

Modelos de Programação Distribuída (parte 2)

CAPÍTULO III

Modelos de Programação Distribuída

Modelos de comunicação por mensagens

- Exemplo: Comunicação por mensagens através de Sockets (em Java)

Modelos Arquitecturais

- Cliente / Servidor
- Múltiplos Servidores
- Proxies
- Peer processes

Modelos de Programação Distribuída

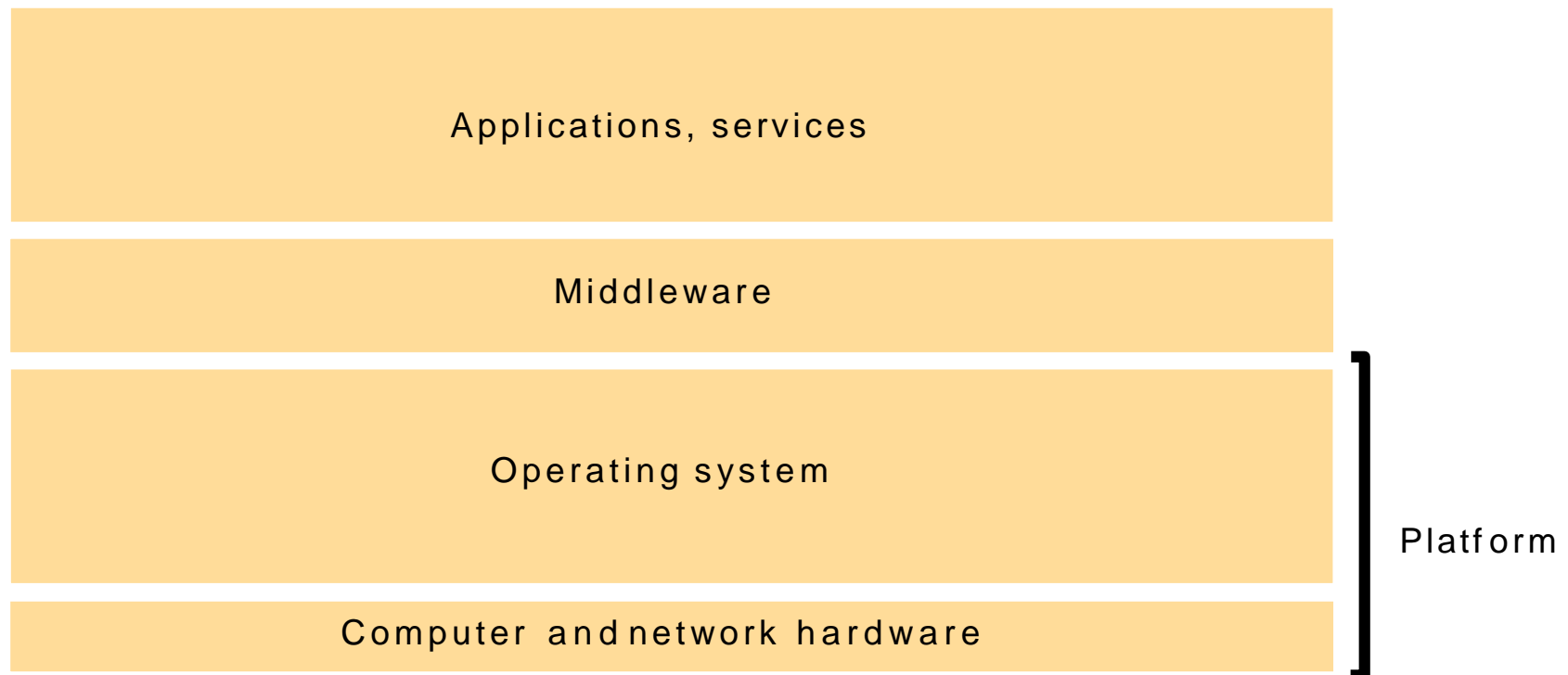
Modelos arquiteturais

- Um modelo arquitetural (ou a arquitetura) de um sistema distribuído é a estrutura do sistema em termos de localização das suas diferentes partes, do papel que cada parte desempenha e como se inter-relacionam.
- A arquitetura tem implicações no desempenho, fiabilidade e segurança do sistema.

Modelos de Programação Distribuída

Modelos arquiteturais

- Camadas de um sistema distribuído:



Modelos de Programação Distribuída

Modelos arquiteturais

- Plataformas:
 - Intel PII/Windows
 - Intel x86/Linux
 - Power PC/Solaris
 - ...
- Middleware:
 - Camada de software que tem o objectivo de mascarar a heterogeneidade de um sistema distribuído fornecendo um modelo de programação uniforme.
 - Sun RPC
 - Java RMI
 - Corba
 - Microsoft .NET
 - ...

