

Arquiteturas de Integração

CAPÍTULO 2. PADRÕES DE INTEGRAÇÃO

PROF. DIOVANI LUIZ MERLO

Arquiteturas de Integração

AULA 2.1. INTRODUÇÃO AOS PADRÕES DE INTEGRAÇÃO

PROF. DIOVANI LUIZ MERLO

Nesta aula



- ❑ Introdução aos padrões de integração.
- ❑ EDI – Electronic Data Interchange.
- ❑ EAI Patterns.

Introdução

A escalada da economia (globalização) e a evolução das soluções de tecnologia da informação (TI) e das redes de comunicação, permitiram que a necessidade de integração se estabelecesse entre as aplicações corporativas.

Atualmente existem vários padrões, princípios e estilos arquiteturais que podem ser utilizados para realizar a integração entre diversas soluções de software, plataformas e produtos.

Eletronic Data Interchange



1

Eletronic Data Interchange (EDI) ou Troca de Informações Eletrônicas, é um conceito estabelecido para os mecanismos de troca eletrônica de informação estruturada e a realização de transações entre empresas (B2B) com mínima ou nenhuma intervenção humana.

2

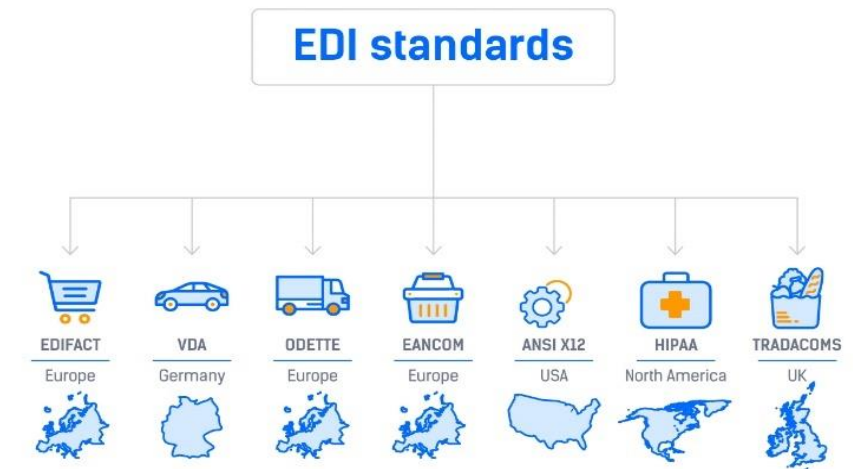
O conceito parece simples, contudo, quando documentos são trocados utilizando o formato EDI, as soluções provêm mais rapidez nas integrações, mais assertividade e mecanismos mais baratos quando se comparado a outros mecanismos de integração (STULTZ, 2001).

Eletronic Data Interchange



Os exemplos mais conhecidos de padrões de EDI são:

- **ANSI X12:** American National Standards Institute foi o responsável pela criação deste formato. É o formato EDI mais popular e utilizado quando comparado aos demais, tendo em vista sua utilização em áreas como saúde, comércio varejista e até o setor automobilístico.
- **EDIFACT:** Stands For Electronic Data Interchange for Administration, Commerce and Transport. Foi desenvolvido pelas nações unidas, mas é adotado em vários países com objetivos de realizar transações.



Fonte: NAIR (2018)

Exemplo EDI ANSI X12/EDIFACT

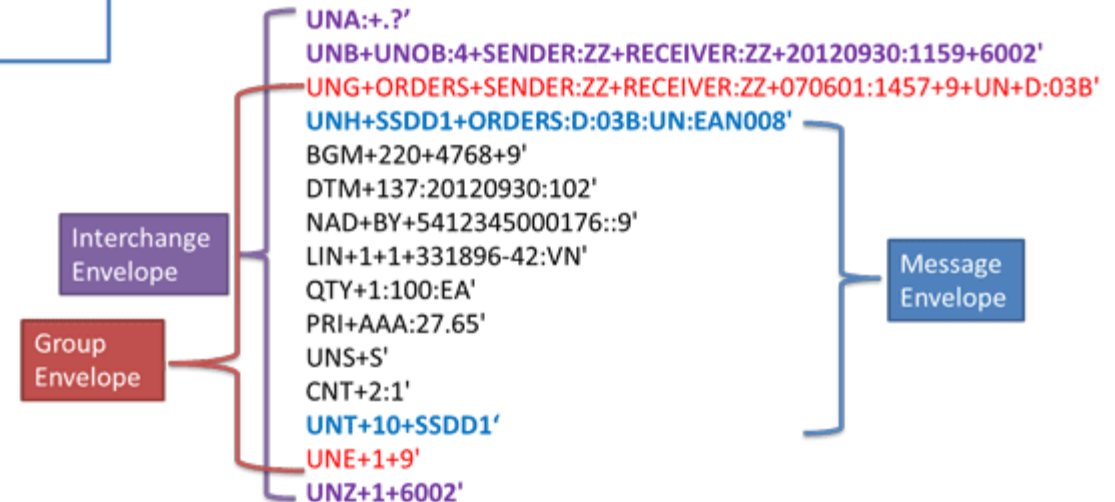


Paper Purchase Order

Purchase Order				
Sold To: XYZ Company		PO Number: 4768		
123 Main Street		PO Date: 9/30/2012		
Fairview, CA 94168				
Item No.	Quantity	Unit of Measure	Price	Product ID
1	100	EA	27.65	331896-42
Total Items: 1		Total Quantity: 100		

ANSI EDI Purchase Order

ST*850*540001■
BEG*00*SA*4768*65*20120930■
N1*SO*XYZ Company■
N3*123 Main Street■
N4*Fairview*CA*94168■
PO1*1*100*EA*27.65**VN*331896-42■
CTT*1*100■
SE*8*54001■

