

30-769

Sistemas Distribuídos

MSc. Fernando Schubert

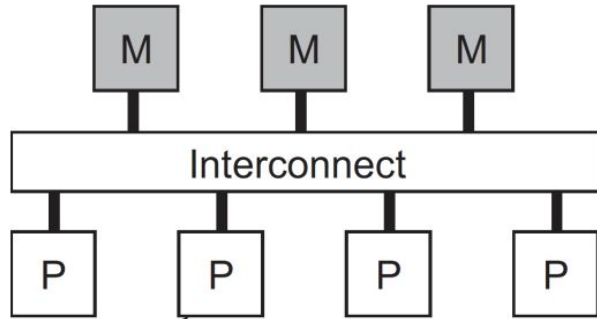
TIPOS DE SISTEMAS DISTRIBUÍDOS

- **Sistemas de Computação Distribuídos de Alto Desempenho**
 - Utilizado para tarefas de computação de alto desempenho
- **Sistemas de Informação Distribuídos**
 - Encontrados em organizações onde uma profusão de aplicações em rede precisa se integrar
 - Interoperabilidade
- **Sistemas Embutidos Distribuídos**
 - São caracterizados por sua mobilidade

Sistemas de Computação Distribuídos de Alto Desempenho

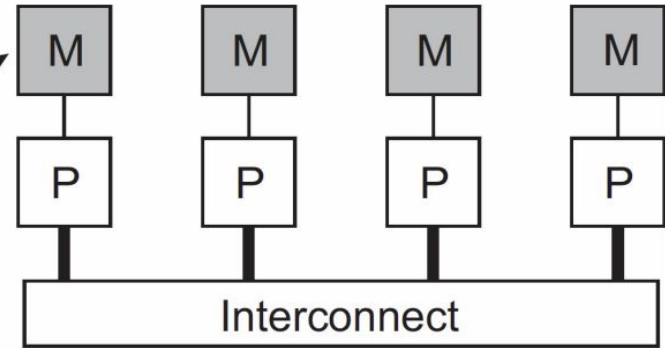
- Começaram com computação paralela

Memória compartilhada



Processador

Memória distribuída

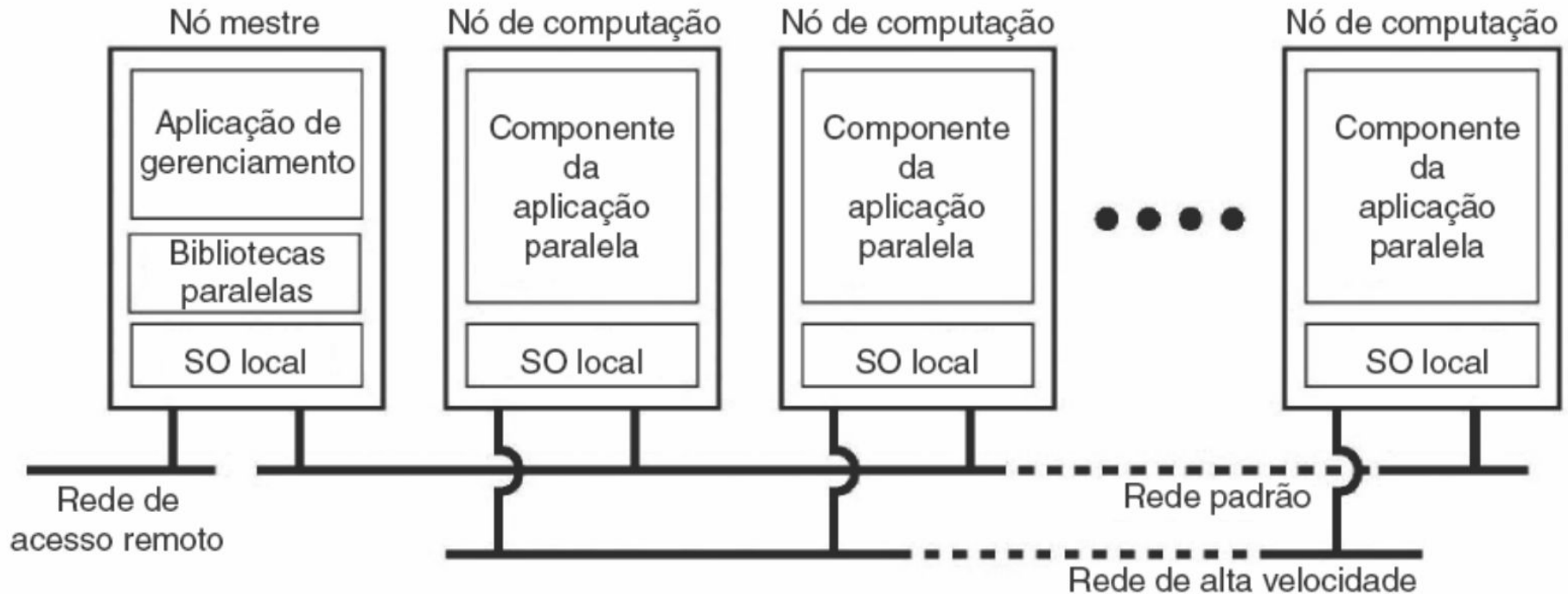


Memória

Sistemas de Computação Distribuídos de Alto Desempenho

- Cluster
 - Se tornaram populares quando a razão preço/desempenho de computadores pessoais e estações de trabalho melhorou
 - Construção de supercomputadores que usam solução de prateleira, conectando vários computadores em uma rede de alta velocidade.
 - É usado para programação paralela no qual um único programa, intensivo em computação, pode ser executado em paralelo em várias máquinas.

Sistemas de Computação Distribuídos de Alto Desempenho



Sistemas de Computação Distribuídos de Alto Desempenho

- Cluster:
 - Conjunto de nós controlados e acessados por um único nó mestre
 - Tarefas típicas do mestre são manipular a alocação de nós a determinado programa paralelo, manter fila de jobs, e fornecer uma interface para o usuário
 - O mestre pode ser responsável por executar um middleware para a execução de programas e o gerenciamento do Cluster
 - Para os nós de computação, basta um SO padrão

Sistemas de Computação Distribuídos de Alto Desempenho

- Grades Computacionais
 - Compostas por recursos heterogêneos conectados por redes de diferentes capacidades
 - Compartilhadas, executando tanto aplicações locais quanto remotas
- Dinâmicas, recursos podem se juntar ou deixar a grade sem aviso prévio
- Recursos podem estar geograficamente distantes

Sistemas de Computação Distribuídos de Alto Desempenho

- Principais diferenças: clusters X grades
 - Recursos heterogêneos
 - alta dispersão geográfica (escala mundial)
 - Compartilhamento
 - múltiplos domínios administrativos
 - controle totalmente distribuído

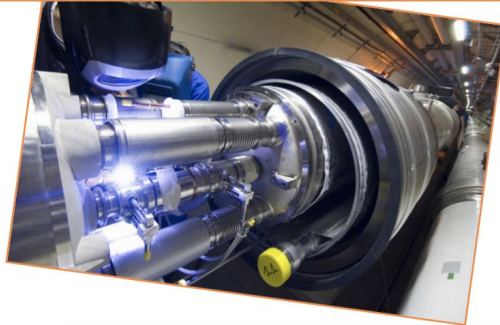
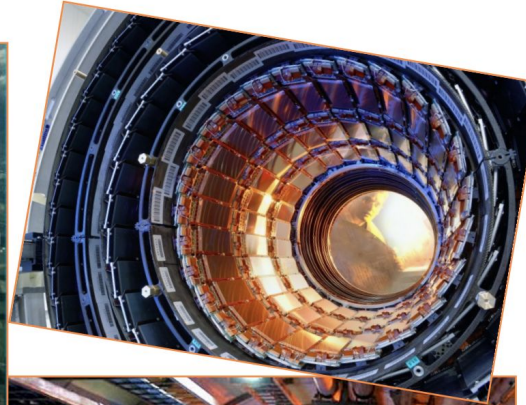
Sistemas de Computação Distribuídos de Alto Desempenho

- O conceito de grades é análogo ao de redes de energia elétrica
- Fornecer energia de acordo com a demanda de qualquer usuário, não importando a sua localização
- O sonho do cientista (The Grid Vision)
 - Oferecer desempenho computacional eficientemente;
 - De acordo com a demanda;
 - A um custo razoável;
 - Para qualquer um que precisar.

