SISTEMAS DISTRIBUÍDOS

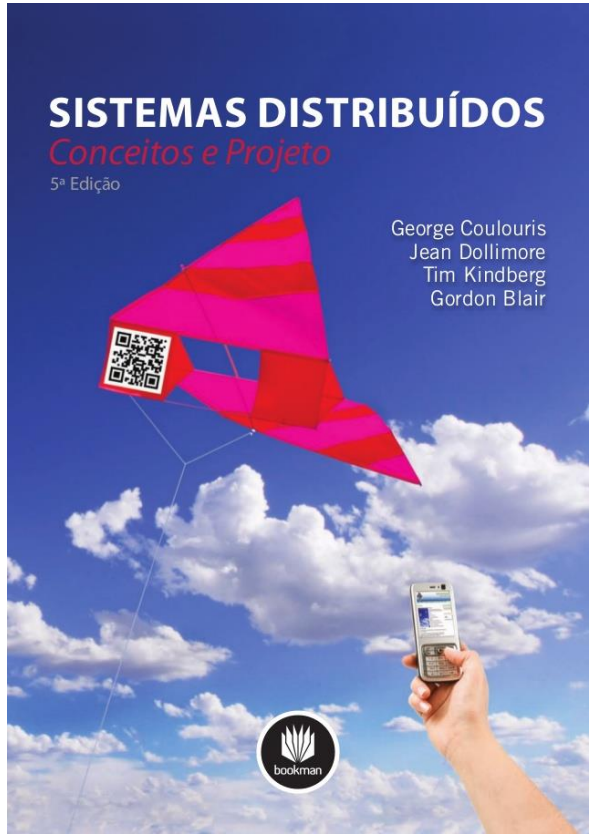
Professor: Johnatan Oliveira

Material adaptado do professor Renato

SUMÁRIO

- Modelos de sistema
  - Introdução
  - Modelos físicos
  - Modelos de arquitetura para Sistemas Distribuídos
  - Modelos fundamentais

Capítulo 2 pg 37



INTRODUÇÃO

MODELOS DE SISTEMAS DISTRIBUÍDOS

- SDs destinados ao uso em ambientes do mundo real devem ser projetados para **funcionar corretamente na maior variedade possível de circunstancias**

- **<u>Muitas dificuldades e ameaças possíveis:</u>**
  - Variados modos de uso
    - **Carga de trabalho**, conexão mal realizada, mínimo de largura de banda exigida
  - Ampla variedade de ambientes de sistema
    - **Hardware**, sistemas operacionais e redes **heterogêneas**
  - Problemas internos
    - Relógios não sincronizados, atualizações conflitantes de dados, falhas
  - Ameaças externas
    - Integridade e **sigilo** dos dados, negação de serviço

MODELOS DE SISTEMAS DISTRIBUÍDOS

- **Modelos físicos**
  - Consideram os tipos de computadores e equipamentos que constituem um sistema e sua interconectividade, sem detalhes das tecnologias específicas

- **Modelos de arquitetura**
  - Descrevem um sistema em termos das tarefas computacionais e de comunicação realizadas por seus elementos computacionais – os computadores individuais ou conjuntos deles interligados por conexões de rede apropriadas

- **Modelos fundamentais**
  - Adotam uma perspectiva abstrata para descrever soluções para os problemas individuais enfrentados pela maioria dos sistemas distribuídos

MODELOS FÍSICOS

MODELOS FÍSICOS

- Representação dos elementos de **hardware de um sistema distribuídos**, de maneira a abstrair os detalhes específicos do computador e das tecnologias de rede empregada

  - **Modelo físico básico:**
    - Conjunto extensível de nós de computadores interconectados por uma **rede de computadores para a necessária passagem de mensagem**
  - Gerações:
    - Até os anos 90: normalmente computadores de mesa, relativamente **estáticos**, **separados** e **autônomos**
    - Atualmente: aumento significativo no nível de **heterogeneidade**
      - Surgimento da **computação móvel**, ubíqua, computação na nuvem e arquitetura de **clusters**
      - Computação em **grade**

MODELOS FÍSICOS

- Sistemas distribuído de sistemas:
  - Exemplo: gerenciamento ambiental para previsão de enchentes
    - Sensores
    - Clusters
    - Alertas via celular

Sistema detecta enchentes e emite alerta para celular

Por Júlio Bernardes - jubern@usp.br

Publicado em 26/junho/2012 | Editoria : Tecnologia | Imprimir | Recommend 24

Equipamento mede nível das águas para prevenir enchentes

Salvar · 0 comentários · Imprimir · Reportar

Publicado por Governo do Estado de Minas Gerais (extraído pelo JusBrasil) - 6 anos atrás

21/07/2013 18h27 - Atualizado em 21/07/2013 18h27

Sistema criado pela USP São Carlos monitora rios para evitar enchentes

Sensor mede intensidade das chuvas e envia dados em tempo real.
Em fase de testes, tecnologia deve ajudar trabalho da Defesa Civil.

MODELOS DE ARQUITETURA PARA SISTEMAS DISTRIBUÍDOS

MODELOS DE ARQUITETURA PARA SISTEMAS DISTRIBUÍDOS

- A ***arquitetura de um sistema*** representa a estrutura desse sistema com relação aos seus diferentes componentes

- O **objetivo** é garantir que a estrutura atenda às demandas atuais e futuras

- Se **preocupando** em **manter** o sistema: confiável, gerenciável, adaptável e rentável

- Analogia: projeto arquitetônico de um prédio

  - Determina sua aparência
  - Determina sua estrutura geral
  - Fornece um padrão de referência coerente para o projeto

ELEMENTOS ARQUITETÔNICOS

ELEMENTOS ARQUITETÔNICOS

- **Elementos arquitetônicos - Quatro perguntas básicas:**
  - Quais são as entidades que estão se comunicando no SD?
  - Como elas se comunicam?
    - Qual o paradigma de comunicação utilizado?
  - Quais as funções e responsabilidades (possivelmente variáveis) que estão relacionadas a eles na arquitetura global?
  - Como eles são mapeados na infraestrutura distribuída física?
    - Qual é a sua localização?

ELEMENTOS ARQUITETÔNICOS

- **Entidades em comunicação:**

  - **Ponto de vista do sistema:**
    - Processos acoplados a paradigmas de comunicação apropriados entre processos
    - Obs: alguns sistemas primitivos os SOs talvez não suportem abstrações de processos. Assim as entidades são ***nós***.
      - Ex.: redes sensores
    - Normalmente processos são completados por **threads**
      - Portanto threads é que são os pontos extremos de comunicação

ELEMENTOS ARQUITETÔNICOS

- **Entidades em comunicação:**
  - **Ponto de vista do problema (programação):**
    - **Objetos**: unidades de decomposição naturais para o domínio do problema dado interagindo. Acessados por meio de interfaces.
    - **Componentes**: semelhantes aos objetos, mas também permite tornar todas as dependências explicitas e fornece um contrato mais completo para a construção de sistemas
      - Estimula e permite o desenvolvimento de componentes por terceiros
      - Remove dependências ocultas
    - Serviços Web: serviços intrinsicamente ligados ao WWW para **representar e descobrir serviços**
      - Parcialmente definidos pelas tecnologias Web que adotam. Ex.: XML na comunicação
      - Geralmente ultrapassam limites organizacionais, diferente dos objetos e componentes

ELEMENTOS ARQUITETÔNICOS

- Paradigmas de comunicação:
  - Como as entidades se comunicam em um SD ?
    - **Três paradigmas:**

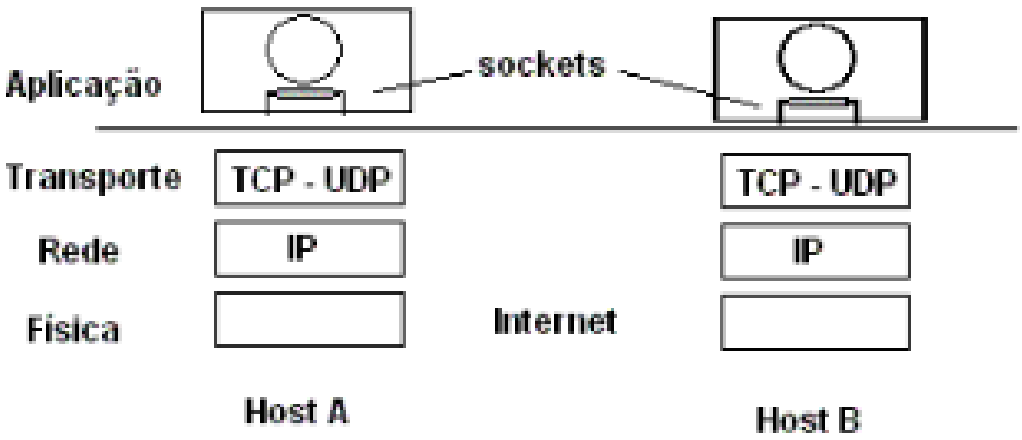
      1) Comunicação entre processo
      2) Invocação remota
      3) Comunicação indireta

ELEMENTOS ARQUITETÔNICOS

- **Paradigmas de comunicação:**

1)Comunicação entre processos

- Inclui primitivas de passagem de mensagens, acesso direto à API oferecida pelos protocolos Internet (soquetes) e também suporte para comunicação multicast