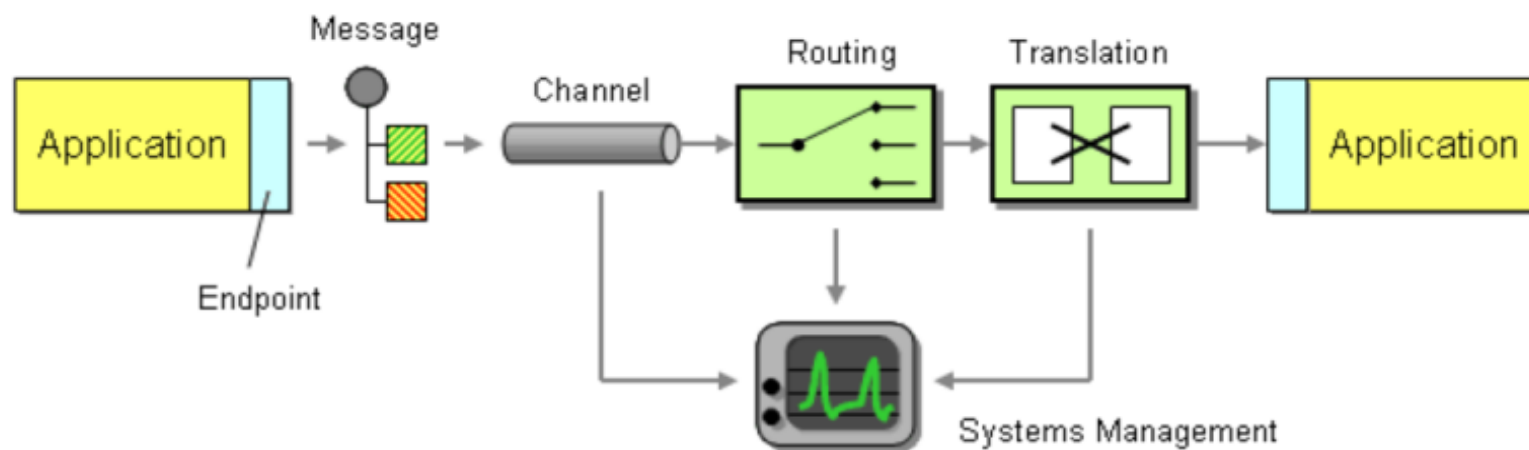


EAI Patterns

Na visão de FOWLER (2011), as aplicações que têm valor raramente funcionam isoladamente. Ou seja, é impossível não falar em interações entre soluções corporativas atualmente.

Ou seja, é impossível não falar em interações entre soluções corporativas atualmente.

Exemplo EAI Patterns



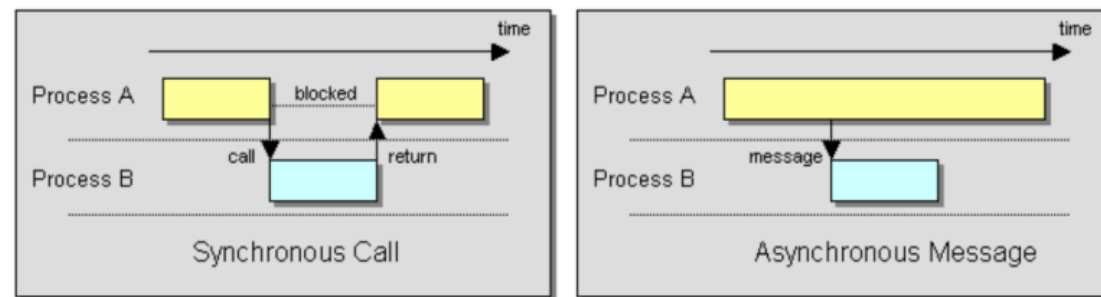
Fonte: FOWLER (2011).

Semântica de integrações

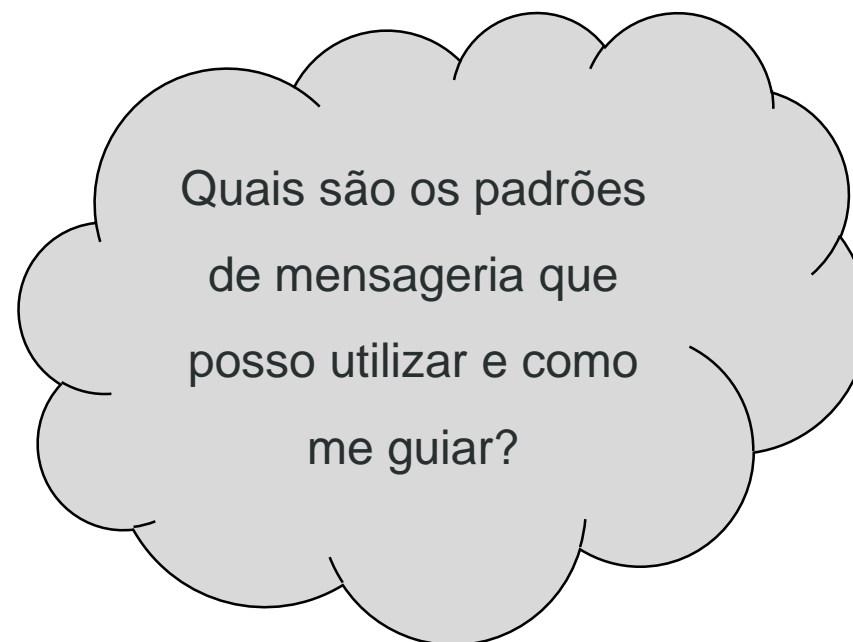
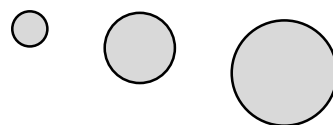


As integrações entre aplicações acontecem de forma síncrona e/ou assíncrona.

A maioria das aplicações corporativas atualmente exigem padrões de integração assíncrono para aumentar a qualidade da experiência do usuário e enfrentar desafios de otimização.

