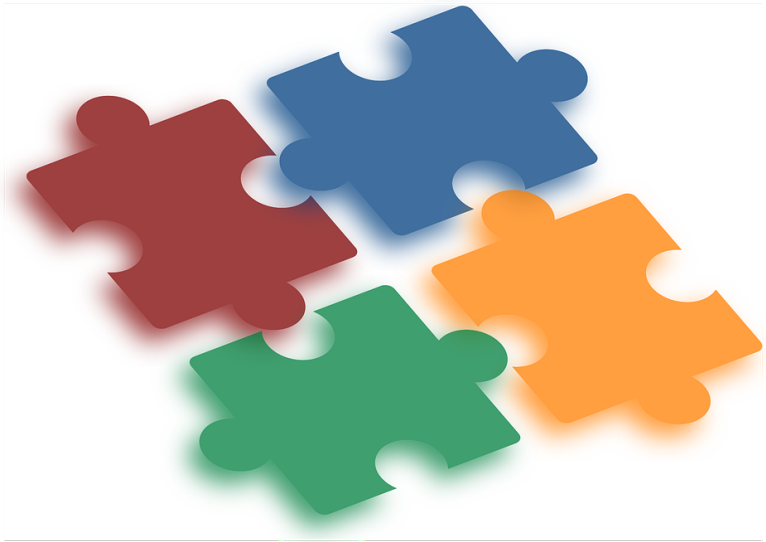
Sistemas Distribuídos (DCC/UFRJ)

Aula 4: Arquiteturas de sistemas distribuídos

Prof. Silvana Rossetto

15 de abril de 2016

Componentes de software distribuídos



- **Sistemas distribuídos**: coleção de peças de software cujos componentes estão espalhados em várias máquinas
- diferentes estratégias para diminuir sua complexidade: ex.: organização lógica X realização física

Arquiteturas de software X arquitetura de sistema

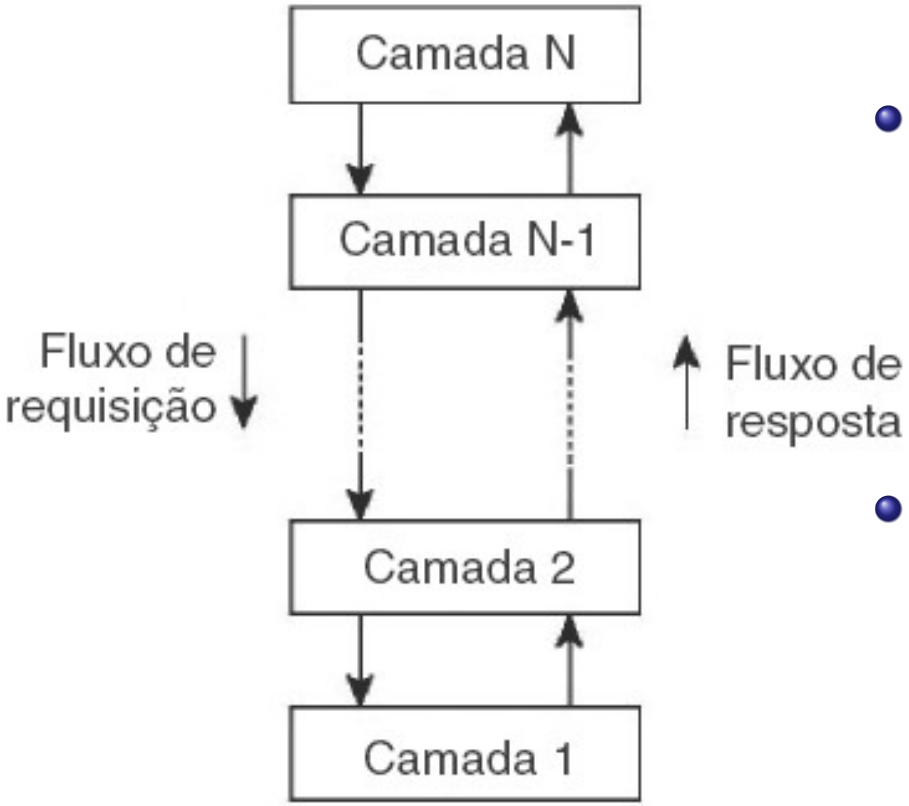
A **arquitetura de software** de SDs trata: *como os componentes de sw são organizados e como eles interagem* (ex. camadas, objetos, eventos, espaço compartilhado)