ELEMENTOS ARQUITETÔNICOS

- **Paradigmas de comunicação:**

2) Invocação remota

- Comunicação mais comum em SDs
  - Protocolos de requisição-resposta: padrão imposto em um serviço de passagem de mensagens para suportar computação cliente-servidor.
    - Troca de pares de mensagens entre cliente e servidor
    - Ex.: HTTP

  - Chamada de procedimento remoto (RPC)
  - Chamada de método remoto (RMI)

ELEMENTOS ARQUITETÔNICOS

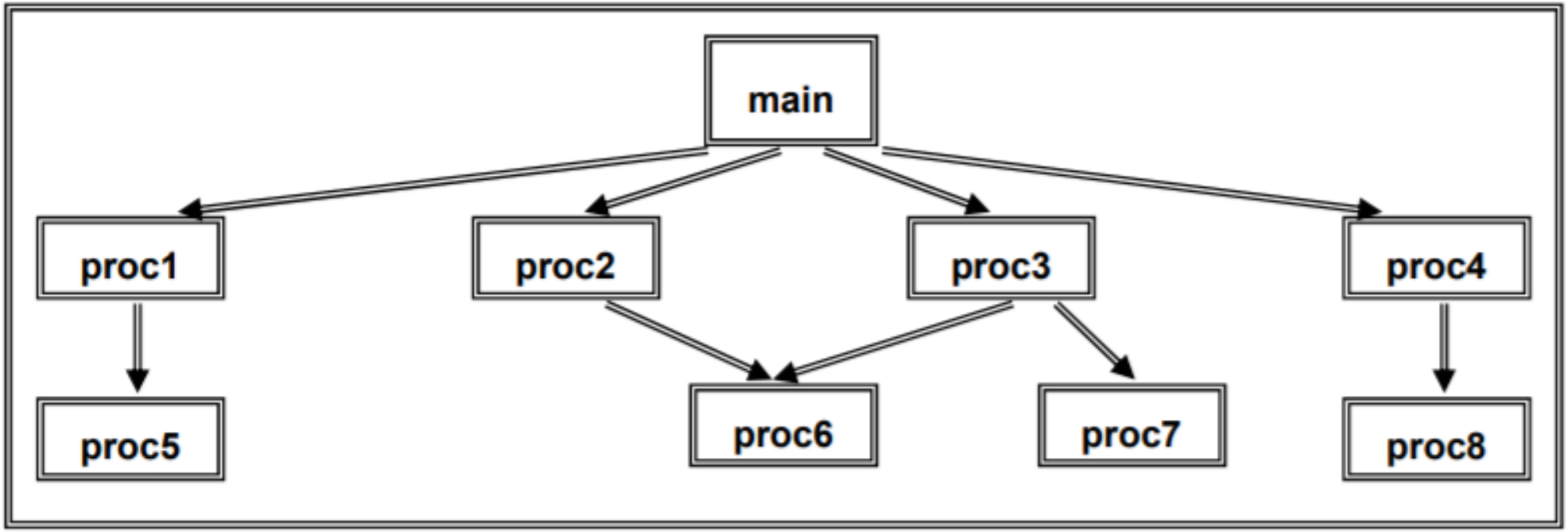
- **Paradigmas de comunicação:**

  - **Chamada de procedimento remoto (RPC)**
    - **Processos** em computadores remotos podem ser chamados como se fossem **procedimentos** no espaço de endereçamento local
    - Oculta aspectos importantes da distribuição
      - Oferecem (no mínimo) transparência de acesso e localização
    - Servidores oferecem um conjunto de operações por meio de uma interface de serviço e os cliente chamam essas operações diretamente

  - **Invocação de método remoto (RMI)**
    - Parecida com RPC, mas voltada para o mundo dos objetos distribuídos
    - Objeto chamador pode invocar um método em um objeto potencialmente remoto

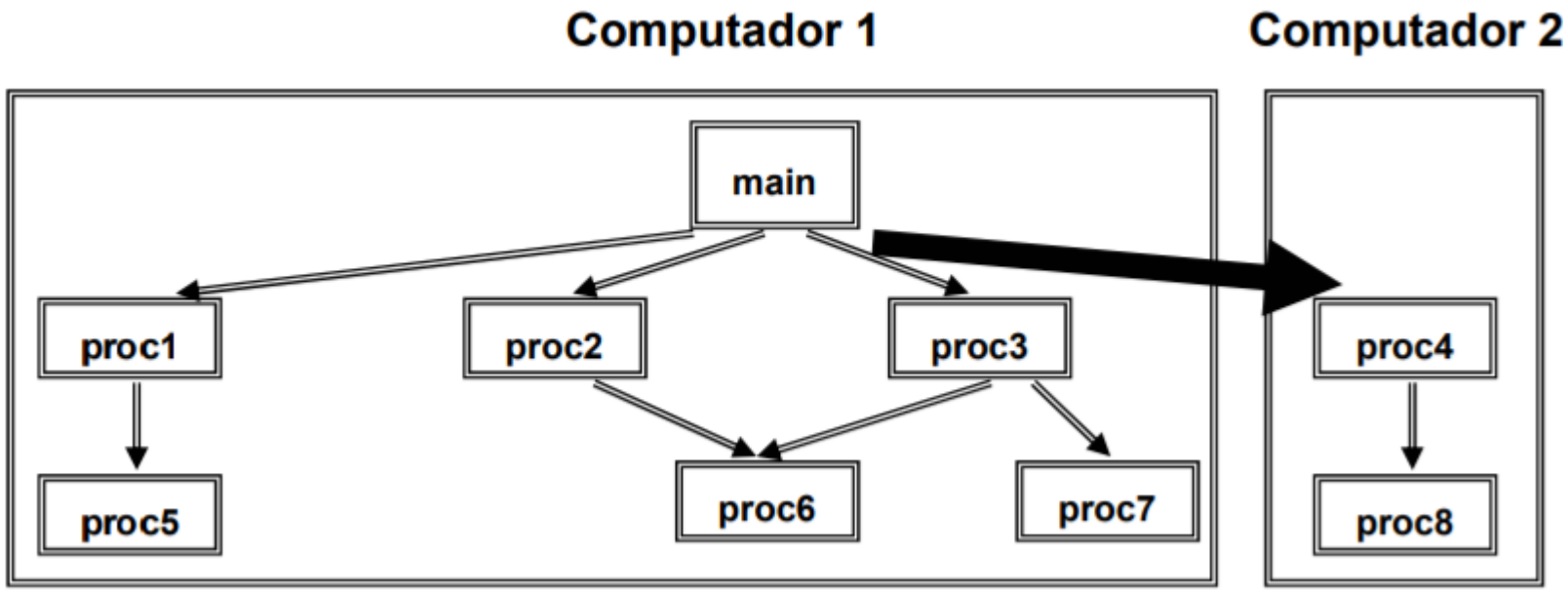
EXEMPLO DE CHAMADAS REMOTAS DE PROCEDIMENTOS

- Um programa convencional consiste de um ou mais procedimentos, geralmente organizados em uma hierarquia de chamadas.
- Uma seta de um procedimento n para um procedimento m significa uma chamada de n para m



