Modelos Arquiteturais

Carlos Ferraz
cagf@cin.ufpe.br

Tópicos da Aula

- Cliente-servidor
- *Peer processes (P2P)*
- Objetos distribuídos

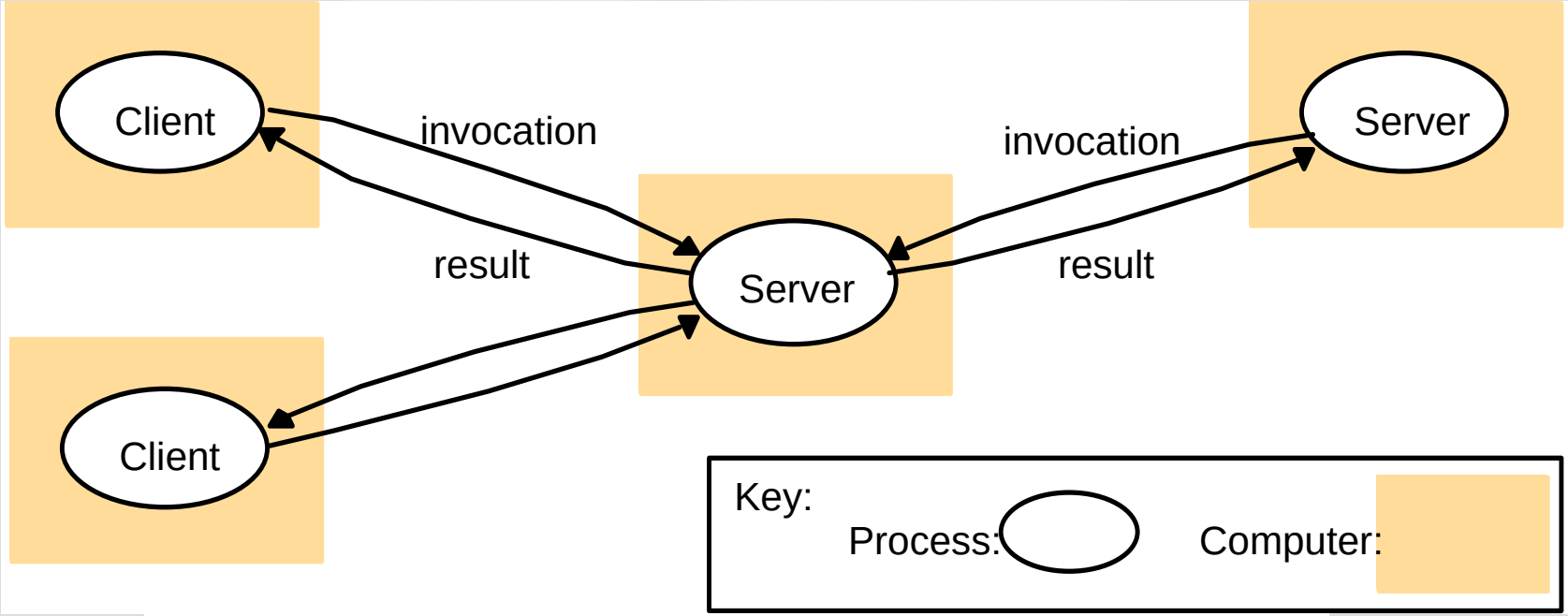
O que é um modelo arquitetural?

- Estrutura em termos de componentes especificados separadamente
- Alocação de componentes em uma rede de computadores
- Interrelações de componentes
- Divisão de responsabilidades entre componentes