Modelos de Programação Distribuída

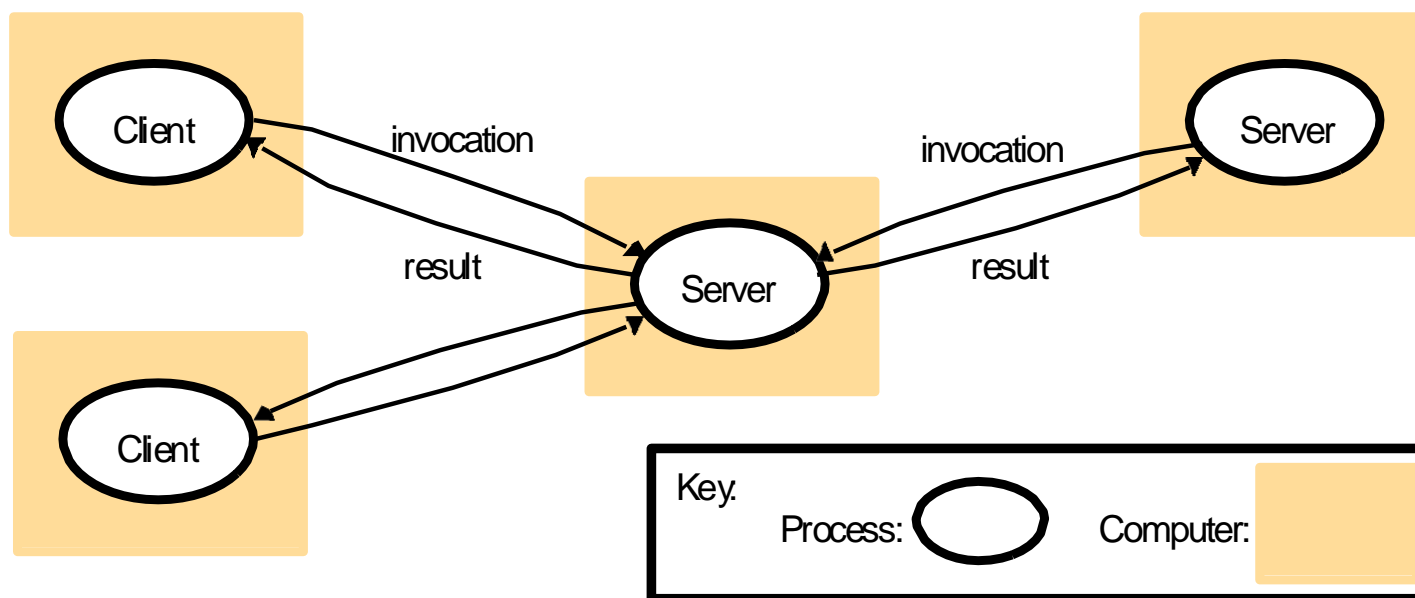
Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Modelo independente do middleware utilizado
 - Servidor (back-end):
 - processo passivo que quando contactado por um cliente envia a resposta.
 - Cliente (front-end):
 - contacta o servidor com o objetivo de utilizar um serviço; envia um pedido (request/invocation) e fica à espera da resposta (reply/result).
 - Cliente e Servidor são papéis que podem ser desempenhados.
 - Uma entidade pode simultaneamente ser cliente e servidor. Um processo para responder a um pedido, pode ter que recorrer a outro serviço, sendo cliente deste.
 - Ex.lo: um motor de pesquisa que usa “web crawlers” para pesquisar servidores web

Modelos de Programação Distribuída

Modelos arquiteturais

- Modelo Cliente – Servidor
 - A máquina que suporta o processo servidor precisa de ter recursos mais poderosos de forma a suportar centenas de pedidos num curto intervalo de tempo.