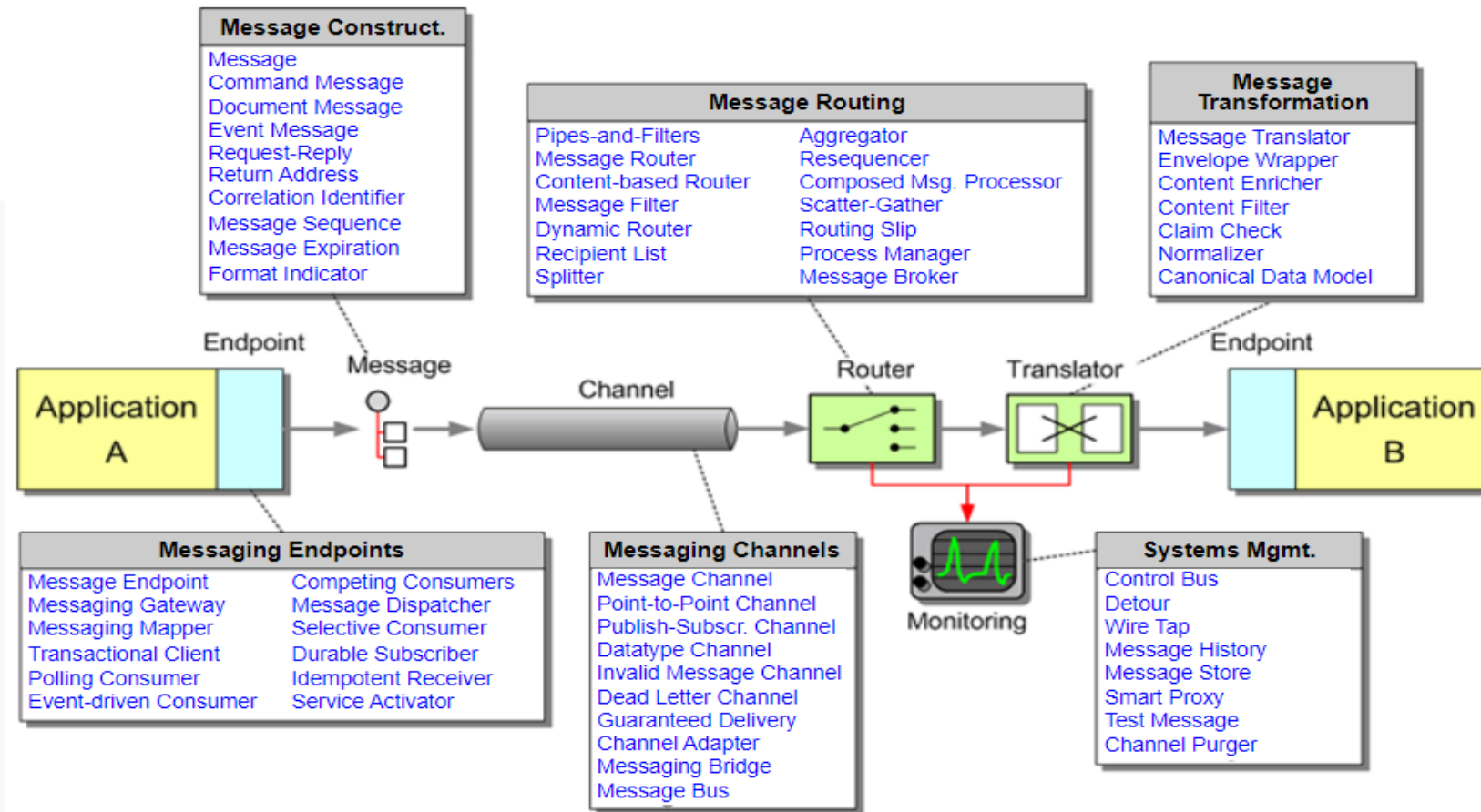


Fonte: FOWLER (2011)



Quais são os padrões
de mensageria que
posso utilizar e como
me guiar?

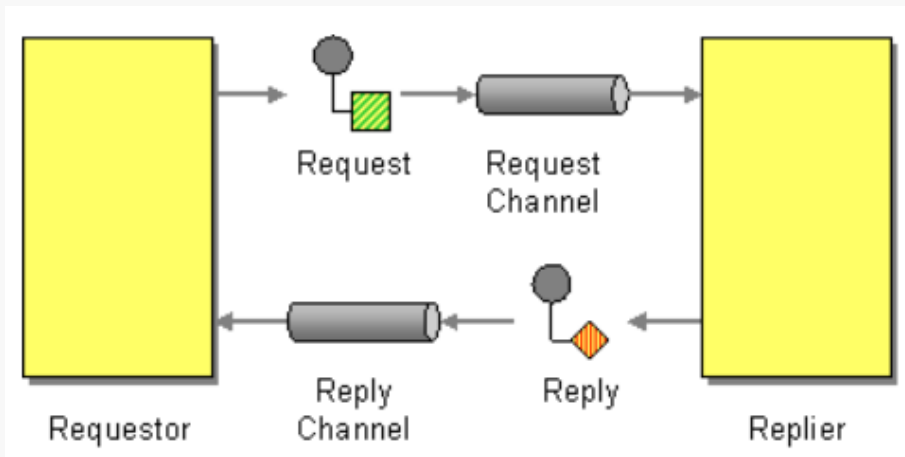
Padrões de mensageria



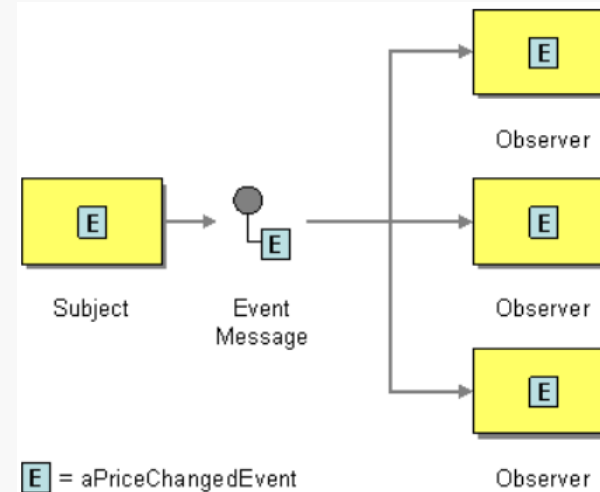
Fonte: FOWLER (2011)

