Model Sistem Terdistribusi

Budi Susanto
budsus@ukdw.ac.id
<->**</->**

dari Distributed System 3th, Colouris chap. 2

Pengantar

- **Model**
  - Menyediakan sebuah gambaran abstrak aspek yang relevan dengan sistem
- **Tujuan**
  - Menyediakan sebuah kerangka kerja untuk memahami per-masalahan dan pemecahannya
- **Model Arsitektural**
  - Hubungan dan interkoneksi seperti apa antara komponen-komponen dari sistem terdistribusi
- **Model Fundamental**
  - Karakteristik apa yang mempengaruhi *dependability* sistem terdistribusi?
  - *Dependability* : *correctness*, *reliability*, *security*

Kesulitan dan Ancaman

- Mode pemakaian
  - Variasi yang beragam terhadap karakteristik pemakaian
  - Contoh : berapa banyak halaman di kunjungi
- Masalah Internal
  - Masalah concurrency
  - Masalah clock
  - Mode kegagalan
- Lingkungan Sistem
  - Sistem terdistribusi harus mengakomodasi heterogenitas hardware, sistem operasi dan jaringan
  - Contoh : berapa banyak versi SO?
- Ancaman Eksternal
  - Serangan terhadap kesatuan data dan keamanannya

Model Arsitektur Sistem Terdistribusi

Model Arsitektur

- Arsitektur
  - Struktur komponen-komponen secara terpisah
- Tujuan
  - Menyakinkan bahwa struktur sistem memenuhi standar saat ini dan yang akan datang
- Model Arsitektur
  - Abstrak fungsi tiap komponen sistem terdistribusi
  - Penempatan komponen pada jaringan komputer
  - Hubungan antar komponen, yaitu peran fungsional per komponen dan pola komunikasi antar komponen

Klasifikasi Proses

- Proses Server
  - Menyediakan layanan dan menangani request
- Proses Client
  - Proses membuat/melakukan request
- Proses Peer
  - Proses yang saling bekerja sama dan berkomunikasi
- Klasifikasi proses tersebut mengidentifikasikan
  - tanggung jawab masing-masing proses
  - dan juga membantu untuk menaksir beban kerja
  - Serta menentukan pengaruh kegagalan dari tiap proses

Software Layer

- Arsitektur Software
  - Lapisan atau modul dalam sebuah komputer tunggal
  - Mendefinisikan layanan yang ditawarkan atau diminta antar proses pada satu atau berbeda komputer

Ide dasarnya

Membagi sistem yang komplek dalam struktur lapisan dan layanan

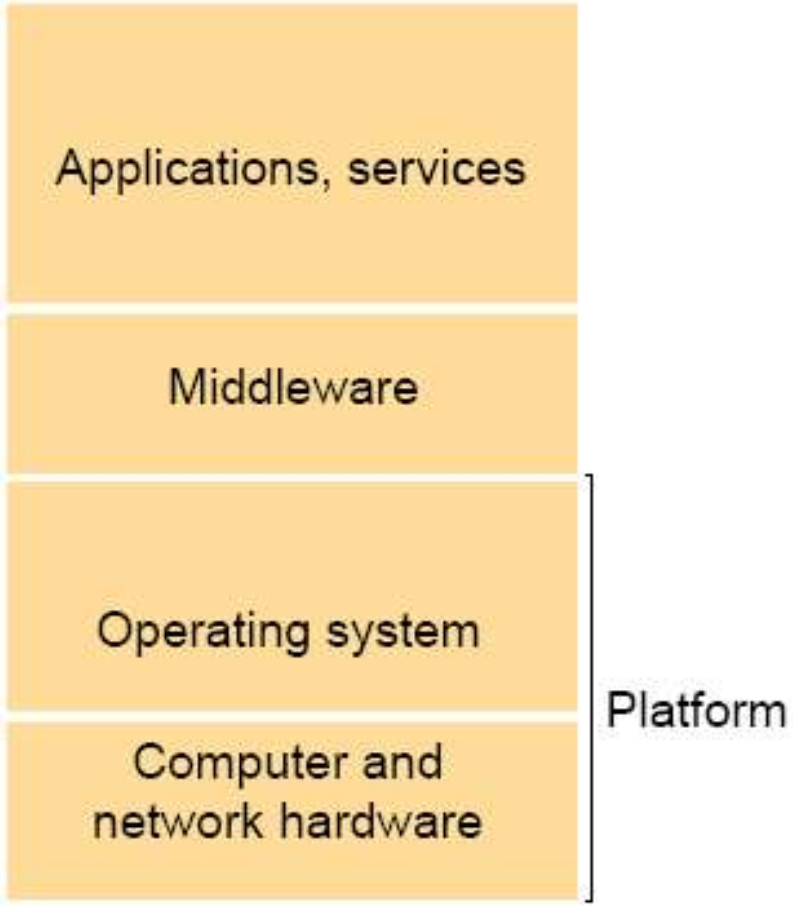
Antar layer mendefinisikan antar muka

Platform : Hardware dan SO

WindowsNT/Pentium Processor

Solaris/SPARC processor

Linux/Pentium Processor



Middleware

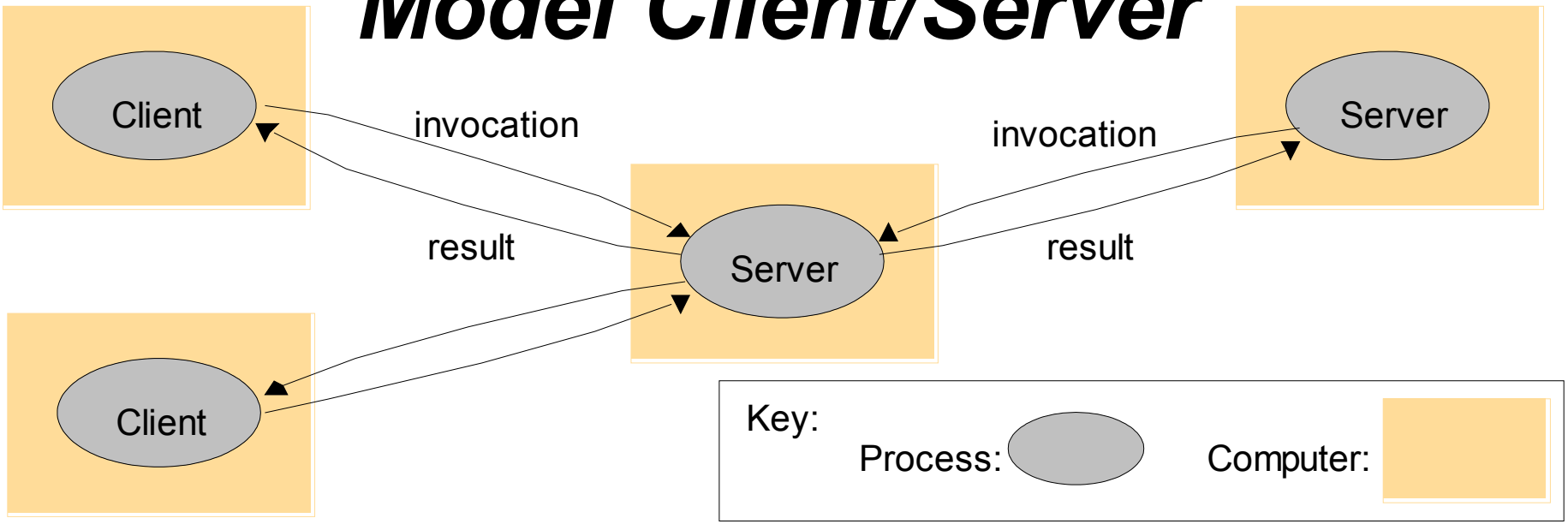
- Menyediakan transparensi terhadap keanekaragaman platform
- Proses dan objek pada sekumpulan mesin yang menerapkan protokol untuk aplikasi ter-distribusi

- Contoh :
  - CORBA (OMG)
  - DCOM (Microsoft)
  - ODP (ITU-T/ISO)
  - Java Remote Method Invocation (Sun)