A **arquitetura de sistema** de SDs trata: *instanciação e implantação da aplicação em máquinas reais*

Arquiteturas de software

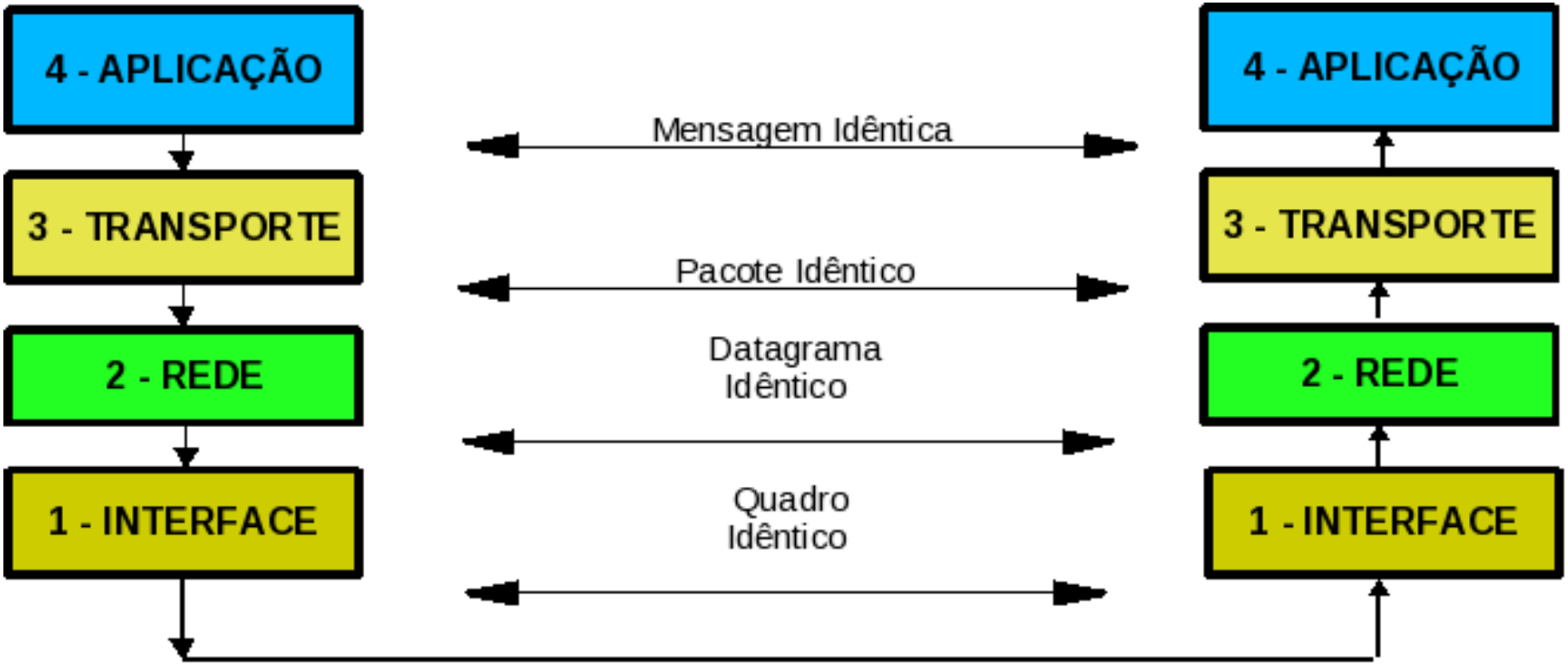
Em sistemas distribuídos, **projetar ou adotar uma arquitetura de software é crucial para o desenvolvimento de grandes sistemas**

