DATASUL

NOVOS
CONCEITOS.
GRANDES
RESULTADOS.

Web Services e Java

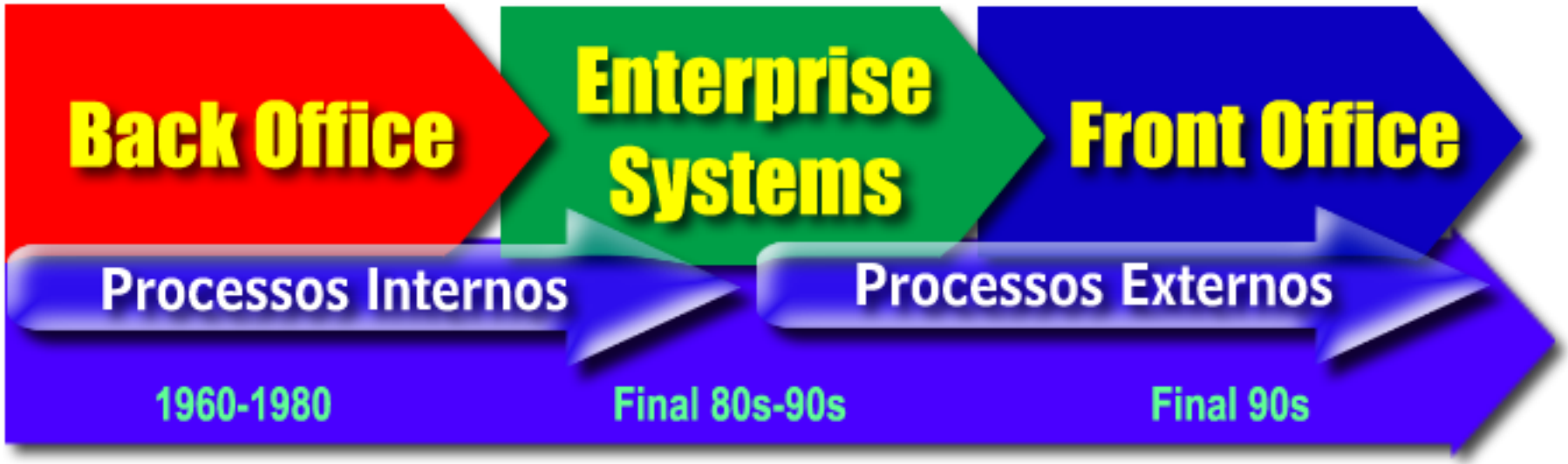
Roger Pedroso – Datasul Tecnologia

Nov/2003

Agenda

- Introdução
- Definição
- Padrões associados
- APIs e ferramentas Java

Evolução da Tecnologia da Informação



Mainframe

- Folha de Pagamento
- Manufatura (MRP)
- Finanças

Client/Server

- Manufatura (ERP)
- Recursos Humanos
- Distribuição
- Planejamento

Web Architecture

- Call Center Support
- Supply Chain
- Sales Force Automation

10-100s de Usuários
Foco Interno

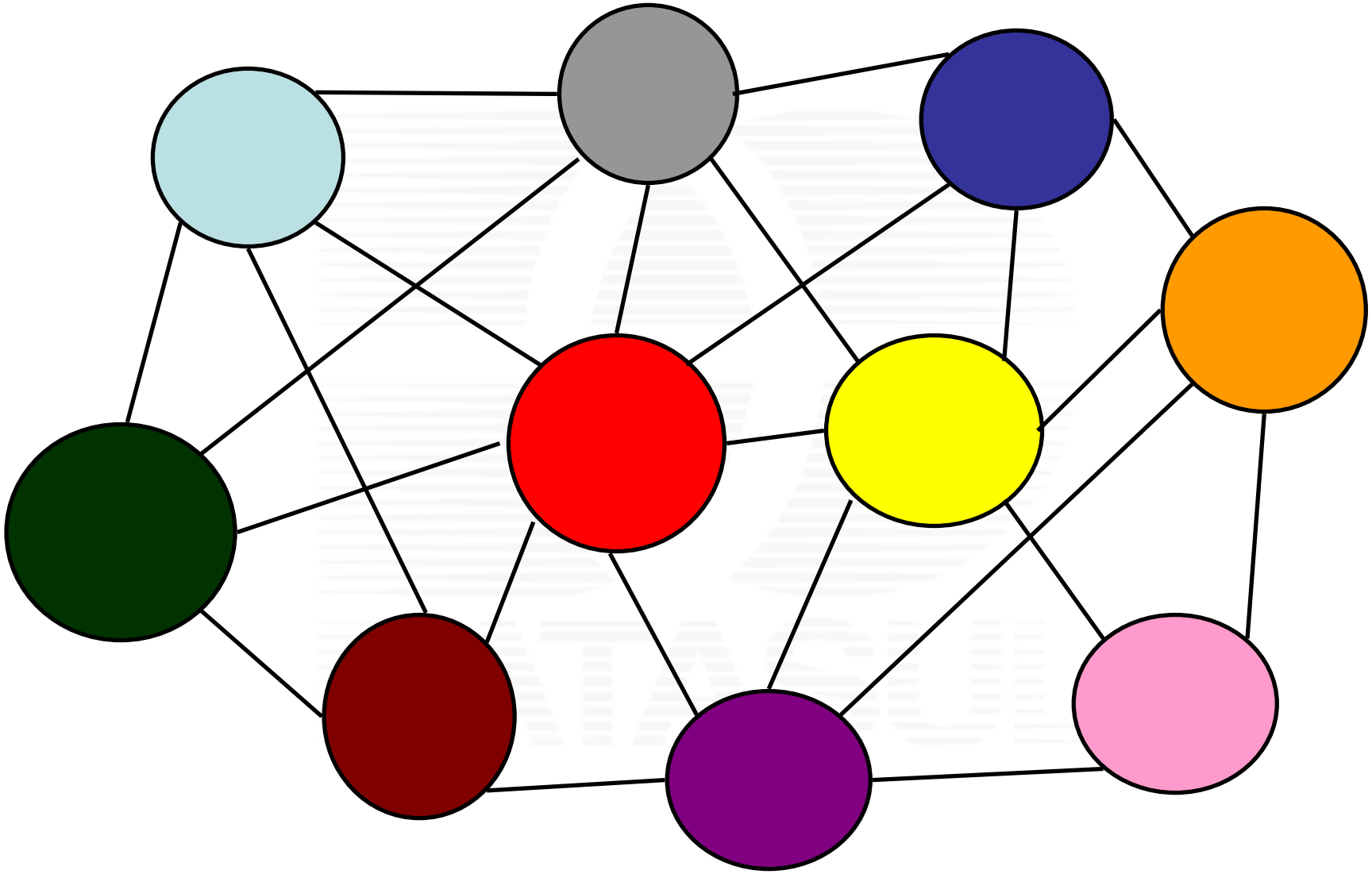
100s-1000s de Usuários
Maior número de usuários