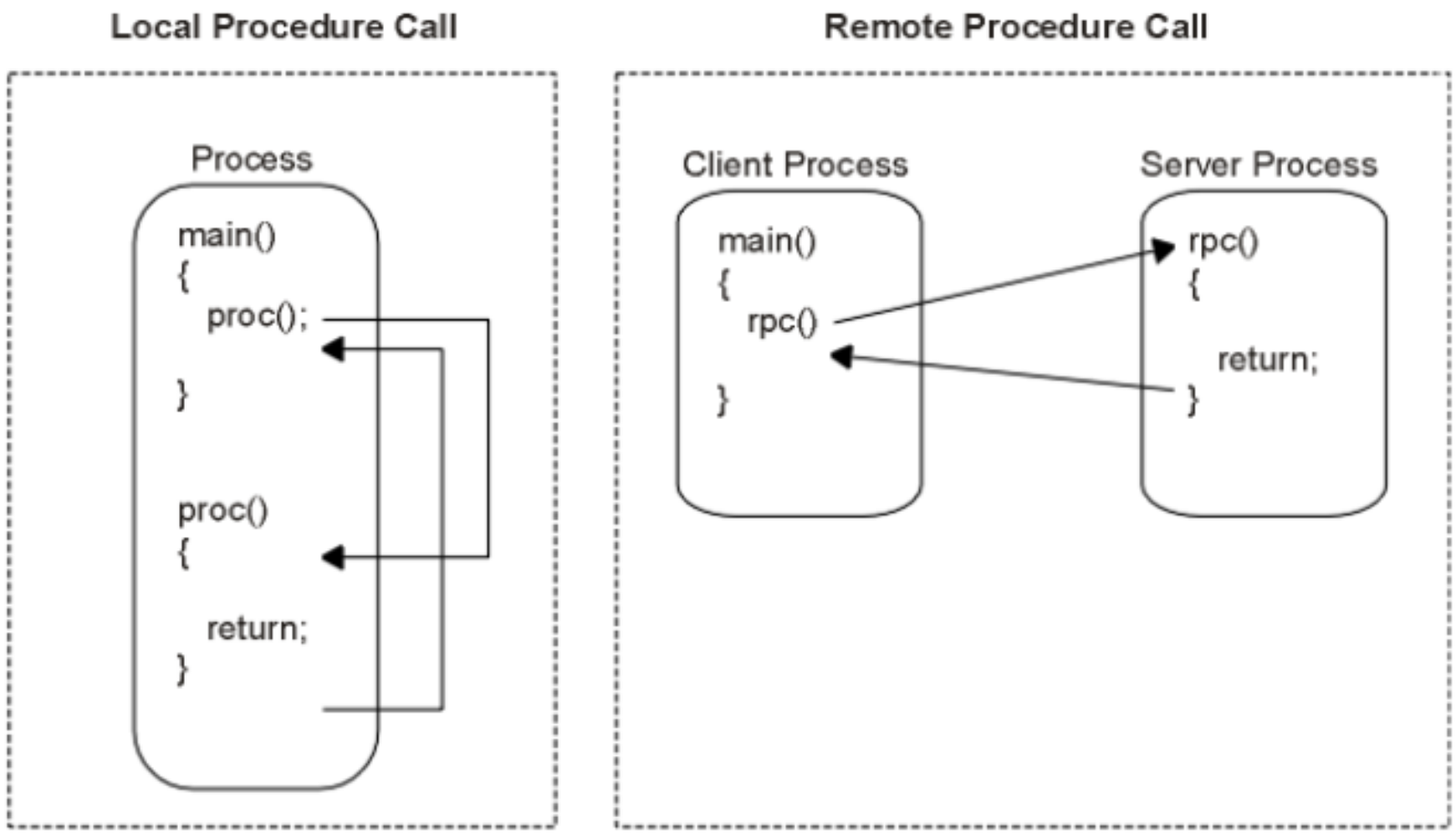
EXEMPLO DE CHAMADAS REMOTAS DE PROCEDIMENTOS

- A divisão ocorre entre o programa principal e o procedimento 4.
- Um protocolo de comunicação é necessário para implementar a chamada remota



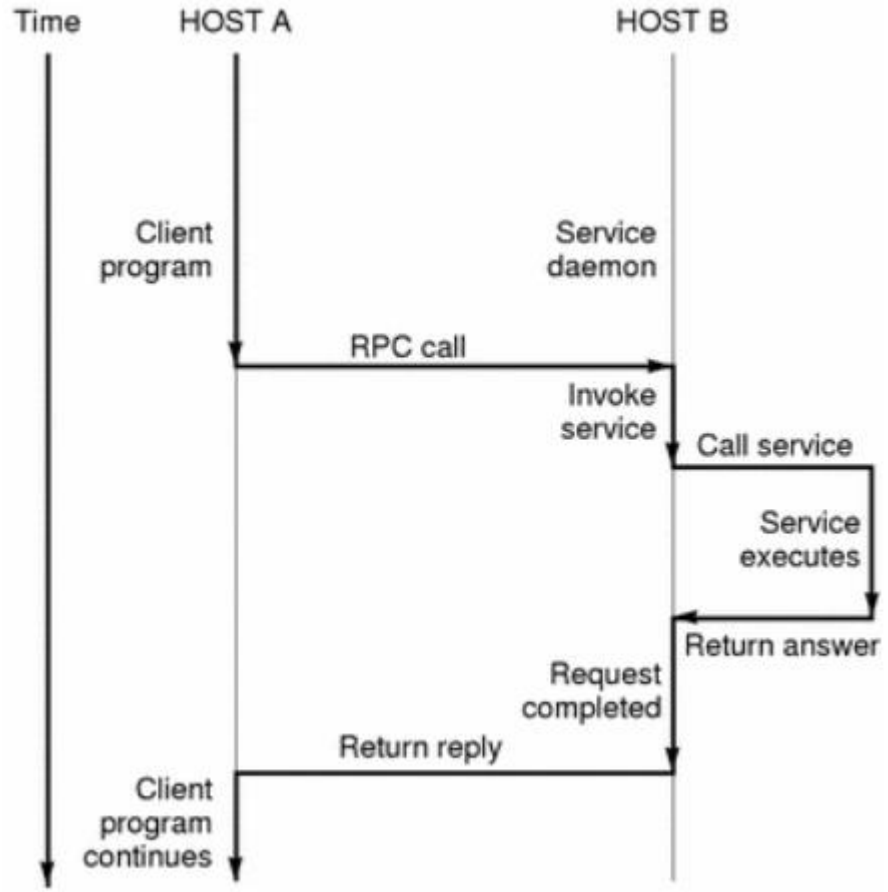
ELEMENTOS ARQUITETÔNICOS

- **Entidades em comunicação – invocação remota (RPC)**



ELEMENTOS ARQUITETÔNICOS

- **Entidades em comunicação – invocação remota (RPC)**
  - Chamada de procedimento remoto e invocação de método remoto são suportadas **pelos protocolos de requisição-resposta**



ELEMENTOS ARQUITETÔNICOS

- **Paradigmas de comunicação:**

3) Comunicação indireta:

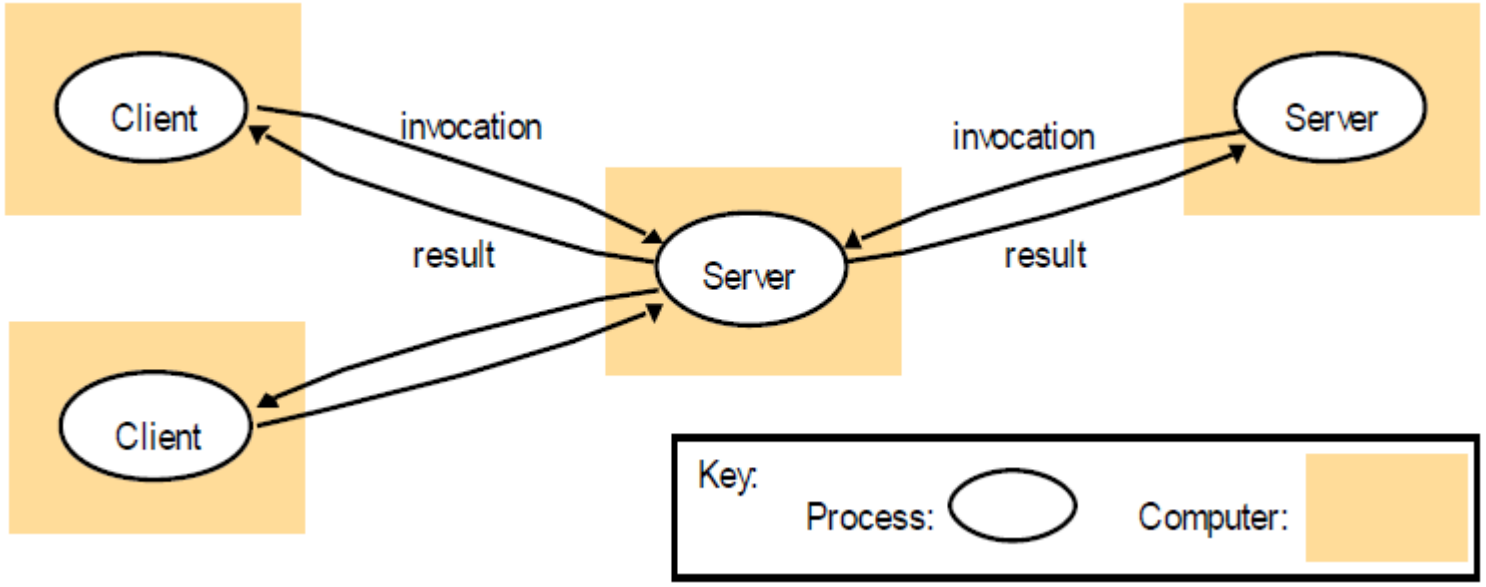
- Possibilita alto grau de desacoplamento entre remetentes e destinatários
  - Desacoplamento espacial: os remetentes não precisam saber para quem estão enviando
  - Desacoplamento temporal: os remetentes e os destinatários não precisam existir no mesmo tempo

ELEMENTOS ARQUITETÔNICOS (AQUI)

- 5 Técnicas:
  - **Comunicação em grupo:** entrega das mensagens para um conjunto de destinatários
  - **Publicar-assinar**: garante que as informações geradas pelos produtores sejam direcionadas para os consumidores que as desejam (um pra muitos)
  - **Fila de mensagens**: igual a publicar-assinar, mas oferece um serviço ponto a ponto (as pessoas aguardam as revistas na fila)
  - **Espaços de tupla**: espaço persistente (leitores e escritores não precisam existir ao mesmo tempo) que o armazenamento e consumo de estruturas denominadas tuplas
  - **Memória compartilhada distribuída**: transparência de distribuição para prover cópias de memória consistentes entre as entidades interessadas na comunicação.

ELEMENTOS ARQUITETÔNICOS

- **Funções e responsabilidades:**
  - Arquitetura **cliente-servidor**
    - Clientes requisitam serviço de um servidor
    - Exemplos: bancos de dados clássicos
    - Mecanismos de busca na Web são classificadas como cliente ou servidor?



