

A integração SOA & Web 2.0

Autor:

José Ricardo de Oliveira Damico

Objetivo

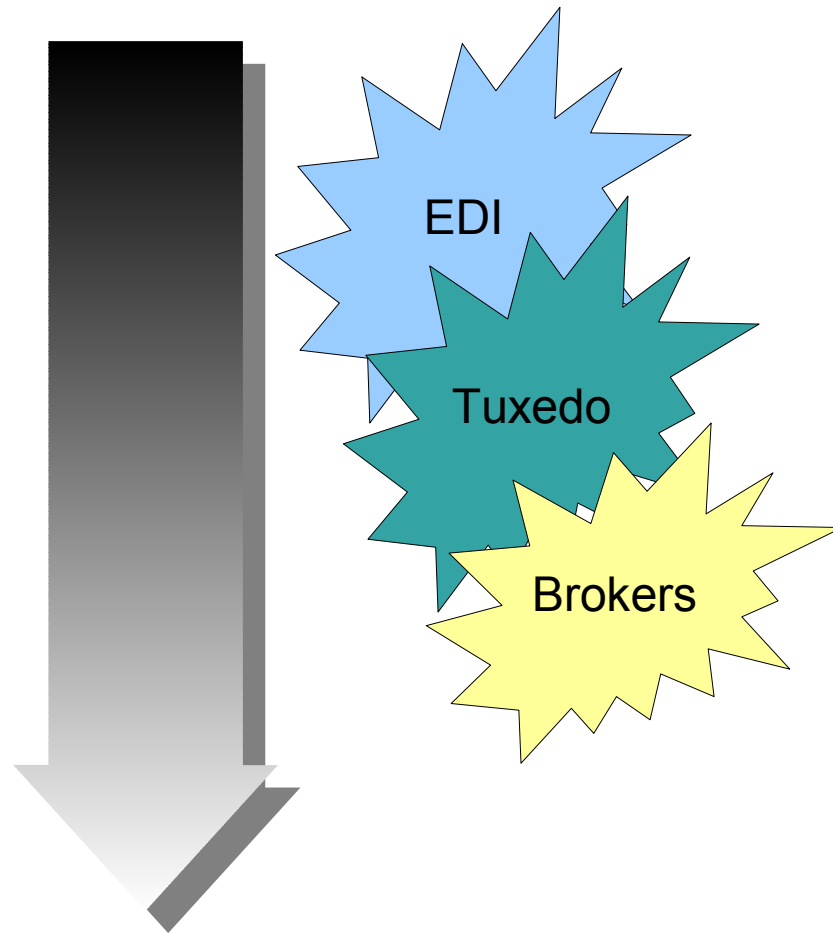
- **Esclarecer os principais aspectos de uma arquitetura orientada à serviços**
- **Apresentar a importância dos padrões XML em arquiteturas orientadas à serviços.**
- **Demonstrar alguns exemplos**
- **Sugerir linhas de estudos**

O Fundamento

- ***“Na era da informação, a produção de linhas de código é um dos mais importantes fundamentos do mercado.”***

Breve Cronologia das Aplicações Empresariais

- **ALL-IN-ONE**
- **Cliente - Servidor**
- **Três camadas**
- **CORBA**
- **RMI**
- **JMS**
- **Web-based**
- **P2P (Grid)**