LHC (LARGE HADRON COLLIDER)

- O Grande Colisor de Hádrõs do CERN
- É o maior acelerador de partículas do mundo e também o de maior energia existente
- O laboratório localiza-se em um túnel de 27 km de circunferência, a 175 metros abaixo do nível do solo na fronteira da França com Suíça

LHC (LARGE HADRON COLLIDER)





LHC (LARGE HADRON COLLIDER)

- Produz por ano 50 a 70 petabytes de dados

A grade computacional LHC possui 1.7 milhões de cores e 1,5 EXAbytes de armazenamento distribuídos em 170 locais e 42 países

<https://www.home.cern/science/computing/grid>

- The project LHC@home foi criado para calibrar a construção do acelerador de partículas
- Permite que usuários cedam tempo ocioso de seus computadores para executar simulações de como as partículas iriam viajar nos túneis do LHC

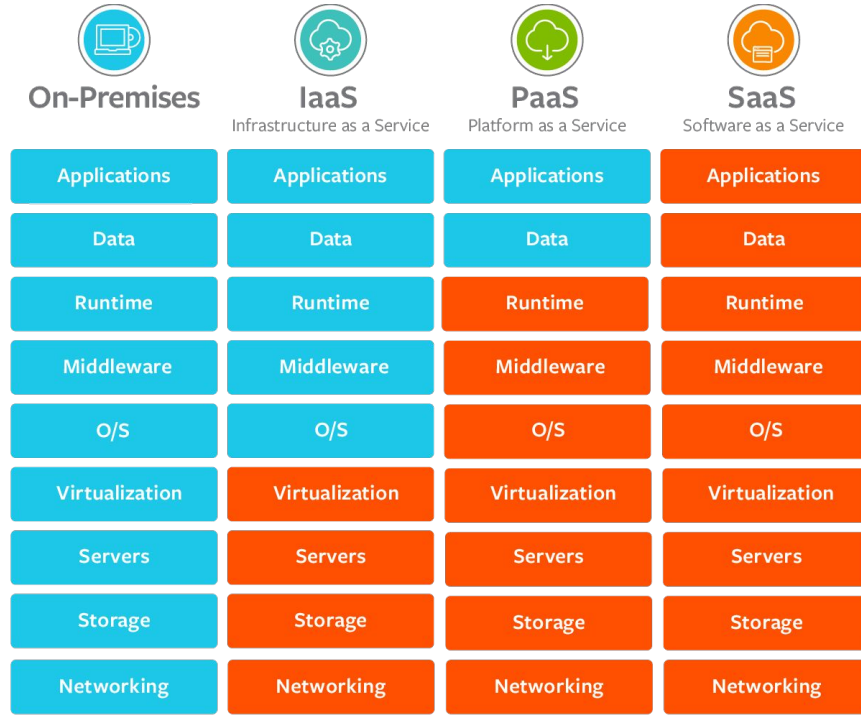
SETI@HOME

- Objetiva usar recursos de computação em uma rede de área ampla, ou mesmo a Internet, quando os recursos de computação locais são escassos.
- Por exemplo:
 - SETI @ home (setiathome.berkeley.edu) utiliza mais de 330 mil computadores obtendo um poder computacional de mais de 528 TeraFLOPS (04 de agosto de 2008)
 - Projeto atualmente em hibernação.

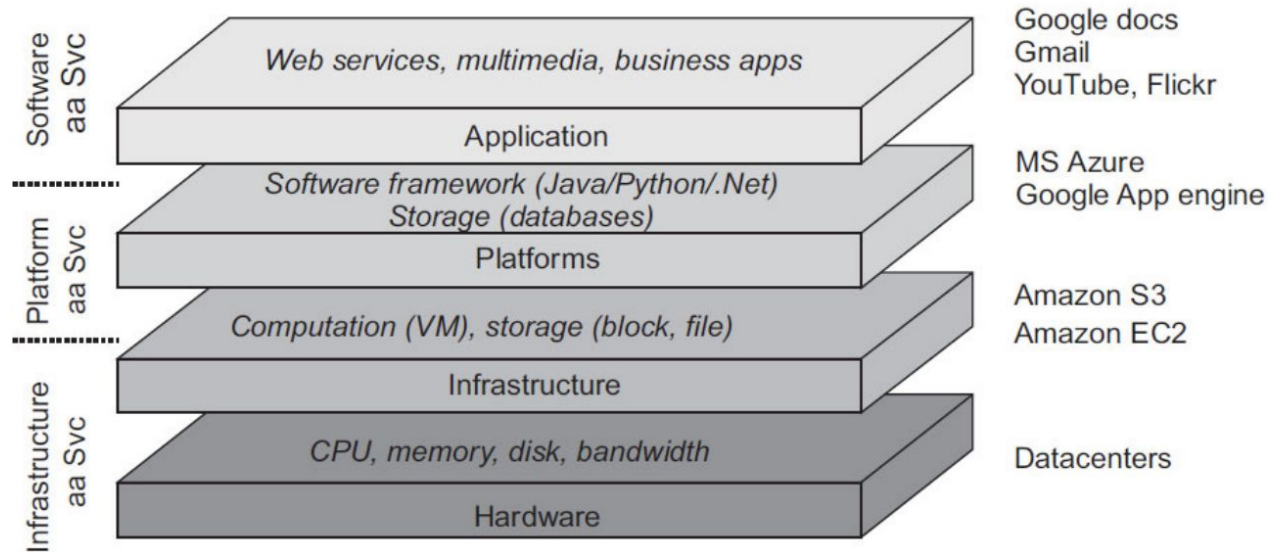
NUVENS COMPUTACIONAIS

- Cloud computing (computação em nuvens)
 - É um modelo de utilização e entrega de recursos
 - Obter recursos (hardware, software) através da rede
- A rede de fornecimento de recursos é chamada de 'nuvem'.
 - O recurso na 'nuvem' parece infinitamente escalável e pode ser utilizado sempre
- Exemplos: Amazon, GCP, Microsoft Azure

NUVENS COMPUTACIONAIS



NUVENS COMPUTACIONAIS



NUVENS COMPUTACIONAIS

- **Hardware:** Processadores, roteadores, sistemas de energia e resfriamento. Usuários normalmente não vêem estes recursos diretamente.
- **Infraestrutura:** Utilizam técnicas de virtualização. Envolve a alocação e o gerenciamento de dispositivos de armazenamento e servidores virtuais.
- **Plataforma:** Fornece abstrações de nível superior para armazenamento e outros recursos computacionais como um sistema operacional. E fornece APIs que incluem chamadas para fazer upload e execução dos programas na nuvem.
 - **Exemplo:** o sistema de armazenamento do Amazon S3 oferece uma API para que os arquivos (criados localmente) sejam organizados e armazenados nos chamados buckets, que são armazenados automaticamente na nuvem.
- **Aplicação:** aplicativos reais, como suítes de escritório (processadores de texto, aplicativos de planilhas, aplicativos de apresentação). Comparável ao pacote de aplicativos enviados com sistemas operacionais