Arquiteturas

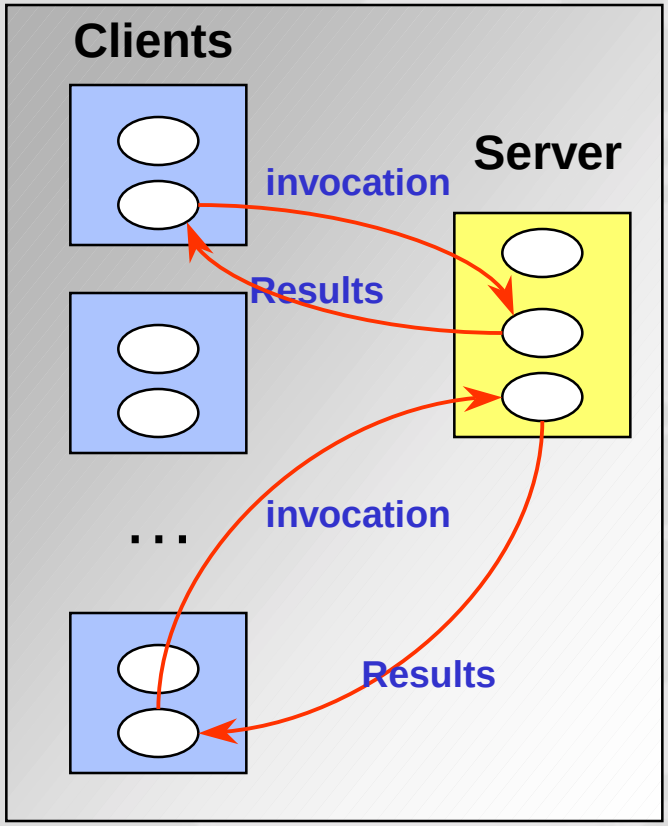
Modelo Cliente-Servidor

Clientes invocando servidores individuais

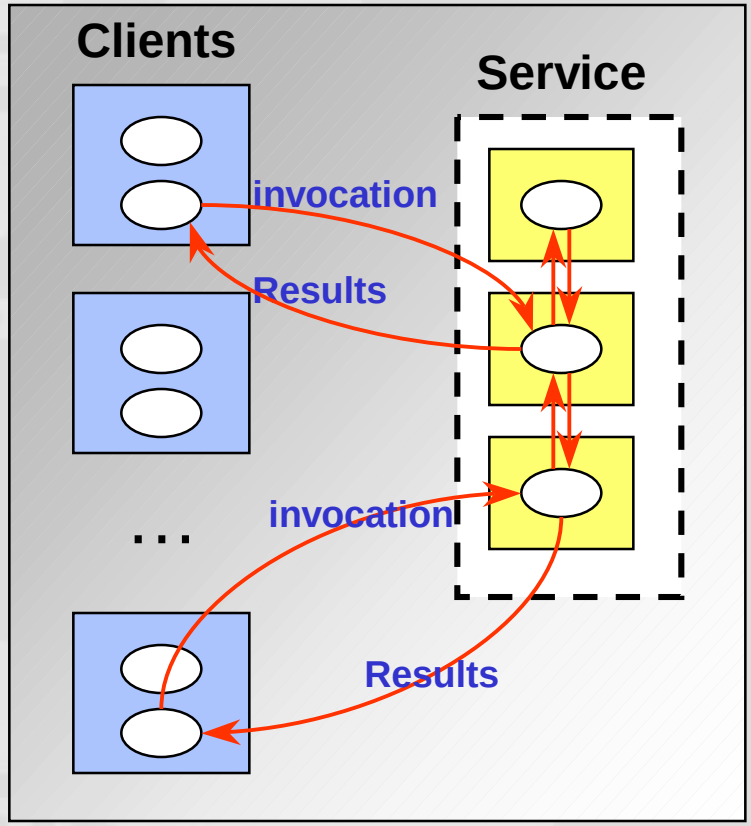


Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 3
© Addison-Wesley Publishers 2000

