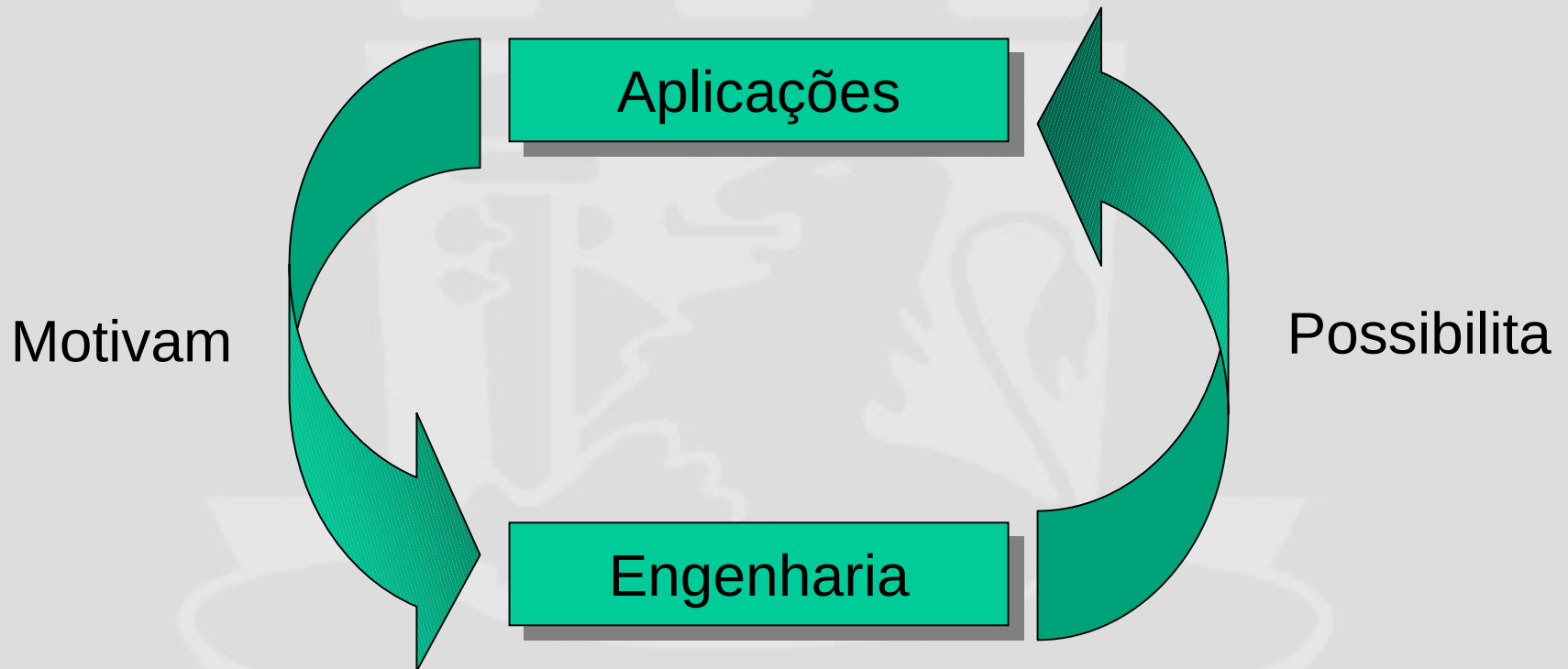




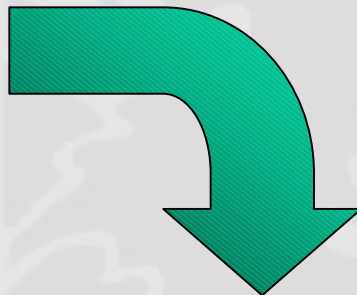
Sistemas Distribuidos

Motivação



Motivação

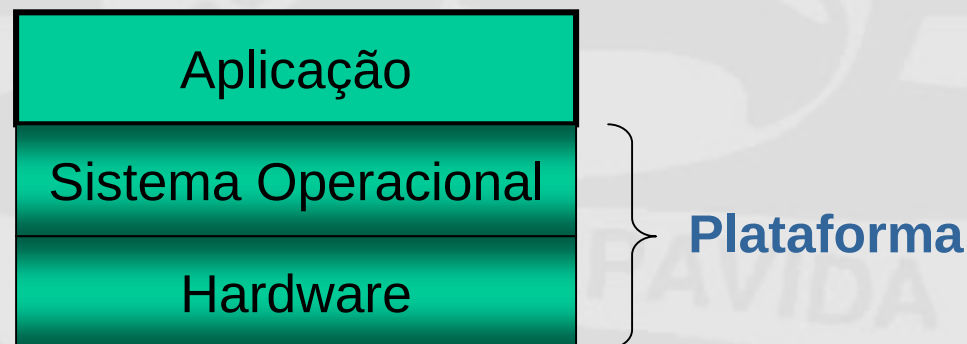
- ❑ Aplicações cada vez mais complexas
- ❑ Qual a técnica mais comum para redução de complexidade?
 - ❖ **Modularização**



Dividir para conquistar!!!

Motivação

- ❑ Modularização aumenta a **legibilidade** de um programa
- ❑ Mas, um único programa com vários módulos (**sistema monolítico**) continua complexo para a **plataforma de suporte**, principalmente porque exige muitos recursos (de processamento, memória...)