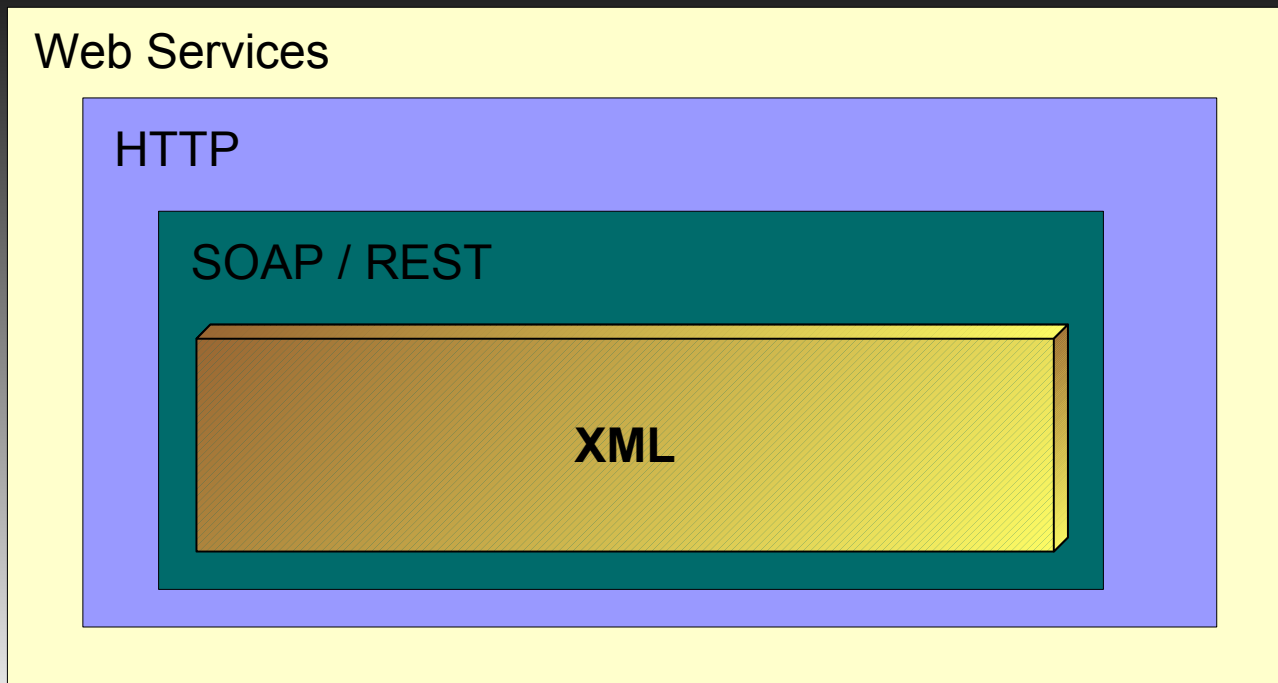


Os passos para evolução

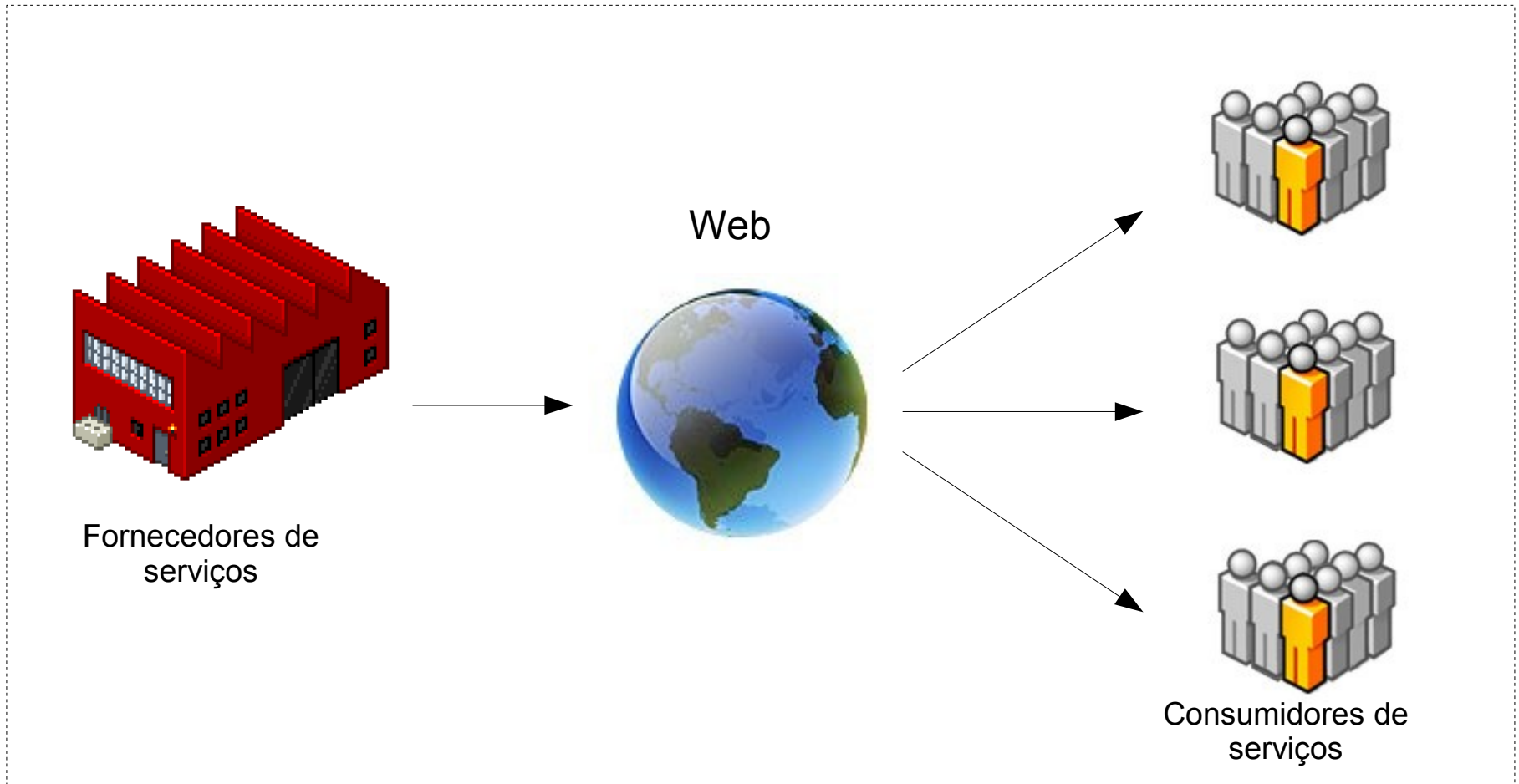
- **Independência de linguagem**
- **Objetos distribuídos**
- **Novo Modelo de mensagem (Conteúdo + Requisição)**
- **Requisições Assíncronas**
- **Nova camada genérica de abstração das regras de negócio**