Arsitektur Sistem

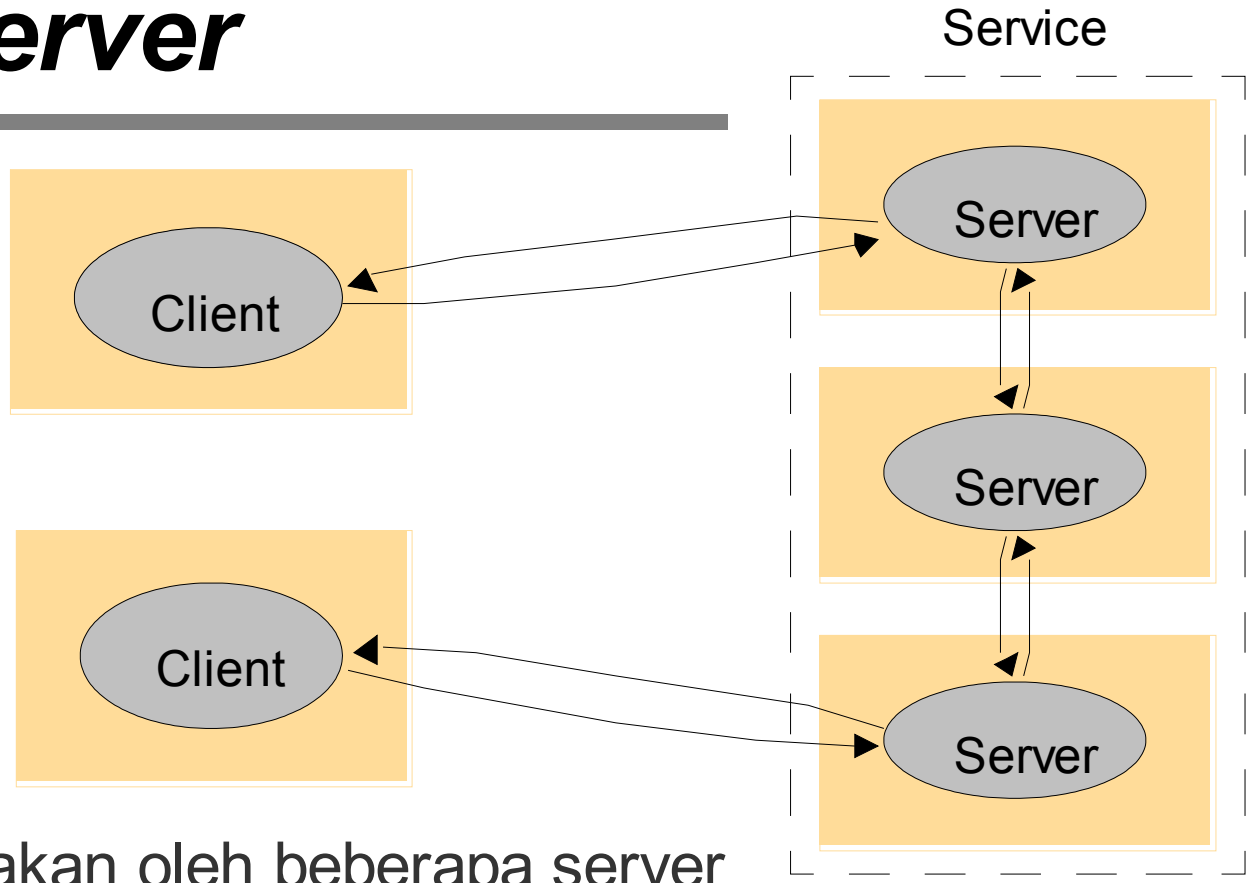
- Perancangan sistem terdistribusi dititikberatkan pada :
  - Pembagian tanggung jawab antara komponen sistem
  - Penempatan komponen pada komputer dalam jaringan
- Pengaruh dari perancangan
  - Unjuk kerja, Kehandalan dan Keamanan secara langsung tergantung pada pilihan yang ditentukan

Model Client/Server



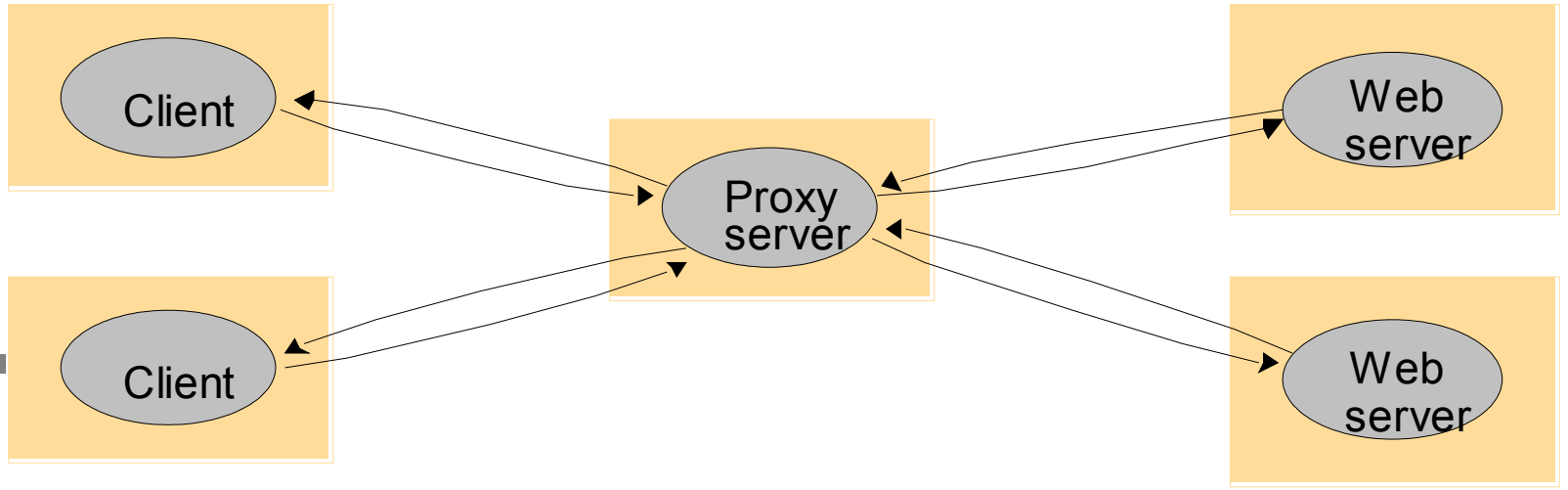
- **Client**: proses untuk mengakses data, menggunakan sesumber atau melakukan operasi pada komputer yang berbeda
- **Server**: proses yang mengatur data dan semua sesumber yang di share di antara server dan client, memungkinkan client mengakses sesumber dan melakukan komputasi
- **Interaction**: pasangan pesan pemanggilan (invocation) / hasil (result)
- Example
  - http server: client (browser) meminta dokumen, server mengirimkan dokumen yang diminta
- **Caching of services (proxy servers)**
  - caching terhadap halaman web yang sering digunakan
- **Peer processes (not client-server)**
  - proses-proses yang secara fungsional identik

Multiple Server



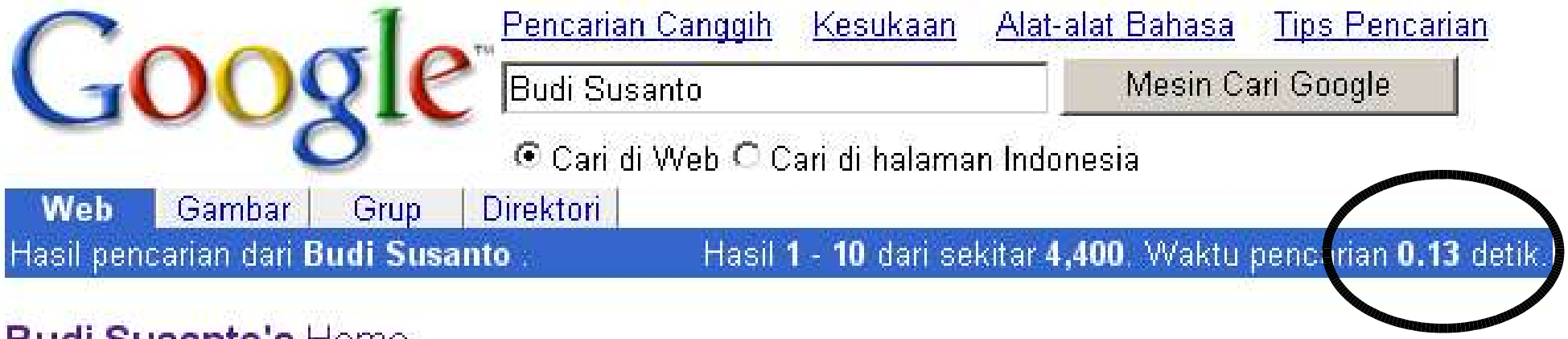
- Service disediakan oleh beberapa server
- Contoh : sebagian besar layanan web komersial diterapkan melalui server fisik yang berbeda
- Motivasi :
  - Unjuk kerja (contoh : cnn.com, download server, dll)
  - kehandalan
- Server menggunakan replikasi atau database terdistribusi