NUVENS COMPUTACIONAIS

- Como resultado, a computação em nuvem modificou a indústria de software inteiramente
- As aplicações passaram a rodar na rede ao invés de estarem no desktop de um usuário
- Afeta arquitetura como desenvolver, implementar, executar e entregar os pedidos

NUVENS COMPUTACIONAIS

- Algumas Diferenças entre Grid e Cloud computing
 - A computação em grades tem foco no compartilhamento de recursos.
 - Todo nó do grade pode requisitar recursos de outros nós.
 - A computação nas nuvens possui ênfase na propriedade
 - Em uma nuvem cada usuário pode requerer seu próprio recurso privado a um serviço de provedor específico. Não é necessário que o usuário contribua com recursos
 - No modelo de negócios de uma nuvem, o usuário paga pelo uso dos recursos/serviços

COMPUTAÇÃO COMO UTILIDADE

- Computação em clusters, grades e nuvens
- Já caminharam no sentido de estabelecer a visão de computação como utilidade.
- Com o grande avanço na área de comunicação e tecnologia da informação
- Existe uma idéia de que a computação seja no futuro um serviço básico essencial
- Água, Gás, Eletricidade, Telefonia e Computação

SISTEMAS DE INFORMAÇÃO DISTRIBUÍDOS

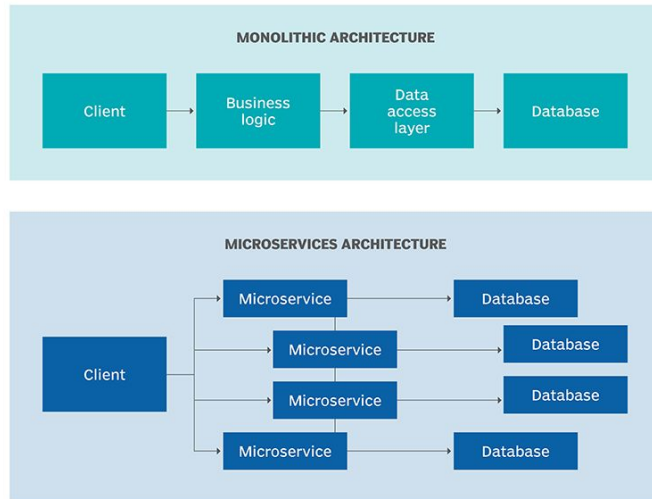
- Exemplo:
 - Uma aplicação de rede consistia simplesmente em um servidor que executava a aplicação e a disponibilizava para clientes remotos
 - Clientes enviavam requisições para o servidor e recebiam uma resposta
- A integração em um nível mais baixo, permitiria que clientes empacotassem várias requisições e enviassem para execução como uma transação distribuída em vários servidores
- À medida que as aplicações se tornaram mais complexas e separadas em componentes independentes, tornou-se necessário que as aplicações também pudessem se comunicar entre si

SISTEMAS DE INFORMAÇÃO DISTRIBUÍDOS

- Integração de Aplicações Empresariais (EAI)
 - Processamento de Transações Distribuídas
 - Modelos diferentes de comunicação entre aplicações
- Transações em bancos de dados
 - Programar a utilização de transações requer primitivas especiais que devem ser fornecidas pelo SD ou pelo sistema de linguagem em tempo de execução

SISTEMAS DE INFORMAÇÃO DISTRIBUÍDOS

Microservices vs. monolithic architecture

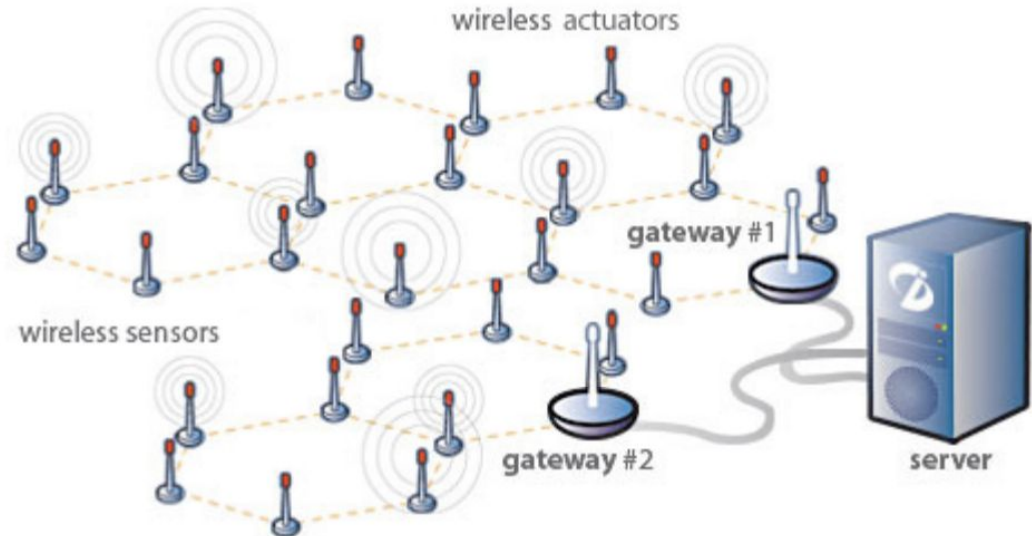


SISTEMAS DISTRIBUÍDOS PERVASIVOS

- Sistemas decorrentes do uso de computação móvel e embutida
comportamento instável é esperado
- Pequeno tamanho
- Alimentados por bateria
- Comunicação sem fio
- Tem como objetivo a mobilidade
- Um dispositivo deve adotar mudanças contextuais

SISTEMAS DISTRIBUÍDOS PERVASIVOS

- Exemplo: redes de sensores
- Monitoramento Ambientais
- Monitoramento de tráfego
- Fins Militares
- Aplicações Médicas



Modelos de Sistema

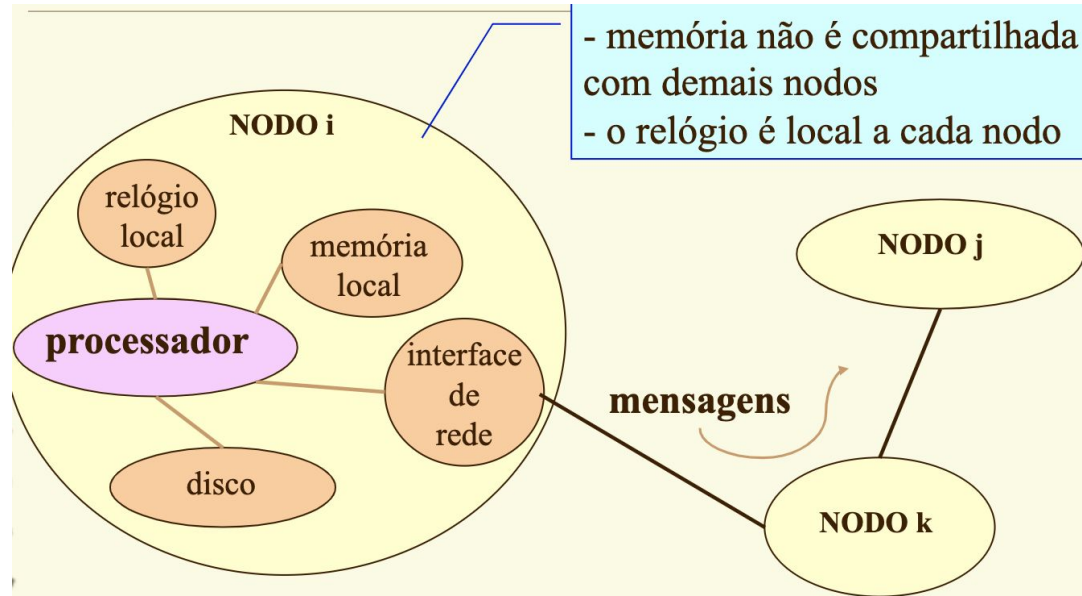
- Os **modelos físicos** consideram os tipos de computadores e equipamentos que constituem um sistema e sua interconectividade, sem os detalhes das tecnologias específicas.
- Os **modelos de arquitetura** descrevem um sistema em termos das tarefas computacionais e de comunicação realizadas por seus elementos computacionais – os computadores individuais ou conjuntos deles interligados por conexões de rede apropriadas. Os modelos cliente-servidor e peer-to-peer são duas das formas mais usadas de arquitetura para sistemas distribuídos.
- Os **modelos fundamentais** adotam uma perspectiva abstrata para descrever soluções para os problemas individuais enfrentados pela maioria dos sistemas distribuídos.

