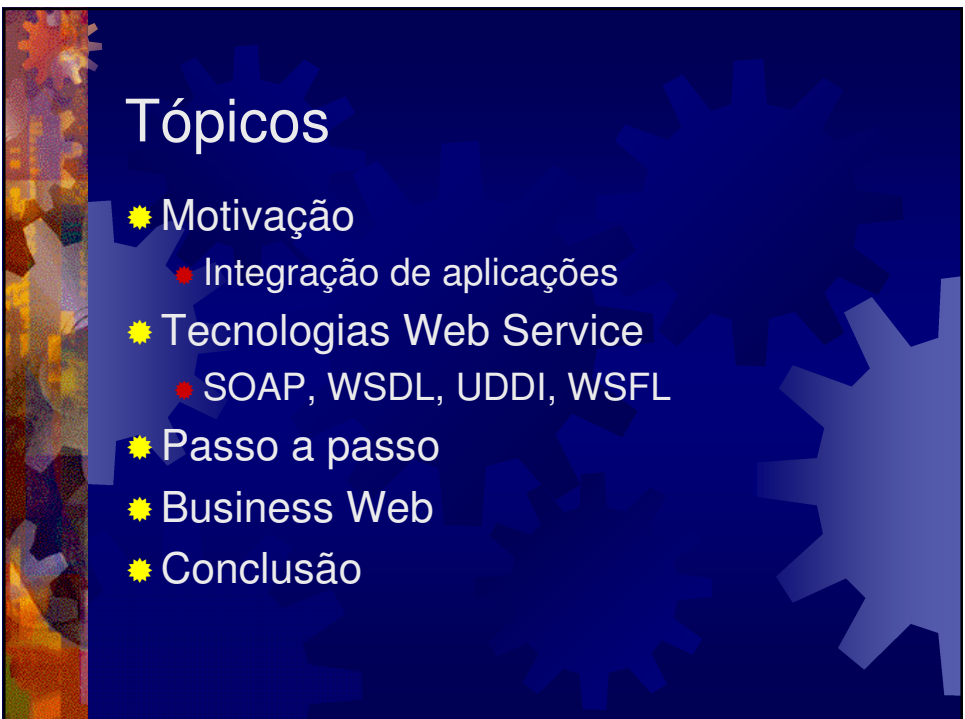




Web Services

Antonio Dirceu – adrvf@cin.ufpe.br



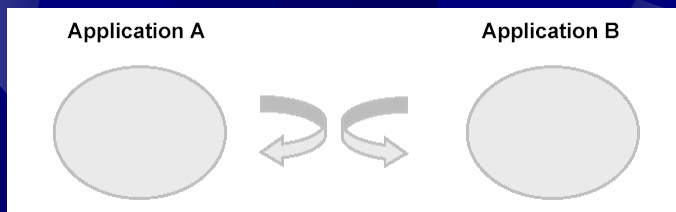
Tópicos

- Motivação
 - Integração de aplicações
- Tecnologias Web Service
 - SOAP, WSDL, UDDI, WSFL
- Passo a passo
- Business Web
- Conclusão

Motivação

Integração de Aplicações

- ☀ O desafio da interoperação entre ambientes heterogêneos

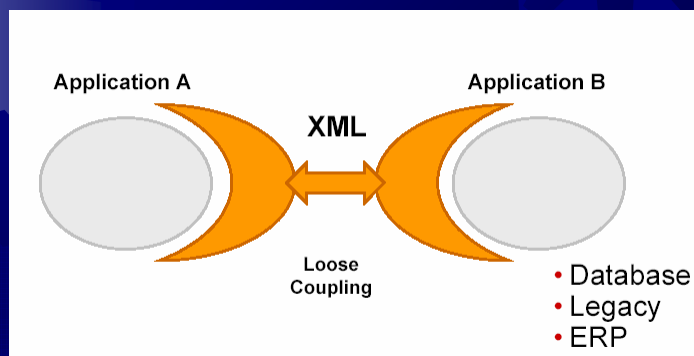


Integração de Aplicações

- Diversidade de componentes
 - EJB, CORBA, DCOM ...
- Diversidade de linguagens
 - Java, C/C++, C# ...
- Firewalls
- Falta de padrões para interoperação

