UNIFESSPA
Universidade Federal do Sul e Sudeste do Pará

Universidade Federal do Sul e Sudeste do Pará

Sistemas Distribuídos

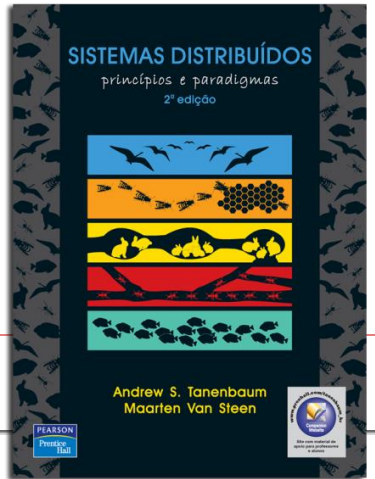
Prof.: Warley Junior
wmvj@unifesspa.edu.br

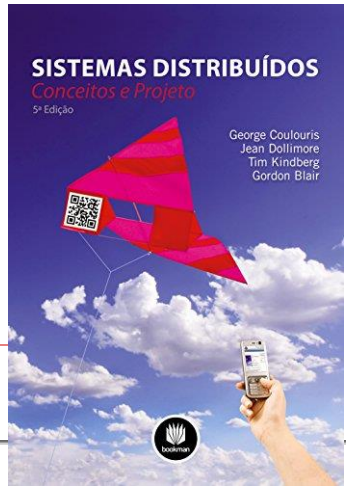
Agenda

- AULA 2
- Estilos de arquitetura
  - Arquitetura de camadas
  - Arquitetura baseada em objetos
  - Arquitetura orientada a serviços
  - Arquitetura centradas em dados
  - Arquitetura baseada em eventos

Leitura Prévia

- COULOURIS, George. Sistemas distribuídos: conceitos e projetos. 5ª ed. Porto Alegre: Bookman, 2013.
  - Capítulo 2.
- TANENBAUM, Andrew S. Sistemas distribuídos: princípios e paradigmas. 2ª ed. São Paulo: Pearson Prentice Hall, 2007.
  - Capítulo 2.

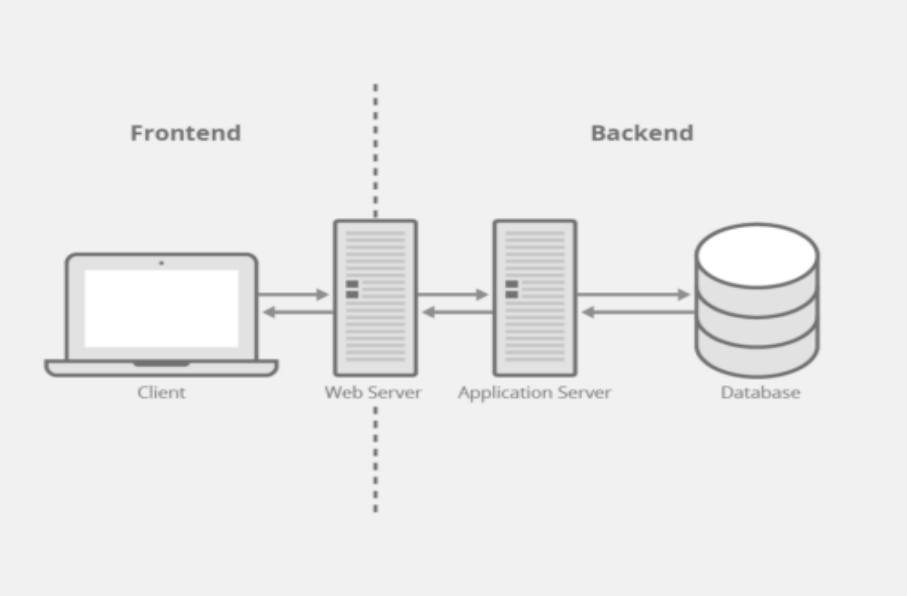


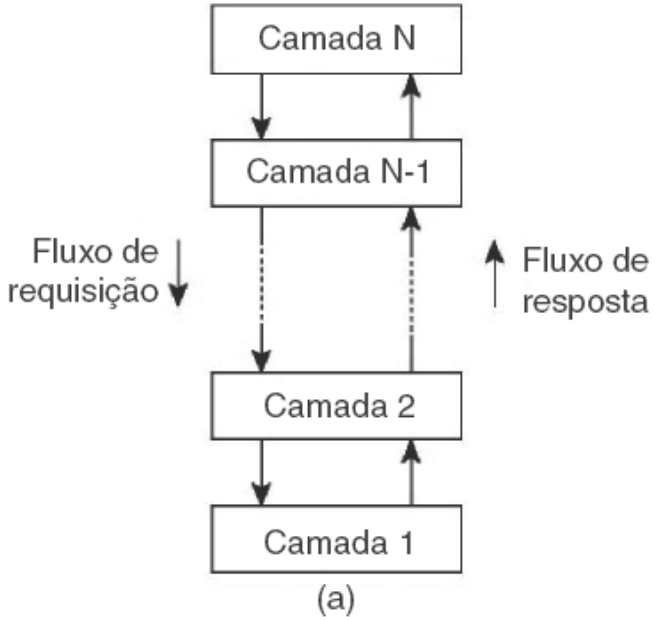


Arquitetura

- Estilo arquitetônico
  - Componentes
  - Conexões
  - Dados intercambiados
  - Formas de configuração