ELEMENTOS ARQUITETÔNICOS

- **Arquitetura cliente-servidor**
  - O que fica no cliente e o que fica no servidor?

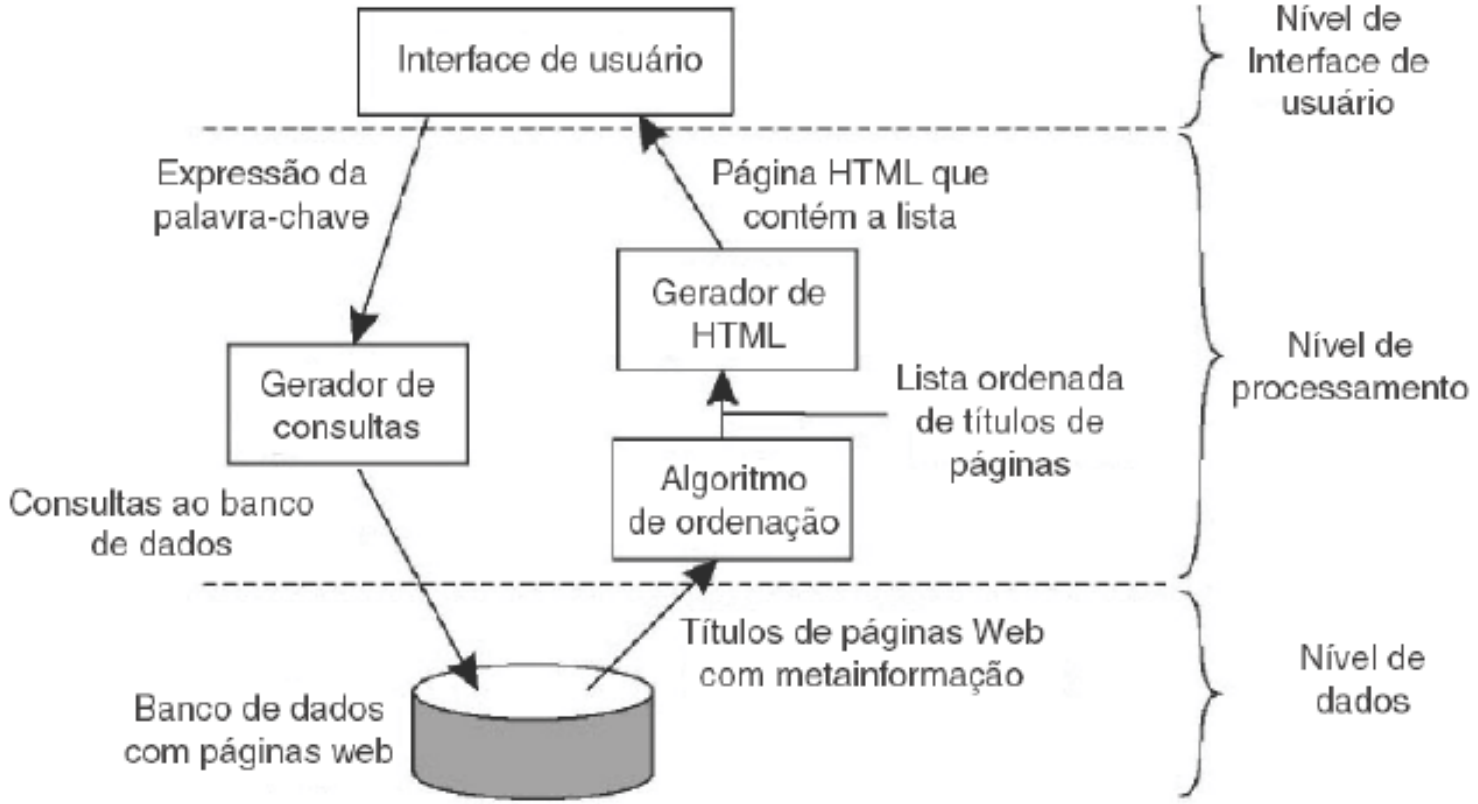


Figura 2.4 Organização simplificada de um mecanismo de busca da Internet em três camadas diferentes.

ELEMENTOS ARQUITETÔNICOS

- Arquitetura **cliente-servidor**
  - Como dividir?

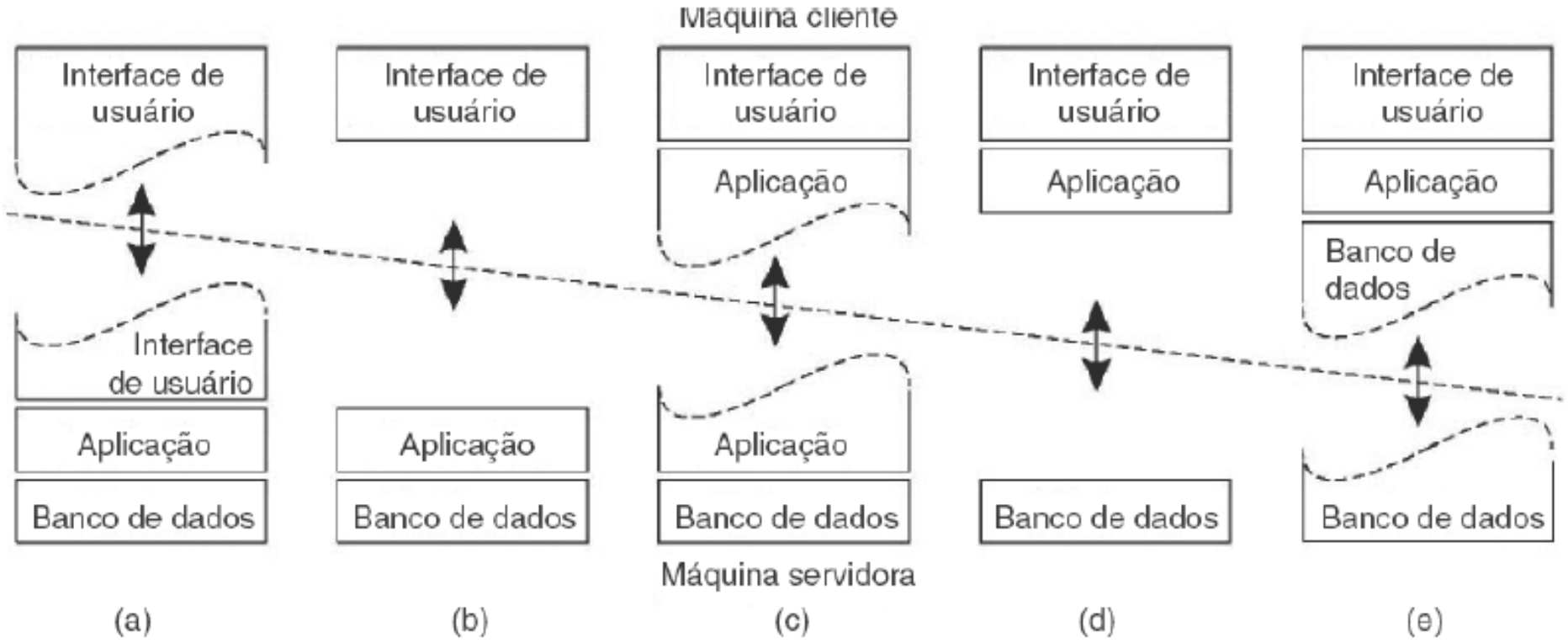


Figura 2.5 Alternativas de organizações cliente–servidor (a)—(e).

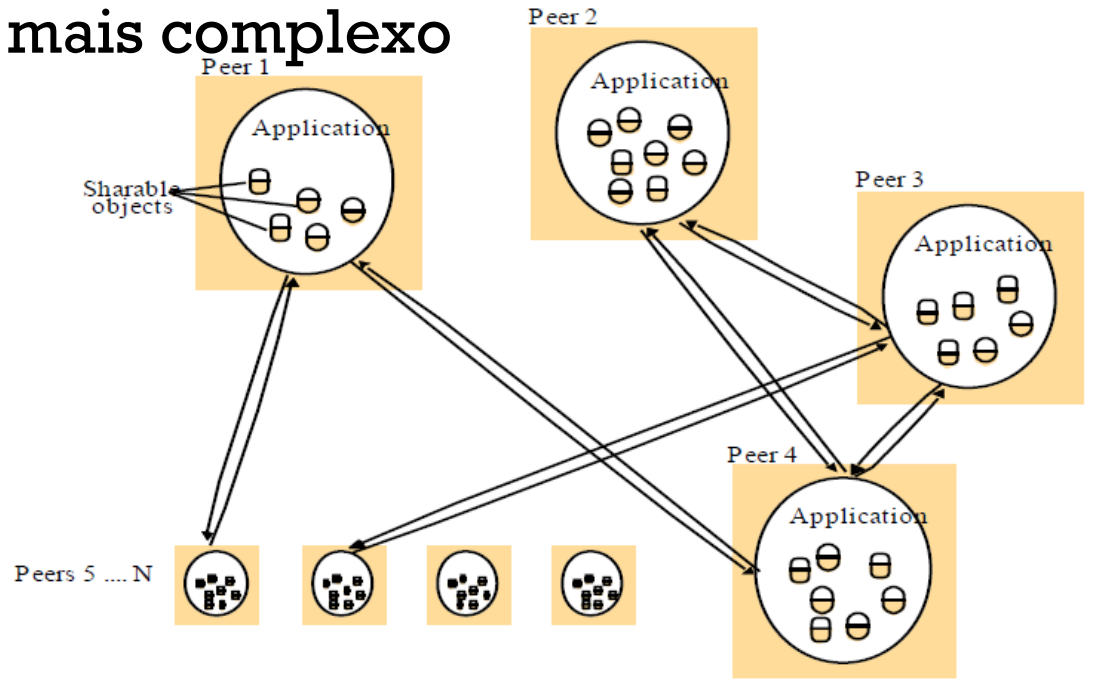
ELEMENTOS ARQUITETÔNICOS

- Arquitetura **cliente-servidor**
  - Como dividir?