Exemplos de construção de mensagens

Requisição/Resposta

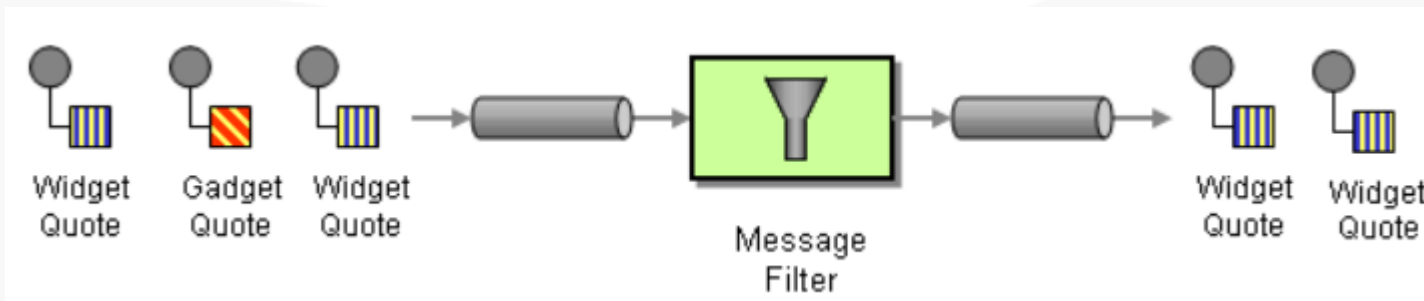


Baseado em eventos

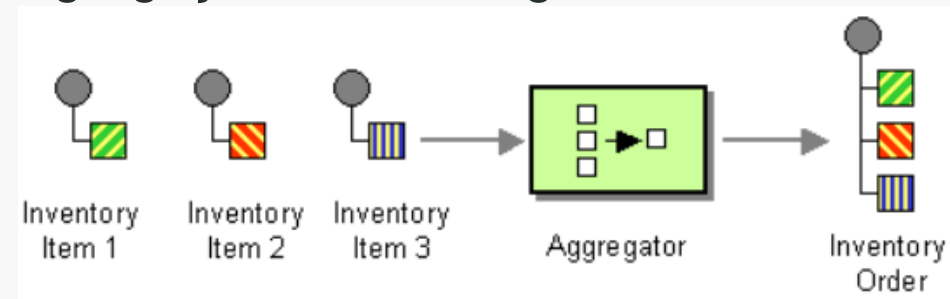


Exemplos de roteamento de mensagens

Filtro de mensagens



Agregação de mensagens



Fonte: FOWLER (2011)

Conclusão



- ✓ Conhecer os padrões de integração é fundamental para a implementação de integrações baseadas em serviços, utilizando o padrão correto.

Próxima aula

01.

Enterprise Service BUS (ESB).

IGTi

Arquiteturas de Integração

AULA 2.2. ENTERPRISE SERVICE BUS

PROF. DIOVANI LUIZ MERLO

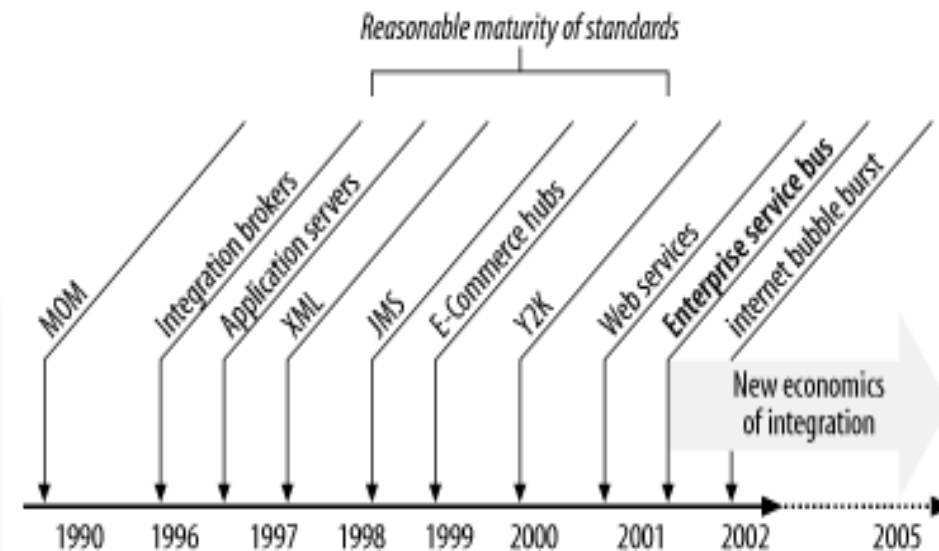
Nesta aula



- ☐ Introdução a ESB.
- ☐ Benefícios para a arquitetura de um ESB.
- ☐ Funcionalidades de um ESB.