1000s-10000s
Clientes, Fornecedores, Parceiros

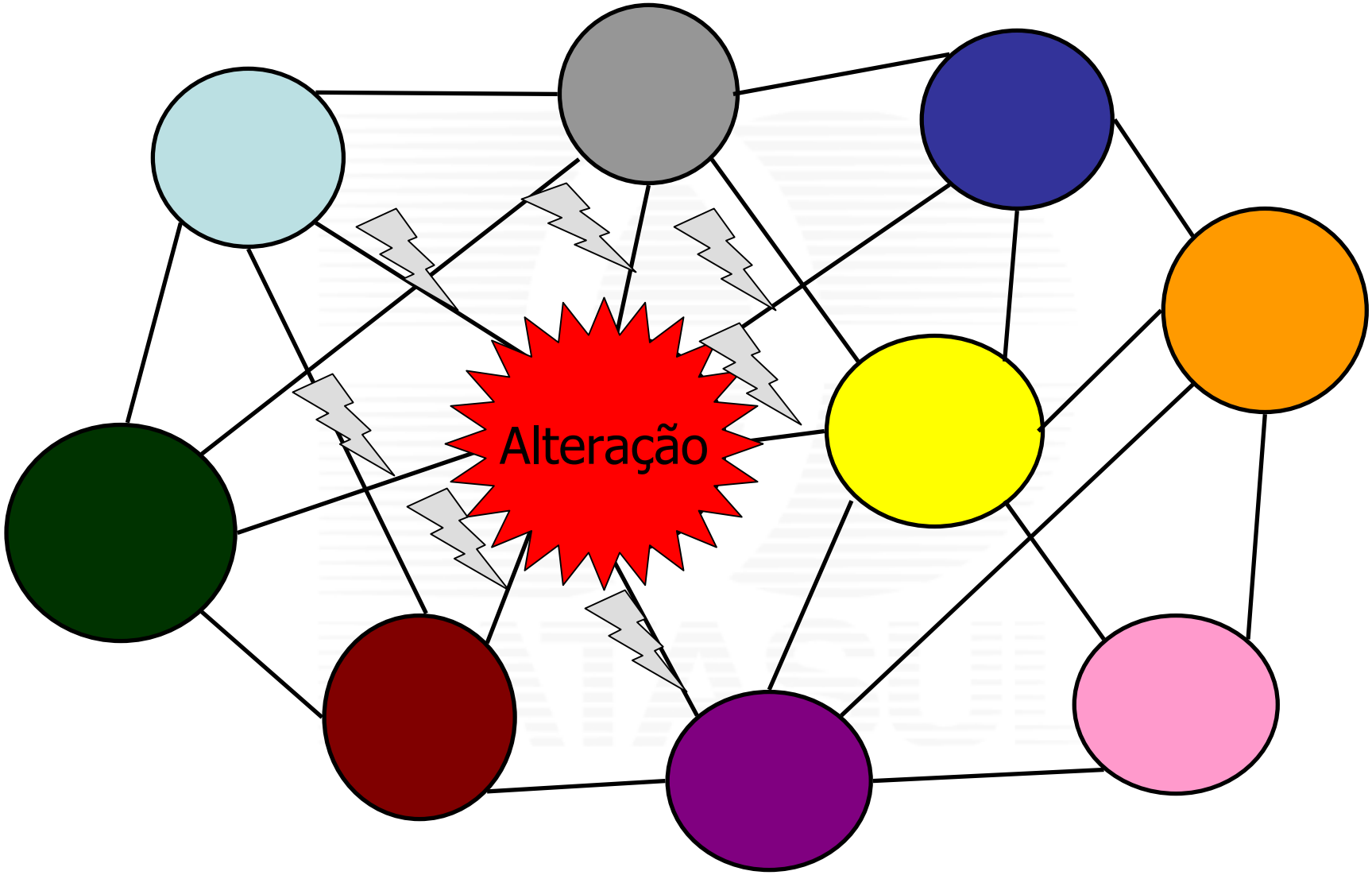
Desafio para Integração

- Diferentes Sistemas Operacionais
  - Unix, Windows, AS/400, etc.
- Diferentes DBMS
  - Oracle, Progress, MS-SQL, DB/2, etc.
- Diferentes Linguagens
  - Progress 4GL, Visual Basic, Java, etc..
- Diferentes Aplicações
  - ERP, CRM, SCM, Legado, MES, etc.

