

1.2 Princípios de Projeto

- Só porque é possível construir sistemas distribuídos não significa necessariamente que seja uma boa ideia
- Vamos discutimos quatro objetivos importantes que devem ser alcançados para que a construção de um sistema distribuído valha o esforço



1.2 Princípios de Projeto

1.2.1 Compartilhamento de recursos

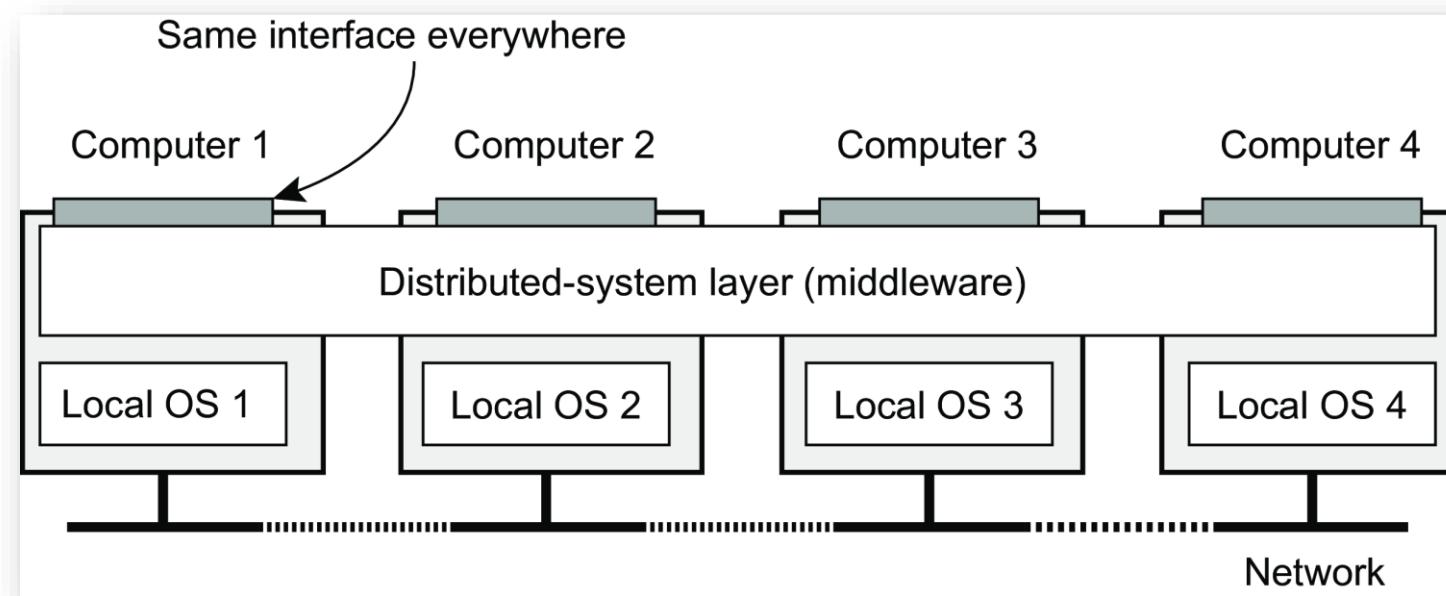
- Um objetivo importante de um sistema distribuído é tornar mais fácil para os usuários (e aplicativos) para acessar e compartilhar recursos remotos
- Os recursos podem ser virtualmente qualquer coisa



1.2 Princípios de Projeto

1.2.2 Transparência de distribuição

- Um objetivo importante de um sistema distribuído é ocultar o fato de que seus processos e recursos estão fisicamente distribuídos por vários computadores
 - Tenta tornar a distribuição de processos e recursos transparente, ou seja, invisível para usuários finais e aplicações
 - Alcançar a transparência da distribuição é realizada por meio do middleware



1.2 Princípios de Projeto

1.2.2 Transparência de distribuição

- **Tipos de Transparência de distribuição**

Transparency	Description
Access	Hide differences in data representation and how an object is accessed
Location	Hide where an object is located
Relocation	Hide that an object may be moved to another location while in use
Migration	Hide that an object may move to another location
Replication	Hide that an object is replicated
Concurrency	Hide that an object may be shared by several independent users
Failure	Hide the failure and recovery of an object

1.2 Princípios de Projeto

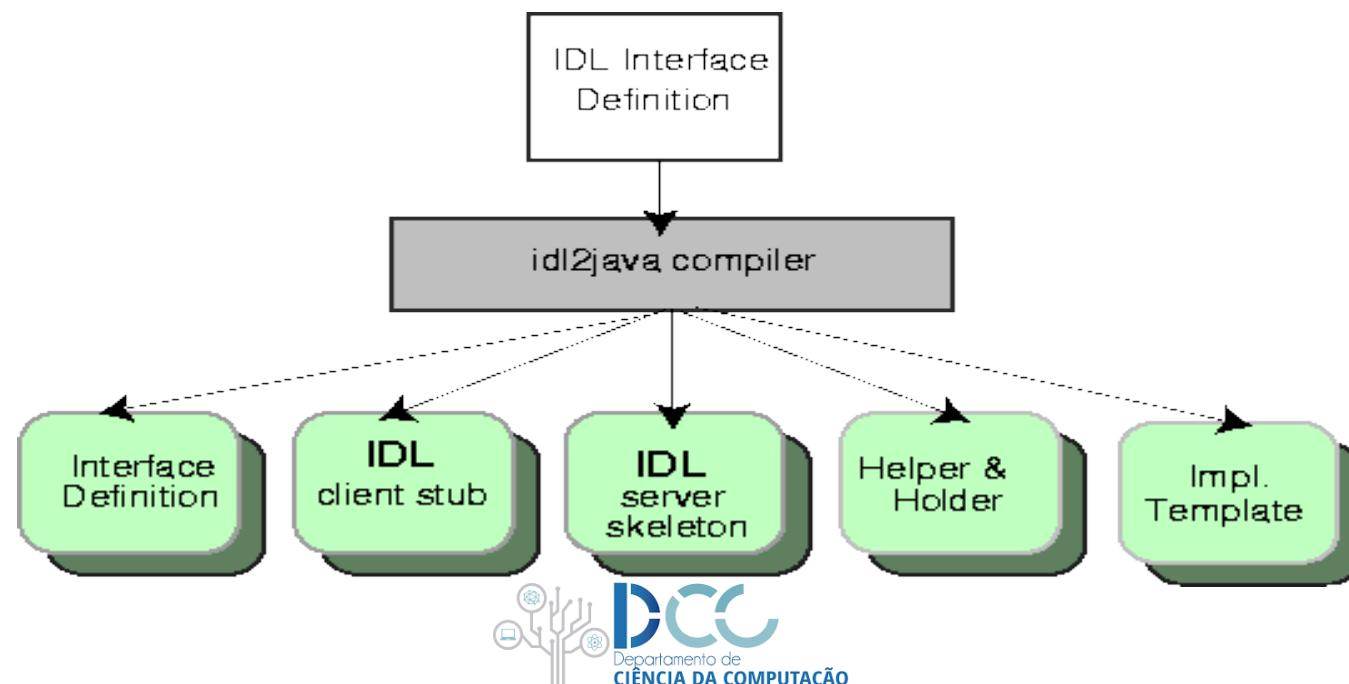
1.2.3 Abertura

- Ser capaz de interagir com serviços de outros sistemas abertos, independentemente do ambiente subjacente:
 - Os sistemas devem estar em conformidade com **interfaces bem definidas**
 - Os sistemas devem interoperar facilmente
 - Sistemas devem suportar a **portabilidade de aplicativos**
 - Os sistemas devem ser **facilmente extensíveis**

1.2 Princípios de Projeto

1.2.3 Abertura

- Os sistemas devem estar em conformidade com **interfaces bem definidas**
 - Os sistemas devem interoperar facilmente
 - O uso de uma Linguagem de Definição de Interface (IDL) pode ajudar a descrever os serviços oferecidos pelos diferentes componentes do sistema de uma forma padronizada e fácil de entender