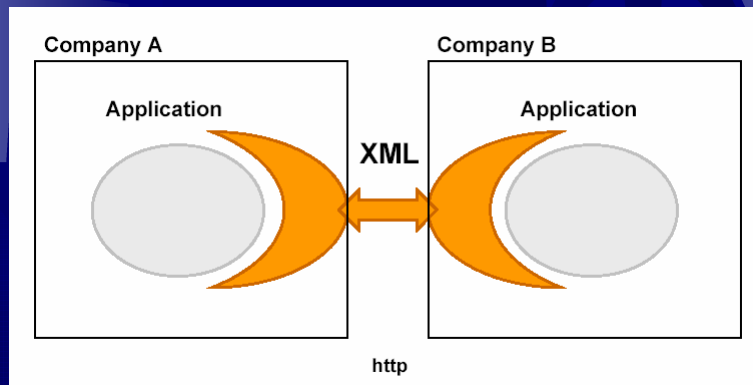
Integração de Aplicações

- Definindo o formato para troca de dados



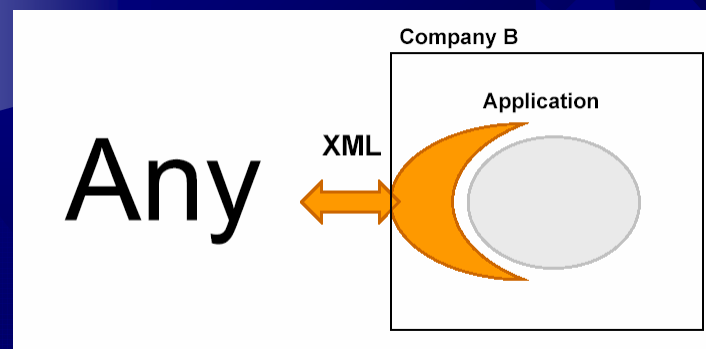
Business-to-Business

- Aplicações se conhecem e conversam entre si



Web Services

- Aplicações oferecem serviços que podem ser acessados dinamicamente



Web Services

- Geração de serviços baseados na internet que utiliza padrões da indústria, como XML, SOAP e UDDI para conectar aplicações e prover novos serviços via Internet
 - serviços mais robustos e integrados
- Nova “onda” de componentes
 - serviços como componentes reutilizáveis
 - “blocos” que integrados produzirão novos serviços

Web Services

- Definição
 - Modelo computacional distribuído
 - fracamente acoplado
 - utiliza mecanismos de transporte padrão (HTTP e HTTPS)
 - modelo de programação síncrona e assíncrona
 - utiliza XML para o transporte de dados
 - Descrito através de metadados (XML)
 - Localizável através de pesquisas em diretórios de serviços

Web Services

- ☀ *Service-oriented*
 - paradigma de orientação a serviços
- ☀ Tecnologia
 - basicamente é XML sobre HTTP
 - XML: porque é texto simples e independente de plataforma
 - HTTP: livre acesso através de firewalls

Web Services

- ☀ A web “costumava” ter foco em pessoas
- ☀ A web está se tornando um plataforma A2A (application-to-application)
 - B2B é um caso especial
- ☀ Web Services é a plataforma computacional distribuída sobre a qual aplicações A2A serão construídas

Tecnologias Web Services

- ✴ *Any technology-service paradigm*
- ✴ Tipicamente
 - SOAP: transporte de dados (XML/HTTP)
 - WSDL: descrição dos serviços
 - UDDI: registro e busca de serviços
 - WSFL: composição de Web Services

SOAP

- ✴ Simple Object Access Protocol
- ✴ Modelo de mensagens independente do protocolo de transporte
 - suporte para HTTP
- ✴ Modelo de codificação para tipos do sistema
 - exemplo: XML para objetos Java, e vice-versa
- ✴ RPC sobre HTTP

WSDL

- Web Services Definition Language
- Provê descrição funcional de serviços
 - *IDL description*
 - *Protocol and deployment details*
 - idealmente deveria prover todas as informações necessárias para acessar o serviço (programaticamente)
- *Machine-readable description*

