Programação Para Internet

Aula: Serviços na Internet.

Prof. Anderson Augusto Bosing

Acesso a Sala de Aula

Google Classroom

Formas de Avaliação da Disciplina

- Trabalhos e Demais Atividades(≅ 3.0)
- Avaliação Bimestral(≅ 7.0)

Meios de Comunicação

Google Classroom

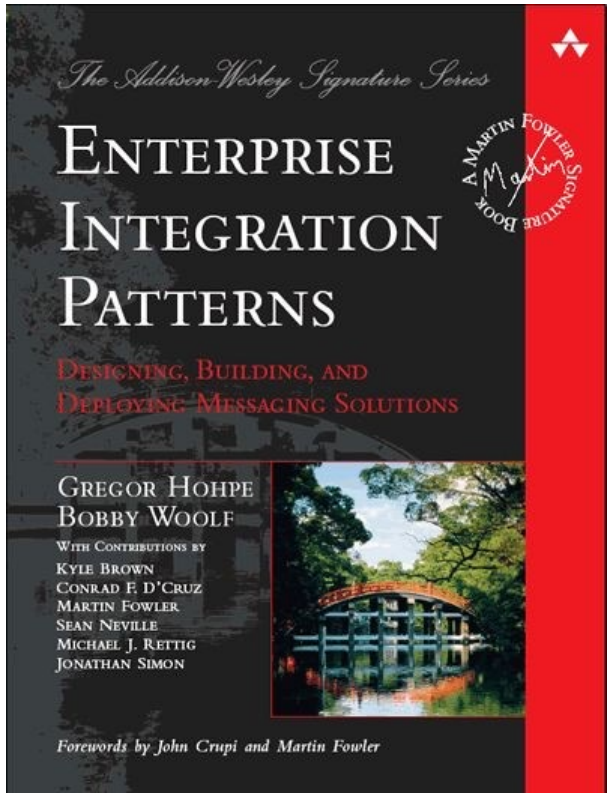
Introdução – Web Services

O que é Web Service?

O que é?
Para que serve?
Resolve que problema?
Qual é o contexto?
Qual é aplicabilidade?

O que é Web Service?

O livro “Enterprise Integration Patterns” classifica os web services como um estratégia de integração de soluções, no qual um sistema necessita acessar dados ou executar operações de outro sistema.



O que é Web Service?

- **Solução** utilizada na integração de sistemas e na comunicação entre independentes aplicações.

- **Tecnologia** que possibilita a integração entre aplicações de forma que elas possam interagir entre si independentemente de plataforma, linguagem, paradigma de programação e fornecedor de tecnologia.

O que é Web Service?

- São componentes que permitem às aplicações enviar e receber informações baseados em um formato pré estabelecido.

- Um serviço web é uma aplicação disponibilizada via (HTTP) intranet ou internet que através de uma URL pode ser acessado por aplicativos clientes usando protocolos acordados.

Para que serve Web Service?

Único e exclusivo objeto é:

Interoperabilidade entre sistemas independentemente de plataforma de desenvolvimento, execução, provedor de tecnologia, linguagem e paradigma de desenvolvimento.

Para que serve Web Service?

Interoperabilidade é a capacidade de um sistema informatizado ou não de se comunicar de forma transparente ou o mais próximo disso com outro sistema semelhante ou não.

Para que serve Web Service?

Interoperabilidade é a capacidade de um sistema informatizado ou não de se comunicar de forma transparente ou o mais próximo disso com outro sistema semelhante ou não.

Quando usar Web Services?

Aplicações corporativas abrangem uma série de cenários simples até os mais complexos que apresentam requisitos de interoperabilidade entre soluções corporativas dentro de um ambiente empresarial heterogêneo.

Cenários de Integração

- Grandes, médias e pequenas empresas possuem investimento significativos e crescentes em sistemas pertencentes a diferentes plataformas ao longo de sua história.

- Web services são a solução para fazer a integração entre o legado e as novas aplicações.

Cenários de Integração

- Na atualidade temos empresas adquirindo outras empresas e cada uma delas apresentando soluções próprias em diferentes plataformas.

- Web services são a solução para fazer a integração entre estas aplicações até então desconhecidas.

Cenários de Integração

- Muitas empresas têm expandido suas redes de negócios no estabelecimento de "parcerias" com outras empresas "sócios" de outros setores que unidas formam uma nova vertente de negócio.

- Web services são a solução para fazer a integração entre estas aplicações de cada empresa, garantindo eficiência no negócio e melhor experiência para o usuário final.

Cenários de Integração

- Muitas empresas têm a necessidade de expor suas soluções para ser acessadas remotamente em diferentes plataformas de execução(smartphones, tablet, TV etc) que são incompatíveis com a tecnologia adotadas.

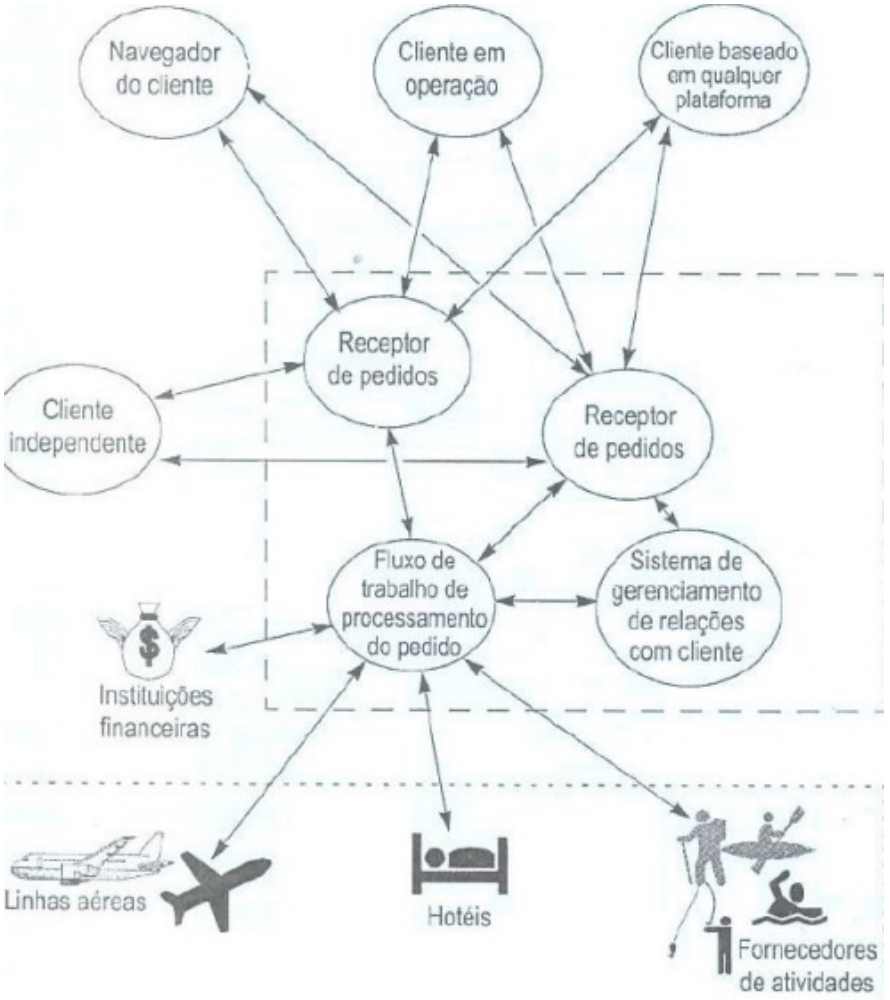
- Web services são a solução utilizada para integrar aplicativos incompatíveis.

Cenários Real

- **Agência de Turismo**

- Fornece catálogo de turismo.
- Construir passeios customizados de acordo com o perfil dos clientes.
- Rastrear status de pedidos.
- Reservar passeios.
- Todas as operações necessitam Interagir com sistemas de outros sócios – linhas aéreas, hotéis, instrutores de passeios, instituições financeiras etc.