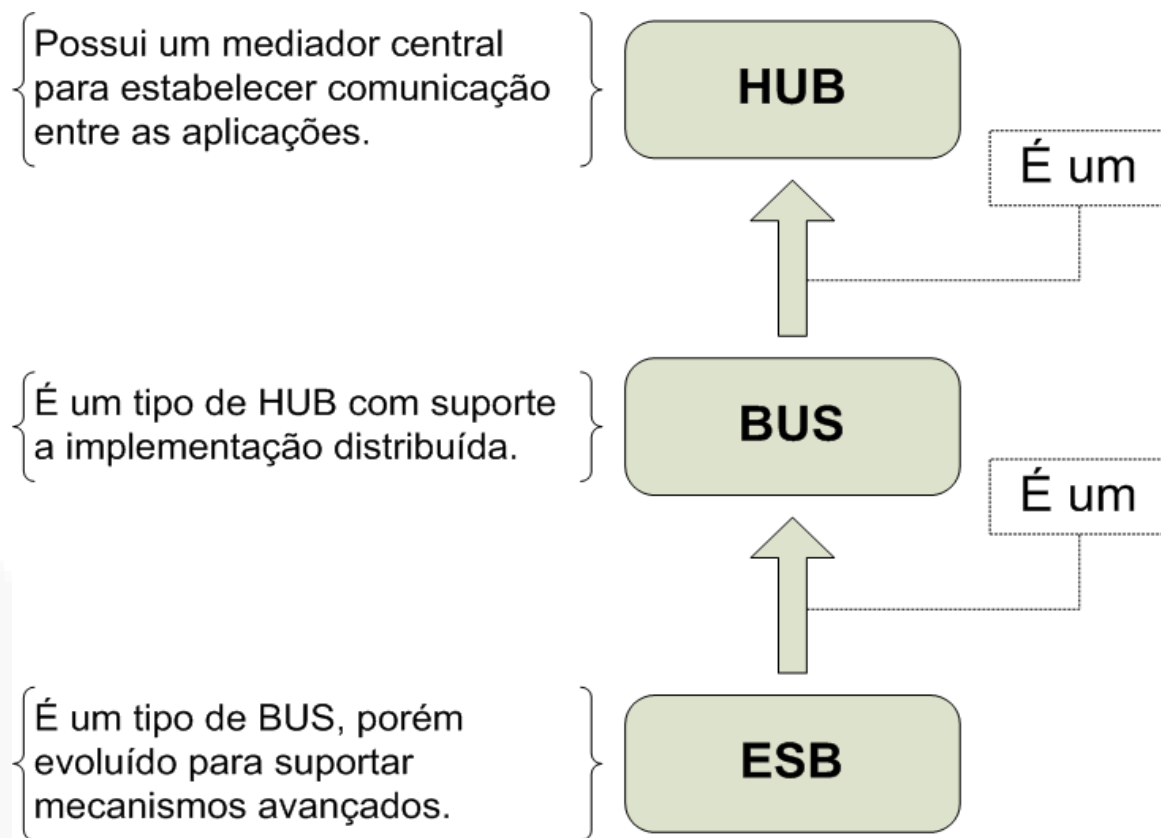
Conceito de um ESB

O conceito de ESB surgiu na indústria de TI por volta 2002. Foi derivado de conceitos como Middleware de mensagens, roteamento e transformação de mensagens. Posteriormente o conceito foi evoluído pelos fornecedores TIBCO, IBM e Oracle.



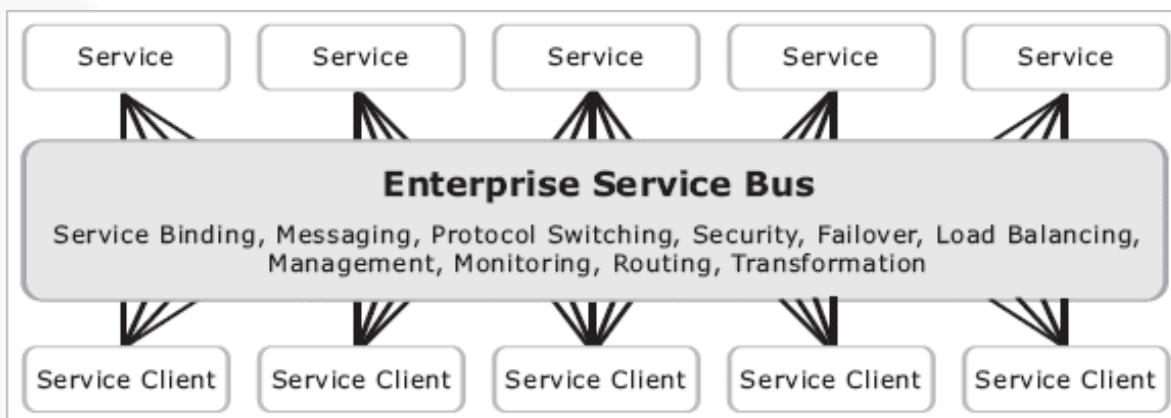
Fonte: CHAPPELL (2004)

Evolução de um ESB

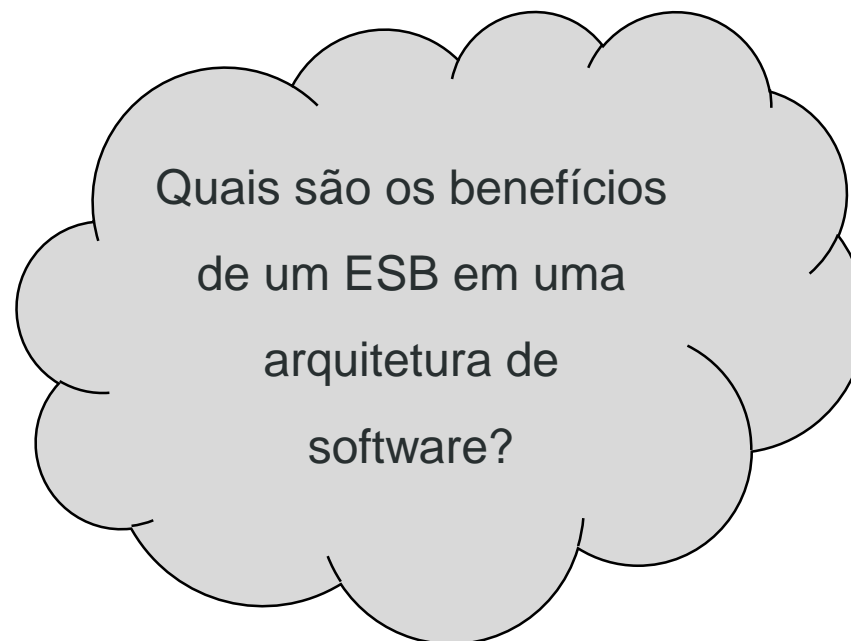
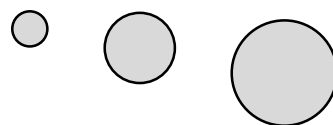


Conceito de um ESB

Em resumo, representa um ambiente desenhado para prover um sofisticado mecanismo de interconectividade entre os serviços. Implementa uma camada intermediária em SOA, a qual prove soluções para problemas como confiabilidade, escalabilidade e heterogeneidade de comunicação entre soluções



Fonte: oracle.com



Quais são os benefícios
de um ESB em uma
arquitetura de
software?

Benefícios de um ESB



1

Promove o reúso de serviços.

Segurança e Monitoramento dos serviços (SLA) em reúso.

Baixo acoplamento entre aplicações.

Padronização de API para integrações.