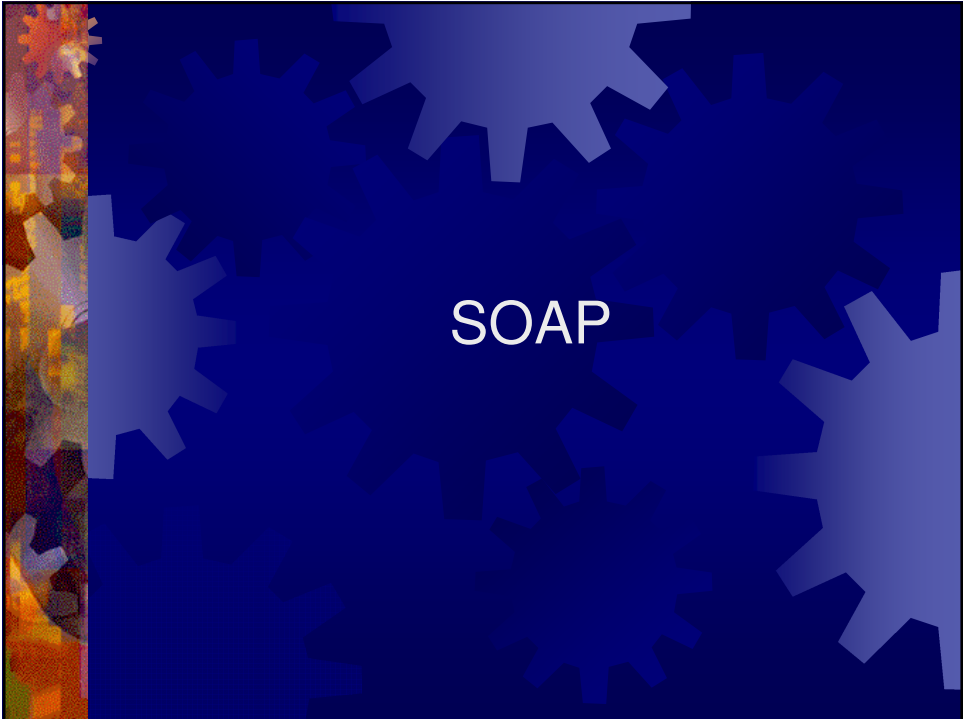
UDDI

- Universal Description, Discovery and Integration
- Coleção de diretórios (*peers*) que contém informações sobre negócios e serviços
- Conjunto de especificações baseadas em padrões para a descrição e pesquisa de serviços
- UBR Node – UDDI Business Registry Node
 - Ex.: <http://uddi.microsoft.com>



WSFL

- Web Services Flow Language
- Composição de Web Services
 - controle do fluxo de mensagens
- Construída sobre WSDL
- Modelos de utilização
 - Flow Model
 - compõe serviços existentes em novos serviços, definindo o *workflow* entre os serviços compostos
 - Global Model
 - descreve a interação entre serviços sem definir a função composta (serviço final)



SOAP

SOAP – Simple Object Access Protocol

☀ Definição:

- É um protocolo que descreve mensagens trocadas entre processos, chamadas remotas a métodos e padrões para transporte via HTTP.
- Baseado em XML
- Se propõe a desempenhar o mesmo papel do IIOP (padrão CORBA), ORPC (padrão DCOM) e JRMP (padrão Java RMI) para Sistemas Distribuídos na Internet.

SOAP – Aspectos Gerais

- SOAP é baseado em texto (XML) enquanto os demais protocolos (IIOP, JRMP, ORPC) são binários
 - Vantagens: facilidade de depuração e maior adaptabilidade aos firewalls
 - Desvantagem: tamanho da mensagem SOAP é bem maior que os protocolos binários podendo gerar perda de performance
- Padrão aberto e independente de plataforma

Arquitetura do SOAP

- Um **envelope** que descreve o conteúdo da mensagem e como processá-la
- Um conjunto de **regras de codificação** que descrevem como os tipos de dados definidos na aplicação são serializados
- Uma convenção para representar **chamadas remotas a métodos** (RPC) e suas respostas