Cenários Real



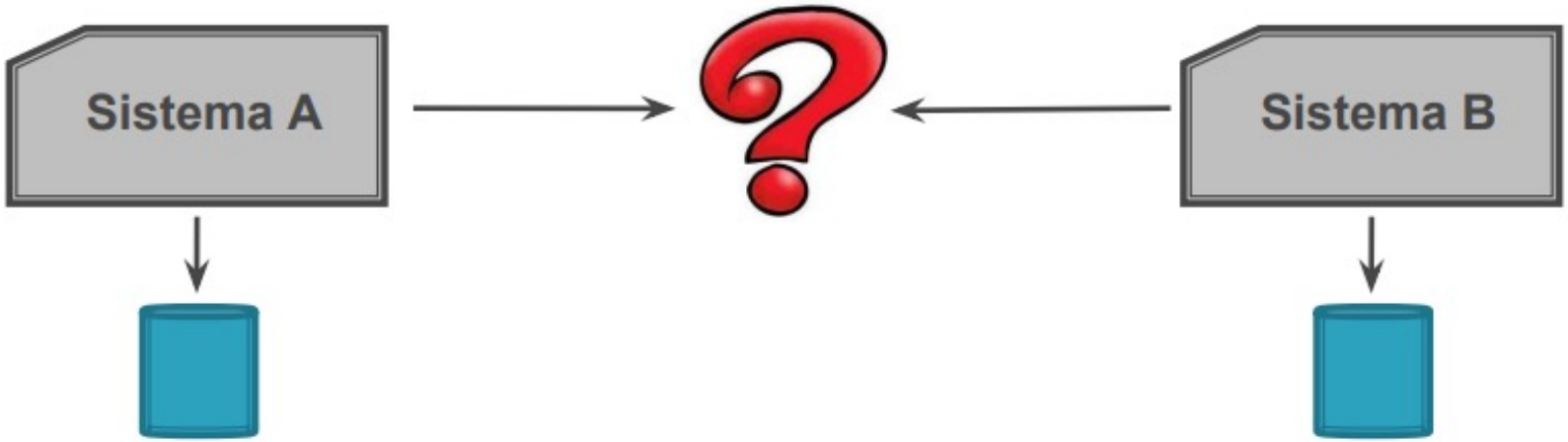
Programação Para Internet

Aula: Arquitetura de Solução – Web Service.

Prof. Anderson Augusto Bosing

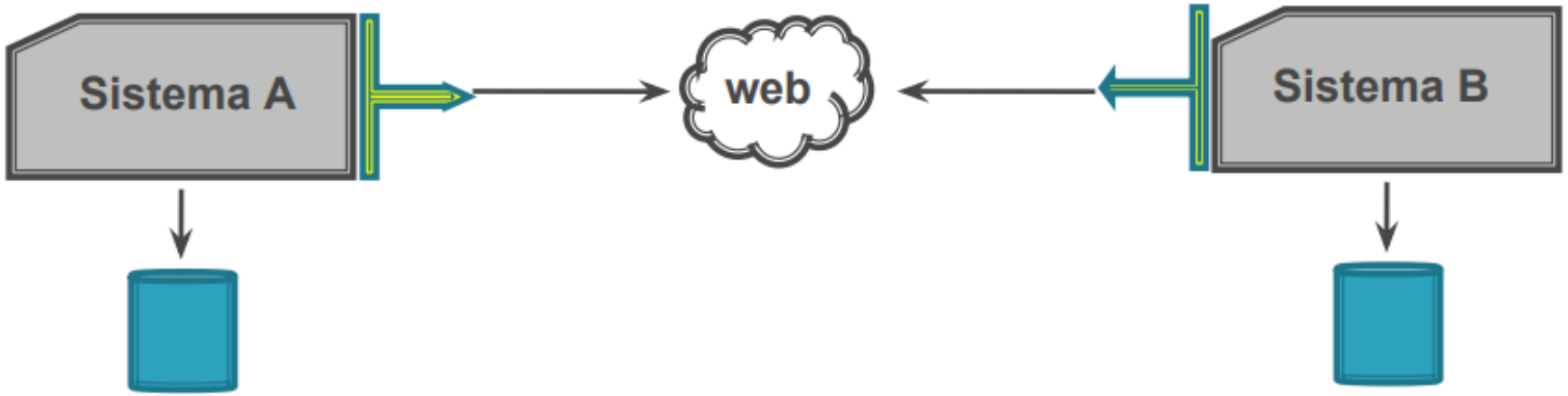
Arquitetura - Web Service

- Surge a necessidade de integrar 2 ou mais soluções diferentes dentro de uma mesma corporação:



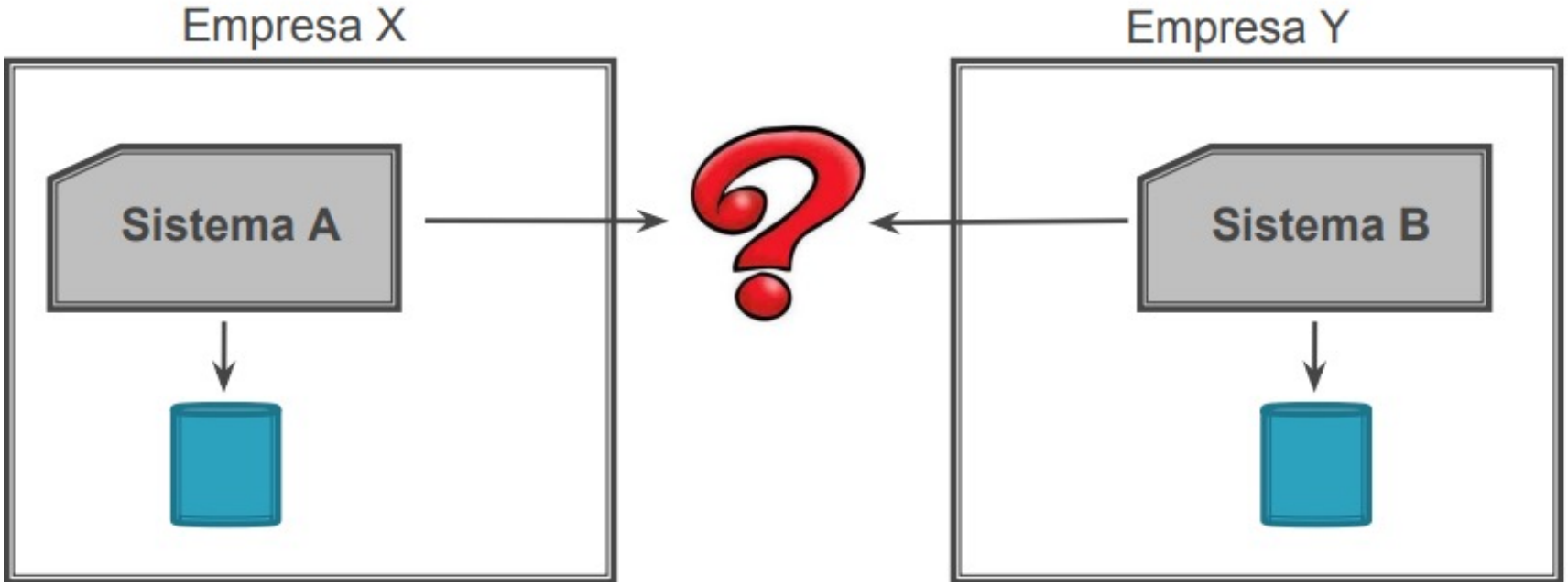
Arquitetura - Web Service

- É acrescentado em cada solução uma "camada de comunicação" que expõe as operações existentes como serviço:



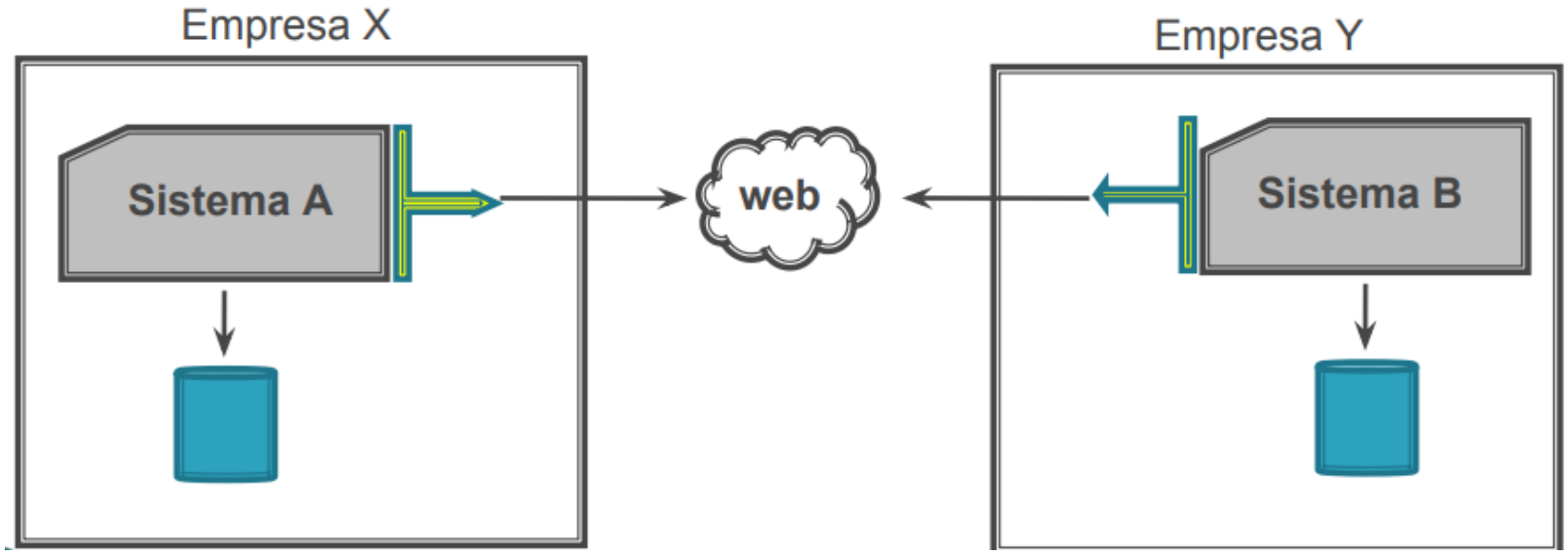
Arquitetura - Web Service

- Surge a necessidade de integrar duas ou mais soluções de diferentes corporações:



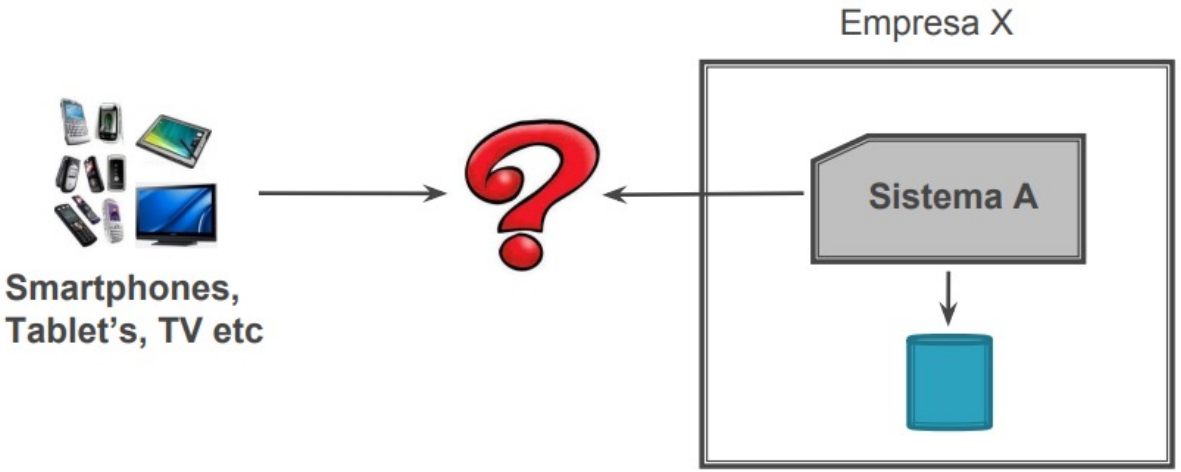
Arquitetura - Web Service

- **É acrescentado nas soluções uma "camada de comunicação" que expõe as operações existentes como serviço:**



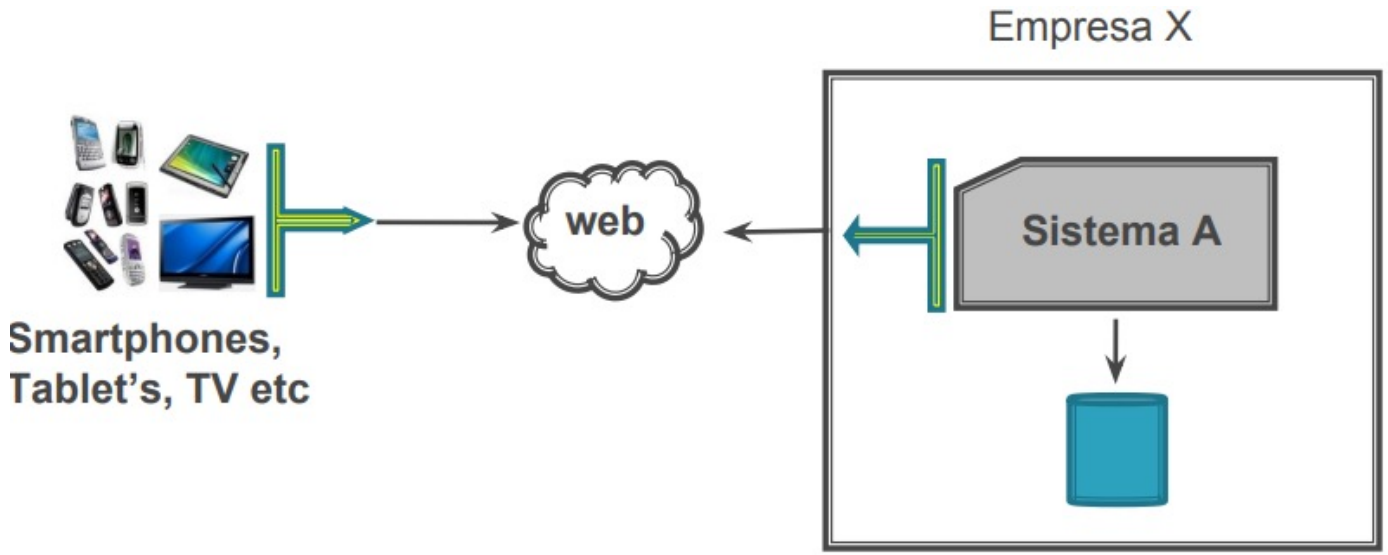
Arquitetura - Web Service

- Surge a necessidade de integrar soluções de dispositivos móveis em uma solução corporativa:



Arquitetura - Web Service

- **É acrescentado na solução da corporação e nas soluções nativas de cada dispositivos uma "camada de comunicação":**



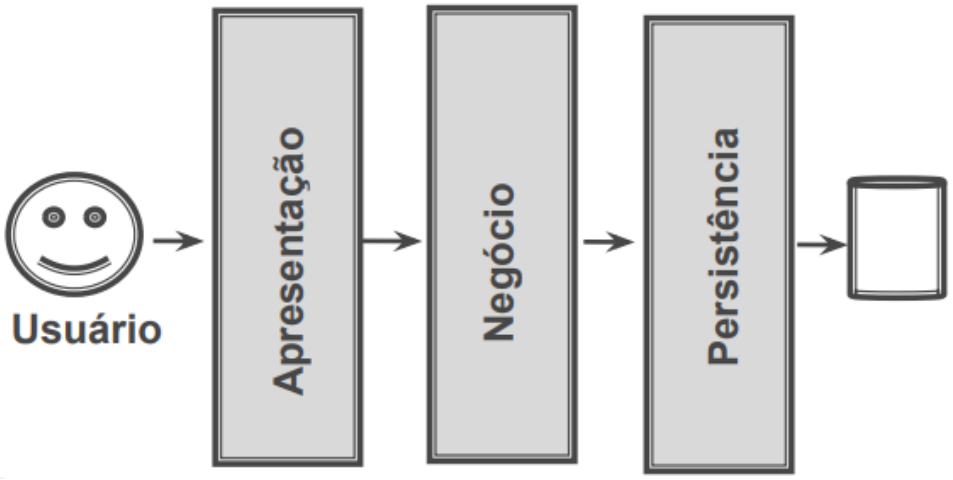
Arquitetura - Web Service

- A aplicação que expõe funcionalidades via web service é chamada de “serviço”.

- A aplicação que invoca as funcionalidade de um serviço é chamada de “consumidor”.