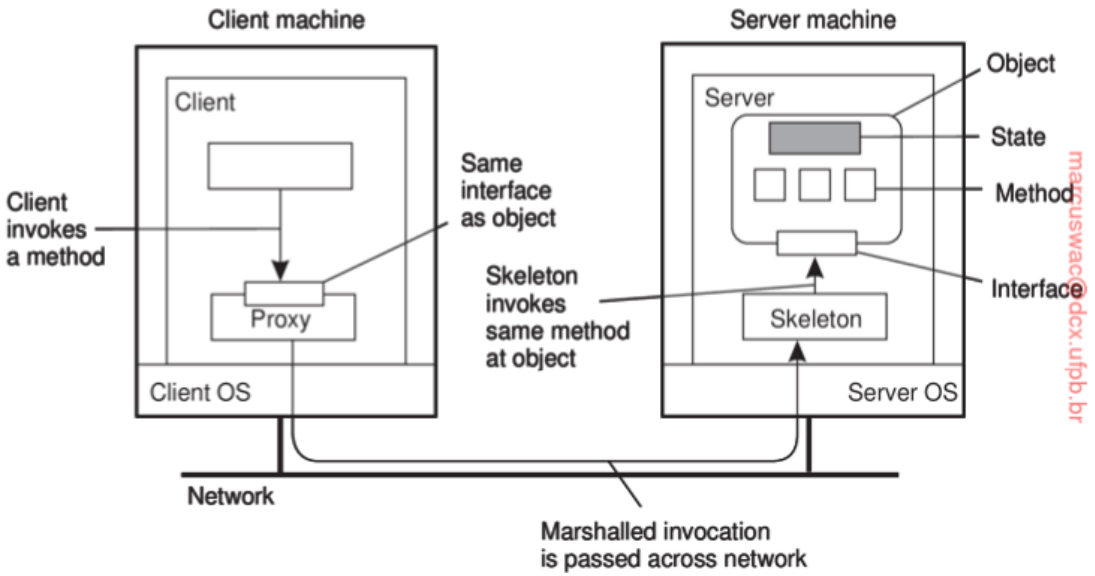
Arquitetura de camadas

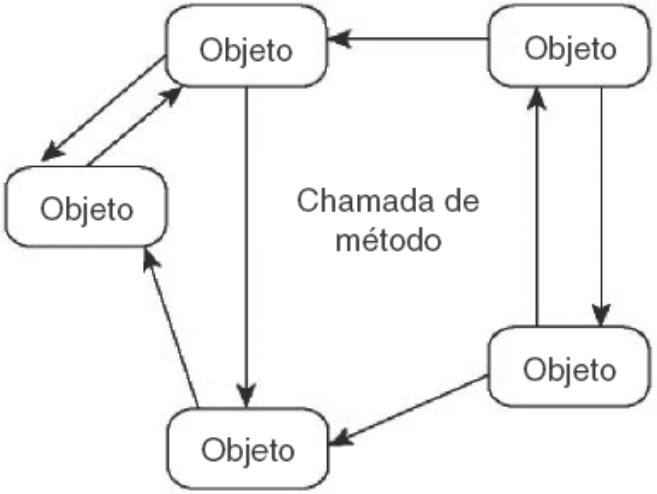




- Componentes são organizados em camadas
- Componente da camada N tem permissão para chamar componentes da camada N-1

Arquitetura baseada em objetos



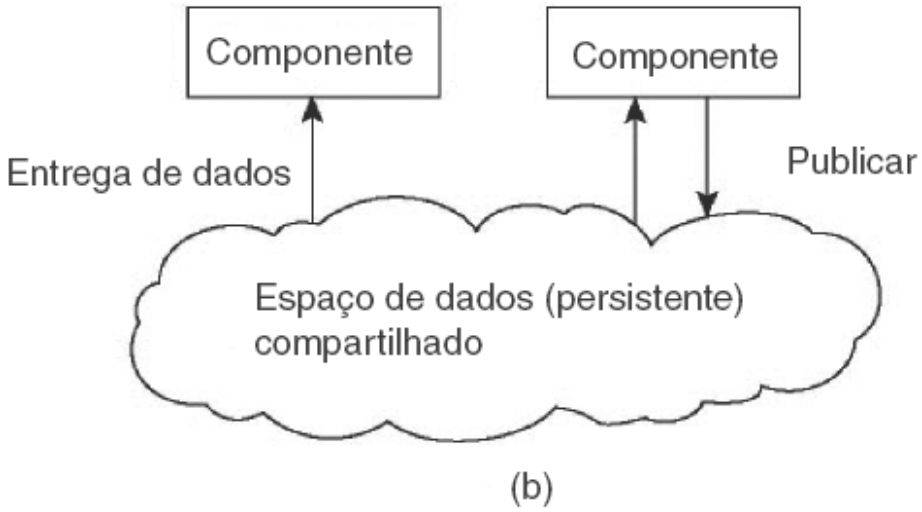


- Objetos distribuídos interagem via chamada remota de métodos
  - Benefícios da orientação a objetos, rodando em máquinas diferentes

Arquitetura orientada a serviços

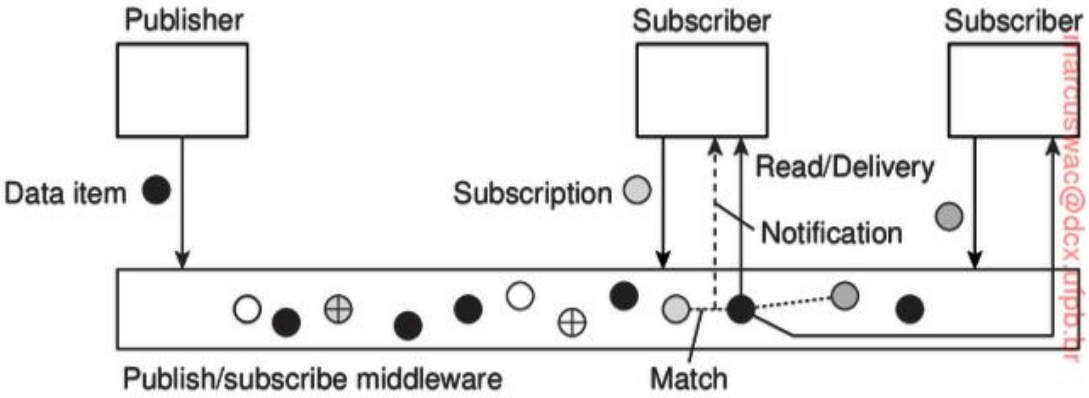
- Separação de serviços que operam de forma independente
  - Ideia similar aos objetos distribuídos
- Aplicação distribuída é composta por vários serviços diferentes
  - Importante ter interfaces bem definidas e padrão de comunicação

