

# Arquiteturas Orientadas a Serviços e Computação em Nuvem

Julio Cezar Estrella

Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo  
email: [jcezar@icmc.usp.br](mailto:jcezar@icmc.usp.br)

# Roteiro I

## 1 Introdução

- Mudança de Foco
- Evolução dos Negócios
- Problemas
- Evolução das Tecnologias

## 2 SOA

- Visão Geral
- Estrutura e Terminologia

## 3 Web Services

- Visão Geral
- Arquitetura Básica
- Tecnologias
- XML
- SOAP
- WSDL
- UDDI

# Roteiro II

- Abstração

## 4 Composição de Web Services

- Visão Geral
- Arquitetura Básica

## 5 Cloud

- Visão Geral

## 6 Conclusão

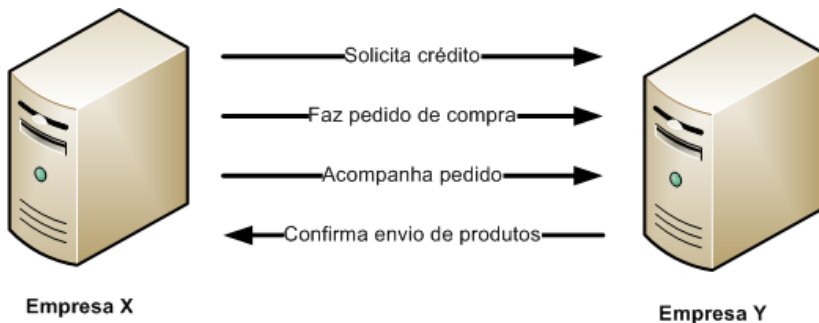
# Mudança de Foco

- A Internet mudou a maneira de se fazer negócios
- Facilidade para acesso às aplicações
- HTML e HTTP → Padrão de troca de informação
- Navegadores web tornaram a comunicação simples e possível

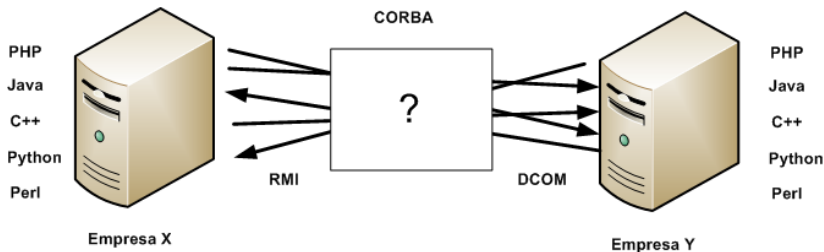
# Evolução dos Negócios

- Evolução dos negócios
- Necessidade de novas soluções
- Transações e trocas de informações tornaram-se essenciais para os negócios
- Automatização nos dias atuais é mais que fundamental

# Evolução dos Negócios



# Evolução dos Negócios



# Problemas

- Há dois problemas que são emergentes [Endrei et al., 2004]:
  - 1 Dificuldades para integrar sistemas heterogêneos;
    - Diferentes empresas utilizam diferentes tecnologias e soluções
    - Variados domínios;
    - Muitos fabricantes;
    - Como integrar todas essas soluções de forma automatizada?
    - Uma solução para cada tecnologia é inviável!
    - Mudar de provedor de serviços pede nova implementação
  - 2 Crescente demanda por manutenção (evolutiva, adaptativa, corretiva, ...)
    - Novos requisitos  $\Rightarrow$  mudanças constantes;
    - Novas tecnologias  $\Rightarrow$  reimplementação;
    - Problemas como  $\Rightarrow$  desempenho, composição de serviços;



# Evolução das Tecnologias

- RSH (Remote Shell): Executar comandos de shell em máquinas remotas
- RPC (Remote Procedure Call): Executar funções em máquinas remotas
- RMI (Remote Method Invocation): API Java para acesso remoto de métodos

