

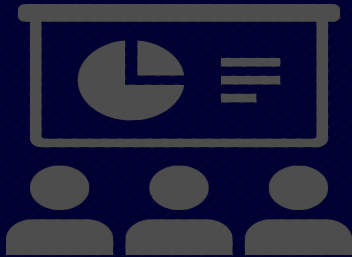
# SISTEMAS DISTRIBUÍDOS

## Web Services

Lattes - linkedin  
euristenhojr@gmail.com  
<http://www.unichristus.edu.br/>

**Euristenho** Queiroz de Oliveira **Júnior**  
**Especialista** em Engenharia de Software  
**MSc** em Engenharia de Software

# AGENDA



1. Apresentação

2. Livros

3. Modelos

4. SOAP

5. Serviços WEB

6. Exercícios

7. Referências

## FORMAÇÃO ACADÊMICA

- ◆ Graduado em Telemática/Telecomunicações - IFCE ( 2002 - 2008)
- ◆ Especialista em Engenharia de Software - FA7 ( 2011 - 2013)
- ◆ MSc em Engenharia de Software - UFPE ( 2011 - 2015)

## CURRÍCULO PROFISSIONAL

- ◆ Atuei 4 anos na empresa privada
- ◆ 9 anos no ambiente Público
- ◆ Atualmente Líder Técnico de 45 Projetos de Tecnologia na SEPOG/PMF

## DOCÊNCIA

- ◆ Professor Substituto das Disciplinas de Sistemas de Informação – FA7 (2011 - 2012)
- ◆ Professor da Especialização em Sistemas WEB – FJN (2011 - 2012)
- ◆ Professor de Bancas de graduação em Sistemas de Informações – FA7 (2012)
- ◆ Professor dos Cursos de Tecnologia da Unifanor (2015 - ATUAL)
- ◆ Professor da Unichristus (2018 - ATUAL)

- **Sistemas Distribuídos - Conceitos e Projeto** - 5ª Ed.  
2013 - George Coulouris, Tim Kindberg, Jean Dollimore
- **Sistemas Distribuídos, Princípios e Paradigmas** - 2ª  
Ed. 2007 - Andrew S. Tanenbaum, Maarten Van Steen



- ◆ Apresentar os conceitos básicos da computação distribuída e seus desafios como Heterogeneidade; Segurança; Tolerância a Falhas; Escalabilidade; Concorrência; Coordenação e Sincronização de processos; Comunicação interprocessos.
- ◆ Desenvolver competências e habilidades que auxiliem o profissional de Ciência da Computação a implementar os conceitos de sistemas distribuídos no desenvolvimento de sistemas de informação.
- ◆ Conhecer a aplicação desses conceitos em estudos de Casos que abordam arquiteturas e tecnologias modernas como RMI, CORBA e Web Services.

# Agenda

1. Introdução
2. Serviços Web
  1. SOAP (Simple Object Access Protocol)
  2. Serviços Web x Objetos Distribuídos

## Introdução



O que são Web Services?



## Introdução



Com qual intuito eles foram criados?

## Introdução



Qual a importância deles no cenário atual?

## Introdução

- Um serviço Web (Web service) fornece **uma interface de serviço** que permite aos clientes **interagirem com servidores** de uma maneira mais geral do que acontece com os navegadores Web
  - O acesso ao serviço Web ocorre, normalmente, por meio de requisições e respostas **formatadas em XML** e transmitidas por **HTTP**
- Importância nos sistemas distribuídos:
  - Suportam **atividade conjunta** na Internet global, incluindo a área fundamental da integração de empresa para empresa (**business-to-business, B2B**)
  - **Cultura de “mashup”**, permitindo que desenvolvedores criem software inovador em cima da base de serviços já existente.
  - Fornecem o **middleware** subjacente para a **computação de grade (grid)** e **em nuvem**.

# Introdução

- Contextualização:
  - O crescimento da Web nas últimas duas décadas prova a eficácia do uso de protocolos simples na Internet (Ex: HTTP)
  - Restrições do navegador de **propósito geral** como cliente:
    - **restringe a abrangência** potencial das aplicações Web
    - Por que??

- Nesse contexto:
  - Os serviços Web (Web services) retornam a esse modelo, no qual **um cliente específico da aplicação** interage pela Internet com um serviço que possui uma **interface funcionalmente especializada**.
    - Possibilitam uma **forma estruturada de interoperabilidade** entre clientes e servidores
    - Um programa cliente em uma organização pode interagir com um servidor em outra organização, **sem supervisão humana**.
    - “Dividir para conquistar”
    - Devido à generalidade de suas interações, os serviços Web **não podem ser acessados diretamente pelos navegadores**.