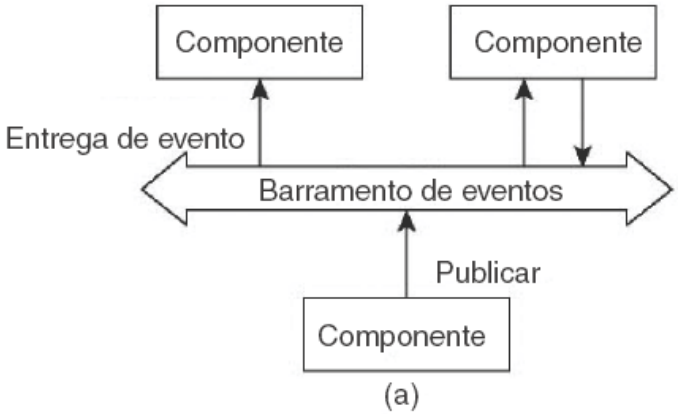
Arquitetura centrada em dados



- Componentes se comunicam através de um repositório comum, como se fosse uma "caixa postal"
- Fracamente acoplado

Arquitetura baseada em eventos





- Sistema publicar-subscrever (*publish-subscribe*)
- Componentes publicam eventos e certificam que somente os que subscreveram recebem estes eventos.

Agenda

- AULA 2
- Arquitetura de Sistemas Distribuídos
  - Arquitetura centralizada
  - Arquitetura descentralizada
  - Arquitetura híbrida

Arquitetura de sistemas

- Arquiteturas centralizadas
  - Cliente-servidor: Netflix, site de notícias
- Arquiteturas descentralizadas
  - Sistemas Peer-to-Peer
- Arquiteturas híbridas
  - Sistemas Peer-to-Peer, como o BitTorrent, Skype e WhatsApp

Arquiteturas centralizadas

- Cliente-Servidor
  - Servidor: processo que implementa um serviço específico
  - Cliente: processo que requisita um serviço específico
  - Forma de interação: requisição-resposta

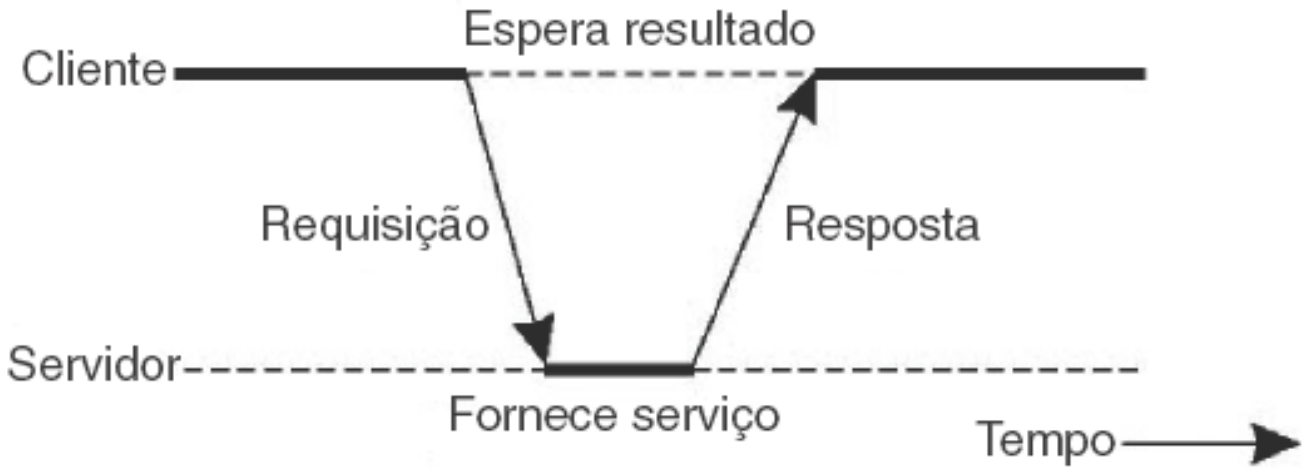


Figura 2.3 Interação geral entre um cliente e um servidor.