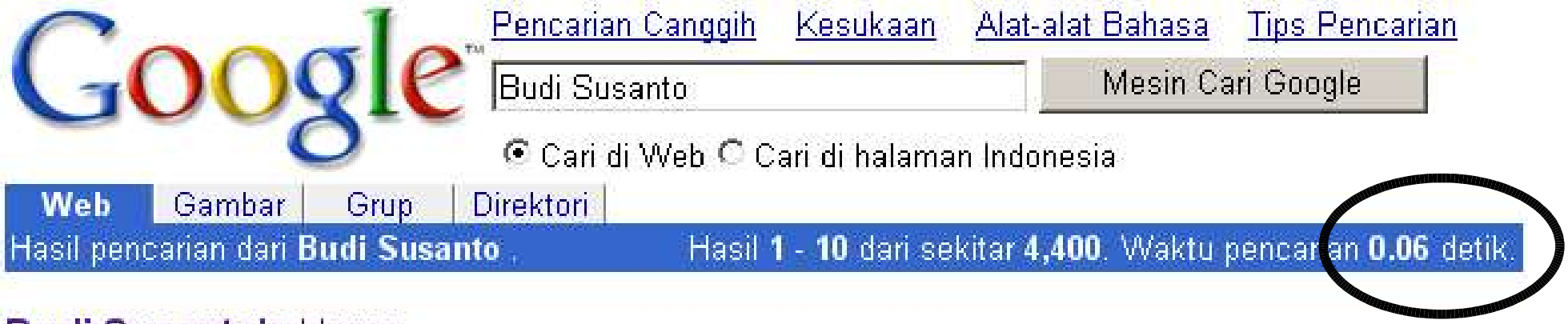
Proxy Server

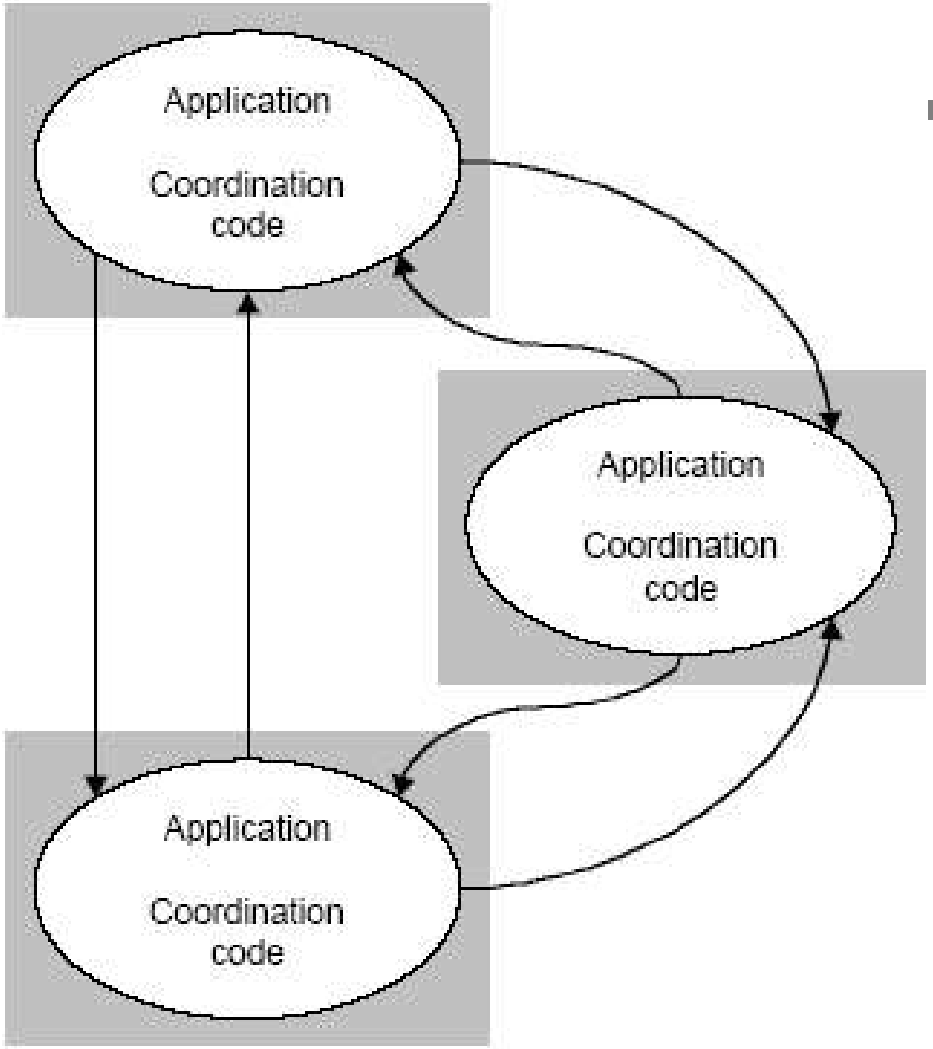


- Server dengan duplikasi informasi yang melayani sebagai proxy
- Caching :
  - Penyimpan lokal untuk item-item yang sering digu-nakan
  - Meningkatkan unjuk kerja
  - Mengurangi beban pada server
- Biasanya digunakan pada search engine

Contoh Cache pada Google



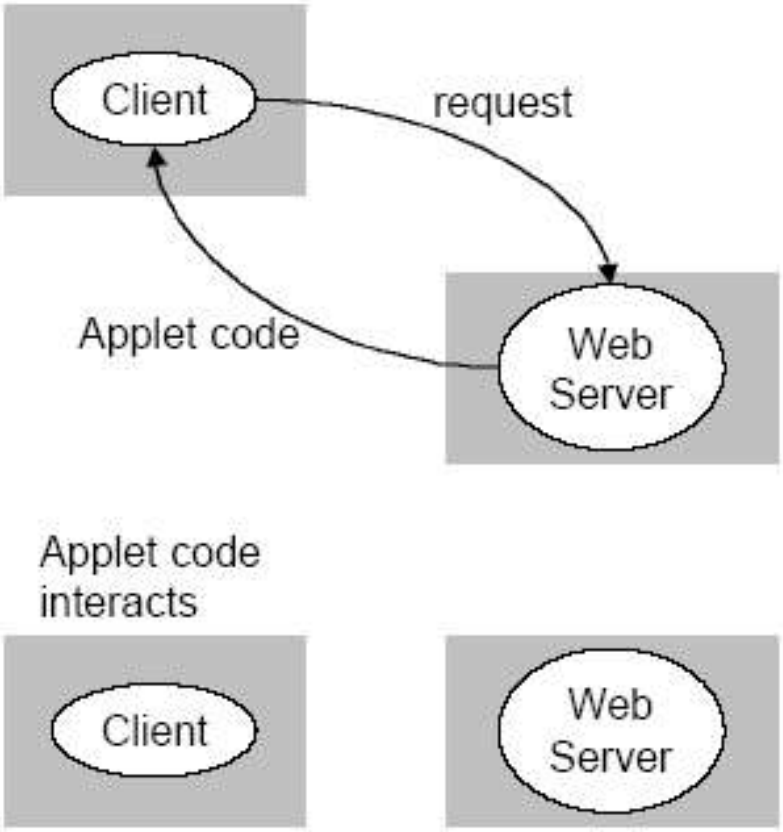




Peer Process

- Peer processes
  – menjaga konsistensi sesumber
  – sinkronisasi aksi
- contoh: Whiteboard
  – menggunakan central server
  – menggunakan peer processes

Mobile Code



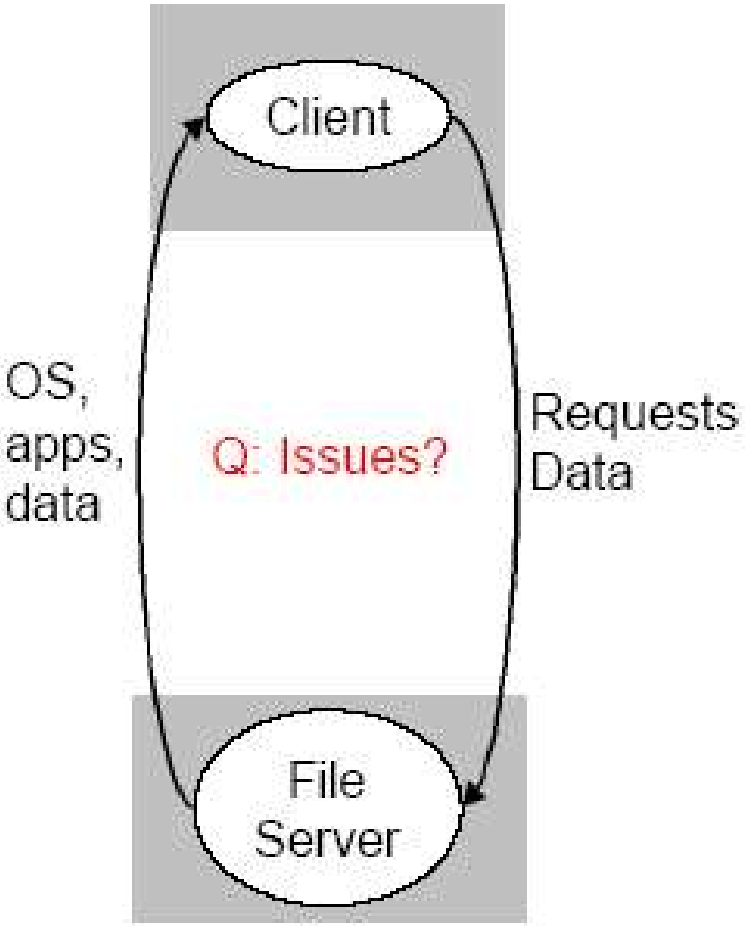
- Mobile code
  - kode yang berpindah dan di-jalankan pada site yang berbeda
- Contoh : applet
- Model
  - pengendali client
  - push model
- Q: masalah keamanan?