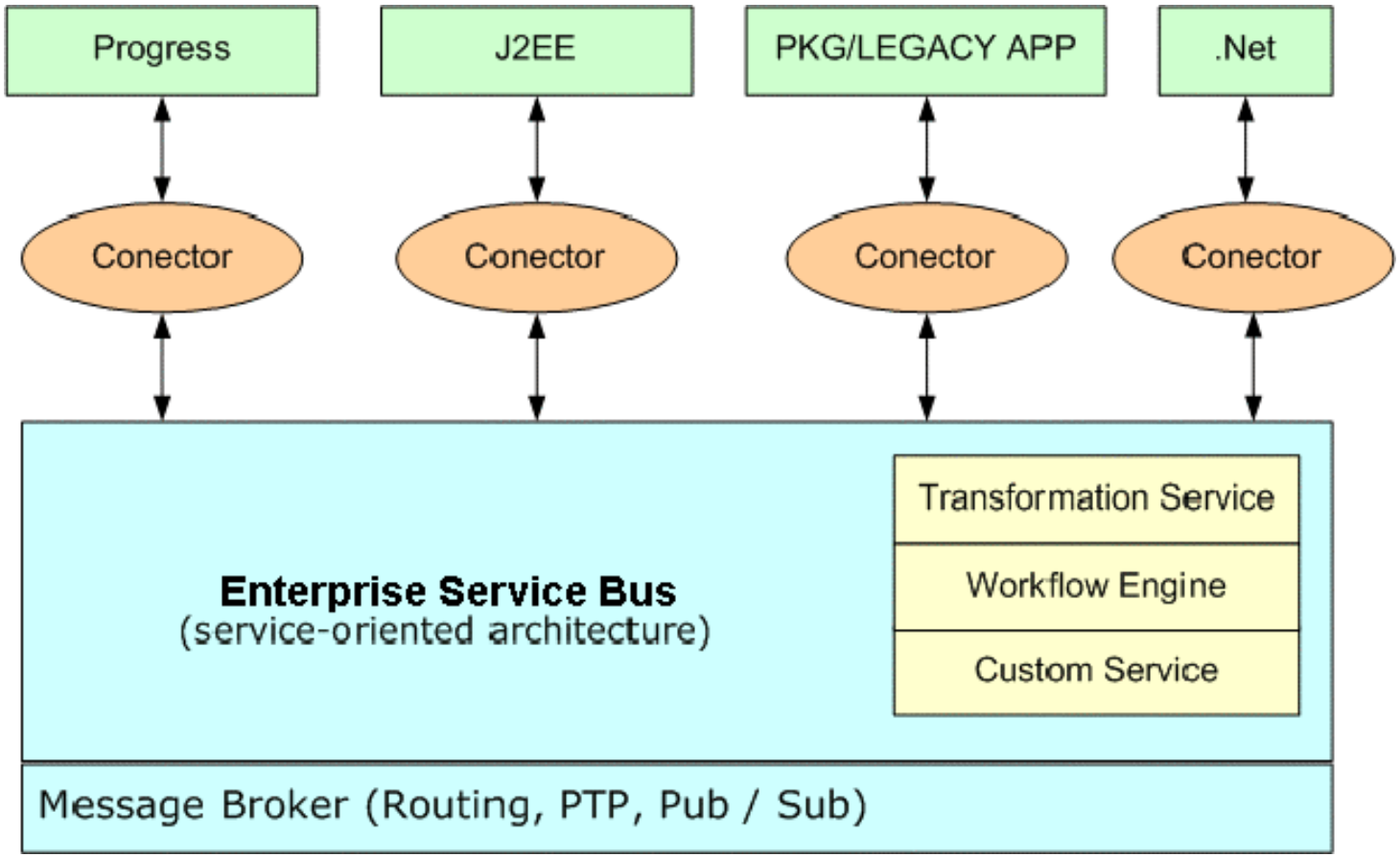
Integração de Aplicações



Integração de Aplicações



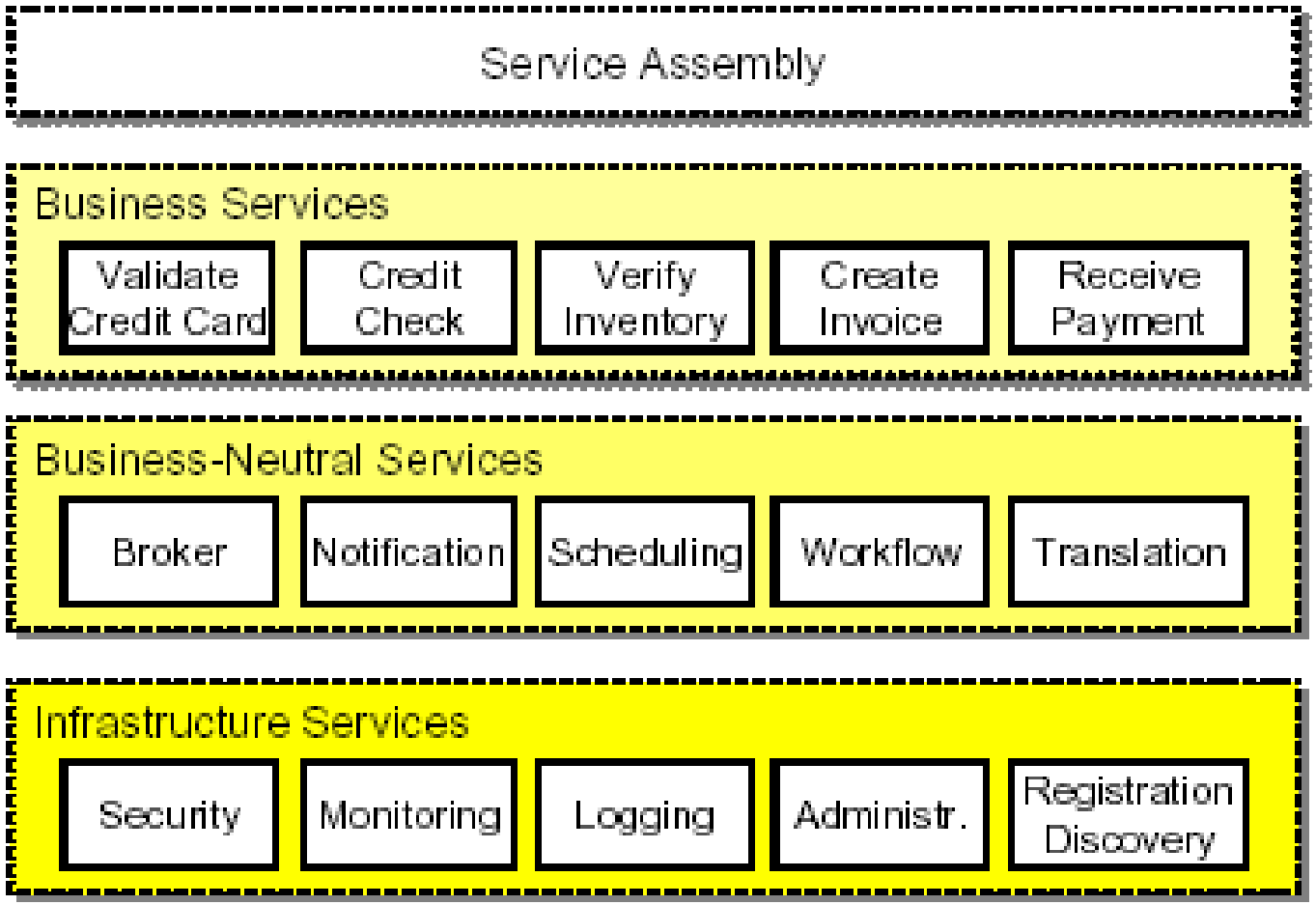
Mecanismos de EAI



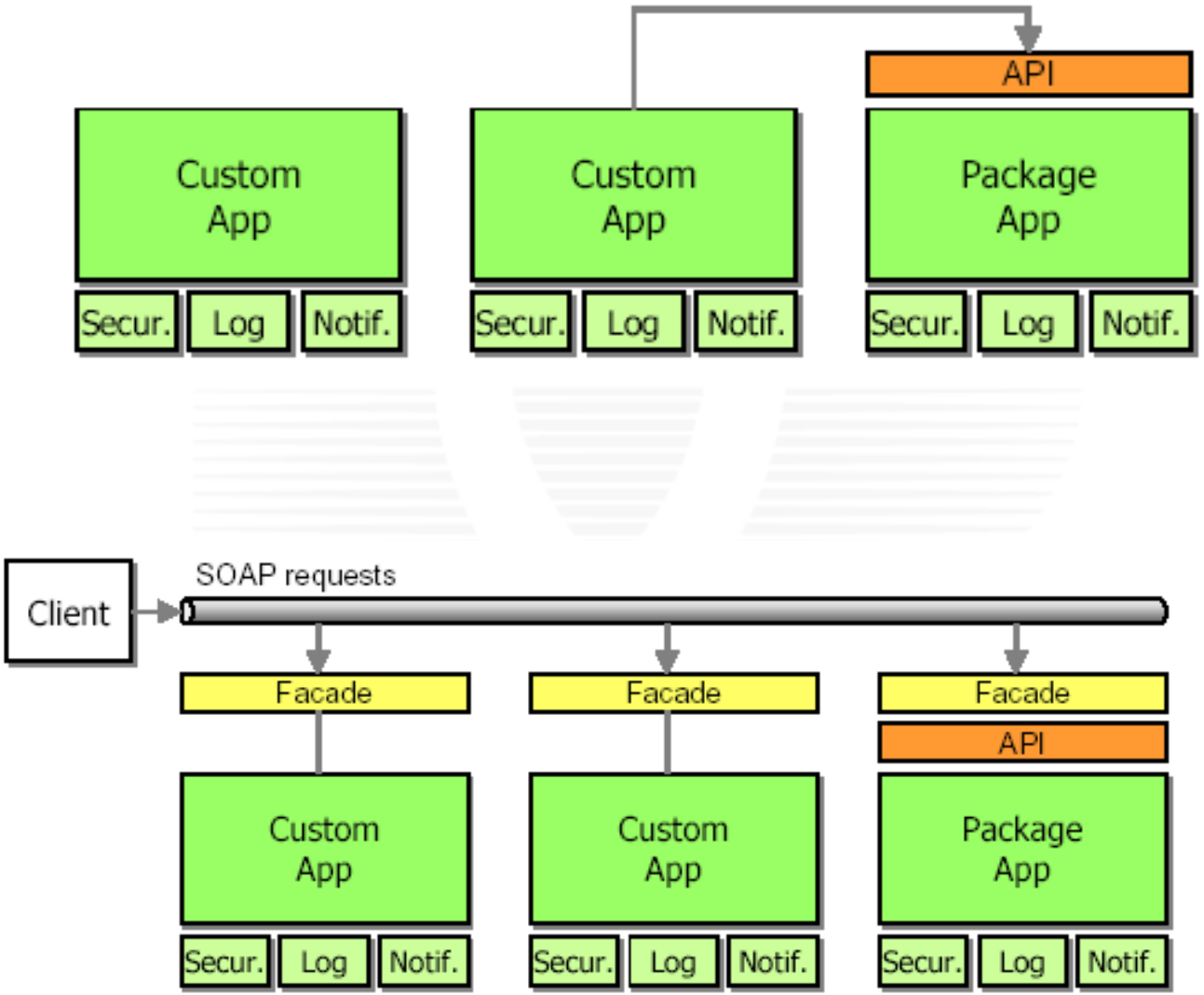
Service-Oriented Architecture (SOA)

- Aplicações monolíticas são dissolvidas em serviços que executam funções de negócio específicas.