Arquiteturas centralizadas

- Camadas de aplicação

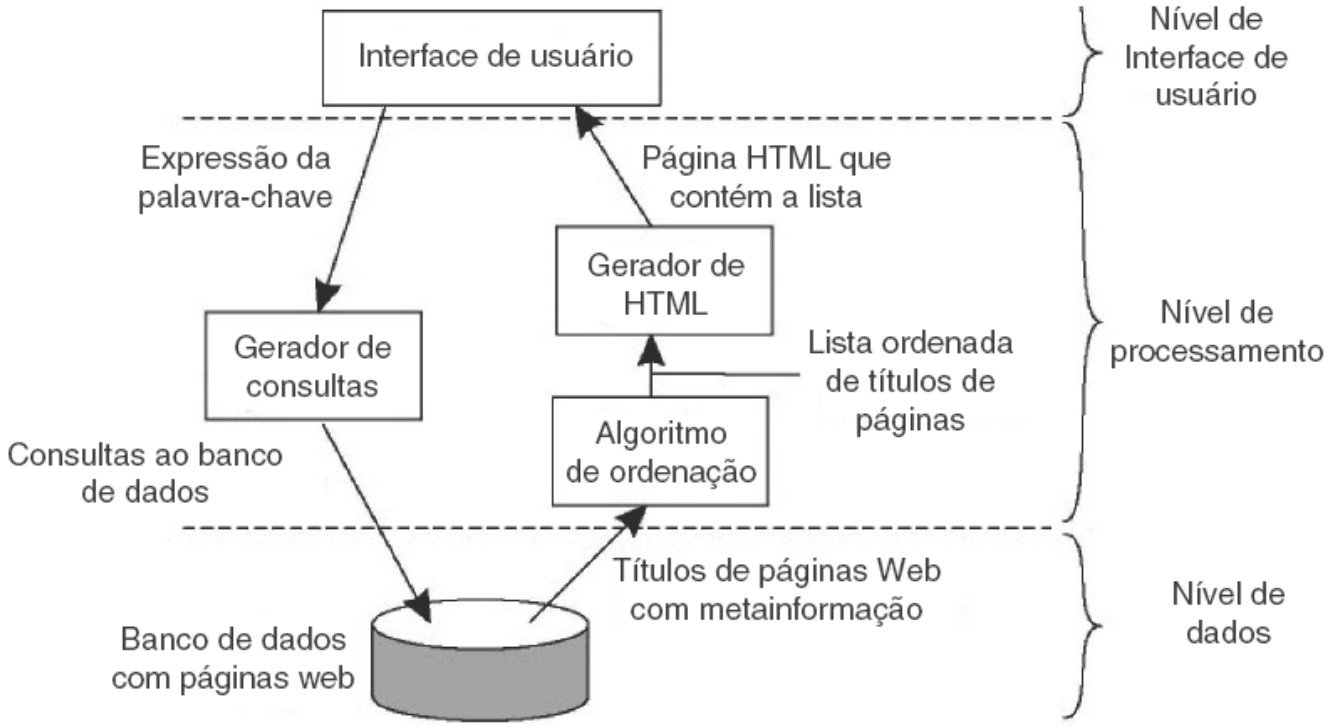


Figura 2.4 Organização simplificada de um mecanismo de busca da Internet em três camadas diferentes.

Arquitetura centralizada com arquiteturas multidivididas

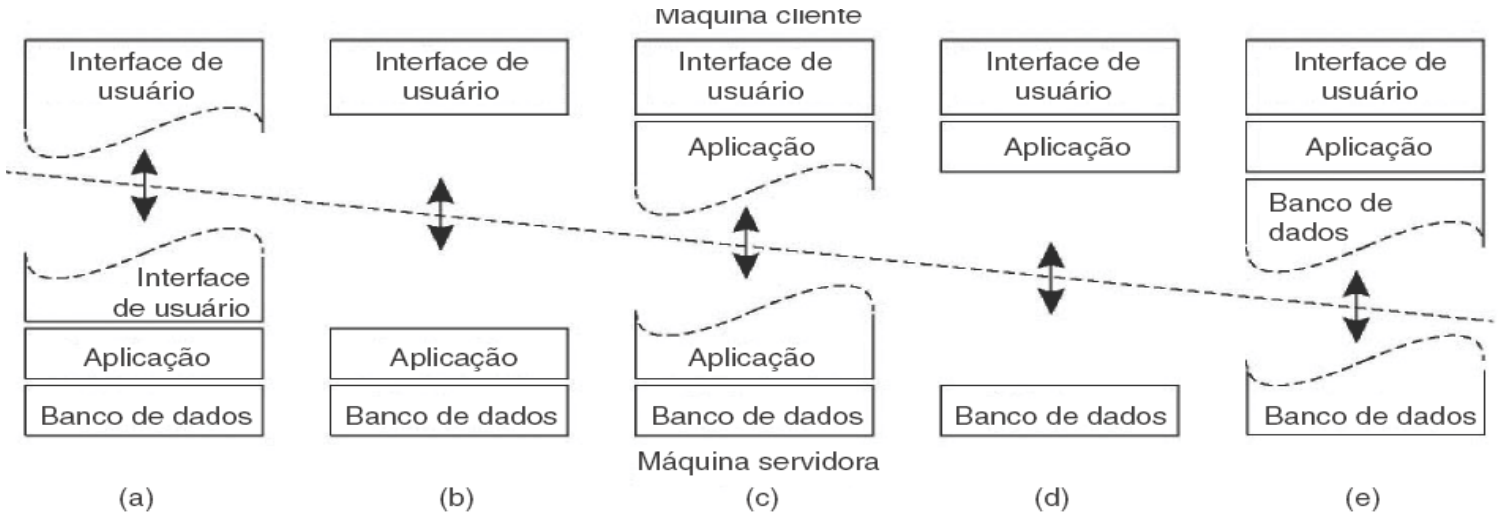
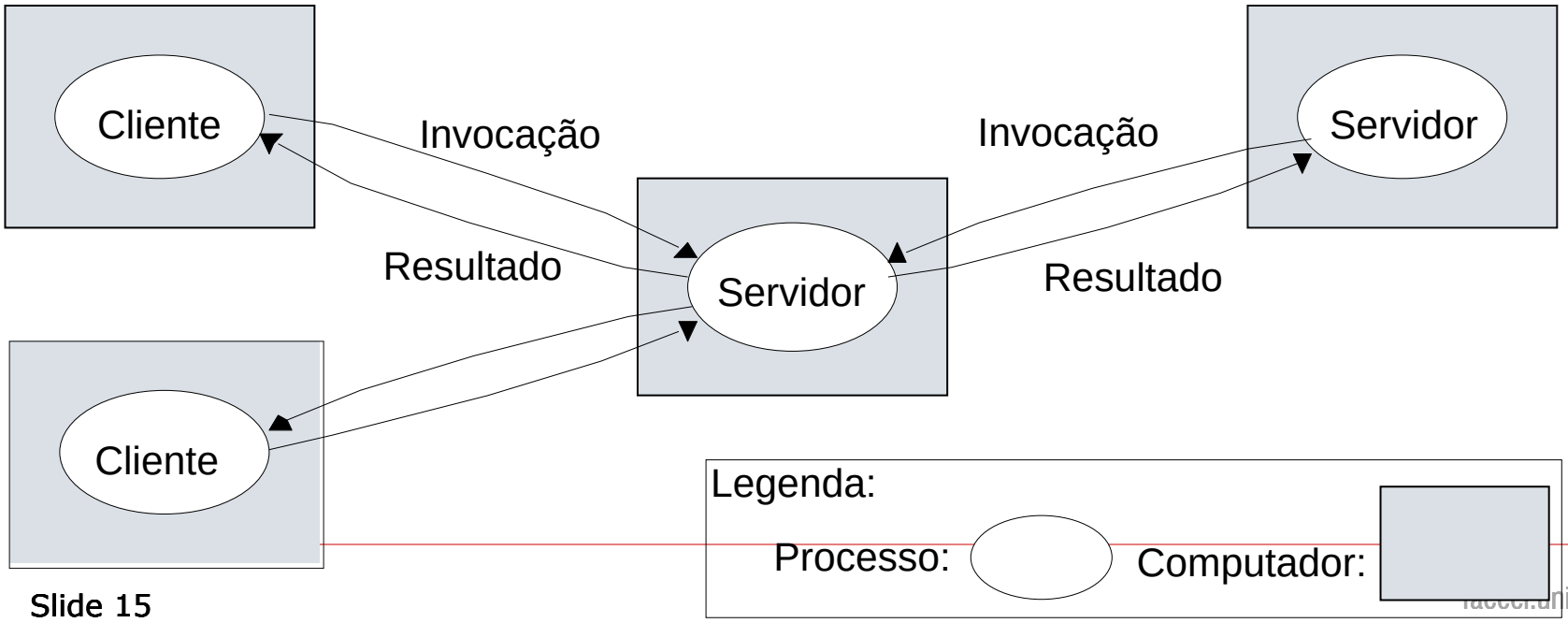