Mobile Agent

- Mobile Agent adalah sebuah program yang berpindah (termasuk data dan kode) dari satu komputer ke lainnya dalam jaringan
- Biasanya melakukan suatu pekerjaan otomatis tertentu
- Beberapa masalah :
  - Authentication
  - Permission dan keamanan
- Alternatif
  - Mengambil informasi melalui remote invocation

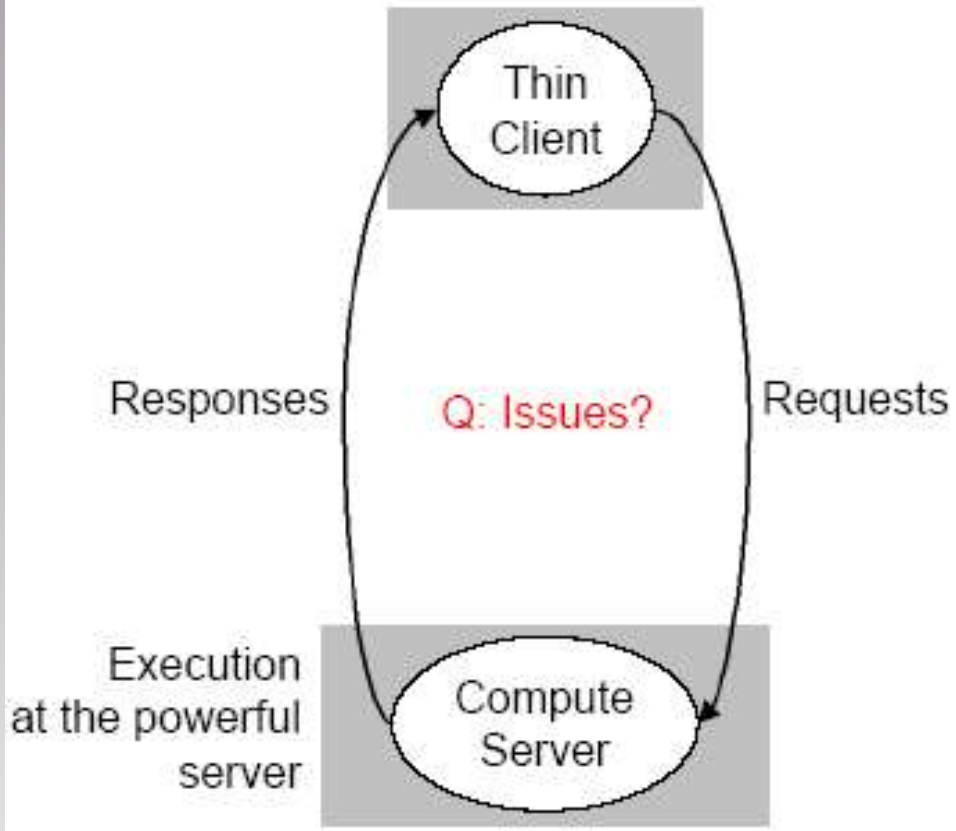
- Contoh :
  - Digunakan untuk install dan memelihara software pada komputer dalam suatu organisasi
  - Membandingkan harga produk dari beberapa vendor

Network Computer



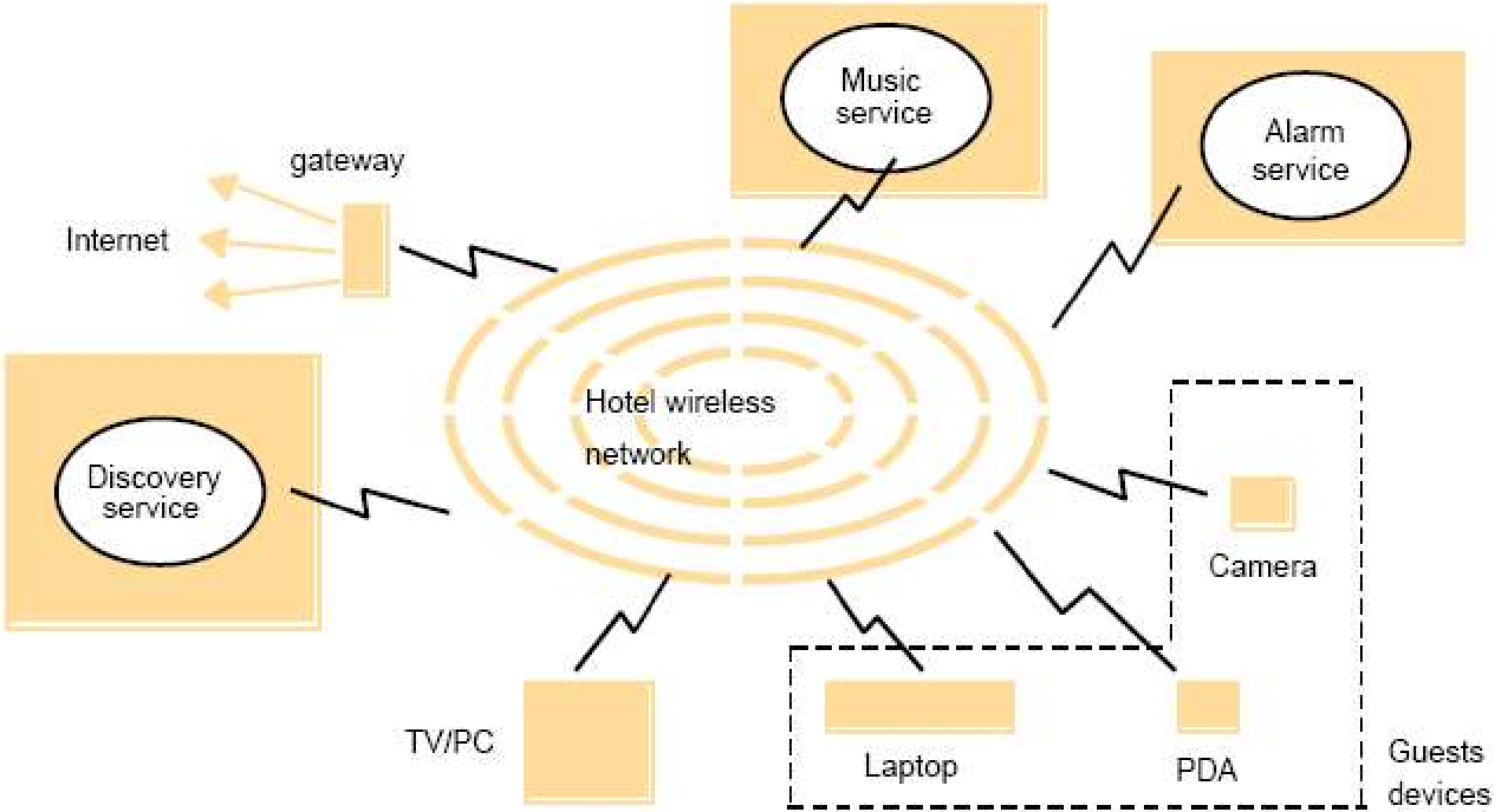
- Masalah
  - Pemeliharaan PC sangat tinggi
- Solusi
  - Mengurangi variasi pada level lokal
- Network Computer
  - OS dan applikasi dari file server
  - Remote file service
  - Disk lokal (jika ada) digu-nakan sebagai cache

Thin Client



- Masalah
  - Membuat PC powerfull sangatlah mahal
- Solusi
  - Menyediakan server komputasi yang powerfull
- Thin Client
  - Lapisan software mendukung aplikasi lokal dengan remote executing
- Contoh :
  - X11 Window (www.xfree86.org)
  - VNC (www.realvnc.com)
  - Citrix WinFrame (www.citrix.com)

Mobile Devices



Spontaneous Networking

- Bentuk distribusi yang menggabungkan peralatan mobile dan peralatan lain dalam suatu jaringan disebut sebagai spontaneous networking
- Meliputi juga aplikasi yang dapat digunakan untuk koneksi antara mobile ataupun non-mobile device ke jaringan.
- **Kemampuan/karakteristik Spontaneous Networking** :
  - Kemudahan koneksi ke jaringan lokal
  - Kemudahan menggabungkan dengan layanan lokal
    - Discovery Service
  - Konektifitas terbatas
  - Keamanan dan privacy
- Antar muka discovery service
  - **Registration service**
    - Menerima penerimaan registrasi dari server, menyimpan properti layanan yang tersedia dalam database
  - **Lookup service**
    - Mencocokan permintaan layanan dengan layanan yang tersedia

Antarmuka Client/Server

- Kumpulan fungsi yang tersedia untuk pemang-gilan (*invocation*) dari sebuah proses (server atau peer) dinyatakn dengan satu atau lebih *definisi antarmuka*.
- Pada C++ dan Java, sebuah antarmuka dapat disusun dengan model berbasis pada objek.
- Contoh : RMI (java), CORBA(semua bahasa OOP)