Arquitetura Cliente-Servidor

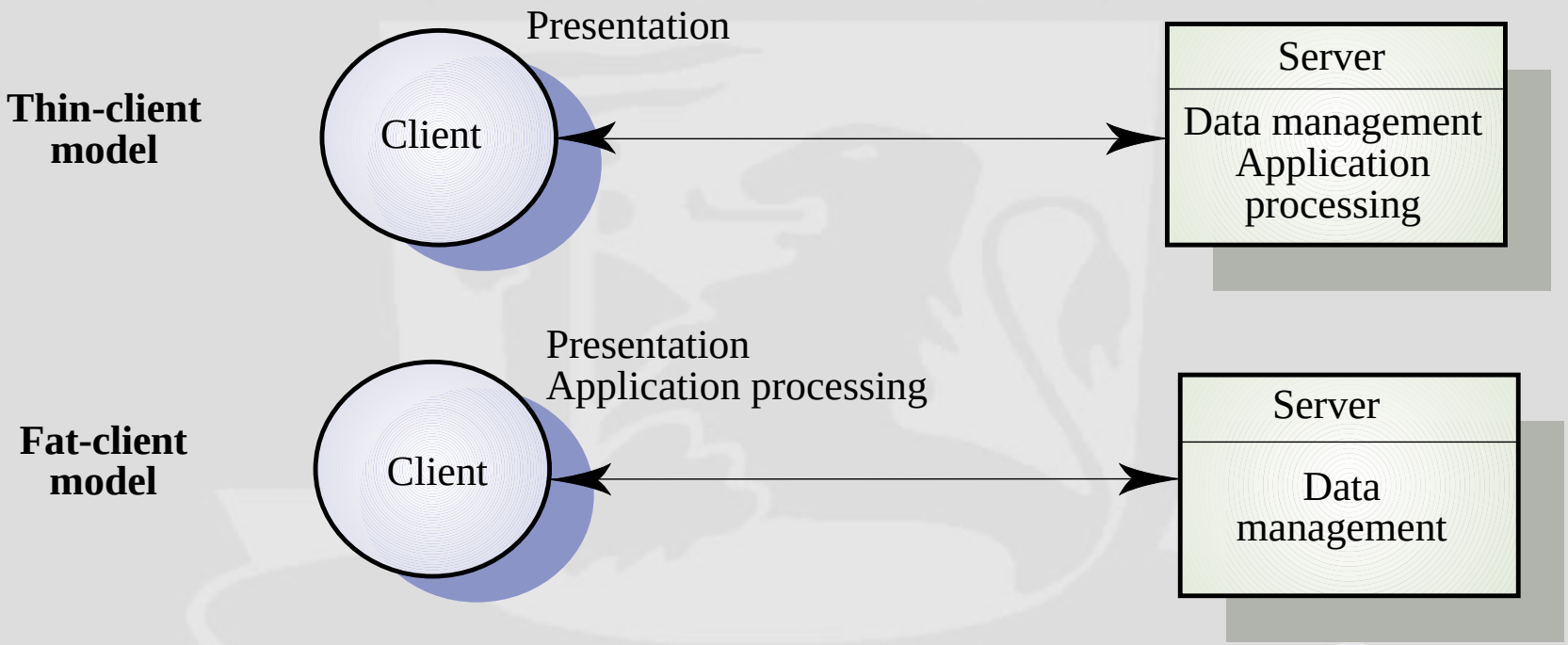


Servidor Único

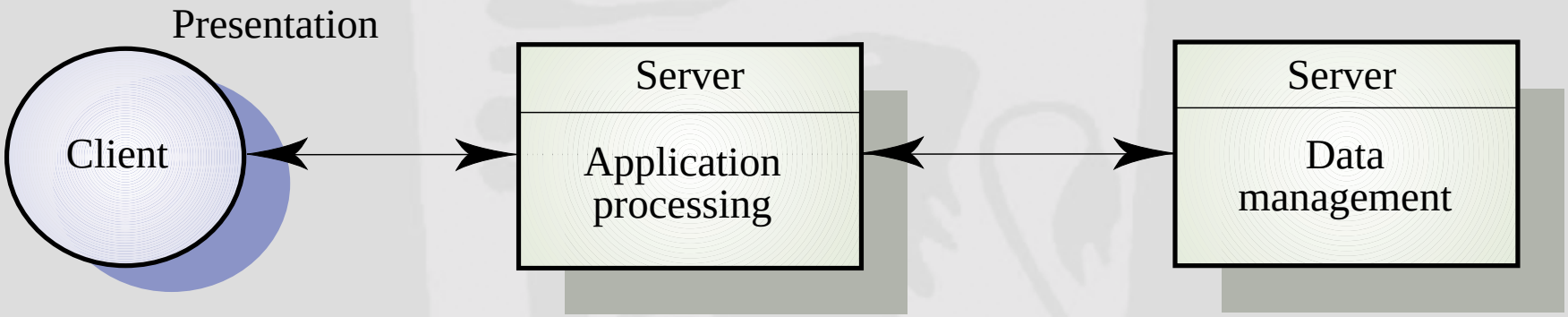


Múltiplos Servidores

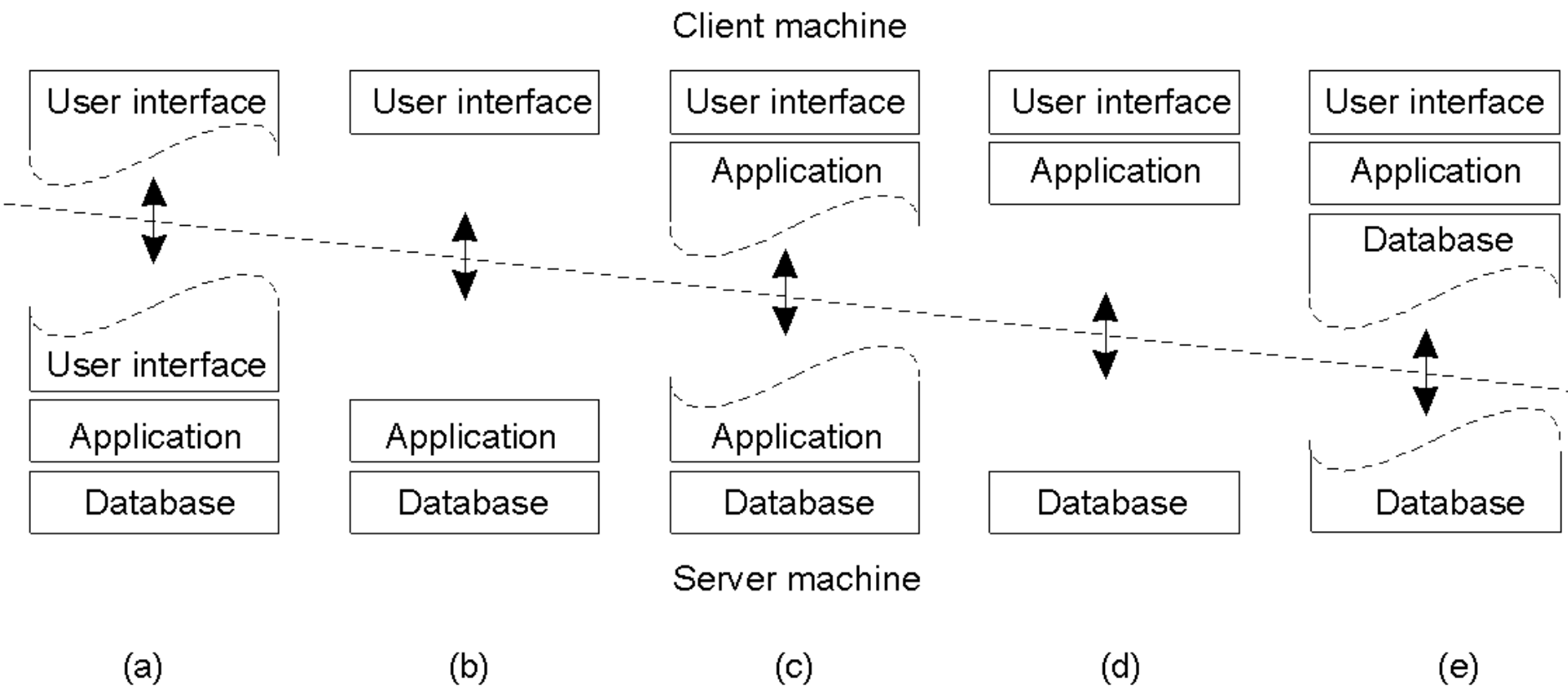
Clintes magros e gordos