Arquitetura de software: em camadas

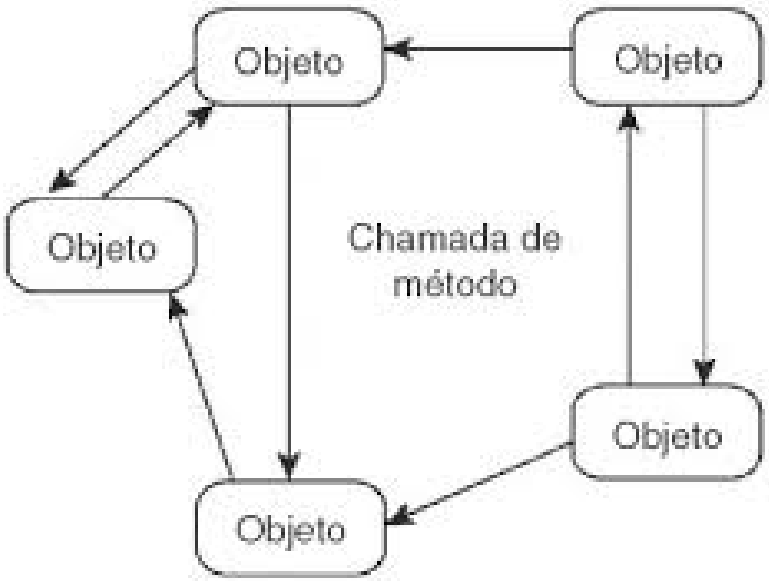


- um componente da camada L_i tem permissão para chamar um componente da camada L_{i-1}, o contrário não é permitido
- o controle flui entre camadas adjacentes: **requisições descem e respostas sobem**

Arquitetura de software: em camadas (exemplo Internet)

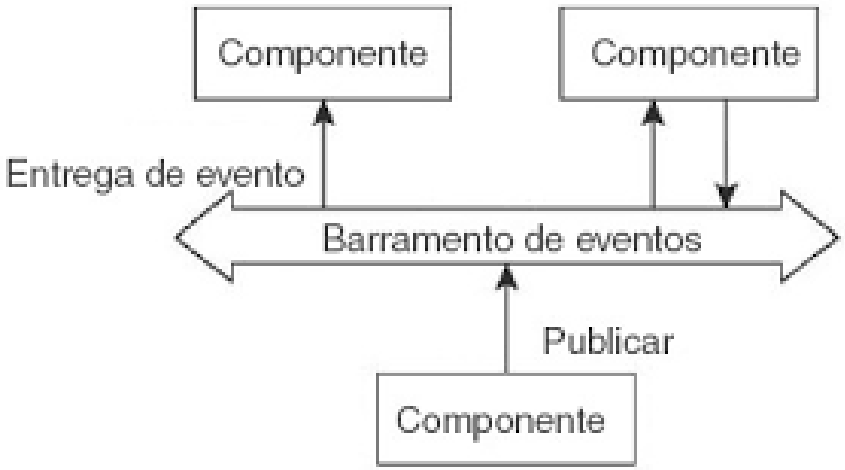


Arquitetura de software: baseada em objetos



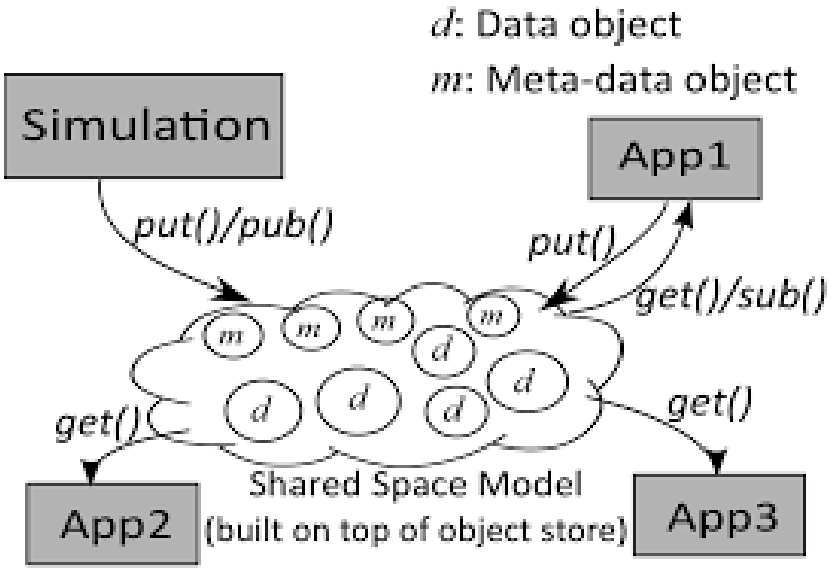
- organização mais "solta" dos componentes (mapeados para objetos)
- **chamada remota de métodos**
- ou **chamada remota de procedimento** (em programação estruturada)

Arquitetura de software: baseada em eventos



- programação baseada em eventos
- **sistemas publish/subscribe**
- desacoplamento no espaço (os processos não precisam se referenciar diretamente)