Exemplo de Mensagem SOAP

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Exemplo: SOAP Body

```
<SOAP-ENV:Body>
  <ns1:sayHelloTo
    xmlns:ns1="Hello"
    SOAP-
    ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <name xsi:type="xsd:string">Maria</name>
  </ns1:sayHelloTo>
</SOAP-ENV:Body>
```

WSDL

WSDL – Web Services Description Language

Definição:

- É a especificação publicada para um diretório UDDI(Universal Description Discovery).
- Tem uma interface e implementação de detalhes específicos dos serviços da Web disponíveis e dos respectivos proprietários
- Baseada em XML, descreve a interface, protocolos, ligações, detalhes de visualização, tipos de dados e localização

A Dupla SOAP e WSDL

- O SOAP foi criado pela Microsoft/IBM
- O WSDL foi criado para descrever os contratos(interfaces) dos serviços WEB invocados através do SOAP
- Estes contratos de serviços indicam à aplicação que pretende chamar os serviços, quais são as suas interfaces, incluindo parâmetros de chamada e retorno

Estrutura do WSDL

```
<?xml version="1.0">
  <definitions name="HelloWorldService" ... Name Space Information ...>
    <types>
      <schema>
        definition of parameter data types in XML Schema (optional)
      </schema>
    </types>
    <message name="HelloIF_sayHello">
      definition of a message (request, reply)
    </message>
    <portType name="HelloIF">
      <operation name="sayHello">
        definition of an operation (request - reply pair)
      </operation>
    </portType>
    <binding name="HelloIfBinding" type="HelloIF">
      definition of a protocol binding (typically SOAP)
    </binding>
    <service name="HelloWorldService">
      <port name="StockQuotePort">
        definition of a port (an Internet address)
      </port>
    </service>
  </definitions>
```

Messages

```
<message name="HelloIF_sayHello">
  <part name="String_1" type="xsd:string"/>
</message>
<message name="HelloIF_sayHelloResponse">
  <part name="result" type="xsd:string"/>
</message>
```

Provê os nomes das mensagens e a passagem de parâmetros

Ports

```
<portType name="HelloIF">
  <operation name="sayHello" parameterOrder="String_1">
    <input message="tns:HelloIF_sayHello"/>
    <output message="tns:HelloIF_sayHelloResponse"/>
  </operation>
</portType>
```

Define a sequência de mensagens correspondentes a uma invocação de serviço

Protocol Binding

```
<binding name="HelloIFBinding" type="tns:HelloIF">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
    style="rpc"/>
  <operation name="sayHello">
    <soap:operation soapAction=""/>
    <input>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded"
        namespace="http://lsirwww.epfl.ch/"></input>
    <output>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="http://lsirwww.epfl.ch/">
    </output>
  </operation>
</binding>
```

Implementa mensagens abstratas de acordo com o protocolo SOAP

Exemplo de Mensagem Soap

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2002/12/soap-envelope">
  <env:Header>
    <m:HelloIF_sayHello xmlns:m="http://lsirwww.epfl.ch/"
      env:role="http://www.w3.org/2002/12/soap-envelope/role/next"
      env:mustUnderstand="true">
    </m:HelloIF_sayHello>
  </env:Header>
  <env:Body>
    <p:String_1 xmlns:p="http://lsirwww.epfl.ch/">
      Hello World!
    </p:String_1>
  </env:Body>
</env:Envelope>
```



Acesso ao serviço

```
<service name="HelloWorldService">
  <port name="HelloIFPort"
    binding="HelloIFBinding">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
```

O location não é conhecido antes do deploy

WSDL depois do Deploy