## Introdução

- Modelo genérico de funcionamento dos serviços Web:

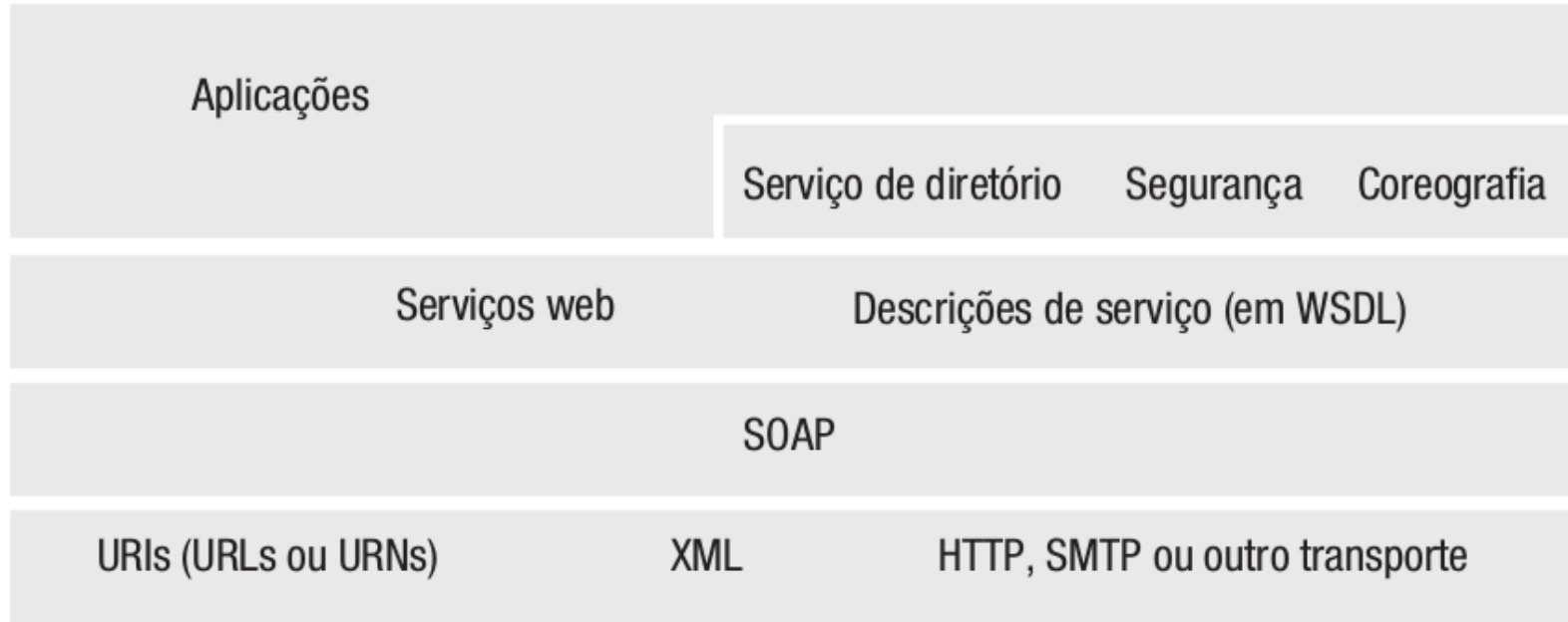


# Introdução

- Características:
  - O fornecimento de serviços Web como um acréscimo aos servidores Web é baseado na capacidade de usar uma requisição HTTP para **provocar a execução de um programa**
  - A **representação de dados externa e o empacotamento das mensagens** trocadas entre clientes e serviços Web são feitos em XML (entre outras alternativas).
  - O **protocolo SOAP (Simple Object Access Protocol)** especifica as regras de uso da XML para empacotar mensagens

## Introdução

- Principais pontos da arquitetura de comunicação





## Introdução

- Principais pontos da arquitetura de comunicação:
  - Um serviço Web é **identificado por um URI** e pode ser acessado pelos clientes usando **mensagens formatadas em XML**.
  - O **protocolo SOAP** é usado para **encapsular essas mensagens e transmiti-las** por HTTP ou outro protocolo, por exemplo, TCP ou SMTP.
  - Um serviço Web divulga para potenciais clientes a interface e outros aspectos dos serviços que implementa por meio das **descrições de serviço**:
    - usado como **base para um entendimento comum** entre cliente e servidor quanto ao serviço oferecido
  - Outra necessidade comum no middleware é um **serviço de atribuição de nomes, ou de diretório**, para permitir que os clientes descubram serviços.

