - Random network
 - Scale free network
 - Example JXTA, Torrent

DHT on Peer-to-Peer

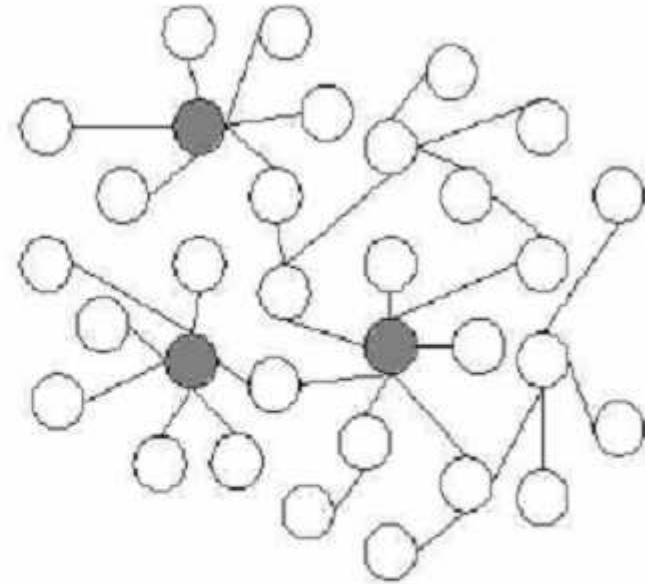


Unstructured peer-to-peer

- Data stored at random nodes

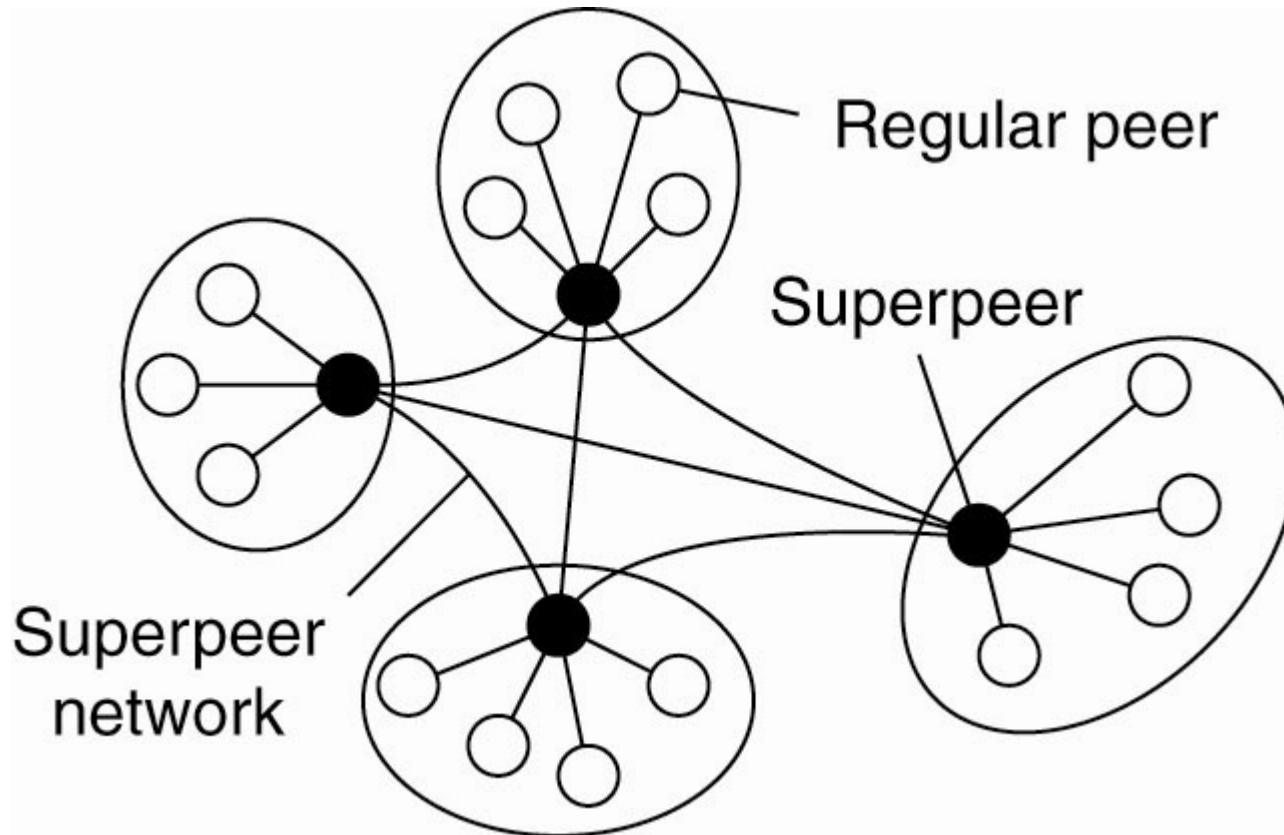


(a) Random network



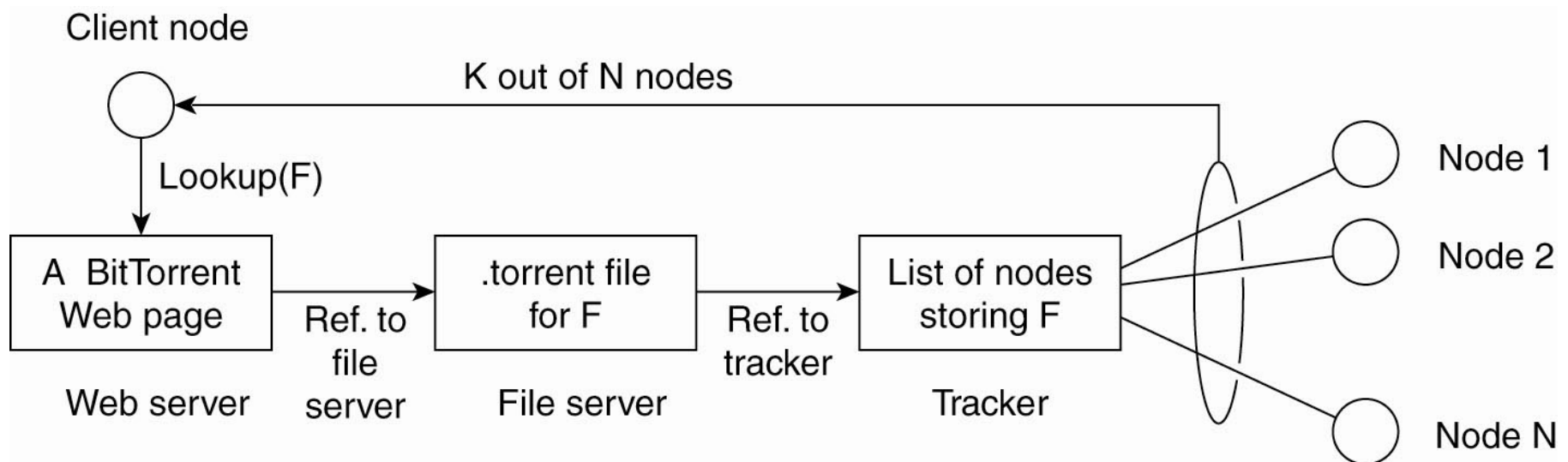
(b) Scale-free network

Superpeers



- Gabungan peer-to-peer & client server

Collaborative Distributed Systems - bitTorrent



- Download file .torrent dari website, yang berisi informasi file yg akan didownload (seed)
- Dicatat oleh Tracker, user akan bergabung dalam node seed dan peer

Fundamental Model SisTer

- Sistem Terdistribusi terdiri dari banyak proses yang berinteraksi dengan cara yang **komplek**.
- Interaksi dilakukan dengan mengirimkan **message** melalui komunikasi (yaitu aliran informasi), dan koordinasi antar proses (sinkronisasi dan urutan aktifitas)
- Perlu dipahami bahwa interaksi sangat dipengaruhi oleh tingkat reliabilitas dan performance jaringan komputer

Contoh :

- Banyak proses yang bekerja sama untuk menyediakan suatu layanan, contoh : **DNS** (replikasi dan partisi data pada Internet), **Sun NIS** (replikasi file password).
- Sekumpulan proses peer dapat bekerja sama dengan lainnya untuk mencapai suatu layanan, contoh : voice conferencing

Communication

- Processes must communicate
 - Synchronization
 - Data sharing
- Caranya:
 - Shared memory
 - Threads
 - Message passing

Faktor-faktor Komunikasi

- Latency, delay antara pengirim dan penerima message
 - Network access time
 - Waktu untuk pengiriman bit pertama dari Network Interface pengirim ke Network Interface Penerima
 - Waktu pemrosesan pada proses pengiriman dan penerimaan
- Throughput, jumlah paket yang terantar per satuan waktu
- Bandwidth, jumlah informasi (bit) yang terkirim per satuan waktu
- Delay jitter, variasi delay antar pesan bertipe sama yang berbeda

NEXT

Interprocess Communication