# Evolução das Tecnologias

- COM+ / DCOM (Distributed COM)
- Padrão para distribuir componentes COM
- Padrão Microsoft
- CORBA (Common Object Request Broker Architecture)
- Baseada em ORBs
- Acesso transparente
- Padrão aberto de mercado (OMG)
- IIOP (Internet Inter-ORB Protocol)
- Permite ao CORBA passar por gateways e ter interoperabilidade entre fabricantes

# Evolução das Tecnologias

- Java RMI: Somente Java
- OMG Corba: Complexidade
- Microsoft DCOM: Uma única implementação e Padrão proprietário

# Evolução das Tecnologias

	<i>Web Services</i>	<i>CORBA</i>	<i>RMI</i>
<i>Comunicação</i>	<i>SOAP</i>	<i>GIOP</i>	<i>JRMP</i>
<i>IDL</i>	<i>WSDL</i>	<i>CORBA IDL</i>	<i>Java</i>
<i>Registro</i>	<i>UDDI</i>	<i>Serviço de nomes</i>	<i>RMI Registry</i>

# SOA – Arquitetura Orientada a Serviços

- *Service-Oriented Architecture*

“SOA é um estilo arquitetural para construção de aplicações de software que utilizam serviços disponíveis em uma rede como a *Web*” [Endrei et al., 2004].

- Mas o que é um serviço?

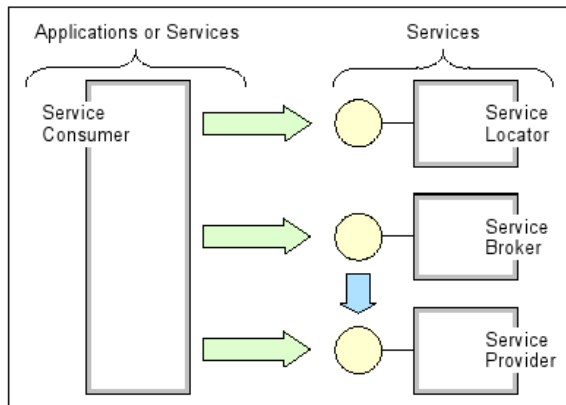
“É a implementação de uma funcionalidade qualquer, bem definida, que pode ser utilizada por clientes em diferentes aplicações e processos de negócios” [Mahmoud, 2005].

# SOA – Visão Geral

- Qual é a novidade então?
- As SOAs são caracterizadas por serem:
  - Fracamente acopladas;
  - Transparentes em localização;
  - Independentes de protocolo.
- Qual o objetivo?
  - 1 Integração de sistemas heterogêneos (interoperabilidade);
  - 2 Facilitar manutenções.
- Clientes não tem conhecimento de detalhes técnicos da implementação dos serviços.
- Uma infra-estrutura subjacente encarrega-se de tratar esses detalhes.

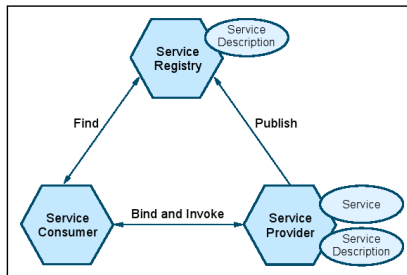
# SOA – Estrutura e Terminologia

- Estrutura e terminologia básica [Endrei et al., 2004]:



# SOA – Estrutura e Terminologia

- Papéis e colaborações [Endrei et al., 2004]:
  - Cliente do serviço: aplicação, componente ou outro serviço que requisita o serviço;
  - Provedor do serviço: entidade de rede endereçável que provê o serviço;
  - Registro de serviços: repositório que habilita a “descoberta” dos serviços.





# Web Services

## Web Service - WS

É um sistema de software projetado para suportar interoperabilidade entre máquinas sobre uma rede. Web Services caracterizam-se por ser uma tecnologia para computação distribuída baseada na Web.