Padrão de Encapsulamento

WEB

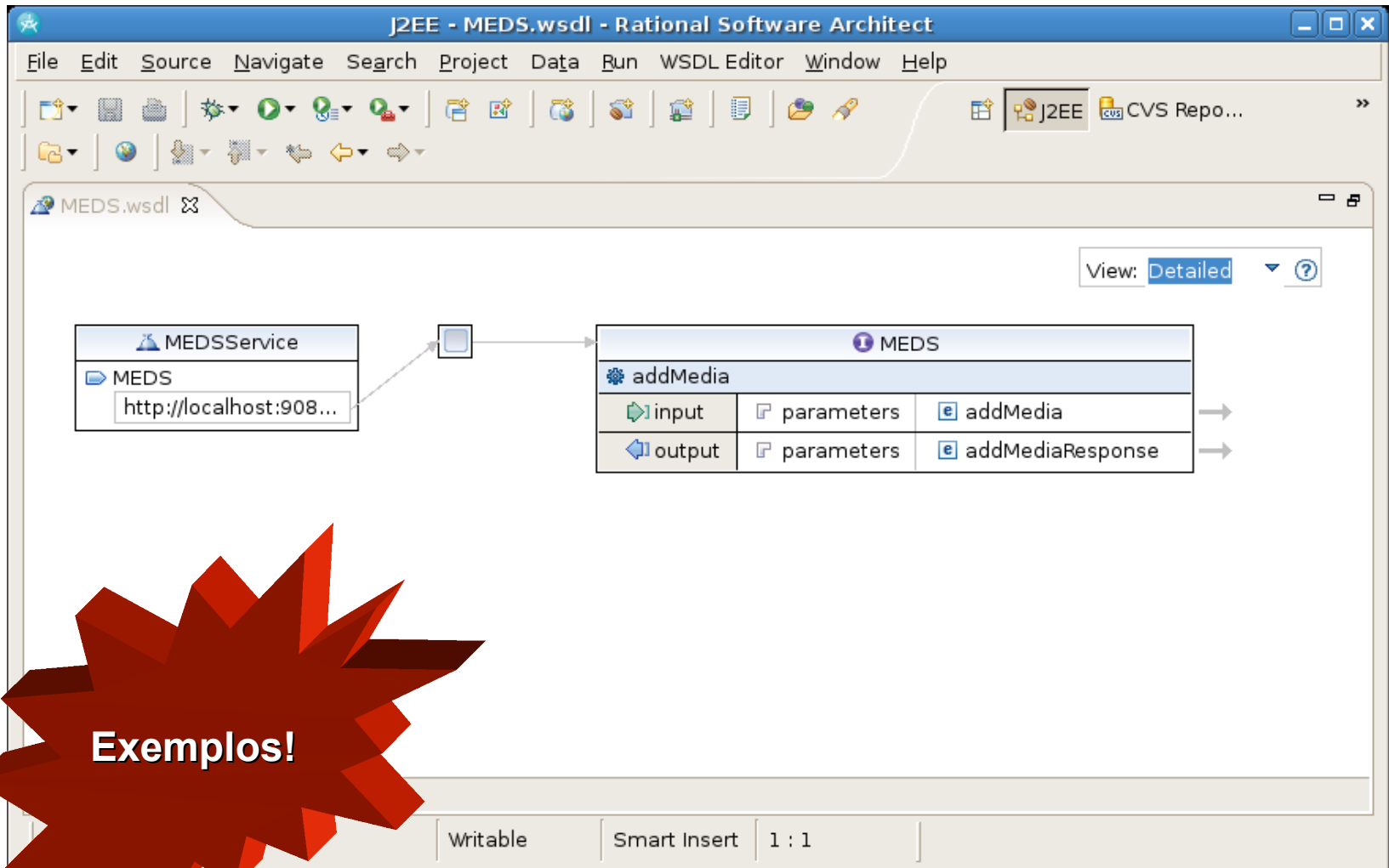


O Modelo de Serviços



