WebServices

Leonardo Marcelino
http://www.itnerante.com.br/profile/LeonardoMarcelino

Arquitetura Orientada a Serviços

*É um **paradigma** para organização e utilização de recursos (capabilities) **distribuídos** que estão sob o controle de diferentes domínios proprietários*.

OASIS, rm-soa 2006

Serviço

*É um mecanismo que permite acessar um conjunto de recursos (capabilities), no qual o acesso é fornecido por meio de uma **interface** descrita e exercitada consistentemente de acordo com **restrições e políticas** especificadas pela **descrição do serviço**. É oferecido por uma entidade (**provedor de serviço**) para uso de terceiros (**consumidor de serviço**), sem contudo, haver necessidade desses terceiros conhecerem o provedor de serviços, podendo inclusive fazer uso do serviço de forma a extrapolar o escopo original concebido pelo provedor.*

OASIS, rm-soa 2006

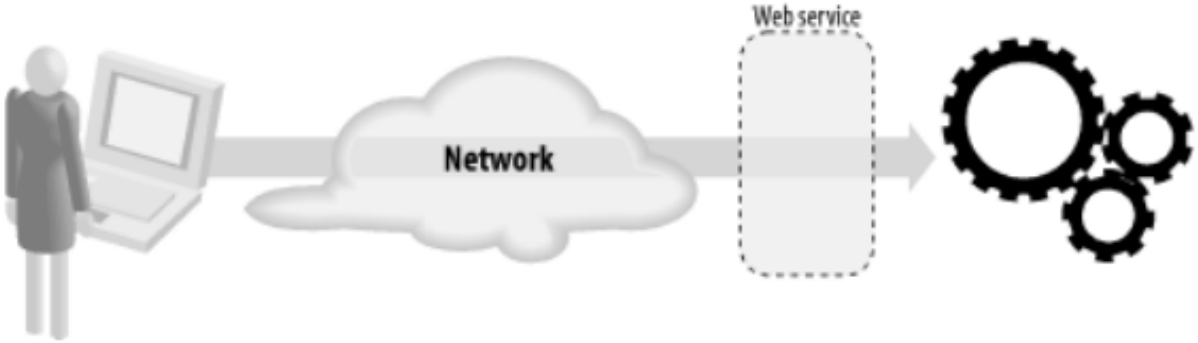
e-PING 2012

- adoção gradual de SOA é diretriz técnica para integração de sistemas
- web services como solução de interoperabilidade
- Especificações **Adotadas (A)**
  - SOAP v1.2 e HTTP 1.1 p/ REST
  - WSDL 1.1 (note)
  - UDDI v3.0.2 (draft)
- Especificações **Recomendadas (R)**
  - WS-Security 1.1
  - WS-Trust 1.4
- Especificações **em Estudo (E)**
  - WSDL 2.0
  - ebXML

web services

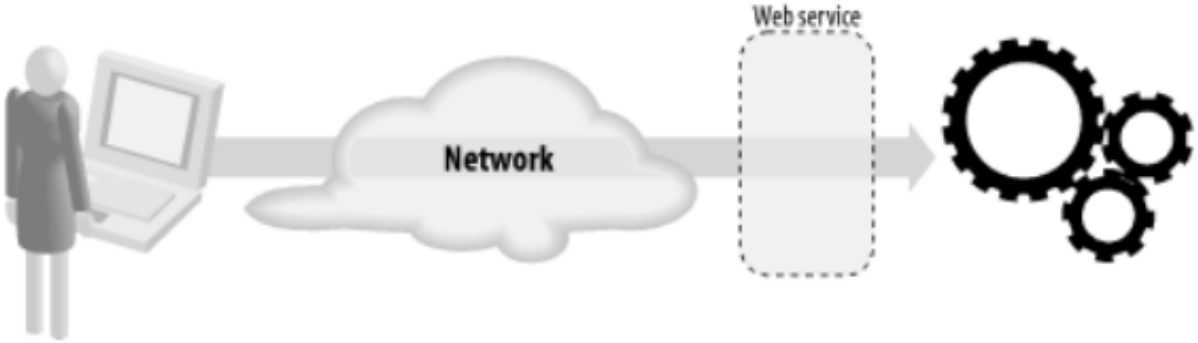
Definição W3Schools

São componentes de aplicativos **baseados em XML**, **auto-contidos** e **auto-descritivos**, que se comunicam usando **protocolos abertos**. Podem ser descobertos com UDDI e ser utilizados por outras aplicações.



web services

- lógica de aplicação disponível na web
- componente de software
  interface para acessar funcionalidades de uma aplicação
  rede de computadores (internet/intranet)
- utilidade
  - reuso de componentes (mashups: mapas, conversão de moedas)
  - conectar aplicações (sistemas legados, plataformas JEE, .NET)



web services: Características

- protocolos padrões da internet (HTTP, XML, etc)
  - troca de mensagens
    - SOAP (mais comum) ou XML-RPC
    - pode usar HTTP (RESTFull)
- Independência
  - plataforma, sistema operacional, linguagem de programação
- baseado em XML (XML-based)
  - representação e transporte de dados
- baixo acoplamento (loosely-coupled)
  - interface abstrata
  - consumidor e provedor
  - entre aplicações (serviços)
- granularidade grossa (Coarse-grained)
  - serviço expõe a quantidade certa de lógica do negócio

Exercícios

[01] CESPE – 2010 – MPU

Acerca de interoperabilidade de sistemas, web services e arquitetura e-Ping, julgue os próximos itens.

A tecnologia Web Services e o uso do simple object access protocol (SOAP) são recomendados pelo e-Ping, que não recomenda o uso do HTTP (hypertext transfer protocol).

Gabarito: ERRADO

Exercícios

[02] CESPE – 2009 – CEHAP-PB

Com relação a SOA e Web services, assinale a opção correta.

a) A interface de Web service define os dados disponíveis e como eles podem ser acessados de modo a que o fornecimento de serviços seja independente da aplicação que o utiliza.

b) Conceitualmente, os scripts de serviços consideram que um provedor de serviço o oferece pela definição de seus dados e pela implementação de sua funcionalidade.

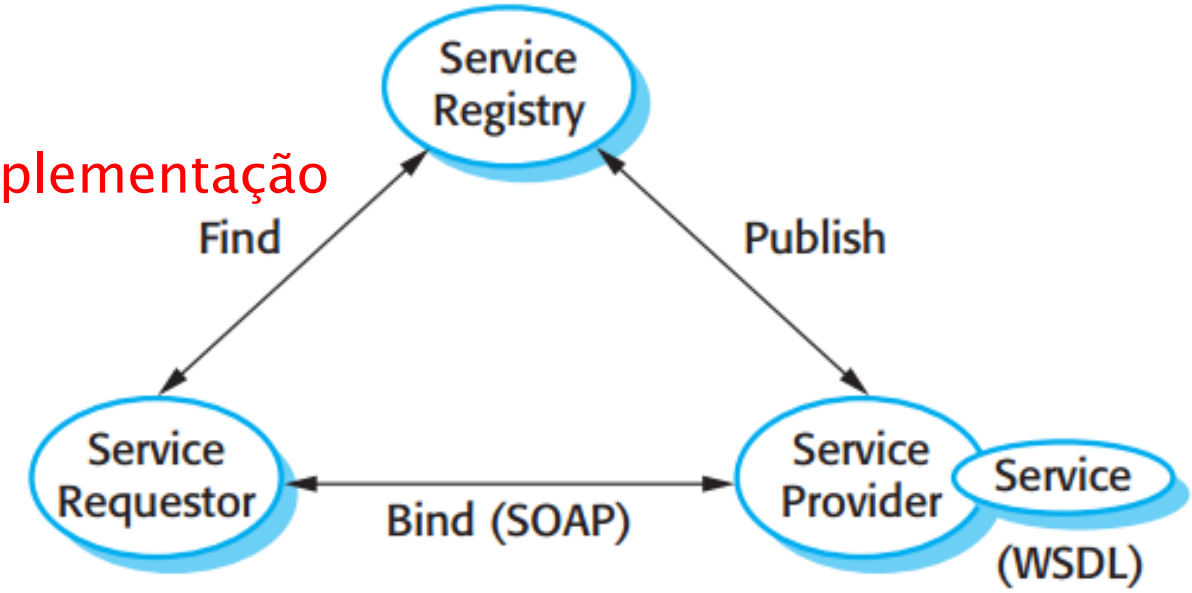
c) Os serviços podem ser oferecidos por qualquer cliente de serviços dentro ou fora de determinada organização e tornam públicas as informações sobre o serviço para que usuários autorizados possam utilizá-la.

d) São padrões de Web services o SOAP, o WSDL e o UDDI, todos baseados em HTTP.

Gabarito: **Letra A**

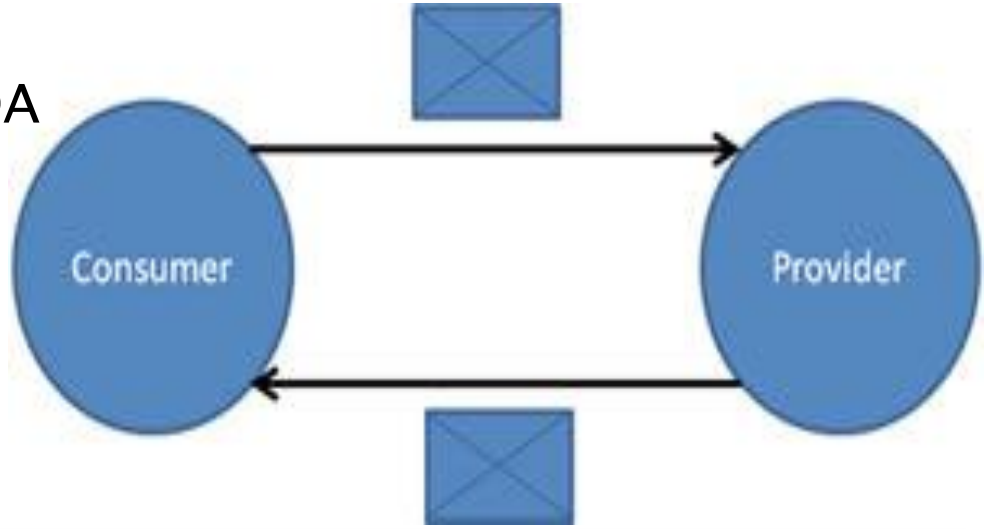
Modelo arquitetural

- modelo baseado em web services
- componentes básicos
  - solicitante do serviço
  - provedor de serviços
  - registro de serviços
- modelo triangular
- SOA independe de implementação



Modelo arquitetural

- modelo SOA primitivo (Erl)
- componentes básicos
  - solicitante do serviço
  - prestador de serviços
- alinhado ao RM-SOA
- função descoberta (registro)
  - característica SOA
  - entregue com princípios SOA



Processo W3C

- processo para aprovação de padrões
- processo
  - Submission
  - Note
  - Working Group
  - Working Draft
  - Candidate Recommendation
  - Proposed Recommendation
  - Recommendation

Arquitetura web services

- não existe uma W3C Recommendation
- Arquitetura Web Services
  - W3C Working **Draft** – Nov/2002
    - descoberta é parte da definição
    - modelo arquitetural triangular (arquitetura básica)
  - W3C Working **Draft** – May/2003
    - abandona modelo triangular
    - Basic Architectural Roles (novo modelo)
  - W3C Working **Draft** – Aug/2003
    - altera definição de web services
  - W3C Working Group **Note** – Feb/2004
    - abandona arquitetura baseada em funções
    - processo geral de contratação
    - meta-modelo com 4 modelos arquiteturais

Arquitetura web services

- W3C Working **Draft** – Nov/2002

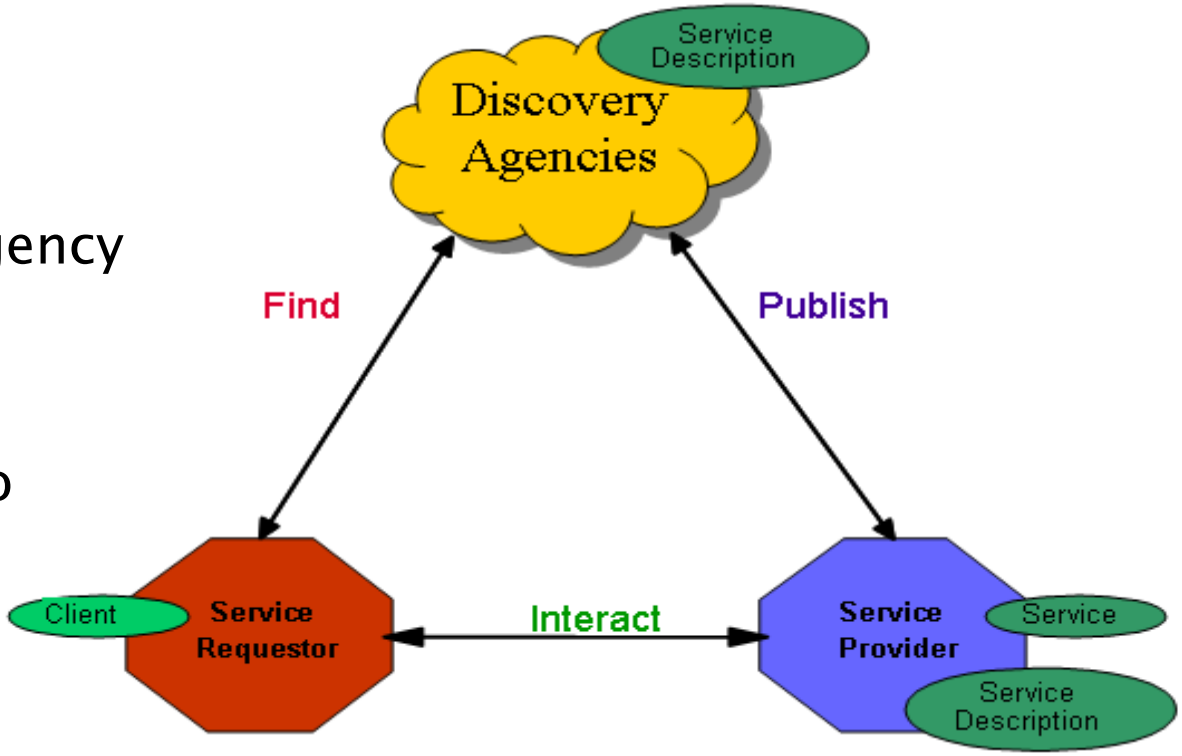
  *serviço web é um sistema de software **identificado por uma URI**, cujas interfaces pública e **ligações** são definidas e descritas usando XML. Sua definição pode ser **descoberta** por outros sistemas de software. Estes sistemas podem então interagir com o serviço web de uma maneira prescrita por sua definição, usando mensagens baseadas em XML transmitidas por protocolos de internet.*