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

ELEMENTOS ARQUITETÔNICOS

- **Funções e responsabilidades:**
  - Arquitetura **peer-to-peer**
    - Processos são tanto servidores quanto clientes
      - Divide cargas de computação e comunicação
      - Melhora consideravelmente a escalabilidade
      - Middleware para suportar é mais complexo



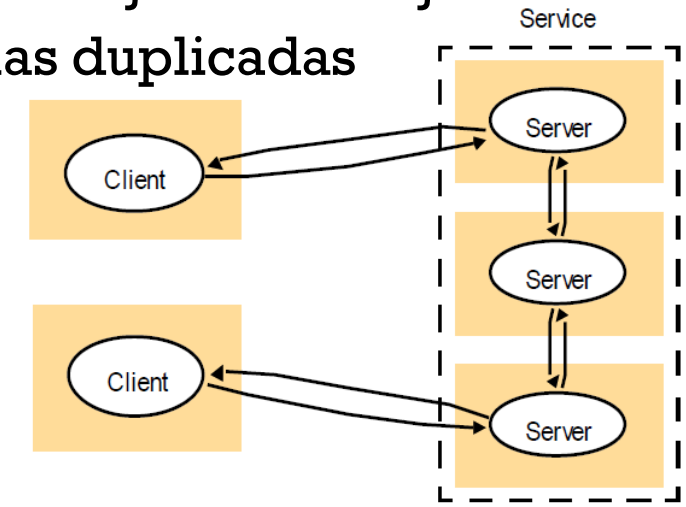
ELEMENTOS ARQUITETÔNICOS

- **Posicionamento:**
  - Modo como entidades objetos ou serviços são mapeadas na **infraestrutura física distribuída**
    - Grande número de máquinas
    - Rede complexa
  - O posicionamento precisa levar em consideração os padrões de comunicação entre as entidades, a confiabilidade das máquinas e sua carga atual e a qualidade da comunicação entre as diferentes máquinas

ELEMENTOS ARQUITETÔNICOS

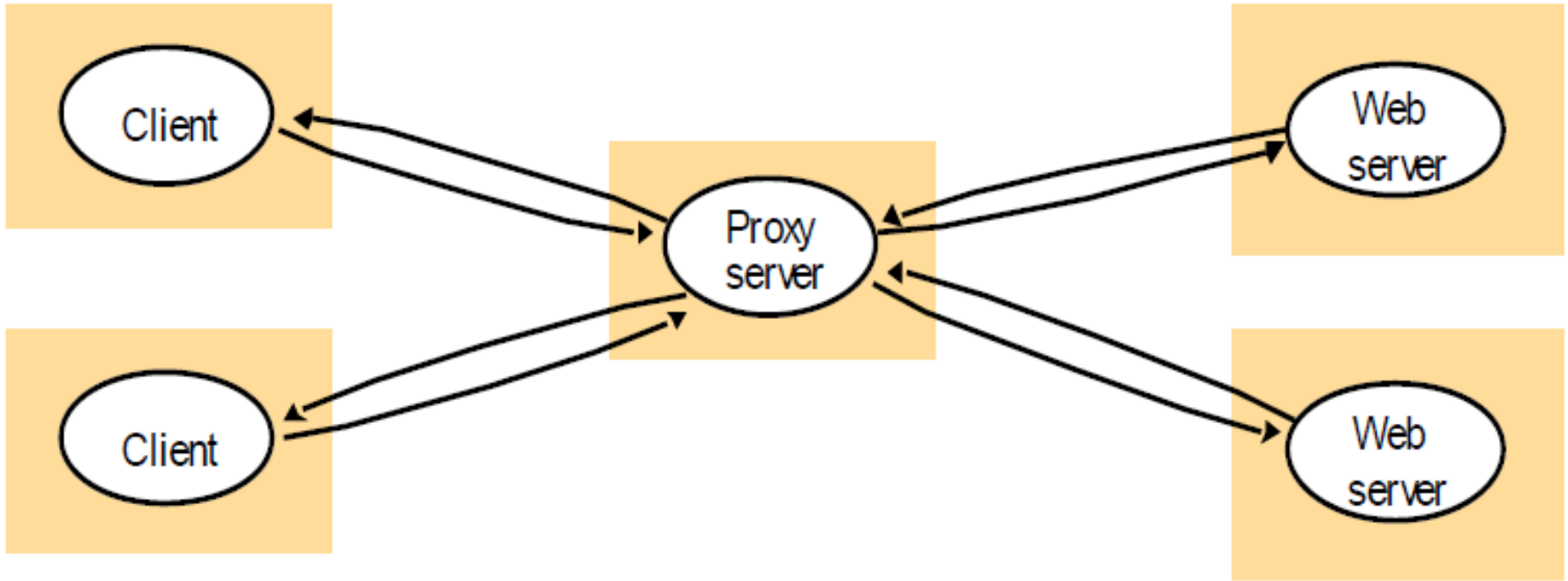
- Posicionamento:
  - Serviço provido por **múltiplos servidores**
    - Vários processos servidores em diferentes computadores hospedeiros, interagindo conforme necessário, para fornecer um serviço para processos clientes

      - Pode:
        - Particionar o conjunto de objetos
        - Manter cópias duplicadas



ELEMENTOS ARQUITETÔNICOS

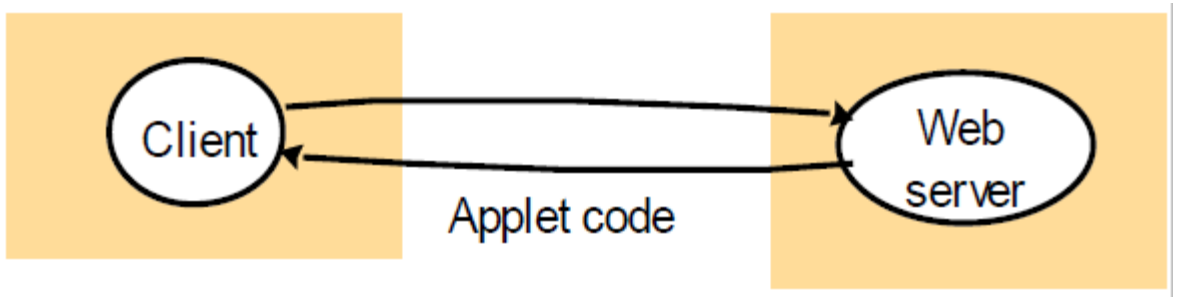
- **Posicionamento:**
  - Servidores de proxy e caches
    - Armazenam objetos recentemente usados em um local mais próximo que a origem



ELEMENTOS ARQUITETÔNICOS

- **Códigos móveis**
  - Oferece boa resposta interativa
  - O código adicional pode em alguns casos comunicar com servidores externos

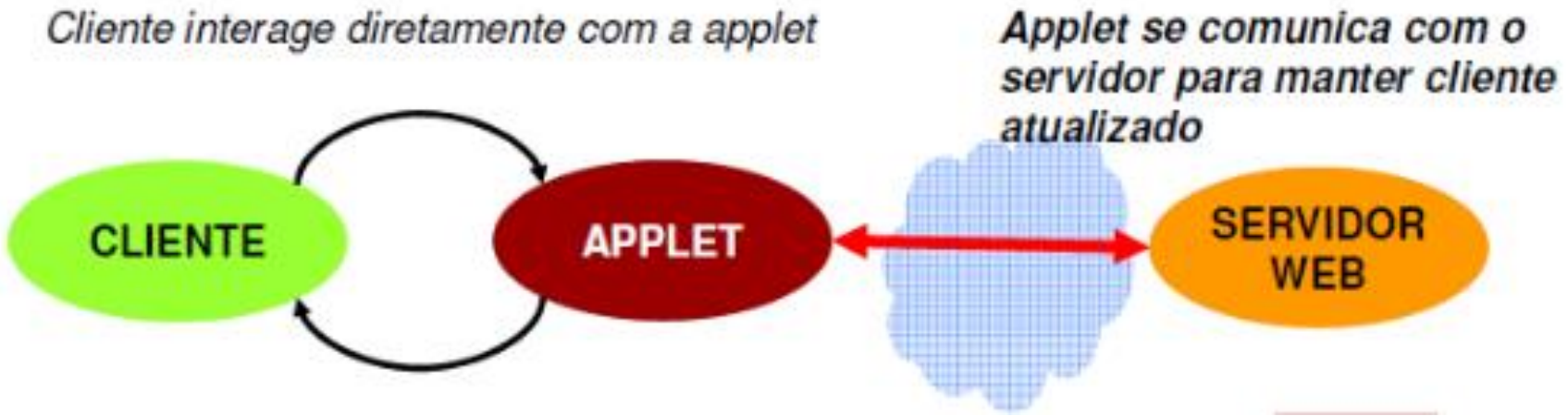
a) Requisição do cliente resulta no download do applet



b) O cliente interage o applet

ELEMENTOS ARQUITETÔNICOS

- **Posicionamento**:
  - Códigos móveis
    - Push



ELEMENTOS ARQUITETÔNICOS

- Agentes móveis
  - **Programas que percorrem uma rede, interagindo com máquinas externas, realizando uma tarefa em nome de seu usuário.**
  - Podem fazer invocações aos recursos locais em cada site que visitam.
  - Exemplos de uso: Instalação de software em computadores de uma organização; Pesquisa de preços de produtos de vendedores visitando o site de cada um e executando operações em base de dados.