Modelos de Programação Distribuída

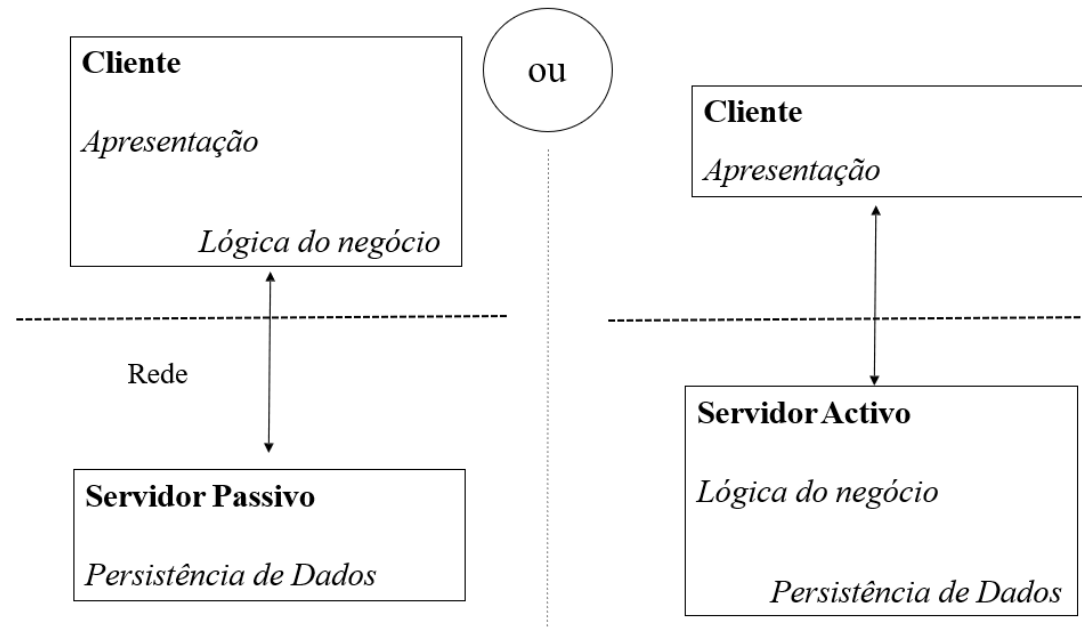
Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Num sistema de informação típico, existem três classes de funcionalidades:
 - Camada de apresentação
 - Parte da aplicação responsável pela interface com o utilizador
 - Camada de lógica de negócio
 - Regras de negócio que controlam o comportamento da aplicação
 - Camada de persistência de dados
 - Parte que assegura o armazenamento e integridade dos dados
 - A primeira arquitetura cliente/servidor a ser desenvolvida foi a arquitetura de duas camadas (2-tiers)

Modelos de Programação Distribuída

Modelos arquiteturais

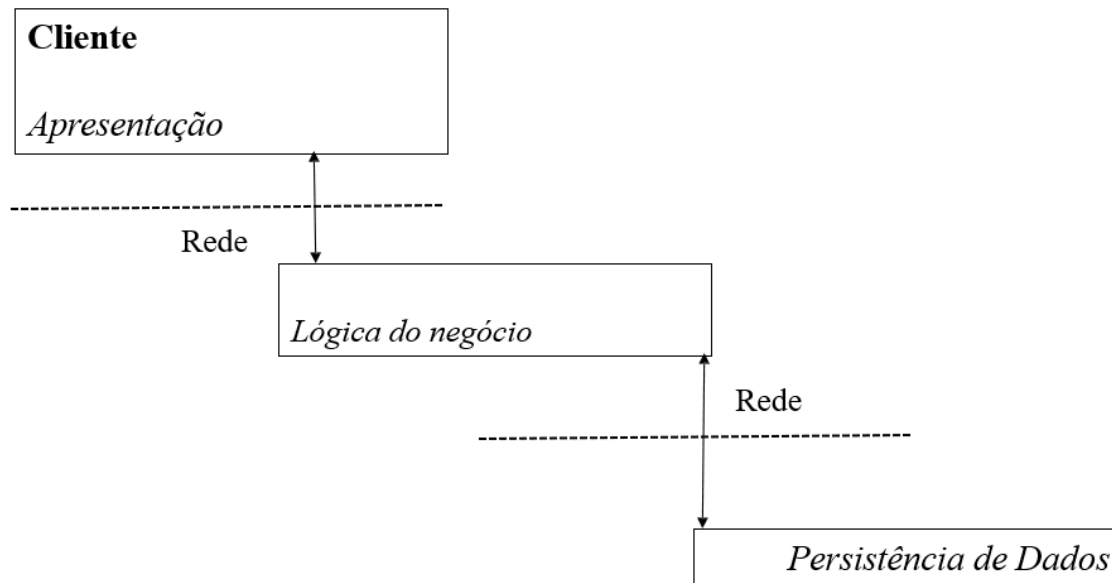
- Modelo (arquitetura) Cliente – Servidor
 - Modelo em duas camadas (2-tiers)



Modelos de Programação Distribuída

Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Modelo em três camadas (3-tiers)



Modelos de Programação Distribuída

Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Modelo em três camadas (3-tiers)**

Web browser

HTML, ...

Servidor Web

PHP, Java, ...