Modelos Físicos

- **Conceito:** Um modelo físico é uma representação dos elementos de hardware de um sistema distribuído, de maneira a abstrair os detalhes específicos do computador e das tecnologias de rede empregadas.
- **Componentes:**
 - Componentes de Hardware
 - Topologia de Rede
 - Data Centers
 - Distribuição Geográfica
 - Tolerância a Falhas e Redundância
 - Consumo de Energia e Eficiência Energética

Modelos Físicos

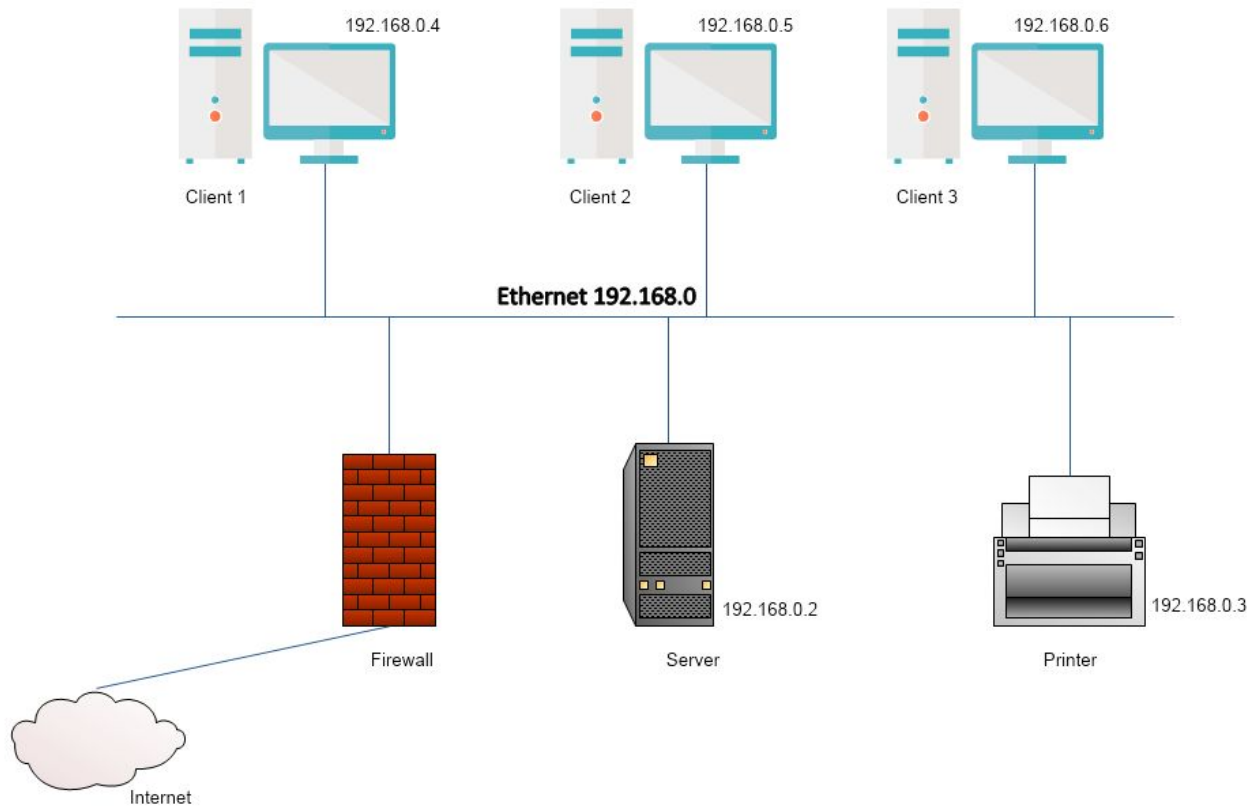
- **Modelo físico básico:** um modelo físico mínimo de um sistema distribuído é aquele que um conjunto extensível de nós de computador interconectados por uma rede de computadores necessária para a passagem de mensagens.



Modelos Físicos

- **Sistemas distribuídos primitivos:** anos 70-80 e início dos anos 90.
Possíveis devido à evolução das redes locais, tamanho variável entre 10 e 100s de nós interconectados com serviços compartilhados de e-mail, arquivos e impressão (cliente-servidor).

Ethernet LAN Diagram



Modelos Físicos

- **Sistemas distribuídos adaptados para a Internet:** aproveitando essa base, sistemas distribuídos de maior escala começaram a surgir nos anos 1990, em resposta ao enorme crescimento da Internet. Ênfase em padrões abertos e middleware como CORBA e serviços Web.