- Novas aplicações são montadas a partir de um conjunto de serviços publicados por fornecedores internos ou externos à empresa.
- O conjunto de serviços disponíveis é gerenciado a partir de um registro central.

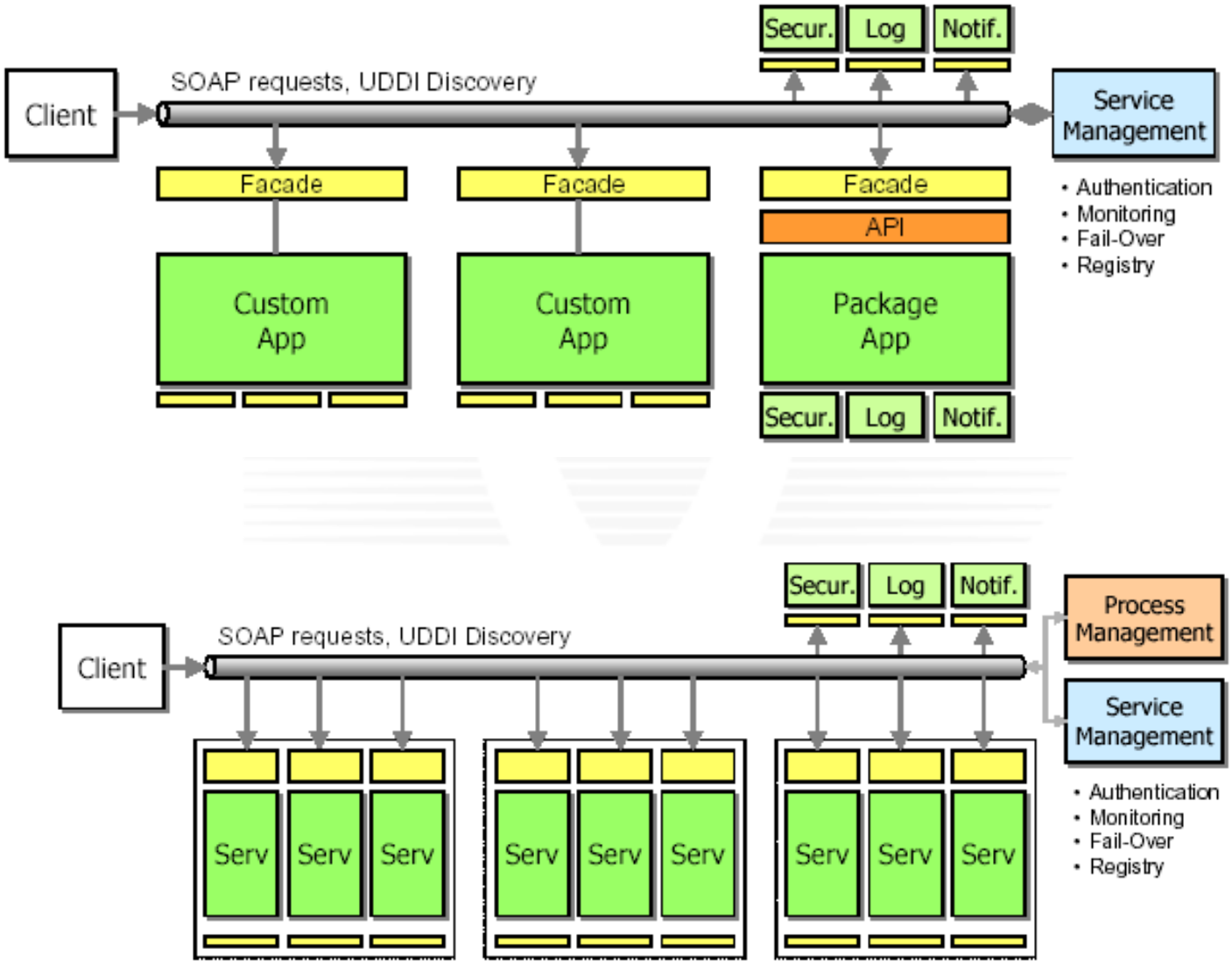
Service-Oriented Architecture (SOA)



Service-Oriented Architecture (SOA)



Service-Oriented Architecture (SOA)



DATASUL

NOVOS
CONCEITOS.
GRANDES
RESULTADOS.