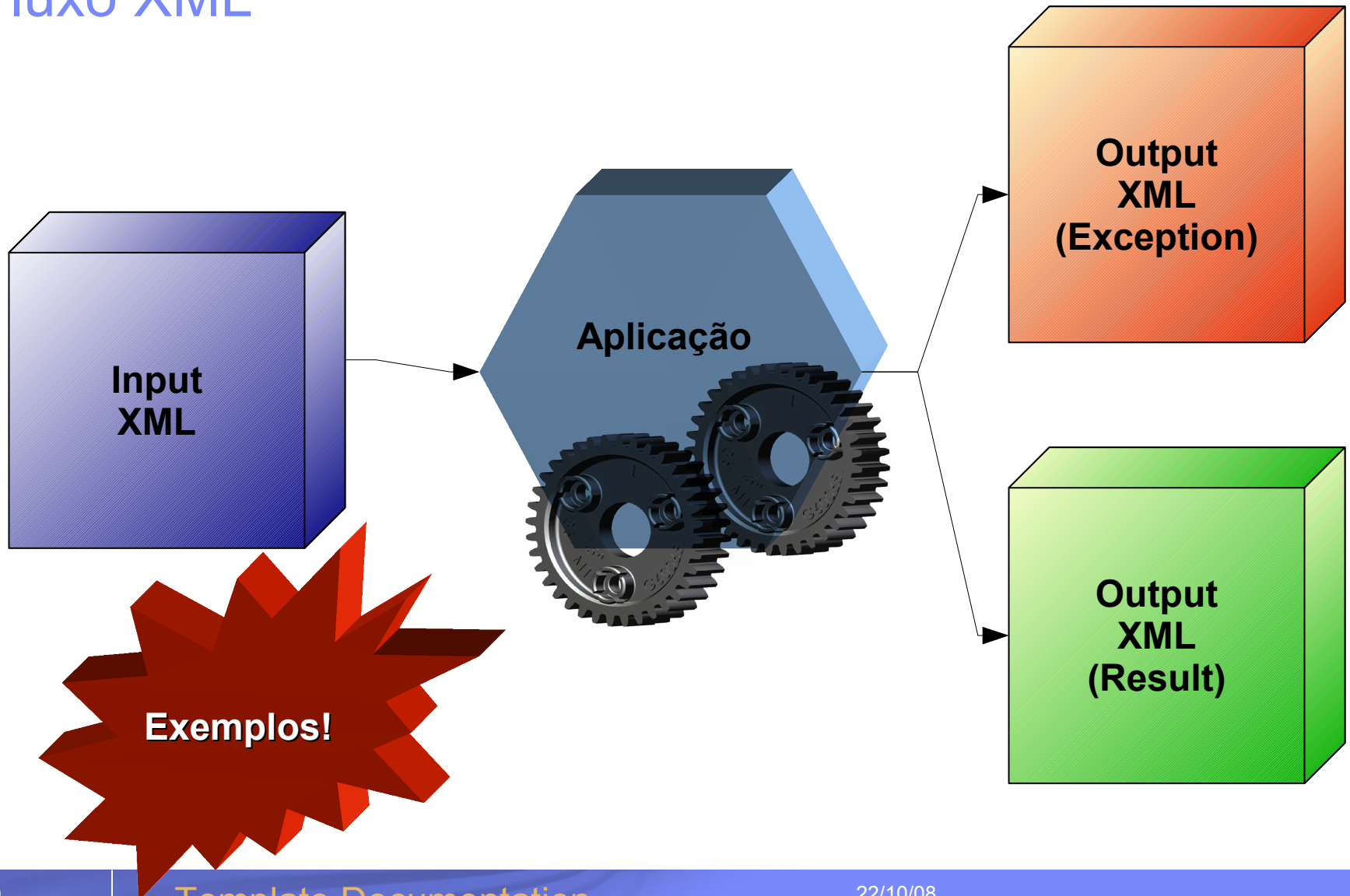
Ambientes, Infraestruturas, Línguas, Protocolos, Sistemas >> HETEROGÊNEOS