# Web Services

- Tecnologia para comunicação entre aplicações via rede
- Independente de linguagem ou plataforma
  - Troca de mensagens XML
    - Descreve uma operação a ser executada ou dados a serem trocados

# Web Services – Vantagens [Endrei et al., 2004]

- Permite diferentes serviços distribuídos executarem em uma variedade de plataformas e arquiteturas
- Integração com sistemas existentes
- Integrar processos de negócios com clientes e fornecedores com um custo menor. Web services permitem compartilhamento de processos sem compartilhar tecnologias

# Web Services – Vantagens [Endrei et al., 2004]

- Oferecer serviços de negócio através da Web
- Liberdade de escolha de ferramentas, produtos e tecnologias
- Usam padrões e protocolos abertos
- Não fica preso a um middleware específico
  - Não importa se está utilizando Java, Microsoft ou CORBA
- Permite o reuso de serviços e componentes

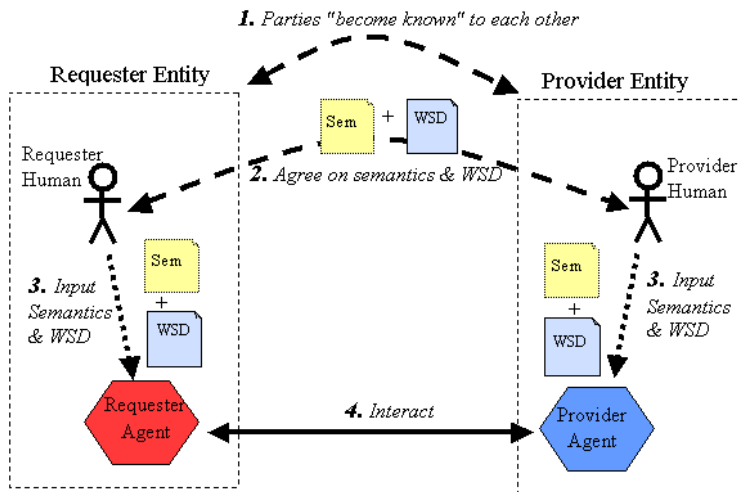
# Web Services – Vantagens [Endrei et al., 2004]

- Como oferecem essas vantagens ?
  - Utilização de XML
  - Utilização de protocolos padrões amplamente utilizados pela indústria
    - HTTP, HTTPS, SOAP
  - A descrição dos serviços é disponibilizada através de um arquivo WSDL (formato XML)

# Web Services – Desvantagens [Endrei et al., 2004]

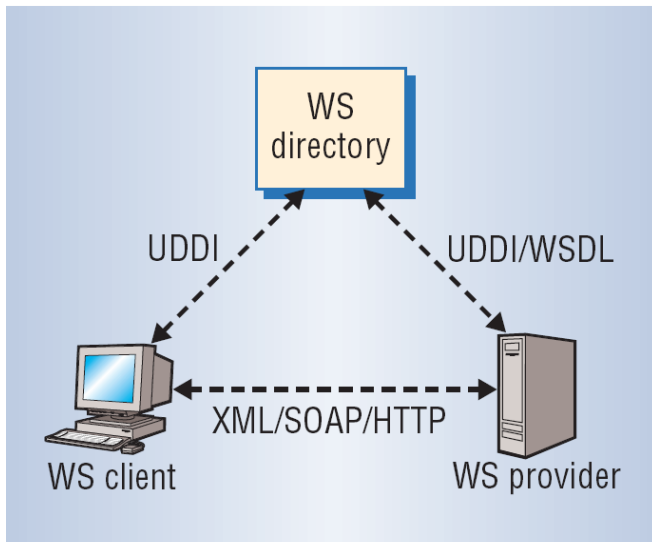
- Integrar Web services dinamicamente requer que o conteúdo do registro UDDI seja confiável. Somente redes privadas UDDI podem promover controle sobre seu conteúdo.
- Problemas de desempenho
  - Custo maior para processar documentos XML
- Padrões para integração de processos de negócio e gerenciamento de transações ainda estão em desenvolvimento

