

# Projeto de Sistemas Distribuídos

Prof. Andrêza Leite  
[andreza.lba@gmail.com](mailto:andreza.lba@gmail.com)



# Exemplos de SD

- Quais podem ser?

# Ex. de SD: Internet

- Internet é um conjunto de redes de computadores, de muitos tipos diferentes, interligadas.
  - Intranets,
  - Provedores de serviços de Internet (ISPs)

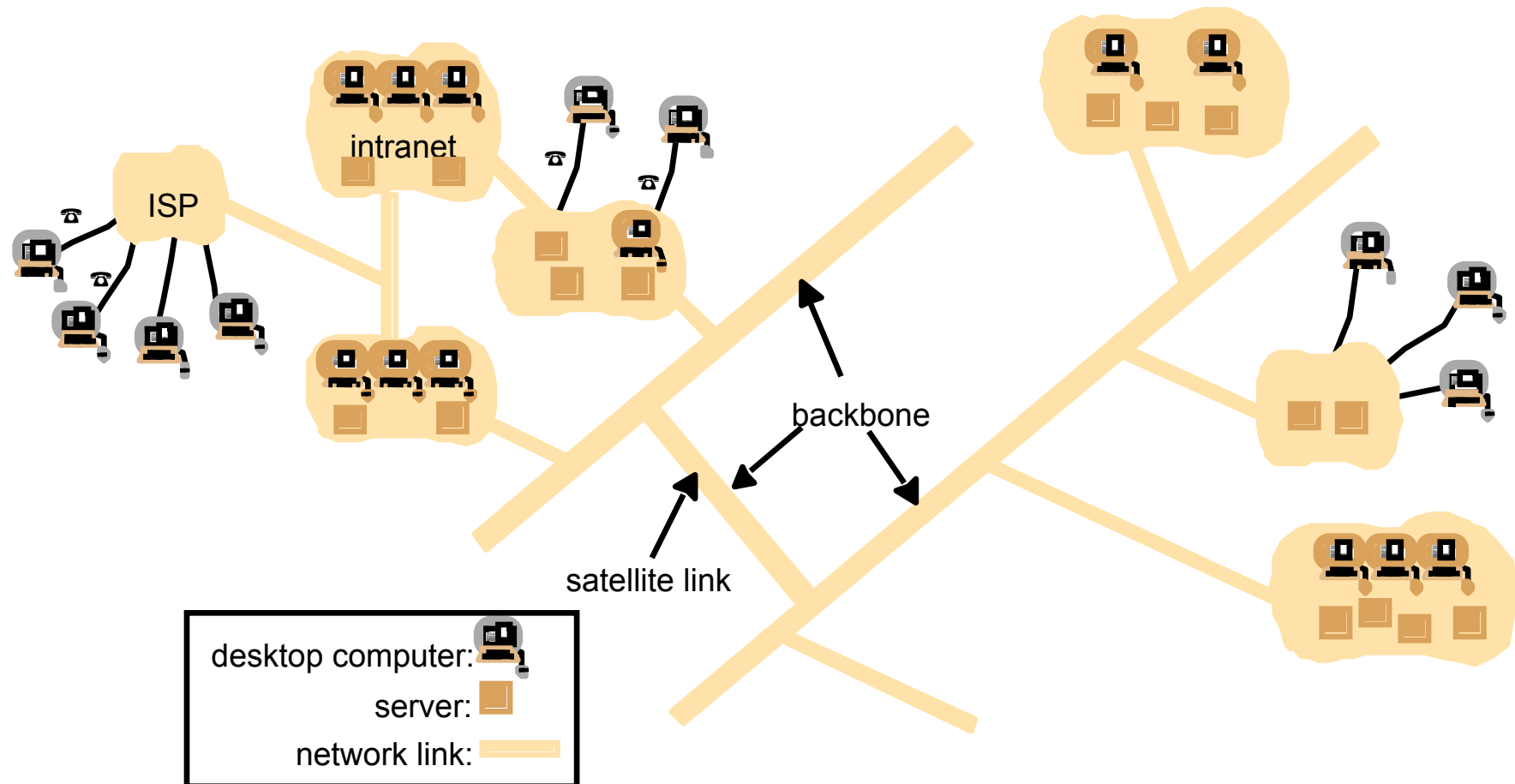


## Ex. de SD: Internet

- A internet é um sistema distribuído muito grande. Ela permite que os usuários, onde que estejam, façam uso de serviços como a World Wide Web, e-mail e transferência de arquivos.