- W3C Working Group **Note** – Feb/2004

  *serviço web é um sistema de software projetado para suportar interação máquina-a-máquina interoperável sobre uma rede. Tem uma **interface** descrita num **formato máquina processável** (especificamente **WSDL**). Outros sistemas interagem com o serviço web de uma maneira prescrita por sua descrição usando **mensagens SOAP**, normalmente transmitidas via HTTP com uma serialização XML em conjunto com outros padrões relacionados à web.*

Arquitetura Básica (draft)

- interação entre agentes com troca de mensagens
- tecnologias web service
  - troca de mensagens
  - descrição dos serviços web
  - publicação e descoberta de serviços web
- modela interações
  - service requestor
  - service provider
  - service discovery agency
- Componentes
  - serviço
  - descrição do serviço
- Ligação ao serviço
  - estática
  - dinâmica



Arquitetura Básica (draft)

Papéis (Roles)

- provedor de serviços
  perspectiva
    - negócio: dono do serviço
    - arquitetural: plataforma que dá acesso
- solicitante de serviço
  perspectiva
    - negócio: requer a funcionalidade
    - arquitetural: aplicativo que interage com o serviço
- agência de discovery
  - repositório de descrições
  - centralizada ou distribuída
  - publicação
    - push: provedor envia a descrição
    - pull: agência busca a descrição

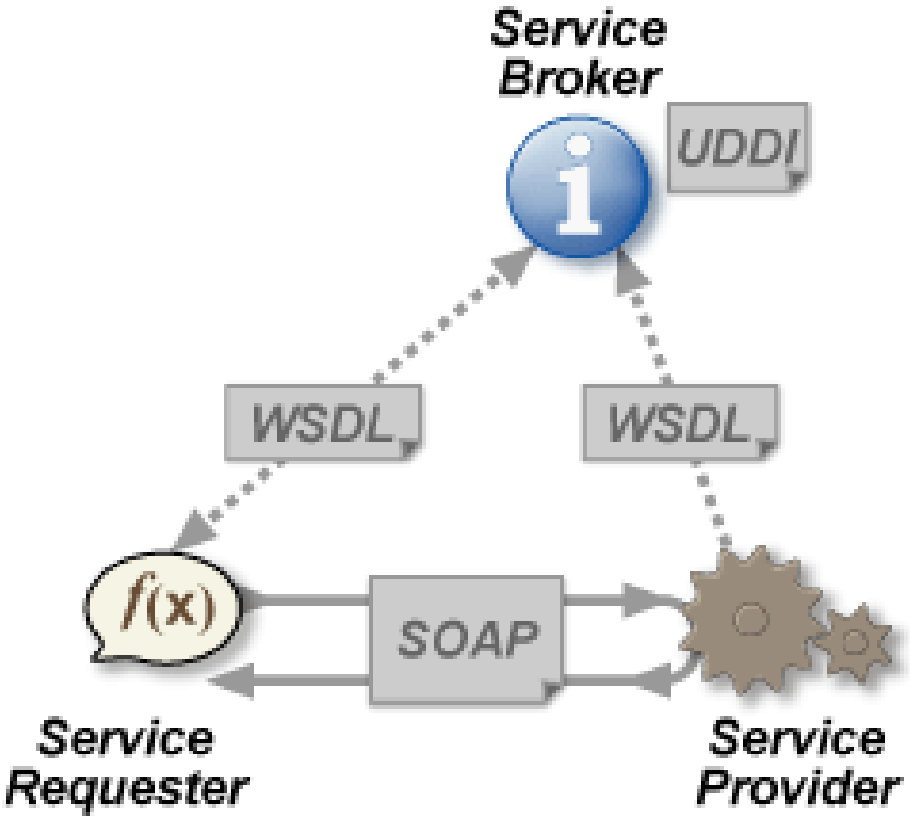
Arquitetura Básica (draft)

Operações

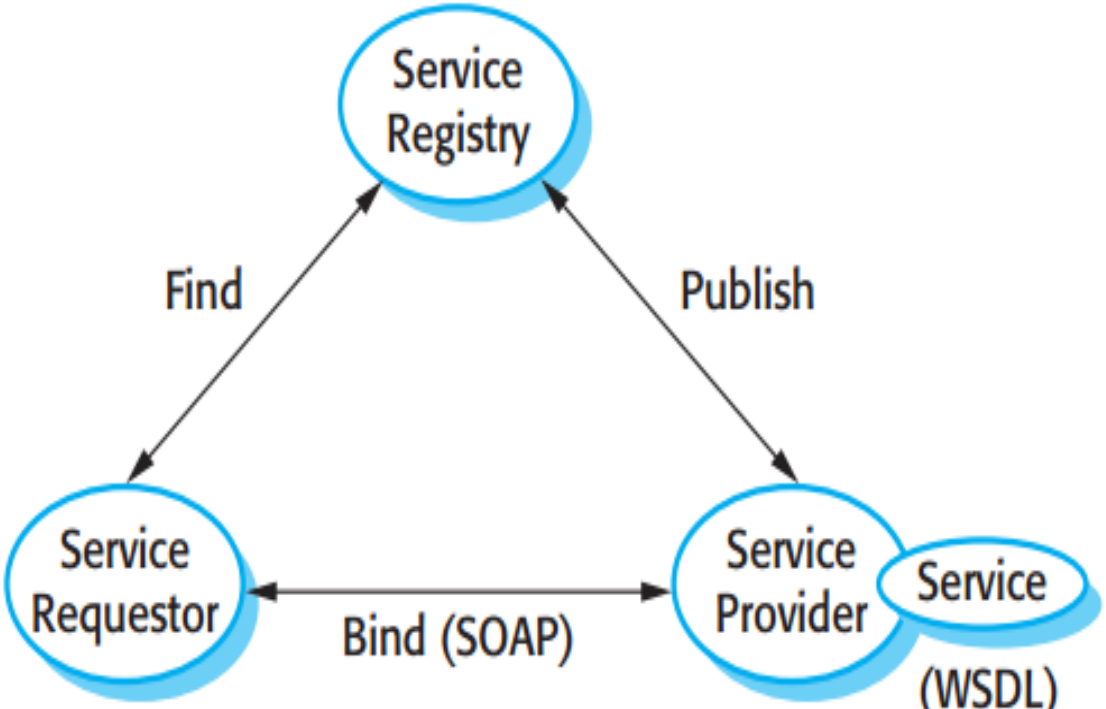
- comportamentos
  - publicar descrições de serviço
  - buscar e recuperar descrições de serviço
  - ligar ou invocar serviços baseados na descrição do serviço
- publicar
  - torna o serviço acessível
- buscar
  - recupera uma descrição do serviço
  - diretamente ou repositório
- interagir
  - invoca um serviço e inicia a interação

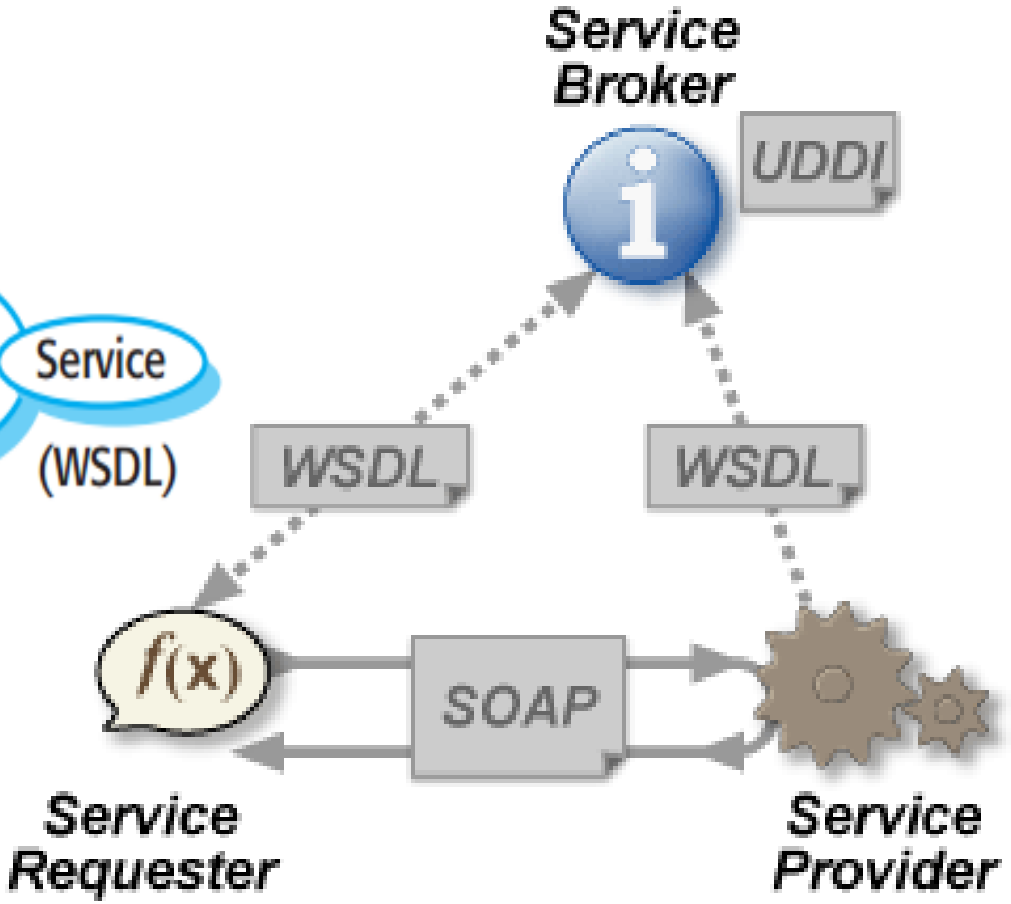
Arquitetura W3Schools

- plataforma básica: XML + HTTP
- elementos básicos da plataforma
  - intercâmbio: SOAP
  - descrição: WSDL
  - publicação: UDDI
- utilizado em
  - reutilização de componentes
  - conexão entre aplicações



Arquitetura W3Schools





Arquitetura (note)

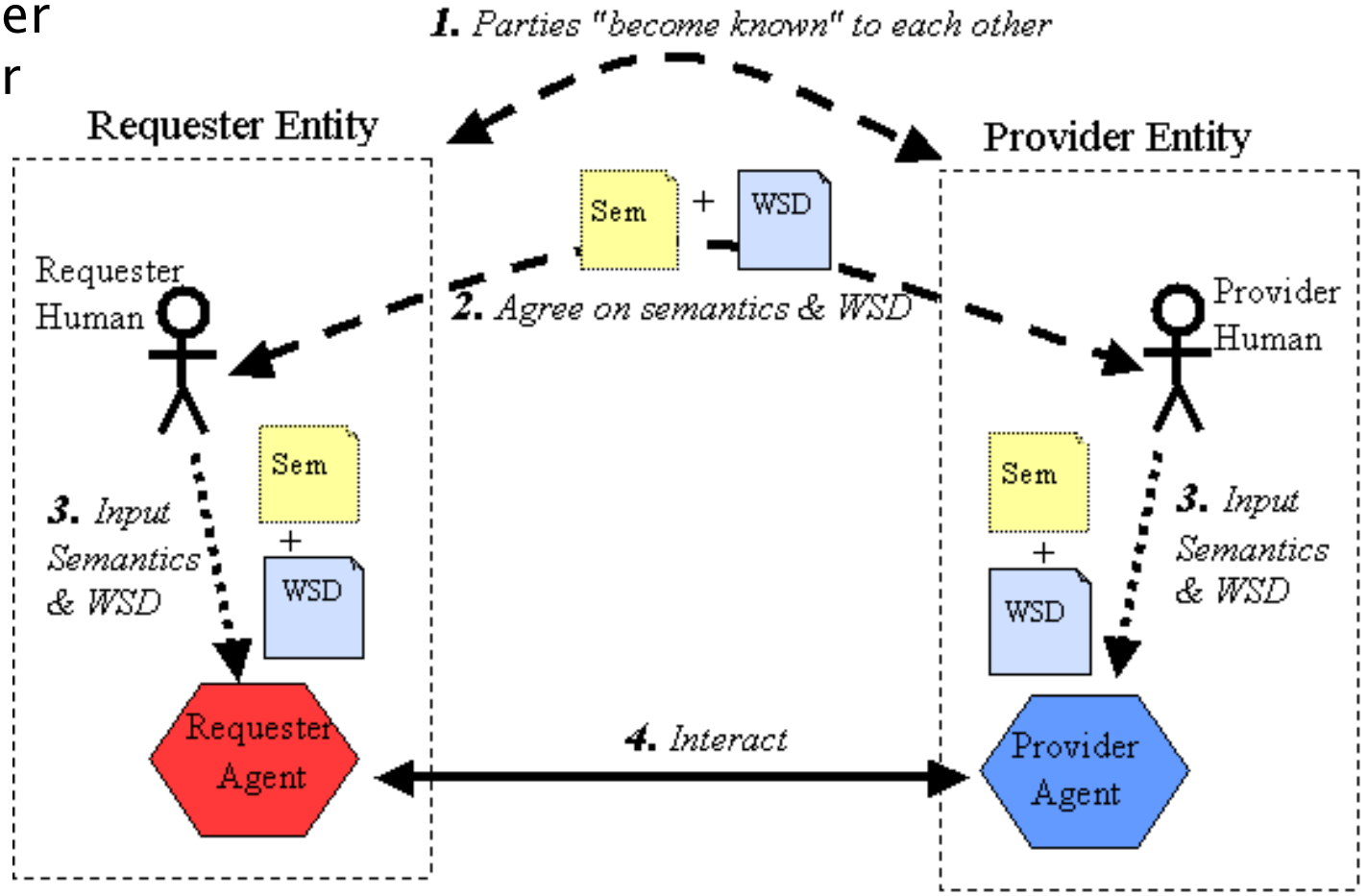
- objetivos
  - fornecer uma definição comum de um serviço web
  - definir um serviço web dentro de uma estrutura de serviços
  - garantir a interoperabilidade entre os serviços web
- descreve características mínimas
  - não especifica como implementar serviços web
  - não impõe restrições sobre combinação de serviços
  - identifica elementos globais de serviços web
- definição de serviço
  - apresenta apenas duas partes
    - solicitante e provedor
  - descoberta não faz parte da definição básica
- Sommerville
  - SOAP, WSDL e WS-BPEL
  - UDDI não foi adotado

Arquitetura (note)

- agente x serviço
  - serviço: abstrato
  - agente: implementa o serviço
- solicitante e provedor
- Web Service Description (WSD)
  - documenta a mecânica da troca de mensagens
  - especificação da interface processável por máquina
  - escrito em WSDL
- Semântica (SEM)
  - inclui informações não definidas no WSD
  - descrita como um documento, mas pode ser verbal
  - formato mutuamente aceitável