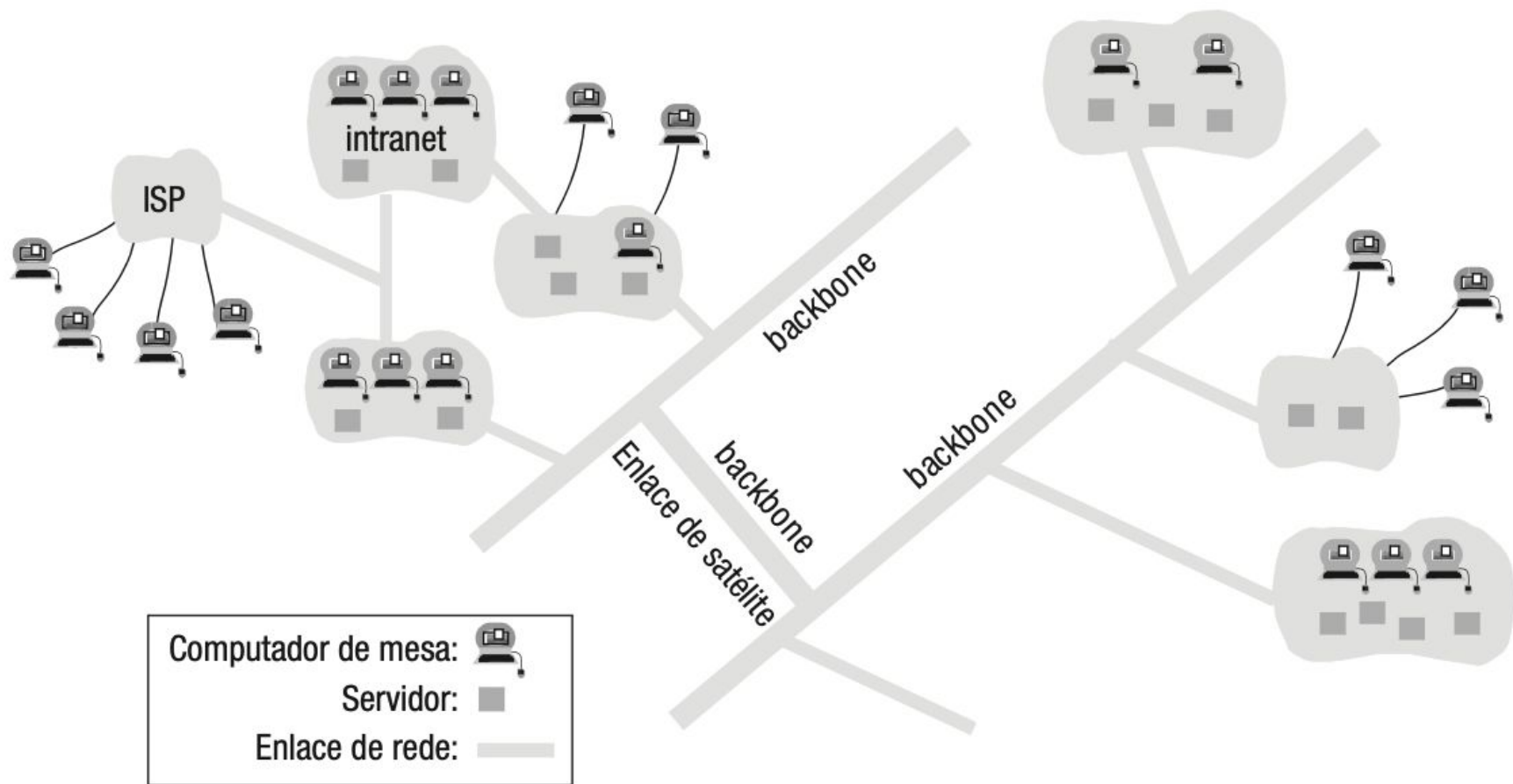
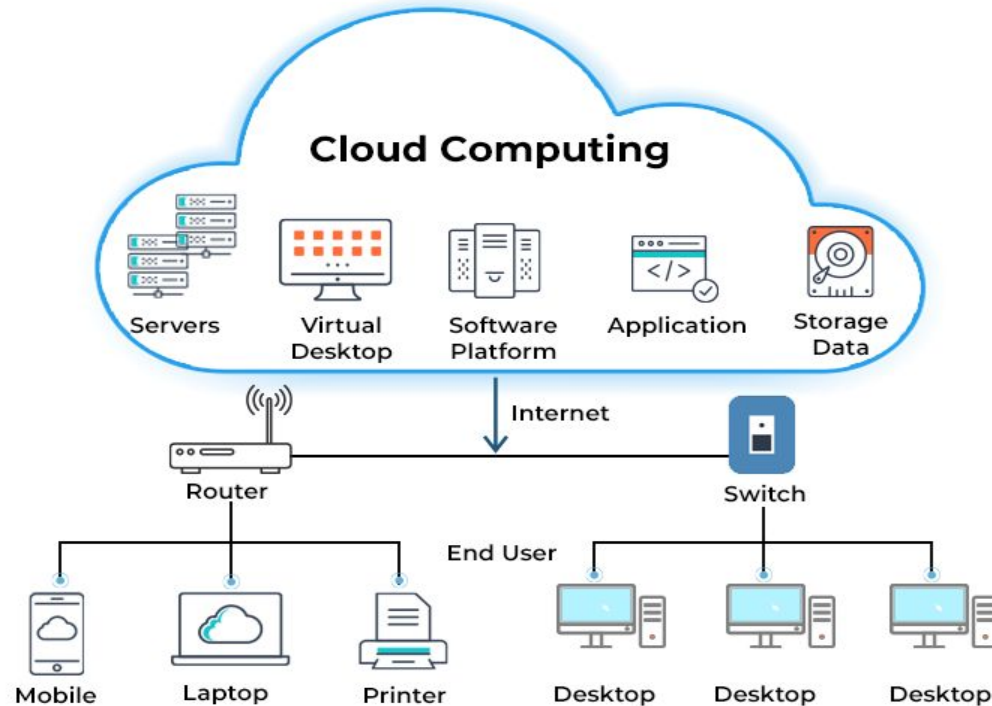


Figura 1.3 Uma parte típica da Internet.

Modelos Físicos

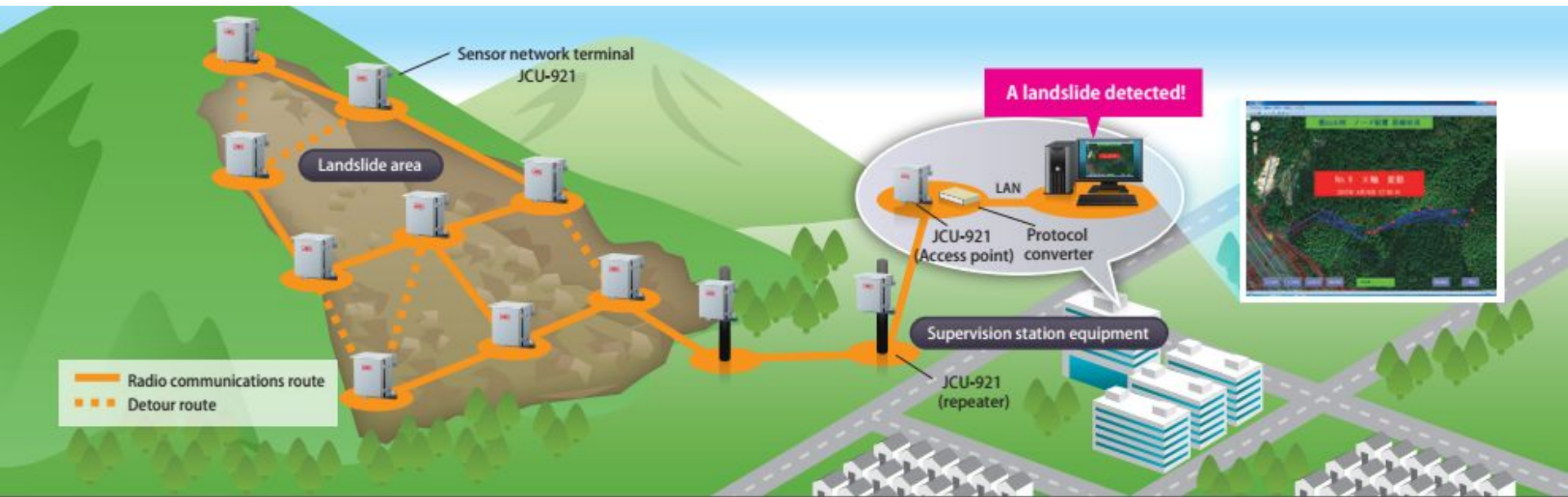
- **Sistemas distribuídos contemporâneos:** nos sistemas anteriores, os nós normalmente eram computadores de mesa e, portanto, relativamente estáticos, separados e autônomos.
 - Computação móvel - necessidade de descoberta de serviços e suporte para operação conjunta espontânea
 - Computação ubíqua: nós heterogêneos de arquiteturas não convencionais como geladeiras, lavadoras, etc.
 - Computação em nuvem e em agregados (clusters): conjunto de nós oferecendo um serviço

CLOUD COMPUTING ARCHITECTURE



Modelos Físicos

- **Sistemas distribuídos de sistemas:** um sistema complexo constituído em uma série de subsistemas distribuídos, eles próprios sistemas que se reúnem para executar uma ou mais tarefas em particular.