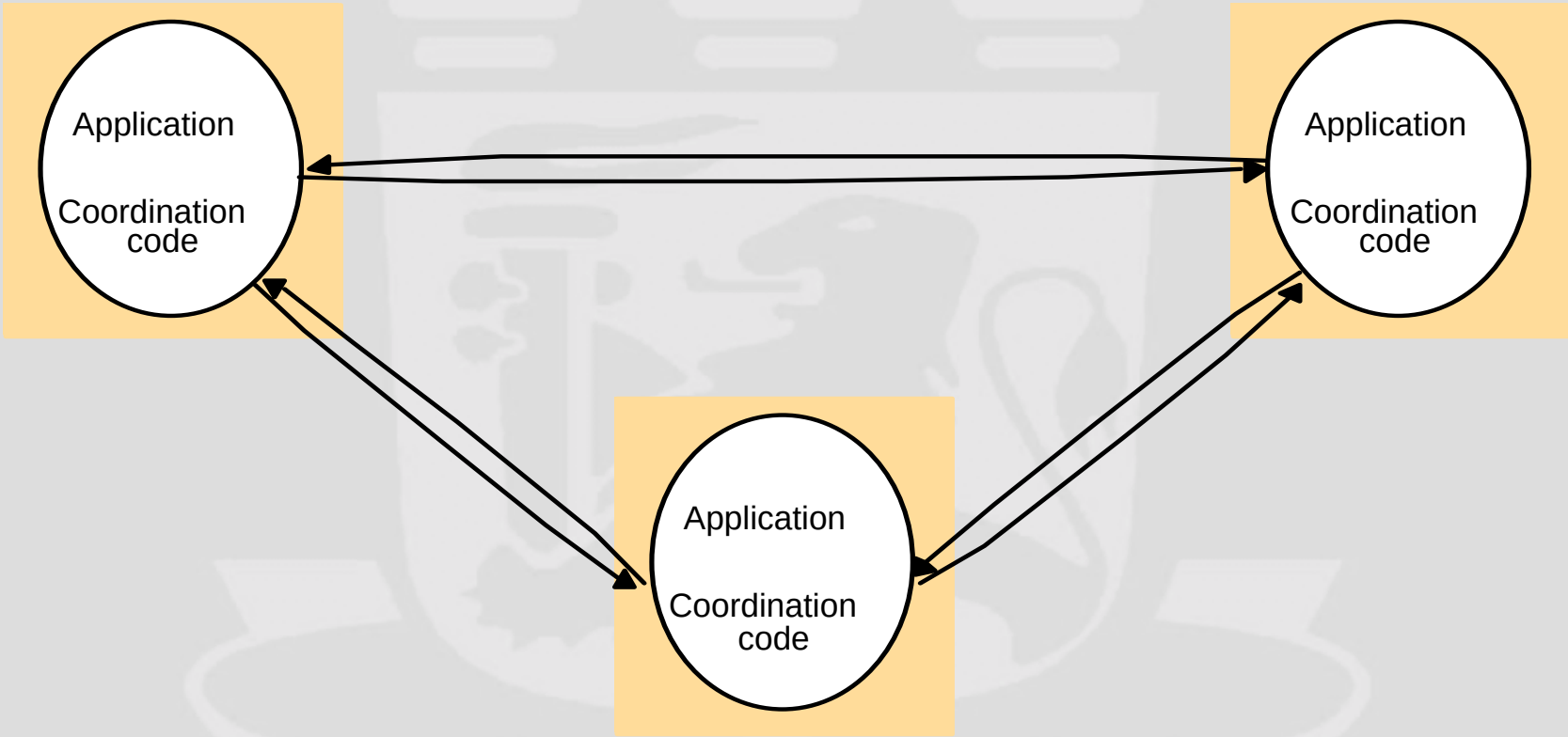
Arquitetura C/S 3-tier



Arquiteturas *Multitiered*



Arquitetura de *Peer processes (Peer-to-Peer)*



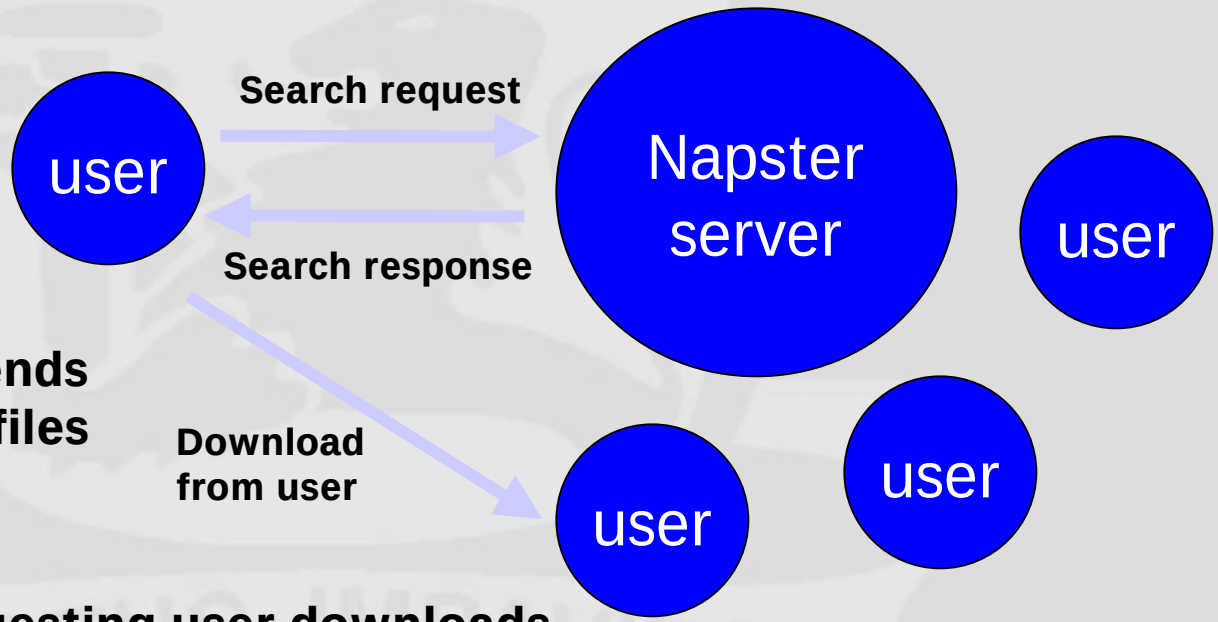
Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 3