Arquitetura (note)

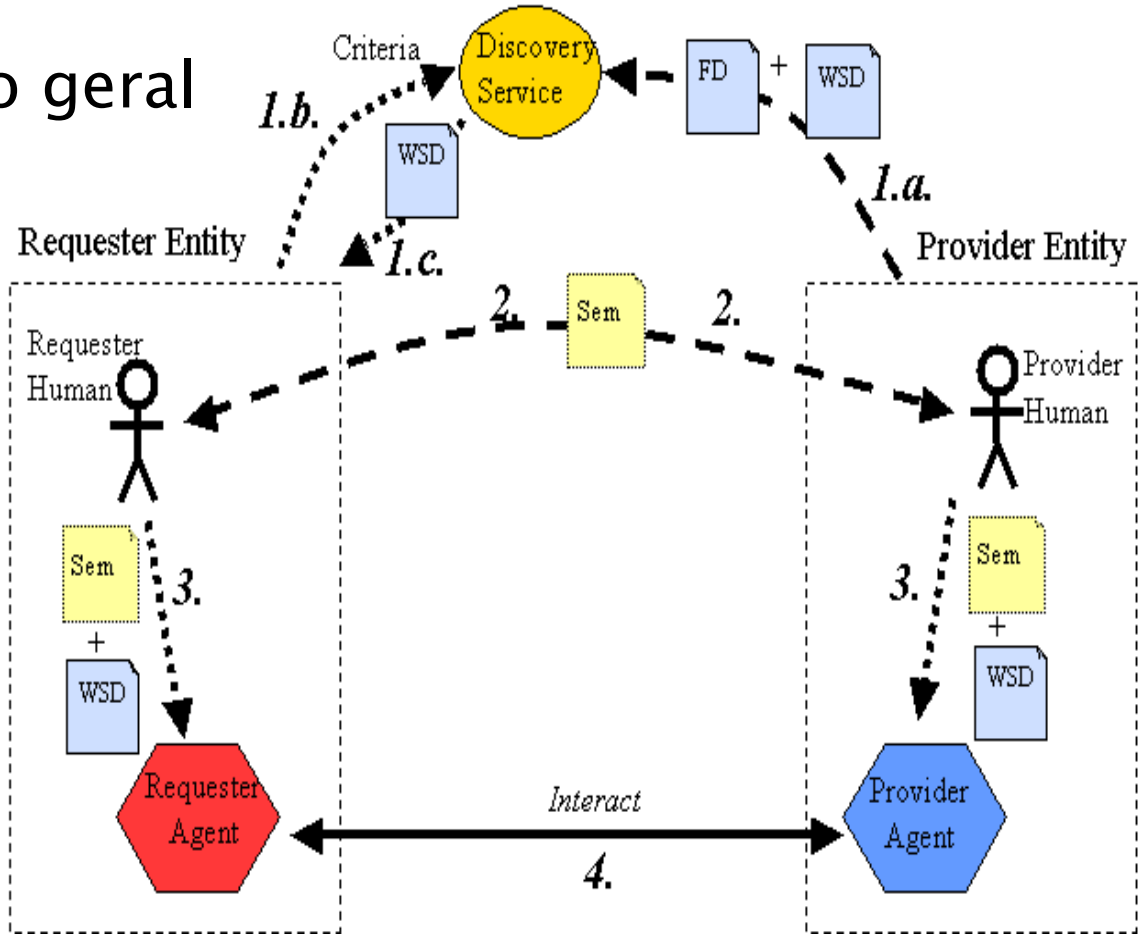
- processo geral para contratar serviço web
- basic architectural roles
  - requester
  - provider



Arquitetura (note)

Discovery

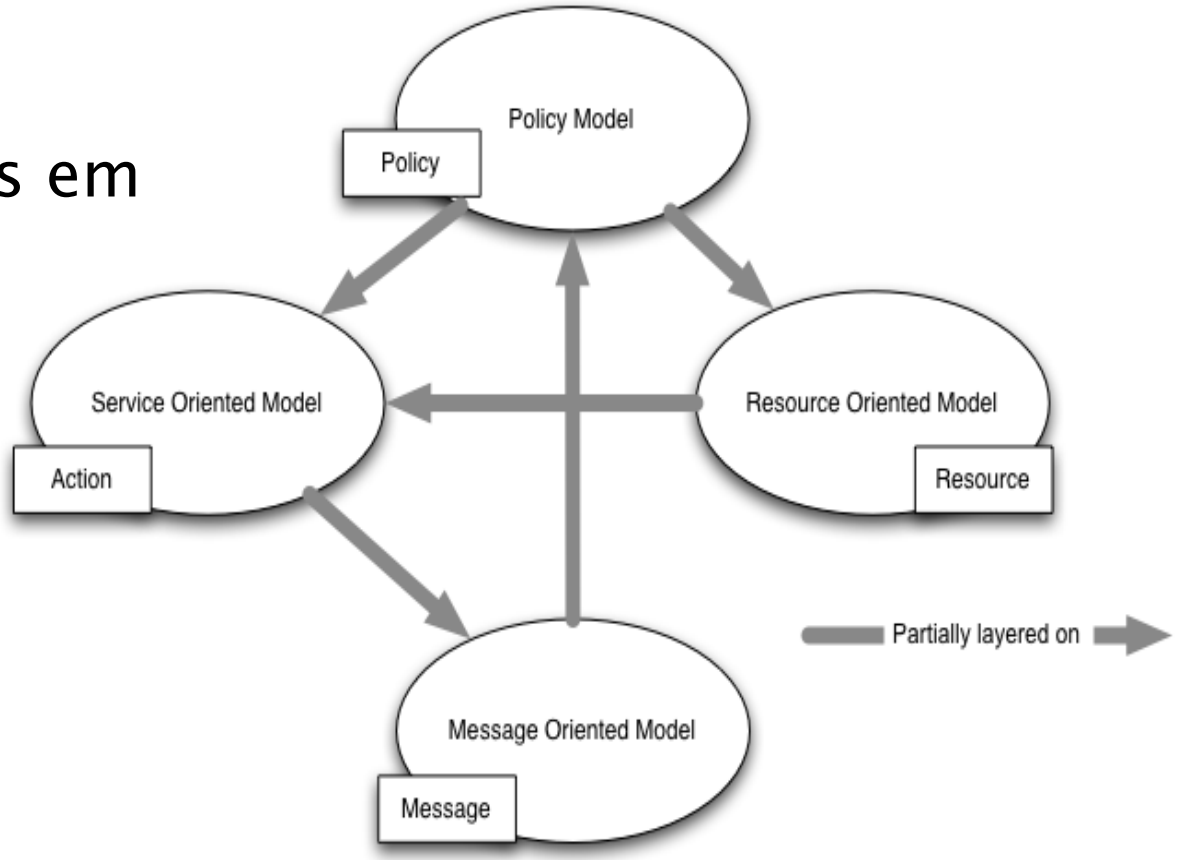
- papel lógico
- expande o processo geral
  - altera o 1º passo
- tipos de Discovery
  - registro
  - índice
  - peer-to-peer



Arquitetura (note)

Modelos arquiteturais

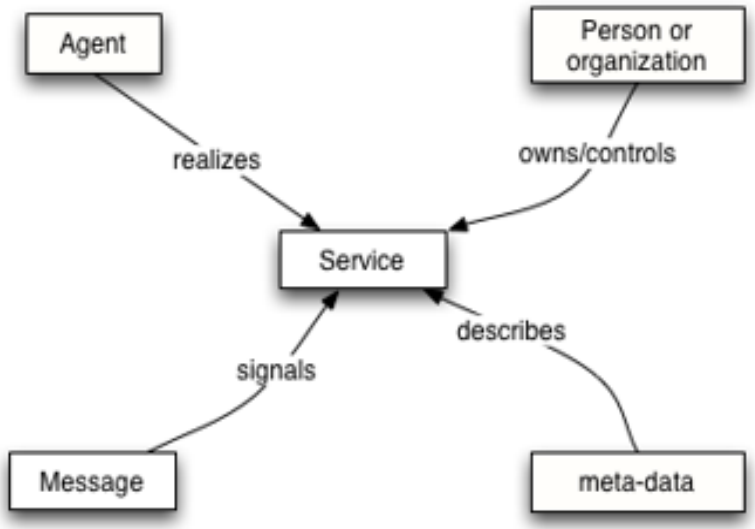
- arquitetura baseada em papéis
  - draft de 2002
  - tutorial W3Schools
- modelos baseados em
  - serviços
  - mensagens
  - recursos
  - políticas



Arquitetura (note)

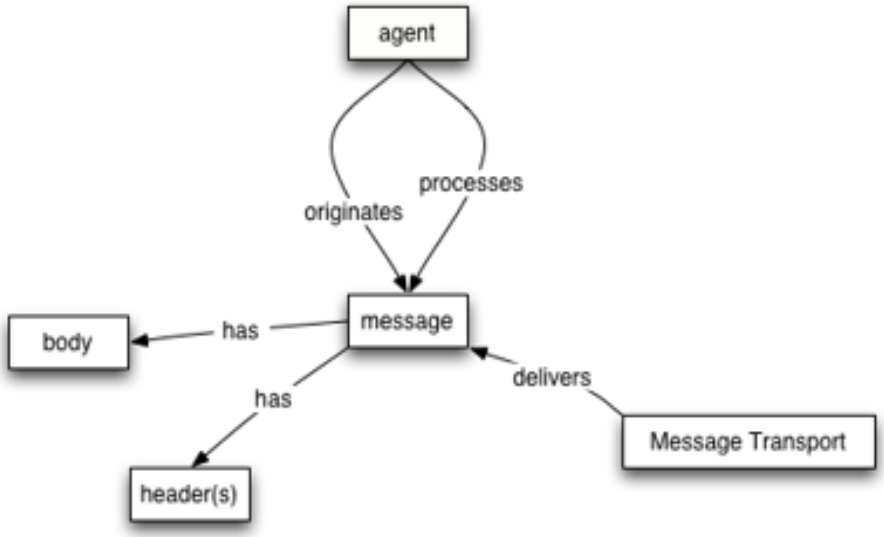
Modelos arquiteturais

modelo orientado a serviços



- mais complexo
- foco no serviço e suas ações
- realizado por meio mensagens
- forte relação com o mundo real
- uso de metadados (descrição)

modelo orientado a mensagem

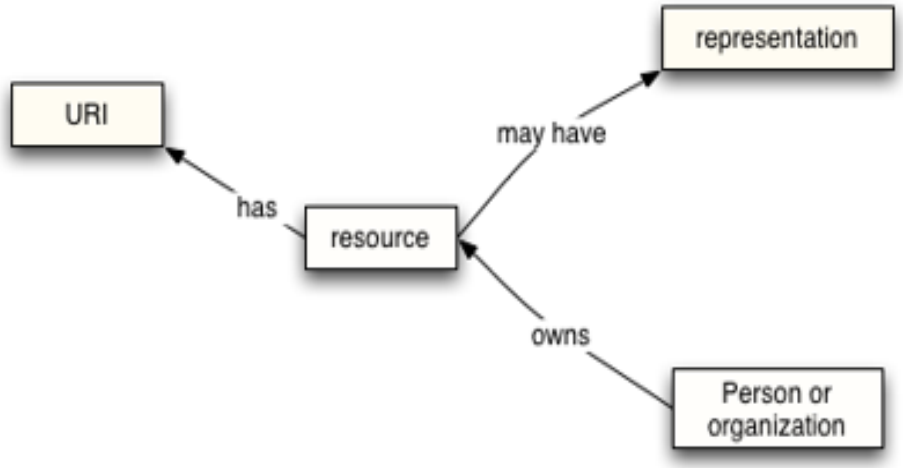


- foco na mensagem, sua estrutura e transporte

Arquitetura (note)

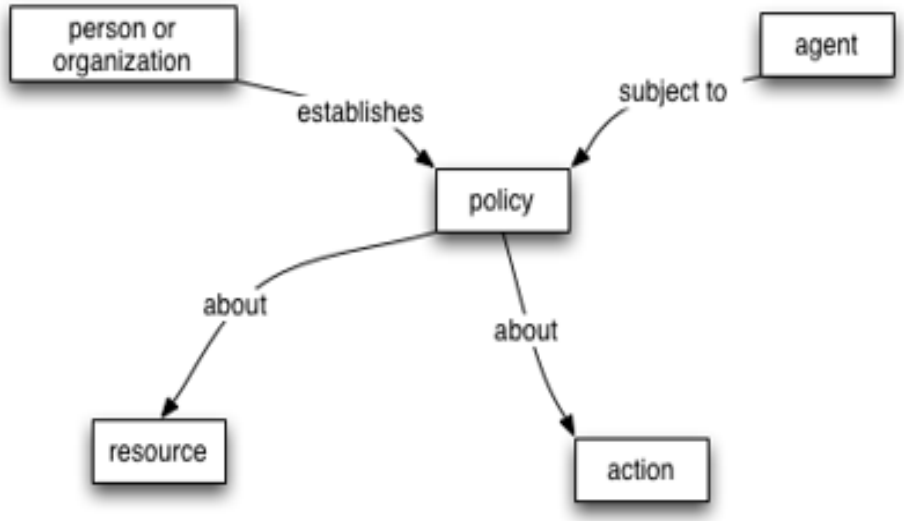
Modelos arquiteturais

modelo orientado a recursos



- foco no conceito de recursos web
- modela relações entre recursos e proprietários