PADRÕES ARQUITETÔNICOS

PADRÕES ARQUITETÔNICOS

- Os padrões arquitetônicos baseiam-se nos elementos de arquitetura mais primitivos discutidos anteriormente (entidades);

  - Os padrões não são soluções completas em si, mas oferecem ideias parciais que, quando combinadas levam o projetista a ótimas soluções;

  - Iremos discutir aqui vários padrões arquitetônicos:
    - Camadas lógicas
    - Camadas Físicas
    - Clientes "Leves"
    - Serviços Web

PADRÕES ARQUITETÔNICOS

- **1) Camadas Lógicas**

  - Um sistema complexo que é **particionado em várias camadas**, utilizando os serviços oferecidos pela camada lógica inferior;

  - Uma palavra forte nas camadas lógicas é a "**abstração**";

  - A camada lógica oferece uma **abstração de software**, com as camadas superiores
    - **Desconhecendo detalhes** da implementação ou até mesmo a **existência** das camadas lógicas inferiores.

  - Motivação: um serviço distribuído pode ser fornecido por um ou mais processos servidores que interagem entre si
    - Grande complexidade = útil organizar em camadas lógicas

PADRÕES ARQUITETÔNICOS

- Camadas Lógicas
  - **2 conceitos importantes:**
  - **Plataforma**:
    - Camadas de hardware e software de nível mais baixo.
    - Essas camadas lógicas de baixo nível promovem uma interface de programação do sistema para um nível que facilita a comunicação e a coordenação;
    - Exemplo: Intel x86/Windows
  - ***Middleware***:
    - É uma camada de software cujo o objetivo é **mascarar a heterogeneidade** e fornecer um modelo de programação conveniente para os programadores de aplicativos.
    - Promove o suporte a comunicação e compartilhamento de recursos em uma aplicação distribuída;

PADRÕES ARQUITETÔNICOS

- **Middleware**

  - Agente de **interoperabilidade** que pode ser entendido como uma camada de software que **não** é uma aplicação propriamente dita e que **não** faz parte do sistema operacional;

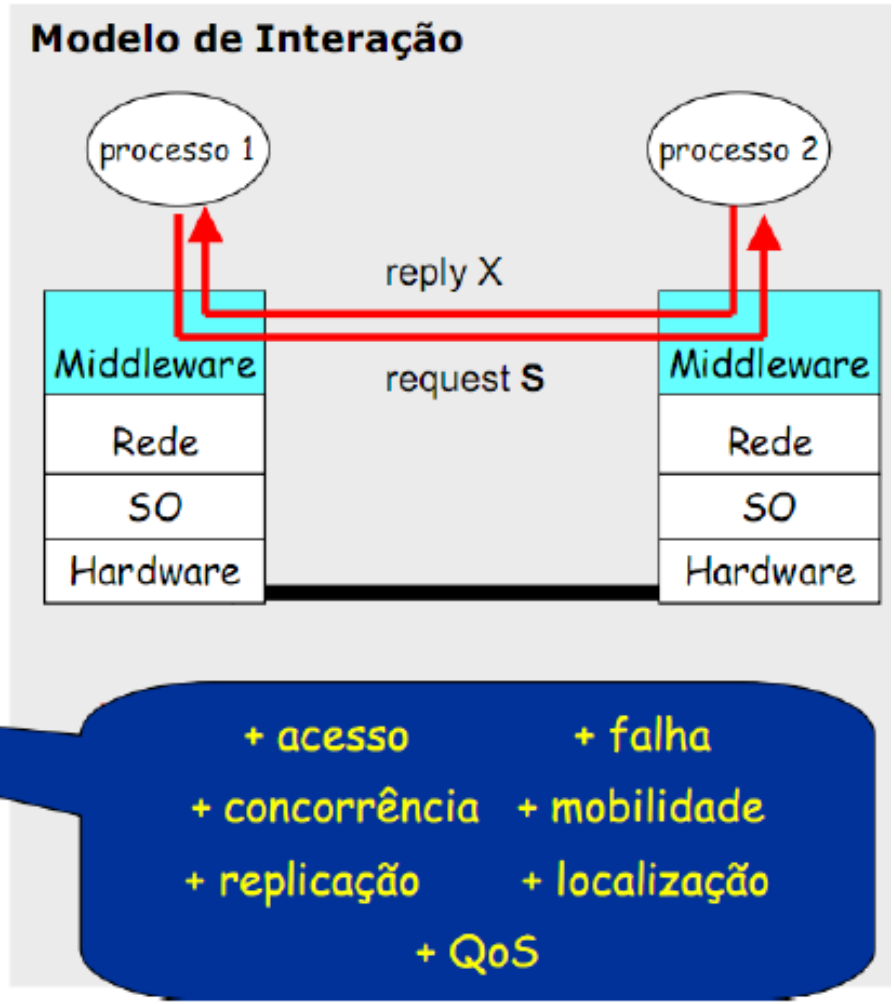
  - Esconde detalhes de dispositivos de **hardware e de software** adicional, para fornecer uma interface abstrata e mais simples de programar às aplicações;

  - Em outras palavras, o ***middleware*** simplesmente torna mais fácil a construção das aplicações na medida em o desenvolvimento pode se focar no propósito específico das aplicações.

PADRÕES ARQUITETÔNICOS

- **Middleware**

- camada de software que permite a comunicação entre aplicações (distribuídas)
- um conjunto de serviços que fornece comunicação e distribuição de forma transparente à aplicação
- componentes
  - ambiente de programação
  - ambiente de execução