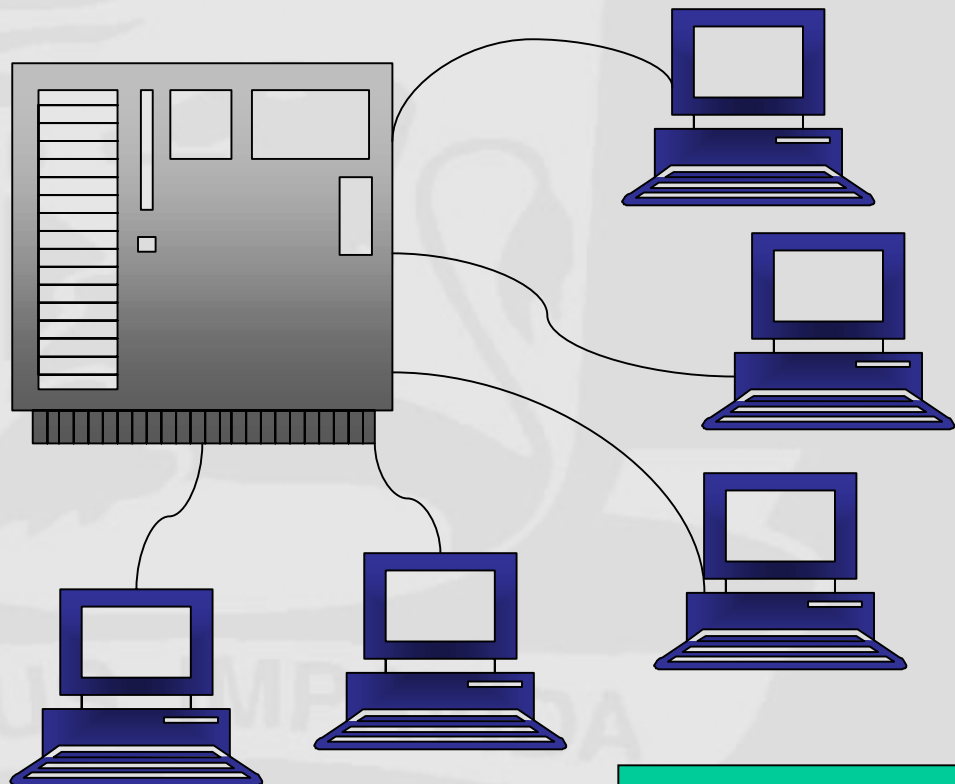
Sistemas Centralizados

Motivação

Computadores com grande capacidade de processamento (*mainframes*)

- ❖ Grande porte físico → limitações para acomodação
- ❖ Grande consumo de energia → sala especial, refrigeração
- ❖ SO único → dependência do fabricante

Terminais sem capacidade de processamento (“burros”)



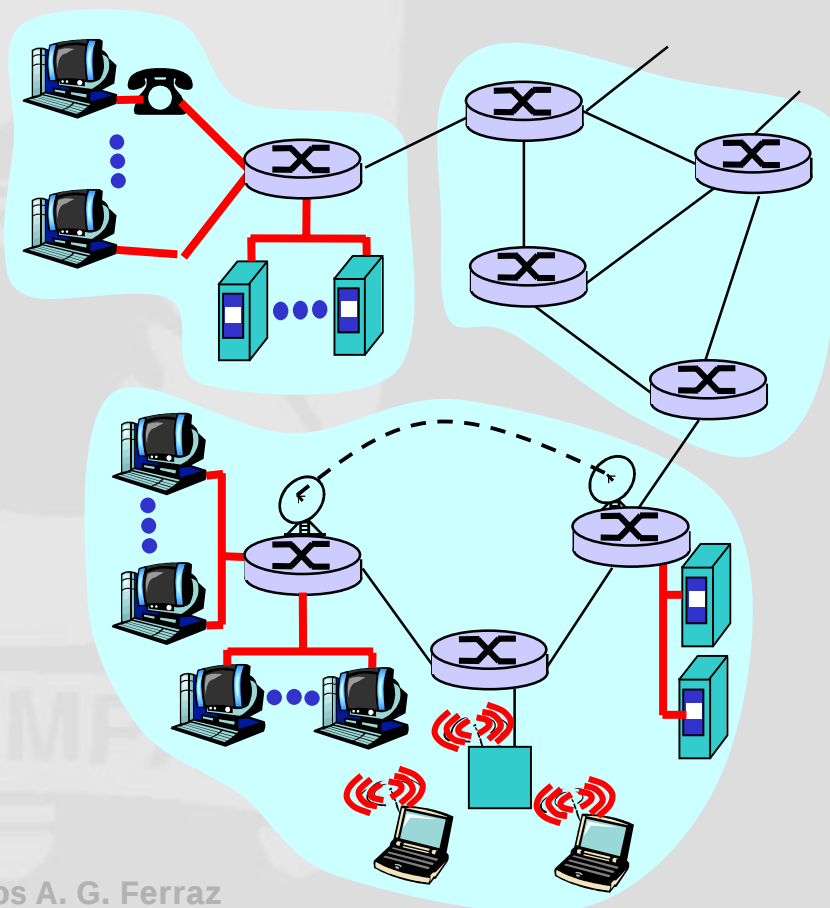
...passado

Sistemas em Rede

Motivação

A realidade dos últimos tempos...