© Addison-Wesley Publishers 2000

P2P Centralized: Napster

- Napster used centralized servers to keep a catalog of available files.

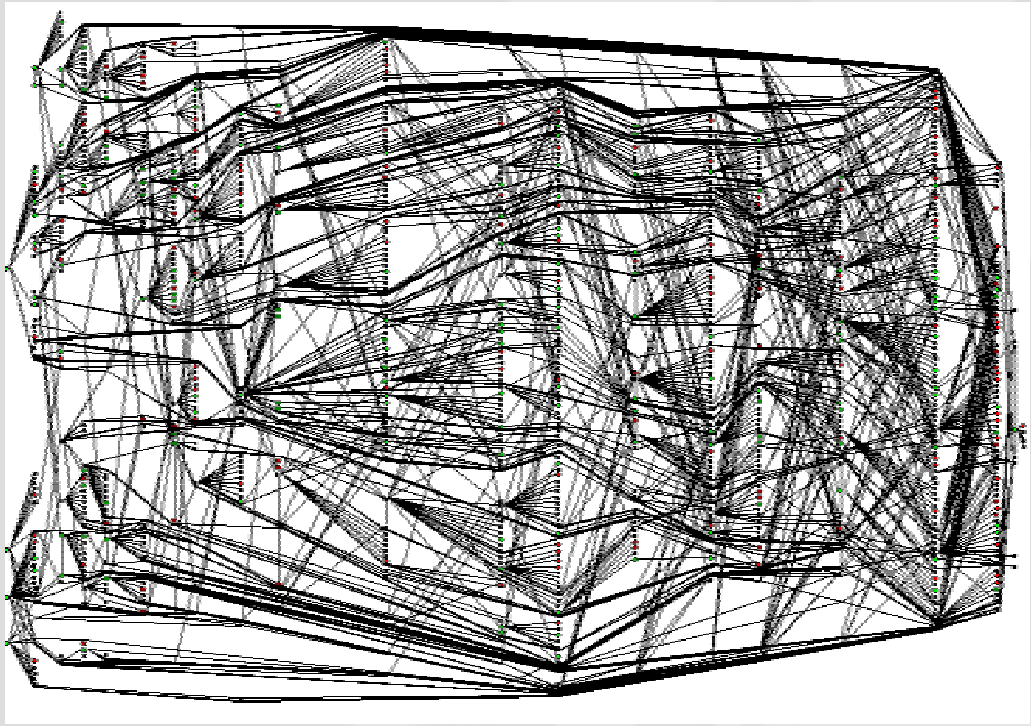
1. User sends out request
Napster searches central database