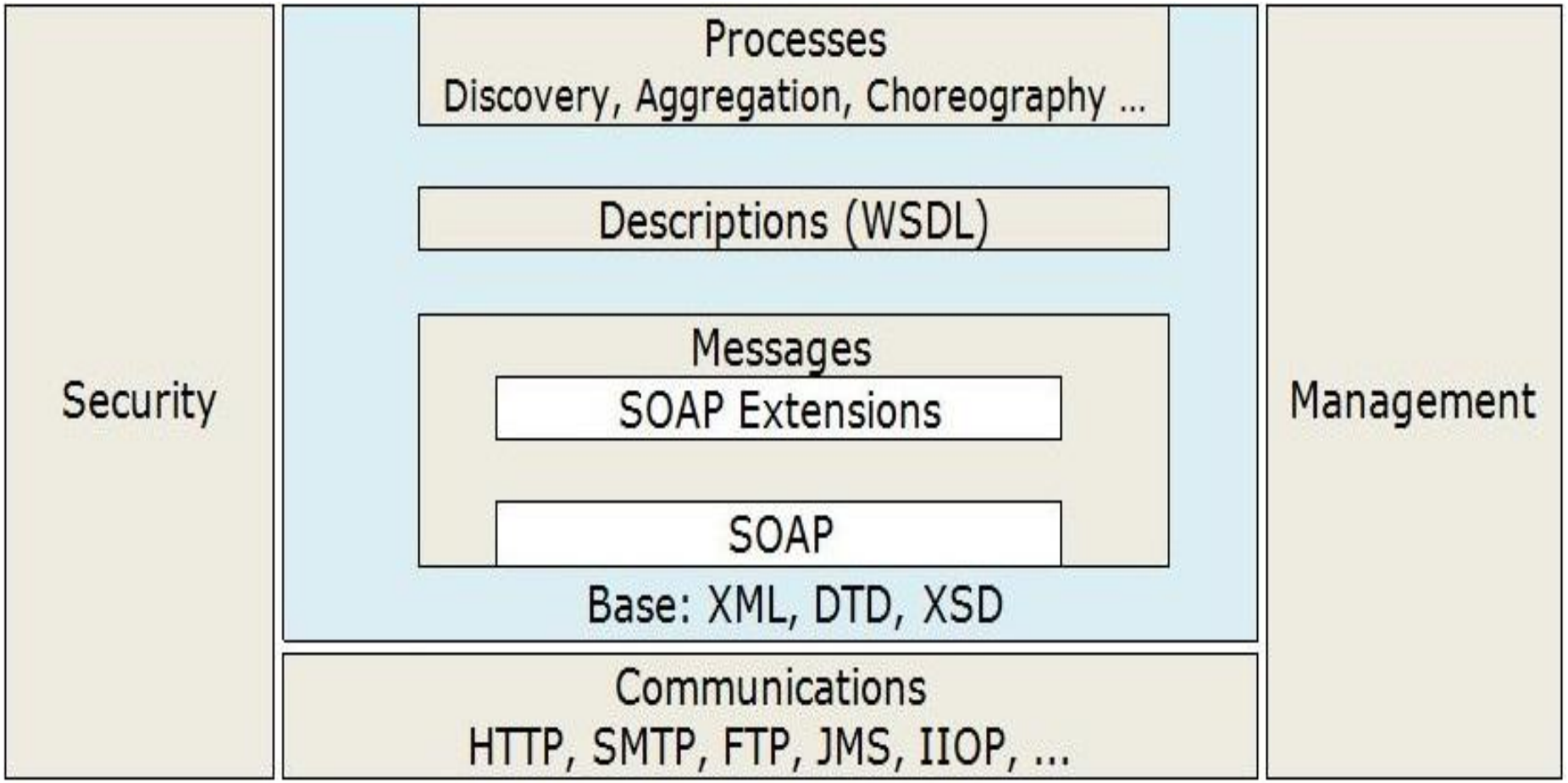
modelo orientado a políticas



- foco em restrições sobre agentes e serviços
- modela a aplicação de políticas

Arquitetura (note)

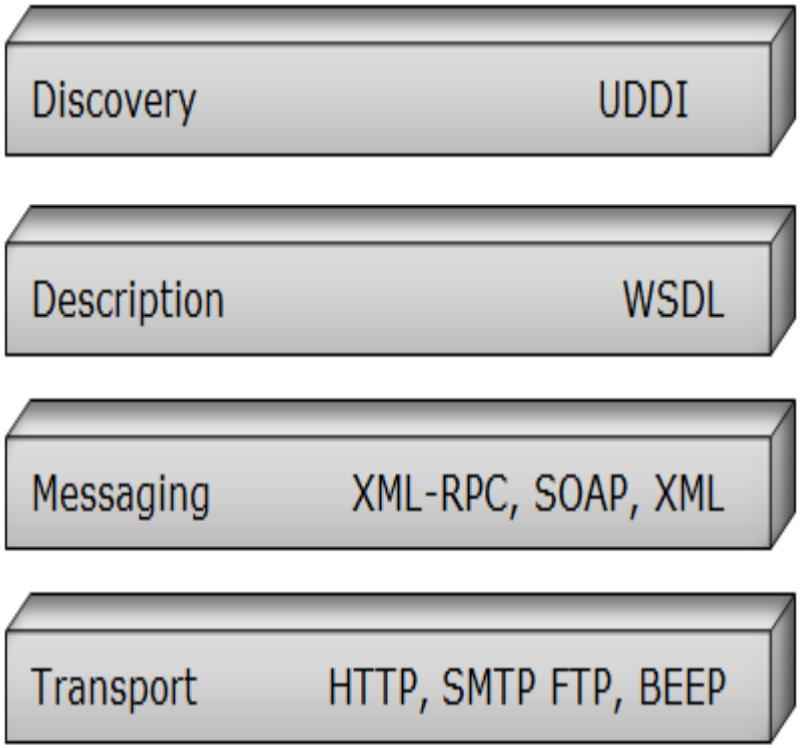
Camadas por tecnologia



web services

Pilha de protocolos (Stack) web services

- transporte
  transporte de mensagens entre aplicações
- mensagem
  método comum de comunicação
  codifica mensagens em XML
- descrição
  descreve a interface do serviço
- descoberta
  centraliza serviços no repositório
  funcionalidades publicar/encontrar



Exercícios

[03] FCC – 2012 – TJ-PE

No tocante a arquitetura orientada a serviços, mais especificamente a serviços web (webservices), considere:

I. Estes serviços proporcionam um padrão de interoperação entre aplicações, podendo ser executados em várias plataformas.

II. Outros sistemas interagem com esse serviço por meio de mensagens no protocolo SOAP, tipicamente enviadas por HTTP.

III. O objetivo deste serviço é proporcionar alguma funcionalidade em favor de seu proprietário (pessoa ou organização).

Está correto o que se afirma em

a) I, II e III.
b) I e II, apenas.
c) II, apenas.
d) I e III, apenas.
e) II e III, apenas.

Exercícios

[04] FCC – 2009 – SEFAZ-SP

Uma vantagem que o Web Service oferece

I. em relação à empresa que desenvolve uma DLL, não precisa distribuí-lo para todos os clientes, pois estará armazenado em um único lugar de onde será acessado.

II. o acesso a ele sempre por meio de HTTP, mas internamente existe uma string XML que está empacotada em um protocolo SOAP (Simple Object Access Protocol).

III. ser transparente para o Firewall de uma empresa, pois, como é uma string XML, é interpretado como um arquivo "texto", não precisando pedir autorização do Firewall para entrar.

Está correto o que consta em

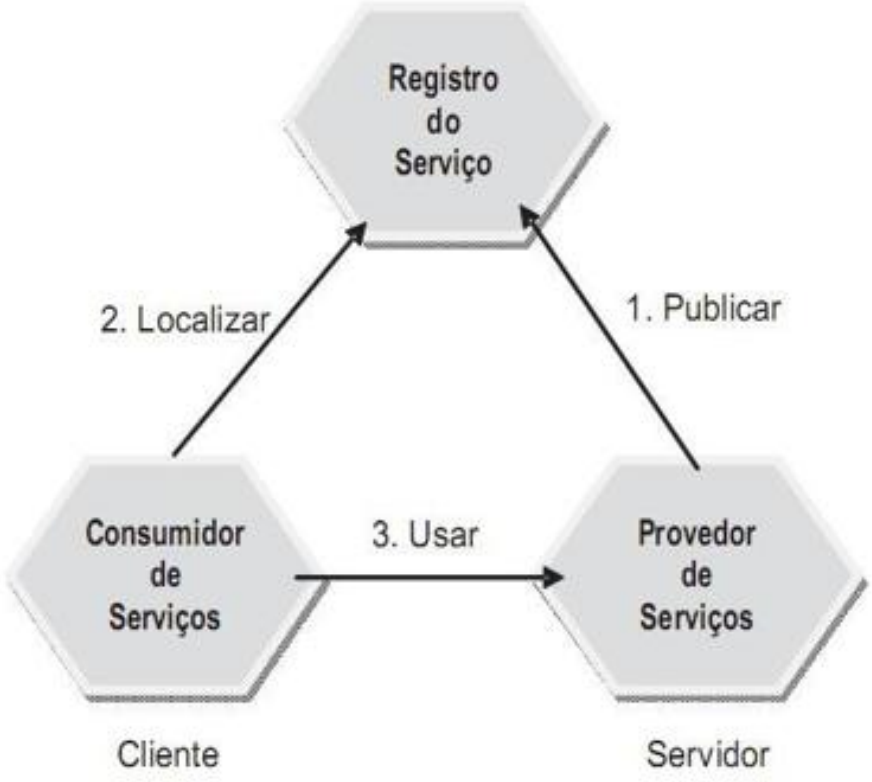
a) I, II e III. b) I e II, apenas. c) I e III, apenas.
d) II e III, apenas. e) II, apenas.

Exercícios

[05] CESGRANRIO – 2010 – IBGE

A figura acima apresenta um modelo básico de interação suportado por Web Services. Os padrões utilizados pelas ligações 1, 2 e 3, respectivamente, são:

a) UDDI, WSDL, SOAP.
b) UDDI, SOAP, WSDL.
c) WSDL, UDDI, SOAP.
d) WSDL, SOAP, UDDI.
e) SOAP, UDDI, WSDL.



Exercícios

[06] CESPE - 2009 - ANTAQ

No que se refere a arquiteturas e tecnologias de sistemas de informação, julgue o item.

Web service é um conjunto de tecnologias utilizadas na integração de sistemas e na comunicação entre aplicações diferentes. Para a representação e estruturação dos dados nas mensagens recebidas/enviadas, é utilizado o XML (eXtensible markup language). As chamadas às operações, incluindo-se os parâmetros de entrada e saída, são codificadas no protocolo UDDI (universal description, discovery and integration). Os serviços (operações, mensagens, parâmetros etc.) são descritos usando-se a linguagem WSDL (web services description language). O processo de publicação, pesquisa e descoberta de web services utiliza o protocolo SOAP (simple object access protocol).

Exercícios

[07] CESPE – 2010 – TRE-MT

Com relação a web services, assinale a opção correta.

a) As arquiteturas de aplicação de web services são arquiteturas firmemente acopladas, nas quais as ligações entre serviços não podem mudar durante a execução.

b) SOAP (Simple Object Access Protocol) é um protocolo com base em HTML que permite troca de informações entre aplicações em um ambiente distribuído.

c) UDDI (Universal Description, Discovery and Integration) é um diretório para armazenamento de informações a respeito de web sevices. Essas informações são descritas em SOAP.

d) A linguagem WSDL (Web Services Description Language) é utilizada para descrever web services.

e) Segundo o W3C (World Wide Web Consortium), web services são apropriados somente para aplicações em que componentes de um sistema distribuído são executados em plataformas semelhantes de um mesmo fornecedor.

Exercícios

[08] FUMARC – 2011 – BDMG

Em relação aos padrões utilizados nas arquiteturas orientadas a serviços, analise os itens a seguir, marcando com **(V)** a assertiva verdadeira e com **(F)** a assertiva falsa.

( ) WSDL é um padrão de linguagem utilizado para definir fluxos de execução envolvendo serviços distintos dentro de um processo.

( ) SOAP é um protocolo que define um padrão para troca de mensagens para dar suporte à comunicação entre serviços.

( ) UDDI fornece informações sobre provedores de serviços, os serviços fornecidos por um provedor e a localização da descrição dos serviços.

( ) BPEL é um padrão de linguagem para definição de serviços Web que descreve as interfaces oferecidas pelos serviços.

Assinale a opção com a sequência **CORRETA**, de cima para baixo.

a) F, V, V, F.
b) V, V, V, F.
c) F, V, F, V.
d) V, F, F, V.

Exercícios

[09] FCC – 2007 – TRT 23ª

Na pilha de protocolos básicos dos Web Services representa-se de baixo para cima nas camadas dois, três e quatro, respectivamente, os protocolos

a) HTTP, FTP e SMTP e na primeira camada os protocolos internet, tais como SOAP e UDDI.

b) SOAP, WSDL e UDDI e na primeira camada os protocolos internet, tais como HTTP e FTP.

c) UDDI, FTP e HTTP e na primeira camada os protocolos internet, tais como SOAP e WSDL.

d) WSDL, SOAP e UDDI e na primeira camada os protocolos internet, tais como SMTP e FTP.

e) UDDI, WSDL e SMTP e na primeira camada os protocolos internet, tais como SOAP e FTP.

Padrões web services

1ª geração
- SOAP 1.1 (Note) e SOAP 1.2 (Rec)
- WSDL 1.1 (Note) e WSDL 2.0 (Rec)
- UDDI 2.0 (Spec) e UDDI 3.0 (Draft)

2ª geração (WS-*)

novos requisitos

- segurança

  WS-Security, XML Signature, XML Encryption, WS-Trust

- Políticas

  WS-Policy, WS-PolicyAssertions

- processos de negócio

  WS-BPEL, WS-CDL

SOAP: Simple Object Access Protocol

- SOAP 1.1: W3C Note (maio de 2000)
- SOAP 1.2

Jul 2001: W3C Working Draft

Out 2001: W3C Working Draft
Part 1: Messaging Framework
Part 2: Adjuncts

Dez 2001: W3C Working Draft
Part 0: Primer
Part 1: Messaging Framework
Part 2: Adjuncts

Jun 2002: W3C Working Draft: Assertions and Test Collection

Jun 2003: W3C Recommendation

Abr 2007: W3C Recommendation Second Edition
Part 0: Primer
Part 1: Messaging Framework
Part 2: Adjuncts
Assertions and Test Collection

SOAP 1.1

- protocolo de comunicação W3C
  - baseado em XML
  - formato para intercâmbio de mensagens
  - usado para encapsulamento/transporte de dados via HTTP
  - transparente a firewalls

  ** recomendação W3C = SOAP 1.2
- Sintaxe (RFC 2119)
  - **deve** indicar namespaces de envelope e encoding (desambiguação)
  - **não deve** ter referências DTD (<!DOCTYPE>)
  - **não deve** ter instruções de processamento

```
<?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">
    ...
```

http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/encoding/

SOAP 1.1


schemas.xmlsoap.org/soap/envelope/
▼<!--
Schema for the SOAP/1.1 envelope
Portions © 2001 DevelopMentor.
© 2001 W3C (Massachusetts Institute of Technolat


www.w3.org/2001/12/soap-envelope
▼<!--
Schema defined in the SOAP Version 1.2 Part 1 specification
17 December 2001 Working Draft:
http://www.w3.org/TR/2001/WD-soap12-part1-20011217/
$Id: soap-envelope.xsd,v 1.1 2001/12/14 13:35:22 ylafon Exp $
Copyright 2001 W3C (Massachusetts Institute of Technology,
Institut National de Recherche en Informatique et en Automatique,
Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
-->