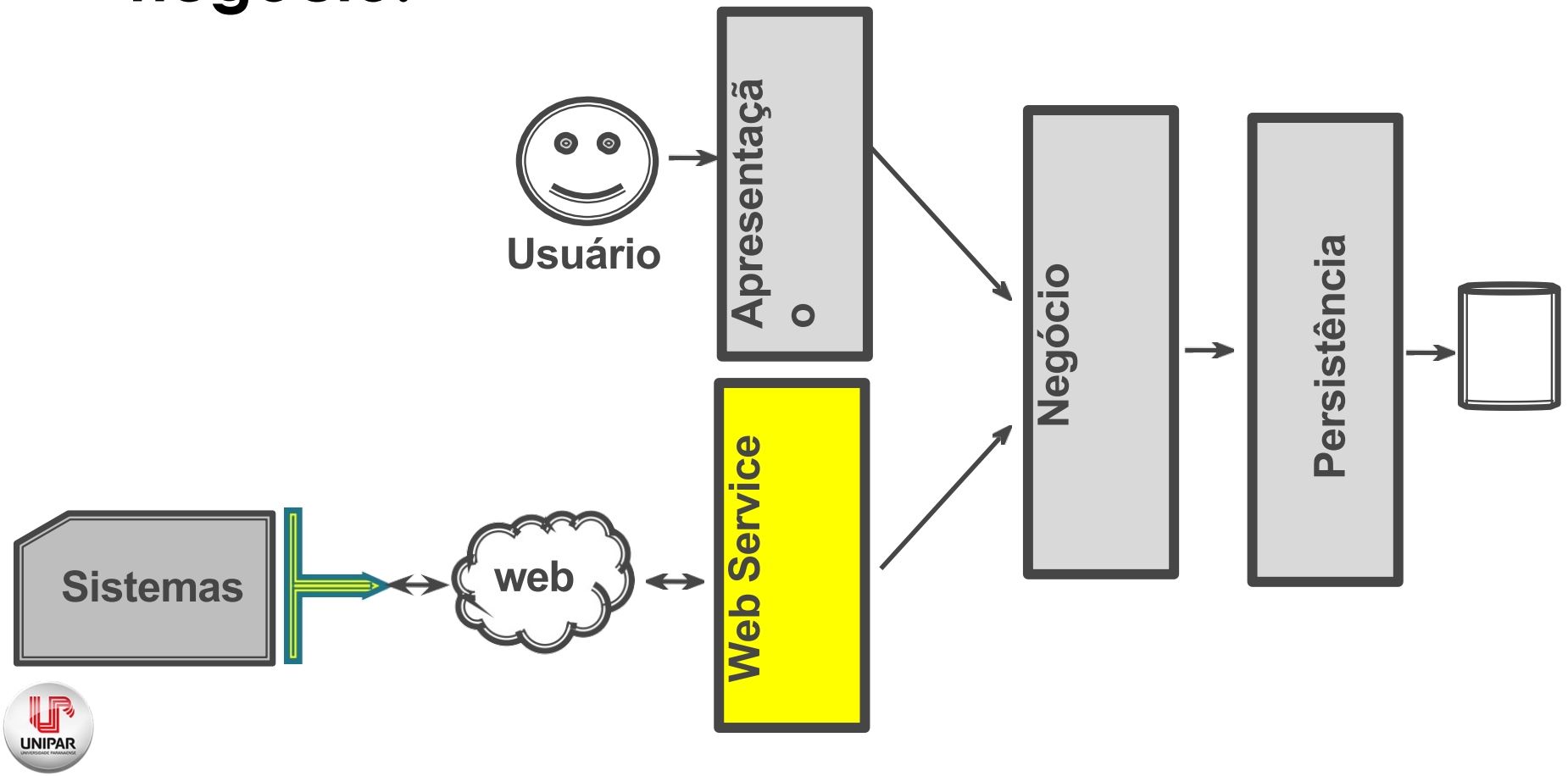
Arquitetura - Web Service

- A solução devidamente projetada esta dívida arquiteturalmente em camadas lógicas que organiza suas responsabilidades.



Arquitetura - Web Service

- Com web services é criado uma camada (responsável por encapsular os detalhes WS) com dependência para a camada de negócio:



Benefícios

- Interoperabilidade em ambientes heterogêneo:
- Como as empresas se desenvolvem ao passar dos tempos, elas adicionam sistemas e soluções requerem diferentes plataformas de diferentes provedores que não se comunicando com as outras.
- Permite que diferentes serviços distribuídos sejam executados em variedade de plataformas de software e de arquitetura.

Benefícios

- **Interoperabilidade em ambientes heterogêneo:**

- **Permite que eles sejam escritos em diferentes linguagens e filosofias de programação, independentemente de fornecedor de tecnologia.**

Benefícios

- Serviços de negócios através da web:
- Uma empresa pode usar web services para alavancar as vantagens em âmbito mundial.
- Ex: Uma empresa de catálogo de produtos pode colocar disponível seu estoque aos seus fornecedores através de web services de modo a conseguir melhor gerenciamento da cadeia de suprimentos.

Benefícios

- Integração com sistemas existentes:

- Uma empresa pode usar web services para alavancar as vantagens em âmbito mundial.

- A maioria das empresas possuem uma quantidade enorme de informações armazenados no sistemas de informações existentes e o custo para substituir isso é tão absurdo que não exista possibilidades de descartar sistemas legados.

Benefícios

- Liberdade de opção:

- O uso de padrões, convenções e diretrizes no desenvolvimento de web services abriram um grande mercado de ferramentas, produtos e tecnologias.

- Isso oferece aos desenvolvedores uma ampla variedade de opções (free, open source e comercial) na seleção de um produto para a criação de web services.

Benefícios

- Suportam qualquer tipo de aplicativos clientes:

- Suportam qualquer tipo de dispositivo como aplicativo cliente de um web service devido independência de plataforma, tecnologias e provedores.

Programação Para Internet

Aula: Arquitetura de Solução – Web Service - SOAP.

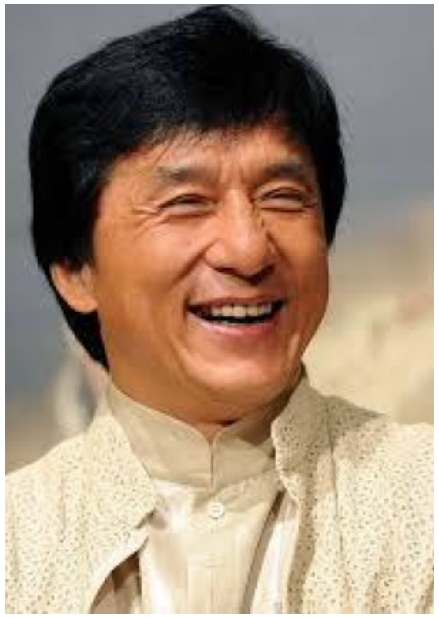
Prof. Anderson Augusto Bosing

O que é SOAP?

O que é?
Para que serve?
O que eu faço?

Java Web Service - SOAP?

Digamos que tem um amigo chinês, solteiro.

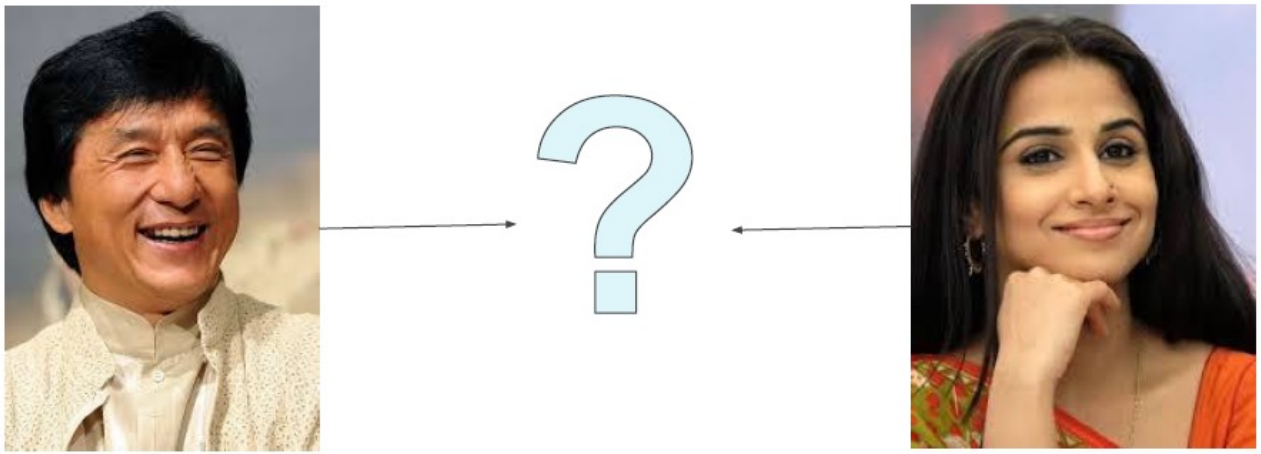


Java Web Service - SOAP?

E você se lembrou que tem uma amiga indiana, solteira.

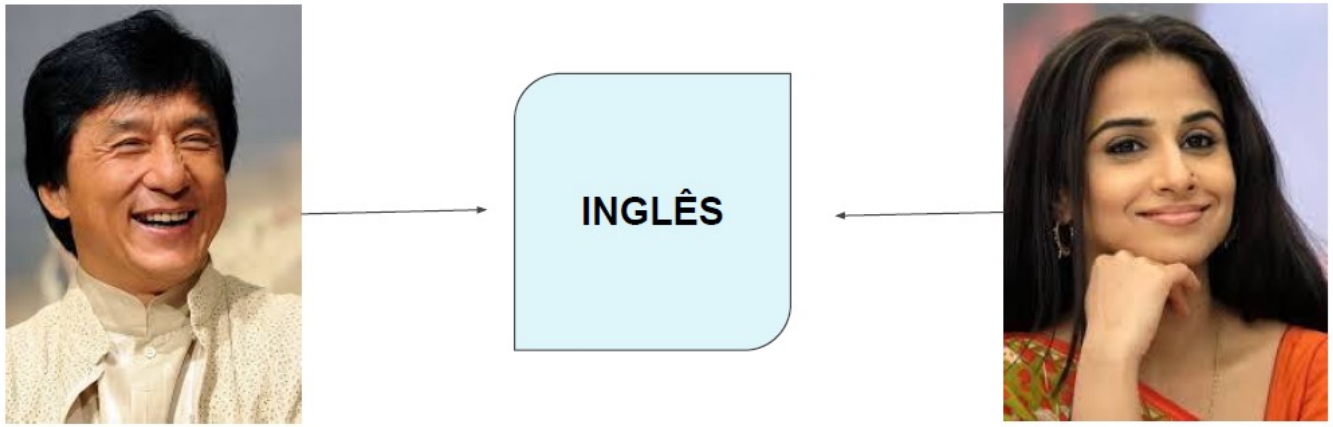
Java Web Service - SOAP?

Como você faria para que os dois pudessem trocar informações?