Mas, Como representar um serviço?

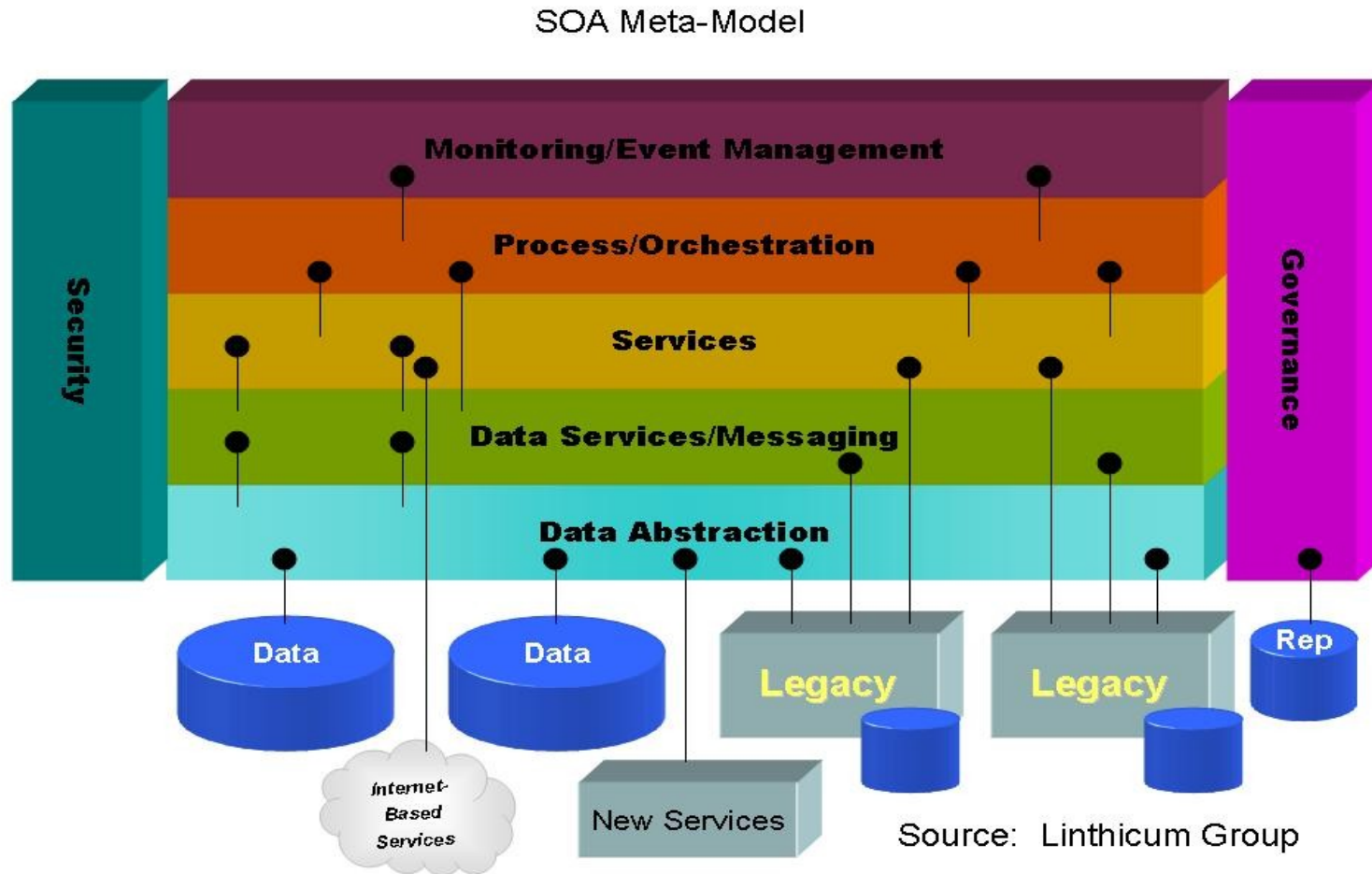


Exemplos!