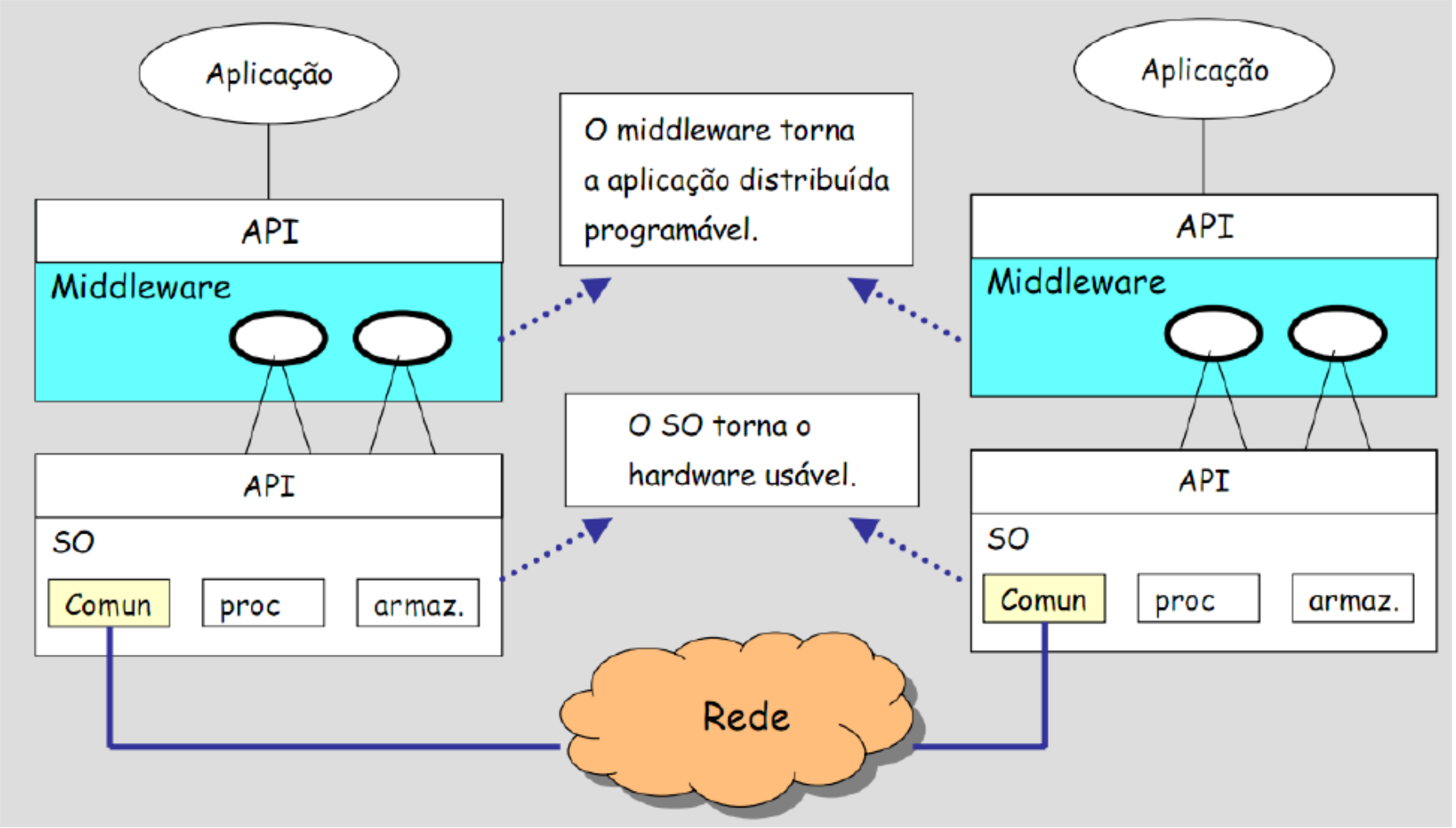


PADRÕES ARQUITETÔNICOS

- **Onde o *Middleware* se encaixa?**
  - Entre aplicações e plataformas distribuídas, com finalidade de proporcionar um grau de transparência à distribuição de dados, processamento e controle;
  - É uma camada de software posicionada entre as outras camadas de software.

PADRÕES ARQUITETÔNICOS

- Contexto do *Middleware*

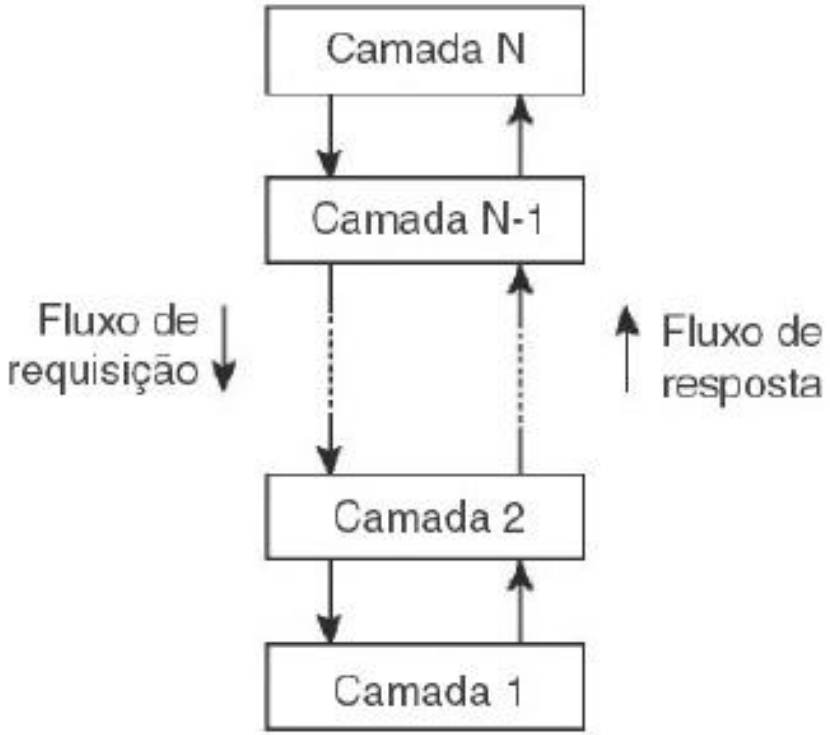


PADRÕES ARQUITETÔNICOS

- 2) Camadas Físicas
  - As arquitetura de camadas físicas são complementares às camadas lógicas;
  - A camada física organiza a funcionalidade de determinada camada lógica em um "servidor" apropriado (em um nó físico);
  - Geralmente neste aspecto surgem os conceitos de arquiteturas de **duas** e **três** camadas físicas;

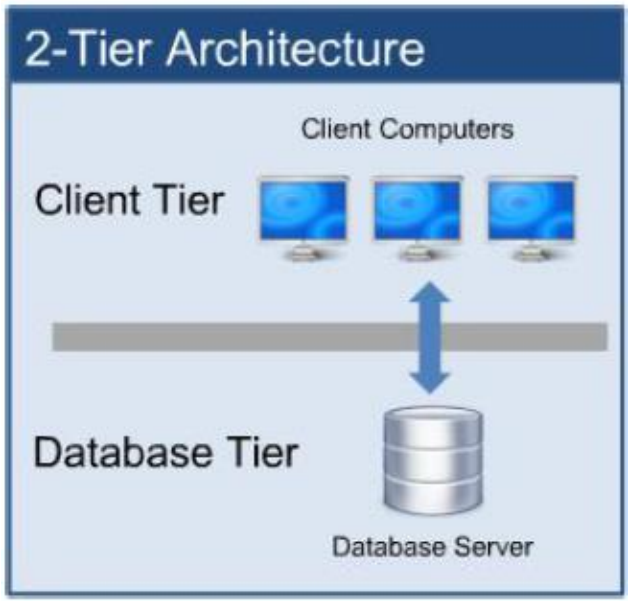
PADRÕES ARQUITETÔNICOS

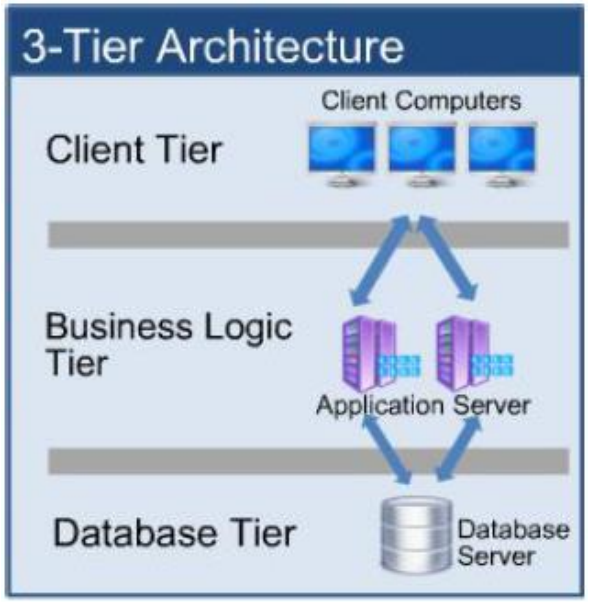
- Camadas Físicas
  - Componentes são organizados em camadas;
  - Componente da **camada N** tem permissão de chamar componentes na **camada N-1**;



PADRÕES ARQUITETÔNICOS

- Camadas Físicas

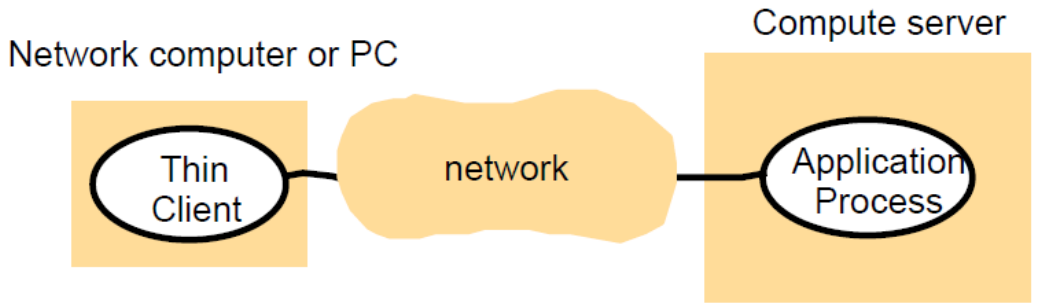




- ***Client Tier***: Modo de visualização do usuário, controles e manipulação de dados;
- ***Business Logic Tier***: Lógica da aplicação;
- ***Database Tier***: Gerenciador de banco de dados.

PADRÕES ARQUITETÔNICOS

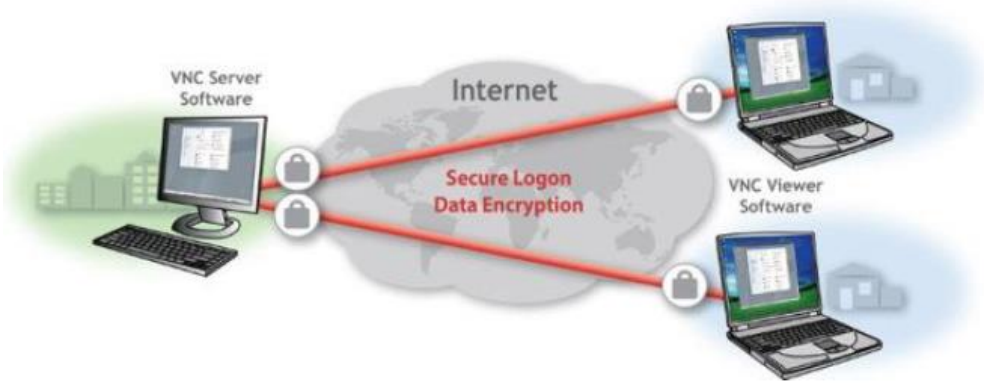
- **3) Clientes "Leves"**
  - A tendência da computação distribuída é retirar a complexidade do equipamento do usuário final e passá-la para os serviços da Internet;
    - Vejam o exemplo da computação em nuvem;
  - Desta forma um equipamento local potencialmente simples pode ser melhorado com diversos serviços e recursos interligados em rede;
  - Todos os programas executam no servidor, o cliente é usado apenas como **interface do usuário**



  - Mas e o processamento de imagens?