- ❑ Computadores diversos, todos com capacidade de processamento
 - ❖ Portes diversos
 - ❖ SOs diversos
 - ❖ Redes diversas (Ethernet, ATM, com fio, sem fio...)
 - ❖ Internet

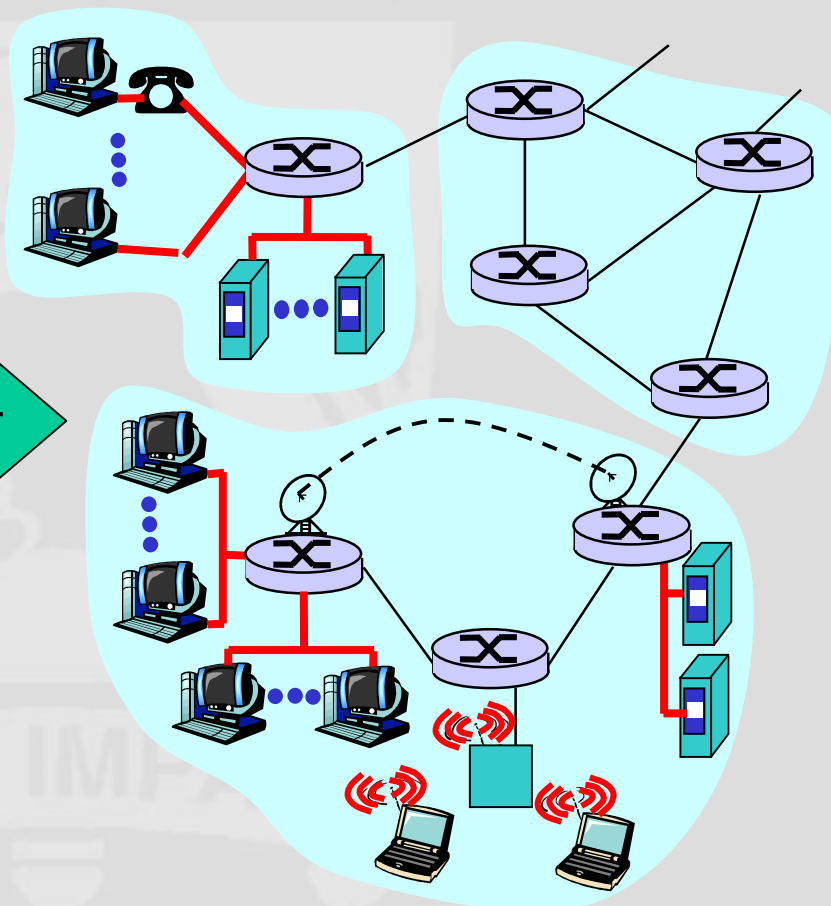
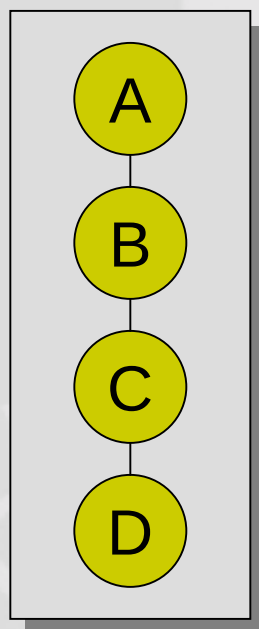


Distribuição de sistemas

Motivação

Dividindo para conquistar!!!

Programa modularizado



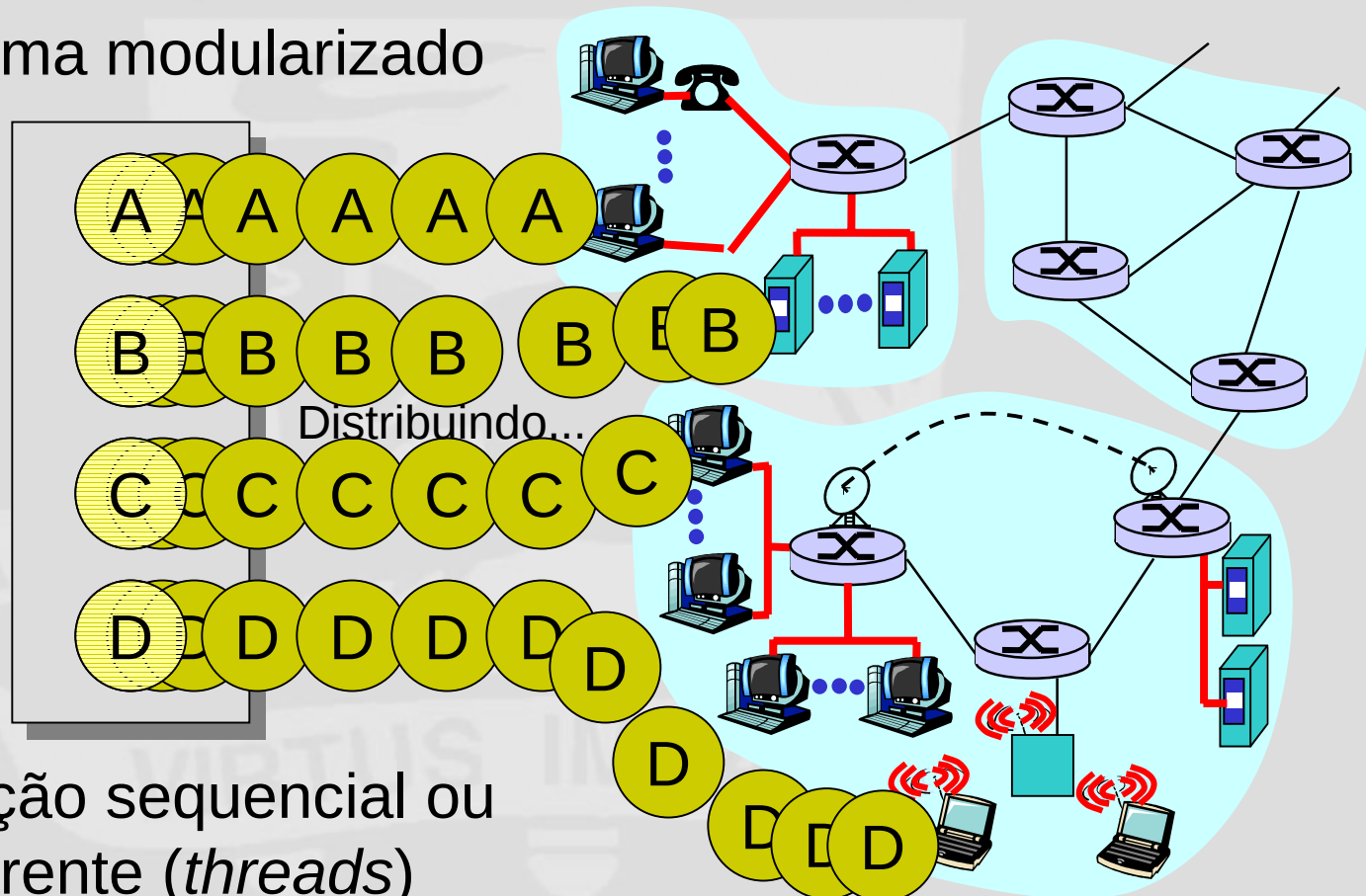
Execução sequencial ou
concorrente (*threads*)