2

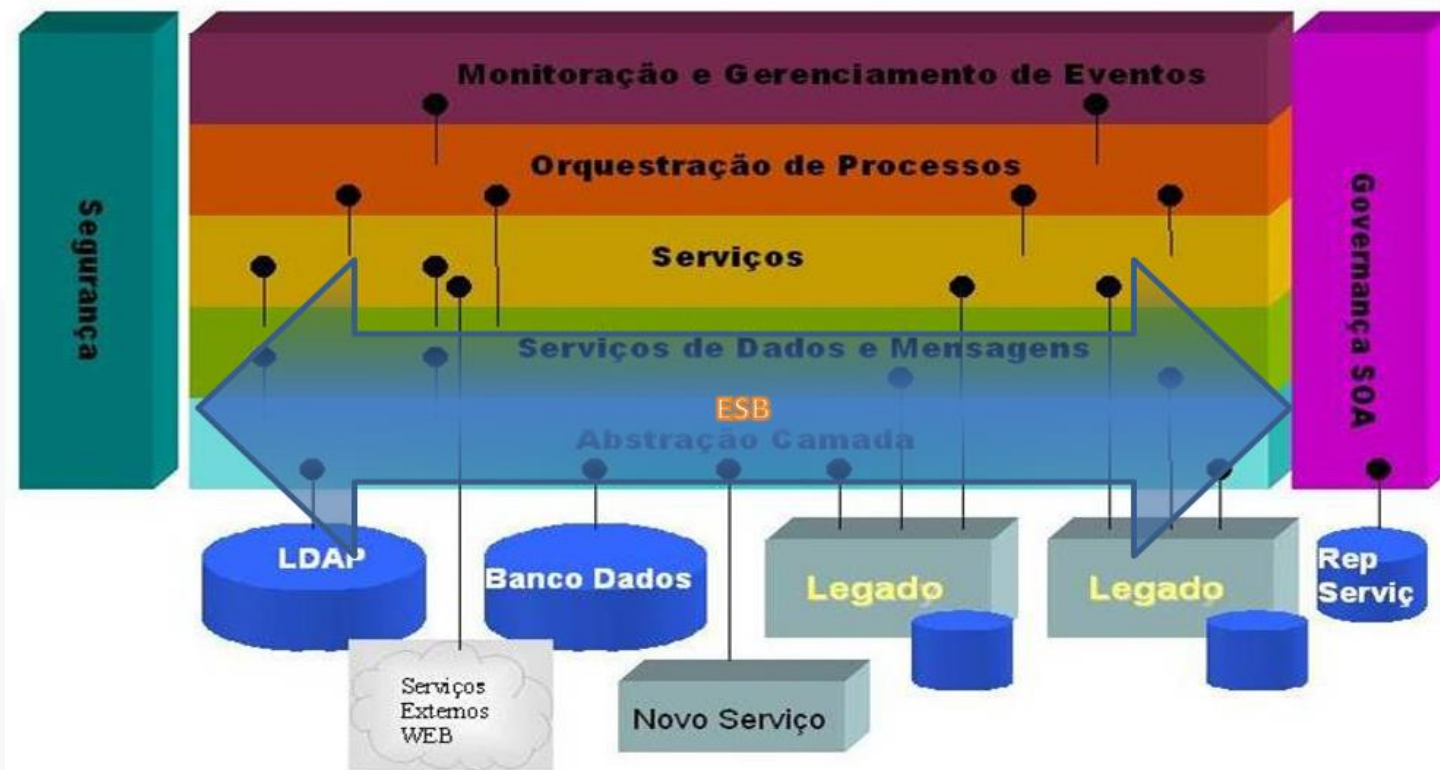
3

Boa prática de EAI.

Simplificação das integrações entre diferentes plataformas.



ESB e SOA



Fonte: CAVALCANTI (2007).

Responsabilidade de um ESB



A diagram on the left side of the slide shows a central blue circle connected by three teal lines to three teal-colored, rounded rectangular shapes. These shapes are arranged vertically and point towards the right, where they align with three horizontal teal bars containing text. The top and bottom teal shapes have a small dark blue circle at their base, while the middle one is a solid dark blue shape.

Integração de serviços.

Transformação de mensagens.

Eliminar contato entre cliente e provedor do serviço.

Adaptadores para aplicações legadas.

Integração de processos de negócio.

Gerar, rotear e tratar eventos de serviços.

Roteamento e entrega de mensagens.

Prover mecanismos de segurança.

Funcionalidades mínimas para um ESB



Category	Capabilities	Reason
Communication	<ul style="list-style-type: none">▶ Routing▶ Addressing▶ At least one messaging style (request / response, pub/sub)▶ At least one transport protocol that is or can be made widely available	<ul style="list-style-type: none">▶ Provide location transparency and support service substitution
Integration	<ul style="list-style-type: none">▶ Several integration styles or adapters▶ Protocol transformation	<ul style="list-style-type: none">▶ Support integration in heterogeneous environments and support service substitution
Service interaction	<ul style="list-style-type: none">▶ Service interface definition▶ Service messaging model▶ Substitution of service implementation	<ul style="list-style-type: none">▶ Support SOA principles, separating application code from specific service protocols and implementations
Management and autonomic	<ul style="list-style-type: none">▶ Administration capability	<ul style="list-style-type: none">▶ A point of control over service addressing and naming

Fonte: www.ibm.com

Conclusão



- ✓ Um barramento de serviços corporativos (ESB) é uma solução robusta, que permite agilidade no desenvolvimento de arquiteturas de integração entre diversos estilos arquiteturais existentes no mercado.

Próxima aula

01.

Web Service.