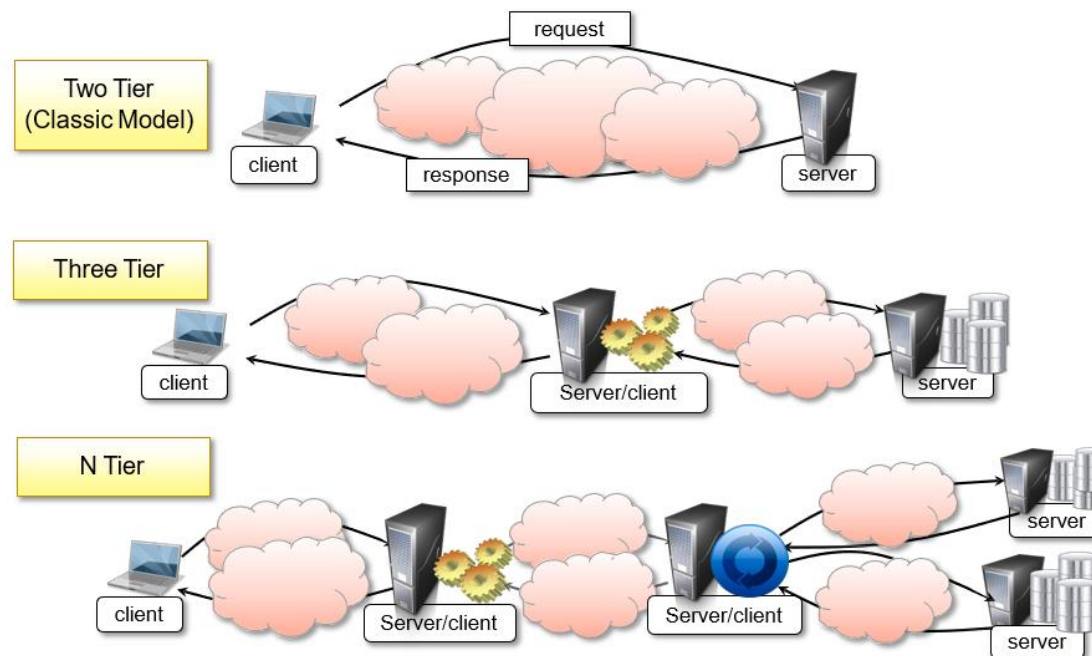
Base de Bados

MySQL, SQLServer, Oracle, ...

Modelos de Programação Distribuída

Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor



Modelos de Programação Distribuída

Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Modelo mais usado na prática
 - Modelo de interação simples
 - Segurança concentrada no servidor
 - Servidor é um ponto de falha único
 - Não é escalável para além de certos limites

Modelos de Programação Distribuída

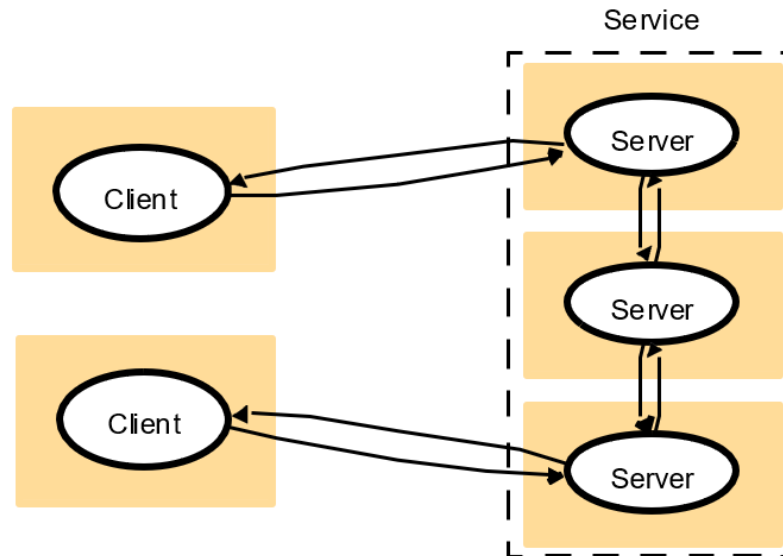
Modelos arquiteturais

- Modelo (arquitetura) Cliente – Servidor
 - Exemplos de grandes Sistemas Distribuídos com arquitetura cliente servidor:
 - Google, GoogleMaps, YouTube, Facebook, etc.
 - Variantes podem colmatar alguns problemas:
 - Particionamento
 - Replicação
 - Hierarquia
 - Etc ...
- Múltiplos Servidores
 - Um serviço pode ser implementado por vários processos servidores localizados em diferentes computadores.