Distribuição de sistemas

Motivação

Dividindo para conquistar!!!

Programa modularizado



Execução sequencial ou
concorrente (*threads*)

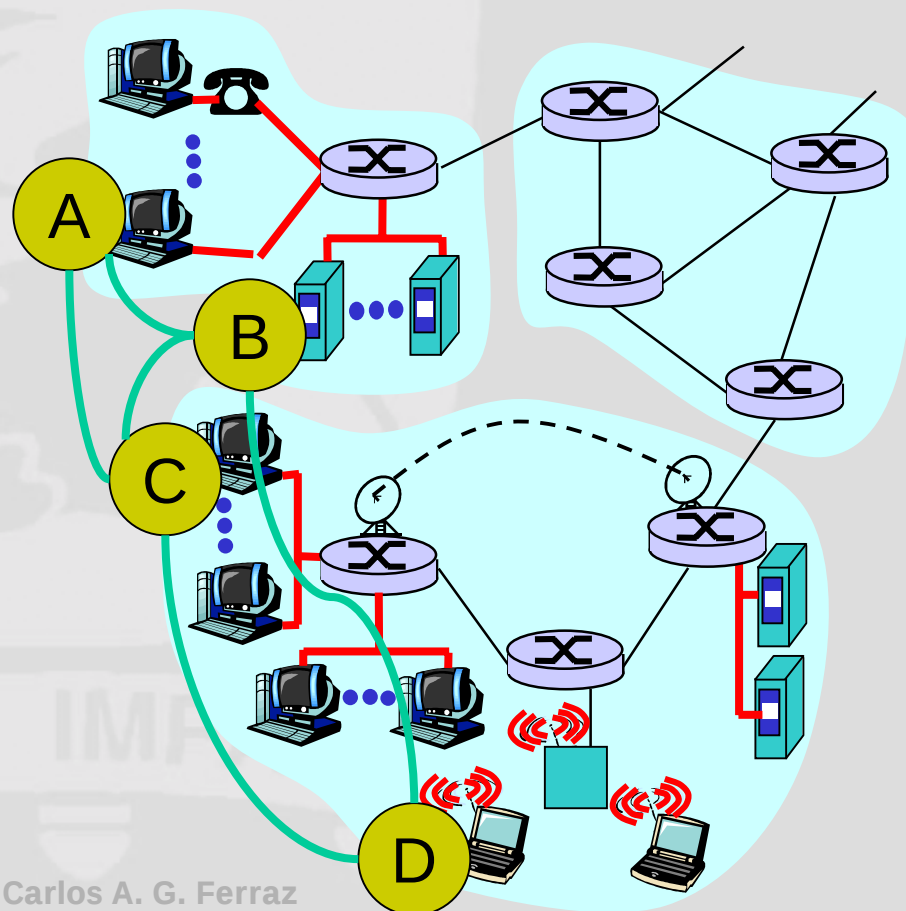
Distribuição de sistemas

Motivação

Dividindo para conquistar!!!

Programa distribuído

- ❑ Componentes interligados
- ❑ Processamento (execução) distribuído ou paralelo



Sistema Distribuído

- ☐ O que é?
- ☐ Conceitos
- ☐ Desafios



VIRTUS IMPAVIDA

O que é um Sistema Distribuído?

- ❑ “Coleção de computadores independentes que aparecem para os usuários do sistema como um único computador.” (**Tanenbaum & van Steen**)
- ❑ “Um sistema em que componentes de *hardware* e *software* localizados em computadores em rede se comunicam e coordenam suas ações por passagem de mensagens.” (**Coulouris et al**)
- ❑ “Um Sistema distribuído é uma coleção de elementos de processamento interconectados, tanto logicamente como fisicamente, para execução cooperativa de programas de aplicação com o controle geral dos recursos centralizado.” (**M. Eckhouse**)
- ❑ Vários componentes
- ❑ Conectados via uma rede
- ❑ Compartilhando recursos

Aplicabilidade

- ❑ Sistemas de software de (grande) porte
- ❑ Um sistema de sw de grande porte precisa de
 - ❖ Grande capacidade de
 - Processamento
 - Memória
 - Armazenamento
 - ❖ Um **sistema centralizado** precisa de
 - Máquina de grande porte (ex. *mainframe*)
→ alto custo de
 - aquisição,
 - manutenção e
 - expansão

Sistemas de Grande Porte (cont)

- ❖ Um **sistema distribuído** pode ter seus componentes
 - Armazenados,
 - Carregados e
 - Processados em
- Diversas máquinas de porte menor interligadas em rede → menor custo de
 - Aquisição,
 - Manutenção e
 - Expansão

Conceitos

- ❑ Os diversos componentes de software de um sistema distribuído, tendo a independência / autonomia de serem processados em máquinas diversas, são chamados de