SOAP 1.1

SOAP: Elementos

- Envelope (obrigatório)
  - identifica o documento XML como SOAP
  - delimita início e fim da mensagem SOAP
- Header (opcional)
  - atributos para o processamento
  - ponto intermediário ou final
- Body (obrigatório)
  - encapsula o payload SOAP
  - requisições e respostas (serviço)
  - chamadas a procedimentos
  - operações UDDI
- Fault (opcional)
  - informações e status de erros

```
<?xml version="1.0"?>
<soap:Envelope

    <soap:Header>
        ...
    </soap:Header>

    <soap:Body>
        ...
        <soap:Fault>
            ...
        </soap:Fault>

    </soap:Body>

</soap:Envelope>
```

SOAP 1.1

SOAP: Envelope (obrigatório)

- atributos
  - namespace (`xmlns:soap`)
    - versionamento
    - soap 1.1: `http://schemas.xmlsoap.org/soap/envelope`
  - encodingStyle
    - define os tipos de dados
    - pode ser usado para indicar regras de serialização
      - pode aparecer em qualquer elemento
      - não há default

    W3Schools:
      - default: `http://www.w3.org/2001/12/soap-encoding`
      - não há default

SOAP 1.1

SOAP: Header (opcional)

- informações específicas para a aplicação
- header entries
  - qualificados com namespace
- atributos
  - mustUnderstand
  - actor
  - encodingStyle
- mustUnderstand
  - indica a obrigatoriedade da entrada
  - boolean: 0|1 (default 0)
- actor
  - indica o destinatário da entrada (final ou intermediário)
  - sintaxe: `soap:actor="URI"`

SOAP 1.1

SOAP: Body (obrigatório)

- contém a mensagem SOAP
- uso
  - empacotar chamadas RPC (marshalling)
  - reportar erros
  - enviar operações UDDI
- Body entries
  - Fault

```
<soap:Body>
  <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
    <m:Item>Apples</m:Item>
  </m:GetPrice>
</soap:Body>
```

```
<soap:Body>
  <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
    <m:Price>1.90</m:Price>
  </m:GetPriceResponse>
</soap:Body>
```

SOAP 1.1

SOAP: Fault (opcional)

- indica mensagens de erro
- sub-elementos
  - faultcode (obrigatório)
  - faultstring (obrigatório)
  - faultactor
  - detail
- FaultCodes
  - VersionMismatch
  - MustUnderstand
  - Client
  - Server

```
<s:Envelope xmlns:s="...">
   <s:Body>
      <s:Fault>
       <faultcode>Client.Authentication</faultcode>
       <faultstring>
          Invalid credentials
       </faultstring>
       <faultactor>http://acme.com</faultactor>
       <details>
          <!-- application specific details -->
       </details>
      </s:Fault>
   </s:Body>
</s:Envelope>
```

SOAP 1.1

SOAP HTTP Binding

- HTTP empacota mensagem SOAP
  - HTTP + XML = SOAP
  - HTTP request/response
  - SOAP pode utilizar métodos HTTP (get, post, etc)
- HTTP POST
  - Headers
    - Content-Type (MIME/Media Type)
      - W3Schools: **`application/soap+xml (RFC 3902)`**
      - Note: **`text/xml (RFC 2376)`**
    - Content-Length
    - SOAPAction (Note)
      - obrigatório para SOAP requests

SOAP 1.1

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction:
"http://electrocommerce.org/abc#MyMessage"

<SOAP-ENV:Envelope
 xmlns:SOAP-ENV=
 "http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle=
 "http://schemas.xmlsoap.org/soap/encoding/"/>

  <SOAP-ENV:Body>
    <m:GetLastTradePriceDetailed
     xmlns:m="Some-URI">
      <Symbol>DEF</Symbol>
      <Company>DEF Corp</Company>
      <Price>34.1</Price>
    </m:GetLastTradePriceDetailed>
  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn


<?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>
```

SOAP 1.2

- Duas edições de recomendações
- consiste de 3 partes
  - 0: Primer
    - não normativo
    - tutorial
  - 1: Messaging Framework
    - kernel do protocolo
    - define
      elementos,
      regras para processamento
      protocolos de ligação
  - 2: Adjuncts
    - complementar
    - define
      - modelo de dados e esquemas de encoding

SOAP 1.2

- prevê o uso de XML Infoset
  - recomendação W3C
  - modelo dados XML (abstrato)
- Envelople

  `http://www.w3.org/2003/05/soap-envelope`
- EncodingStyle
  - proibido para elemento envelope
  - sub-elementos de Header, Body e Detail

  `soap-encoding` e `soap-envelope/encoding/none`
- Header Blocks (header entry)
  - atributos MustUnderstand (true | false)
  - elemento NotUnderstood
- proíbe elementos após o <**Body**>
  - SOAP 1.1 permitia sem definir no esquema
  - SOAP 1.2: envelope só tem dois filhos: header e body

SOAP 1.2

- substitui atributo **actor** por **role**
  - next, ultimateReceicer e none

    `role="http://www.w3.org/2003/05/soap-envelope/role/next"`
- elemento **Fault**
  - pode aparecer

    dentro de Header Blocks ou de filhos de Body

    escopo semântico diferente

  - Sub-elementos (namespace)
    - obrigatórios
      - **code** (faultcode)
      - **reason** (faultstring)
    - opcionais
      - **node** e **role**
      - **detail**

SOAP 1.2

- elemento **Fault**
  - Fault Codes
    - VersionMismatch
    - MustUnderstand
    - DataEncodingUnknown
    - Sender
    - Receiver

```
<env:Body>
   <env:Fault>
      <env:Code>
         <env:Value>env:Sender</env:Value>
         <env:Subcode>
            <env:Value>m:MessageTimeout</env:Value>
         </env:Subcode>
      </env:Code>
      <env:Reason>
         <env:Text xml:lang="en">Sender Timeout</env:Text>
      </env:Reason>
      <env:Detail>
         <m:MaxTime>P5M</m:MaxTime>
      </env:Detail>
   </env:Fault>
</env:Body>
```

SOAP 1.2

- Binding
- SOAP Protocol Binding Framework (parte 1: abstrato)
- SOAP HTTP Binding (parte 2: concreta)
  - media type: “application/soap+xml” (RFC 3902)

    ```
    application/soap+xml; charset=utf-8; action="uri"
    ```

  - padrões de troca de mensagens (obrigatórios)
    - SOAP Request-Response MEP
    - SOAP Response MEP
  - recursos (obrigatórios)
    - Web Method

      POST: Request-Response

      GET: Response
    - SOAP Action (opcional)

      ```
      application/soap+xml; action="uri"
      ```

SOAP 1.2

- Binding

- Suporte a HTTP GET
  - padrão de troca de mensagens
    - SOAP Response MEP

```
GET /itinerary?reservation=1234567890 HTTP/1.1
Host: travelcompany.com
Accept: application/soap+xml
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml
Content-Length: nnnn
<?xml version="1.0"?>
<e:Envelope xmlns:e='...'>
    <e:Body>
        <f:itinerary xmlns:f='...'>
            <f:itinid>1234567890</f:itinid>
            ...
        </f:itinerary>
    </e:Body>
</e:Envelope>
```

SOAP 1.2

	SOAP 1.1	W3Schools	SOAP 1.2
	note	draft 1.2	recommendation
<Envelope>			
namespaces (envelope e encoding)	http://schemas.xmlsoap.org/soap	http://www.w3.org/2001/12/	http://www.w3.org/2003/05/
elementos de envelope	Header, Body e outros		Header e Body apenas
Encoding	qualquer elemento não há default	qualquer elemento há e não há default	filhos de Header, Body e Detail não há default
<Header>			
MustUnderstand	0 ou 1	0 ou 1	true ou false header NotUnderstood
actor/roles	actor: Next		roles: Next, None, UltimateReceiver

SOAP 1.2

	SOAP 1.1	W3Schools	SOAP 1.2
<Body>			
elementos de fault	e:Fault faultcode faultstring faultactor detail	Fault faultcode faultstring faultactor detail	e:Fault e:Code, e:Subcode, e:Value e:Reason e:Node, e:Role e:Detail, e:Text
fault codes	VersionMismatch MustUnderstand Client Server	VersionMismatch MustUnderstand Client Server	VersionMismatch MustUnderstand Sender Receiver DataEncodingUnknown
estrutura de fault codes	notação de ponto Server.Memory		formato XML-like Code Value Receiver Subcode Value Memory
Binding			
Content-Type	text/xml	application/soap+xml	application/soap+xml
HTTP SOAP request	obrigatório SOAPAction HTTP Header		opcional atributo do media type
HTTP Binding	HTTP POSTt (request-response)	HTTP POSTt (request-response)	HTTP POST (request-response) HTTP GET (response)

Exercícios

[10] FCC – 2010 – TRE-RS

SOAP é

I. um protocolo de comunicação leve destinado a troca de informações estruturadas em um ambiente distribuído.

II. dependente de plataforma e linguagem de programação.

III. baseado em XHML.

IV. uma recomendação da W3C.

Está correto o que consta em

a) III e IV, somente.

b) II e III, somente.

c) I e IV, somente.

d) II, III e IV, somente.

e) I, II, III e IV.

Exercícios

[11] FCC – 2008 – MPE-RS

A identificação do documento XML, como uma mensagem SOAP, está contida no elemento da estrutura SOAP denominado

a) root.

b) body.

c) envelope.

d) fault.

e) header.

Exercícios

[12] CESGRANRIO – 2006 – EPE

Sobre os Serviços Web são feitas as seguintes afirmativas.

I – O SOAP pode ser transportado por protocolos como HTTP, SMTP e JMS.

II – Uma mensagem SOAP permite encapsular chamadas RPC.

III – Uma mensagem SOAP é um documento XML que pode conter três partes: o envelope, o cabeçalho e o corpo.

Está(ão) correta(s) a(s) afirmativa(s):

a) I, apenas.
b) II, apenas.
c) III, apenas.
d) I e II, apenas.
e) I, II e III.

Exercícios

[13] CESPE – 2013 – ANP

Acerca do SOAP (simple object access protocol) e web services, julgue os itens subsequentes.

[61] Na versão SOAP 1.2 não é permitido qualquer elemento após a tag body.

[62] Em uma mensagem SOAP que possui o atributo env:mustUnderstand=true no cabeçalho, o bloco deve ser processado de acordo com as especificações constituintes do SOAP.

Exercícios

[14] FCC – 2013 – TRT/PR

SOAP é um protocolo de acesso a um Web Service, baseado em XML e recomendado pela W3C, que permite que aplicativos troquem informações usando HTTP. Define um formato para envio de mensagens. Uma mensagem SOAP é um documento XML comum contendo diversos elementos, como o exemplo a seguir:

Sobre SOAP é correto afirmar que

a) o elemento Header é um elemento obrigatório que contém informações específicas sobre a mensagem SOAP.

b) uma mensagem SOAP pode conter um ou mais elementos Fault que são opcionais e usados para indicar mensagens de erro.

c) o atributo encodingStyle do elemento Envelope é utilizado para definir os tipos de dados utilizados no documento. Este atributo pode aparecer apenas no elemento Envelope.

d) o xmlns:soap namespace deve sempre ter o valor "http://www.w3.org/2001/12/soap-envelope". Se um namespace diferente for usado, o aplicativo gera um erro e descarta a mensagem.

e) se houver um elemento Fault na mensagem, ele deve aparecer obrigatoriamente como um elemento filho do elemento Header.

Exercícios

[15] FCC – 2013 – SEFAZ – SP

Segundo o texto, a troca de mensagens entre a aplicação e o **web service** utiliza o protocolo SOAP. Uma mensagem SOAP é um documento XML que pode conter diversos elementos. Sobre esses elementos, é correto afirmar que

a) se um elemento **Fault** está presente na mensagem, ele deve aparecer como um elemento filho do elemento **Header**.

b) se o elemento **Header** estiver presente na mensagem SOAP, ele deve ser o primeiro elemento filho do elemento SOAP.

c) o elemento **Fault** é obrigatório e é usado para recuperar mensagens de erro e informações de **status** resultantes do envio da mensagem.

d) o elemento SOAP é o elemento raiz de uma mensagem, e define que o documento XML é uma mensagem SOAP.

e) um elemento **Fault** não pode aparecer mais que uma vez em uma mensagem SOAP.

Exercícios

[16] CESPE – 2010 – TCU

Considere que o líder tenha solicitado a um programador do projeto que comentasse o seguinte trecho de código.