Modelos Físicos

<i>Sistemas distribuídos</i>	<i>Primitivos</i>	<i>Adaptados para Internet</i>	<i>Contemporâneos</i>
<i>Escala</i>	Pequenos	Grandes	Ultragrandes
<i>Heterogeneidade</i>	Limitada (normalmente, configurações relativamente homogêneas)	Significativa em termos de plataformas, linguagens e <i>middleware</i>	Maiores dimensões introduzidas, incluindo estilos de arquitetura radicalmente diferentes
<i>Sistemas abertos</i>	Não é prioridade	Prioridade significativa, com introdução de diversos padrões	Grande desafio para a pesquisa, com os padrões existentes ainda incapazes de abranger sistemas complexos
<i>Qualidade de serviço</i>	Em seu início	Prioridade significativa, com introdução de vários serviços	Grande desafio para a pesquisa, com os serviços existentes ainda incapazes de abranger sistemas complexos

Figura 2.1 Gerações de sistemas distribuídos.

Modelos de arquitetura para sistemas distribuídos

- A arquitetura de um sistema é sua estrutura em termos de componentes especificados separadamente e suas inter-relações.
 - O objetivo global é garantir que a estrutura atenda às demandas atuais e, provavelmente, às futuras demandas.
 - As maiores preocupações são tornar o sistema confiável, gerenciável, adaptável e rentável.



Modelos de arquitetura para sistemas distribuídos

- Elementos arquitetônicos
 - Quais são as entidades que estão se comunicando no sistema distribuído?
 - Como elas se comunicam ou, mais especificamente, qual é o paradigma de comunicação utilizado?
 - Quais funções e responsabilidades (possivelmente variáveis) estão relacionadas a eles na arquitetura global?
 - Como eles são mapeados na infraestrutura distribuída física (qual é sua localização)?

Modelos de arquitetura para sistemas distribuídos

- Elementos arquitetônicos
 - Quais são as entidades que estão se comunicando no sistema distribuído?
 - Como elas se comunicam ou, mais especificamente, qual é o paradigma de comunicação utilizado?
 - Quais funções e responsabilidades (possivelmente variáveis) estão relacionadas a eles na arquitetura global?
 - Como eles são mapeados na infraestrutura distribuída física (qual é sua localização)?
- Componente é uma unidade modular com interfaces requeridas e fornecidas e bem definidas que é substituível dentro de seu ambiente.

Modelos de arquitetura para sistemas distribuídos

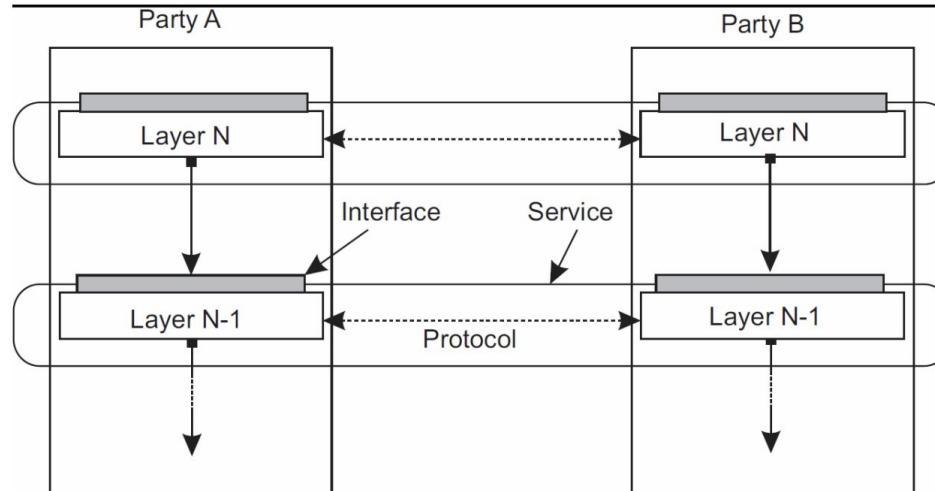
- Conector é um mecanismo que serve como mediador da comunicação ou da cooperação entre componentes.
- Usando componentes e conectores, podemos chegar a várias configurações, que foram classificadas em estilos arquitetônicos.
 - Arquitetura em camadas
 - Arquitetura baseada em objetos e orientada a serviços
 - Arquitetura centrada em recursos
 - Arquitetura baseada em eventos

Modelos de arquitetura para sistemas distribuídos

- Arquitetura em camadas
- Os componentes são organizados em camadas
- Cada componentes da camada Li pode chamar um componente da camada Li-1, mas não o contrário
- Modelo amplamente adotado pela comunidade de redes
- O controle flui de camada para camada
 - As requisições descem pela hierarquia
 - Os resultados fluem para cima

Modelos de arquitetura para sistemas distribuídos

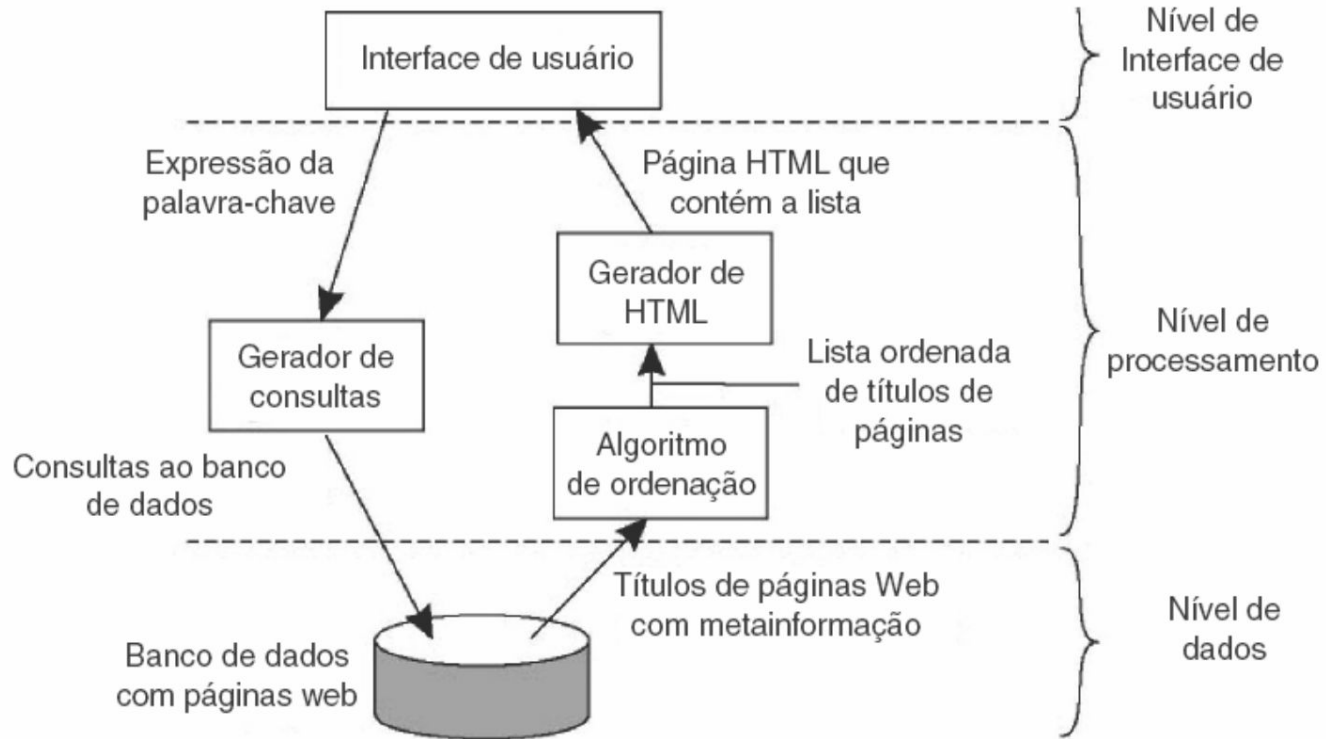
Exemplo de Arquitetura em camadas Protocolo de Redes