```
<service name="HelloWorldService">  
  <port name="HelloIFPort" binding="tns:HelloIFBinding">  
    <soap:address location=  
      "http://lsir-cis-pcx:8009/hello/helloService"/>  
  </port>  
</service>
```

Esta é a URL que poderá ser publicada no UDDI

UDDI



UDDI

(Universal Description, Discovery and Integration)

- Permite informações sobre negócio e serviços sejam eletronicamente publicadas e consultadas. "Register once, published everywhere"
- Define como os negócios interagem sobre a internet e compartilham informações em uma arquitetura de registro global.
- É um framework para integração de Web Services.



UDDI

(continuação)

- UDDI é uma especificação global. Toda empresa espalhada pelo mundo é capaz de se registrar em um *UDDI Business Registry*.
- Business Registry implementa a especificação UDDI.
- Nos *Registries* ocorre uma sincronização de seus conteúdos regularmente.
- Permissões de acesso – Somente o usuário com as credenciais que conferem com as credenciais usadas para criação dos registros podem alterá-los.

Exemplo UBR

The screenshot shows the IBM UDDI Business Test Registry interface. The main heading is "UDDI Business Test Registry" with the subtitle "Universal Description, Discovery, and Integration". The page is titled "Add a Business". A text block explains that the only required field is the business name, and users can press "Continue" to add more details or "Cancel" to stop. Below this is a form section titled "Enter the name of your business" with a "Language" dropdown set to "English" and a "Name" text input field. "Cancel" and "Continue" buttons are at the bottom of the form. The left sidebar contains links for "Publish", "Find", "Account", and "Related Links". The footer includes "About IBM", "Privacy", "Legal", and "Contact".

UDDI

- ☀ Informação fornecida a um registro UDDI consiste de três componentes:
 - White pages
Informações de contato da empresa.
 - Yellow pages
Categorização dos serviços pela taxonomia padrão.
 - Green pages
Informações técnicas sobre o serviço exposto (regras de negócio, descrição do serviço,...)



Benefícios imediatos de UDDI

- ✦ A capacidade de descobrir serviços e interagir com eles de forma rápida e dinâmica na internet.
- ✦ Agregar valor aos serviços já fornecidos aos seus clientes.
- ✦ Potencializar o crescimento do comércio B2B.



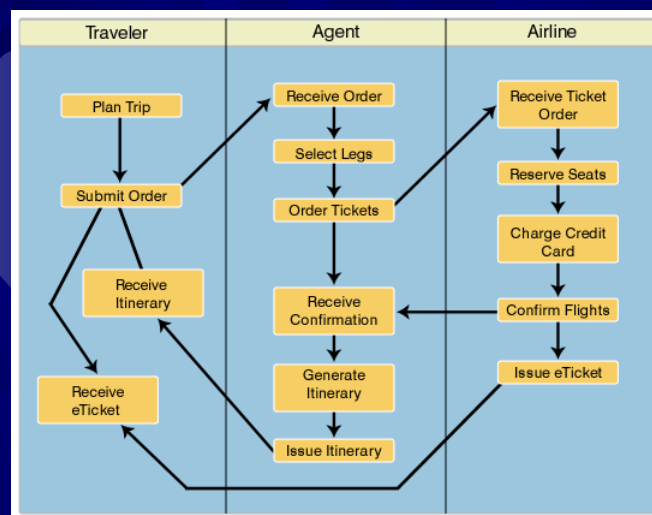
WSFL

WSFL

Web Service Flow Language

- Definição
 - Modelo de fluxo
 - Modelo global

Exemplo de Fluxo de Serviços



Modelo de Fluxo

```
<flowModel name="bookTickets" serviceProviderType="airlineFlow">

  <flowSource name="ticketFlowSource">
    <output name="processInstanceData"
      message="tio:receivedTicketOrder"/>
  </flowSource>

  <serviceProvider name="agent" type="agentType"/>
  <serviceProvider name="traveler" type="travelerType"/>

  <activity name="reserveSeats">
    <input name="reserveSeatsInput" message="tio:receivedTicketOrder"/>
    <output name="reserveSeatsOutput" message="tio:reservation"/>
    <performedBy serviceProvider="local"/>
    <implement>
      <internal>
        <!-- .. call to credit card service .. -->
      </internal>
    </implement>
  </activity>