```
POST /objectURI HTTP/1.1
Host: www.foo.com
SOAPMethodName: urn:develop-com:IBank#getBalance
Content-Type: text/xml
Content-Length: 1234
```

O comentário do programador teria sido correto se ele dissesse que esse código é provavelmente o esqueleto de um pedido http que foi invocado sobre o servidor http (*hypertext transfer protocol*) de endereço www.foo.com, embasado no modelo de comunicação SOAP (simple object access protocol), que apenas o cabeçalho do pedido está sendo apresentado e que o pedido completo deve possuir em seu corpo um documento XML com 1.234 *bytes* de tamanho.

WSDL

WSDL (Web Services Description Language)

- Evolução
  - WSDL 1.1: W3C Note – Mar/2001
  - WSDL 1.2: W3C Working Drafts
    - Jul/2002: documento único
    - Jun/2003
      - Part 1: Core Language
      - Part 2: Message Patterns
      - Part 3: Bindings
  - WSDL 2.0
  - nov/2003: renumerada de 1.2 para 2.0
  - jun/2007: Recommendation
    - Part 0: Primer (não-normativa)
    - Part 1: Core Language
    - Part 2: Adjuncts

WSDL

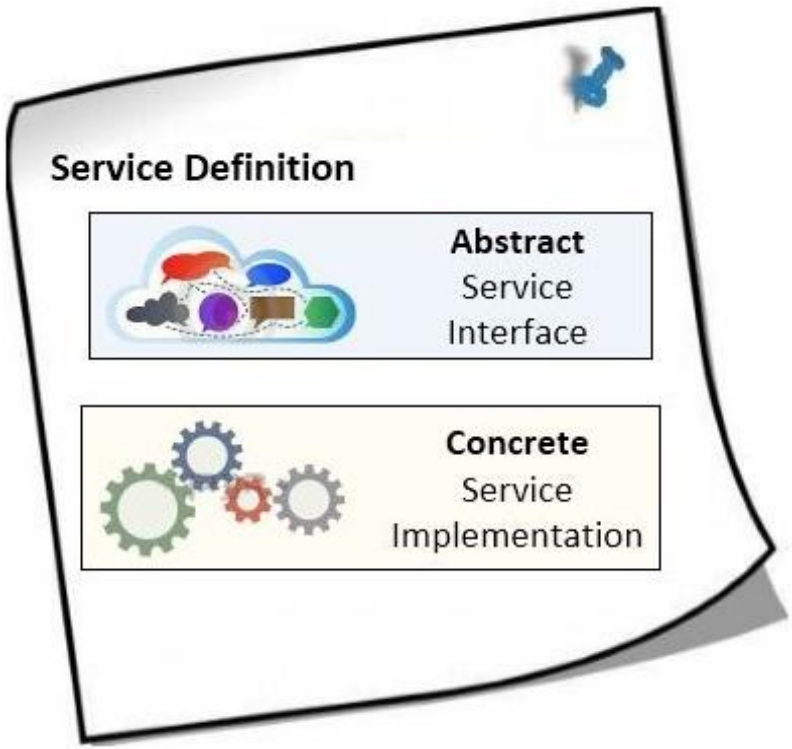
Definições

- WSDL 1.1
  definição baseada nos elementos WSDL
  descrever serviços de rede

- W3Schools
  - descrever Web Services e como acessá-los
  - descreve como acessar um serviço
  - usado para localizar serviços web
- WSDL 2.0
  - descrever serviços web
  - foco na separação de preocupações (abstrato x concreto)

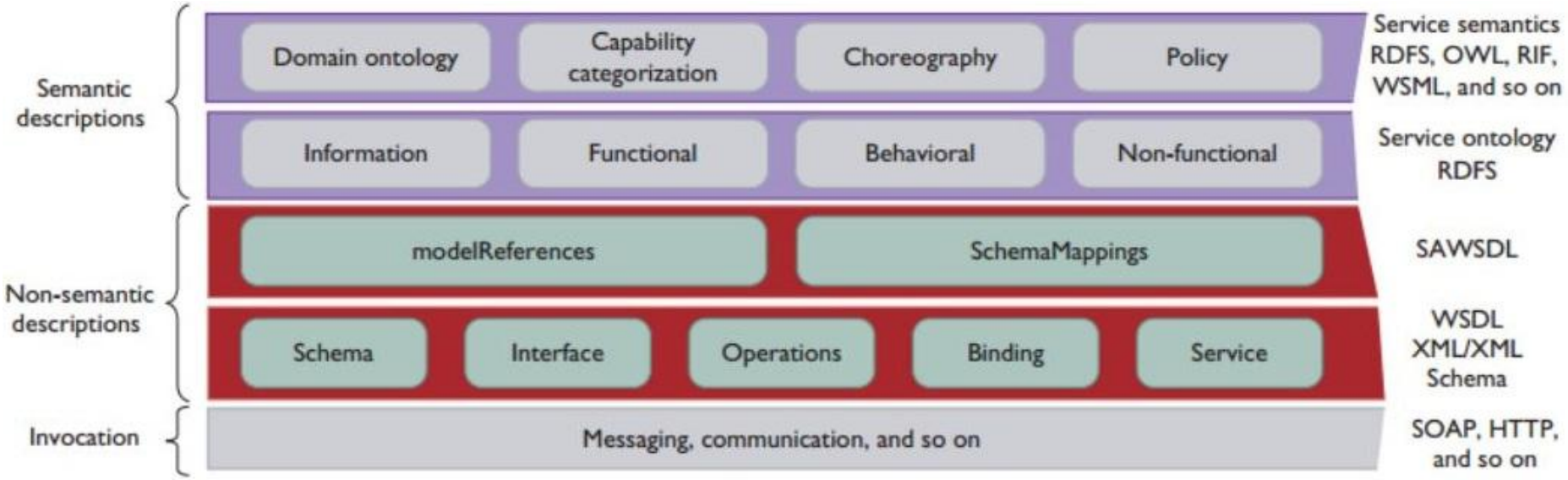
WSDL

- descreve o serviço sob duas perspectivas
  - reusabilidade e independência de preocupações (concern)
  - **abstrata** (definição)
    o que o serviço faz
    operações, entradas e saídas
  - **concreta** (implementação)
    forma de comunicação e conexão



WSDL

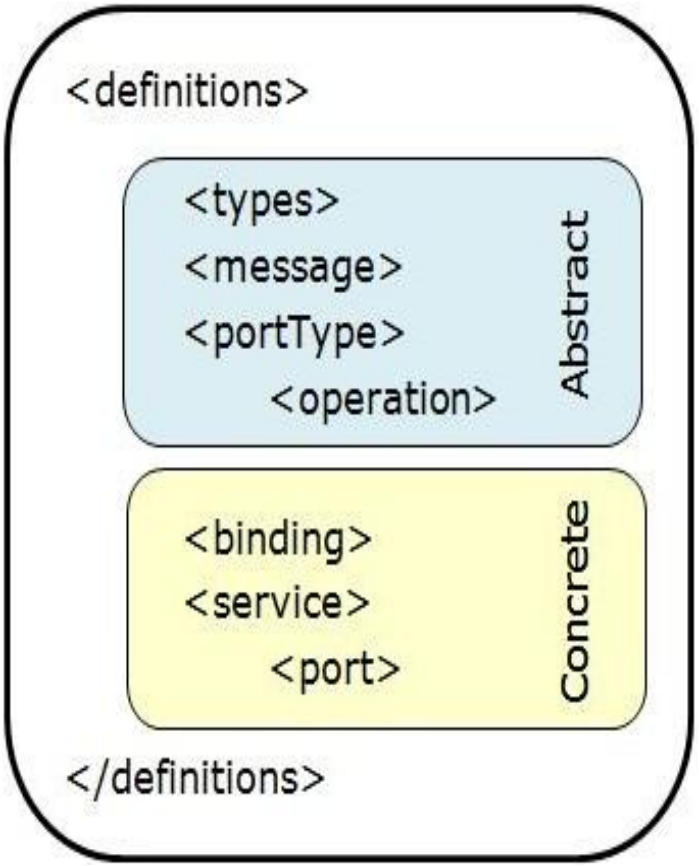
- Semântica
  - WSDL = estrutura sintática
  - não define características não funcionais
    desempenho, segurança, QoS, etc.
  - Semantic Annotations for WSDL (SAWSDL)
    - conjunto de atributos
      modelReference, SchemaMapping(2) e attrExtensions



WSDL 1.1 – Elementos (Note)

- definitions
  delimita o início e fim de um WSDL

  - Types
  - Message
  - Operation
  - Port Type
  - Binding
  - Port
  - Service



WSDL 1.1 – Elementos (Note)

- types
  definições de tipo de dados (XSD)
- message
  dados a serem transmitidos
  consiste de elementos lógicos
    <part>
      name, element e type
    associados a <types> ou XSD
  tipos
    input (parâmetros)
    output (retorno)

```
<message name="SayHelloRequest">
  <part name="firstName"
        type="xsd:string"/>
</message>

<message name="SayHelloResponse">
  <part name="greeting"
        type="xsd:string"/>
</message>
```

WSDL 1.1 – Elementos (Note)

- portType
  conjunto de operações
  container
- operation
  descrição abstrata de uma ação
  funcionalidades
- patterns (primitivas)
  - **one-way (input)**
  - **request-response**
    input, ouput e fault
  - solicit-response
    ouput, input e fault
  - notification (output)

```
<portType name="Hello_PortType">
  <operation name="sayHello">
    <input message="tns:SayHelloRequest"/>
    <output message="tns:SayHelloResponse"/>
  </operation>
</portType>
```

WSDL 1.1 – Elementos (Note)

- binding
  transmissão do portType
  - soap:binding
    protocolo de comunicação
    atributos
      transport (obrigatório)
      style: RPC ou Document
  - soap:operation
    ligação entre operação e SOAP
    atributos
      style e soapAction
  - soap:body
    ligação <parts> e <Body>
    atributos
      use (obrigatório)
      parts
      namespace e encodingStyle

```
<binding name="Hello_Binding"
         type="tns:Hello_PortType">
  <soap:binding style="rpc|document"
     transport="URI"/>
  <operation name="sayHello">
    <soap:operation
         soapAction="sayHello"/>
    <input>
      <soap:body use="encoded"
         encodingStyle="URI"/>
    </input>
    <output>
      <soap:body use="encoded"
         encodingStyle="URI"/>
    </output>
  </operation>
</binding>
```

WSDL 1.1 – Elementos (Note)

- service
  coleção de portas suportadas pelo serviço
  uma porta por protocolo
- port
  define um endpoint
  binding + endereço de rede
  atributos
    - name e binding
  sintaxe
    - um endereço por porta
    - apenas endereço
- soap:address
  informações de endereço
- documentation
  legível por humano

```
<service name="Hello_Service">

  <documentation>
        WSDL File for HelloService
  </documentation>

  <port binding="tns:Hello_Binding"
        name="Hello_Port">
    <soap:address
        location=
        "http://www.examples.com/SayHello/">
  </port>
</service>
```

WSDL 1.1 – Elementos (Note)

```
<?xml version="1.0"?>
<definitions name="StockQuote"

    <types>
        <schema targetNamespace="http://example.com/stockquote.xsd"
            xmlns="http://www.w3.org/2000/10/XMLSchema">
            <element name="TradePriceRequest">
                <complexType>
                    <all>
                        <element name="tickerSymbol" type="string"/>
                    </all>
                </complexType>
            </element>
            <element name="TradePrice">
                <complexType>
                    <all>
                        <element name="price" type="float"/>
                    </all>
                </complexType>
            </element>
        </schema>
    </types>

    <message name="GetLastTradePriceInput">
        <part name="body" element="xsd1:TradePriceRequest"/>
    </message>

    <message name="GetLastTradePriceOutput">
        <part name="body" element="xsd1:TradePrice"/>
    </message>
```

WSDL 1.1 – Elementos (Note)

```
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns: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>

<service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>

</definitions>
```

WSDL 1.1 – Elementos (W3Schools)

- elementos principais
  - <types>
  - <message>
  - <portType>
  - <binding>
- não apresenta
  - <port>
  - <service>
- portType
  - elemento mais importante
  - descreve:
    - serviço web
    - operações
    - mensagens
  - <operation>

```
<definitions>

   <types>
      ....
   </types>

   <message>
      ....
   </message>

   <portType>
      ....
   </portType>

   <binding>
      ....
   </binding>

</definitions>
```

WSDL 2.0

- recomendação
  - três documentos
- suporte a IRI (RFC 3987)
  - internacionalização
  - URI: alfabeto inglês
  - IRI: vários idiomas
- <description>
  - definition (wsdl 1.1)
  - componentes de alto nível
    - WSDL 2.0
      interface, binding e service
    - type system
      Element Declarion
      Type Definition (obrigatório)
  - atributo
    targetNamespace (requerido)

```
<description
    targetNamespace="xs:anyURI">
    <documentation />
    [<import /> | <include />]
    <types />
    [<interface />
    <binding />
    <service />]
</description>
```

WSDL 2.0 – Elementos

- <types>
- <message>
  - removido
  - definido em operation
- <interface>
  - portType (WSDL 1.1)
  - atributos
    - styleDefault
      - RPC
      - IRI
      - Multipart
    - extends (herança)

```
<interface
  name="xs:NCName"
  extends="xs:QName"
  styleDefault="xs:anyURI">
  <documentation />
  [<fault /> | <operation /> ]
</interface>
```

WSDL 2.0 – Elementos

- <operation>
  - atributos
    - pattern (requerido)
      - in-only
      - robust in-only
      - in-out
    - style
      - RPC
      - IRI
      - Multipart
  - elementos
    - input e output
      - messageLabel
      - element
    - infault e outfault
      - ref (requerido)
      - messageLabel

```
<operation name="opCheckAvailability"
  pattern="http://www.w3.org/ns/wsdl/in-out"
  style="http://www.w3.org/ns/wsdl/style/iri"

  <input messageLabel="In"
    element="ghns:checkAvailability" />

  <output messageLabel="Out"
    element="ghns:checkAvailabilityResponse" />

  <infault ref="tns:invalidDataFault"
    messageLabel="In"/>

  <outfault ref="tns:invalidDataFault"
    messageLabel="Out"/>
</operation>
```

WSDL 2.0 – Elementos

- <binding>
  mensagem e protocolo
  atributos
    - interface
    - type (requerido)
  WSDL 1.1
    binding e soap:binding
- <operation>
    formato da mensgem
    atributo
      ref (requerido)
    elementos
      input e output
      infault e outfault
- extensões
    whttp e wsoap

```
<binding name="reservationHTTPBinding"
  interface="tns:reservationInterface"
  type="http://www.w3.org/ns/wsdl/http">

  <operation ref="tns:opCheckAvailability"
    whttp:method="GET"/>
</binding>

<binding name="reservationSOAPBinding"
  interface="tns:reservationInterface"
  type="http://www.w3.org/ns/wsdl/soap"
  wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">

  <operation ref="tns:opCheckAvailability"
   wsoap:mep="http://www.w3.org/2003/05/soap/mep/request-response"/>
</binding>
```

WSDL 2.0 – Elementos

- <service>
  container para endpoints
  atributos
  - interface (requerido)
- <endpoint>
  WSDL 1.1: <port>
  endereço do serviço
  atributos
  - binding(requerido)
  - address (opcional)

```
<service name="reservationService"
  interface="tns:reservationInterface">

  <endpoint name="reservationHttpEndpoint"
    binding="tns:reservationHTTPBinding"
    address="http://www.example.com/rest/checkAvailability"/>

  <endpoint name="reservationSoapEndpoint"
    binding="tns:reservationSOAPBinding"
    address="http://www.example.com/soap/checkAvailability"/>