Arquitetura de software: espaço compartilhado

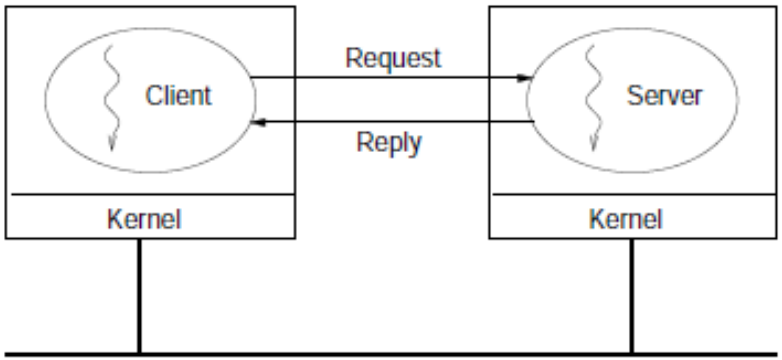


- programação baseada em eventos combinada com um repositório comum de dados
- desacoplamento no espaço e no tempo (os processos não precisam estar ativos quando ocorre a comunicação)

Arquiteturas de sistema

- como o sistema é organizado fisicamente...
- onde os componentes de software são colocados...
- como esses componentes interagem...

Arquitetura cliente/servidor

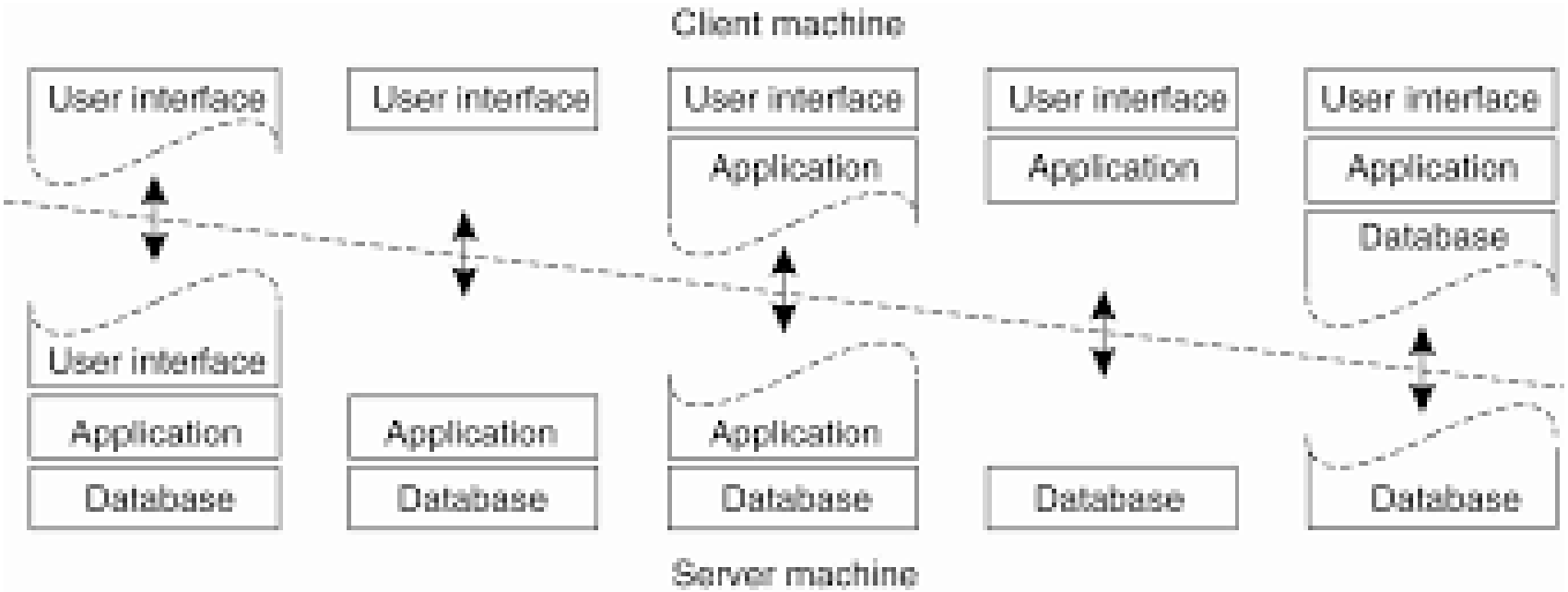


- amplamente usada no desenvolvimento de SDs
- os processos assumem papéis distintos: **servidor** ou **cliente**
- os *servidores* provêem serviços (*ex., sistema de arquivo, banco de dados*)
- os *clientes* usam esses serviços