Processos

- ❑ Um dos módulos do sistema operacional, o **escalonador**, gerencia a fila de processos que usarão o processador como um

Recurso

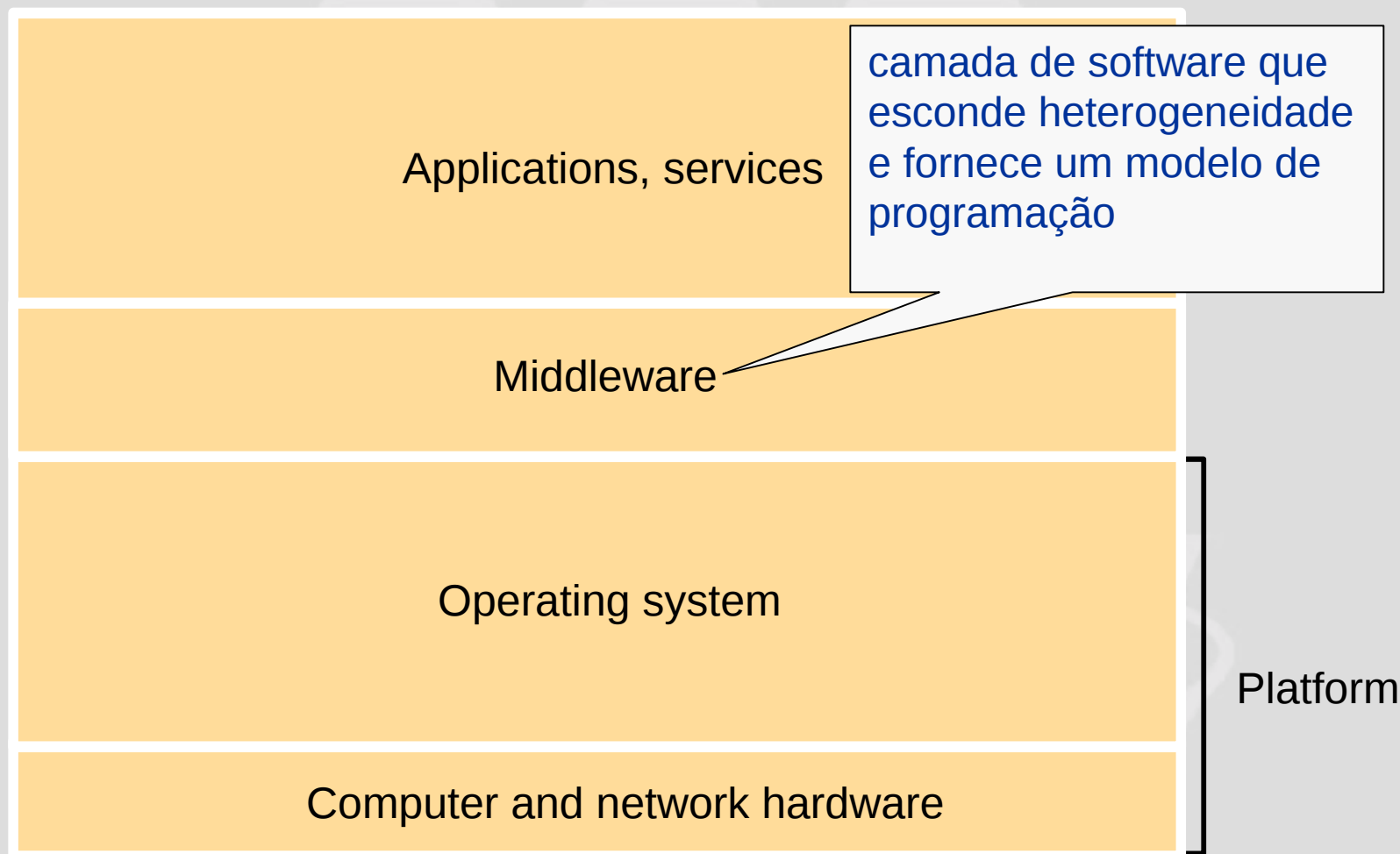
❖ Outros exemplos de recursos:

- Hw: memória, discos, impressoras
- Sw: arquivos, bancos de dados, compiladores

Conceitos (cont)

- ❑ **Threads**: considerados **processos leves**
 - ❖ Permitem que **atividades concorrentes** internas a um processo possam ser intercaladas pelo escalonador, podendo aumentar a eficiência do sistema
- ❑ **Plataforma** = hardware + sistema operacional
 - ❖ As soluções atuais de suporte visam tornar os sistemas distribuídos independentes de plataformas

Conceitos (cont)