1.2 Princípios de Projeto

1.2.3 Abertura

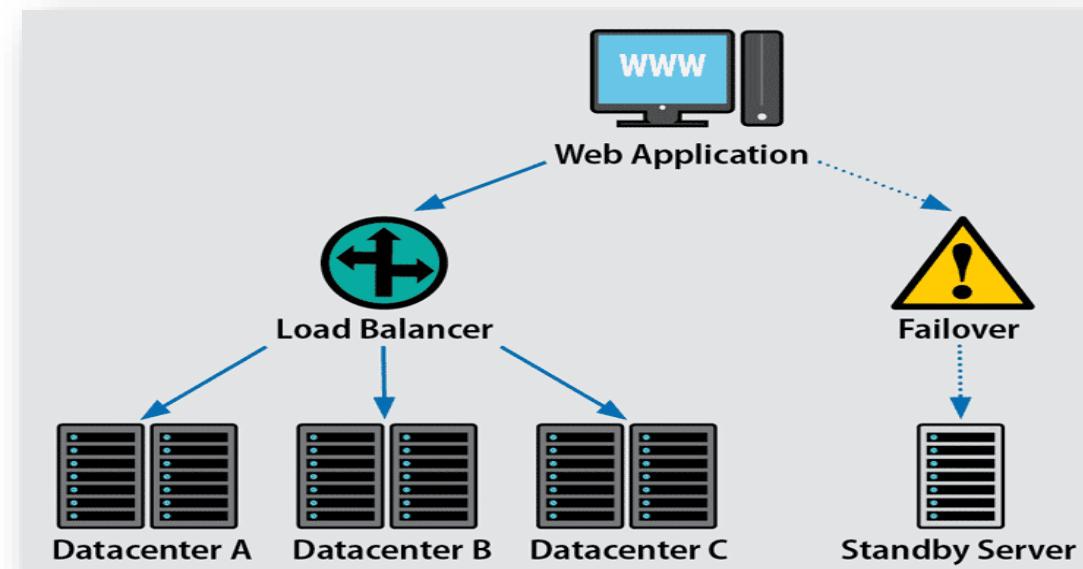
- Sistemas devem suportar a **portabilidade de aplicativos**
- Os sistemas devem ser **facilmente extensíveis**
 - Um sistema distribuído aberto deve permitir que novos componentes sejam adicionados ou que os componentes existentes sejam substituídos por novos, sem afetar o funcionamento do sistema como um todo



1.2 Princípios de Projeto

1.2.4 Confiabilidade

- Grau em que um sistema de computador pode ser confiável para operar conforme o esperado
- A confiabilidade em sistemas distribuídos pode ser **bastante complexa devido a falhas parciais**
- Um objetivo importante de sistemas distribuídos é mascarar essas falhas, assim como mascarar a recuperação dessas falhas: **tolerância a falhas**



1.2 Princípios de Projeto

1.2.4 Confiabilidade

- Confiabilidade é um termo que abrange vários requisitos úteis:
 - **Confiabilidade:** refere-se à propriedade que um sistema pode executar continuamente sem falha
 - **Disponibilidade:** definida como a propriedade de que um sistema está pronto para ser usado imediatamente
 - **Segurança:** refere-se à situação em que, quando um sistema falha temporariamente em operar corretamente, nenhum evento catastrófico acontece
 - **Manutenibilidade:** refere-se à facilidade com que um sistema com falha pode ser reparado

1.2 Princípios de Projeto

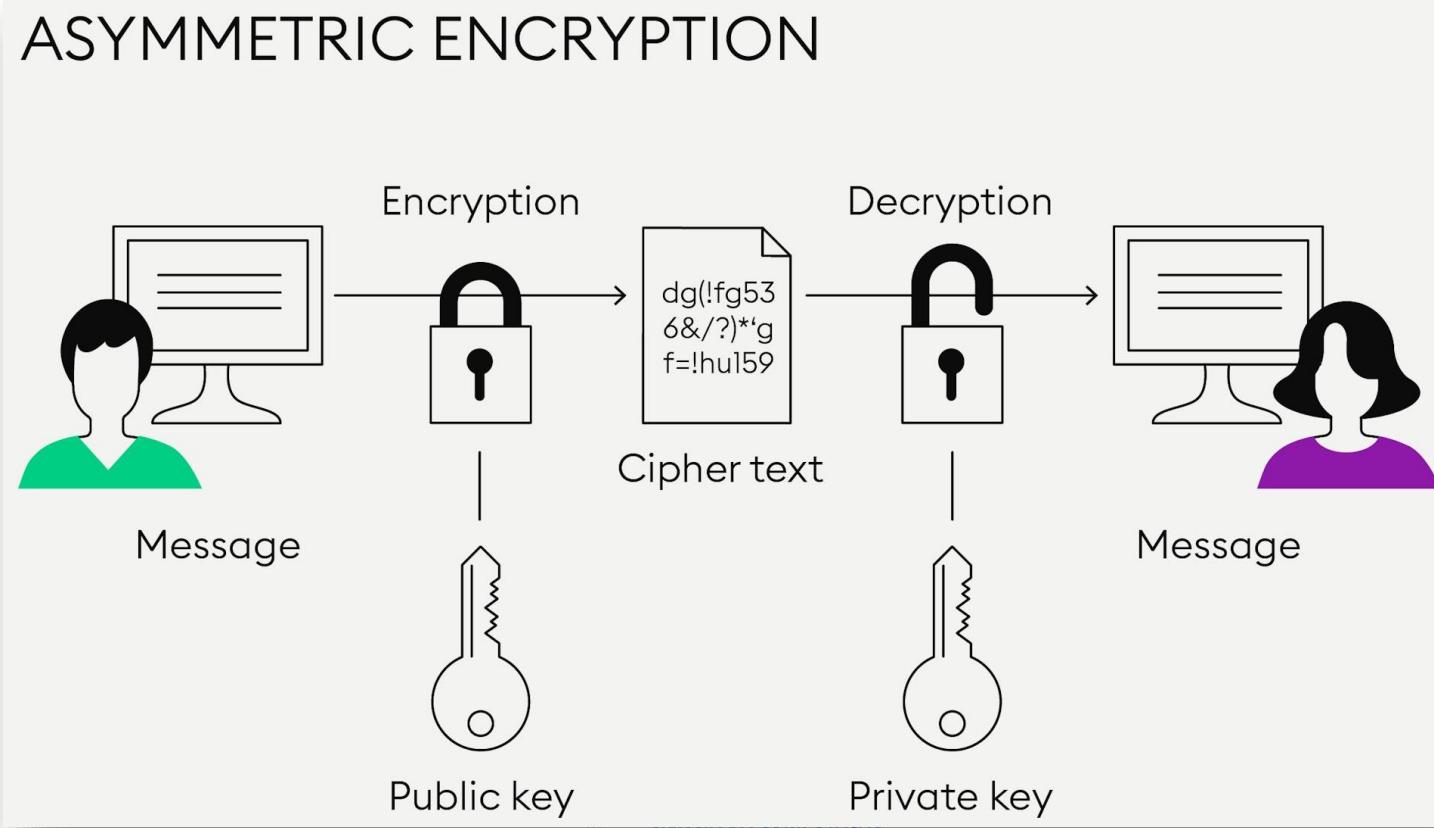
1.2.5 Segurança

- Um sistema distribuído que não é seguro não é confiável
- É necessário garantir a confidencialidade e a integridade
 - Ambas diretamente relacionadas à divulgação e acesso autorizados de informações e recursos
 - A autorização é feita verificando se uma entidade identificada tem os direitos de acesso adequados
 - Isso significa que o sistema deve saber que está lidando de fato com a entidade correta
 - Se um sistema pode autenticar positivamente uma pessoa, qual é o valor dessa autenticação se a pessoa não pode ser confiável?
 - Por exemplo, em sistemas financeiros, a autorização pode limitar a quantidade de dinheiro que uma pessoa pode transferir entre várias contas.

1.2 Princípios de Projeto

1.2.5 Segurança

- Uso de criptografia de chaves assimétricas



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- Muitos desenvolvedores de sistemas distribuídos modernos usam facilmente o adjetivo “**Escalável**” sem deixar claro por que o sistema realmente é dimensionado
- Em Sistemas Distribuídos existem três componentes:
 - Número de usuários e/ou processos: **escalabilidade de tamanho**
 - Distância máxima entre nós: **escalabilidade geográfica**
 - Número de domínios administrativos: **escalabilidade administrativa**



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- **Escalabilidade de Tamanho**
- Causas principais para problemas de escalabilidade em sistemas centralizados:
 - A capacidade computacional, limitada pelas CPUs
 - A capacidade de armazenamento, incluindo a taxa de transferência entre CPUs e discos
 - A rede entre o usuário e o serviço centralizado



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- Escalabilidade de Tamanho
- Atraso em filas de redes
 - https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/queuing-loss-applet/index.html