# Web Services – Funcionamento [W3C, 2004]



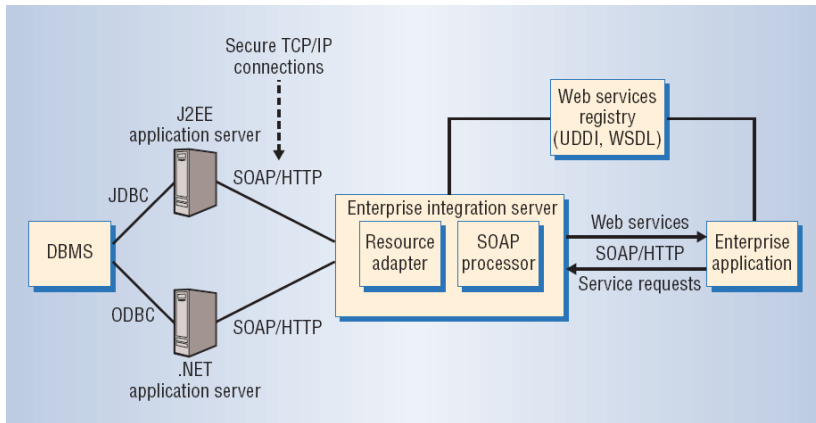
# Web Services – Arquitetura Básica

[Vaughan-Nichols, 2002]





# Web Services – Processamento [Vaughan-Nichols, 2002]



# Web Services – Tecnologias Envolvidas [Ort, 2005]

- XML;
- SOAP;
- WSDL;
- UDDI;

# XML - Linguagem de Marcação Extensível

- *eXtensible Markup Language*.
- Padrão para descrição de dados para serem trocados pela Web.
- Base para todos os demais.
  - Mensagens baseadas no protocolo SOAP;
  - Descrições de serviços escritas em WSDL;
  - Descrições de localização como entradas UDDI.

# SOAP - Protocolo Simples de Acesso a Objetos

- *Simple Object Access Protocol.*
- Formato de dados comum para troca de dados cliente–servidor.
- Utilizado para troca de informações em ambientes distribuídos.
- **Mensagem SOAP**, formada por:
  - Envelope;
  - Cabeçalho (*header*);
  - Corpo principal (*mandatory body*);

# SOAP - Exemplo

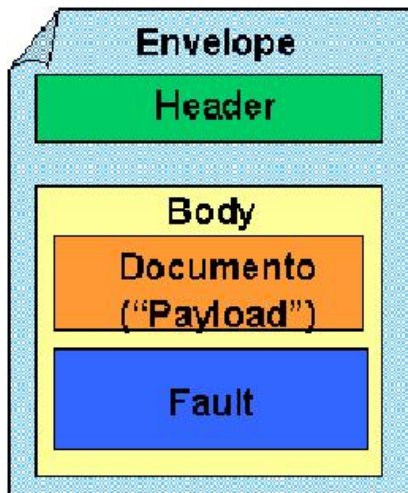
```
<SOAP-ENV: Envelope
  xmlns:SOAP-ENV=
    "http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle=
    "http://schemas.xmlsoap.org/soap/encoding/">

  <SOAP-ENV: Header >
    <t:Transaction
      xmlns:t="some-URI"
      SOAP-ENV:mustUnderstand="1">
    </t:Transaction>
  </SOAP-ENV:Header>

  <SOAP-ENV: Body >
    <m:GetBookPrice xmlns:m="some-URI">
      <title>My Life and Times</title>
      <author>Felix Harrison</author>
    </m: GetBookPrice>
  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

# Web Services – Funcionamento [W3C, 2004]



# WSDL - Linguagem de Descrição de Serviços Web

[Wolter, 2001]

- *Web Service Description Language.*
- Linguagem para composição de documentos que descrevem a interface de um WS.
- É um documento XML que descreve um conjunto de mensagens SOAP e como essas mensagens são trocadas.
- Especifica o que uma mensagem de requisição deve conter como será a mensagem de resposta.
- Define a localização do WS e qual protocolo de comunicação é utilizado para conversar como serviço.

# WSDL - Exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="BookSearch"
  targetNamespace="http://myexample.com/booksearch.wsdl"
  xmlns:...
  <types>
    ...
  </types>
  < message name="BookSearchInput">
    <part name="body" element="xsd:BookRequest"/>
  </ message>
  < message name="BookSearchOutput">
    <part name="body" type="xsd:BookResponse"/>
  </ message>
  <portType name="BooksPortType">
    < operation name="getBooks">
      <input message="tns:BookRequest"/>
      <output message="tns:BookResponse"/>
    </ operation>
  </portType>
  <binding name="BooksSearchBinding" type="tns:BooksPortType">
    ...
  </binding>
  <service name="BookSearchService">
    <port name="BooksSearchPort" binding="tns:BooksSearchBinding">
      <soap:address location="http://example.com/booksearch"/>
    </port>
  </service>
</definitions>
```