</service>
```

WSDL

WSDL 1.1	W3Schools	WSDL 2.0
<definitions>	<definitions>	<description>
<types>	<types>	<types>
<message>	<message>	- x -
<portType>	<portType>	<interface>
<operation>	<operation>	<operation>
<binding>	<binding>	<binding>
<service>		<service>
<port>		<endpoint>

Exercícios

[17] FCC – 2008 – MPE-RS

NÃO é uma informação requerida para invocar um serviço de Web e encapsulada pelo WSDL na forma de um documento XML:

a) O local do serviço.

b) As operações que o serviço apoia.

c) Os parâmetros que o serviço espera.

d) Os detalhes das mensagens do serviço.

e) Os meios para publicar e localizar o serviço.

Exercícios

[18] CESPE - 2008 - STJ

Acerca de SOA e serviços web, julgue os próximos itens.

O WSDL separa a parte abstrata de uma descrição de serviço da parte concreta; nessa descrição, a parte concreta contém as definições de tipos usados pelo serviço e a parte abstrata especifica como e onde o serviço pode ser contatado. Os documentos WSDL podem ser acessados via um serviço de diretório como o UDDI; as definições WSDL podem ser geradas a partir de definições de interfaces escritas em outras linguagens.

Exercícios

[19] CESPE – 2011 – STM

A respeito de SOA, de *web services* e do modelo de acessibilidade do governo eletrônico, julgue os itens subsequentes.

Na especificação WSDL 2.0, são propriedades de um componente description: portType, bindings, services, elemento declarations e type definitions. A única propriedade obrigatória é services.

Exercícios

[20] CESPE – 2010 – TCU

Considere que o líder da equipe solicite a um programador do projeto que analise o seguinte trecho de código de um documento XML.

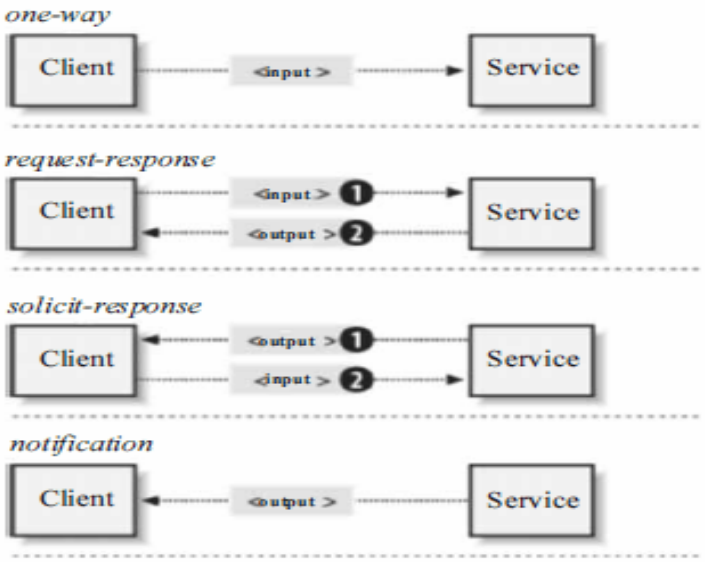
```
<service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
```

Nessa situação, se o programador disser que esse trecho de documento é provavelmente de declaração de serviço web com base na tecnologia WSDL (web services description language) e que, embora o serviço descrito contenha uma única porta, é possível a definição de várias portas associadas a um mesmo serviço, a análise feita deverá ser considerada correta.

Exercícios

[21] CESPE – 2009 – TCU

Considerando a figura, julgue o item.



Na figura mostrada, na notification, o serviço envia uma mensagem e a operação tem um único elemento de saída. O padrão request-response é pouco utilizado nos serviços SOAP.

Exercícios

[22] CESPE – 2009 – TCU

Em WSDL, os elementos do tipo types descrevem todos os tipos de dados usados entre cliente e servidor. O WSDL está exclusivamente ligado a um sistema de tipagem específico pois utiliza, como padrão, um esquema de especificação W3C XML.

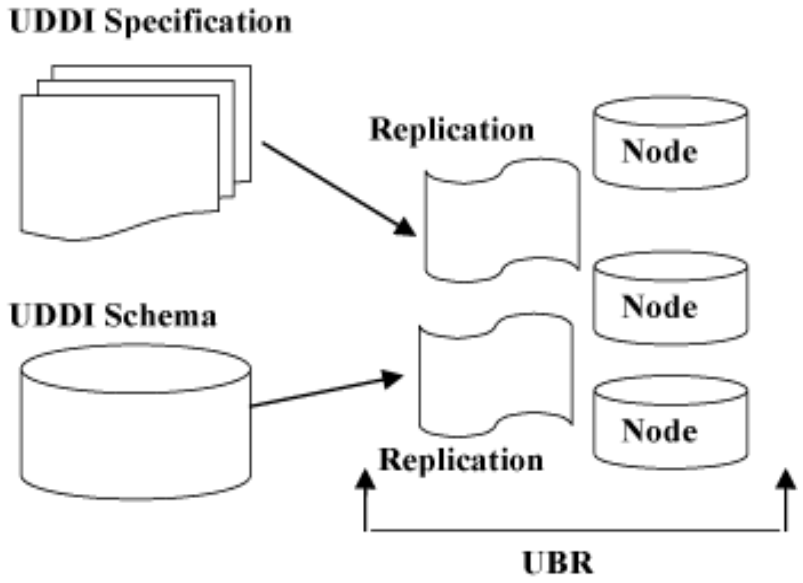
UDDI

- Universal Description, Discovery and Integration
  - serviço de descoberta
    - modelo triangular
    - 1ª etapa: entidades se tornam conhecidas
- padronizado pela OASIS
  - UDDI 2.03 (standard 2002)
  - UDDI 3.0.2 (draft 2004) (standard 2005)
- definição
  - padrão para publicar e descobrir serviços web
  - suporta descrição e descoberta
    - negócios, organizações e provedores
    - serviços e interfaces
- W3Schools

  descrever serviços, descobrir negócios e integrar serviços de negócios

UDDI

- framework
  - data model
    - estruturas de dados XML Schema
  - API specification
    - operações
      - publicação
      - procura
  - Cloud services (UBR)
    - repositório
    - diretório
      - logica centralizada
      - nós físicos distribuídos
    - site (nó) operador
      - implementa UBR
      - custodian



UDDI Technical Architecture

UDDI

- API specification

  funções CRUD para UDDI

  encapsuladas por SOAP

  - **Publisher Interface**

    função: Publish

    16 operações

    save_business, delete_service, etc.

  - **Inquiry Interface**

    função: find

    10 operações

    find_business, get_serviceDetail, etc.

- Interface web

  IBM e Microsoft

  ambiente de teste e produção

  https://www.ibm.com/services/uddi/protect/publish

  http://www-3.ibm.com/services/uddi

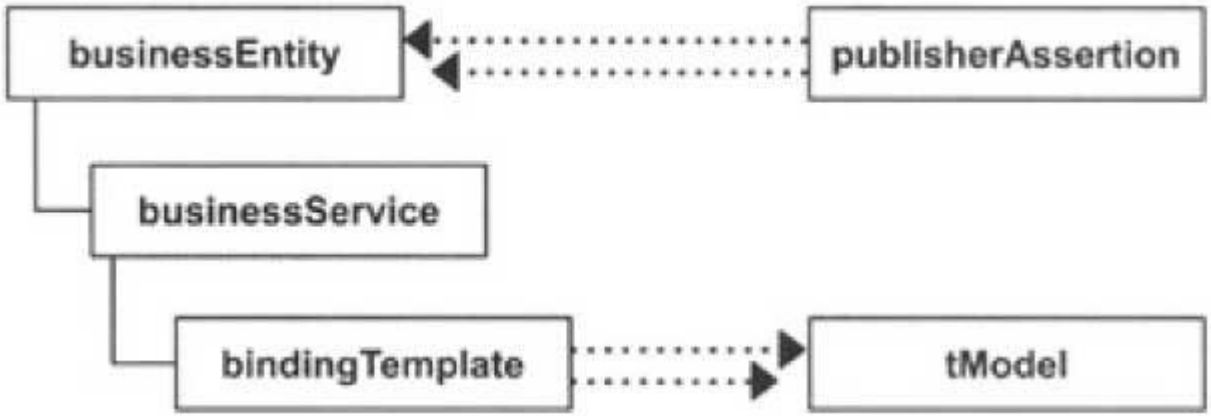
UDDI

```
POST /save_business HTTP/1.1
Host: www.XYZ.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "save_business"
<?xml version="1.0" encoding="UTF-8" ?>

<Envelope
xmlns="http://schemas/xmlsoap.org/soap/envelope/">
  <Body>
    <save_business generic="2.0"
    xmlns="urn:uddi-org:api_v2">
      <businessKey="" />
      <name>XYZ, Pvt Ltd.</name>
      <description>
        Company is involved in...
      </description>
      <identifierBag />
    </save_business>
  </Body>
</Envelope>
```

```
POST /get_businessDetail HTTP/1.1
Host: www.XYZ.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "get_businessDetail"
<?xml version="1.0" encoding="UTF-8" ?>