# Ex. de SD: Internet

Nesta rede quais seriam os recursos compartilhados ?





## Ex. de SD: Intranet

- Uma intranet é uma parte da internet administrada separadamente, cujo limite pode ser configurado para impor planos de segurança locais.

## Ex. de SD: Intranet

- Sub-rede operada por empresas e outras organizações.
- Composta de varias redes locais (LANs) interligadas por conexões de *backbone*

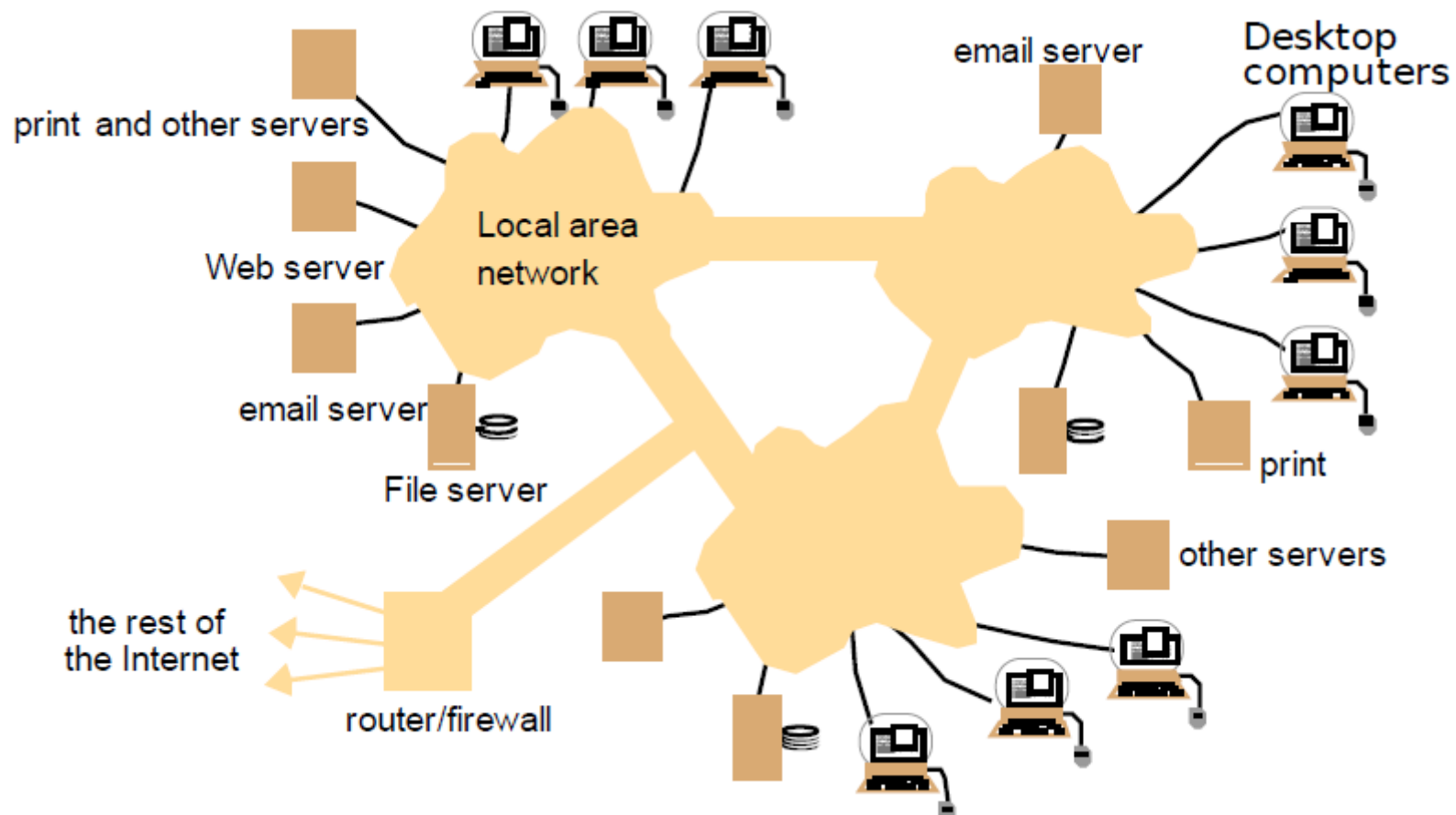
## Ex. de SD: Intranet

- *Backbone* é um enlace de rede com alta capacidade de transmissão, empregando conexões via satélite, cabos de fibra ótica ou outros meios físicos de transmissão que possuam uma grande largura de banda.



# Ex. de SD: Intranet

Nesta rede quais seriam os recursos compartilhados ?





# Ex. de SD: Rede Wireless

## Motivação

- Computação móvel e ubíqua
  - miniaturização de dispositivos e interligação em rede sem fio;
  - Integração de equipamentos de computação pequenos e portáteis com SDs.