1.2 Princípios de Projeto

1.2.5 Escalabilidade

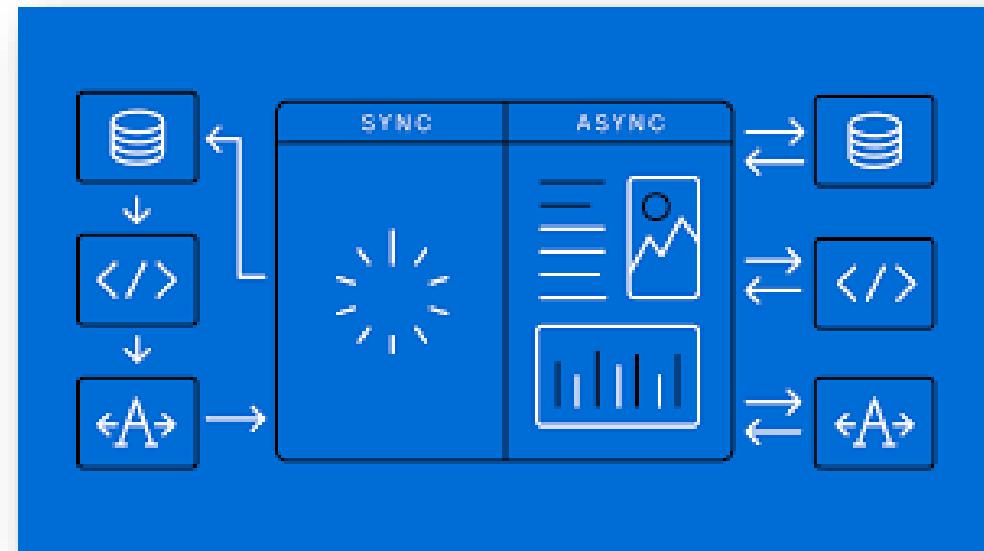
- **Escalabilidade de Tamanho**
- A maioria dos sistemas centralizados tem escalabilidade de tamanho limitada
- **Solução comum:** vários servidores poderosos operando independentemente em paralelo



1.2 Princípios de Projeto

1.2.5 Escalabilidade

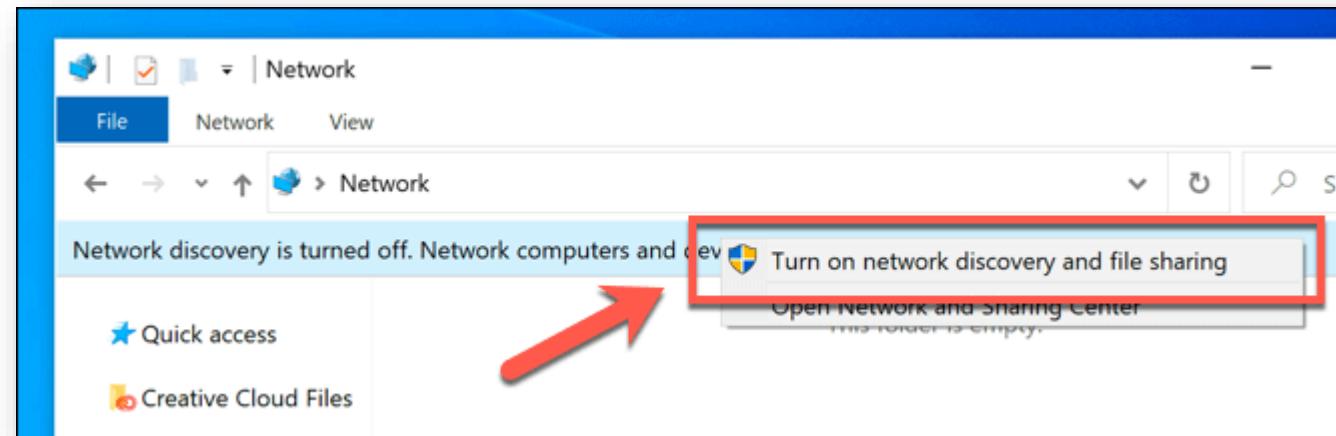
- **Escalabilidade Geográfica**
- Não se pode simplesmente migrar de uma LAN para WAN: muitos sistemas assumem interações **cliente-servidor síncronas**. A latência pode facilmente proibir esse esquema
- Os links de WAN são inherentemente pouco confiáveis: basta mover o fluxo de vídeo de uma LAN para WAN está fadado a falhar



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- Escalabilidade Geográfica
- Falta de comunicação multiponto, de modo que uma simples busca por broadcast não pode ser implantada

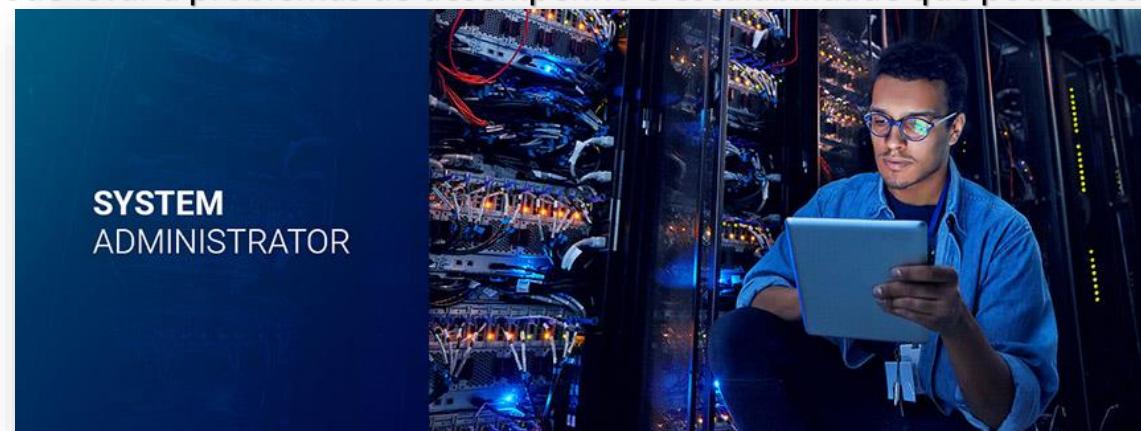


- A solução é desenvolver serviços de nomes e diretórios separados (com seus próprios problemas de escalabilidade).

1.2 Princípios de Projeto

1.2.5 Escalabilidade

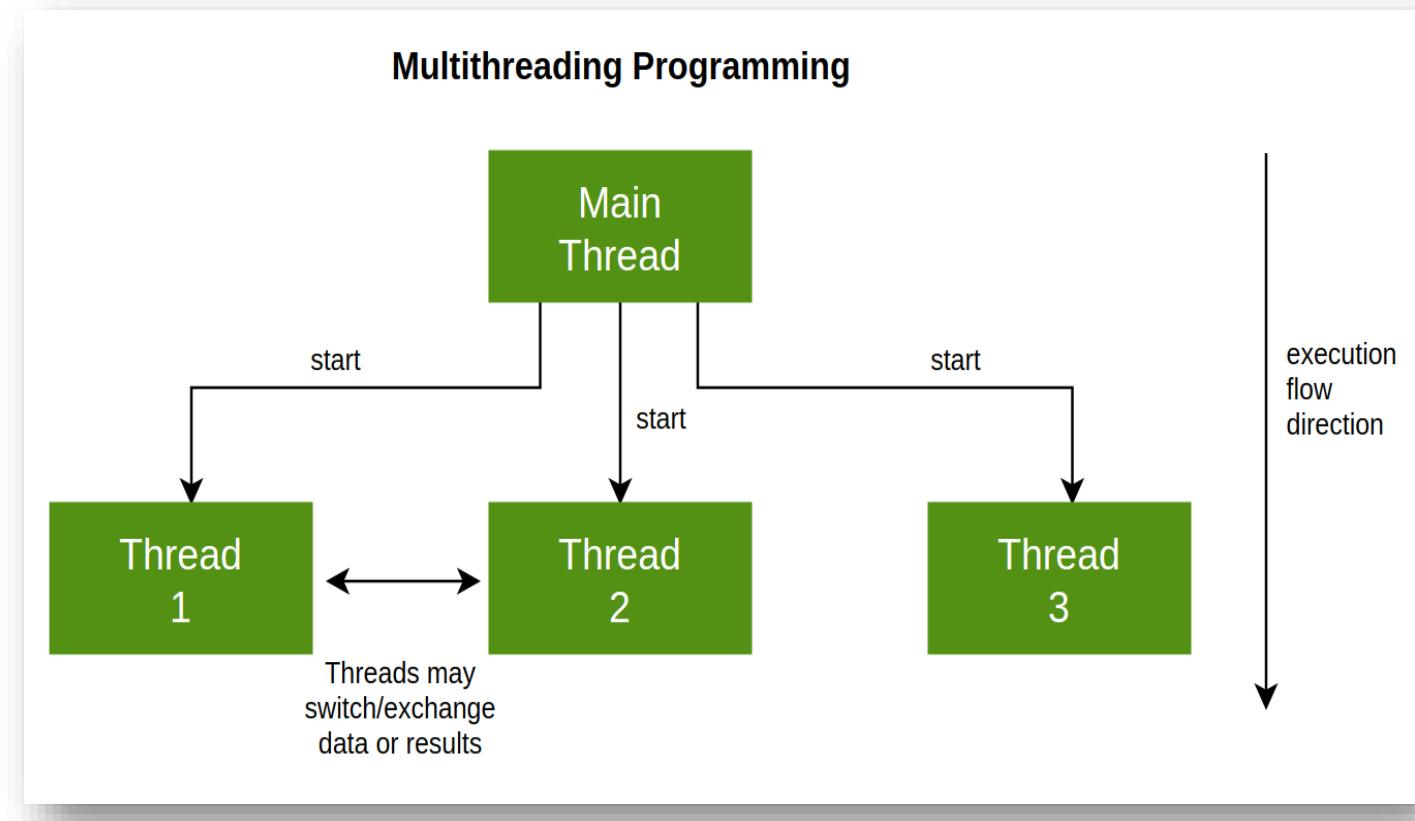
- **Escalabilidade Administrativa**
- Existem vários pontos relevantes a serem considerados para garantir a escalabilidade administrativa em sistemas distribuídos. Abaixo estão alguns dos principais pontos:
 - **Complexidade:** Isso pode levar a problemas de gerenciamento de configuração, provisionamento de recursos e orquestração, especialmente quando há muitos componentes envolvidos
 - **Confiabilidade:** A falha de um componente pode afetar outros componentes do sistema, o que pode levar a problemas de escalabilidade e desempenho
 - **Segurança:** Garantir que o sistema esteja protegido contra ameaças internas e externas pode ser um desafio, especialmente quando há muitos pontos de entrada para o sistema
 - **Monitoramento:** Isso pode levar a problemas de desempenho e escalabilidade que podem ser difíceis de detectar e corrigir