Modelos de Programação Distribuída

Modelos arquiteturais

- Múltiplos Servidores
 - Entidades que fornecem o serviço estão distribuídas por diferentes máquinas
 - ou
 - Cópias replicadas pelas diferentes máquinas: > disponibilidade e > tolerância a falhas



Modelos de Programação Distribuída

Modelos arquiteturais

- **Cliente / Servidor replicado**
 - Existem vários servidores, capazes de responder aos mesmos pedidos
 - Vantagens:
 - Permite distribuir a carga, melhorando o desempenho
 - Não existe um ponto de falha único
 - Principal problema:
 - Manter estado do servidor coerente em todas as réplicas

Modelos de Programação Distribuída

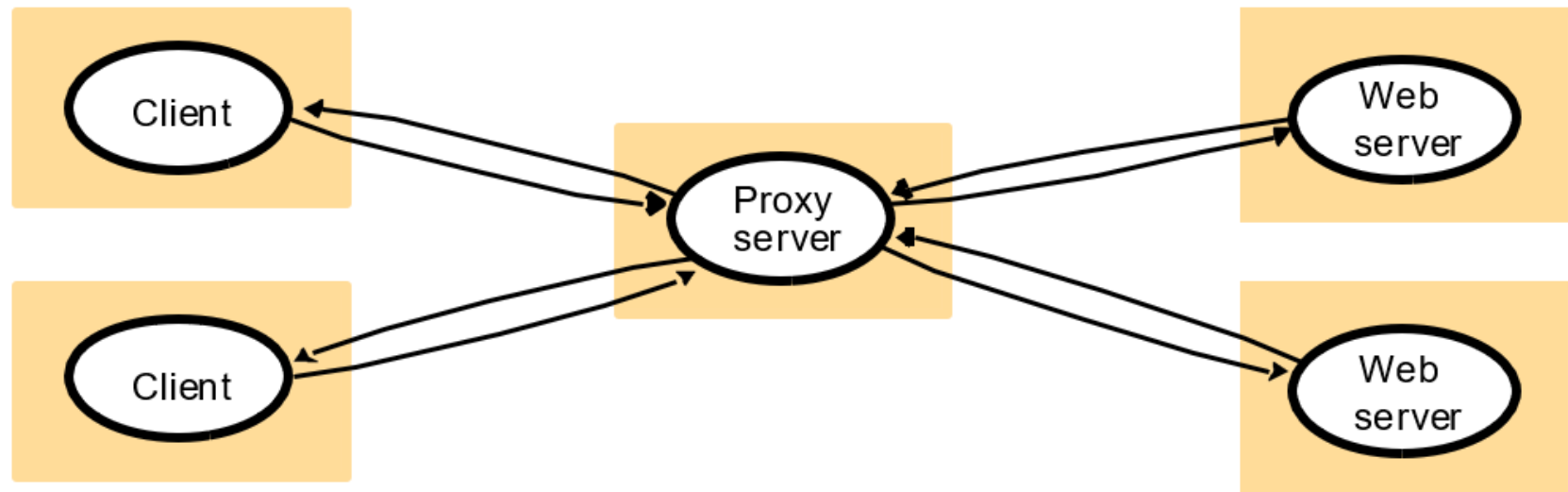
Modelos arquiteturais

- Cliente / Servidor particionado
 - Existem vários servidores com a mesma interface, cada um capaz de responder a uma parte dos pedidos (ex. DNS)
 - Servidor redirige o cliente para outro servidor (iterativo)
 - Servidor invoca o pedido noutro servidor (recursivo)
 - Vantagens → Escalabilidade
 - Permite distribuir a carga, melhorando o desempenho
 - Não existe um ponto de falha único
 - Problemas
 - Falha de um servidor impede acesso aos dados presentes nesse servidor

Modelos de Programação Distribuída

Modelos arquiteturais

- Proxy servers and Caches
 - Uma “cache” permite o armazenamento, numa localização mais próxima, de dados/objectos recentemente usados