Modelo em camadas + C/S

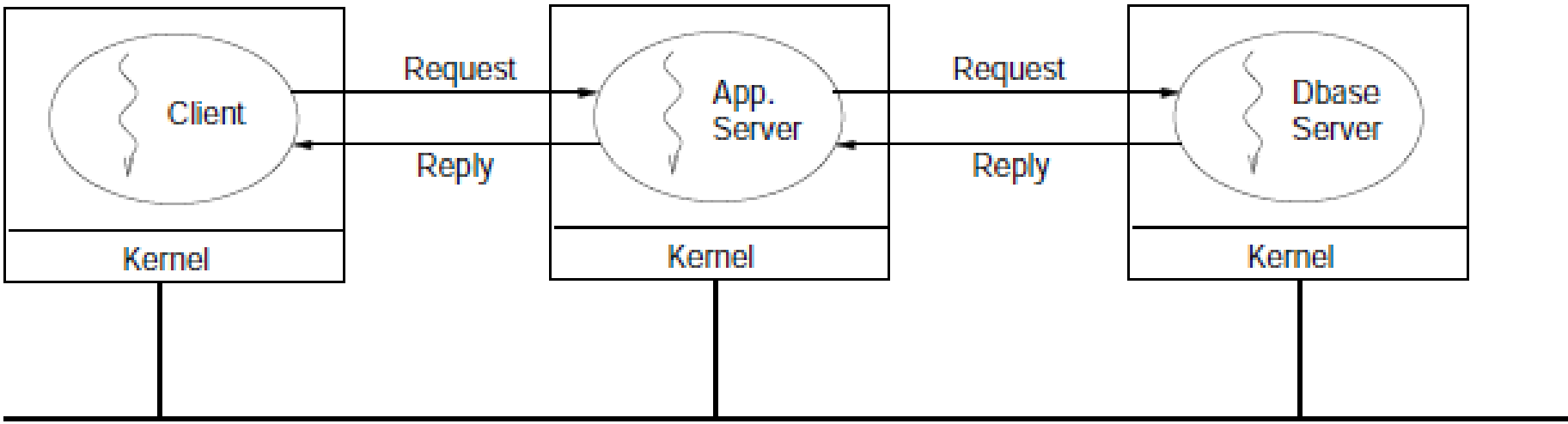
Muitas aplicações C/S são projetadas em três camadas principais:

- interface de usuário
- processamento
- gerência de dados (persistência)



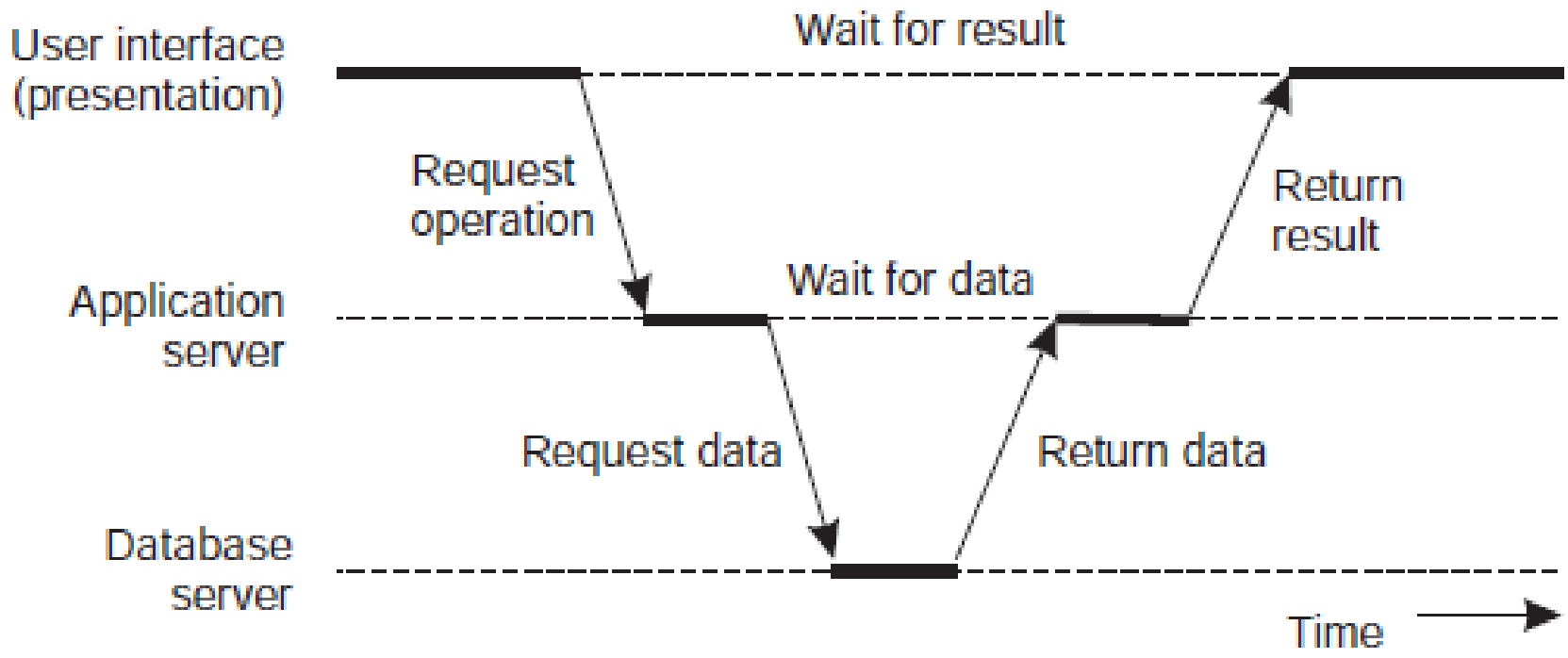
Arquitetura C/S (distribuição vertical)