Desafios

☐ Heterogeneidade

❖ Software

- SO
- Linguagens de programação

❖ Hardware

❖ Rede

☐ Abertura

☐ Segurança

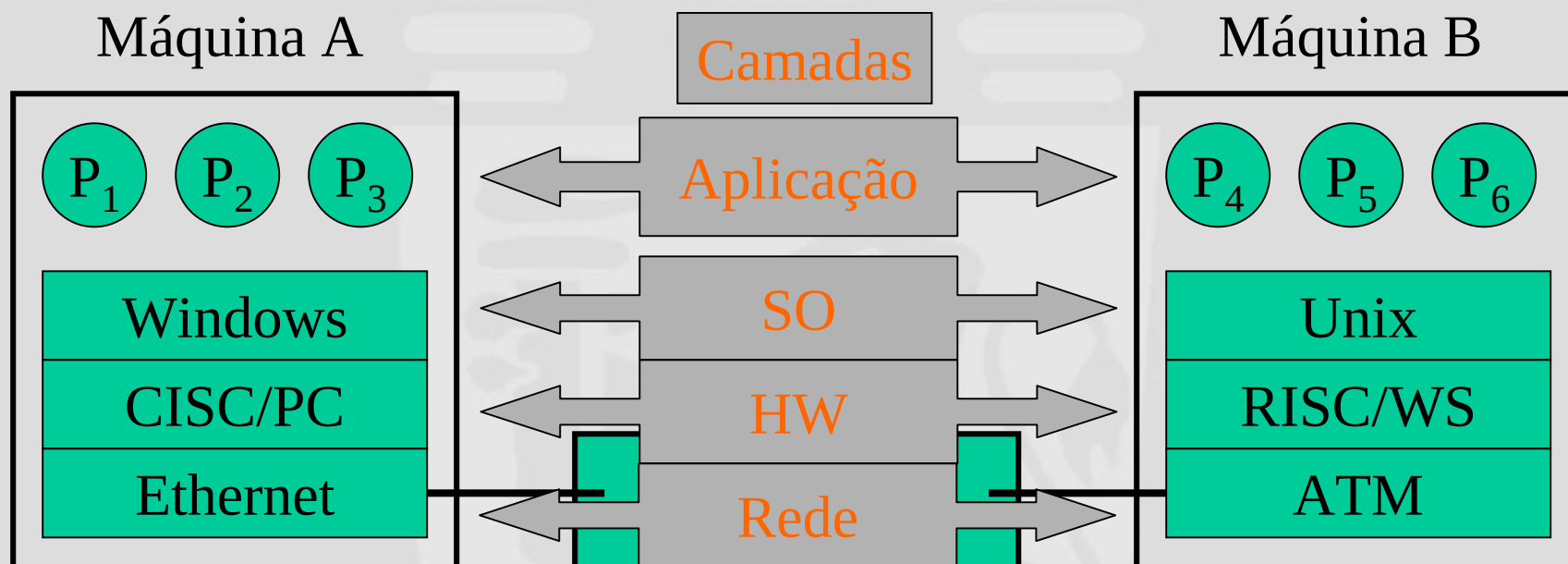
☐ Escalabilidade

☐ Tratamento de falhas

☐ Concorrência

☐ Transparência

Sistemas Heterogêneos



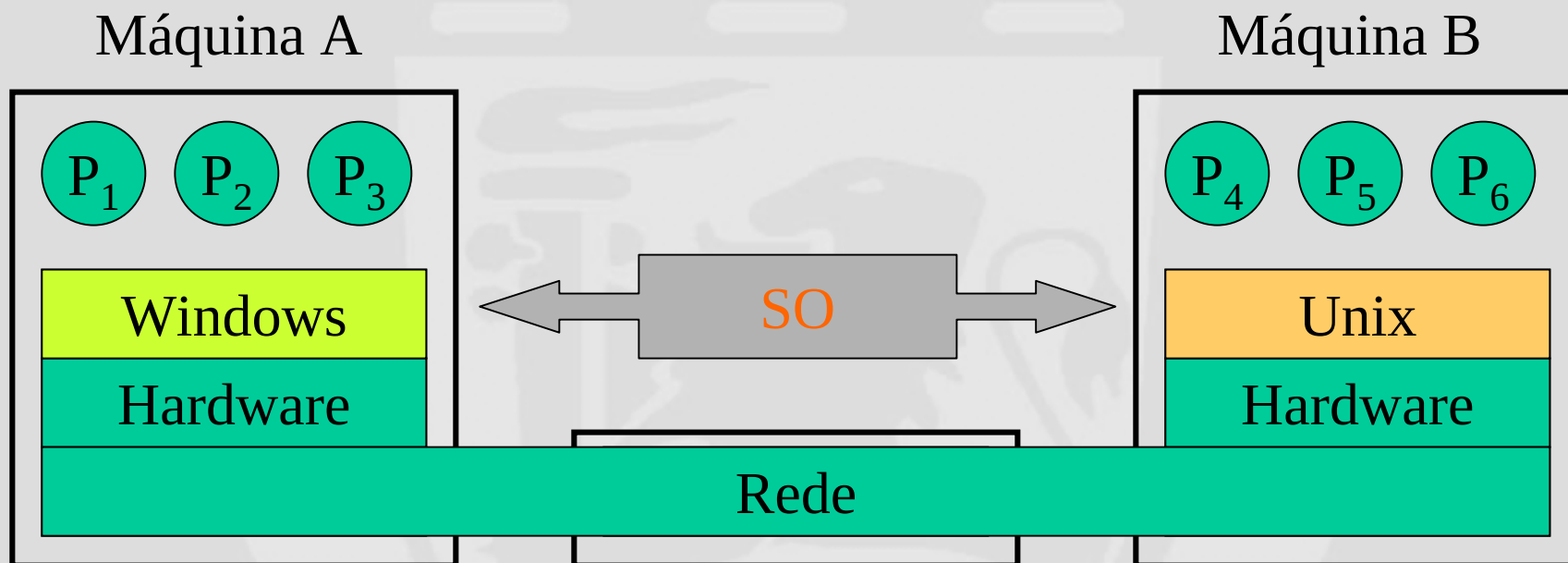
❑ Obs.: SOs são otimizados para determinadas arquiteturas de hardware

- ❖ Ex. 1: Windows, Linux para CISC/PCs
- ❖ Ex. 2: Unix para RISC/WS (Workstations)

Mas podem ser rodados/emulados em plataformas diversas; ex.: SAMBA emula Windows em RISC/WS

Sistemas Heterogêneos (cont)

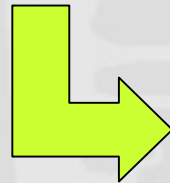
Considerando apenas a camada SO:



- ❑ Como rodar P₄ no Windows se ele foi construído (desenvolvido e compilado) para Unix?
 - ❖ Simplesmente recompilar? E se o programa usou construções específicas (*system calls*) do Unix?
Pode ser necessário **reprogramar** para **recompilar** no Windows...