1.2 Princípios de Projeto

1.2.5 Escalabilidade

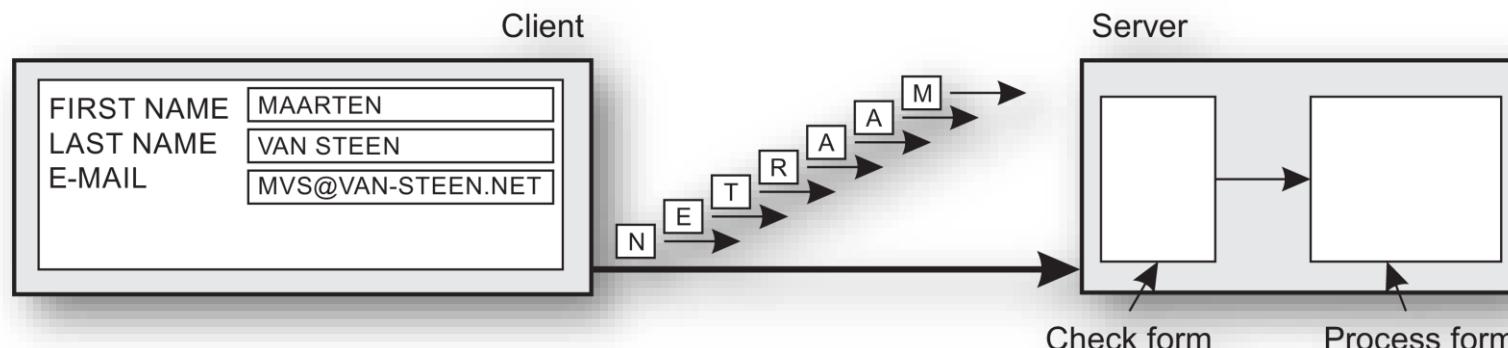
- Técnicas de aumento de escala
- Ocultar latências de comunicação
 - Faça uso de comunicação assíncrona
 - Ter um handler separado para a resposta recebida
- **Problema:** nem toda aplicação se encaixa nesse modelo



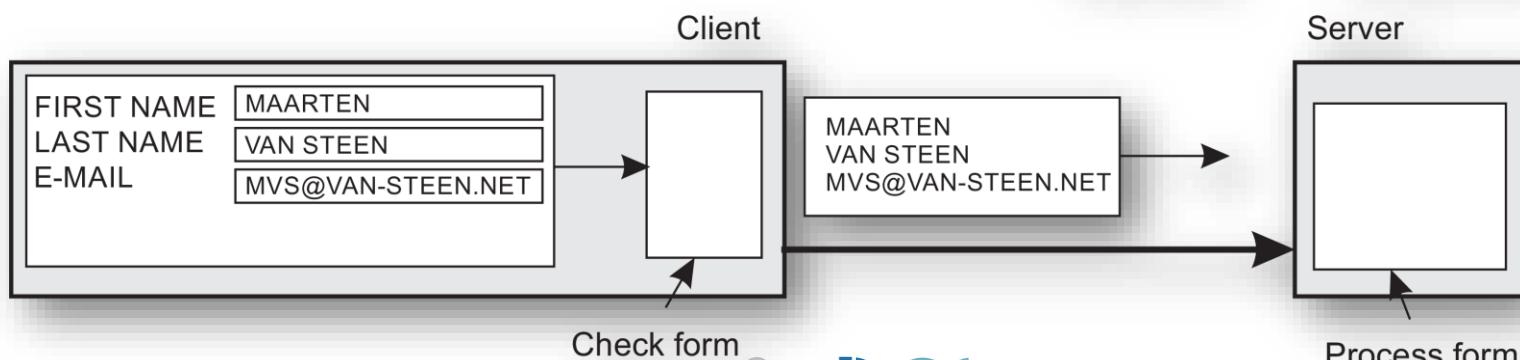
1.2 Princípios de Projeto

1.2.5 Escalabilidade

- Técnicas de aumento de escala



-



1.2 Princípios de Projeto

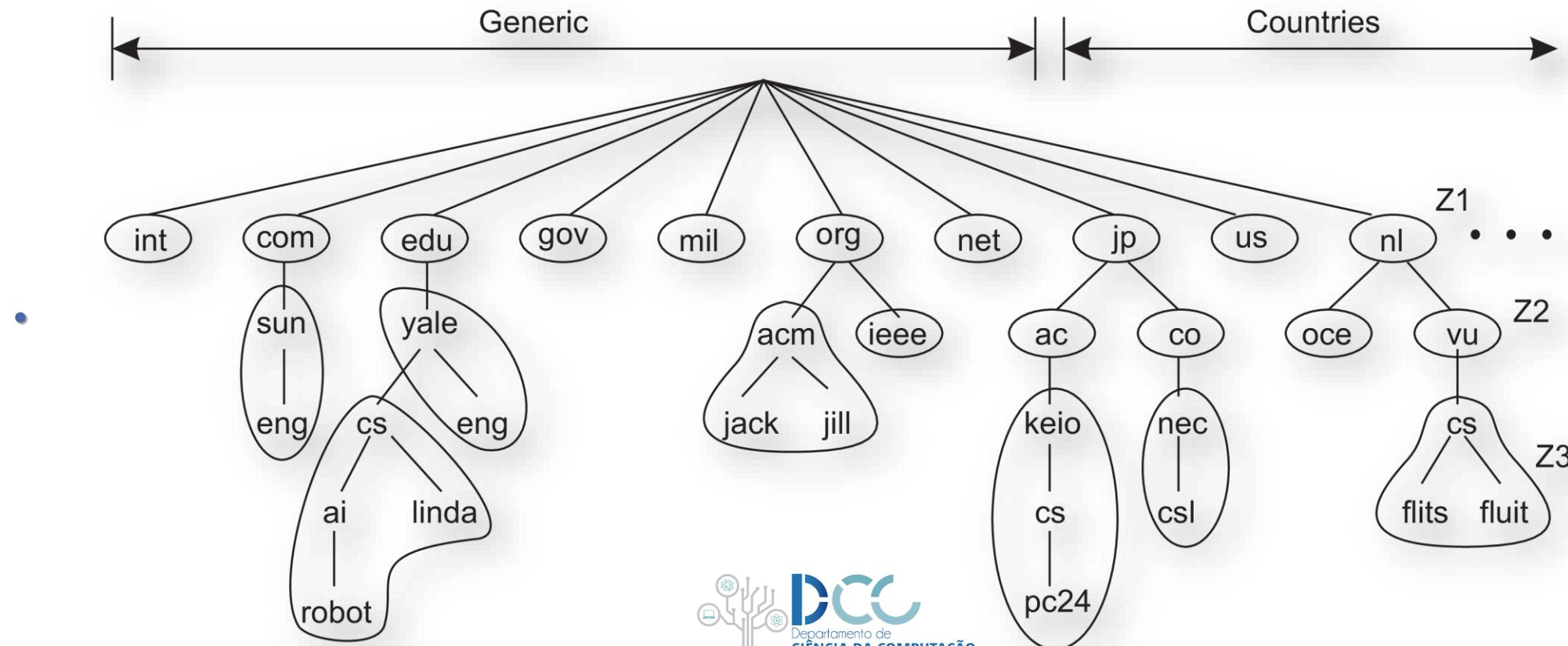
1.2.5 Escalabilidade

- Técnicas de aumento de escala
- Particionar dados e computação em várias máquinas
 - Mover computação para clientes (applets Java ou javascript)
 - Serviços de nomes descentralizado (DNS)
 - Sistemas de informação descentralizados (WWW)
-