Web Services - Definição

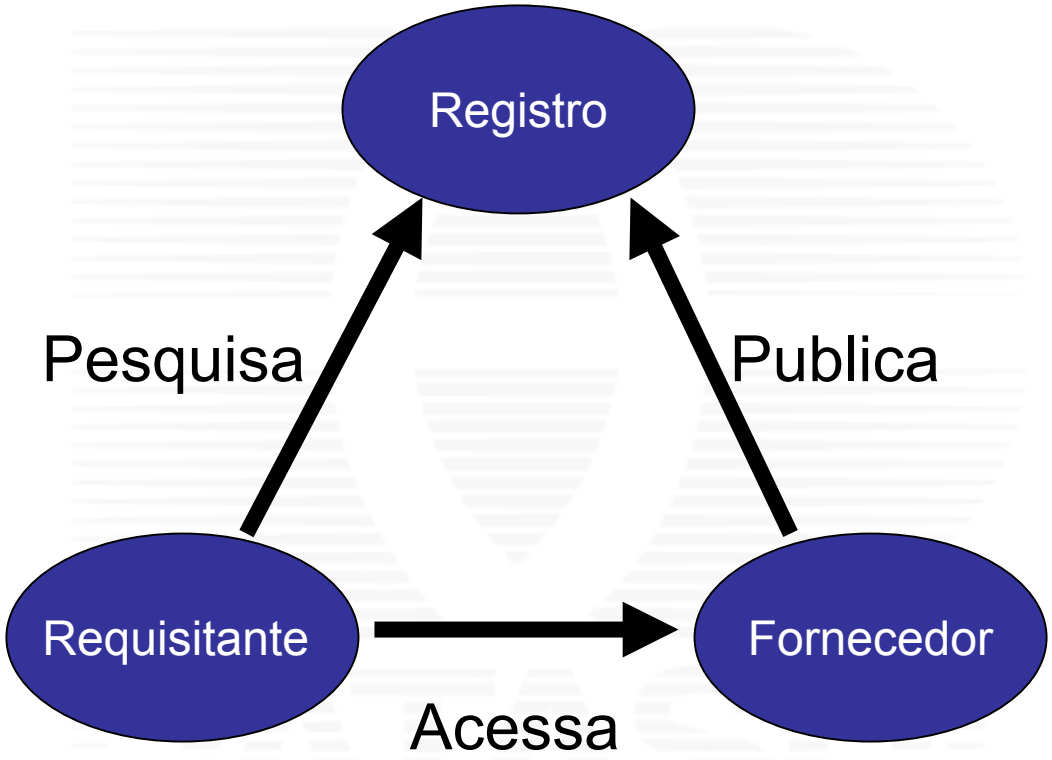
Nov/2003

Web Services - Definição

- Web Service é uma maneira de expor funcionalidades para usuários Web através de protocolos padrão.

- Web Service é uma aplicação identificada por uma URI (Uniform Resource Identifier), cujas interfaces podem ser descobertas e definidas através de artefatos XML, e que suporta interações diretas com outros softwares utilizando mensagens XML através de protocolos padrão da Internet (W3C).

Web Services - Papéis



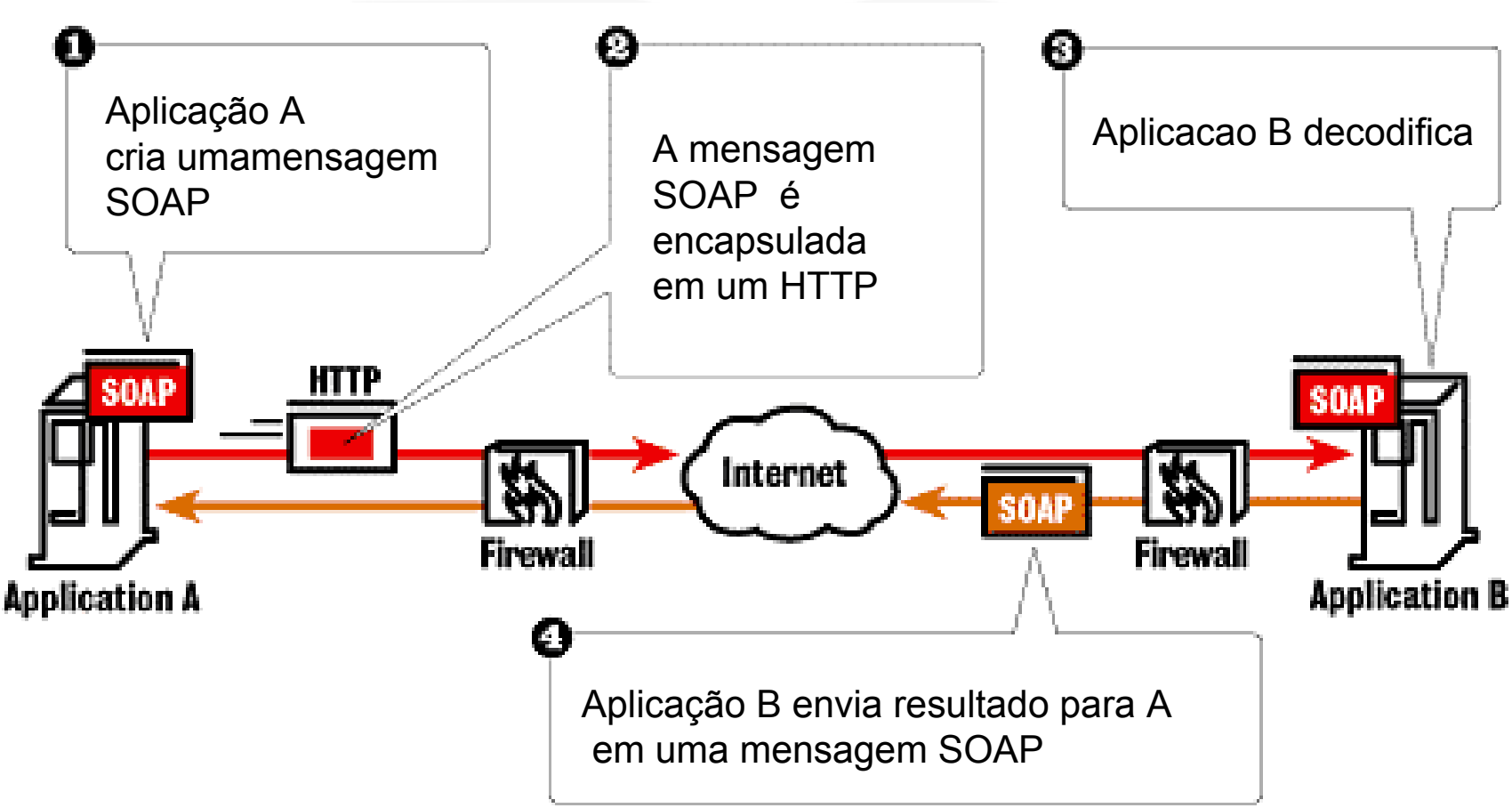
Propriedades

- Baseados em padrões da indústria.
- Independentes de liguagens e plataformas.
- Transparentes para firewalls.
- Auto-descritíveis.
- Fracamente acoplados.

Padrões

- SOAP (Simple Object Access Protocol)
  - Define o formato que as mensagens devem ter.
- WSDL (Web Services Description Language)
  - Descreve as interfaces dos Web Services e como invocar uma operação.
- UDDI (Universal Description, Discovery, and Integration)
  - Padrão de especificações para descrição e descoberta de serviços
  - UDDI Registry: local onde os serviços são encontrados

Executando um serviço



DATASUL

NOVOS
CONCEITOS.
GRANDES
RESULTADOS.