# UDDI - Descrição, Descoberta e Integração Universal

- *Universal Description, Discovery and Integration.*
- Repositório/registro/diretório de WS.
- Constitui as “**páginas amarelas**” dos WS.
  - Localização de WS;
  - Obtenção de informações sobre os serviços;
  - Modo de utilizar o serviço (interface);
- Fornece todos os detalhes de como utilizar os serviços.

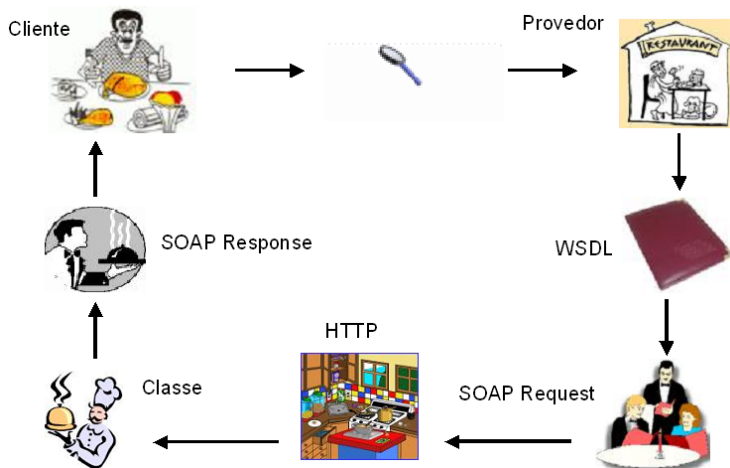
# UDDI - Exemplo de Entrada

```
<businessEntity businessKey="35AF7F00-1419-11D6-A0DC-000C0E00ACDD"
  authorizedName="0100002CAL"
  operator="www-3.ibm.com/services/uddi">
  <name>BooksToGo</name>
  <description xml:lang="en">
    The source for all professional books
  </description>
  <contacts>
    <contact>
      <personName>Benjamin Boss</personName>
      <phone>(877)1111111</phone>
    </contact>
  </contacts>
  ...
```

- Exemplo de mensagem SOAP para busca de serviços:

```
<find_business generic="2.0" xmlns=um:uddi-org:api-v2">
  <name>Books%</name>
</find_business>
```

# Web Services – Abstração



# Composição de Web Services

## Por que composição de Web Services?

A arquitetura de Web Services é satisfatória para interações cliente servidor, mas se for necessário combinar aspectos de vários serviços para gerar um novo serviço complexo, o ideal seria a utilização de composição de serviços para definir a execução dessas interações.

# Composição de Web Services

- Principais Motivações:
  - Possibilidade de criação de serviços mais elaborados a partir de serviços já existentes
    - Difícil localização de serviços candidatos
  - Possibilidade de integração de negócios

# Exemplo1

Usuário necessita de uma tradução  
Português - Russo! O que fazer?