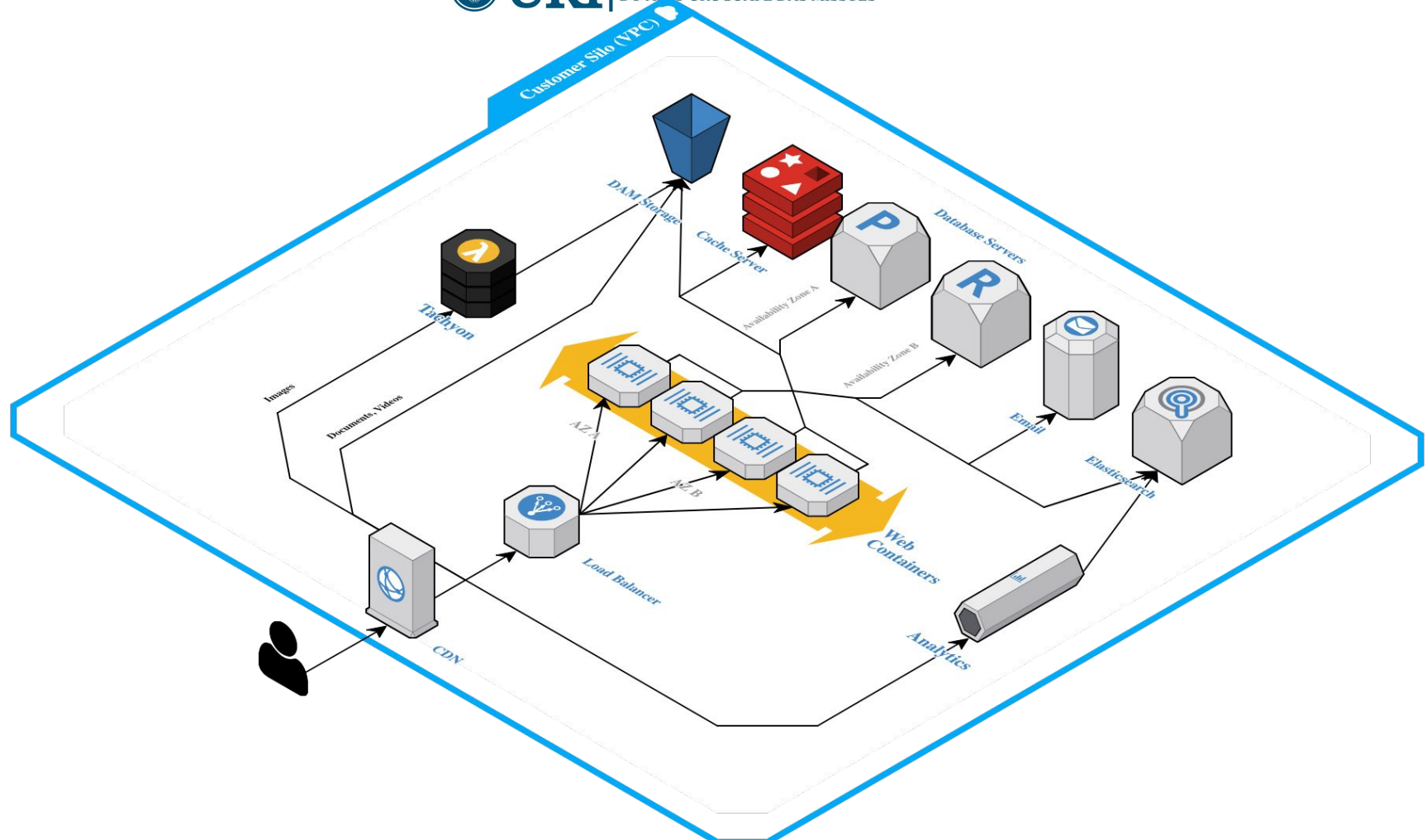


Modelos de arquitetura para sistemas distribuídos

- Aplicação em camadas
- Visão tradicional em 3 camadas
 - Nível de interface contém componentes para fazer a interface da aplicação com usuários ou aplicações externas
 - Nível de processamento contém as funções da aplicação, sem dados específicos
 - Nível de dados contém os dados que o cliente deseja manipular através da aplicação

Modelos de arquitetura para sistemas distribuídos





Modelos de arquitetura para sistemas distribuídos

- Arquiteturas baseadas em **objetos e orientadas a serviços**
- Objetos correspondem às definições de componentes
- Objetos (componentes) são conectados por meio de chamadas de procedimento remotas (RPCs)
- Um tipo de organização mais solta
- Arquiteturas em Camadas e Baseadas em Objetos são os estilos mais importantes para sistemas de SW de grande porte.

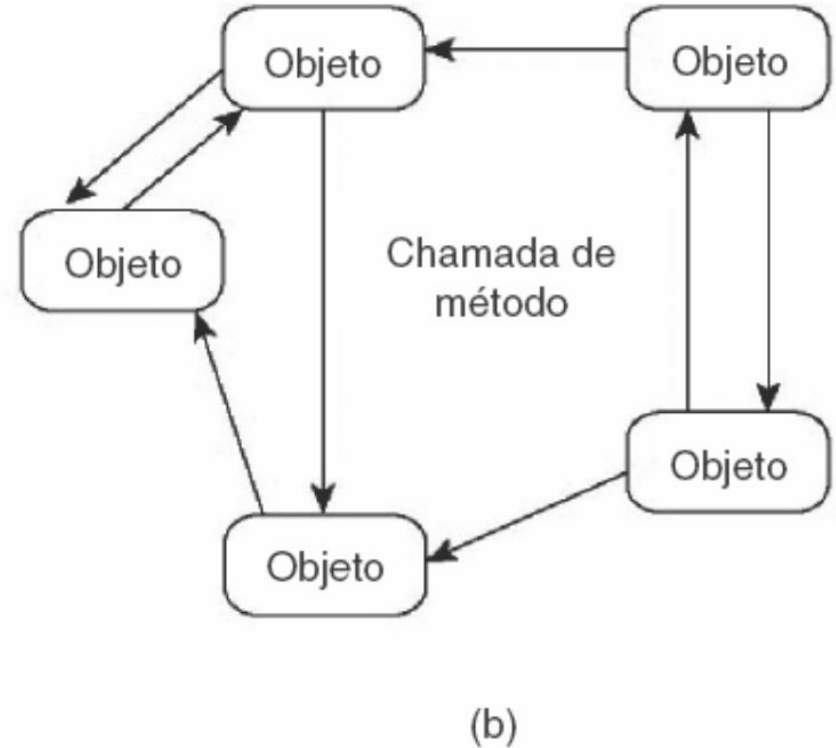
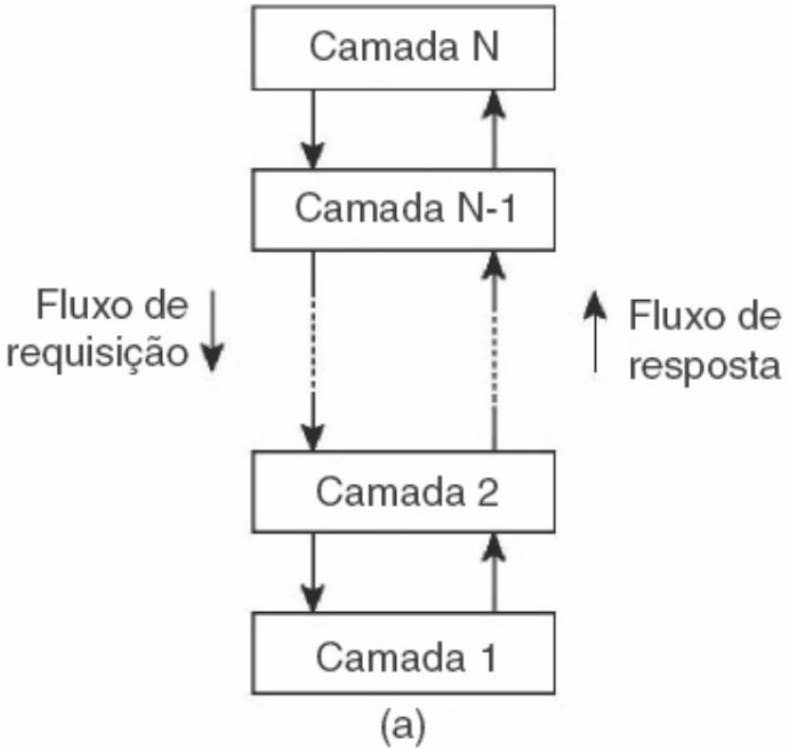