Web Services – Padrões associados

Nov/2003

SOAP - Simple Object Access Protocol

- Padrão que define a estrutura que uma mensagem XML deve ter para ser utilizada em Web Services.

- Sua implementação padrão trabalha sobre HTTP, o que permite que as mensagens passem por firewalls.

SOAP - Simple Object Access Protocol

- Uma mensagem SOAP é um documento XML contendo os seguintes elementos:
  - Envelope (obrigatório): Identifica o documento XML como uma mensagem SOAP.
  - Header (opcional): Contém informações de controle para o processamento da mensagem.
  - Body (obrigatório): Contém a carga útil da mensagem.
  - Fault (opcional): Fornece informação sobre erros que ocorreram durante o processamento da mensagem

SOAP - Simple Object Access Protocol

- Exemplo:

```
POST /test/simple.asmx HTTP/1.1
Host: 131.107.72.13
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://soapinterop.org/echoString"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns="http://soapinterop.org/"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <tns:echoString>
            <inputString>Teste</inputString>
        </tns:echoString>
    </soap:Body>
</soap:Envelope>
```

WSDL - Web Services Description Language

- Um documento WSDL é um documento XML que descreve Web Services como um conjunto de pontos de serviço (endpoints) que operam baseados em trocas de mensagens.

- As operações e mensagens relativas a um serviço são descritas de forma abstrata e em seguida ligadas a protocolos de rede e formatos de mensagens concretos com o objetivo de definir um ponto de serviço.

WSDL - Web Services Description Language

- O uso de WSDL na arquitetura de Web Services é em geral dividido em duas partes:
  - interface do serviço
  - implementação do serviço.

- Cada parte pode ser definida de maneira independente e, conseqüentemente, reutilizada por outras aplicações.

WSDL - Elementos da Interface de Serviço

- **types**
  - Definem os tipos de dados que são utilizados para descrever as mensagens. Para melhor interoperabilidade e independência de plataforma indica-se o uso de XSD (XML Schema Documents).

```
<wsdl:types>
   <xsd:schema>
     <xsd:complexType name="DVDAbstractType">
              <xsd:sequence>
                   <xsd:element name="title" type="xsd:string"/>
                   <xsd:element name="artist" type="xsd:string"/>
                   <xsd:element name="releaseDate" type="xsd:date"/>
              </xsd:sequence>
     </xsd:complexType>
   </xsd:schema>
</wsdl:types>
```

WSDL - Elementos da Interface de Serviço

- **message**
  - Agrupa dados (cujos tipos foram estruturados na seção *type*) em uma assinatura, dando-lhes um nome que será referenciado em operações.

```
<wsdl:message name="Subscribe">
   <wsdl:part name="email" type="tns:EmailAddressType"/>
   <wsdl:part name="until" type="xsd:date"/>
</wsdl:message>
```

WSDL - Elementos da Interface de Serviço

- **portType**
  - É equivalente ao conceito de interface do Java. Agrupa referências ao elemento <message> na forma de operações e lhes dá um nome.

```
<wsdl:portType name="DVDRenting">
   <wsdl:operation name="SubscribeToSpecialsAlertList">
     <wsdl:input message="tns:Subscribe"/>
   </wsdl:operation>
   <wsdl:operation name="RentDVD">
     <wsdl:input message="tns:RentalRequest"/>
     <wsdl:output message="tns:PurchaseResponse"/>
     <wsdl:fault name="tns:RentFault" message="tns:PaymentFault"/>
   </wsdl:operation>
</wsdl:portType>
```

WSDL - Elementos da Interface de Serviço

- **binding**
  - Define detalhes de comunicação para um elemento portType.

```
<binding name="StockQuoteSoapBinding" type="StockQuotePortType">
   <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http"/>
   <operation name="GetLastTradePrice">
     <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
     <input>
             <soap:body use="literal"/>
     </input>
     <output>
             <soap:body use="literal"/>
     </output>
   </operation>
</binding>
```

WSDL - Elementos da Implementação do Serviço

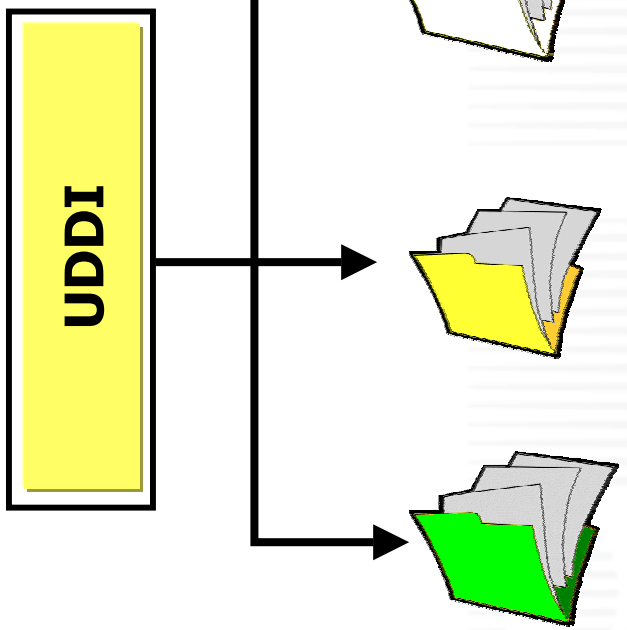
- **service**
  - Especifica um processo que pode atender as solicitações para um determinado elemento binding.

```
<service name="StockQuoteService">
   <documentation>My first service</documentation>
   <port name="StockQuotePort" binding="tns:StockQuoteBinding">
     <soap:address location="http://localhost:6080/stockquote"/>
   </port>
</service>
```

UDDI (Universal Description, Discovery, and Integration)

- UDDI é uma especificação para criar registros distribuídos de serviços na Internet.
- Um registro UDDI armazena informações sobre empresas, serviços oferecidos por estas empresas e informações técnicas sobre estes serviços.
- Um serviço publicado num registro UDDI não precisa ser, necessariamente, um Web Services.

UDDI (Universal Description, Discovery, and Integration)



- Páginas brancas: contêm o endereço, pessoas de contato e outros identificadores relativos ao negócio onde a empresa atua;
- Páginas amarelas: incluem categorizações industriais baseados no tipo do serviço;
- Páginas verdes: Contêm informações técnicas sobre os serviços expostos pelo provedor.

DATASUL

NOVOS
CONCEITOS.
GRANDES
RESULTADOS.

Web Services - APIs e ferramentas Java

Nov/2003

JWSDP - Java Web Services Developer Pack

- Java API for XML Processing (JAXP)
- Java Architecture for XML Binding (JAXB)
- Java API for XML-based RPC (JAX-RPC)
- Java API for XML Messaging (JAXM)
- Java API for XML Registries (JAXR)

Java API for XML Processing (JAXP):

- Oferece a possibilidade de fazer a interpretação de documentos XML utilizando o modelo DOM ou SAX.
- Permite fazer a transformação de documentos XML através de XSLT.