IGTi

Arquiteturas de Integração

AULA 2.3. WEB SERVICES

PROF. DIOVANI LUIZ MERLO

Nesta aula



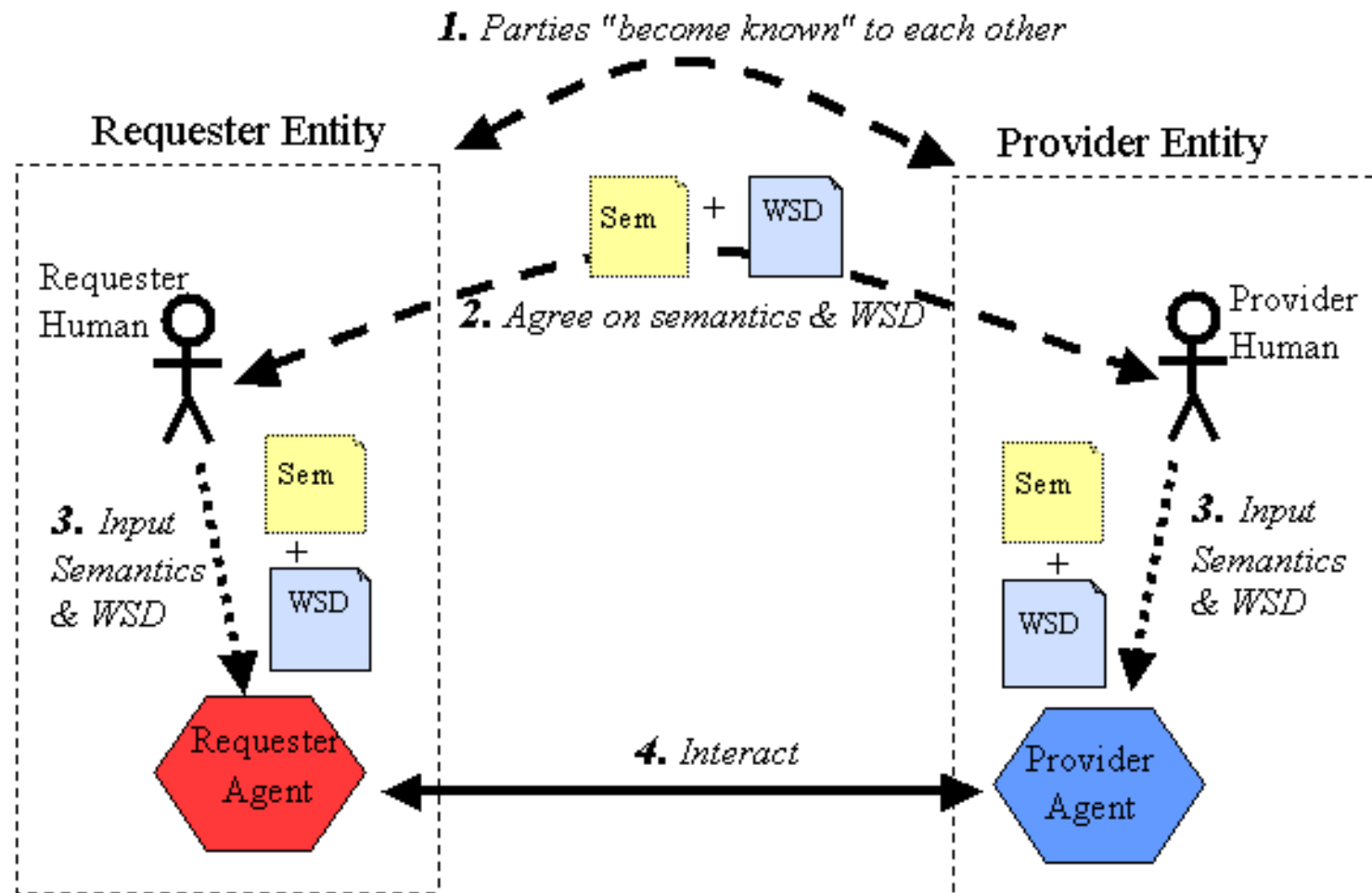
- ☐ Introdução aos conceitos de Web Services.
- ☐ Formatos de mensagem para Web Services.
- ☐ Protocolos utilizados para Web Services.

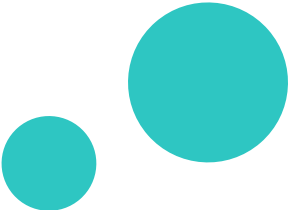
Introdução

O conceito de Web Service é um dos mais importantes atualmente, pois é uma das implementações mais utilizadas para materializar serviços em uma arquitetura orientada a serviços (SOA).

Segundo W3C (2020), Web Service é uma solução de software projetado para suportar uma interação/interoperabilidade entre máquinas sobre uma determinada rede.

Processo Geral uso em Web Service



Two teal circles of different sizes, one larger than the other, positioned above the thought bubble.

Quais são as
vantagens no uso de
Web Services para a
integração de
aplicações?

Vantagens no uso de Web Service



Favorece o uso de aplicações em nuvem (Cloud).

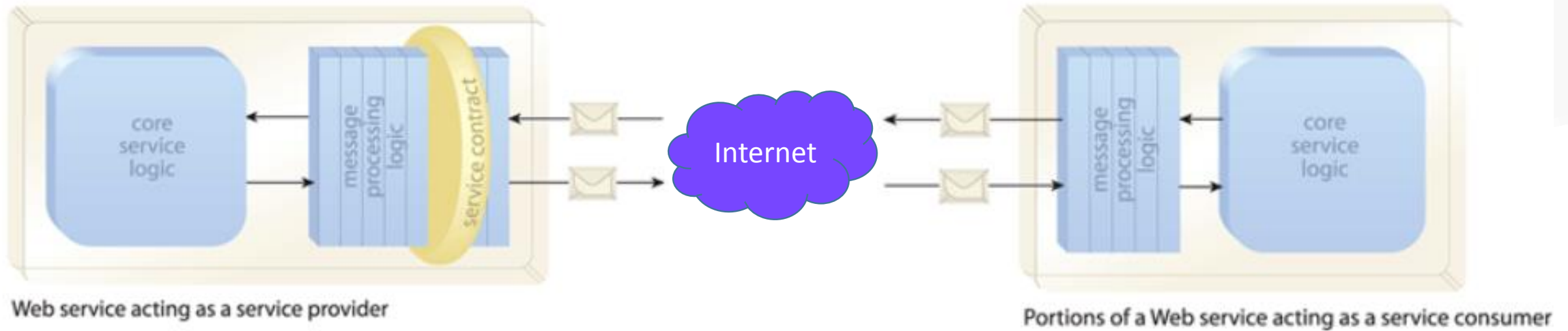
Permite o uso por diferentes plataformas tecnológicas.