Merancang Kebutuhan untuk Sistem Terdistribusi

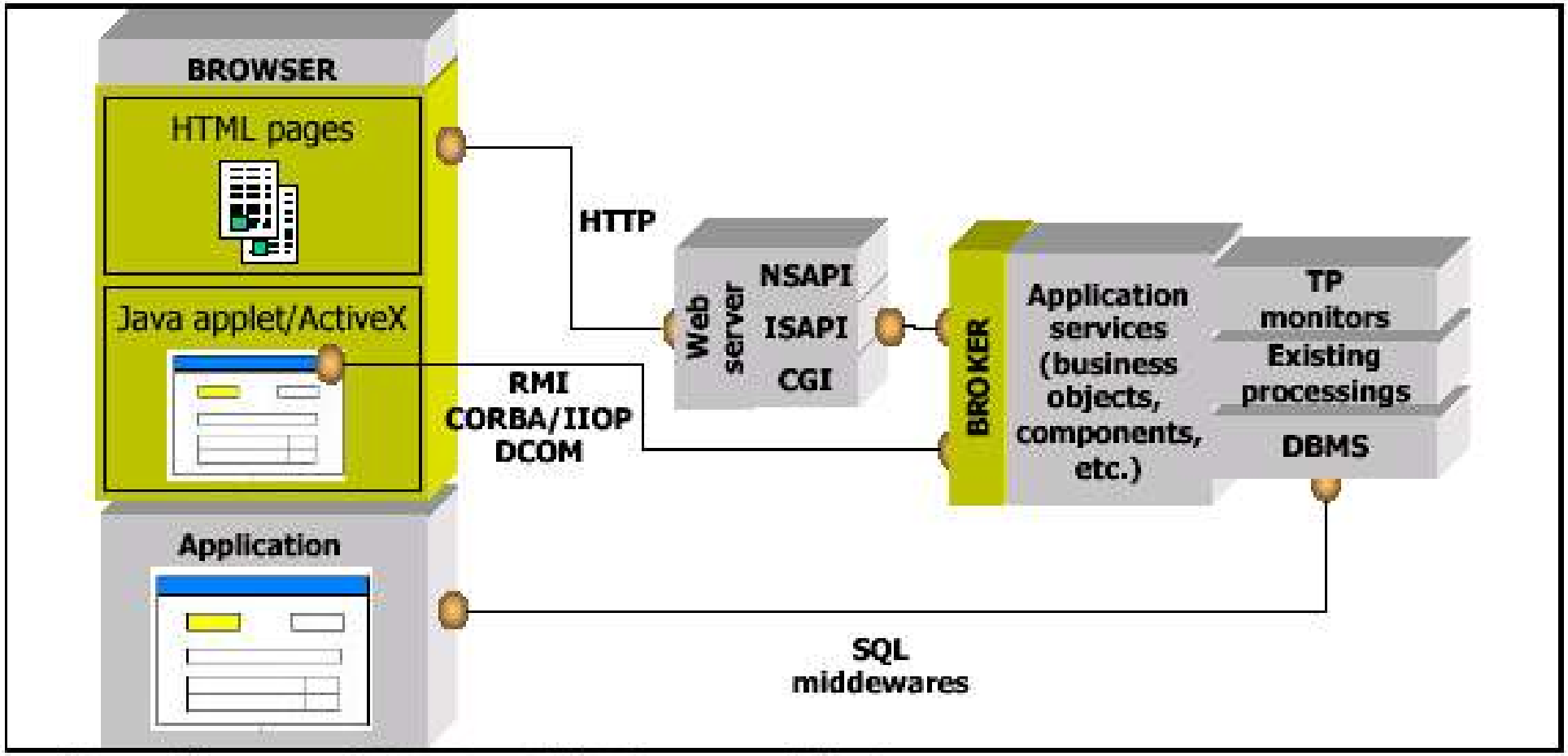
- **Masalah Kinerja (Performance)**
  - Responsiveness
  - Throughput
  - Balancing computational load
- **Quality of Service**

  pengukuran seberapa tingkat pemakaian fungsi, yaitu
  - Reliability
  - Security
  - Performance
  - *Adaptability*, kemampuan untuk menyesuaikan dengan perubahan konfigurasi sistem

Merancang Kebutuhan untuk Sistem Terdistribusi (lanjut)

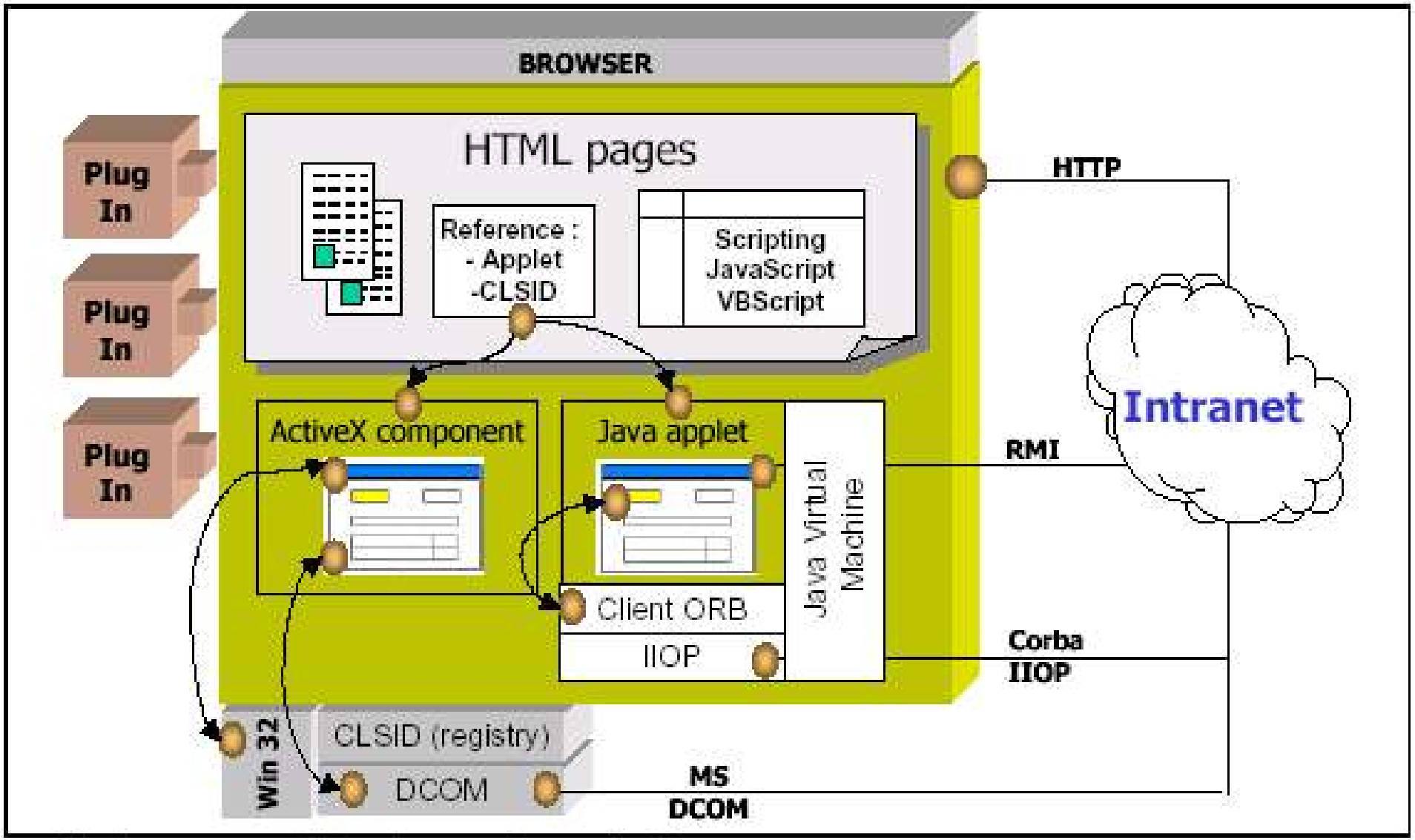
- **Pemakaian caching dan replikasi**
  - Local copy informasi
  - Cache consistency
  - Web caching protocol
  - Replikasi : beberapa copy dari service
- **Masalah Dependability**
  - Fault tolerance :
  - Keamanan