2. The central server sends back a list of available files for download

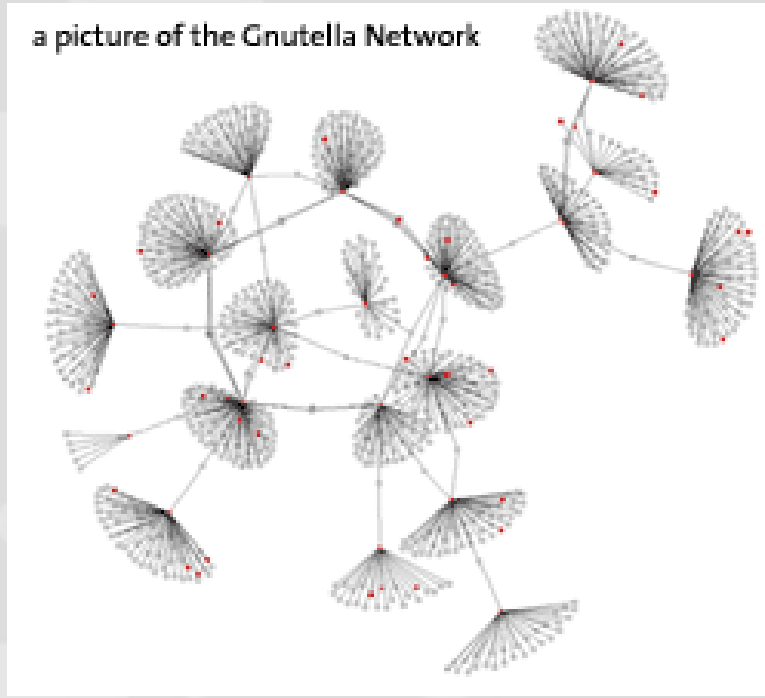


3. Requesting user downloads the file directly from another Napster user computer

P2P Decentralized: Gnutella



Partial Map of the Gnutella Network
http://dss.clip2.com

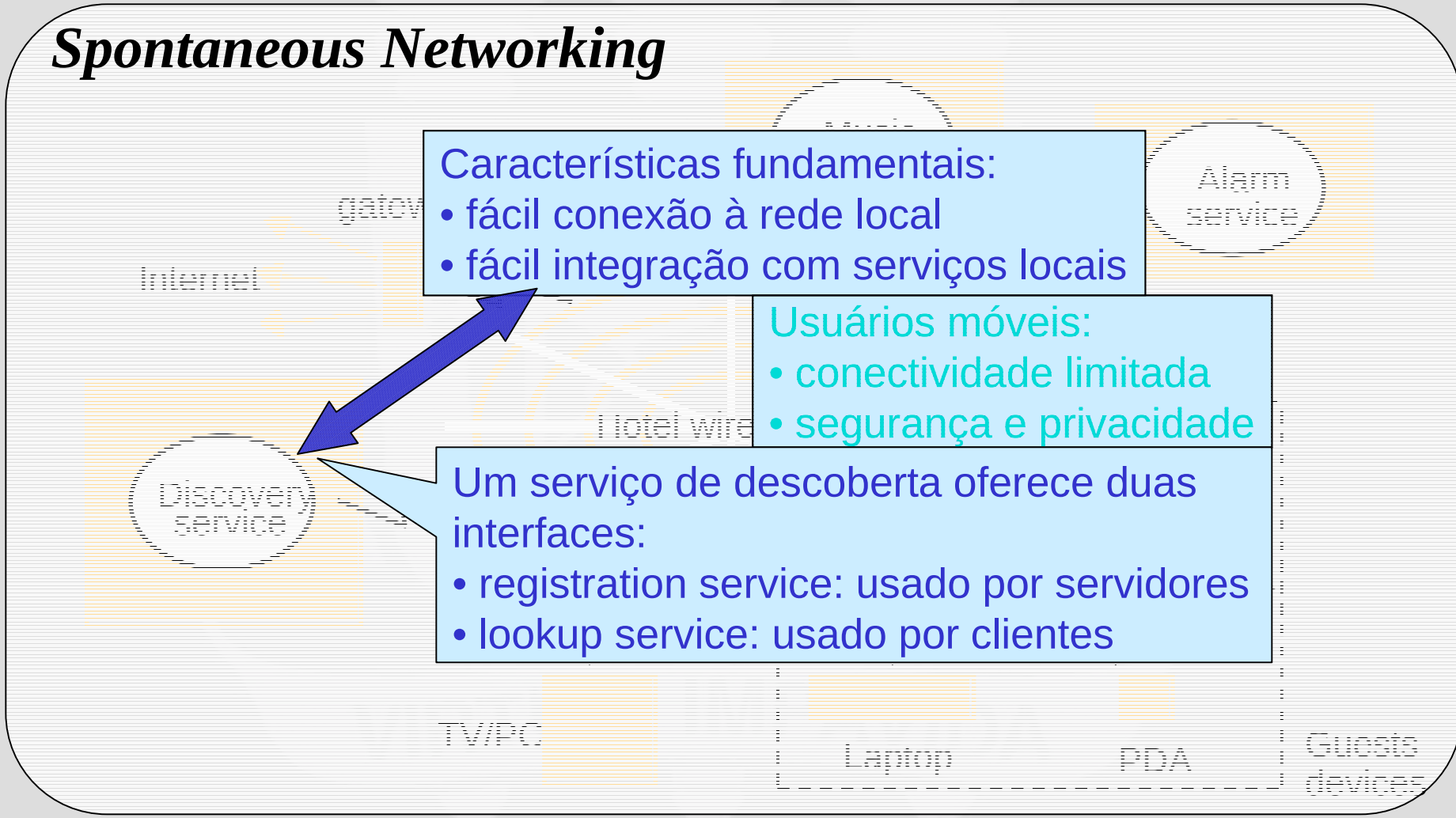


http://www.limewire.com

- See also ***gnuTellaVision: Real Time Visualization of a Peer to Peer Network***