Português  
<-> Inglês

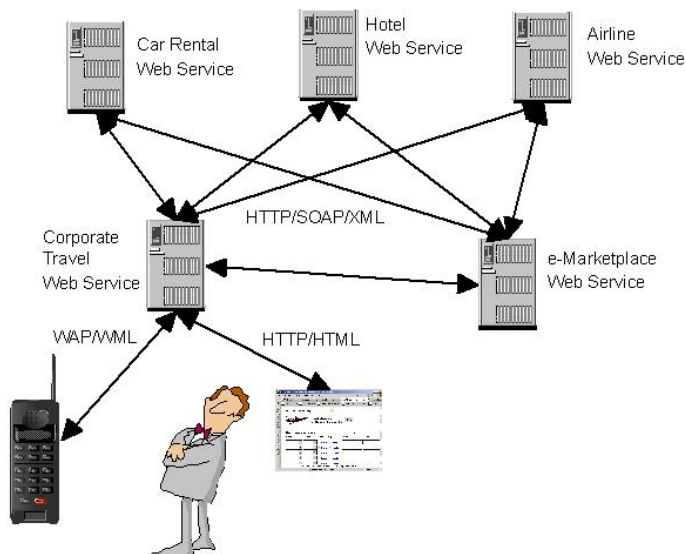
Inglês <->  
Alemão

Alemão <->  
Russo

# Exemplo1



# Exemplo2





# Modelos de Composição

## Orquestração

A orquestração descreve o fluxo de execução coordenado por um engine (mecanismo) centralizado ou distribuído para fazer a integração dos serviços. A principal linguagem utilizada para se fazer orquestração é WS-BPEL.

# Modelos de Composição

## Coreografia

A coreografia descreve a colaboração que cada participante deve ter, definindo, de um ponto de vista global, seu comportamento através de troca de informações para que ambos entrem em consenso sobre as regras de integração entre as entidades. Para descrever essas interações, de modo que não houvesse ambiguidade na documentação e responsabilidade dos participantes surgiu a WS-CDL (Web Services Choreography Description Language).

# Tipos de Composição

- Composição de Web Services Manual;
- Composição de Web Services Automática;
- Sub-tipos de Composição;
  - Estática - associada a serviços concretos(implementação do serviço);
  - Dinâmica - associada a serviços abstratos(descrição lógica do serviço);

# Internet

## O Fundamento Principal

- Acesso globalizado
- Versatilidade
- Permite acesso às informações

# Internet

## O Fundamento Principal

“Eu diria que o computador do futuro é a Internet. Hoje, se você tem um problema no computador, está tudo perdido, é terrível. Mas, com a computação nas nuvens, não importa se você usa o celular, o computador ou qualquer outro aparelho, tudo estará guardado na Internet”. Eric Schmidt - Presidente do Google

# O que é Computação em Nuvens?

Refere-se à possibilidade de utilizarmos computadores que podem se conectar à Web e utilizar; todas as ferramentas on-line. Assim, o computador seria simplesmente uma plataforma de acesso às aplicações, que estariam em uma grande nuvem – a Internet.

# Alguns Serviços na Nuvem



**FREE ONLINE STORAGE**

UPLOAD, ACCESS, SHARE

YOUR MUSIC, VIDEO, PHOTOS

# Modelos de Serviços Baseados em Cloud

- Storage-as-a-service
- Backup-as-a-service
- Database-as-a-service
- Information-as-a-service
- Process-as-a-service
- Application-as-a-service
- Platform-as-a-service
- Infrastructure-as-a-service



# Modelos mais Utilizados no Mercado Mundial

SaaS	Software-as-a-Service	Google Apps, Microsoft "Software+Services"
PaaS	Platform-as-a-Service	IBM IT Factory, Google AppEngine, Force.com
IaaS	Infrastructure-as-a-Service	Amazon EC2, IBM Blue Cloud, Sun Grid

# Software-as-a-Services (SaaS)

- O software é oferecido como serviço. O usuário não precisa adquirir licenças de uso para instalar ou comprar computadores ou servidores para executá-lo.
- Neste caso, pode ser pago um valor periódico - como se fosse uma assinatura - pelos recursos utilizados em função do tempo, por exemplo.