Modelos de Programação Distribuída

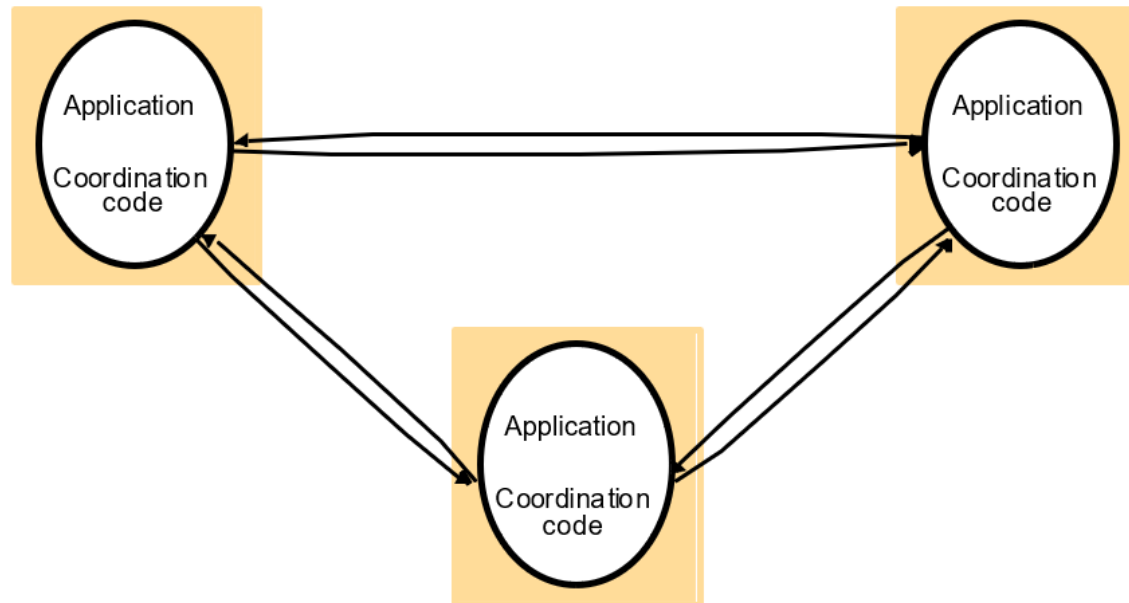
Modelos arquiteturais

- Proxy servers and Caches
 - Quando um cliente necessita de um objecto, o serviço de “caching” verifica se possui uma cópia actualizada do objecto, em caso afirmativo fornece essa cópia
 - Uma “cache” pode estar localizada no cliente ou em servidores “proxy” que são partilhados por vários clientes
 - Objectivo: aumentar a disponibilidade e a performance do serviço

Modelos de Programação Distribuída

Modelos arquiteturais

- Peer processes
 - Todos os processos desempenham papéis similares. Cada processo é responsável pela consistência dos seus dados (recursos) e pela sincronização das várias operações.



Modelos de Programação Distribuída

Modelos arquiteturais

- Peer processes
 - Cada processo pode assumir (simultaneamente ou alternadamente) o papel de cliente e servidor do mesmo serviço
 - Paradigma de distribuição em que os serviços são suportados diretamente pelos seus clientes / utilizadores, sem recurso a uma infra-estrutura criada e mantida explicitamente para esse fim.
 - A ideia base é conseguir explorar os recursos disponíveis nas máquina ligadas em rede: cpu, disco, largura de banda ...
 - Modelos de interação e coordenação mais complexos (que em sistemas cliente/servidor)
 - Algoritmos mais complexos
 - Não existe ponto único de falha
 - Grande potencial de escalabilidade
 - Adequado para ambientes em que todos os participantes querem cooperar para fornecer um dado serviço

Modelos de Programação Distribuída

Modelos arquiteturais

- Peer processes
 - Exemplos de grandes Sistemas Distribuídos com arquitetura peer-to-peer:
 - Napster, Kazaa, Gnutella, BitTorrent, Skype, ...
 - Duas arquiteturas principais:
 - Um servidor de diretório centralizado (Ex. – Napster)
 - Quando um novo processo se liga, informa o servidor central (do seu endereço, do seu conteúdo)
 - A partir daí pode comunicar com os outros pares
 - Serviço de diretório distribuído:
 - Vários grupos de processos, ex. – Kazaa
 - Quando um novo processo se liga, liga-se a um grupo
 - O líder do grupo regista o conteúdo de todos os elementos do grupo
 - Cada processo acede ao líder do seu grupo para localizar o que pretende.
 - Cada líder pode aceder aos outros líderes.

Modelos de Programação Distribuída

Modelos arquiteturais

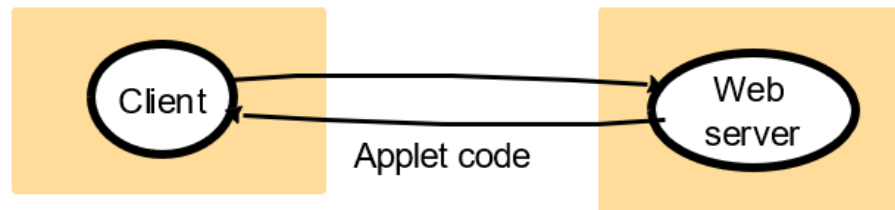
- Outros modelos
 - Variantes do modelo cliente servidor resultantes de:
 - Uso de código móvel;
 - Uso de sistemas com hardware limitado;
 - Requisitos de adicionar/remover ao/do sistema periféricos móveis;