Figura 2.1 Estilo arquitetônico (a) em camadas e (b) baseado em objetos.

Modelos de arquitetura para sistemas distribuídos

- Arquiteturas centradas em recursos (RESTful)
- O sistema distribuído é visto como uma coleção de recursos gerenciados individualmente pelos componentes. Os recursos podem ser adicionados, removidos, recuperados e modificados por aplicações (remotas)
 - 1. Os recursos são identificados por um único esquema de nomeação
 - 2. Todos os serviços oferecem a mesma interface
 - 3. As mensagens enviadas para um serviço ou recebidas do serviço são totalmente auto-contidas
 - 4. Após a execução de uma operação em um serviço, o componente esquece tudo sobre o chamador



Rest API Basics

Typical HTTP Verbs:

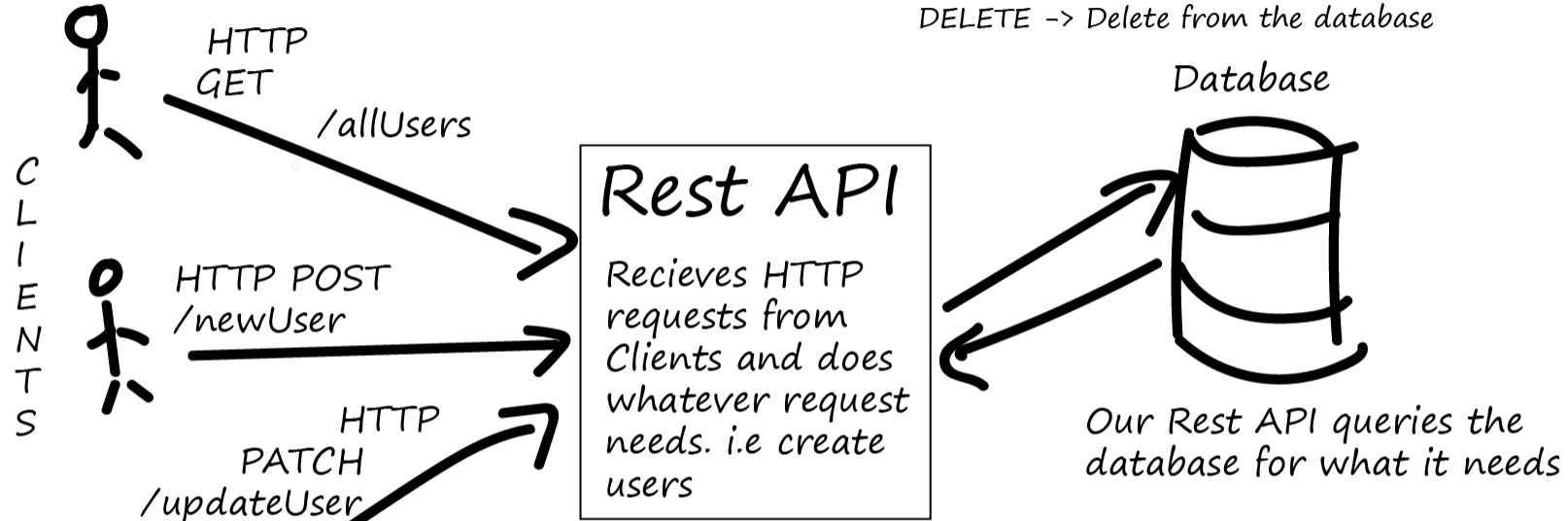
GET -> Read from Database

PUT -> Update/Replace row in Database

PATCH -> Update/Modify row in Database

POST -> Create a new record in the database

DELETE -> Delete from the database



Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

Modelos de arquitetura para sistemas distribuídos

O que são APIs REST?

- API significa Interface de Programação de Aplicativos.
- REST significa Transferência de Estado Representacional.
- APIs REST são um conjunto de diretrizes arquiteturais para a criação de serviços web escaláveis e interoperáveis.

Modelos de arquitetura para sistemas distribuídos

Como Funcionam as APIs REST?

- Baseado em Recursos: Cada recurso (como usuários, posts, etc.) é acessado através de um URI único.
- Operações sobre Recursos: Utiliza os métodos HTTP para operações sobre os recursos (GET, POST, PUT, DELETE, etc.).
- Representação dos Recursos: Os recursos são representados geralmente em formato JSON ou XML.
- Estado do Servidor: As APIs REST são stateless, o que significa que cada requisição do cliente deve conter todas as informações necessárias para entender e processar a requisição.

Modelos de arquitetura para sistemas distribuídos

Exemplo de arquitetura centrada em recursos

- Sistema de armazenamento da Amazon (S3)
- Objetos (arquivos) são colocados em buckets (diretórios).
- Buckets não podem ser colocados em buckets. Operações em ObjectName bucket BucketName requerem o seguinte identificador (URI):
 - <http://BucketName.s3.amazonaws.com/ObjectName>
- Todas as operações são executadas através de pedidos HTTP:
- Cria um bucket/objeto: POST URI
- Lista objetos: GET no nome do diretório
- Lê um objeto: GET URI

Modelos de arquitetura para sistemas distribuídos

Exemplo de arquitetura centrada em recursos

- Muitos usuários gostam dessa abordagem RESTful porque a interface para o serviço é simples. O problema é que precisa muito trabalho na identificação dos parâmetros. Amazon provê uma interface baseada em serviços.

Amazon S3 SOAP interface

Bucket operations	Object operations
ListAllMyBuckets	PutObjectInline
CreateBucket	PutObject
DeleteBucket	CopyObject
ListBucket	GetObject
GetBucketAccessControlPolicy	GetObjectExtended
SetBucketAccessControlPolicy	DeleteObject
GetBucketLoggingStatus	GetObjectAccessControlPolicy
SetBucketLoggingStatus	SetObjectAccessControlPolicy

Modelos de arquitetura para sistemas distribuídos

Arquiteturas baseadas em eventos

- Processos se comunicam por meio de propagação de eventos
- Opcionalmente estes eventos podem transportar dados
- Sistemas Publisher/Subscriber
 - Processos publicam eventos
 - O middleware assegura que somente os processos assinantes irão receber esses eventos.
 - Os processos são fracamente acoplados
 - Processos não precisam se referir especificamente uns aos outros

Modelos de arquitetura para sistemas distribuídos