# Aplicações em Software as a Service - SaaS





### Google Apps Professional

US\$ 50/conta de usuário/ano  
avaliação gratuita por 30 dias\*

### Content, Community, Collaboration



# Aplicações em Software as a Service - SaaS

Microsoft Office Live Workspace Beta

Brasil (alterar) Ajuda Comentários

Você já tem uma conta do Microsoft Office Live? [Entrar](#)

**Acesso de qualquer lugar**

**Armazenamento e compartilhamento online**

**Funciona com programas que você conhece**

**Compartilhamento e armazenamento online gratuito de documentos**

**É fácil com o Microsoft Office Live Workspace**

- ✓ Armazene até 5 GB online
- ✓ Funciona com o Microsoft Office Word, Excel e PowerPoint
- ✓ Visualize, edite, compartilhe documentos protegidos por senha

[COMECE AGORA](#)

[Perguntas Frequentes](#) | [Saber mais](#)

# Infrastructure-as-a-Services (IaaS)

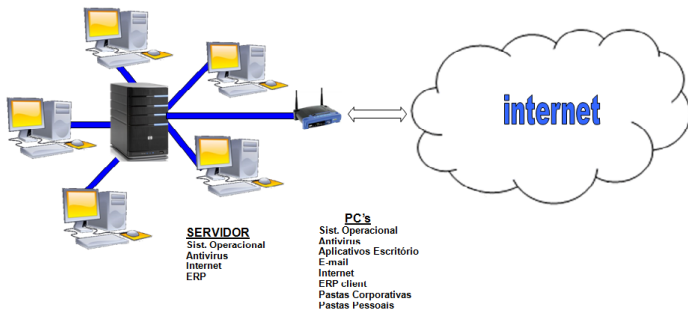
- Um ambiente completo e gerenciado. O IaaS oferece:



# Empresas que oferecem Infrastructure-as-a-Services (IaaS)



# A Infraestrutura Hoje?

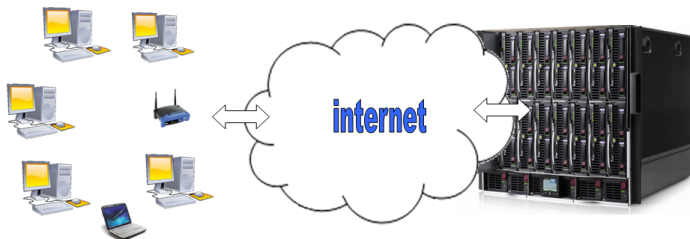




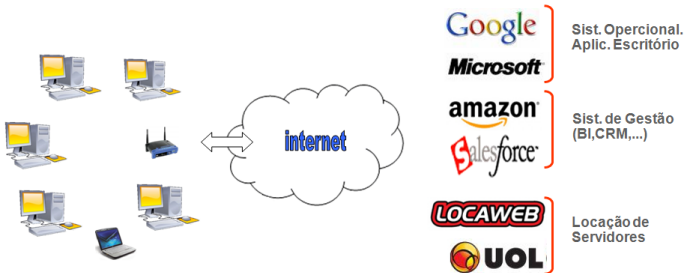
# A Infraestrutura Hoje?



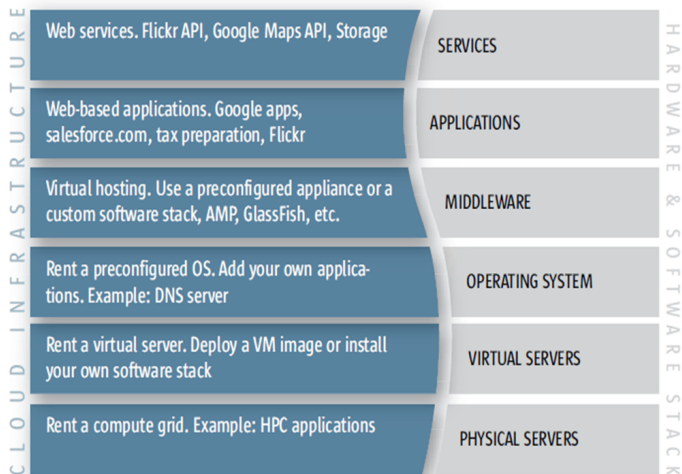
# A Infraestrutura Amanhã?