Fluxo XML



XML é a chave da transferência entre as camadas do modelo SOA



O que acontece quando o XML acessa a aplicação? *(Parte I)*

- **XML acessa o serviço via SOAP ou REST**
- **O serviço é formado por um conjunto de classes ou por um ou mais componentes**
- **O XML tem em si definido uma das classes e métodos de entrada**
- **Além disso o XML de entrada também defin, e quando aplicável, os valores a serem passado para classe**



O que acontece quando o XML acessa a aplicação? *(Parte II)*

- **O XML é validado**
- **Faz a chamada do serviço**
- **Um XML de saída passa a ser construído ou transformado a partir do XML de entrada**
- **A transformação de XML em objetos (e vice-versa) é comum (SDO é uma boa abordagem)**

<http://www.ibm.com/developerworks/library/ws-sdoxmlschema/>



Caixa de Ferramentas XML

- **XSD:** Manter a forma
- **XSLT:** Transformar sem perder o modelo
- **XQuery:** Consulta e modelagem dinâmica de conteúdo

Bibliotecas essenciais:

- Xalan (<http://xerces.apache.org/>)
- Xerces (<http://xalan.apache.org/>)
- Saxon (<http://saxon.sourceforge.net>)
- JAXP (<http://java.sun.com/webservices/jaxp>)



Exemplos!

Integração do XML com Bancos de Dados



Java DB



DB2 Enterprise 9

Information Management

software

http://www.regdeveloper.co.uk/2006/11/08/java_data

<http://www-306.ibm.com/software/data/db2/9/edition-e>

<https://www6.software.ibm.com/developerworks/demc>

A Evolução do SQL

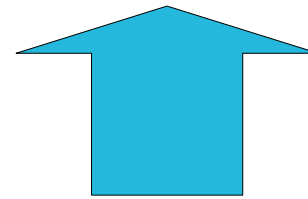
```
SELECT ID,
       XMLSERIALIZE (
         XMLQUERY('string(//student[text() = "BC"]/@age)' PASSING BY REF xcol EMPTY ON EMPTY)
         AS VARCHAR(50))
FROM x_table
WHERE
       XMLEXISTS('//student[text() = "BC"]' PASSING BY REF xcol)
```

```
update clientes set contatos=
(xmlparse(document
'\<Cliente>
  <Endereco>
    <rua>Rua dos Pássaros 154</rua>
    <cidade>Sao Paulo</cidade>
    <estado>SP</estado>
    <cep>88888888</cep>
  </Endereco>
  <telefones>
    <comercial>551833457898</comercial>
    <residencial>551833454444</residencial>
    <celular>551897054849</celular>
  </telefones>
  <email>email@email.com.br</email>
</Cliente>
' ) )
where codigo = 4;
```

```
INSERT INTO x_table VALUES
(1,
 XMLPARSE(DOCUMENT '
  <roster>
    <student age="18">AB</student>
    <student age="23">BC</student>
    <student>NOAGE</student>
  </roster>'
  PRESERVE WHITESPACE)
)
```

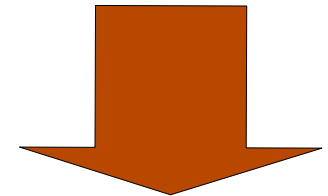
SOA + Web 2.0

- Parseamento de XML no Browser
- Volta do JavaScript



Aumento de
Processamento
Local

Diminuição
custo
Infraestrutura



Necessidade de um modelo
hierarquico mais leve

Influência de
padrões abertos



SOAP & REST

- **Serviços entre Máquinas**
 - Detalhamento lógico e o mapeamento de opções leva ao resultado
- **Serviços entre Máquinas e Pessoas**
 - Estado representacional leva ao resultado