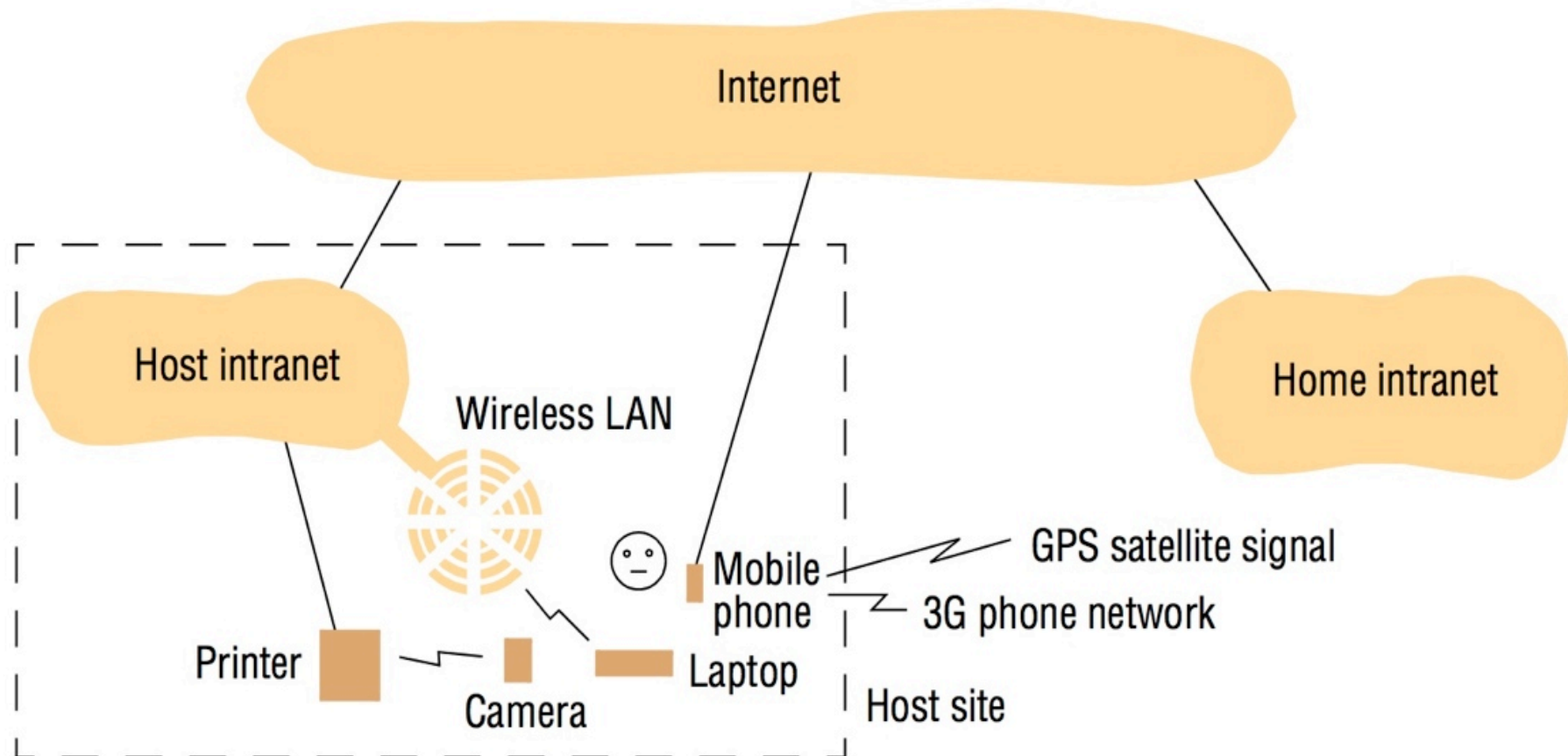
# Ex. de SD: Rede Wireless

Ubíquo?

- Noção de que o acesso a serviços de computação está onipresente (disponível em qualquer lugar).
- Então, computação móvel e ubíqua se sobrepõem.

# Ex. de SD: Rede Wireless


Nesta rede quais seriam os recursos compartilhados ?






# Compartilhamento de Recursos e a Web

- Geralmente compartilhamos recursos de hardware (impressoras), de dados (arquivos), e os com funcionalidades mais específicas (mecanismos de busca).



# Compartilhamento de Recursos e a Web

- Muitos SDs podem ser totalmente construídos na forma de clientes e servidores interagindo:
  - Web;
  - E-mail;
  - Impressoras interligadas;



# Compartilhamento de Recursos e a Web

## WEB

- sistema em evolução (web, 2.0, 3.0 ...) para publicação e acesso a recursos e serviços pela internet.




# Compartilhamento de Recursos e a Web

## WEB

- Baseada em três componentes básicos
  - Quais??!