http://www.sims.berkeley.edu/~rachna/courses/infoviz/gtv/paper.html

Exemplo de um SD em um hotel



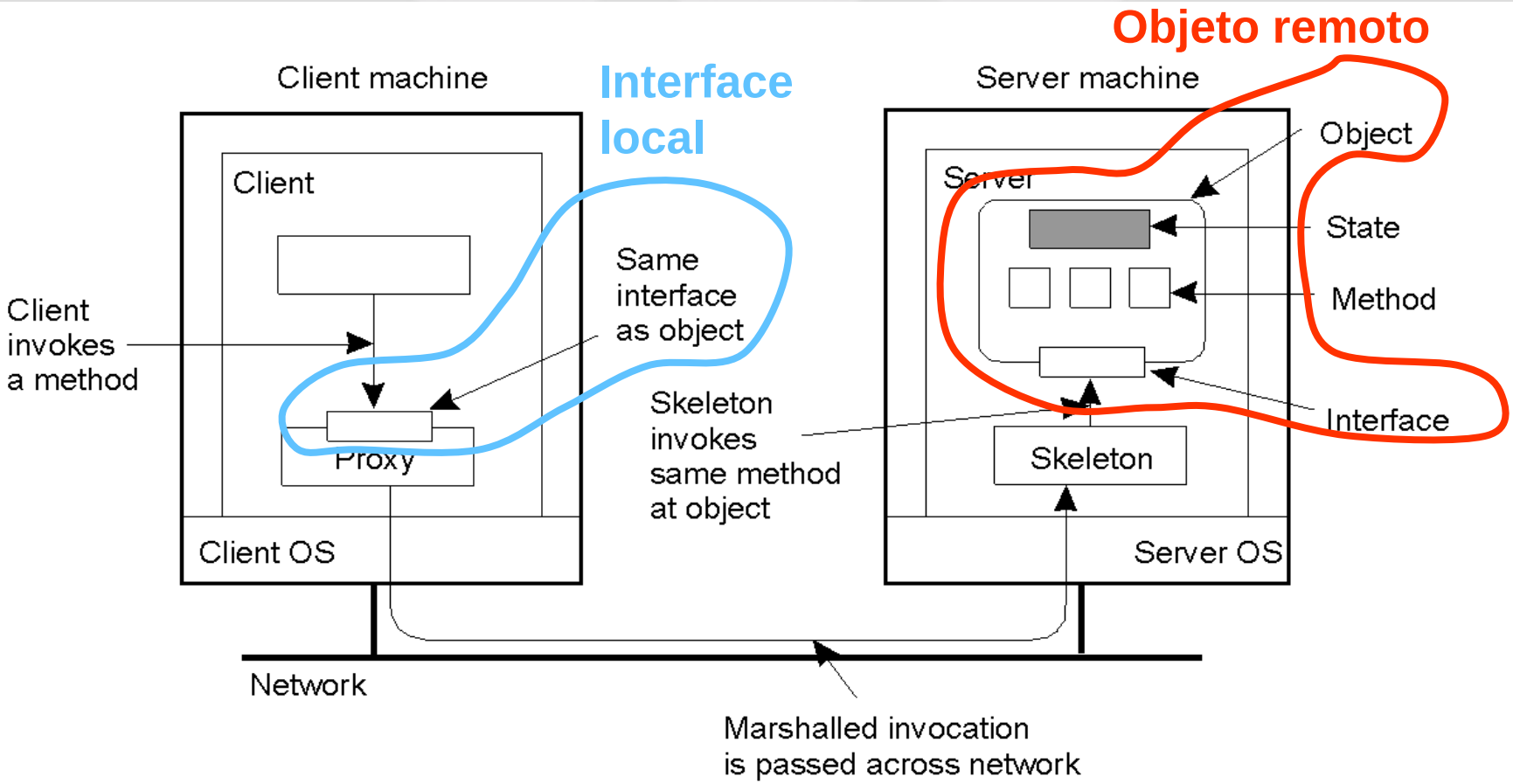
Objetos Distribuídos

- Uma aplicação distribuída pode ser vista como um conjunto de objetos

- Objetos:
  - Consistem de dados + código
  - Podem ser clientes, servidores ou ambos
  - Interface esconde detalhes de implementação