Diagram Arsitektur Intranet dan CS



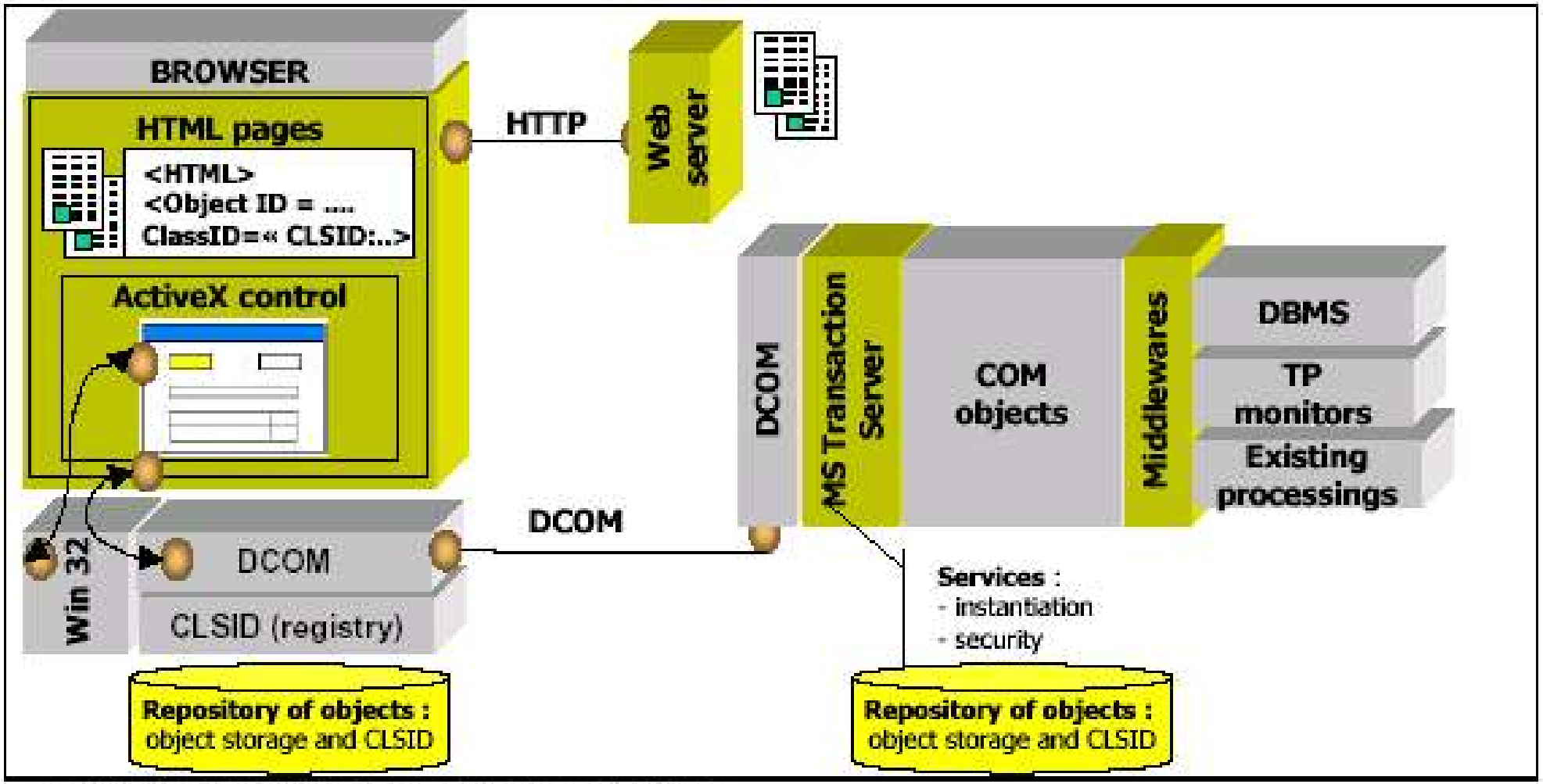
General diagram of intranet and client-server architectures

Browser



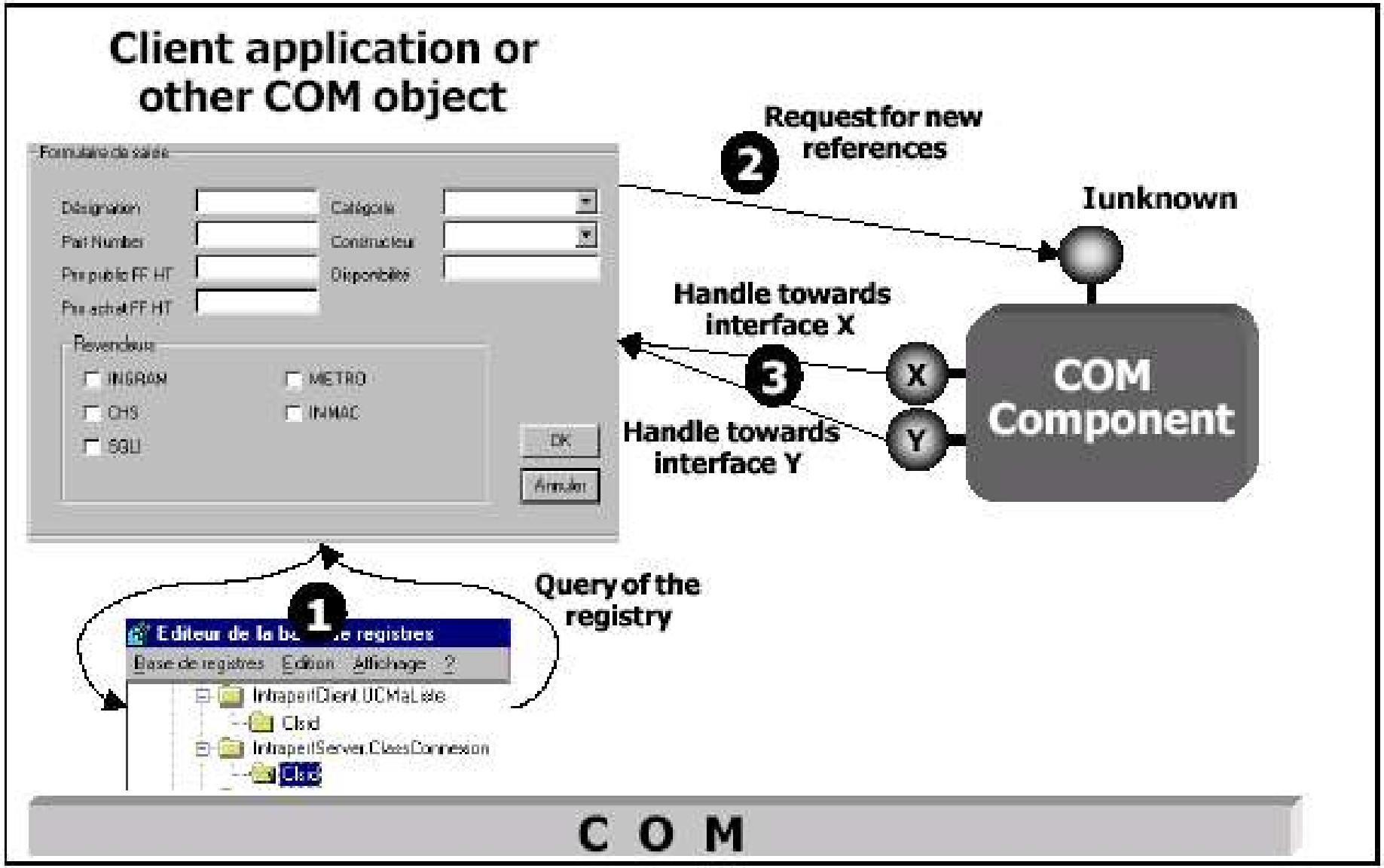
The browser, a platform for running applications.

ActiveX - DCOM



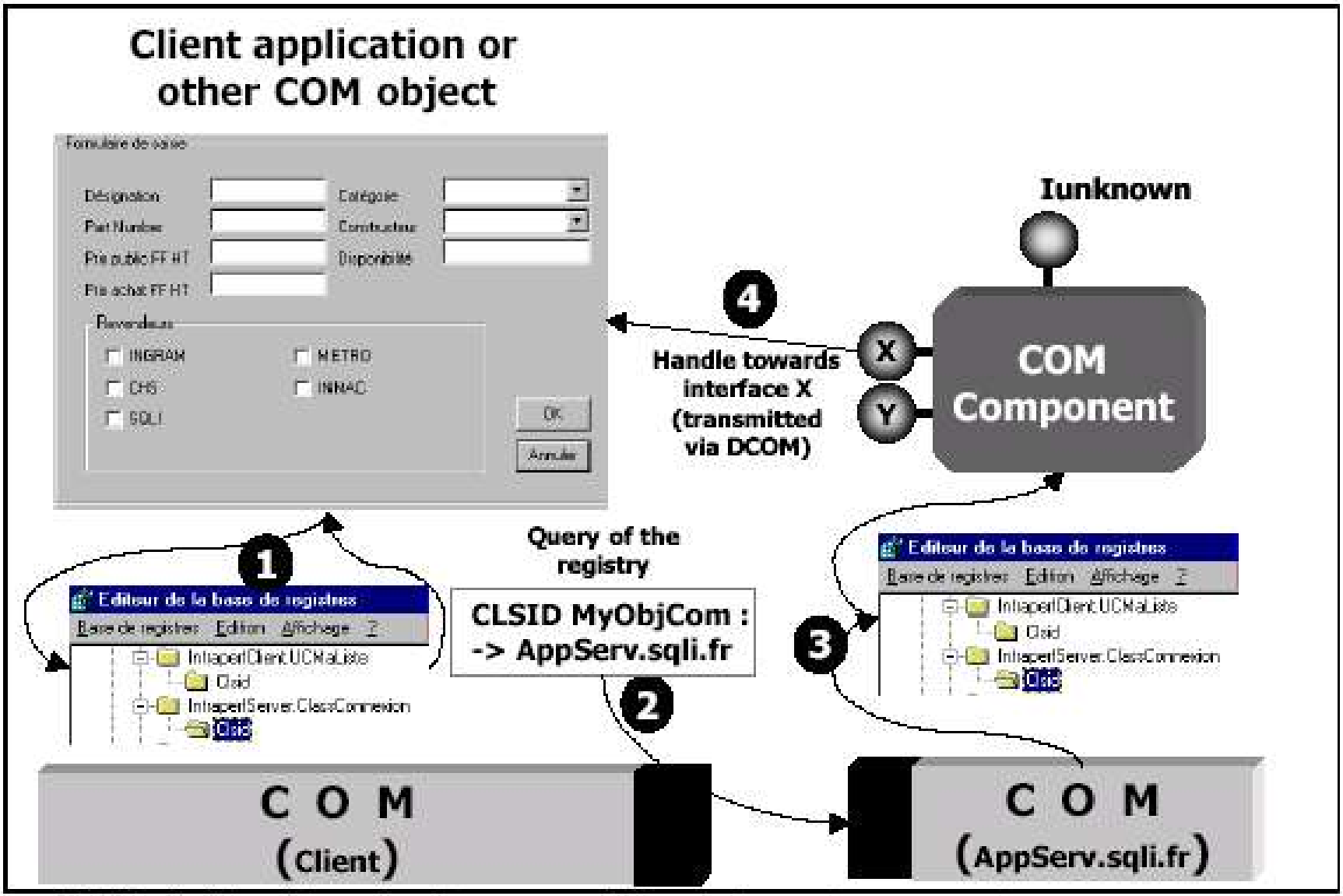
ActiveX and DCOM communication principles.