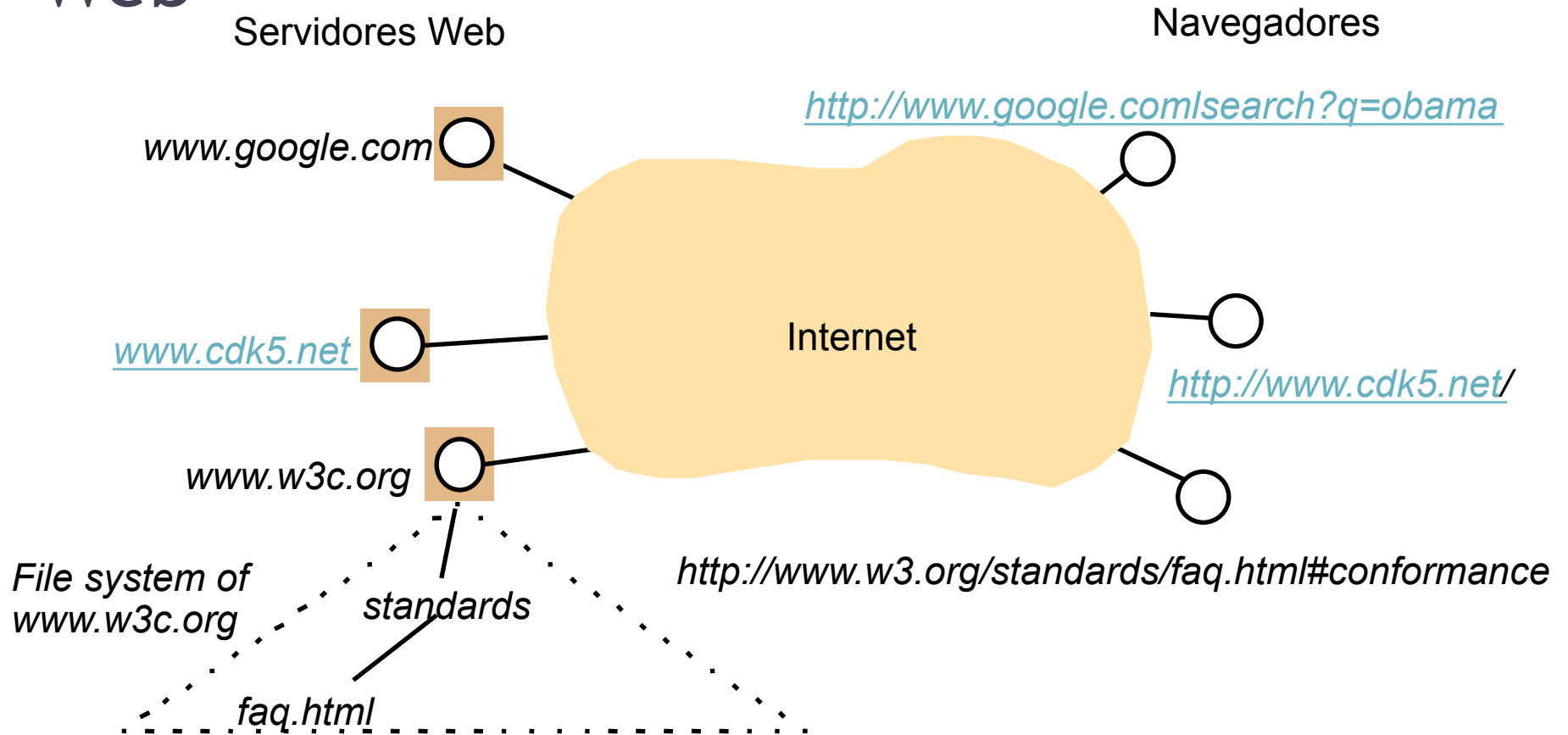




# Compartilhamento de Recursos e a Web

- HTML – linguagem para especificar o conteúdo;
- URLs – identificam os documentos e recursos armazenados como parte da web; e
- HTTP – protocolo de interação por meio dos quais navegadores(e outros tipos de clientes) buscam documentos e outros recursos dos servidores web

# Compartilhamento de Recursos e a Web






## Próxima aula ...

- Principais desafios para implantação de SD.



# Principais Desafios para Implantação de SD





# Principais Desafios para a Implantação

- Heterogeneidade;
- Abertura;
- Segurança;
- Escalabilidade;
- Tratamento de falhas;
- Concorrência;
- Transparência;



# Desafios::Heterogeneidade

- Variedade e diferença em aspectos de:
  - Redes;
  - Hardwares de computadores;
  - Sistemas operacionais;
  - Linguagens de Programação; e
  - Implementações de diferentes desenvolvedores;



# Desafios::Heterogeneidade

- Quais as soluções atuais?





# Desafios::Heterogeneidade

- Middleware
  - Uma camada de software que fornece uma abstração de programação, assim como o mascaramento da heterogeneidade.
    - Ex. CORBA (*Common Object Request Broker*); e
    - Java RMI (*Remote Method Invocation*).



# Desafios::Heterogeneidade

- **Middleware**
  - Alguns (RMI) suportam apenas uma linguagem;