Mecanismo de implementação fácil e de baixo custo.

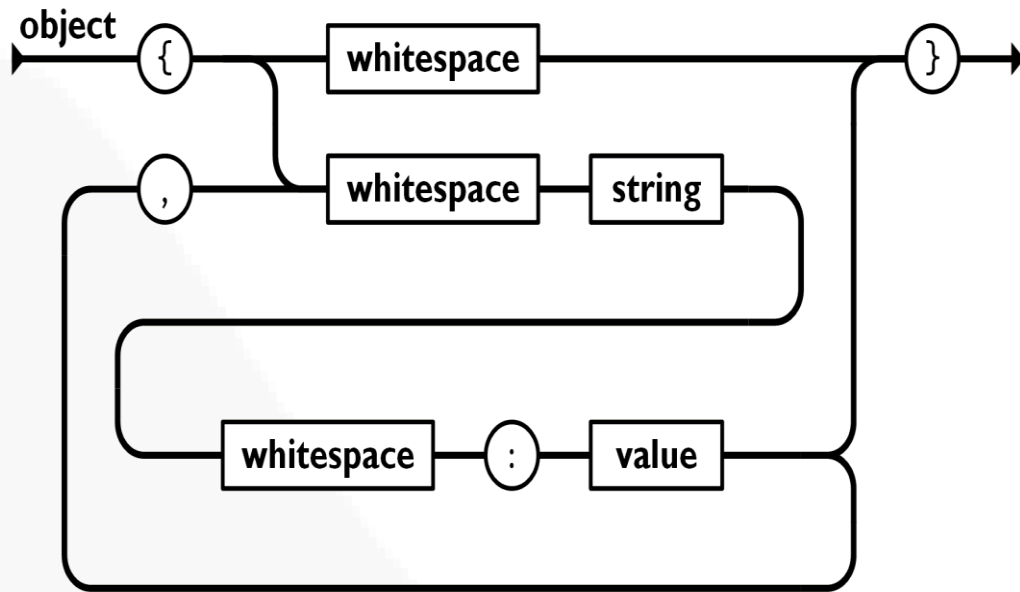
Componentes Web Service



Componentes Web Service



Formato JSON



```
{
  "id": "00000234567894",
  "name": "Jane Doe",
  "birthday": "04/18/1978",
  "gender": "female",
  "type": "user",
  "work": [{
    "employer": {
      "id": "106119876543210",
      "name": "Doe Inc."
    },
    "start_date": "2007 - 08"
  },
  {
    "start_date": "2004",
    "end_date": "2007"
  }
  ]
}
```

Formato XML



O formato XML é o mais rico em detalhes sobre sua estrutura, a partir do uso de *tags* para declaração dos atributos/campos, contudo trata-se de um formato mais pesado para ser trafegado e processado pelos serviços em uma implementação de Web

```
<customer-details id="AcPharm39156">
  <name>Acme Pharmaceuticals Co.</name>
  <address country="US">
    <street>7301 Smokey Boulevard</street>
    <city>Smallville</city>
    <state>Indiana</state>
    <postal>94571</postal>
  </address>
</customer-details>
```

Fonte: W3C (2020)

REST

REST é uma implementação de serviços sem necessidade de protocolos adicionais ao HTTP. É um estilo arquitetural.

RESTFul é apenas uma nomenclatura para definir a capacidade de aplicar conceitos estabelecidos no estilo arquitetural REST.



RESTFul

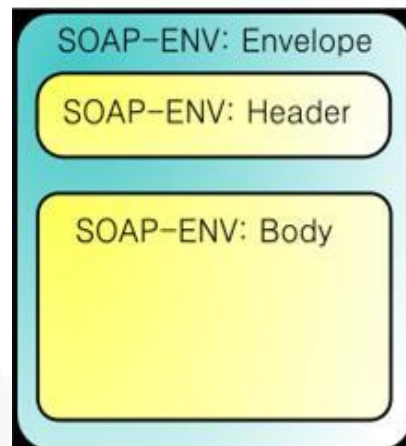
REST

SOAP



SOAP é um protocolo para troca de mensagens estruturadas em um ambiente distribuído, baseado na linguagem XML para seu formato.

Estrutura Lógica



Exemplo

```
<?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>
```

Fonte: W3C (2020)

WSDL



Outro conceito muito importante para o entendimento das soluções que são suportadas por Web Services é o *Web Service Definition Language* (WSDL), uma linguagem para definição de Web Services baseado em XML.

WSDL

Um arquivo WSDL contém os seguintes objetos:

- Type: são os tipos de dados.
- Message: parâmetros de entrada/saída.
- PortType: conjunto de operações.
- Binding: mapeia as operações para um protocolo específico.

WSDL



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="urn:server.hello" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:server.hello">
  <types>
    <xsd:schema targetNamespace="urn:server.hello">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    </xsd:schema>
  </types>
  <message name="helloRequest">
    <part name="name" type="xsd:string" />
  </message>
  <message name="helloResponse">
    <part name="return" type="xsd:string" />
  </message>
  <portType name="server.helloPortType">
    <operation name="hello">
      <documentation>Retorna o nome</documentation>
      <input message="tns:helloRequest" />
      <output message="tns:helloResponse" />
    </operation>
  </portType>
  <binding name="server.helloBinding" type="tns:server.helloPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="hello">
      <soap:operation soapAction="urn:server.hello#hello" style="rpc" />
    </operation>
    <input>
      <soap:body use="encoded" namespace="urn:server.hello" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:server.hello" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </binding>
  <service name="server.hello">
    <port name="server.helloPort" binding="tns:server.helloBinding">
      <soap:address location="http://localhost/imasters2/nuSOAP/server2.php" />
    </port>
  </service>
</definitions>
```

Tipos de Dados

Parâmetros

Operações

Protocolo

Conclusão



- ✓ Atualmente, com o advento de soluções em nuvem e também pelo uso em grande maioria da própria internet, o uso de Web Service tornou-se uma das opções mais viáveis para uma arquitetura de integração de aplicações de tecnologia da informação.

Próxima aula

01.

Introdução aos princípios de
design de serviços.

IGTi