PADRÕES ARQUITETÔNICOS

- **Clientes "Leves"**
  - **Computação em rede virtual (VNC):**



  - Fornecer acesso remoto para interfaces gráficas do usuário;
  - Um cliente VNC interage com um servidor VNC por intermédio de um protocolo VNC;
  - Desta forma um usuário pode acessar seus recursos de computador a partir de qualquer lugar.
  - Exemplos de aplicações: RealVNC, TightVNC e UltraVNC

PADRÕES ARQUITETÔNICOS

- Outros Padrões
  - Proxy
    - um ***proxy*** é criado no espaço de endereçamento local para representar o objeto remoto. Esse ***proxy*** oferece a mesma interface do objeto remoto;
  - *Brokerage*
    - Esse padrão consiste no trio **provedor** de serviço, **solicitante** de serviço e um **corretor** de serviço-*broker (combina os serviços solicitados*) ;
  - Reflexão
    - Suporte a **introspecção** (descoberta dinâmica de propriedades do sistema) e **intercessão** (capacidade de modificar a estrutura ou comportamento dinamicamente).
  - ...

MODELOS FUNDAMENTAIS DE SISTEMAS DISTRIBUÍDOS

MODELOS FUNDAMENTAIS

- Um modelo contém apenas o essencial para que possamos entender e raciocinar a respeito de aspectos do comportamento de um sistema

  - Propriedades fundamentais:
    - Processos que se comunicam enviando mensagens por meio de uma rede de computadores
    - Características de projeto
    - Características de desempenho
    - Confiabilidade dos processos e das redes
    - Segurança dos recursos presentes no sistema

MODELOS FUNDAMENTAIS

- Objetivos:
  - Tornar explícitas todas as suposições relevantes sobre os sistemas que estamos modelando
  - Fazer generalizações a respeito do que é possível ou impossível, dadas essas suposições

MODELOS FUNDAMENTAIS: ASPECTOS CONSIDERADOS

- **Aspectos observados nos modelos fundamentais:**
- **Interação***:*
  - Deve refletir o fato de que a comunicação ocorre com atrasos, que, frequentemente, têm duração considerável
- **Falha***:*
  - Define e classifica as formas pelas quais o sistema pode falhar
    - Computador, software e rede
- **Segurança**:
  - Define e classifica as formas de ataque que podem comprometer o sistema

MODELO DE INTERAÇÃO

- Sistemas distribuídos são compostos por vários processos que interagem entre si.
- **Exemplos**
  - Vários processos servidores podem cooperar entre si para fornecer um serviço
    - Ex.: **DNS**
  - Um conjunto de processos peer-to-peer pode cooperar entre si para atingir um objetivo comum

ALGORITMOS DISTRIBUÍDOS

- Um algoritmo distribuído é uma definição dos passos a serem dados por cada um dos processos que compõem o sistema incluindo a transmissão de mensagens entre eles.
- **Dificuldades**:
  - Prever a velocidade de execução de cada processo e a sincronização das mensagens trocadas entre eles
  - Descrever todos os estados do algoritmo, dadas as falhas podem ocorrer em processos ou mensagens
- **Fatores que afetam a interação dos processos:**
  - Desempenho da comunicação
  - Impossibilidade de manter uma noção global de tempo única

FATORES QUE AFETAM A INTERAÇÃO

- **Desempenho dos canais de comunicação**
  - Latência: tempo entre o início da transmissão de uma mensagem do processo de origem até o início da recepção pelo processo de destino
  - Largura de banda: volume total de dados que podem ser transmitidos em um certo tempo
  - *Jitter*: variação no tempo de transmissão de uma série de mensagens

FATORES QUE AFETAM A INTERAÇÃO

- **Relógios e temporização de eventos**
  - Cada computador possui seu próprio relógio
  - Processos em máquinas diferentes podem associar tempos diferentes aos seus eventos, mesmo lendo seus relógios ao mesmo tempo
  - *drift* ou taxa de desvio do tempo real faz os relógios divergirem
    - Drift se refere à quantidade relativa pela qual um relógio de computador se difere de um relógio em referência perfeito
  - Como saber qual evento aconteceu primeiro em um sistema distribuído?

MODELOS DE INTERAÇÃO

- **Sistemas distribuídos síncronos**
  - Possui suposições fortes a respeito do tempo de interação
    - **Tempo superior e inferior para executar cada etapa**
    - Tempo para receber mensagem dentro de um canal
    - Taxa de desvio de tempo real tem um valor máximo conhecido
  - Torna possível fazer estimativas
    - Porém, é muito difícil chegar a valores realistas e dar garantias dos valores escolhidos, tornando o projeto menos confiável
- **Sistemas distribuídos assíncronos**
  - Não faz suposições de tempo de interação
    - Ex.: Internet

MODELO DE FALHAS

- Em um sistema distribuído, tanto os processos como os canais de comunicação podem falhar
  - Falha: sair do comportamento esperado
- Tipos de falha:
  - Omissão
  - Tempo (sincronização)
  - Arbitrária

FALHAS POR OMISSÃO

- **Deixa de executar uma ação que deveria**
- Em processos:
  - *Crash*: outros não notam
  - *Fail-stop*: perceptível
- Em canais de comunicação:
  - Mensagem enviada não chega ao destino
- Ambas são consideradas falhas "benignas" e são as mais frequentes

FALHAS ARBITRÁRIAS (OU BIZANTINAS)

- **Qualquer tipo de erro pode ocorrer**
  - Ex: processos respondem com **valores incorretos**
- Em processos:
  - Processo omite passos esperados do processamento ou realiza **passos indesejados**
- Em canais de comunicação:
  - Mensagens corrompidas, envio de mensagens inexistentes, vários envios da mesma mensagem

FALHAS DE TEMPO

- Acontecem quando limites de tempo são desrespeitados em **sistemas síncronos**
  - Falha de relógio: o relógio local do processo ultrapassa os limites de sua taxa de desvio em relação ao tempo real
  - Falha de desempenho (processo): o processo ultrapassa os limites de tempo entre duas etapas
  - Falha de desempenho (canal): a transmissão de uma mensagem demora mais do que o limite definido

MODELO DE SEGURANÇA

- A segurança de um SD pode ser obtida tornando seguros os processos e os canais usados por suas interações e **protegendo contra acesso não autorizado os objetos que encapsulam**

- Normalmente modelamos um adversário(ou inimigo), suas capacidades e seus recursos

- A partir dele, fazemos um modelo de ameaças

AMEAÇA AOS PROCESSOS

- O adversário é capaz de fazer requisições não-autorizadas
  - Serviços sem senha

- É necessário que o serviço seja capaz de verificar a *identidade* do requisitante

- E o mesmo vale se o adversário puder se passar pelo *servidor*
  - O cliente deve conseguir verificar a identidade do servidor

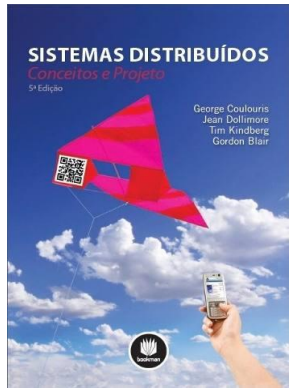
AMEAÇA À INTERAÇÃO DOS PROCESSOS

- Ataques possíveis para um adversário:
  - Se passar por outro processo e enviar uma mensagem
  - Escutar mensagens nos canais de comunicação e alterá-las, omiti-las ou repeti-las
  - Enviar tantas requisições a um servidor que o paralisa (*Denial of Service*)
- **De quais desses ataques nosso adversário é capaz?**

LIDANDO COM AMEAÇAS

- Criptografia: ciência de manter mensagens seguras
  - Criptografar =embaralhar de forma a só ser desembaralhado por quem conhece uma chave
  - É possível perceber se alguém alterou uma mensagem criptografada
- Autenticação: prova uma identidade usando criptografia
- Autorização: define que identidades podem fazer o que
- Com autenticação + criptografia temos canais seguros:
  - Partes sabem a identidade do outro lado do canal
  - Esses canais proveem confidencialidade e integridade
  - TSL e SSL provê essa abstração para um desenvolvedor

LIVRO TEXTO

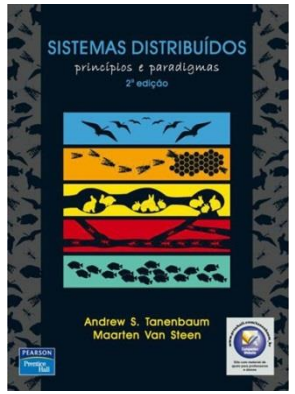


- (Coulouris, 2013)

  COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.

  Sistemas Distribuídos: Conceito e Projeto

  Artmed, 5ª edição, 2013



- (Tanenbaum, 2008)

  TANENBAUM, A. S.; STEEN, M. V.

  Sistemas Distribuídos

  Pearson, 2ª edição, 2008