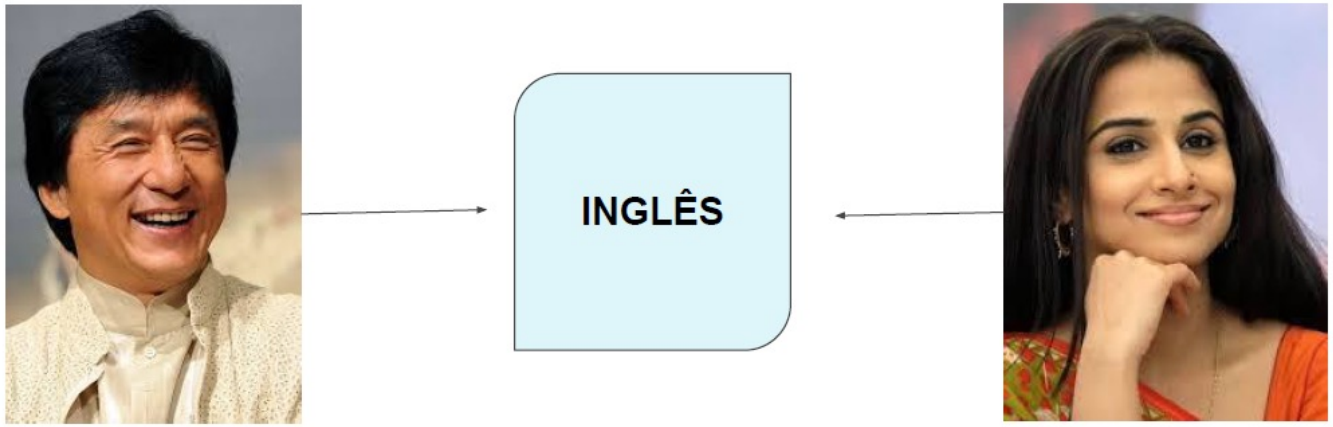
Java Web Service - SOAP?

Seria necessário um acordo de "linguagem" em comum entre os 2.



Java Web Service - SOAP?

O inglês seria o protocolo de acordo entre eles para troca de informações.



Java Web Service - SOAP?

SOAP é o acrônimo de *Simple Object Access Protocol* - Protocolo Simples de Acesso a Objetos.

É um protocolo utilizado troca de informações estruturadas descentralizada e distribuída entre diferentes plataformas de desenvolvimento e execução.

Java Web Service - SOAP?

SOAP é uma especificação para a troca de informação entre sistemas, ou seja, uma especificação de formato de dados para envio de estruturas entre serviços, estabelecendo um padrão de mercado oficial padrão e mundialmente aceito para permitir a interoperabilidade entre eles.

Os órgãos responsáveis pelas especificações são:

- W3C (World Wide Web Consortium)
- WS-I (Web Services Interopeability Industry Consortium)

Java Web Service - SOAP?

Para alcançar 100% de interoperabilidade, os órgãos responsáveis especificaram os seguintes recursos:

1. Linguagens de comunicação.
2. Formato de intercâmbio de mensagens.
3. Protocolo de tráfego.
4. Descrição de serviços.

Java Web Service - SOAP?

1. Linguagens de Comunicação:

Comunicação entre diferentes plataformas obrigam uma terminologia em comum. Uma linguagem através do qual os requisitantes e o serviço possam se comunicar.

Uma linguagem independente de plataforma que fosse capaz de trocar informações neutras.

Java Web Service - SOAP?

1. Linguagens de Comunicação:

XML - eXtensible Markup Language é uma linguagem de marcação para o transporte de dados organizados hierarquicamente.

Independente de plataforma e de hardware uma vez que é representado como simples arquivo texto.

Java Web Service - SOAP?

1. Linguagens de Comunicação:

Classificada como extensível porque permite definir os elementos de marcação.

Recomendada pela W3C.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:

Para uma comunicação efetiva as partes devem ser capazes de trocar mensagens de acordo com um formato padrão.

Diante deste formato, a partes estranhas e incompatíveis podem então se comunicar eficientemente.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:

SOAP - Simple Object Access Protocol (Protocolo Simples de Acesso a Objetos) é um protocolo usando o paradigma de objetos, criado especialmente para troca de informações estruturadas em uma plataforma descentralizada e distribuída para web services.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:

SOAP é construído usando XML para seu formato de mensagem e tem o objetivo de formar a camada base de uma pilha de protocolos de web services, fornecendo um framework de mensagens básico sob o qual os serviços web podem ser construídos.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:

SOAP consiste basicamente em três partes:
Envelope: Toda mensagem SOAP deve contê-lo. É o elemento raiz do documento XML. O Envelope pode conter declarações de namespaces e também atributos adicionais como o que define o estilo de codificação.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:
Header: É um cabeçalho opcional. Ele carrega informações adicionais. Quando utilizado, o Header deve ser o primeiro elemento do Envelope.

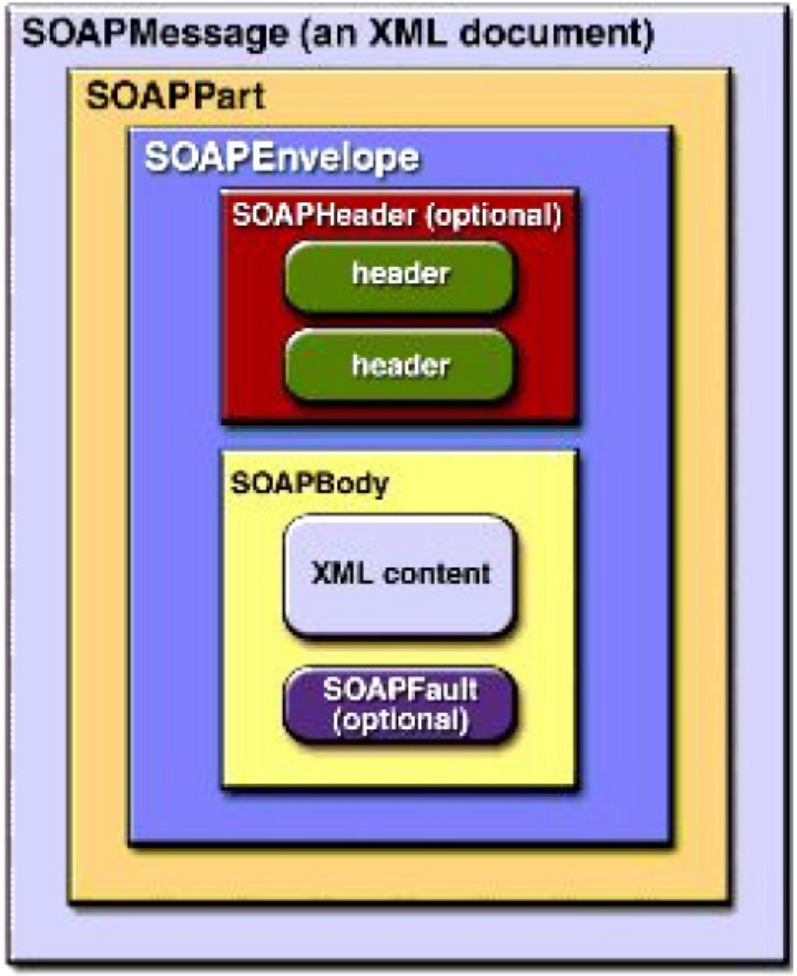
Body: Elemento obrigatório e contém a informação a ser transportada para o seu destino final. O elemento Body pode conter um elemento opcional Fault, usado para carregar mensagens de status e erros retornadas pelos web services.

Java Web Service - SOAP?

2. Formato de Intercâmbio de Mensagens:
Header: É um cabeçalho opcional. Ele carrega informações adicionais. Quando utilizado, o Header deve ser o primeiro elemento do Envelope.

Body: Elemento obrigatório e contém a informação a ser transportada para o seu destino final. O elemento Body pode conter um elemento opcional Fault, usado para carregar mensagens de status e erros retornadas pelos web services.

Java Web Service - SOAP?



Java Web Service - SOAP?