</flowModel>
```

Modelo Global

```
<globalModel name="bookTripNTickets" serviceProviderType="agentType">
  <serviceProvider name="travelAgent"
    serviceProviderType="tio:agentType">
    <export>
      <source portType="tio:tripHandler" operation="sendItinerary"/>
      <target portType="tio:tripNTicketHandler"
        operation="sendItinerary"/>
    </export>
    <export>
      <source portType="tio:tripHandler" operation="receiveTripOrder"/>
      <target portType="tio:tripNTicketHandler"
        operation="receiveTripOrder"/>
    </export>
    <locator type="static" service="abc:agent"/>
    <locator type="uddi" bindTime="startup" selectionPolicy="first">
      <uddi-api:find_service businessKey="uuid_key"
        generic="1.0" xmlns:uddi-api="urn:uddi-org:api">
      </uddi-api:find_service>
    </locator>
  </serviceProvider>

</globalModel>
```

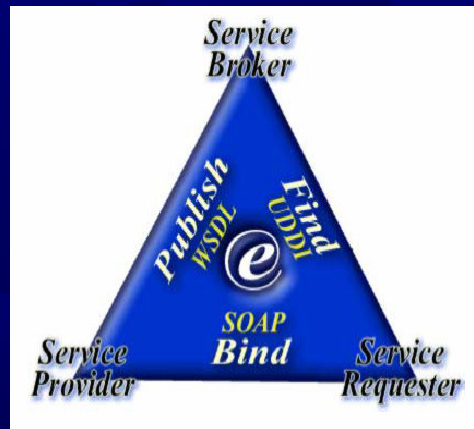
Web Services

● Papéis

- Service provider
- Service broker
- Service requester

● Operações

- Publish
- Find
- Bind



Web Services

● Passo 1: Implementar o serviço

- Utilizando qualquer linguagem que converse http