1.2 Princípios de Projeto

1.2.5 Escalabilidade

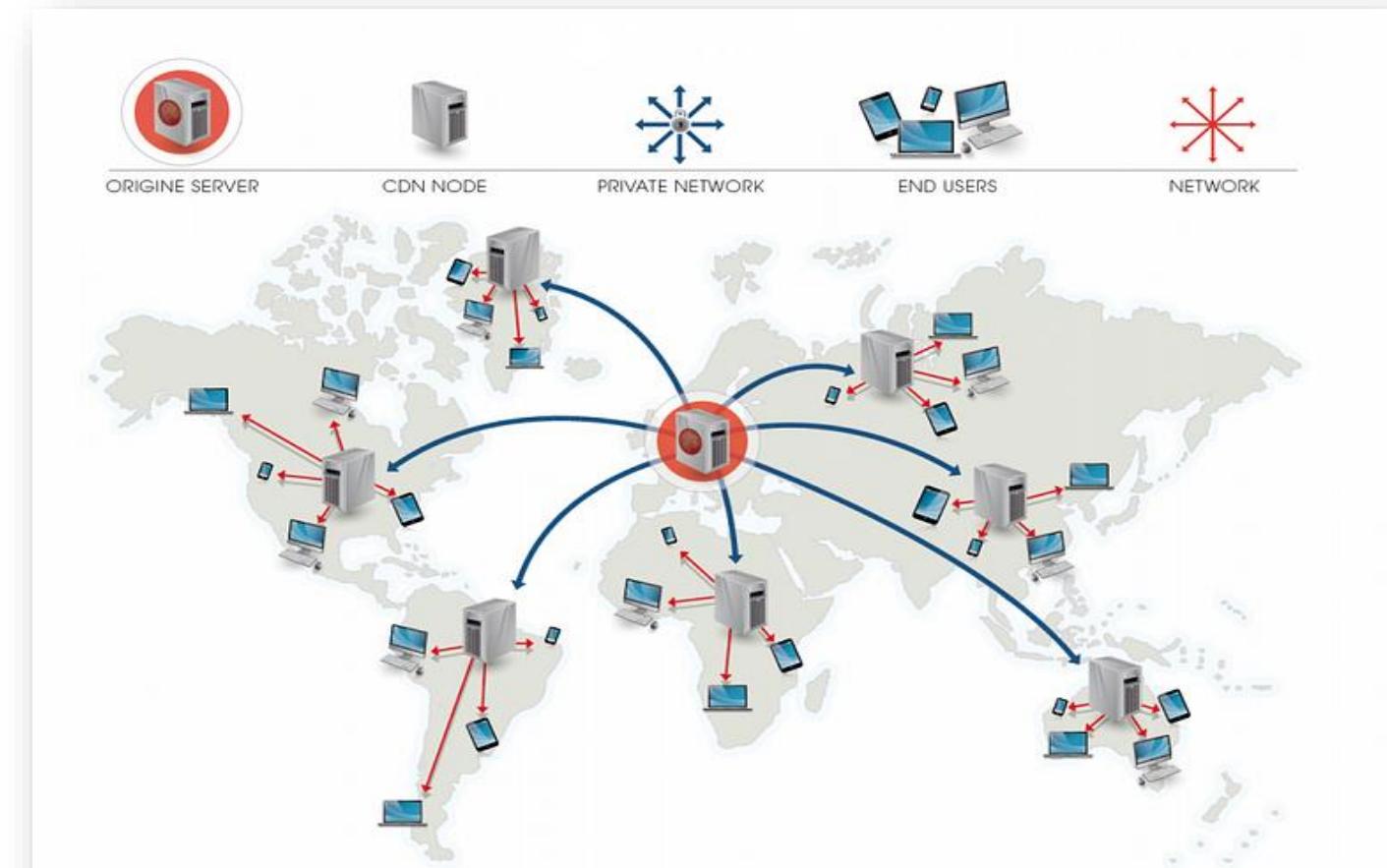
- Técnicas de aumento de escala



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- **Técnicas de aumento de escala**
- **Replicação e armazenamento em cache:**
 - Disponibilize cópias de dados em diferentes máquinas
 - Servidores de arquivos e bancos de dados replicados
 - Sites espelhados
 - Caches da Web (em navegadores e proxies)
 - Armazenamento em cache de arquivos (no servidor e no cliente)



1.2 Princípios de Projeto

1.2.5 Escalabilidade

- Técnicas de aumento de escala
- Aplicar replicação é fácil, exceto por uma coisa
 - Ter várias cópias (armazenadas em cache ou replicadas) **leva a inconsistências**: modificar uma cópia torna essa **cópia diferente do resto**
 - Manter cópias sempre consistentes **requer sincronização global** em cada modificação
 - A sincronização global **impede soluções em larga escala**
- Se pudermos **tolerar inconsistências**, poderemos reduzir a **necessidade de sincronização**, mas **tolerar inconsistências depende do aplicativo**
-

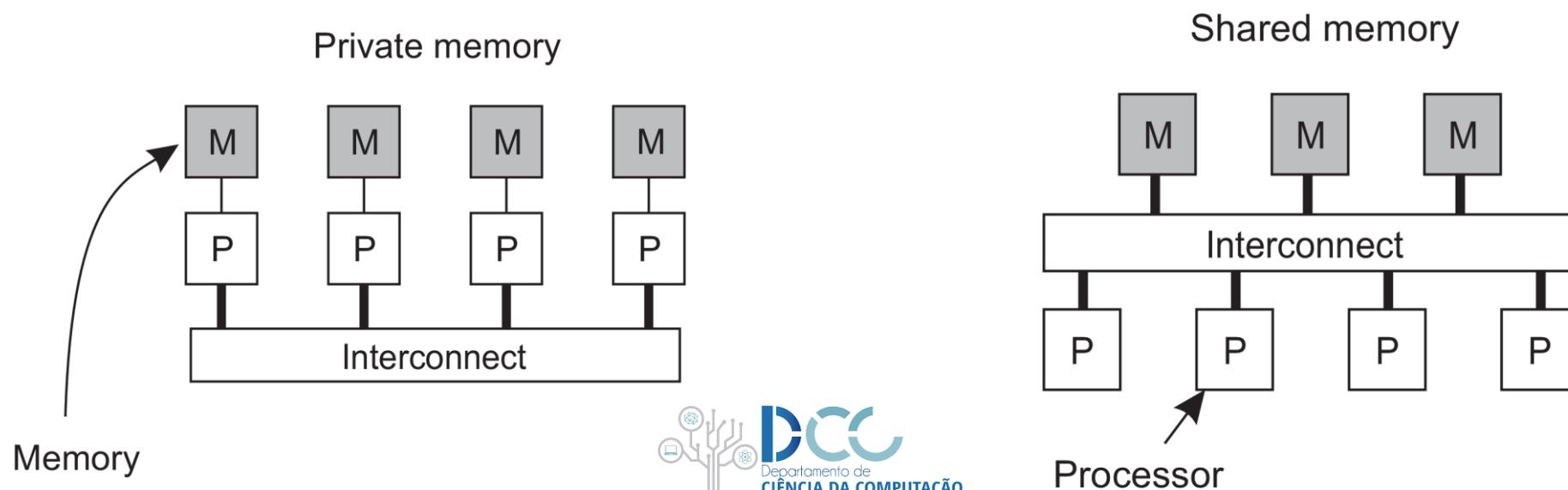
1.3 Classificação de Sistemas Distribuídos

- Sistemas desenvolvidos para alta performance
- Sistemas desenvolvidos para computação pervasiva, ou seja, para a "Internet das coisas"

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Computação Paralela:** Começou com a introdução de máquinas multiprocessadoras
 - Nesse caso, várias CPUs são organizadas de tal forma que todas têm acesso à mesma memória física
 - Um sistema multicomputador, vários computadores são conectados por meio de uma rede e não há compartilhamento de memória principal
 - O modelo de memória compartilhada mostrou-se altamente conveniente para melhorar o desempenho de programas e foi relativamente fácil de programar.