  - A maioria é implementada sobre protocolos de internet e escondem a diferença entre redes;
  - Todo Middleware, em si, trata de diferenças em nível de SO e hardware.

# Desafios::Heterogeneidade

- **Middleware**

- Além da heterogeneidade, fornece um modelo computacional uniforme para ser usado pelos programadores de serviços e de aplicativos distribuídos. Os modelos possíveis incluem:

- Invocação remota de objetos;
    - Notificação remota de eventos;
    - Acesso remoto a BD; e
    - Processamento de transação distribuído.



# Desafios::Heterogeneidade

- Middleware
  - EX: CORBA fornece invocação remota a objetos permitindo que um objeto, em um programa sendo executado em um computador, invoque um método de um objeto em um programa executando em outro computador.

# Desafios::Heterogeneidade

- Código Móvel (migração de código)
  - Referencia-se ao código que pode ser enviado de um computador para outro e ser executado no destino;
  - *Applet* - software aplicativo que é executado no contexto de outro programa;
  - Diferentemente de um programa, um *applet* não pode rodar independentemente;



# Desafios::Heterogeneidade

- Código Móvel (migração de código)
- Exemplos:
  - *Applets Java*;
  - Vídeos em flash; e
  - Windows Media Player (plugin navegadores).
  - Como isso acontece?