● Passo 2: Descrever o Web Service utilizando WSDL

- O WSDL pode ser gerado automaticamente

● Passo 3: Efetuar o Deploy

- É gerado o proxy do servidor. O *Service Location* é então incluído no WSDL

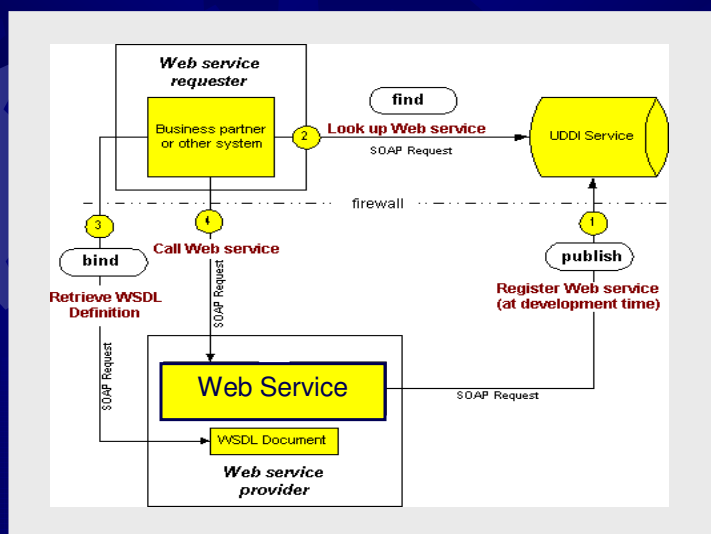
● Passo 4: Publicar o serviço (opcional)

- O cliente poderá localizar o serviço utilizando o UDDI

Web Services

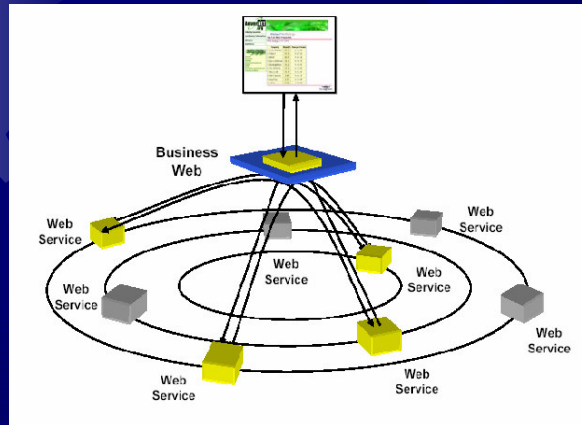
- Passo 5: Gerar o proxy do cliente
 - Também chamado de *Stub*. Pode ser gerado automaticamente através do WSDL
- Passo 6: Escrever a chamada do cliente ao método remoto
 - Através do *Stub*
- Passo 7: Executar o cliente
 - O cliente invoca o serviço (quase) como uma chamada a um método local

Web Services



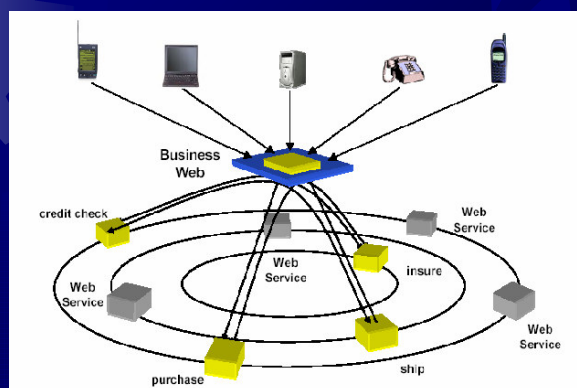
Business Web

- ☀ Novo tipo de aplicação

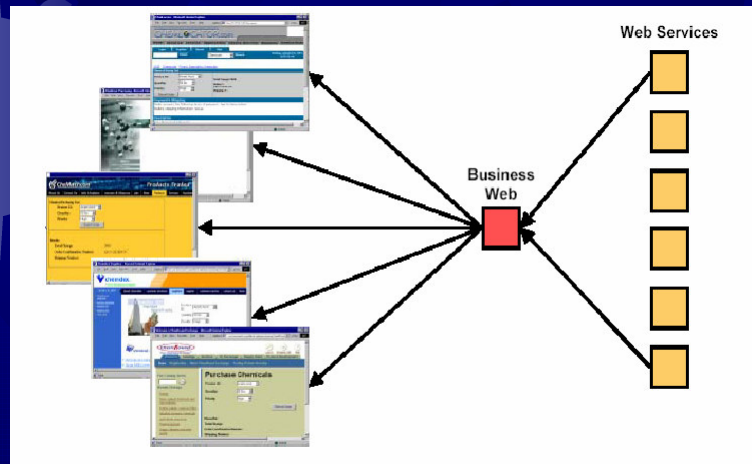


Business Web

- ☀ É um Web Services composto por outros Web Services



Business Web



Web Services

☀ Benefícios

- baixo acoplamento entre aplicações
- evolução independente de aplicações
- B2B a baixo custo (reuso)
- EAI (Enterprise Application Integration) não intrusiva