A funcionalidade de um servidor pode ser **dividida entre vários servidores** (servidores são clientes de outros servidores):



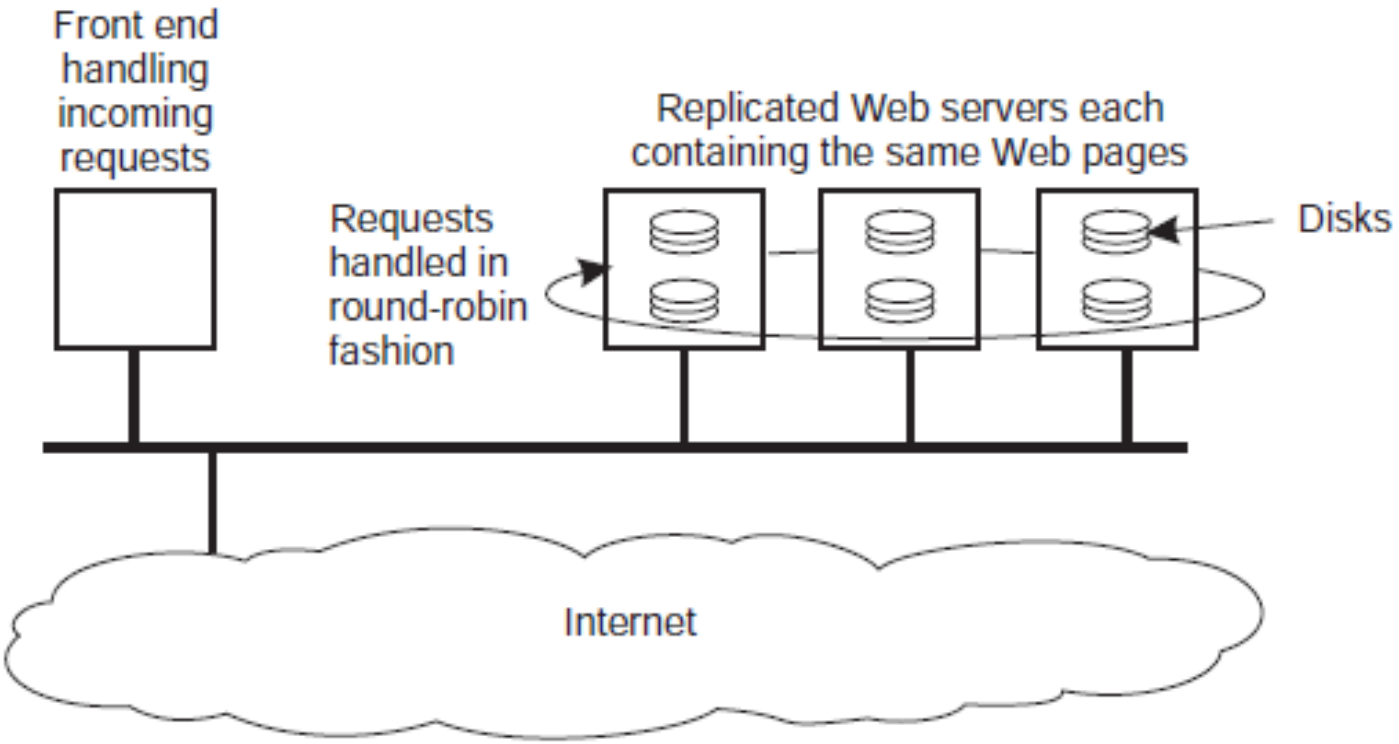
Arquitetura C/S (distribuição vertical)