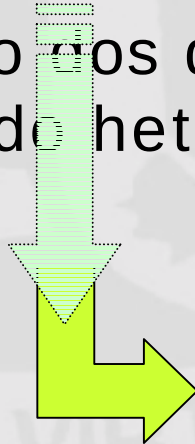
Sistemas Heterogêneos (cont)

- ❑ Para evitar reprogramação...
- ❑ Usar camada intermediária padronizada que



Middleware

- ❑ isole (torne independentes) os programas de aplicação dos diversos SOs de um sistema distribuído heterogêneo



Transparência de
heterogeneidade

Conceito / Desafio

❑ **Transparência**

- ❖ Esconde do usuário e do programador de aplicação a separação de componentes em um SD
- ❖ O sistema é percebido como um todo, em vez de uma coleção de componentes independentes

Outros Tipos de Transparência

- ☐ Localização: esconde onde o recurso está localizado
- ☐ Acesso: operações idênticas para acesso local e remoto
- ☐ Migração: esconde que um recurso pode se mover para outra localização
- ☐ Relocação: esconde que um recurso pode ser movido para outra localização enquanto está em uso
- ☐ Concorrência: compartilhamento de recursos sem interferência entre processos concorrentes
- ☐ Falha: esconde a falha e recuperação de um recurso
- ☐ Replicação: esconde de usuários ou programadores de aplicação a existência de réplicas de recursos

Desafio: Abertura

- ☐ Grau com o qual novos serviços compartilhados podem ser adicionados e usados
- ☐ Requer **publicação** de interfaces para acesso a recursos compartilhados
- ☐ Requer mecanismo de **comunicação uniforme**
- ☐ Conformidade de cada componente com o padrão publicado deve ser testada e verificada

Desafio: Segurança

❑ Segurança tem três componentes:

- ❖ Confidencialidade (proteção contra abertura para indivíduos não autorizados)
- ❖ Integridade (proteção contra corrupção)
- ❖ Disponibilidade (proteção contra interferência nos meios para acessar recursos)

❑ Dois difíceis desafios de segurança:

- ❖ Ataques de *Denial of Service* (DOS)
- ❖ Segurança de código móvel: pode acessar recursos locais?

Desafio: Escalabilidade

- ❑ Permanecer efetivo mesmo quando há um aumento significativo no número de recursos e usuários:
 - ❖ Controlar o custo dos recursos
 - ❖ Controlar a perda de desempenho
 - ❖ Prevenir que os recursos de software acabem (ex., endereços IP)
 - ❖ Evitar gargalos de desempenho

Desafio: Tratamento de Falhas

❑ Falha parcial

- ❖ Detecção (crash de máquina ou atraso de rede?)
- ❖ Mascaramento (retransmissão, replicação)
- ❖ Cliente desiste após número pré-determinado de tentativas, timeout etc
- ❖ Recuperação (*checkpoint* e *rollback recovery*)
- ❖ Redundância

Desafio: Concorrência

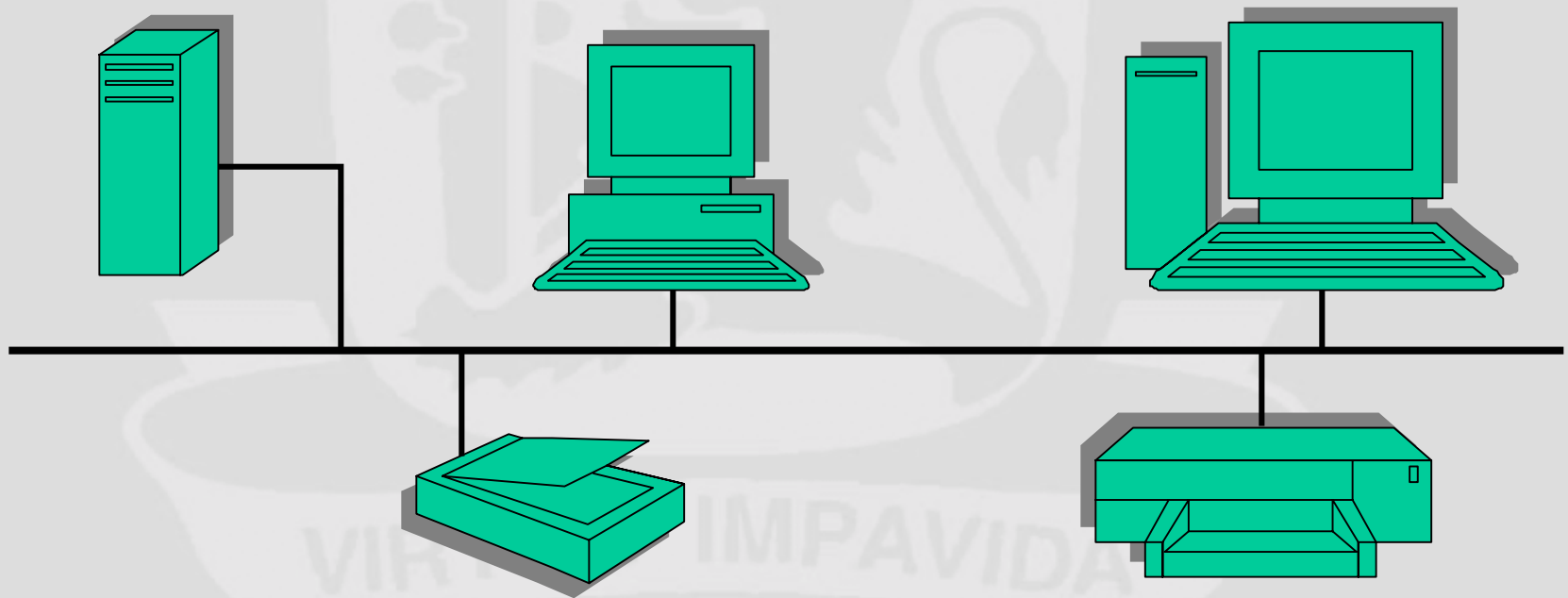
- ❑ É um problema quando dois ou mais usuários acessam o mesmo recurso ao mesmo tempo
 - ❖ *Threads*
 - ❖ Mecanismos de exclusão mútua (ex. semáforos)

Sistemas Distribuídos

- ☐ Por que ter?
- ☐ Para que?
- ☐ Como funcionam?