# Desafios::Heterogeneidade

- Código Móvel (migração de código)
  - A estratégia de máquina virtual oferece uma maneira de tornar um código executável em qualquer tipo de processador e SO.
  - O compilador de uma linguagem específica gera código pra uma máquina virtual, em vez de processador e SO específicos.
    - Java Virtual Machine - JVM



## Desafios::Sistemas Abertos

- Sistemas que podem ser estendidos e reimplementados de diversas maneiras;
- O fato de um SD ser aberto ou não é determinado pelo grau com que novos serviços podem ser adicionados e disponibilizados para uso por uma variedade de programas clientes;
- EX. APIs públicas.



# Desafios::Sistemas Abertos

- A característica de sistema aberto é obtida a partir do momento em que a especificação e a documentação das principais interfaces de software dos componentes de um sistema estão disponíveis para os desenvolvedores de software;
- As principais interfaces são publicadas e os sistemas podem ser ampliados/estendidos em nível de hardware, pela adição de computador na rede, e em nível de software, pela introdução de novos serviços ou pela reimplementação dos antigos.

## Desafios::Segurança

- Muitos recursos de informação que se tornam disponíveis em SD tem alto valor para seus usuários.
- Nos SDs os clientes enviam pedidos para acessar dados gerenciados por servidores, o que envolve o envio de informações em mensagens por uma rede.

# Desafios::Segurança

- A segurança de recursos de informação tem três componentes:
  - Confidencialidade – Proteção contra exposição para pessoas não autorizadas;
  - Integridade – Proteção contra alteração ou dano;
  - Disponibilidade – Proteção contra interferência com os meios de acesso aos recursos.
- Envolve questões de ocultar conteúdo de mensagens, saber com certeza a identidade do usuário, ou outro agente, em nome de quem a mensagem foi enviada.

# Desafios::Segurança

- Desafios ainda não totalmente resolvidos:
  - Negação de serviço (*Denial of Service*) - ocorre quando um usuário interrompe um serviço.
    - Como pode ser conseguido? Como se chama?
  - Segurança de Código móvel – precisa ser manipulado com cuidado. Considere alguém recebendo um programa executável como anexo em um e-mail: os efeitos da execução são imprevisíveis.



## Desafios::Escalabilidade

- SDs funcionam de forma efetiva em escalas diferentes, variando desde uma pequena intranet até a internet;
- Um sistema é escalável quando permanece eficiente em aumento do número de recursos e usuários;
- EX. A internet – número de computadores e serviços vem aumentando substancialmente.



# Desafios::Escalabilidade

- Projetos de SDs escaláveis apresenta desafios de:
  - Controlar custos dos recursos físicos;
  - Controlar a perda de desempenho;
  - Impedir que recursos de sw se esgotem;
  - Evitar gargalos de desempenho.



# Desafios::Tratamento de falhas

- Falhas em sistemas distribuídos são parciais: alguns falham outros não.
- Técnicas atuais:
  - Detecção;
  - Mascaramento;
  - Tolerância;
  - Recuperação; e
  - Redundância – os serviços podem fazer uso de componentes redundantes.



## Desafios::Concorrência

- Garantir que processos operem corretamente em ambientes concorrentes.
- Para que um objeto mantenha coerência em um ambiente concorrente, suas operações devem ser sincronizadas de maneira que seus dados permaneçam consistentes.
- Pode ser obtidos por técnicas padrões como semáforos, disponíveis na maioria dos S.Os.



# Desafios::Transparência

- É definida como a ocultação, para usuário final, da separação dos componentes em um SD de modo que o sistema seja percebido como um todo, em vez de uma coleção de componentes independentes.
  - Acesso;
  - Localização;
  - Concorrência;
  - Replicação;
  - Falhas;
  - Mobilidade;
  - Desempenho; e
  - Escalabilidade.

# Próxima aula: Arquiteturas

## **Acesso Remoto**

- Acesso distribuído a recursos centralizados (ex. mainframe via terminal, FTP)

## **Distribuição de Arquivos e Memória**

- Mais comum em redes locais (Disco, memória, etc)

## **Acesso Remoto Avançado**

- Terminal server, X11



# Arquiteturas

## **Arquiteturas Cliente/Servidor**

- Acesso a métodos de outros processos para execução de tarefas específicas;
- Duas e três camadas;

## **Código Móvel**

- Applets Java

# Projeto de Sistemas Distribuídos

Prof. Andrêza Leite  
[andreza.lba@gmail.com](mailto:andreza.lba@gmail.com)