A funcionalidade de um servidor pode ser **dividida entre vários servidores** (servidores são clientes de outros servidores):



Arquitetura C/S (distribuição horizontal)

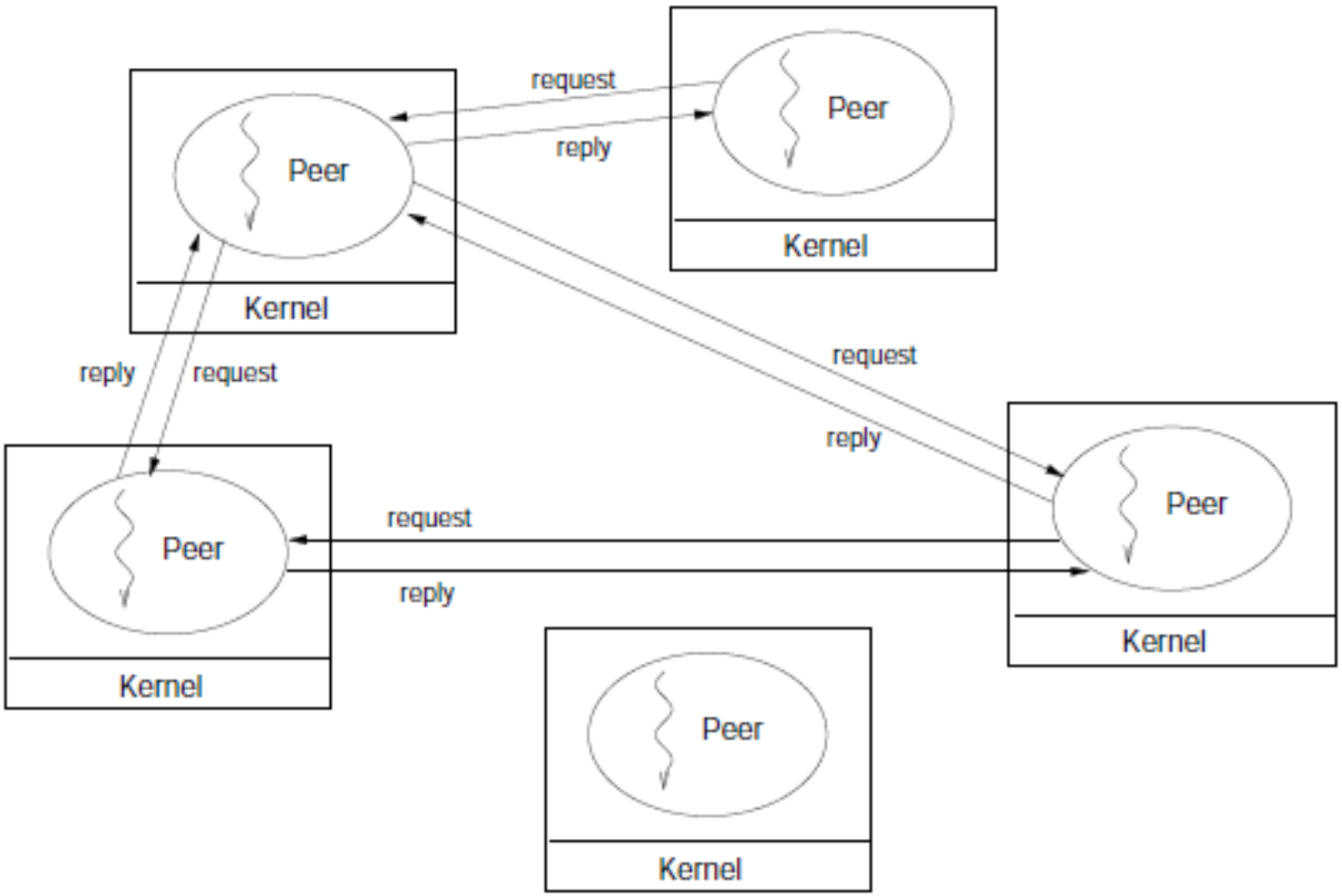
A funcionalidade de um servidor pode ser **subdividida em várias máquinas**:



Arquitetura Peer-to-Peer

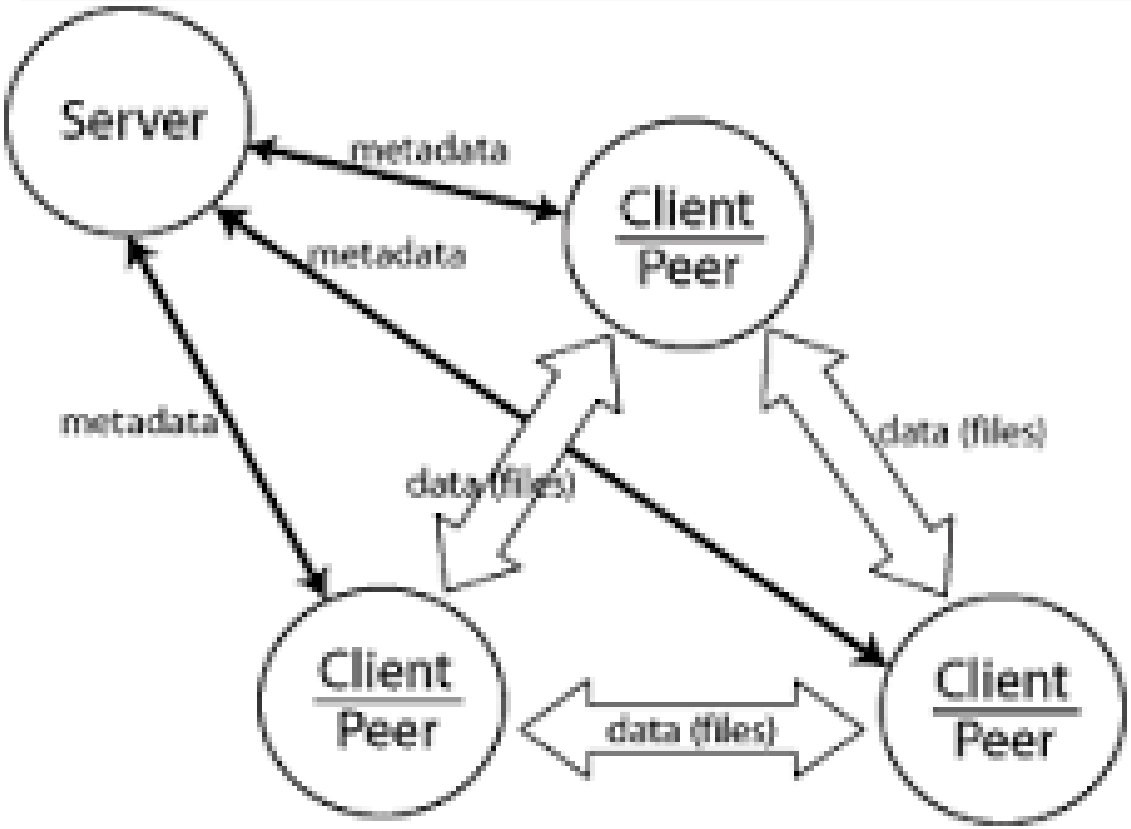
- Adota a abordagem inversa, assume que todos os processos desempenham o mesmo papel, e por isso são "pares" uns dos outros
- Todos os processos da aplicação oferecem os mesmos serviços lógicos
- Os nós da arquitetura P2P formam uma **rede sobreposta** (rede virtual na qual os nós tem conhecimento apenas dos nós com quem compartilham informação)
- Cada nó mantem uma lista de vizinhos (visão parcial da rede)

Arquitetura Peer-to-Peer



Arquiteturas híbridas

combinam arquiteturas centralizadas (C/S) com arquiteturas decentralizadas (P2P)



Exercício

Implementar um *chat* distribuído

- definir a arquitetura do sistema
- definir a arquitetura de software

Referências bibliográficas

1. A. S. Tanenbaum and M. Van Steen, *Sistemas Distribuídos: princípios de paradigmas*, 2007