1.3 Classificação de Sistemas Distribuídos

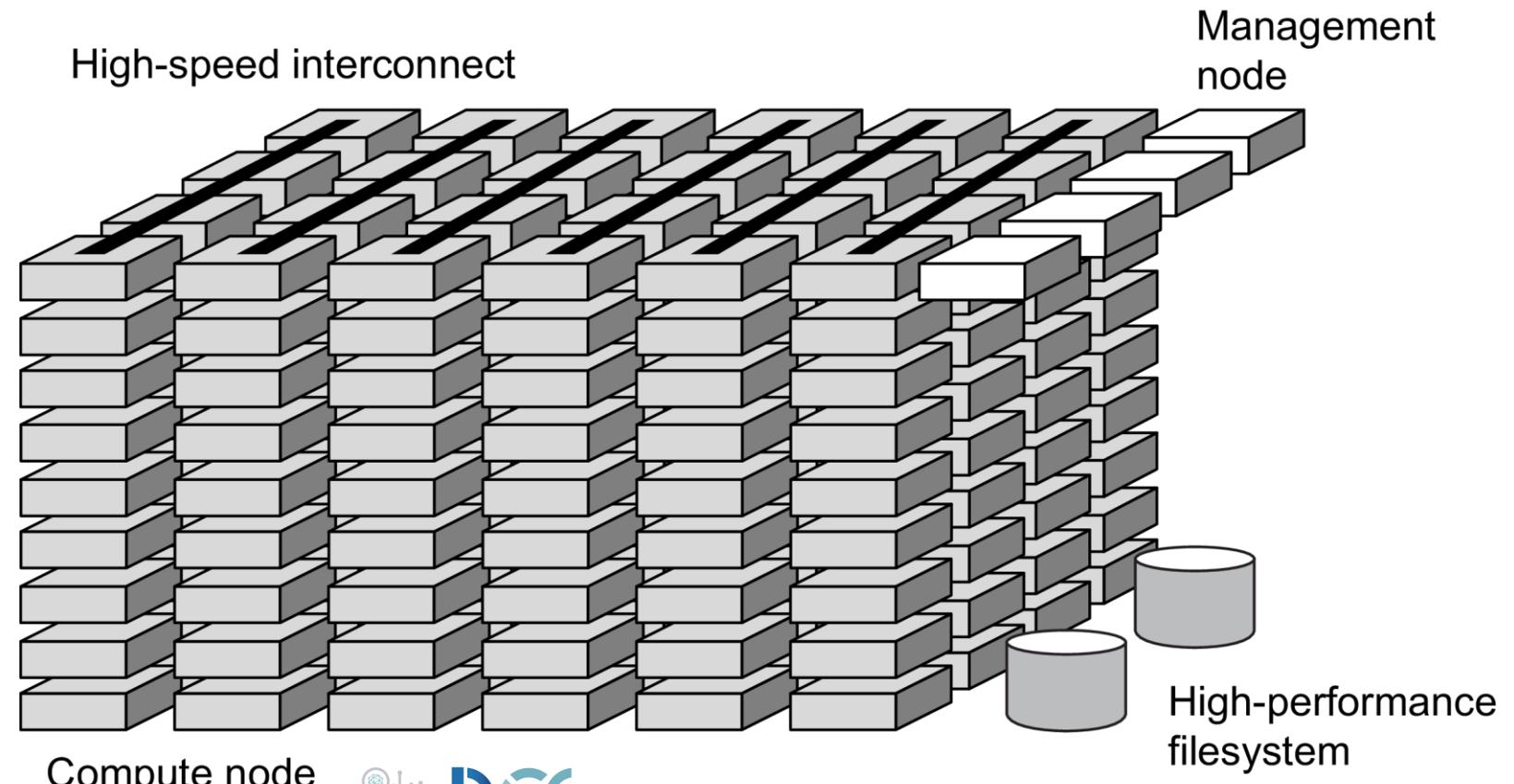
1.3.1 Computação distribuída de alta performance.

- **Cluster Computing:** Cluster computing é uma técnica de computação distribuída que envolve a conexão de vários computadores independentes para trabalhar juntos como se fossem uma única máquina virtual
 - Esses computadores, chamados de nós, são conectados por meio de uma rede de alta velocidade e geralmente executam o mesmo sistema operacional
 - Cada nó do cluster executa um conjunto de tarefas em paralelo, e o conjunto de nós trabalhando em conjunto é capaz de realizar tarefas computacionais complexas com alta eficiência e escalabilidade
 - Essa técnica é amplamente utilizada em várias áreas, como pesquisa científica, modelagem de sistemas complexos, processamento de grandes quantidades de dados e computação de alta performance

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Cluster Computing**

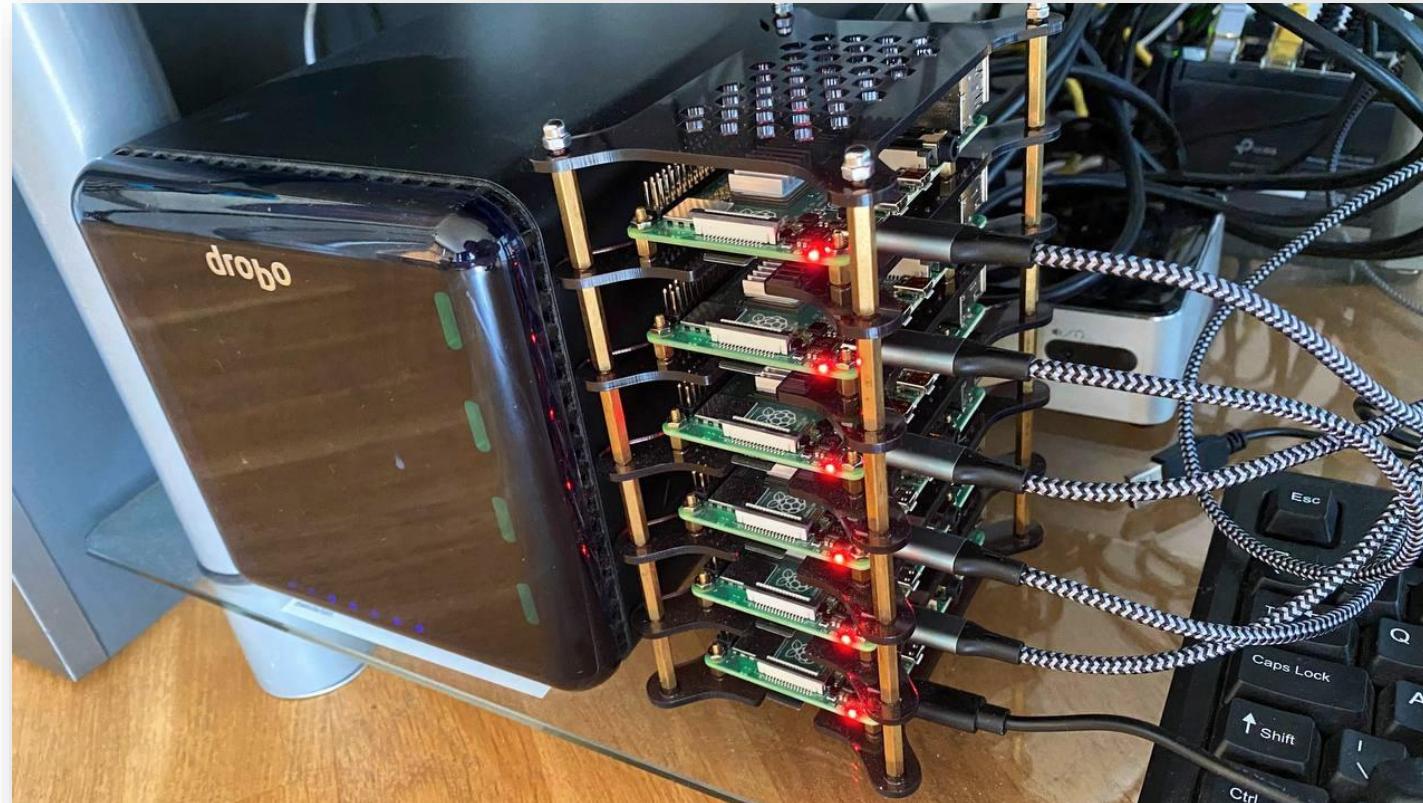


Compute node

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- Cluster Computing



1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Cluster Computing:** Existem vários softwares populares disponíveis para criação de clusters de computação:
 - **Apache Hadoop** - um software de código aberto para processamento de grandes conjuntos de dados distribuídos em clusters
 - **Apache Spark** - um mecanismo de processamento de dados em larga escala que fornece uma API unificada para processamento de dados em lote, processamento em tempo real e aprendizado de máquina
 - **Kubernetes** - uma plataforma de orquestração de contêineres para automação de implantação, dimensionamento e gerenciamento de aplicativos em contêineres
 - **Docker Swarm** - uma ferramenta para criação de clusters de contêineres Docker para implantação e gerenciamento de aplicativos
 - **OpenMPI** - uma biblioteca de mensagem para programação paralela em clusters de computação