Prinsip COM



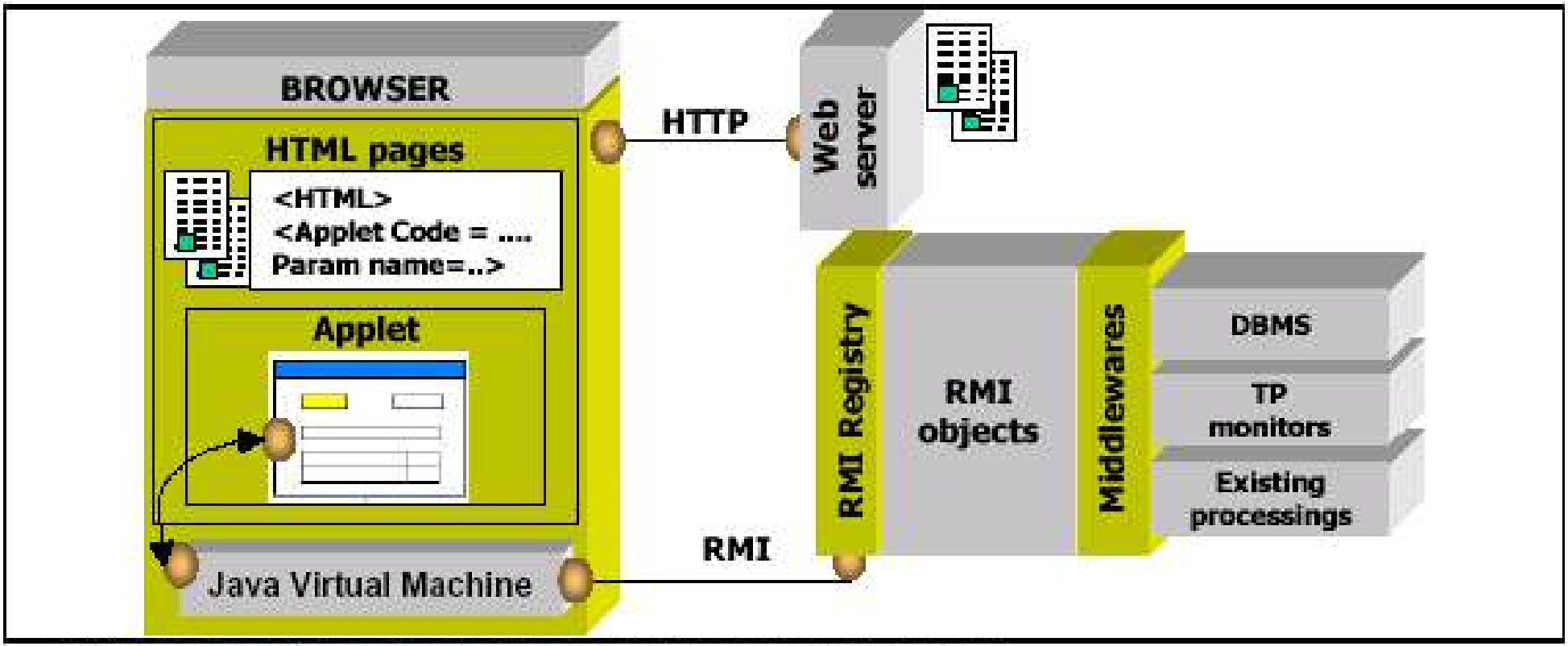
COM principles: interaction between application and component.

Prinsip DCOM

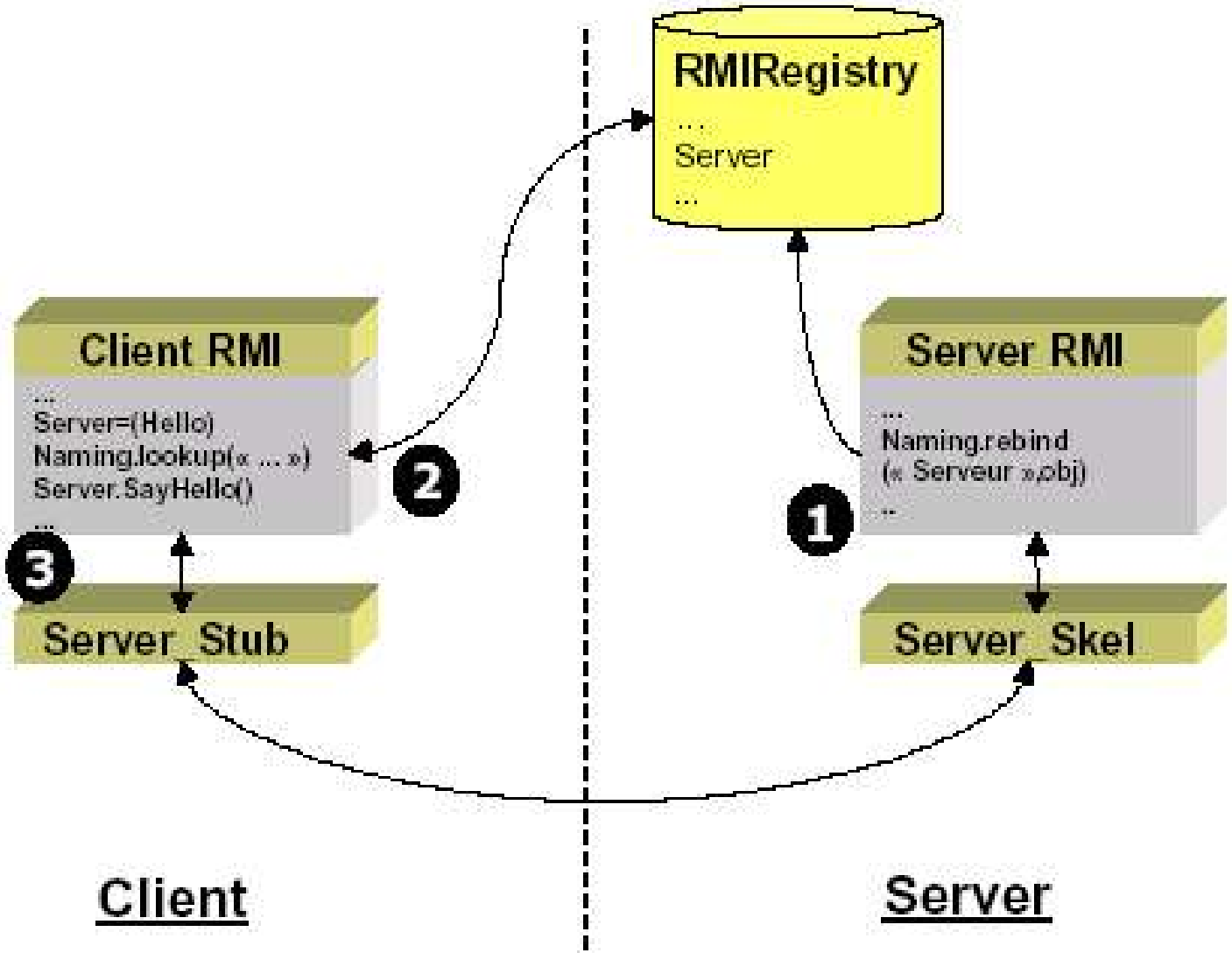


DCOM principles: interaction between application and remote component.