JWSDP - JAXP - Parser SAX

- Utilizado apenas para interpretar mensagens.
- São disparados eventos enquanto a mensagem é analisada.
- É construída uma classe que estende a classe DefaultHandler, que por sua vez é a implementação padrão da interface ContentHandler.

```
<priceList> [parser calls startElement]
    <coffee> [parser calls startElement]
        <name>Mocha Java</name> [parser calls startElement, characters, and endElement]
        <price>11.95</price> [parser calls startElement, characters, and endElement]
    </coffee> [parser calls endElement]
</priceList> [parser calls endElement]
```

JWSDP - JAXP - Parser DOM

- Utilizado para interpretar e alterar/criar mensagens.
- Monta em memória uma representação do documento numa estrutura de árvore.

```
Node rootNode = document.getDocumentElement();
NodeList list = document.getElementsByTagName("coffee");
for (int i=0; i < list.getLength(); i++) {
thisCoffeeNode = list.item(i);
Node thisNameNode = thisCoffeeNode.getFirstChild();
if (! thisNameNode.getFirstChild() instanceof org.w3c.dom.Text) continue;
    String data = thisNameNode.getFirstChild().getNodeValue();
    if (! data.equals("Mocha Java")) continue;
    Node newCoffeeNode = document.createElement("coffee");
    Node newNameNode = document.createElement("name");
    Text tnNode = document.createTextNode("Kona");
    newNameNode.appendChild(tnNode);
    newCoffeeNode.appendChild(newNameNode);
    …
    rootNode.insertBefore(newCoffeeNode, thisCoffeeNode);
    break;
}
```

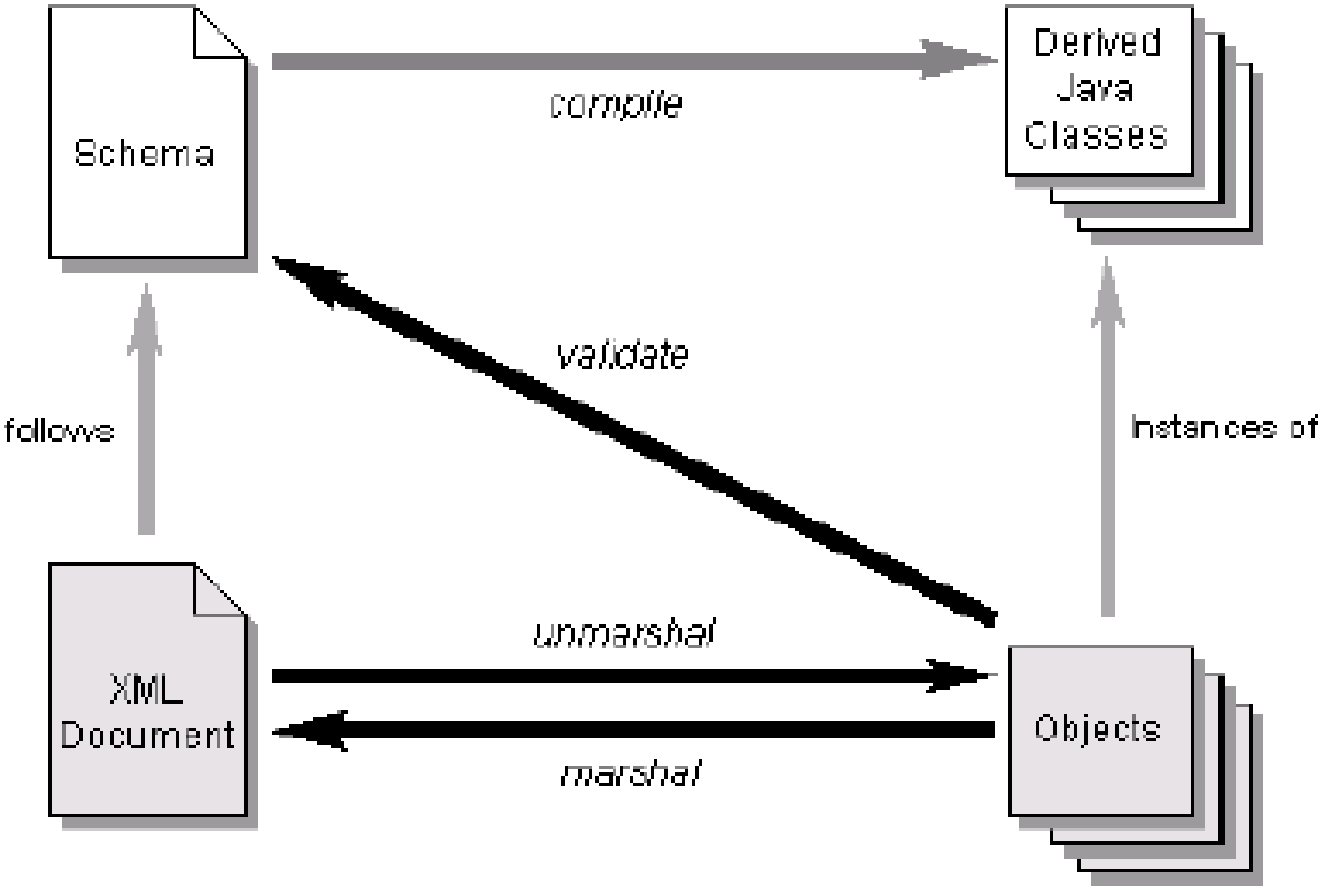
Java Architecture for XML Binding (JAXB):

- Permite gerar classes a partir de XML Schemas.
- Elimina a necessidade de codificar classes que manipulem XML.

JWSDP - JAXB

XML Schema	Java Class Files
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">	
<xsd:element name="purchaseOrder" type="PurchaseOrderType"/>	PurchaseOrder.java
<xsd:element name="comment" type="xsd:string"/>	Comment.java
<xsd:complexType name="PurchaseOrderType"> <xsd:sequence> <xsd:element name="shipTo" type="USAddress"/> <xsd:element name="billTo" type="USAddress"/> <xsd:element ref="comment" minOccurs="0"/> </xsd:sequence> <xsd:attribute name="orderDate" type="xsd:date"/> </xsd:complexType>	PurchaseOrder- Type.java
<xsd:complexType name="USAddress"> <xsd:sequence> <xsd:element name="name" type="xsd:string"/> <xsd:element name="street" type="xsd:string"/> <xsd:element name="city" type="xsd:string"/> <xsd:element name="state" type="xsd:string"/> <xsd:element name="zip" type="xsd:decimal"/> </xsd:sequence> <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/> </xsd:complexType>	USAddress.java
</xsd:schema>	