Figura 2.5 *Alternativas de organizações cliente–servidor (a)—(e).*

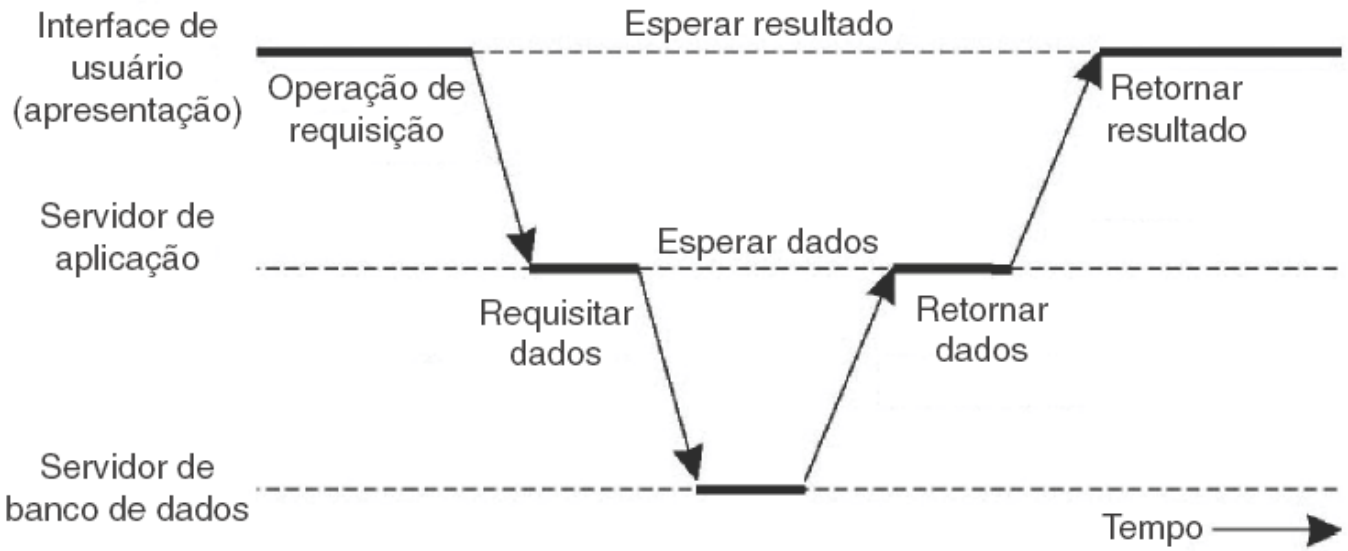
- Gerenciamento de sistema:
  - Clientes gordos (*fat clients*)
  - Clientes magros (*thin clients*)

Arquiteturas centralizadas

- Cliente-servidor de 3 divisões: processos clientes interagem com processos servidores, localizados em distintos computadores hospedeiros, para acessar os recursos compartilhados que estes gerenciam.