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Grid Computing:** Grid computing é uma forma de computação distribuída na qual recursos computacionais, como processadores, armazenamento e rede, são compartilhados em larga escala entre diversas organizações ou indivíduos
 - O objetivo do grid computing é maximizar a utilização dos recursos disponíveis, aumentando a capacidade de processamento e armazenamento, reduzindo custos e melhorando a eficiência
 - **Heterogênea**
 - **Dispersada em várias organizações**
 - **Pode facilmente estender uma rede de longa distância (WAN)**
 - **IBM grid computing**
 - <https://www.youtube.com/watch?v=esVzoSqQ1Pc>

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Cloud Computing:** ou computação em nuvem, é um modelo de entrega de serviços de computação que permite acesso sob demanda a recursos de computação compartilhados, como servidores, armazenamento, aplicativos e outros serviços de TI, por meio da Internet
 - Os serviços de cloud computing geralmente são oferecidos por provedores de nuvem, que gerenciam e mantêm a infraestrutura de TI subjacente necessária para fornecer esses serviços
 - Os usuários podem acessar esses serviços de nuvem por meio de uma conexão com a Internet, pagando apenas pelo uso que fazem dos recursos



1.3 Classificação de Sistemas Distribuídos

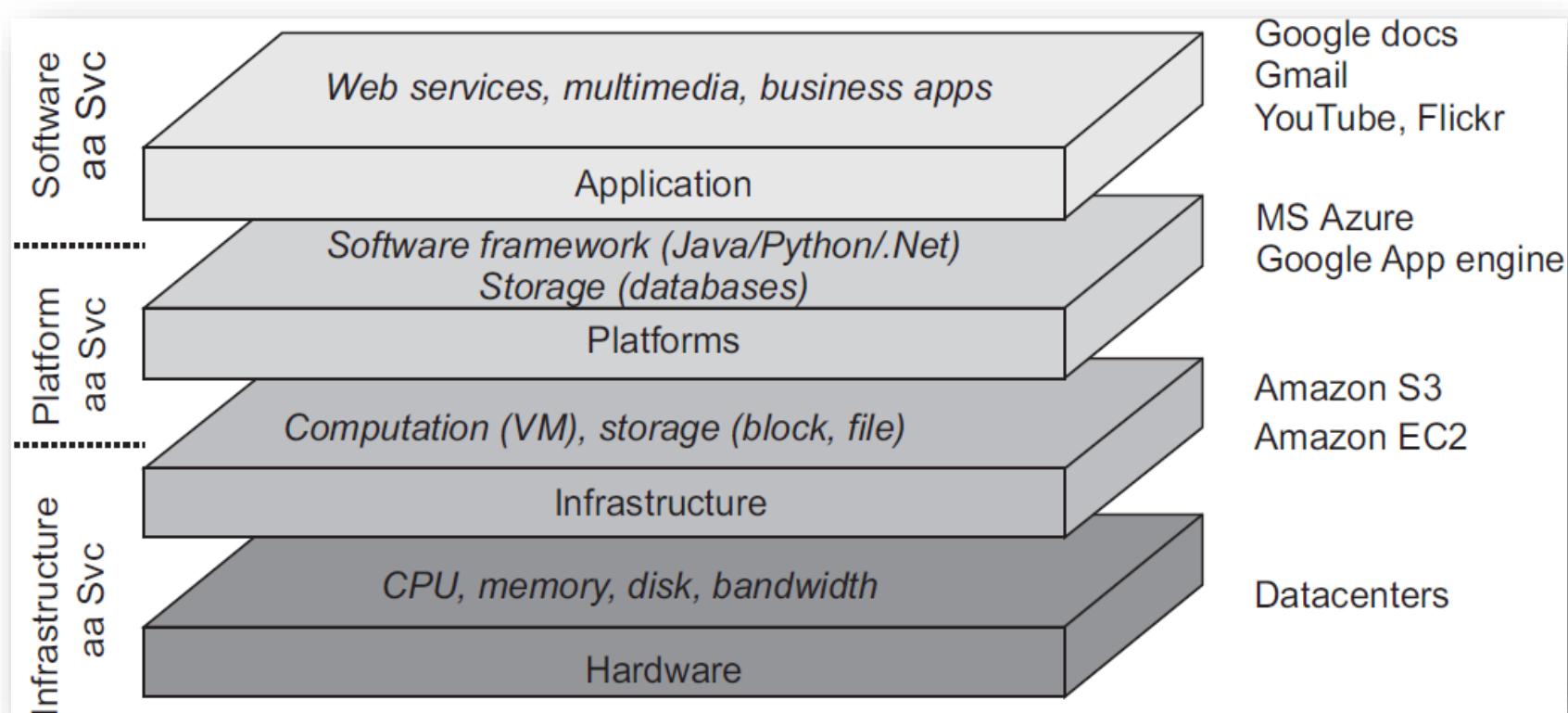
1.3.1 Computação distribuída de alta performance.

- **Cloud Computing:** Existem três principais modelos de serviços em nuvem (cloud computing):
 - **Infrastructure as a Service (IaaS)** - Fornece infraestrutura de TI sob demanda, incluindo servidores, armazenamento, rede e recursos de virtualização. Os usuários podem implantar e executar seus próprios sistemas operacionais, aplicativos e softwares no ambiente de nuvem
 - **Platform as a Service (PaaS)** - Fornece uma plataforma de desenvolvimento e execução de aplicativos na nuvem, sem que os usuários precisem gerenciar a infraestrutura subjacente. A plataforma inclui recursos como sistemas operacionais, banco de dados, middleware, ferramentas de desenvolvimento e serviços de gerenciamento de dados
 - **Software as a Service (SaaS)** - Fornece aplicativos de software hospedados na nuvem, acessíveis aos usuários por meio da Internet. Os usuários podem acessar os aplicativos por meio de um navegador da web ou um aplicativo móvel, sem precisar instalar o software em seus próprios dispositivos.

1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Cloud Computing:** Existem três principais modelos de serviços em nuvem (cloud computing):



1.3 Classificação de Sistemas Distribuídos

1.3.1 Computação distribuída de alta performance.

- **Cloud Computing:** Algumas das principais plataformas de cloud computing incluem:
 - Amazon Web Services (AWS)
 - Microsoft Azure
 - Google Cloud Platform (GCP)



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

- Próxima geração emergente de sistemas distribuídos nos quais os nós são pequenos, móveis, e muitas vezes incorporado em um sistema maior, caracterizado pelo fato de que o sistema se mistura naturalmente ao ambiente do usuário.
- Três subtipos (sobrepostos)
 - Sistemas computacionais ubíquos: difundidos e continuamente presentes, ou seja, existe uma interação contínua entre sistema e usuário.
 - Sistemas de computação móvel: difundidos, mas a ênfase está no fato de que dispositivos são inherentemente móveis.
 - Redes de sensores (e atuadores): pervasivas, com ênfase (colaborativa) no sensoriamento e atuação no ambiente



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

- **Computação Ubíqua:** principais características:
 - **Distribuição:** Os dispositivos são ligados em rede, distribuídos e acessíveis em maneira transparente
 - **Interação:** Interação entre usuários e dispositivos é altamente discreta
 - **Conhecimento de contexto:** O sistema está ciente do contexto de um usuário para otimizar a interação
 - **Autonomia:** Dispositivos operam de forma autônoma sem intervenção humana, e são, portanto, altamente autogeridos
 - **Inteligência** O sistema como um todo pode lidar com uma ampla gama de ações dinâmicas e interações
 - Microsoft: Productivity Future Vision
 - <https://www.youtube.com/watch?v=w-tFdreZB94>