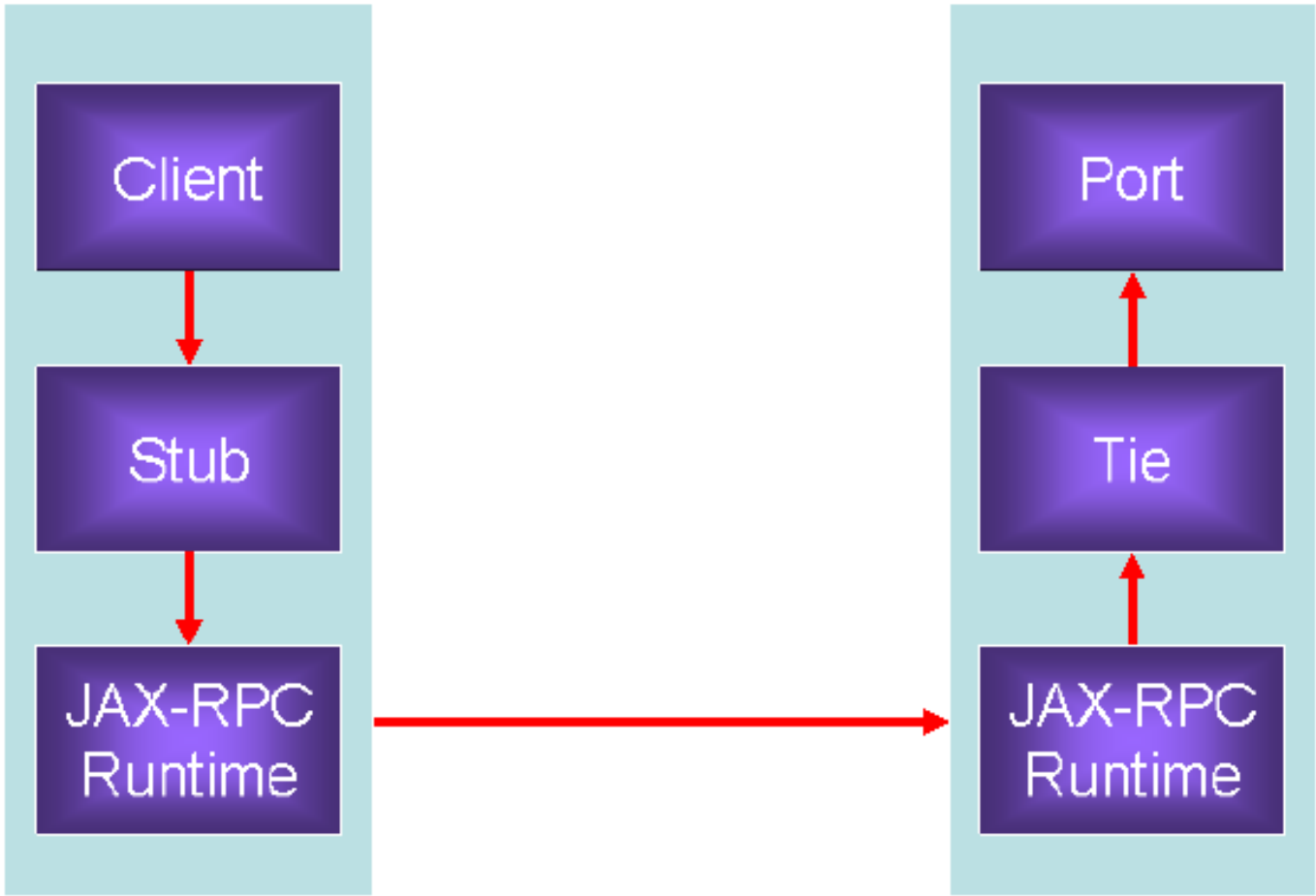
JWSDP - JAXB


Schema
compile
Derived
Java
Classes
validate
follows
Instances of
XML
Document
unmarshal
marshal
Objects

Java API for XML-based RPC (JAX-RPC):

- Facilita o uso de Web Services no modelo de Remote Procedure Call (RPC).
- Permite gerar boa parte da infra-estrutura de um Web Service.
- Dispensa a manipulação de mensagens SOAP.

JWSDP - JAX-RPC



JWSDP - JAX-RPC - Lado Servidor

- Criar uma interface que exponha os métodos disponibilizados pelo Web Service.
- Criar uma classe que implemente estes métodos.
- Utilizar a ferramenta *wsdeploy* para gerar o "tie" e o arquivo WSDL.
- Empacotar a aplicação em um arquivo .war.
- Fazer o deploy da aplicação em um Web Container.

JWSDP - JAX-RPC - Lado Servidor

```
package coffees;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface CoffeeOrderIF extends Remote {
   public Coffee [] getPriceList() throws RemoteException;
   public String orderCoffee(String coffeeName, int quantity) throws RemoteException;
}
```

```
package coffees;
public class CoffeeOrderImpl implements CoffeeOrderIF {

   public Coffee [] getPriceList() throws RemoteException; {
      . . .
   }

   public String orderCoffee(String coffeeName,int quantity) throws RemoteException; {
   . . .
   }

}
```

JWSDP - JAX-RPC - Lado Cliente

- Utilizar a ferramenta *wscompile* para gerar classes para um "stub" e um "stub factory" a partir do arquivo WSDL.
- Obter uma instância do "stub" a partir do "stub factory".
- Executar os métodos do Web Service como se eles fossem do "stub".

JWSDP - JAX-RPC – Lado Cliente

```
package coffees;
public class CoffeeClient {
   public static void main(String[] args) {
      try {
         CoffeeOrderIF coffeeOrder = new CoffeeOrderServiceImpl().getCoffeeOrderIF();
         Coffee [] priceList = coffeeOrder.getPriceList();
         for (int i = 0; i < priceList.length; i++) {
            System.out.print(priceList[i].getName() + " ");
            System.out.println(priceList[i].getPrice());
         }
      }
      catch (Exception ex) {
         ex.printStackTrace();
      }
   }
}
```

JWSDP - Java Web Services Developer Pack

- Java API for XML Messaging (JAXM)
  - Fornece uma maneira padrão de transmitir documentos XML pela Internet, seguindo as especificações SOAP 1.1 e SOAP with Attachments.

- Java API for XML Registries (JAXR)
  - Fornece uma maneira simples de acessar registros de negócios na Internet. Estes tanto podem ser baseados em padrões abertos (como ebXML) ou especificações de consórcios de empresas (como UDDI).

Axis - Apache EXtensible Interaction System

- Framework que é uma evolução do Apache SOAP, que por sua vez nasceu como SOAP4J da IBM.
- Fornece um servidor que pode ser acoplado a "servlet engines" como o Tomcat, dentre outras coisas.
- Ferramentas para geração de WSDL a partir de classes Java e o inverso.
- Oferece uma maneira de transformar classes Java em Web Services simples (JWS).
- Web Services mais complexos podem ser configurados através de Web Service Deployment Descriptor (WSDD).

IBM WebSphere SDK for Web Services (WSDK)

- Oferece uma implementação para a especificação JSR-101 (JAX-RPC).
- Estende o framework Axis.
- Oferece integração com a IDE Eclipse e com o WebSphere Application Server.
- Speed-start Web Services: Conjunto de produtos da IBM para o desenvolvimento de Web Services e tutoriais a este respeito.

Obrigado!

Roger Pedroso
Datasul Tecnologia