Modelos de Programação Distribuída

Modelos arquiteturais

- Web Applets

a) client request results in the downloading of applet code



b) client interacts with the applet



Modelos de Programação Distribuída

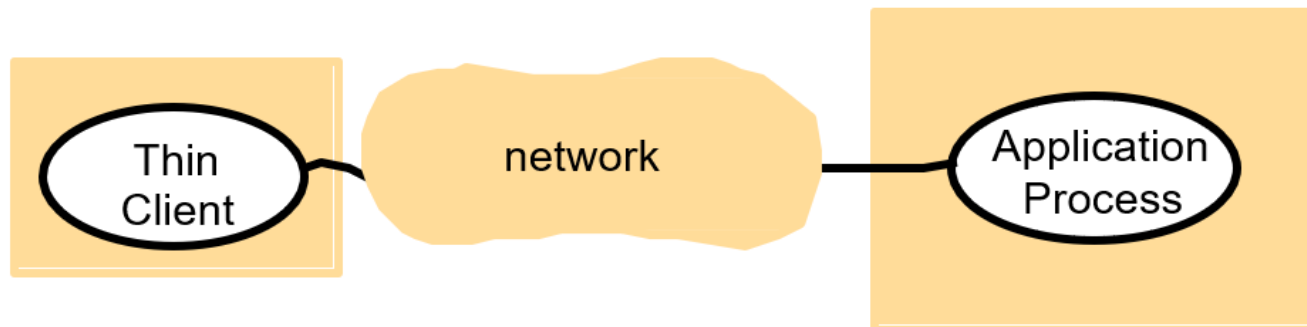
Modelos arquiteturais

- Agentes móveis
 - Um Agente é um programa executável que pode “mover-se” de uma máquina para outra.
 - Age em nome de um utilizador específico, e num dado computador para o qual se transfere realiza algum serviço para o seu proprietário, podendo obter informações que mais tarde transmitirá ao local de origem.
 - Entidade capaz de interagir autonomamente com o ambiente que o rodeia, podendo apresentar características de adaptabilidade, mobilidade, cooperação ou competição.
 - Problemas de segurança
 - Dificuldades em realizar o trabalho devido a problemas impossíveis de prever

Modelos de Programação Distribuída

Modelos arquiteturais

- Thin Clients
 - Interface gráfica, baseada em windows, na máquina local ao utilizador
 - As aplicações executam no servidor problemas para aplicações gráficas interactivas
 - Ex. Citrix WinFrame



Modelos de Programação Distribuída

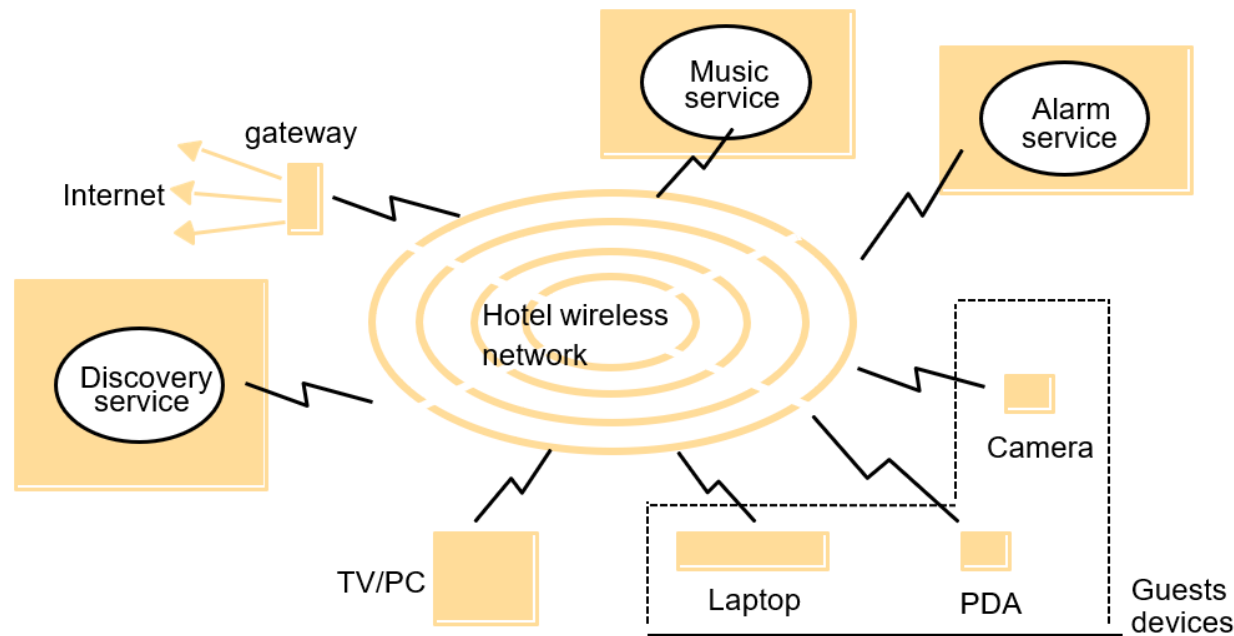
Modelos arquiteturais

- Equipamentos móveis e redes ad-hoc
 - Laptops, PDA (Personal Digital Assistant), telemóveis, câmeras digitais, máquinas de lavar roupa, relógios, etc
 - Protocolos Wireless: BlueTooth, Infrared, HomeRF
 - Principais características das redes sem fios:
 - Configuração é feita automaticamente sem intervenção humana
 - Os equipamentos digitais móveis descobrem por si os serviços disponíveis
 - Problemas:
 - Conexão limitada
 - Se os dispositivos se afastam demasiado do local de transmissão?
 - Segurança e privacidade