- diversidade de componentes não afeta interoperabilidade
- diversidade de linguagens não afeta interoperabilidade
- padronização (futura) dos mecanismos de interação

Web Services

★ Melhorias da versão 2.0

- WS-Addressing
- WS-Policy
- WS-Security
- WS-SecurityPolicy
- WS-SecureConversation
- WS-Trust
- ...



WEB SERVICES
INTEROPERABILITY
ORGANIZATION

www.ws-i.org

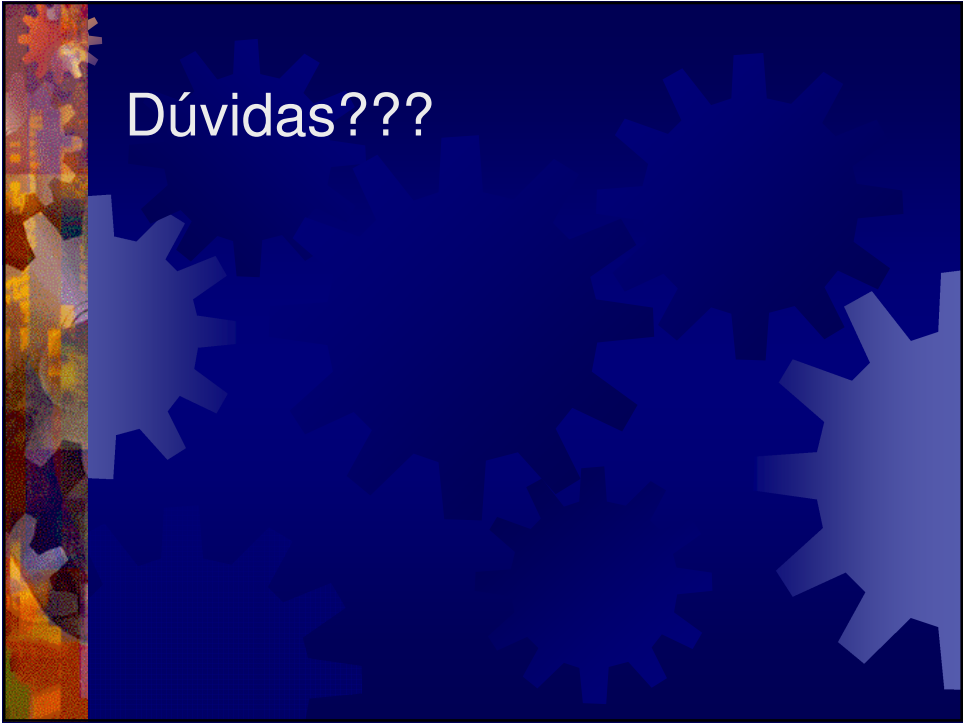
- ★ 170+ fabricantes software envolvidos
- ★ Interoperabilidade entre plataformas, aplicações e linguagens
- ★ Maior evidência de que a indústria está alinhada com os web services



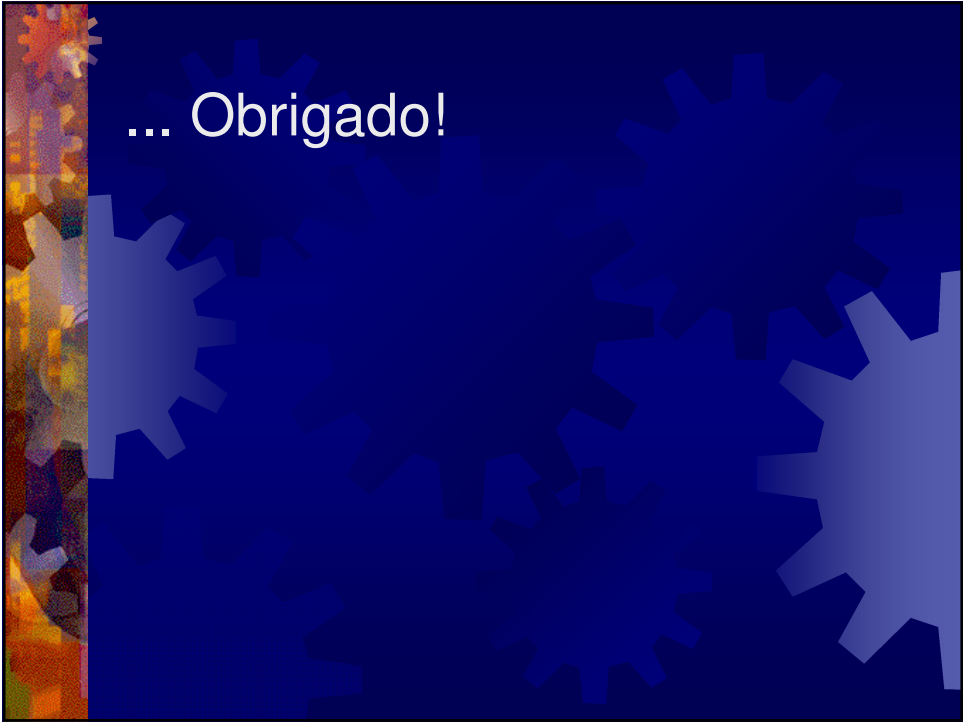
Web Services

● Conclusão

- WebService é uma forma nova de utilizar tecnologias e conceitos já existentes
 - viabilizada pelo contexto tecnológico e comercial atual
- Traz uma tecnologia real para distribuição de serviços na Internet
 - Utiliza o protocolo http
 - Traz interoperabilidade entre sistemas heterogêneos
- Características como segurança e controle de transações ainda estão sendo melhor especificadas estarão disponíveis em breve



Dúvidas???



... Obrigado!