# A Infraestrutura Amanhã?



# Infraestrutura de Cloud



Fonte: Introduction to Cloud Computing (Sun Microsystems, 2009)

# Categorias de Serviços

- Nuvem pública: Disponível a todos na Internet;
- Nuvem privada: Ambiente proprietário que fornece os serviços apenas um número limitado de usuários;
- Nuvem híbrida: Fornece serviços executados em uma infra-estrutura de nuvem pública, mas limita o acesso a esta, com uma rede privada virtual (VPN).

# Vantagens

- Possibilidade de utilização de softwares sem que estes estejam instalados no computador;
- Na maioria das vezes o usuário não precisa se preocupar com o sistema operacional e hardware que está usando em seu computador pessoal, podendo acessar seus dados na “nuvem computacional” independentemente disso;
- O trabalho corporativo e o compartilhamento de arquivos se tornam mais fáceis, uma vez que todas as informações se encontram no mesmo “lugar”.

# Vantagens

- Melhor controle de gastos ao usar aplicativos, pois a maioria dos sistemas de computação em nuvem, quando não gratuitas, são pagas somente pelo tempo de utilização dos recursos;
- Redução considerável do preço de computadores devido ao fato de que um computador não necessita de muitos recursos para acessar a Internet;

# Reflexões

- Arquitetura em nuvem é muito mais que apenas um conjunto (embora massivo) de servidores interligados;
- Requer uma infra-estrutura de gerenciamento desse grande fluxo de dados que, incluindo funções para provisionamento e compartilhamento de recursos computacionais, equilíbrio dinâmico do workload e monitoração do desempenho;
- Idéia de que “tudo é de todos e ninguém é de ninguém” nem sempre é verdade.



# Reflexões

- A Segurança é crítica, considerando que dados ficam online o tempo todo.

# Conclusão

- SOA e Web Services são temas atuais
  - Grandes empresas tem investido em arquiteturas orientadas a serviços (IBM, HP)
  - Os padrões para web services estão em constante evolução
  - É preciso formar profissionais para lidar com SOA.
- O Conceito de Computação em Nuvens não é atual
  - Redução de custos
  - Recursos para executar aplicaçõesm estão em um servidor ou conjunto de servidores - Dados podem ser acessados de qualquer lugar do mundo
  - A Internet é a sustentação da Computação em Nuvens



### Introduction to web services and the wsdk v5.1.

Online.

disponível em [www.ibm.com/developerWorks](http://www.ibm.com/developerWorks) - último acesso em 30/04/2008.



### Web services and corba.

Online.

disponível em [www.xs4all.nl/~irmen/comp/CORBA\\_vs\\_SOAP.html](http://www.xs4all.nl/~irmen/comp/CORBA_vs_SOAP.html) - último acesso em 30/04/2008.



### Web services and soap.

Online.

disponível em <http://www.w3c.org/2002/ws> - último acesso em 30/04/2008.



### Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2003).

Web services: Concepts, architectures, and applications.

*V. Springer Verlag.*



### Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., and Newling, T. (2004).

*Patterns: Service-Oriented Architecture and Web Services.*

IBM Redbooks Series. IBM Corporation.

disponível em <http://www.redbooks.ibm.com/abstracts/sg246303.html> - último acesso em 30/04/2008.



Mahmoud, Q. H. (2005).

Service-oriented architecture (SOA) and web services: The road to enterprise application integration (EAI).

*Sun Technical Articles.*

disponível em <http://java.sun.com/developer/technicalArticles/WebServices/soa/index.html> - último acesso em 30/04/2008.



Ort, E. (2005).

Service-oriented architecture and web services: Concepts, technologies, and tools.

*Sun Technical Articles.*

disponível em

<http://java.sun.com/developer/technicalArticles/WebServices/soa2/> - último acesso em 30/04/2008.



Vaughan-Nichols, S. J. (2002).

Web services: Beyond the hype.

*IEEE Computer*, 35(2):18–21.



W3C (2004).

Web services architecture.

Online.

disponível em <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/> - último acesso em 30/04/2008.



Wolter, R. (2001).

Xml web services basics.

*Microsoft MSDN Library.*

disponível em <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/webservbasics.asp> - último acesso em 30/04/2008.