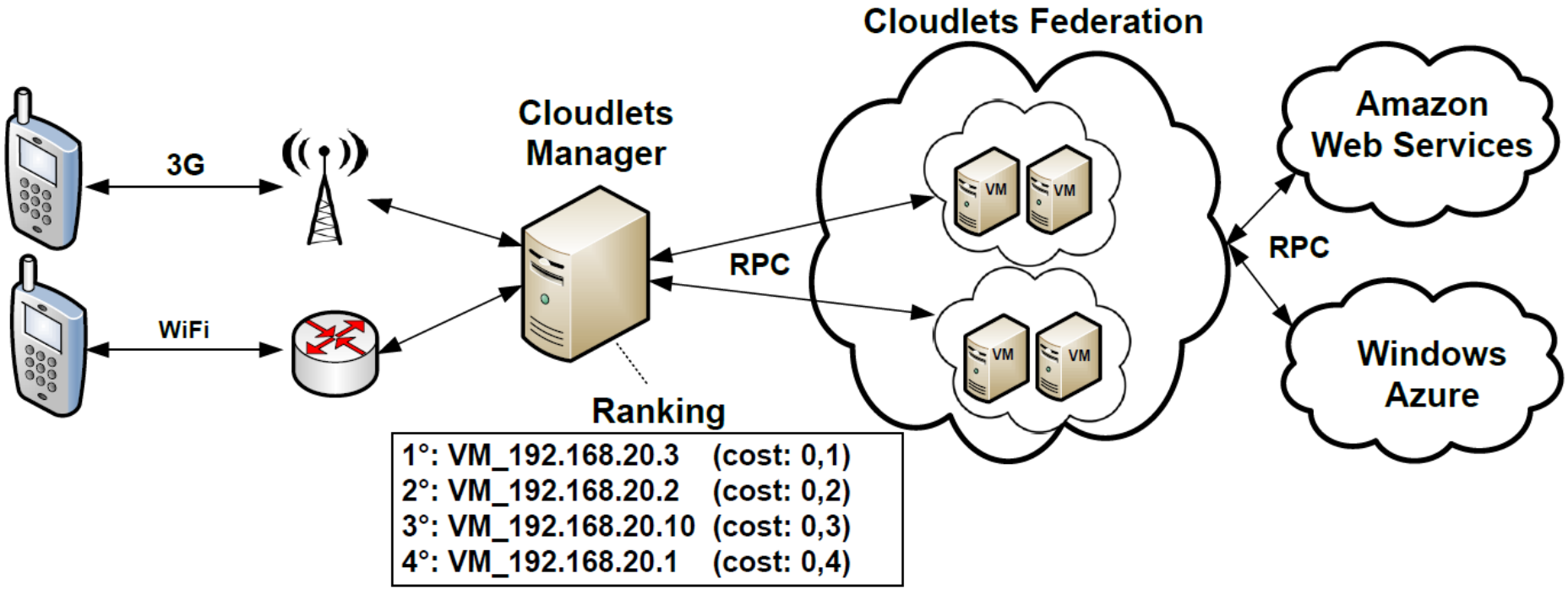
Arquiteturas centralizadas



- Arquitetura de três divisões:
  - Cliente
  - Servidor
  - Servidor que pode agir como cliente

Arquiteturas centralizadas

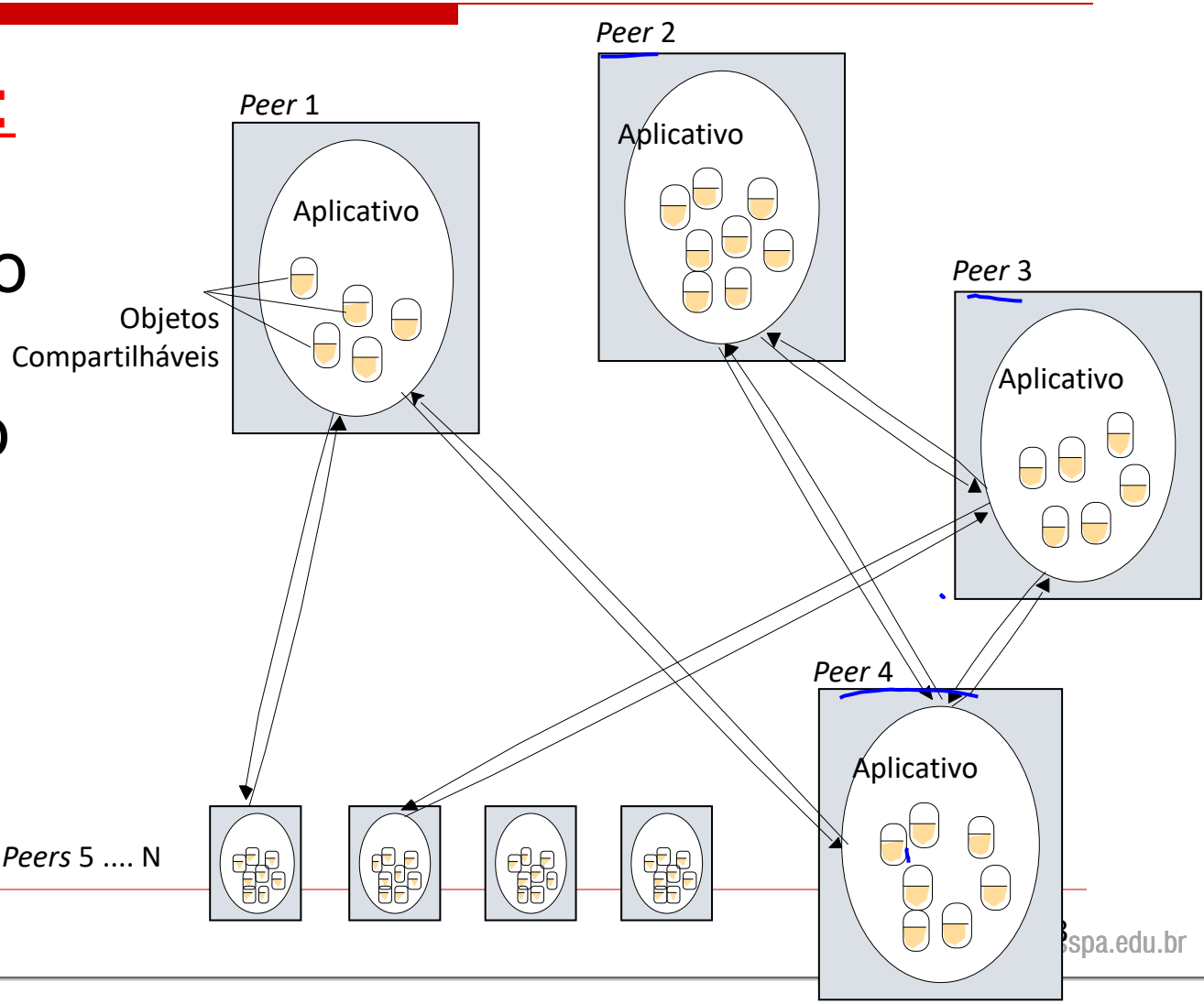
- Exemplo de arquitetura de três divisões:



Arquiteturas descentralizadas

- Peer-to-peer:

Processos são clientes e servidores ao mesmo tempo.



Arquiteturas descentralizadas

- Cliente-servidor possuem duas distribuições:
  - Distribuição vertical
  - Distribuição horizontal

- Rede de sobreposição: rede onde os nós são formados pelos processos e os enlaces denotam os canais de comunicação.