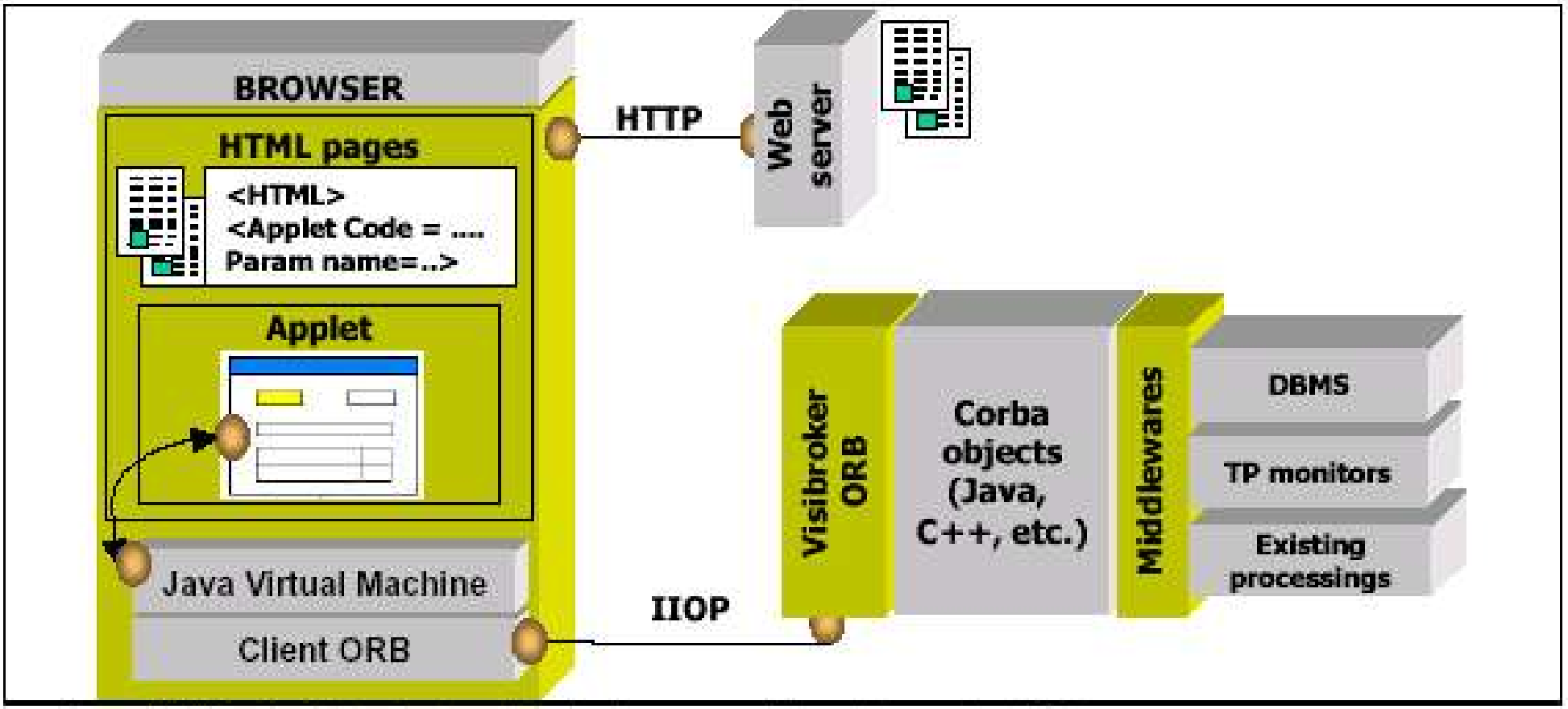
Arsitektur RMI



RMI principles: interaction between applet and remote objects.

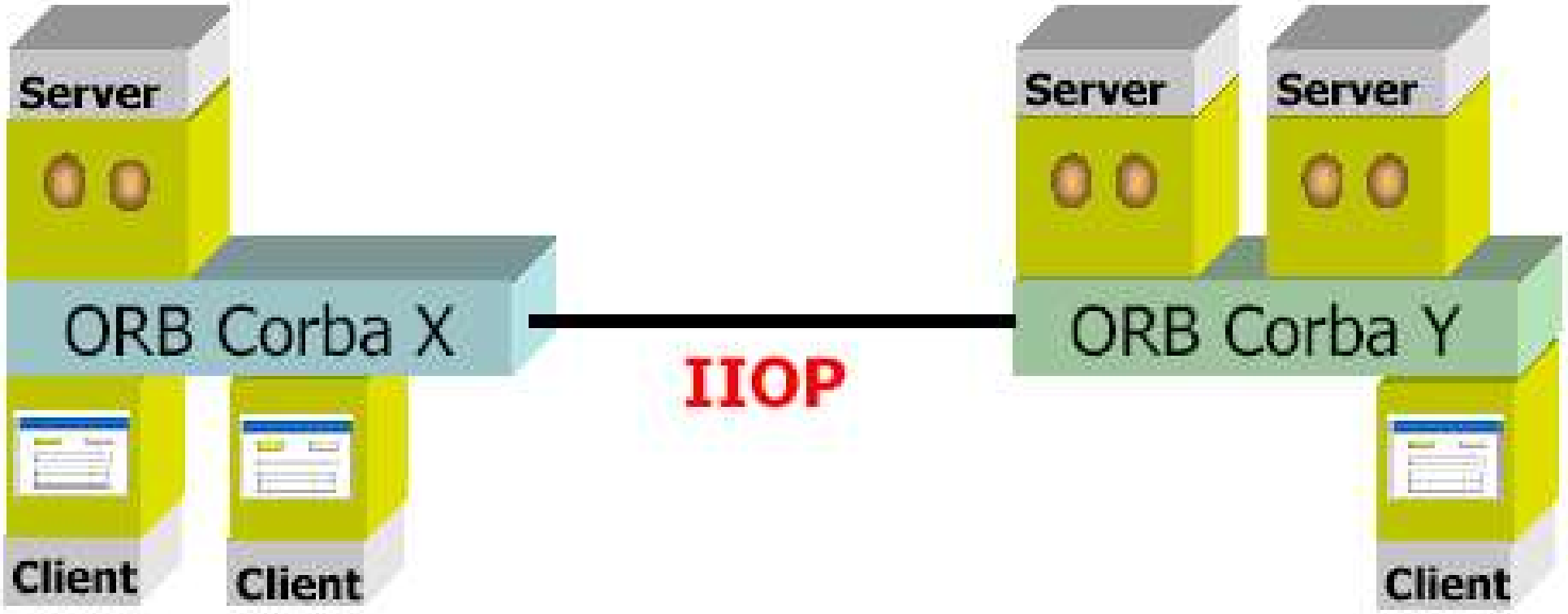

RMIRegistry
...
Server
...
Client RMI
...
Server=(Hello)
Naming.lookup(« ... »)
Server.SayHello()
...
Server RMI
...
Naming.rebind
(« Serveur »,obj)
..
1
2
3
Server_Stub
Server_Skel
Client
Server

Java CORBA

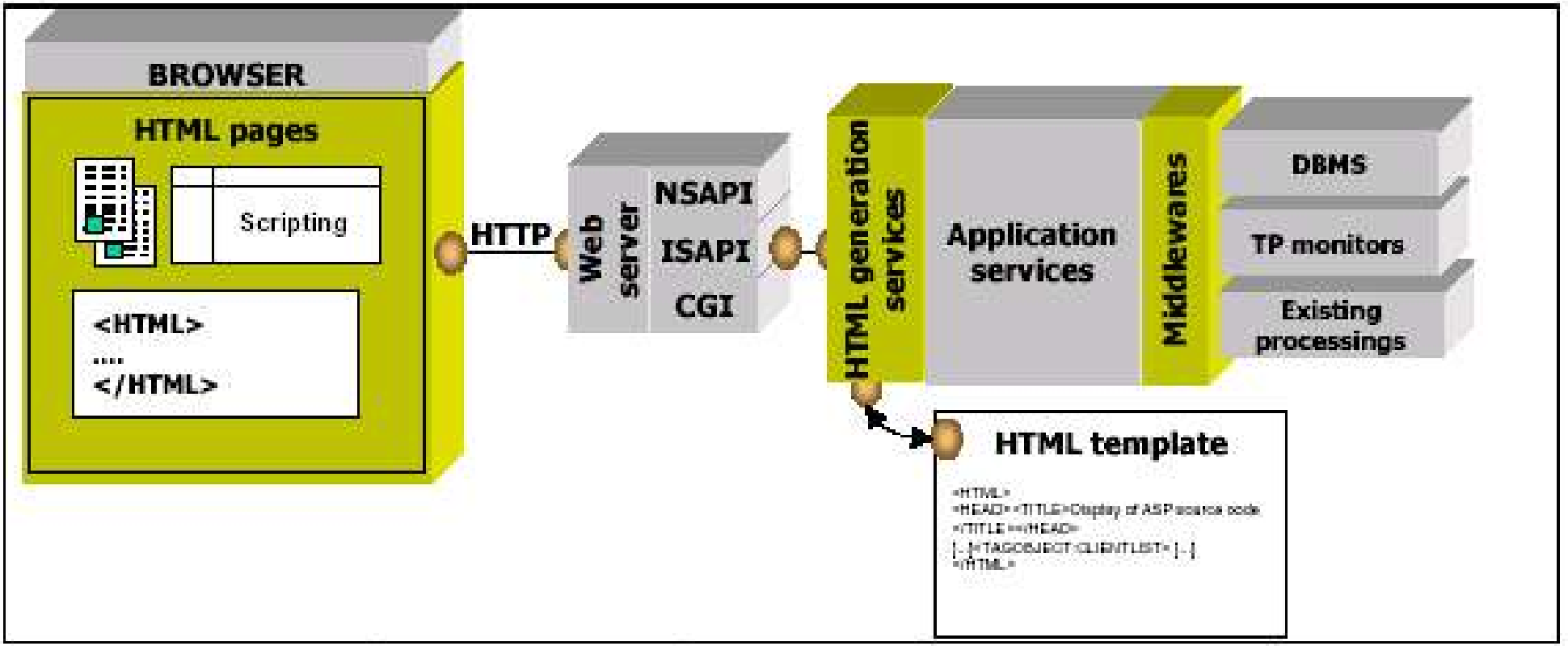


Java - IIOP principles: interaction between applet and remote objects.

Arsitektur CORBA



HTML - HTTP



HTML - HTTP principles: communication between component.