<Envelope
xmlns="http://schemas/xmlsoap.org/soap/envelope/">
  <Body>
    <get_businessDetail generic="2.0"
    xmlns="urn:uddi-org:api_v2">
      <businessKey="C90D731D-772HSH-5303370C2" />
    </get_businessDetail>
  </Body>
</Envelope>
```

UDDI 2.03

- Data model
  - modelo de informação
  - core data types
    - businessEntity
    - businessService
    - bindingTemplate
    - tModel
    - publisherAssertion
  - descrição de serviços



UDDI 2.03

- businessEntity
  - provedor de serviços
  - dados gerais sobre o negócio
  - define categorias, listas de serviços
  - componentes
    - businessKey (requerido): UUID
    - identifierBag (D-U-N-S)
    - categoryBag (NAICS, UNSPSC, ISO 3166)
- businessService
  - serviço web individual
  - define tipo de serviço, conexão e categorias
  - componentes
    - serviceKey (requerido) (UUID)
    - businessKey
    - bindingTemplates
    - categoryBag

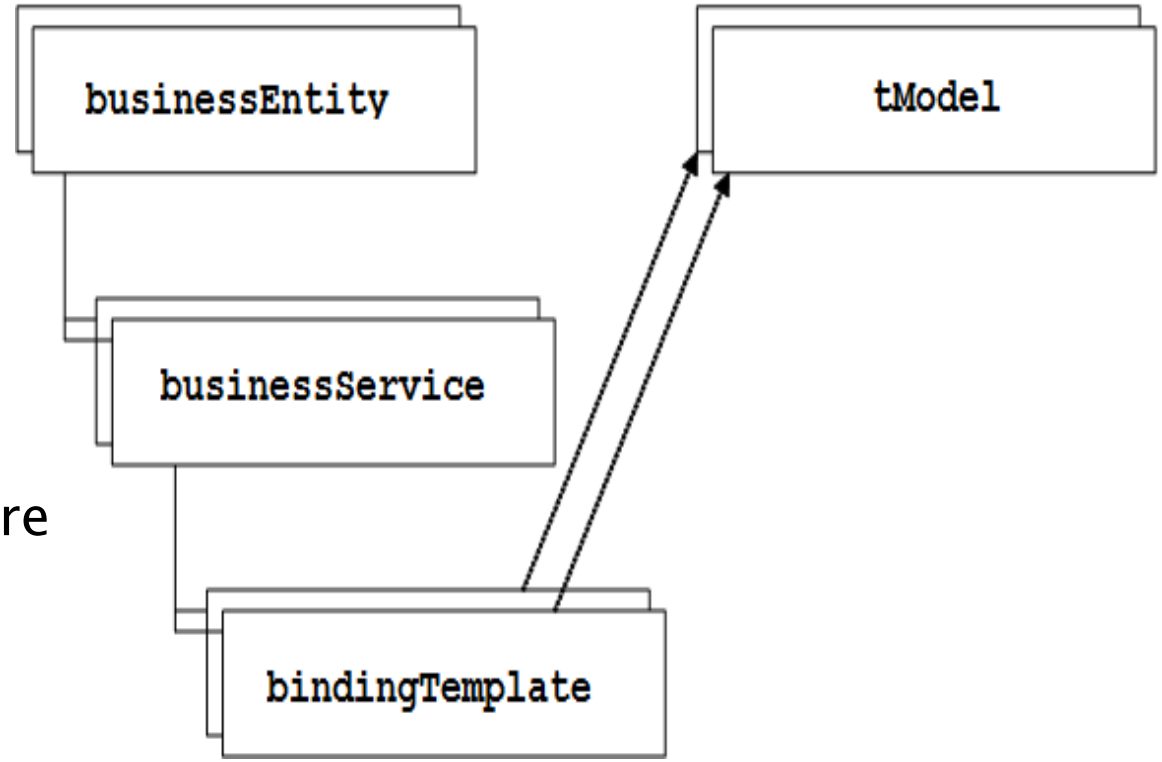
UDDI 2.03

- bindingTemplate
  - descrições técnicas
  - indica como se conectar ao serviço
  - componentes
    - bindingKey (requerido)
    - accessPoint (requerido)
    - hostingRedirector
- tModel
  - representa uma especificação técnica externa (fingerprints)
  - componentes
    - tModelKey (requerido)
    - overwiewDoc
- publisherAssertion
  - relacionamento entre businessEntity
  - componentes
    - fromKey, toKey e keyedReference – (requeridos)

UDDI 3.0.2

- core data structure
  - businessEntity
  - businessService
  - bindingTemplate
  - tModel

  publisherAssertion
  não faz parte do core

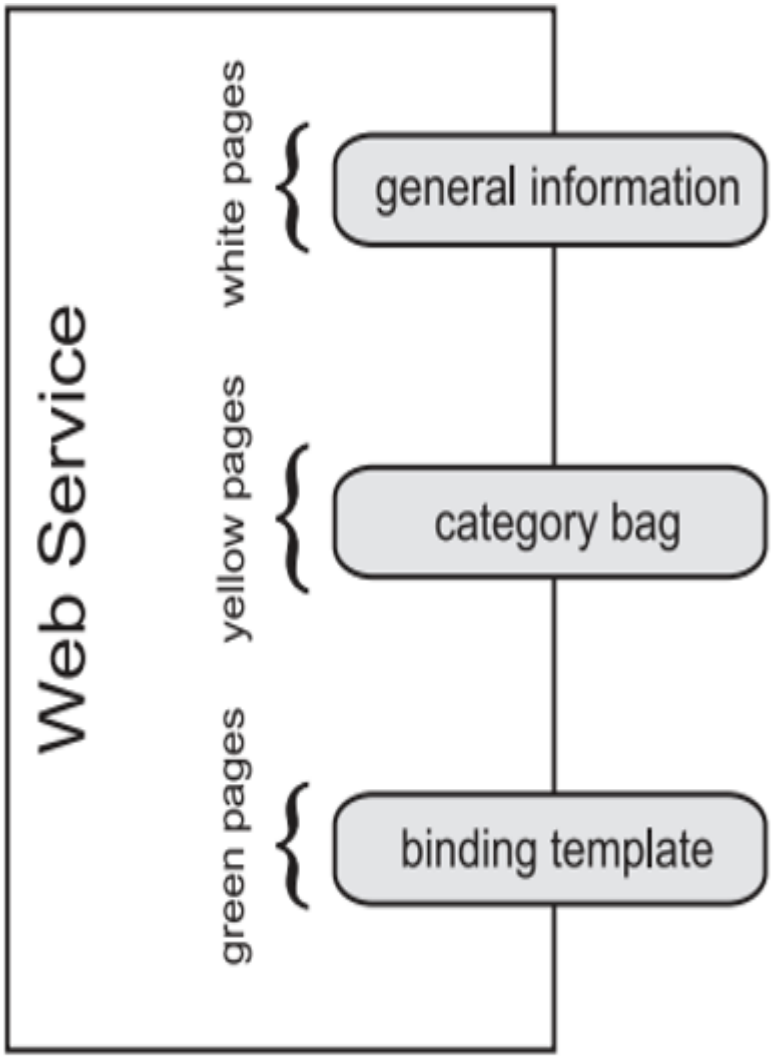


- internacionalização

- segurança
  assinatura digital
  UDDI policy
  transferência de propriedade

UDDI: Pages

- Páginas Brancas
  entidade que fornece o serviço
- Páginas Amarelas
  classificação da entidade/serviço
- Páginas Verdes
  informações técnicas sobre o serviço
- Páginas Azuis
  semântica
  WS-Specification
    <processes>
    <concepts>



Exercícios

[23] CESPE - 2008 - TRT-5

Com relação a SOA, *web services*, UDDI, WSDL, SOAP, XML, XSLT, *e-ping*, SGC, W3C e e-MAG, julgue os próximos itens.

O UDDI é uma especificação técnica que tem como objetivo descrever, descobrir e integrar *web services*; é embasado na tecnologia XML, que fornece uma plataforma neutra de dados e permite descrever relações hierárquicas de modo natural.

Exercícios

[24] CESPE – 2008 – STJ

O serviço UDDI fornece uma interface para publicar e atualizar informações acerca de serviços web; possibilita pesquisar descrições WSDL pelo nome; provê uma interface que possibilita executar consultas de modo a recuperar uma entidade que corresponda a uma chave ou recuperar entidades que correspondam a um conjunto de critérios de busca.

Exercícios

[25] ESAF – 2012 – AFC (CGU)

Assinale a opção correta.

a) A *API* de solicitação de *MDD* é usada para consultar um registrador *MDD* por condições de acesso.

b) A *APL* de atualização de *UDDI* é usada para consultar um usuário *UDDI* por informações sobre localização de uma empresa.

c) A *UDDI* é usada para manter a consistência de registradores API de propriedade de uma empresa.

d) A *API* de solicitação de *UDDI* é usada para consultar um registrador *UDDI* por informações sobre uma empresa.

e) A *API* de solicitação de *UDDL* é usada para consultar um usuário de *UDDL* por informações sobre interesses de negócio de uma empresa.

Exercícios

[26] ESAF – 2013 – AFC (STN)

As estruturas de dados da especificação UDDI são as seguintes:

a) businessEvent, businessService, blendingTransfer, tModel, publisherApproach.

b) businessEvent, userService, bindingTemplate, tModel, publisherAssertion.

c) businessEntity, businessService, bindingTemplate, tModel, publisherAssertion.

d) businessEntity, businessSite, blendingTransfer, tModel, publisherApproach.

e) businessEntity, businessSite, bindingTemplate, tModel, publicAssertion.

Exercícios

[27] (FCC – 2010 – DPE-SP) [adaptada]

Em relação à arquitetura de aplicações para o ambiente Internet, considere:

I. Na UDDI a estrutura de dados é composta por businessEntity, contendo informação sobre a organização; businessService, com a descrição do serviço; bindingTemplate, contendo a informação de como invocar o serviço; e tModel, que contem informação sobre especificações técnicas do serviço.

II. Na UDDI a informação de categoria green pages contém informação técnica sobre um Web service, geralmente incluindo um ponteiro para uma especificação externa e um endereço para invocar o serviço, que pode ser baseado em SOAP e outros.

É correto o que se afirma em

a) I e II estão corretas.
b) apenas I está correta.
c) apenas II está correta.
d) nenhuma está correta.

Exercícios

[28] CESPE – 2013 – CNJ

Acerca de interoperabilidade de sistemas, julgue os itens subsequentes.

Nos registros de negócio UDDI, a descrição da forma de acesso aos web services é um procedimento contido nas páginas verdes (green pages).

[29] CESPE – 2011 – MCE

Julgue os itens que se seguem, relativos a arquitetura de aplicações para Internet e Web, SOA e web services.

O UDDI (universal description discovery and integration), que corresponde a um registro de web services, é dividido em páginas brancas, amarelas e verdes, nas quais são prestadas aos clientes informações sobre a empresa, os serviços por ela oferecidos e as especificações WSDL desses serviços.

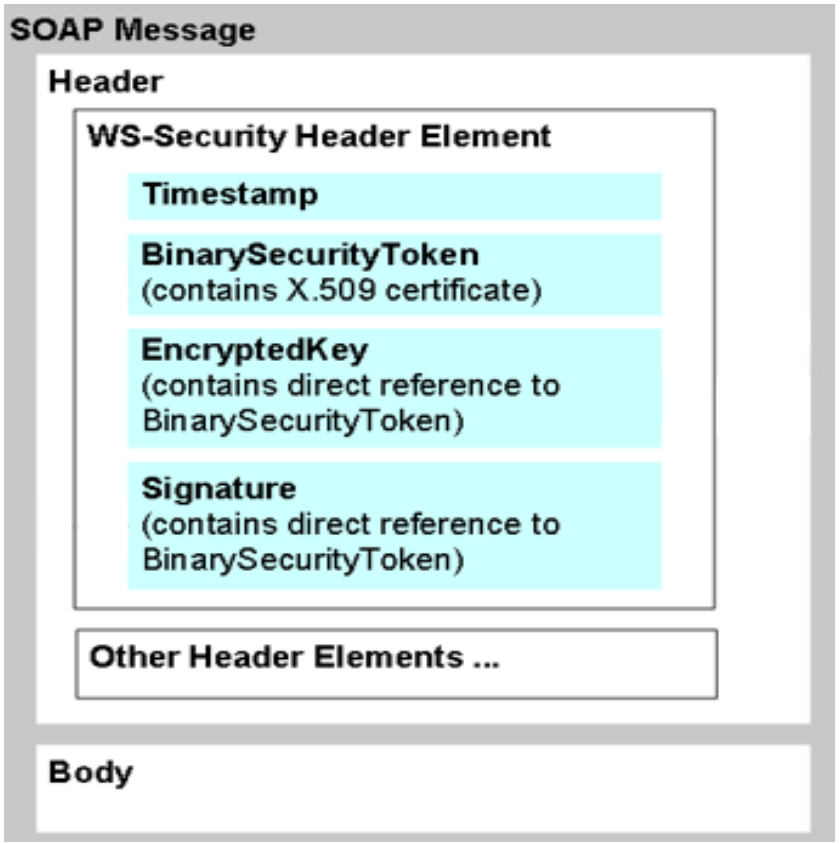
WS-Security

- Segurança para web service
  - nível de transporte
    - protege o protocolo de rede (HTTPS)
    - canal seguro com SSL
  - nível de mensagem
    - protege partes do documento XML (SOAP)

- SOAP
  - HTTP + XML = texto em claro
  - Especificação não prevê segurança
    - uso de extensões
      - sigilo
      - integridade
      - autenticação
    - extensões inseridas em <header block>

WS-Security

- WS-Secutiry (WSS)
  - parte da família WS-*
  - publicado pela OASIS
  - informa como a mensagem foi protegida
  - fornece segurança fim-a-fim
- mecanismos
  - confidencialidade
    - XML Encryption
  - integridade
    - XML Signature
  - credenciais de segurança
    - autenticidade
    - security tokens (X509, kerberos)



WS-Security: Encryption

- XML-Encryption
  - padrão W3C (2002)
  - granularidade flexível
    - documento
    - elemento
    - conteúdo
- elementos
  - EncryptedData
    - EncryptionMethod
    - CipherData
    - KeyInfo
  - EncryptedKey
    - ReferenceList
- EncryptedHeader
    - EncryptedData

```
<Envelope>
  <Header>
    <wsse:Security>
      <xenc:EncryptedKey>
        <xenc:EncryptionMethod
          Algorithm= "http://www.w3.org/.../xmlenc#rsa-1_5"/>
        <xenc:CipherData>
          <xenc:CipherValue>...</xenc:CipherValue>
        </xenc:CipherData>
        <xenc:ReferenceList>
          <xenc:DataReference URI="#enc1"/>
        </xenc:ReferenceList>
      </xenc:EncryptedKey>
    </wsse:Security>
  </Header>

  <Body>
    <xenc:EncryptedData
      Type="http://www.w3.org/2001/04/xmlenc#Element"
      Id="enc1">
      <xenc:EncryptionMethod
        Algorithm="http://www.w3.org/.../xmlenc#tripledes-cbc"/>
      <xenc:CipherData>
        <xenc:CipherValue>...</xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedData>
  </Body>
</Envelope>
```

WS-Security: Signature

- XML-Signature
  - padrão W3C (2008)
  - granularidade flexível
- elementos
  - Signature
    - SignedInfo
      - CanonicalizationMethod
      - SignatureMethod
      - Reference
        - Transforms
        - DigestMethod
        - DigestValue
    - SignatureValue
    - KeyInfo

```
<ds:Signature>
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
        Algorithm=".../xml-exc-c14n#"/>
    <ds:SignatureMethod
        Algorithm=".../xmldsig#rsa-sha1"/>
    <ds:Reference URI="#body">
      <ds:Transforms>
        <ds:Transform
           Algorithm=".../xml-exc-c14n#"/>
      </ds:Transforms>
      <ds:DigestMethod
           Algorithm=".../xmldsig#sha1"/>
      <ds:DigestValue>...</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>...</ds:SignatureValue>
  <ds:KeyInfo>
    <wsse:SecurityTokenReference>
      <wsse:Reference URI="#X509Token"/>
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
...
<Body Id="body">
```

WS-Security

- Security Tokens
  credenciais de segurança
  autenticação
  - tipos
    - UserNameToken
    - BinarySecurityToken
      X509, Kerberos
    - XML token
      SAML e XrML
- Timestamp
  Created
  Expires

```
<wsse:BinarySecurityToken
    Id="X509Token"
    ValueType="...#X509v3"
    EncodingType="...#Base64Binary">
  ...QEmtJZc0rqrKh5i...
</wsse:BinarySecurityToken>

<wsse:UsernameToken>
  <wsse:Username>scott</wsse:Username>
  <wsse:Password
      Type="wsse:PasswordText">
    ...qr5i...
  </wsse:Password>
</wsse:UsernameToken>

<wsu:Timestamp wsu:Id="T0">
  <wsu:Created>2001-09-
  13T08:42:00Z</wsu:Created>
  <wsu:Expires>2001-10-
  13T09:00:00Z</wsu:Expires>
</wsu:Timestamp>

<ds:Signature>
  <ds:SignedInfo>
    <ds:Reference URI="#X509Token">
    <ds:Reference URI="#T0">
  </ds:SignedInfo>
```

Exercícios

[30] CESPE – 2011 – MEC

Uma das formas de se atribuir segurança aos web services é adotar o XML encryption, que permite criptografar partes confidenciais de um documento, podendo outras partes estar legíveis sem restrição de processamento

[31] CESPE – 2013 – CNJ

Ao se utilizar a tecnologia web services para garantia de uma autenticação segura, devem ser usados signed security tokens, que consistem em elementos criptografados a partir de uma chave de acesso previamente acordada.

[32] CESPE –2013 – TER-MS [adaptada]

O WS-Security propõe uma série de extensões para aprimorar a segurança dos *web services* no UDDI e no WSDL. Por questão de compatibilidade, essas extensões não afetam os cabeçalhos do envelope SOAP.

RESTful

- Representational State Transfer
  - abordagem baseada em recursos
  - estilo arquitetural
    - cliente-servidor, stateless e “cacheável”
    - serviço identificado pela URI
- RESTful web services
  - conjunto de recursos identificados por URIs
  - múltiplas representações por recurso
  - operações mapeadas para métodos HTTP
    - Create (POST), Reader (GET), Update (PUT) e Delete (DELETE)
- RESTful x WS-*
  - vantagens: mais simples, menor overhead
  - desvantagens: segurança,

Exercícios

[33] CESGRANRIO - 2008 - Petrobrás

A interoperabilidade entre aplicações nos dias atuais é fortemente baseada no uso de web services. Duas abordagens arquiteturais distintas para o projeto e implementação de web services têm-se firmado no cenário de tecnologia. São elas:

a) REST e WS-*

b) SOAP e WSDL

c) RPC e RMI

d) SGML e HTML

e) B2B e B2C

Exercícios

[34] CESGRANRIO – 2007 – REFAP [adaptada]

O estilo arquitetural REST (Representational State Transfer) para WEB tem como característica:

a) permitir o uso de RPC diretamente sobre SSL, para aplicações seguras.

b) acelerar a transferência do FTP com a implementação de cache.

c) utiliza a XML como mecanismo de solicitação/resposta para Web Services.

d) usar SOAP para interoperabilidade entre sistemas heterogêneos.

e) utilizar os métodos HTTP: GET, POST, PUT e DELETE.

Exercícios

[35] CESPE – 2010 – MPU

Web services é uma tecnologia utilizada para fazer a integração de sistemas e a comunicação entre aplicações diferentes. Essa tecnologia possibilita que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis entre si. Os *web services* são componentes que permitem às aplicações enviar e receber dados em formato XML: cada aplicação pode ter a sua própria linguagem, que é traduzida para uma linguagem universal, o formato XML.

Com base nessas informações, julgue o item seguinte.

REST (*Representationals State Transfer*) é uma tecnologia que está sendo utilizada em *web services*, como substituta das tecnologias SOAP (*Simple Object Access Protocol*) e WSDL.

Exercícios

[36] CESPE – 2010 – TCU

* O sistema deverá ser integrado, interoperável, portável e seguro.
* O sistema deverá apoiar tanto o processamento online, quanto o suporte a decisão e gestão de conteúdos.
* O sistema deverá ser embasado na plataforma JEE (Java enterprise edition) v.6, envolvendo servlets, JSP (Java server pages), Ajax, JSF (Java server faces) 2.0, Hibernate 3.5, SOA e web services.

Com relação às diferentes arquiteturas e tecnologias que, se escolhidas, impactarão as características do sistema descrito no texto, julgue o item.

O estilo de arquitetura de *software* denominado REST (*representational state transfer*) demanda mais recursos computacionais que o modelo de desenvolvimento de sistemas embasado em SOAP (*single object access protocol*), por isso não é recomendável a adoção do padrão REST de arquitetura de *software* no desenvolvimento do sistema em questão

Gabarito

01 - ERRADO	02 - A	03 - A	04 - A	05 - C	06 - ERRADO
07 - D	08 - A	09 - B	10 - C	11 - C	12 - E
13 - C(61) E(62)	14 - D	15 - E	16 - C	17 - E	18 - ERRADO
19 - ERRADO	20 - CERTO	21 - ERRADO	22 - ERRADO	23 - CERTO	24 - CERTO
25 - D	26 - C	27 - A	28 - CERTO	29 - CERTO	30 - CERTO
31 - ERRADO	32 - ERRADO	33 - A	34 - E	35 - CERTO	36 - ERRADO

WebServices

Leonardo Marcelino
http://www.itnerante.com.br/profile/LeonardoMarcelino