Modelos de Programação Distribuída

Modelos arquiteturais

- Equipamentos móveis e redes ad-hoc



Modelos de Programação Distribuída

Modelos arquiteturais

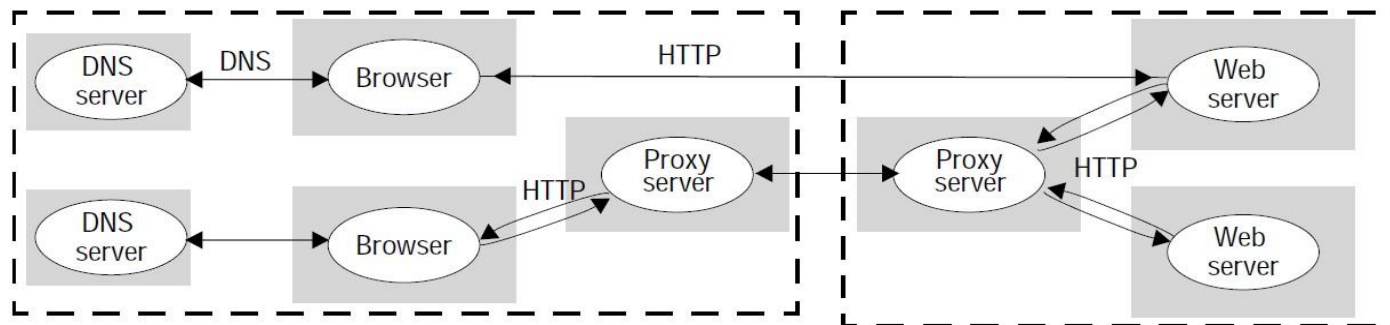
- Equipamentos móveis e redes ad-hoc
 - Redes espontâneas exigem:
 - Um meio de os clientes (equipamentos móveis) descobrirem que serviços estão disponíveis na rede a que se ligaram.
 - Um “discovery service” é um servidor (um ou mais processos) que mantém uma lista dos tipos e características dos serviços disponíveis dentro da rede local sem fios.
 - Redes espontâneas oferecem dois tipos de serviços:
 - Registo de serviços
 - Aceita pedidos para registar numa base de dados os detalhes de cada serviço disponível
 - Lookup de serviços
 - Aceita “queries” aos serviços disponíveis, fornecendo detalhes suficientes para que o cliente se possa ligar ao serviço que escolher

Modelos de Programação Distribuída

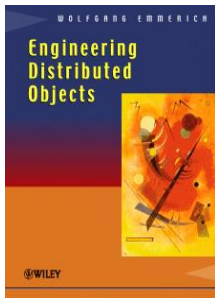
Modelos arquiteturais

- Exemplo: Arquitetura de uma aplicação WEB
 - Um browser é cliente do servidor de DNS e cliente do servidor Web;
 - Algumas intranets, usam servidores Proxy.
 - Quando o proxy está localizado do lado do cliente, minimiza o tráfego e atrasos da rede.
 - Quando o proxy está localizado do lado do servidor reduzem a carga do servidor.
 - O proxy e o servidor cooperam para fornecer o serviço. O proxy é responsável por manter a consistência, verificando com frequência a data em que as páginas foram modificadas no servidor

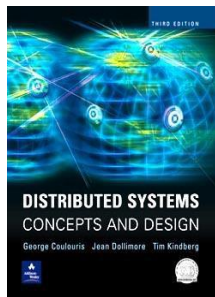
Web:



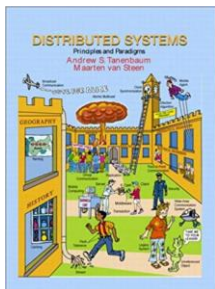
Bibliografia



From: Wolfgang Emmerich
Engineering Distributed Objects
John Wiley & Sons, Ltd 2000



From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design
Edition 4 © Addison-Wesley 2005



From: Andrew S., Tanenbaum and Van Steen, Maarten
Distributed Systems: Principles and Paradigms
Edition 2 © Pearson 2013

Questões?