  - Modelar com objetos não implica no uso de programação orientada a objetos

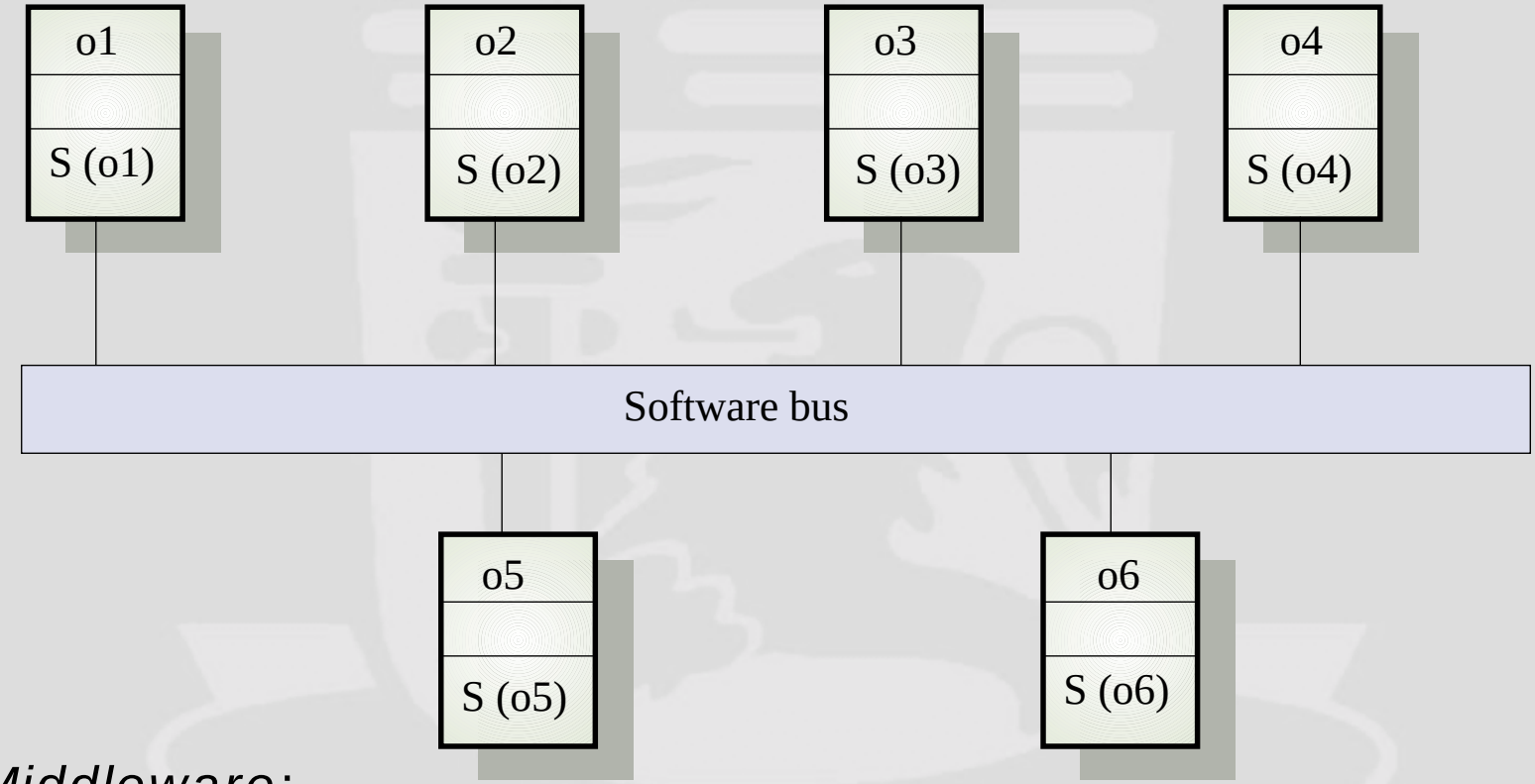
Objetos Distribuídos



Tanenbaum and van Steen. Distributed Systems: Principles and Paradigms © Prentice Hall 2002

- Observe a separação entre interface e objeto

Arquitetura de Objetos Distribuídos



- *Middleware*:
  - *Object brokers*: permitem que objetos se encontrem em um sistema distribuído, e interajam uns com os outros
  - *Object services*: permitem criar, nomear, mover, armazenar e gerenciar objetos

Ligação (*Binding*) Cliente-Objeto

- Ligação implícita

```
Distr_object* obj_ref;          // Declare a systemwide object reference
obj_ref = ...;                  // Initialize the reference to a distributed object
obj_ref-> do_something();       // Implicitly bind and invoke a method
```

- Ligação explícita

```
Distr_object objPref;           // Declare a system-wide object reference
Local_object* obj_ptr;          // Declare a pointer to local objects
obj_ref = ...;                  // Initialize the reference to a distributed object
obj_ptr = bind(obj_ref);        // Explicitly bind and obtain pointer to local proxy
obj_ptr -> do_something();      // Invoke a method on the local proxy
```

Tanenbaum and van Steen. Distributed Systems: Principles and Paradigms
© Prentice Hall 2002

Comentários finais

- O uso de objetos distribuídos melhora a capacidade de manutenção e adaptabilidade de um sistema
- Arquiteturas cliente-servidor fornecem uma infra-estrutura versátil que suporta a inserção de novas tecnologias mais rapidamente
- Arquiteturas de software cliente-servidor têm sido usadas desde os anos 80 → maturidade
- Um número de *tradeoffs* deve ser considerado para selecionar a arquitetura mais apropriada, incluindo:
  - O crescimento potencial do número de usuários,
  - Custo e
  - Homogeneidade do ambiente computacional atual e futuro