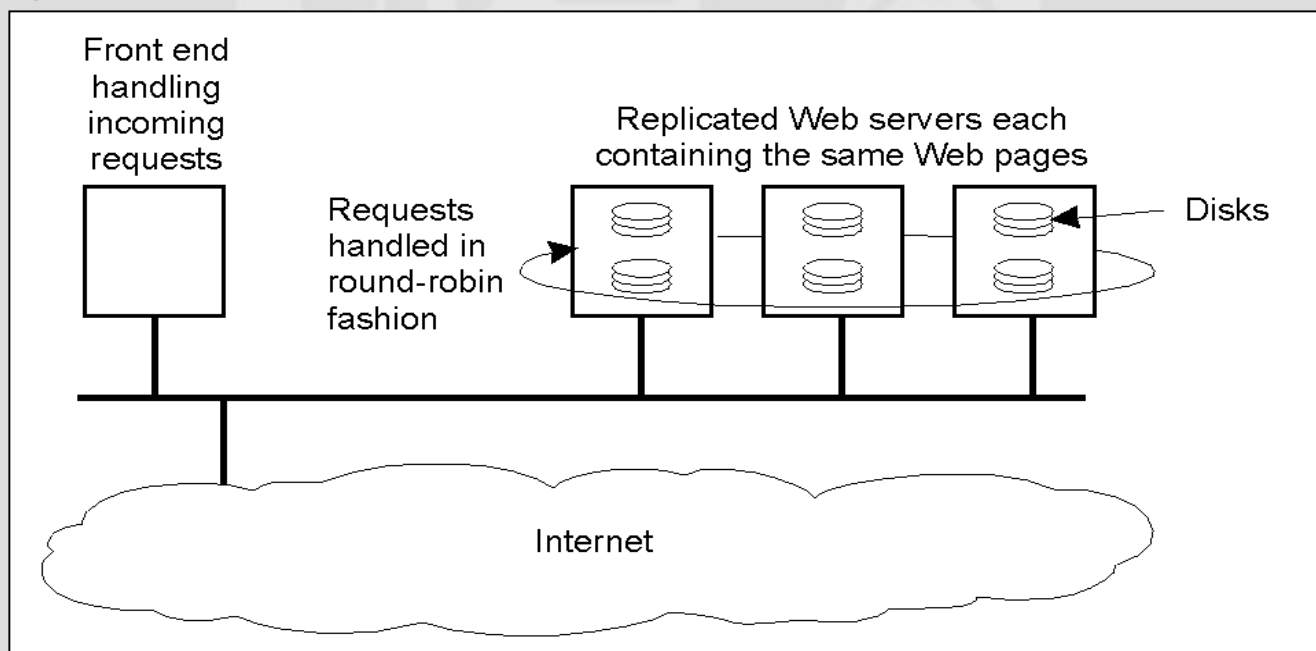
Por que ?

- ❑ Aproveitar recursos de software e hardware distribuídos



Para que ?

- ❑ Escalabilidade: atendimento à demanda
 - ❑ Desempenho: múltiplos processadores
 - ❑ Segurança
 - ❑ Disponibilidade: replicação; tolerância a falha
- Exemplos
- ARPAnet: Hardware, software e pessoas distribuídos
 - Serviço Web moderno:



Como funcionam?

❑ Suporte de comunicação

- ❖ Redes velozes e confiáveis



❑ Suporte de software

- ❖ Sistemas operacionais de rede
- ❖ Sistemas operacionais distribuídos
- ❖ *Middleware*

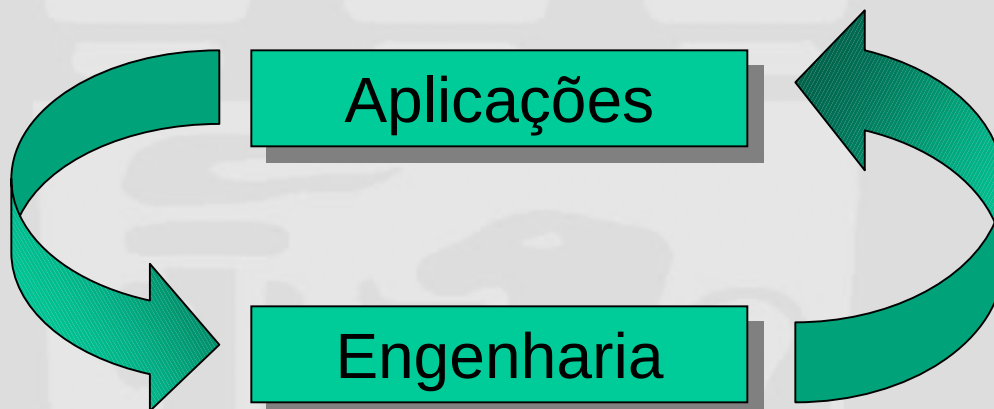


Desenvolvimento de Tecnologia

O papel dos Sistemas Distribuídos hoje

Antes...

Motivam



Possibilita

Hoje...

Motivam



Possibilitam