Input SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body> </soap:Envelope>
```

Input REST

<http://en.wikipedia.org/wiki/SOAP>
<http://dictionary.reference.com/browse/day>

Implementação AJAX

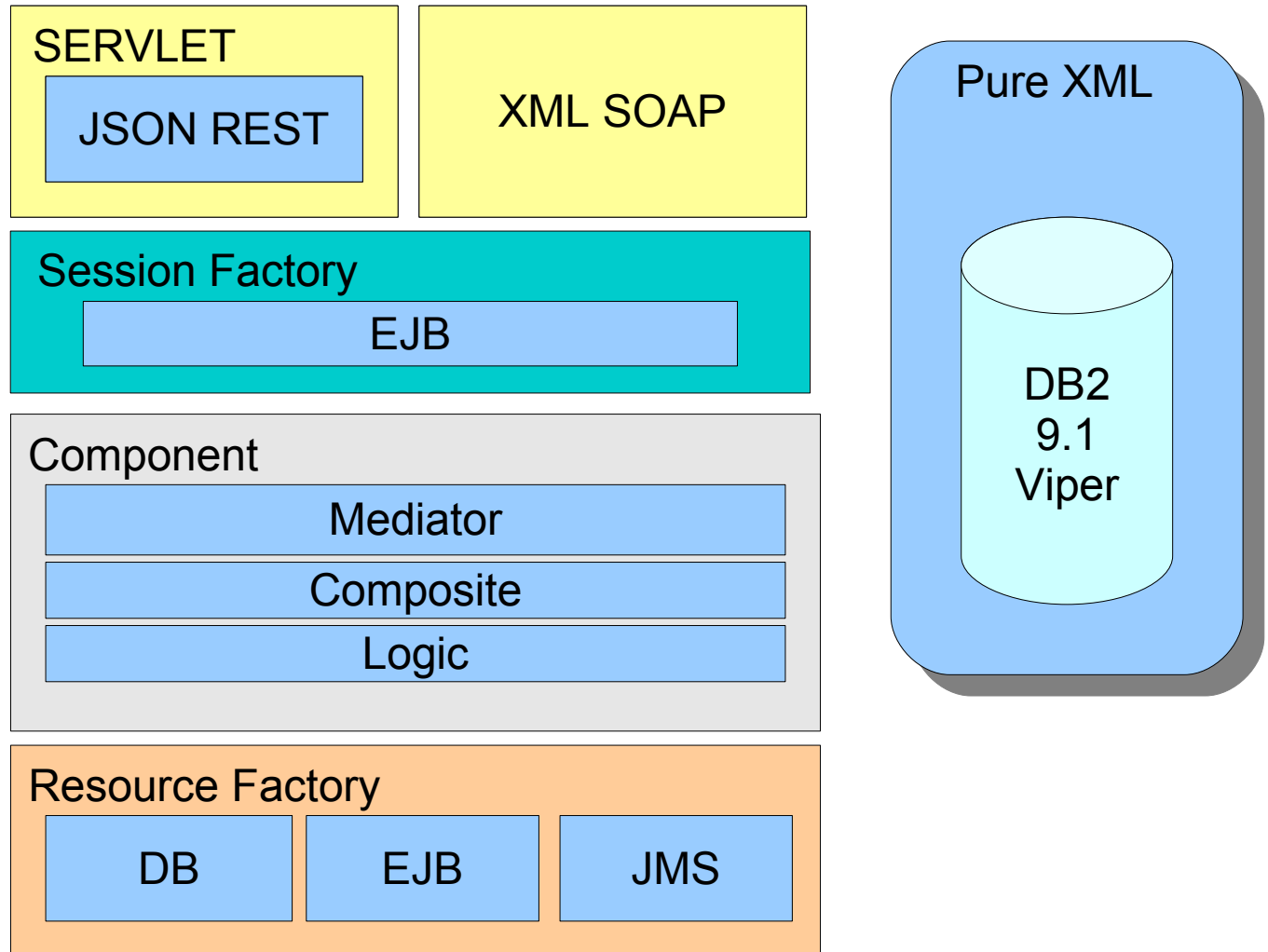
DOJO

JSON