- Arquitetura: estruturadas, não estruturadas e hierárquicos

Arquitetura Peer-to-Peer estruturada

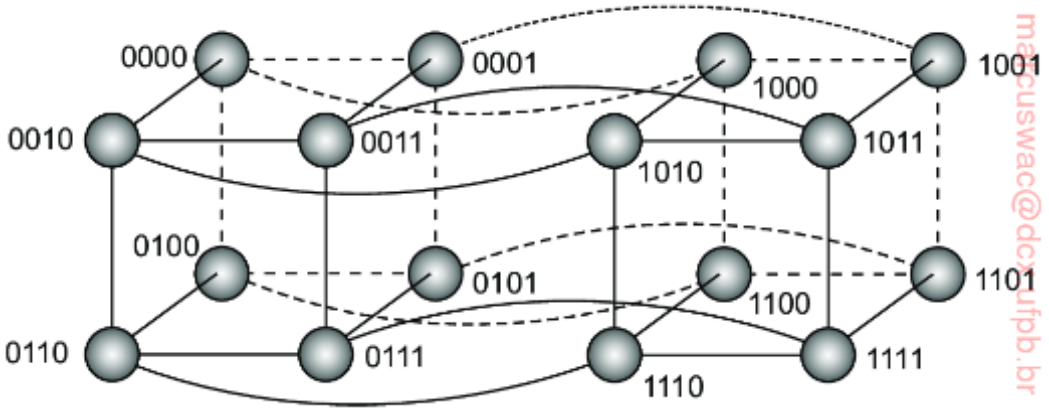
- Mecanismo usado comumente: DHT (*Distributed Hash Table*)

- Dados e nós recebem uma chave aleatória (128~160 bits).

Desafio: dada uma chave de um dado, mapear para o identificador de um nó.

Arquitetura Peer-to-Peer estruturada

- Topologia é bem determinada (anel, árvore, grade, etc)
- Ex: cada *peer* tem um *id* e se comunica no máximo com 4 peers; para buscar chave *x*, repassa requisição para vizinho mais próximo



Arquitetura Peer-to-Peer estruturada

- Chord: implementação de um DHT para redes peer-to-peer.
- Cada nó recebe um identificador aleatório (semelhante a um RG)

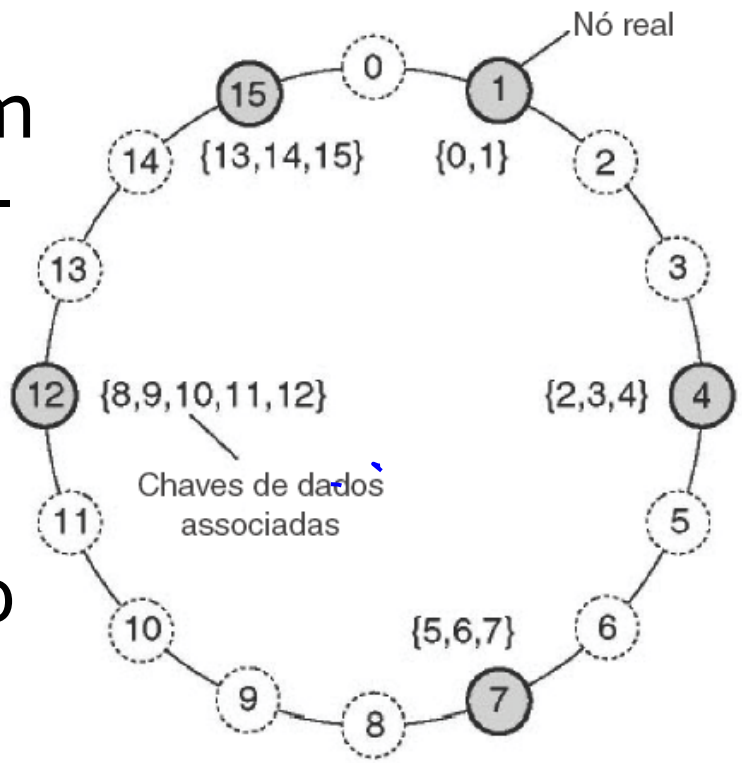
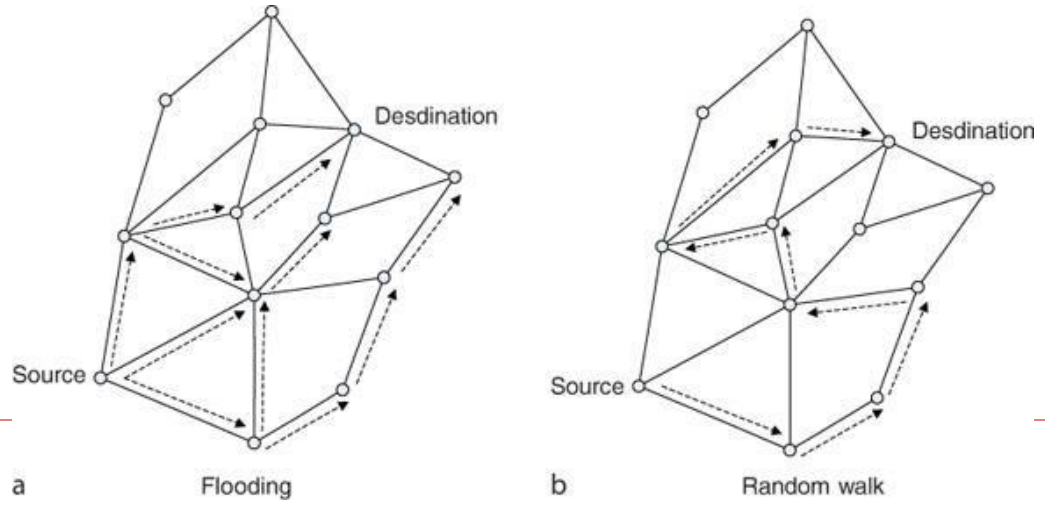


Figura 2.7 Mapeamento de itens de dados para nós em Chord.