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:c="http://www.acmeOrders.com/OrderService"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <c:OrderMessage>
            <localElement>
                <FirstName>John</FirstName>
                <LastName>Smith</LastName>
                <Street>High Street</Street>
                <City>London</City>
                <ZipCode>W1A1AA</ZipCode>
                <PartNumber>ABC1234</PartNumber>
                <Quantity>1</Quantity>
            </localElement>
        </c:OrderMessage>
    </soap:Body>
</soap:Envelope>
```

Java Web Service - SOAP?

3. Protocolo de tráfego:
SOAP foi especificado como protocolo de serviço, com o objetivo de ser independente do protocolo de comunicação. Assim SOAP pode trafegado dentro de qualquer tipo de protocolo – HTTP, SMTP, FTP etc...

Para web services corporativos, foi selecionado o HTTP como protocolo de tráfego oficial, justamente pelo popularização e facilidades da internet.

Java Web Service - SOAP?

4. Descrição de Serviços:

Além dos formatos de mensagem e linguagem de marcação padrão, deve haver uma formato em que todos os fornecedores de web services possam utilizar para *especificar detalhes de seus serviços*.

Java Web Service - SOAP?

4. Descrição de Serviços:

- Como por exemplo:
- Quais as operações disponíveis?
- Como acessar cada uma delas?
- Quais parâmetros enviar em cada operação?
- O que será retornado caso sucesso na comunicação?
- O que será retornado caso falha da comunicação?

Java Web Service - SOAP?

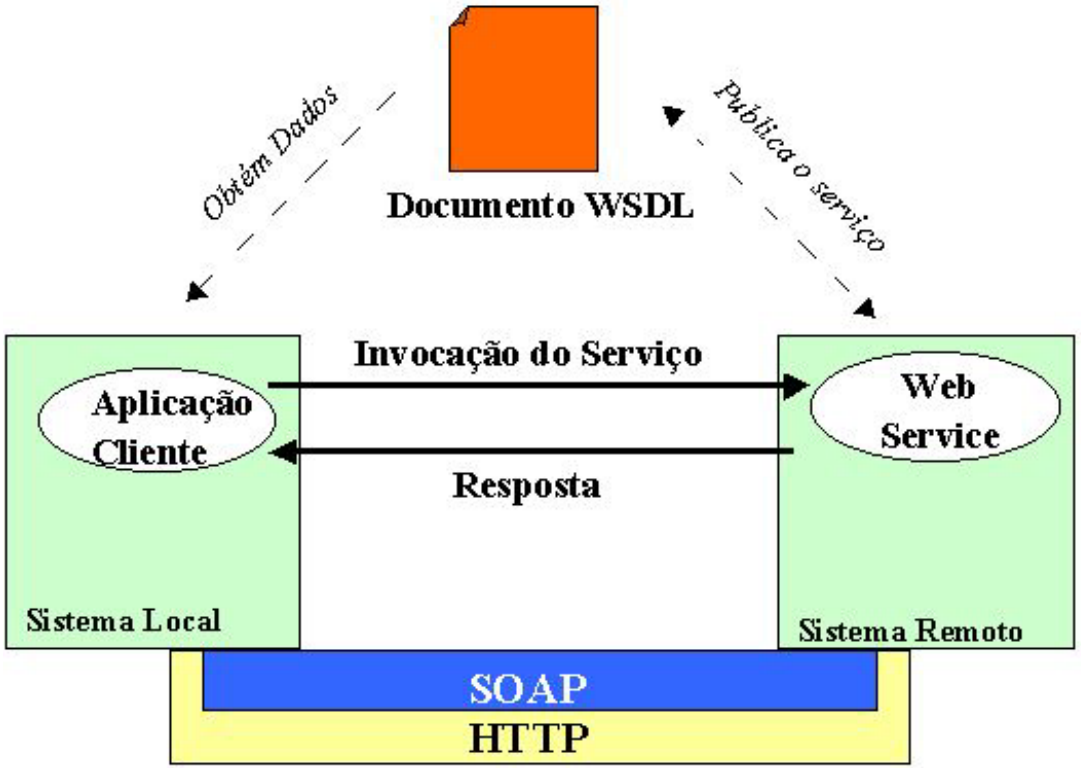
4. Descrição de Serviços:

WSDL - Web Services Description Language é uma linguagem baseada em XML utilizada para descrever Web Services funcionando como um contrato do serviço. Trata-se de um documento escrito em XML que além de descrever o serviço, especifica como acessá-lo e quais as operações ou métodos disponíveis.

Java Web Service - SOAP?