```
{"universe":{"payload":{"entity":  
{"type":"Usr","uld":"vfc"}}, "msgHeader":  
{"retCd":"0"}}}
```

REST

Arquitetura SOA Básica



Conclusão

- ***“O modelo de uma arquitetura orientada à serviços depende do fluxo de informações e requisições baseadas em documentos XML. Mesmo pequenas aplicações devem estar de acordo com este modelo para garantir o futuro de seus ciclos de vida.”***

Referências

- **WebServices**

- <http://www.ibm.com/developerworks/webservice>

- **SOAP**

- <http://www.ibm.com/developerworks/webservice>

- **REST**

- <http://en.wikipedia.org/wiki/REST>

- **SOA**

- <http://www.ibm.com/developerworks/webservice>

- **ESB**

- <http://www.ibm.com/developerworks/library/ar-e>

- **Jetty**

- <http://docs.codehaus.org/display/JETTY/Embec>

- **XSLT**

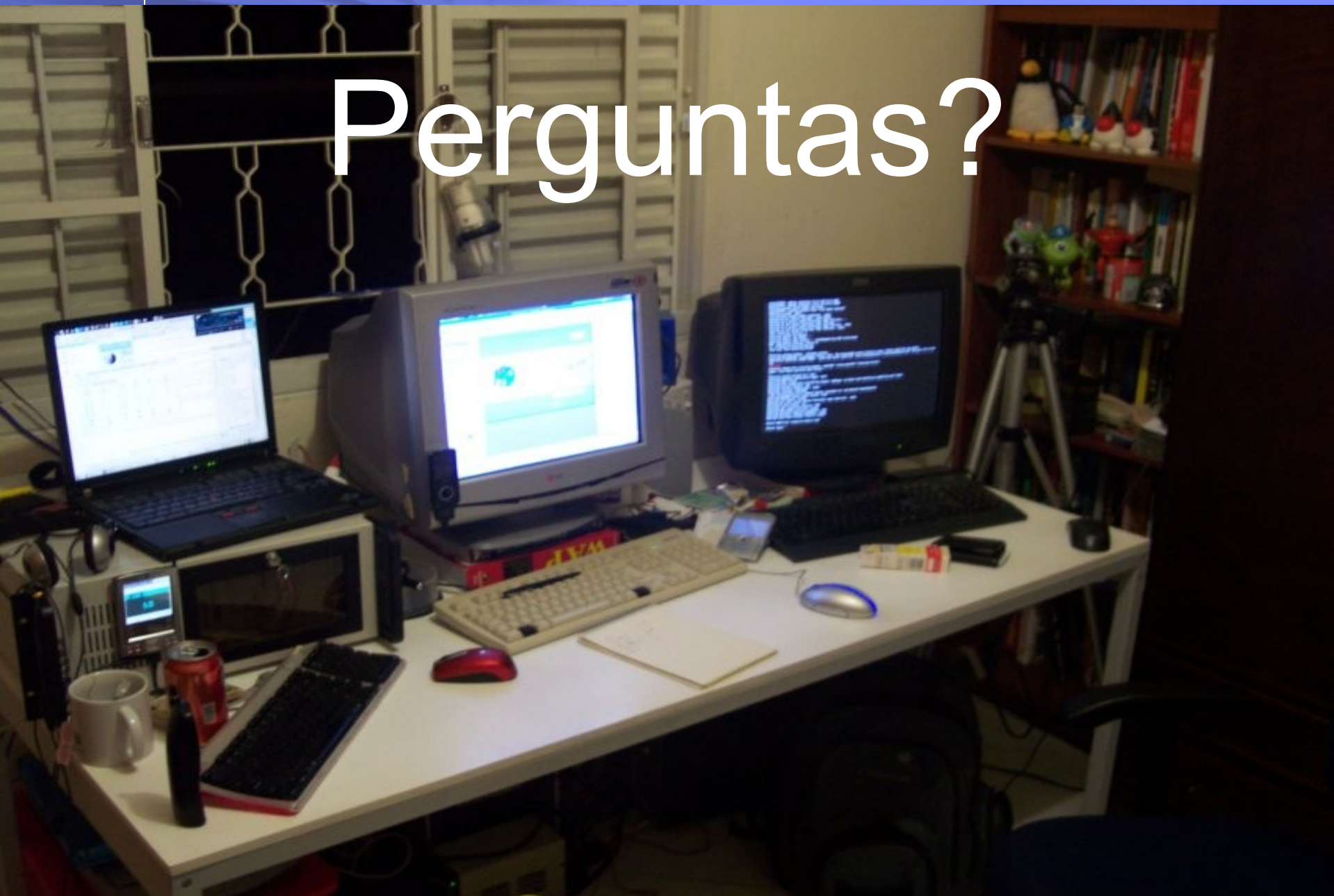
- <http://www.ibm.com/developerworks/xml/library>

- **Xquery**

- <http://www.ibm.com/developerworks/xml/library>

- <http://dcon.com.br/XQueryREST/>

Perguntas?





IBM Software *Live!*

Thank You!

<http://dcon.com.br>