Arquitetura Peer-to-Peer Não-estruturada

- A lista de vizinhos de cada *peer* é montada sem estrutura definida
  - Busca com ***Flooding***: repassa requisição para todos os vizinhos
  - Busca com ***Random walk***: repassa para alguns vizinhos aleatórios



Arquitetura Peer-to-Peer Não-estruturada

- Cada nó mantém uma lista de nós vizinhos.
- Dados são armazenados aleatoriamente.
- Como realizar a busca?
  - Inunda toda a rede. Tempestade de *broadcast.*

Arquitetura Peer-to-Peer: Hierárquicos (ou Superpares)

- A medida que a rede cresce, localizar itens de dados em sistemas P2P não estruturados pode ser problemático.
- Sempre que um nó comum se junta a rede, se liga a um dos superpares.
- Problema: Seleção do líder.

Arquitetura Peer-to-Peer: Hierárquicos (ou Superpares)

- Os superpares devem ter vida longa e alta disponibilidade.

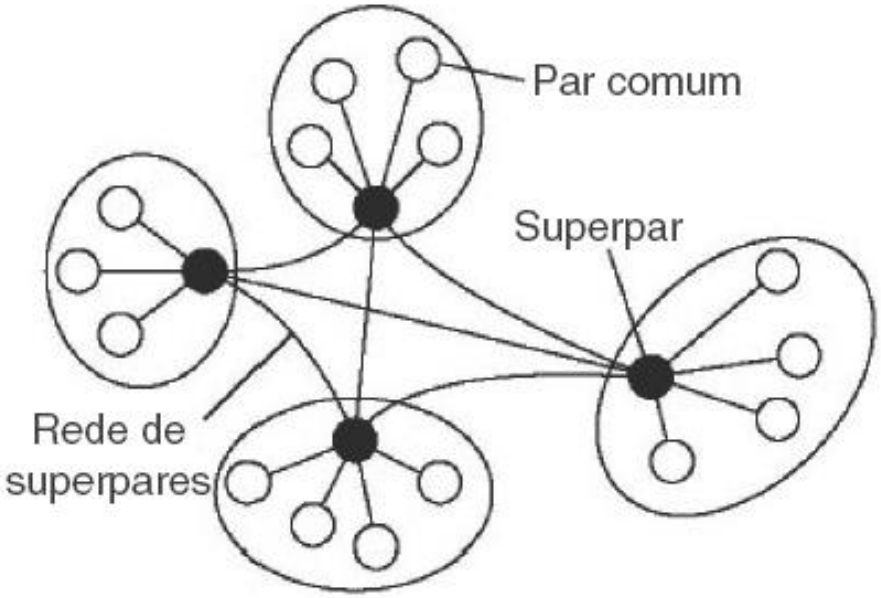


Figura 2.12 Organização hierárquica de nós em uma rede de superpares.

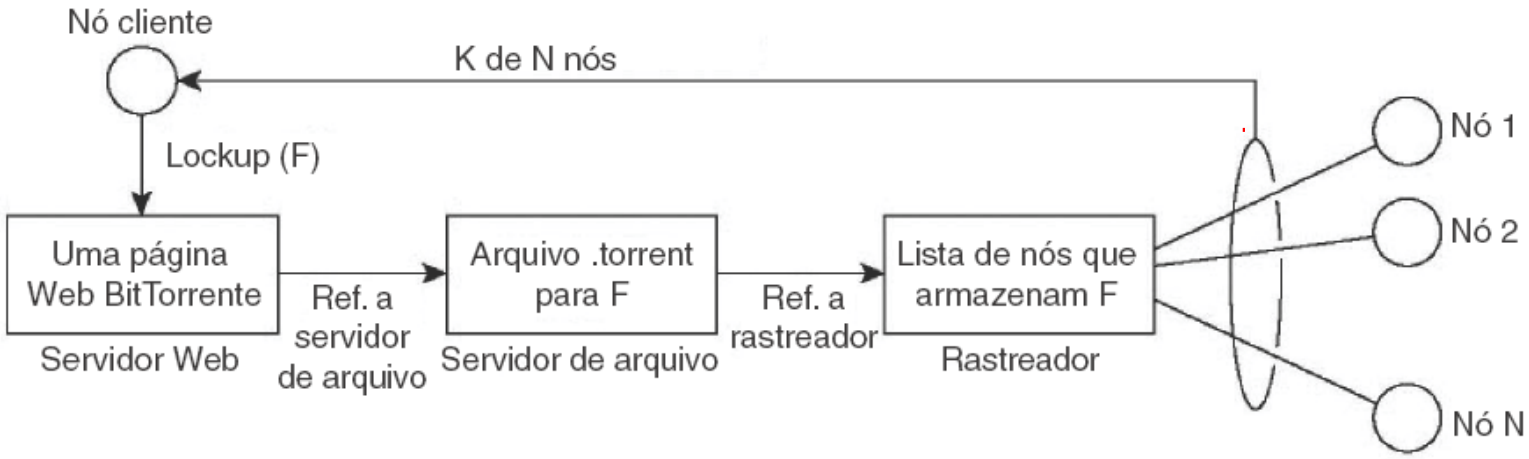
Arquiteturas híbridas

- Híbrida = centralizada + descentralizada

- Centralizada: Servir de diretório
- Descentralizada: Distribuição do conteúdo.
  - Ex.: Skype, BitTorrent, WhatsApp.

Arquiteturas híbridas

- Sistemas distribuídos colaborativos
  - Inicialmente, um esquema de procura pelo cliente-servidor tradicional.
  - Subsequentemente, o nó junta-se para dar um mecanismo descentralizado de colaboração.



Arquiteturas híbridas

- *Edge computing*: servidores são posicionados na "borda" da rede
  - Ex: limite entre a Internet e a rede de um provedor de Internet
  - Otimiza a distribuição de conteúdo e processamento
  - Dispositivos de clientes também podem fazer parte (*fog computing*)