1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

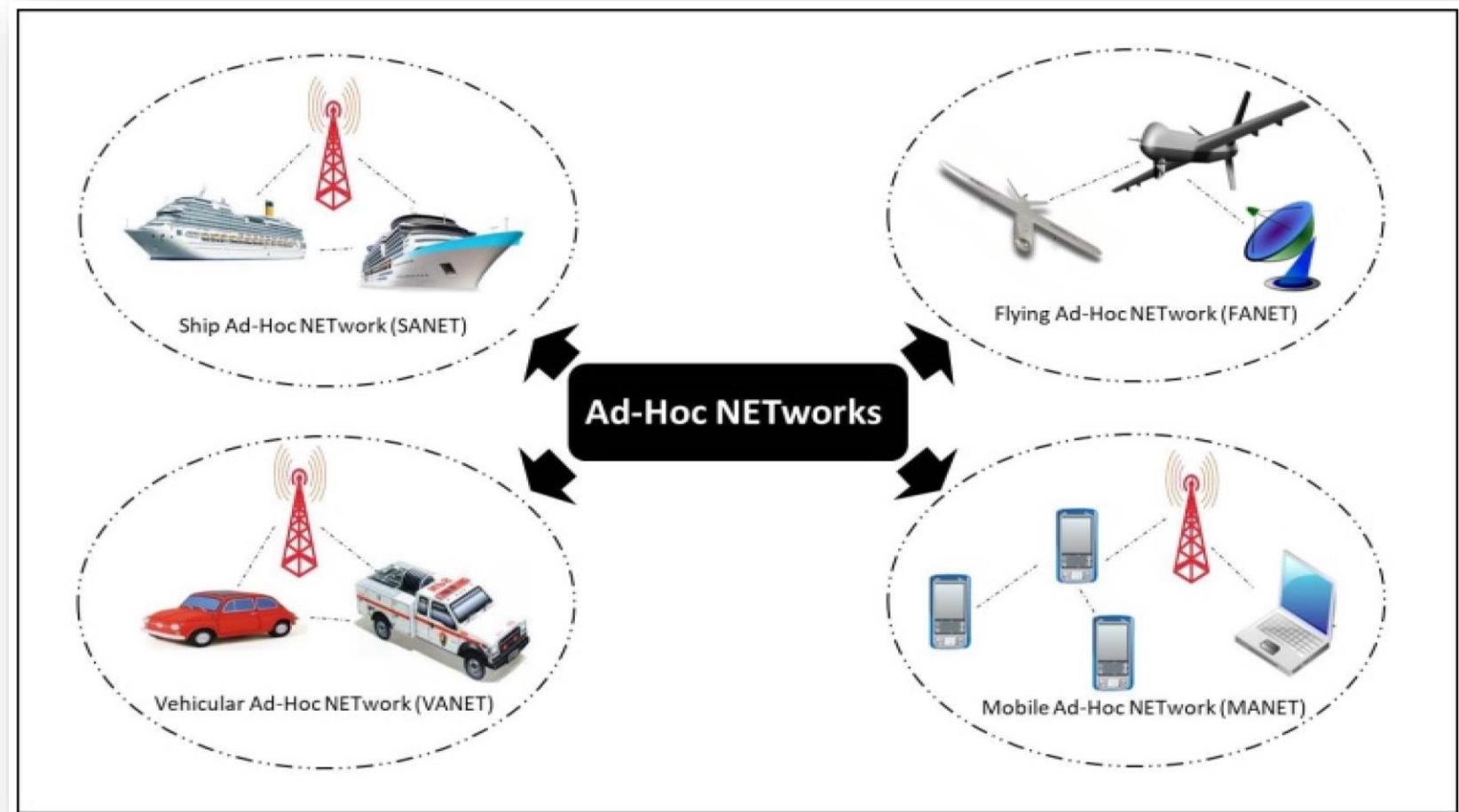
- **Computação Móvel:** Uma miríade de diferentes dispositivos móveis (smartphones, tablets, dispositivos GPS, controles remotos)
 - Móvel implica que a localização de um dispositivo deve mudar com o tempo
 - Mudança de serviços locais, acessibilidade, etc. **Palavra-chave: descoberta**
 - A comunicação pode se tornar mais difícil: nenhuma rota estável, mas também talvez nenhuma conectividade garantida



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

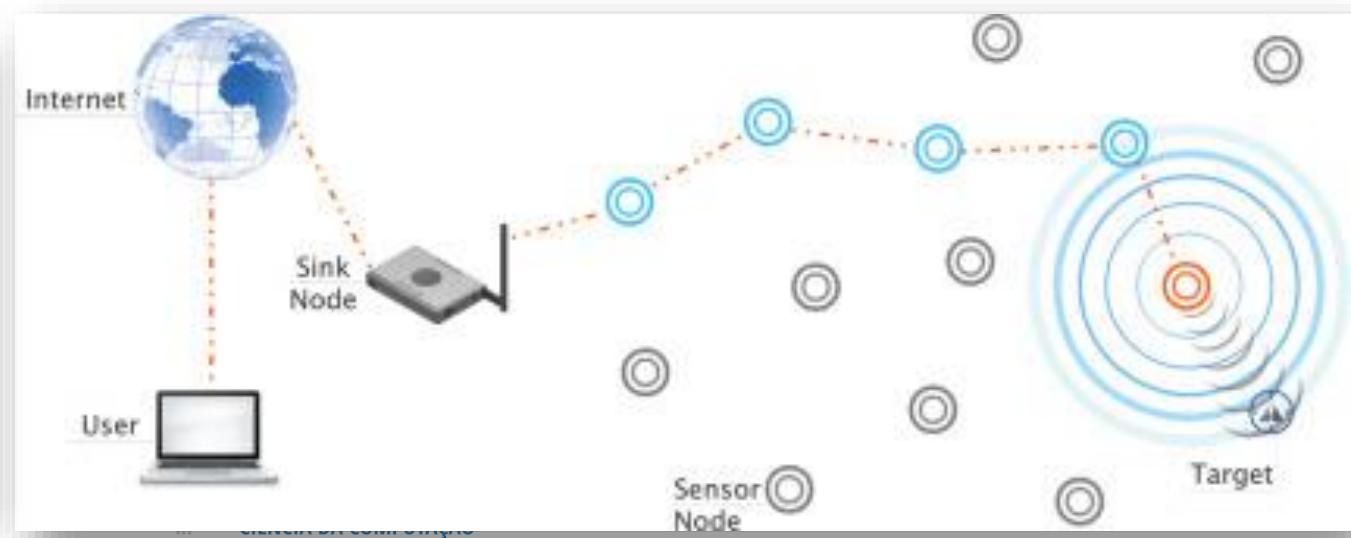
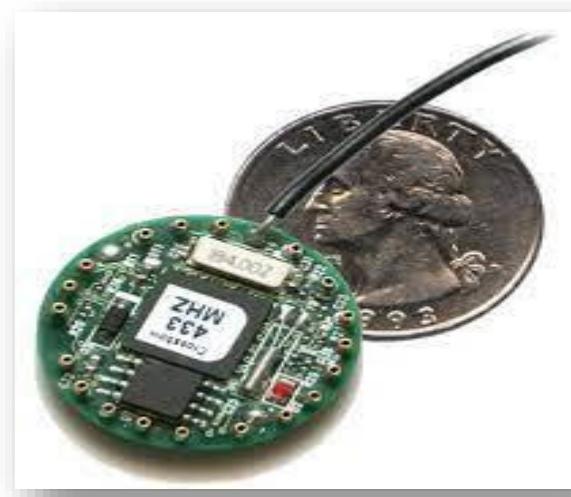
- Computação Móvel:



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

- **Redes de Acesso Móveis e Sem Fio:** São redes compostas por um grande número de dispositivos sensores distribuídos em uma área geográfica específica
 - Esses dispositivos são capazes de capturar informações sobre o ambiente ao seu redor (como temperatura, umidade, luminosidade, etc.) e transmitir esses dados para um sistema central de processamento
 - Essa tecnologia é utilizada em diversas áreas, como monitoramento ambiental, agricultura de precisão, automação industrial, entre outras



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

- **Internet das Coisas (IoT)**: é uma rede de objetos físicos conectados à Internet, que coletam e compartilham dados sem intervenção humana direta
 - Dispositivos sensores e atuadores coletam dados e agem sobre o ambiente físico
 - Esses dados são transmitidos através de redes de comunicação sem fio, como Wi-Fi, Bluetooth e RFID



1.3 Classificação de Sistemas Distribuídos

1.3.2 Sistemas Pervasivos

- **Internet das Coisas (IoT):** Existem muitas aplicações de IoT disponíveis no mercado atualmente. Alguns exemplos populares incluem:
- Smart Home: sistemas de automação residencial que permitem controlar remotamente a temperatura, iluminação, fechaduras e eletrodomésticos, por meio de dispositivos conectados
- Wearables: dispositivos vestíveis, como smartwatches e pulseiras de fitness, que coletam dados biométricos, como frequência cardíaca, níveis de atividade física e sono
- Cidades Inteligentes: sensores e dispositivos conectados que coletam e analisam dados sobre o tráfego, qualidade do ar, iluminação pública, segurança e muito mais, para melhorar a qualidade de vida dos moradores da cidade
- Agricultura de precisão: sensores e sistemas de monitoramento que ajudam os agricultores a tomar decisões informadas sobre irrigação, fertilização, controle de pragas e muito mais, para melhorar a produtividade e reduzir o desperdício
- Indústria 4.0: sensores, robôs e sistemas de automação que permitem a monitoração e controle remoto de processos industriais, melhorando a eficiência e reduzindo custos

Fim!



obrigado!