```
- <wsdl:definitions name="AdderService" targetNamespace="http://adder.remote.ipojo.felix.apache.org/">
  - <wsdl:types>
    - <xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
      targetNamespace="http://adder.remote.ipojo.felix.apache.org/">
        <xsd:element name="add" type="tns:add"/>
      - <xsd:complexType name="add">
        - <xsd:sequence>
            <xsd:element name="arg0" type="xsd:int"/>
            <xsd:element name="arg1" type="xsd:int"/>
          </xsd:sequence>
        </xsd:complexType>
        <xsd:element name="addResponse" type="tns:addResponse"/>
      - <xsd:complexType name="addResponse">
        - <xsd:sequence>
            <xsd:element name="return" type="xsd:int"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:schema>
    </wsdl:types>
  - <wsdl:message name="addResponse">
      <wsdl:part element="tns:addResponse" name="parameters"> </wsdl:part>
    </wsdl:message>
  - <wsdl:message name="add">
      <wsdl:part element="tns:add" name="parameters"> </wsdl:part>
    </wsdl:message>
  - <wsdl:portType name="AdderServicePortType">
    - <wsdl:operation name="add">
        <wsdl:input message="tns:add" name="add"> </wsdl:input>
        <wsdl:output message="tns:addResponse" name="addResponse"> </wsdl:output>
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding name="AdderServiceSoapBinding" type="tns:AdderServicePortType">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    - <wsdl:operation name="add">
        <soap:operation soapAction="" style="document"/>
      - <wsdl:input name="add">
          <soap:body use="literal"/>
        </wsdl:input>
      - <wsdl:output name="addResponse">
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
```

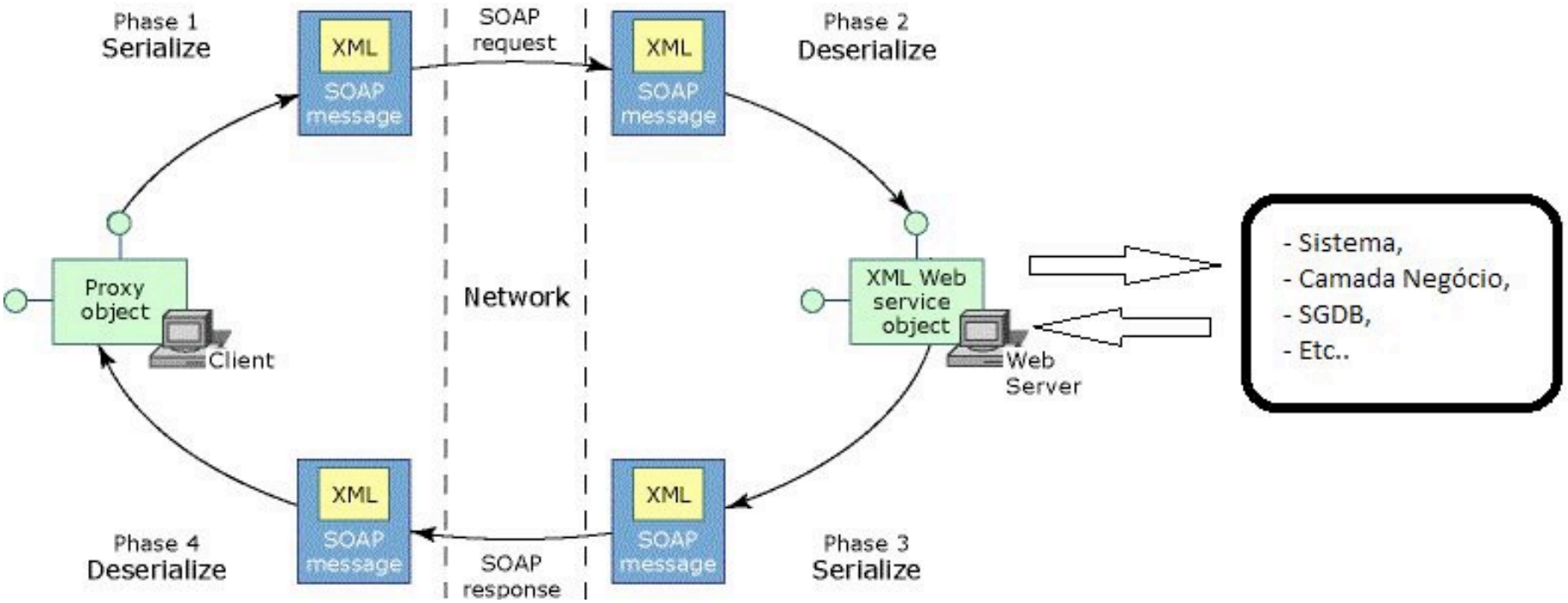
Java Web Service - SOAP?



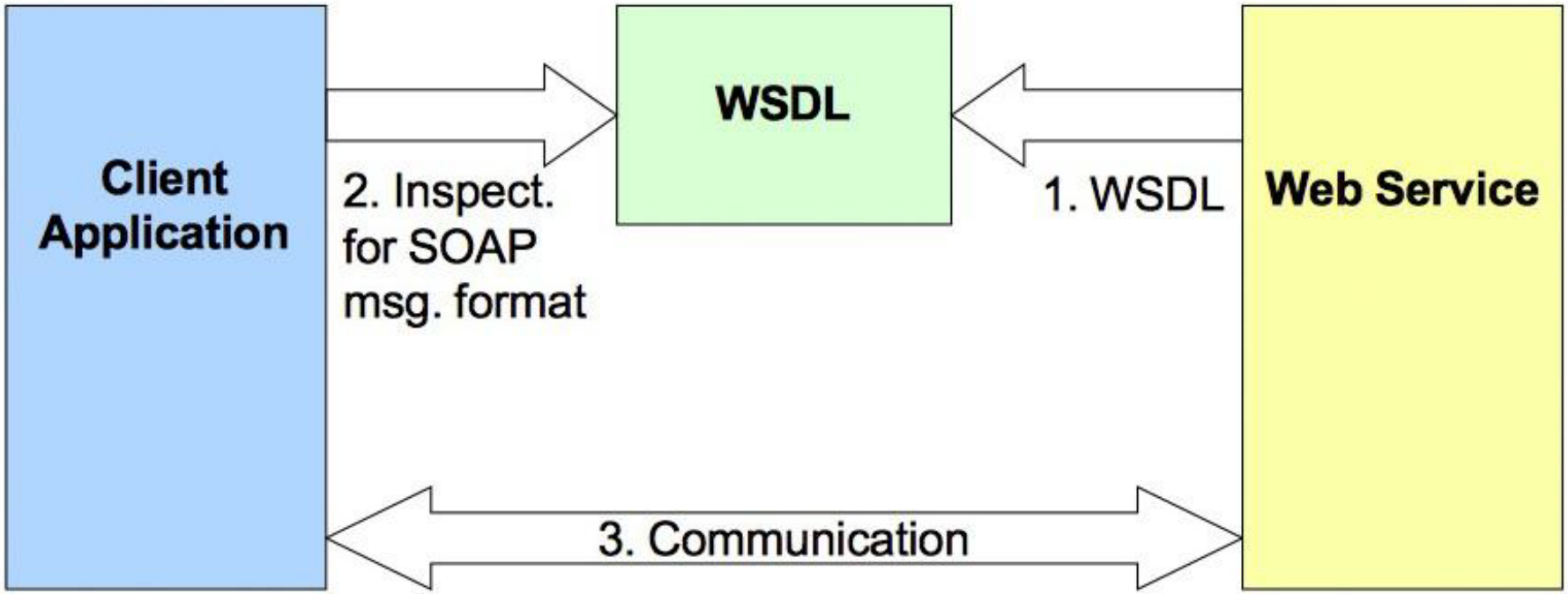
Arquitetura SOAP

O que é?
Como se inicia?
Como funciona?

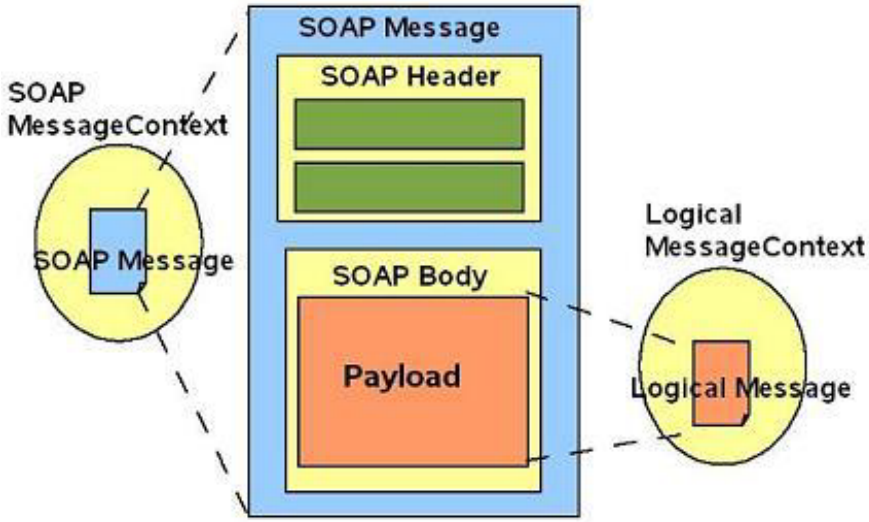
Arquitetura SOAP



Arquitetura SOAP



Arquitetura SOAP



Como eu faço para criar um web service SOAP então?

Java Web Service - SOAP

1. De posse de uma aplicação existente, defina quais serão os serviços expostos para serem invocados por outros sistemas.

2. Aprenda XML e XSD.
3. Aprenda a especificação SOAP e seus XSDs.
4. Aprenda a especificação WSDL e seus XSDs.

5. Gere o arquivo WSDL correspondente ao seu serviço, declarando os operações disponíveis.

Java Web Service - SOAP

- Implementar uma aplicação web que:
- Receba requisições HTTP.
- Processe a requisição, lendo o SOAP de requisição.
- Execute as regras de negócio de serviço - camada de
- negócio.
- Processe a resposta, gerando a SOAP de resposta.

Quanto tempo eu demoraria para aprender todas as especificações e implementar toda a infraestrutura necessária ?

Existem produtos prontos no Java para eu não perder meu tempo com infraestrutura de SOAP e seus XSDs?

Java Web Service - SOAP

Opções de produtos soap Java proprietários:

- Axis –http://axis.apache.org/axis2/java/core/
- CXF – http://cxf.apache.org/
- Spring SOAP Web Services - http://www.concretepage.com/spring-4/spring-4-soapweb-service-producer-consumer-example-with-tomcat
- Groovy-wslite - https://github.com/jwagenleitner/groovy-wslite

- Muitos outros...

Java Web Service - SOAP

Suporte para o desenvolvimento de web services SOAP com Java está oficialmente disponível na especificação desde o JSE 6 sobre o nome de:

JAX-WS: Java API for XML and Java Web Services.

Java Web Service - SOAP

Opções de providers de JAX-WS:

- **Metro – https://javaee.github.io/metro/**
- **Axis –http://axis.apache.org/axis2/java/core/**
- **CXF – http://cxf.apache.org/**

Java Web Service - SOAP

A implementação de referência de JAX-WS é desenvolvida como um projeto de código aberto e é parte do projeto GlassFish, um servidor de aplicações Java EE de código aberto.

Chamada de Metro, a JAX-WS é uma RI e destinada a ser uma implementação de qualidade referência para a comunidade.
Link oficial - https://javaee.github.io/metro/

Java Web Service - SOAP

Desde a versão JDK6, a JSE padrão já vem com o provider Metro de JAX-WS (jars), não sendo necessário baixar implementação para esse produto.

Baixe somente em casos no qual se deseje usar outro provider em específico.

Programação Para Internet

Aula: WSDL Overview

Prof. Anderson Augusto Bosing

WSDL

O que é?
Qual seu objetivo?
Ele composto por quais sessões?

WSDL Overview

WSDL Overview

Objetivo de WSDL:

1. Estabelecer um contrato entre o serviço e a aplicação cliente.

2. Descrever detalhes de serviço – "o que", "como" e "aonde".

XML do WSDL é composto por 5 sessões.

WSDL Overview

<types > - define ou importa dos tipos de dados XML via XSD utilizados pelo serviço.

<message > - define as mensagens disponíveis pelo serviço.

<portType> - nomeia abstratamente as mensagens disponíveis pelo serviço, como se fossem uma classe/interface Java.

WSDL Overview

<binding> - implementação concreta do serviço, fornecendo detalhes do serviço. Seria a implementação concreta da sessão <portType>.

<service> - define as conexões disponíveis e o endereço físico a ser usado como se fosse uma aplicação distribuída.

WSDL Overview

Para todas as informações acesse a especificação oficial -

http://www.w3.org/TR/wsdl

WSDL Overview

Criar um web service SOAP, como provedor de serviços consiste em:

1. Fazer o WSDL, estabelecendo o contrato e os serviços disponibilizados.

2. Fazer uma aplicação web que implemente o contrato dos serviços, recebendo as chamadas e processando das respostas SOAP.

WSDL Overview

Desenvolvedores Java não precisam se preocupar esses detalhes infraestruturais, pois o JAX-WS:

1. Gera 100% automático o arquivo WSDL a partir de do modelo de objetos POJO.

2. Disponibiliza o WSDL via URL padrão on-line.

3. Faz 100% automático parse de requisição SOAP, com base no modelo de objeto.

4. Faz 100% automático parse da resposta SOAP, com base no modelo de objeto.

WSDL Overview

Observação:

O arquivo de WSDL gerado pode conter variações de provider para provider, podendo apresentar grandes diferenças entre eles.

O que são servidores de aplicação?

O servidor de aplicação é um software voltado para desenvolvedores web. Também chamados de middleware, eles têm como objetivo conceder uma plataforma para que outras ferramentas sejam instaladas e executadas.

Exemplos de servidores de aplicação.

- Apache Tomcat
- IBM WebSphere
- GlassFish Server
- IIS
- Wildfly
- Projeto Payara

Instalação das ferramentas

Intellij

JDK 21

Wildfly

SOAP UI

Instalação das ferramentas

Wildfly 35

Para adicionar um usuário admin ao wildfly precisamos executar o seguinte processo:

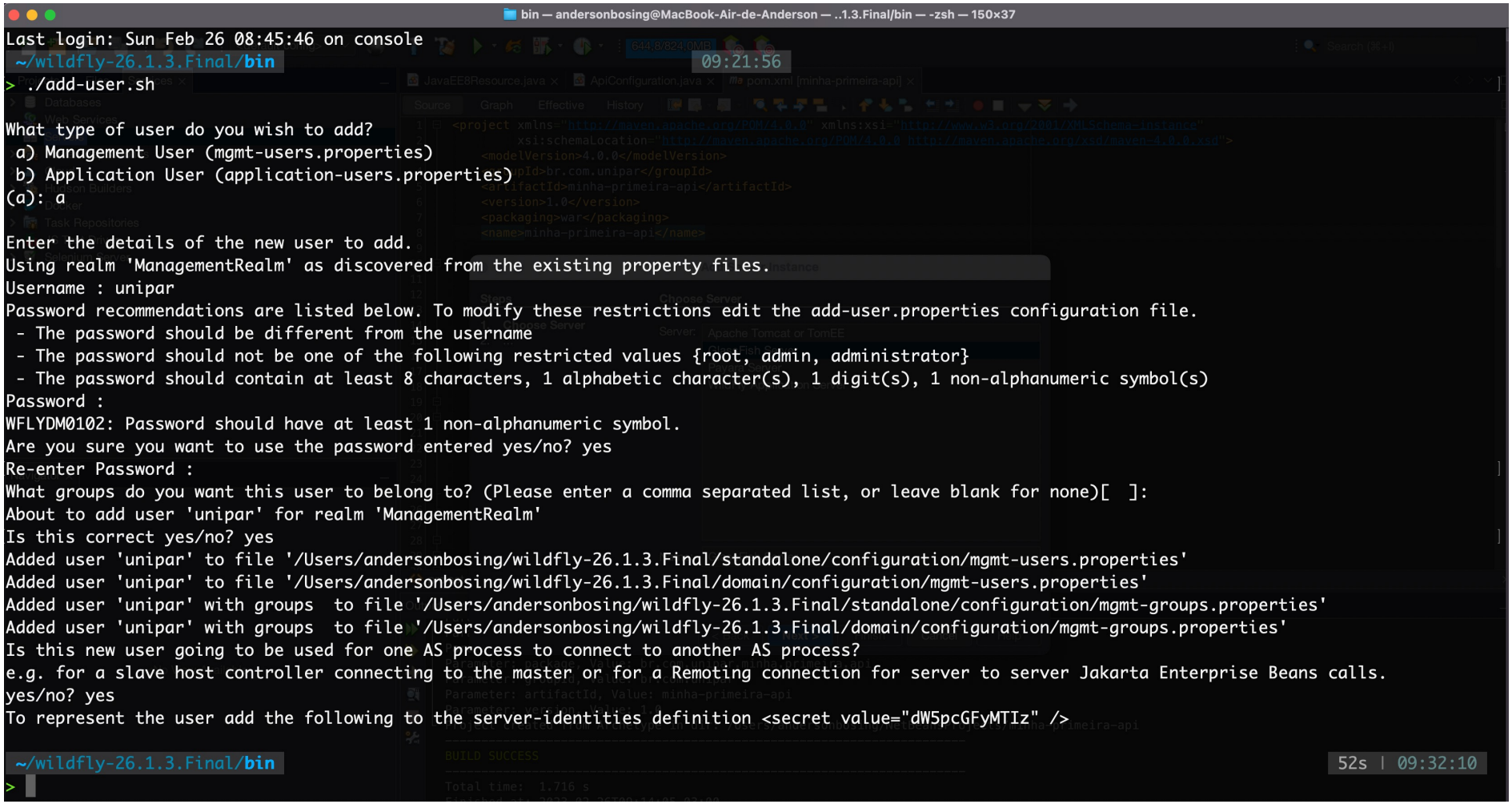
Acesse o seguinte path onde a instalação do wildfly foi descompactada: `~/wildfly-26.1.3.Final/bin`

Execute o script add-user.

Linux/Mac **./add-user.sh**

Windows **add-user.bat**

Instalação das ferramentas



WSDL - JAX-WS
Exercício 1:

- Implementar um serviço expondo operações de calculadora.
- Acessar o WSDL gerado.

  ?wsdl

O que são annotations no Java ?

Annotations são funções que podem ser usadas em classes, métodos ou atributos.

As annotations tem a seguinte assinatura @ seguido do nome da anotação.

As annotations podem servir para diferentes casos, como no exemplo acima a annotation @Table() serve para dizer que essa classe se refere a tabela de nome pessoa. Enquanto a annotation @id diz ao banco que aquele campo é uma chave primária e por fim a annotation @IsNotNull diz ao banco que aquela coluna é não nula.

Anotações JAX- WS
JAX-WS - Conceitos

Web services SOAP em Java são chamados – Endpoint Interface .

Endpoint interface em português “interface endpoint de serviço”, é um termo usado na plataforma Java Enterprise Edition ao expor qualquer componentes Java como um serviço web SOAP.

A maneira adequada para implementar um Endpoint SOAP em Java é:

Anotações JAX- WS
JAX-WS - Conceitos

1. Criando uma interface que defina os métodos que serão expostos no WSDL. Esta interface é chamada de SEI: Interface Endpoint Service.

2. Criando um objeto POJO que implemente polimorficamente a interface, fornecendo a implementação concreta dos métodos que serão expostos como serviços. Este objeto é chamada de SIB: Service Implementation Bean.

Anotações JAX- WS

Para habilitar o web service, usar as anotações do JAX-WS localizados no pacote javax.jws* na SEI e SIB para configurar declarativamente os detalhes e comportamento na exposição do web services. Veja as anotações básicas e suas funções:

Nas versões mais atuais do java as anotações do JAX-WS se encontram no pacote: jakarta.jws.*

Anotações JAX- WS

@WebService – indica que a classe ou interface implementa um web service. Deve ser especificado no SEI e SIB.

Todos os métodos public serão automaticamente publicado no serviço.

Anotações JAX- WS

@WebService.targetNamespace – customiza o namespace do WSLD gerado. Usa o nome da classe totalmente qualificado com padrão. Deve ser especificado no SEI e SIB.

@WebService. serviceName – customiza o nome do serviço no WSLD gerado. Usa o nome da classe totalmente qualificado com padrão. Deve ser especificado no SEI e SIB.

@WebService.endpointInterface – indica que o SIB implementa a interface SEI. Deve ser especificado somente no SIB, amarrando com a interface SEI.

Anotações JAX- WS

@WebMethod - Personaliza um método exposto como uma operação de serviço Web. O método associado deve ser público e seus parâmetros retornam valor, e as exceções devem seguir as regras definidas pelo padrão SOA. Deve ser especificado somente no SEI, amarrando com a interface SIB.

@WebResult - Personaliza o mapeamento do valor de retorno para uma parte WSDL e um elemento XML. Deve ser especificado somente no SEI, amarrando com a interface SIB.

Anotações JAX- WS

@WebParam - Customiza o mapeamento de um parâmetro individual para uma parte de mensagem para um elemento XML de serviço da Web.

Aplique esta anotação aos métodos em um SEI (Service Endpoint Interface) de cliente ou servidor ou em uma classe de implementação de terminal de servidor.

Propriedade header = true determina que será recepcionado dado no header da chamada.

Falhas de SOAP

Seguindo a especificação SOAP, quando uma requisição a um serviço soap falha por erro de negócio, ele deve gerar um SOAP FAULT e gerar na resposta do SOAP.

Desenvolvedores java não precisam se preocupar com isso, sendo que o JAX-WS mapeia automaticamente qualquer exception padrão Java para a especificação SOAP FAULT, tanto no servidor quando no cliente, fazendo o parse de server e de cliente de forma transparentemente.

Anotações JAX- WS

@WebFault - A anotação @WebFault mapeia falhas de WSDL em exceções Java. Ela é utilizada para capturar o nome da falha durante a serialização do tipo JAXB que é gerado a partir de um elemento global mencionado por uma mensagem de falha de WSDL. Ela pode ser utilizada também para customizar o mapeamento de exceções específicas do serviço para falhas de WSDL.

Essa anotação pode ser aplicada apenas a uma classe de implementação de falha no cliente ou servidor.

Falhas de SOAP

Exercicio:

- **Implementar um serviço de cadastro de produtos que gere falhas de soap quando o nome e valor do produto não for corretamente preenchido.**

- **Acessar o WSDL gerado.**

- **Verifique o tipo do <fault message> da operação de retorno.**

Anotações JAX- WS

Exercício 2:

Implementar um web service de Livraria On-Line, customizando o nome do serviço WSDL.

Acessar o WSDL gerado.

Consumir o WSDL no SOAP UI.

Anotações JAX- WS

Observação:

Veja que o desenvolvedor Java também não precisa se preocupar com a criação do XSD de objeto de negócio trafegado dentro do protocolo SOAP, sendo que o JAX-WS gera o XSD no WSDL. Acesse o <xsd:import> do WSDL no exercício anterior e veja o XSD online do objeto Livro.

Anotações JAX- WS

Para excluir métodos use na SEI: Use a anotação @WebMethod(exclude=true) somente naquele(s) métodos que se deseje excluir.

Eles não serão gerados no WSDL e não serão invocados pelo SOAP.