# Agenda

1. Introdução
2. Serviços Web
  1. SOAP (Simple Object Access Protocol)
  2. Serviços Web x Objetos Distribuídos

## Serviços Web (Web Services)

- Geralmente, uma interface de serviço Web consiste em um **conjunto de operações** que podem ser usadas por um cliente na Internet
  - Um serviço Web pode ser gerenciado por um servidor Web, junto a páginas Web, ou pode ser um serviço totalmente separado.
- A principal característica da maioria dos serviços Web é que eles podem **processar mensagens SOAP formatadas em XML**.
  - Alternativa é a estratégia **REST (REpresentational State Transfer)**
- Cada serviço Web usa sua **própria descrição** para tratar das **características específicas das mensagens** que recebe

## Serviços Web (Web Services)

- Ex: Web Service da Amazon
  - fornece operações que permitem aos clientes obter informações sobre produtos, adicionar um item a um carrinho de compras ou verificar o status de uma transação.
    - O que podemos fazer com isso?



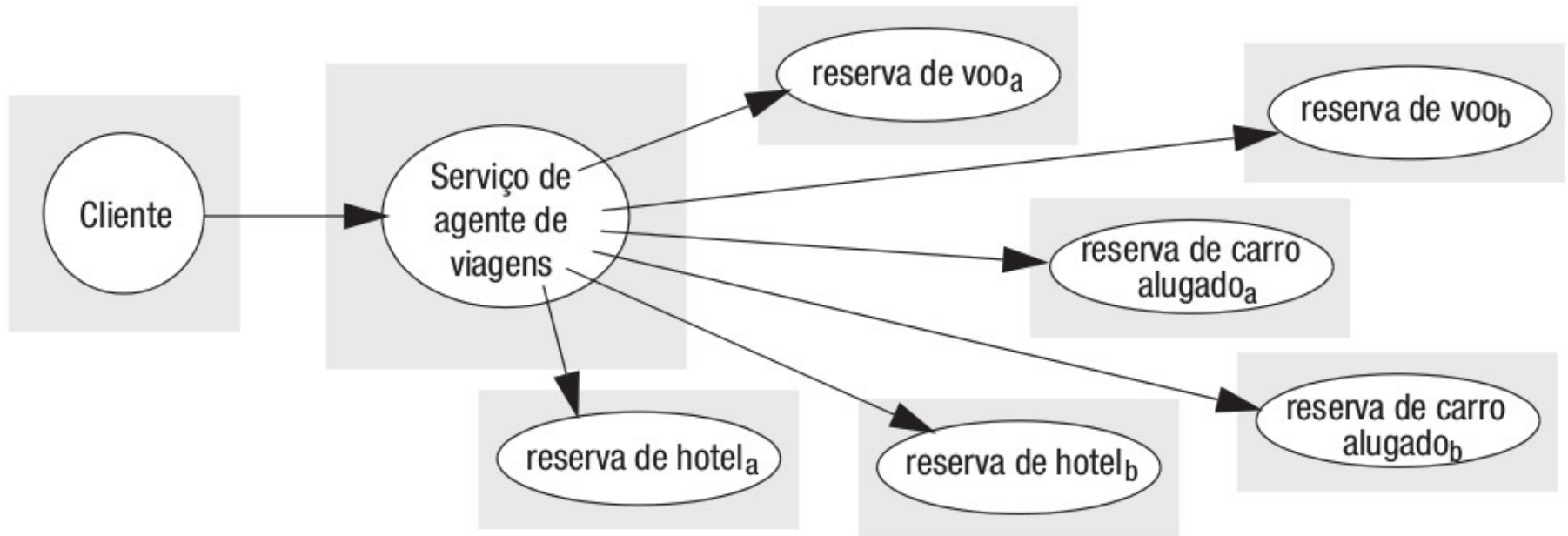
## Serviços Web (Web Services)



Quais características um Web Service possui?

## Serviços Web (Web Services)

- Características gerais:
  - **Combinação de serviços Web**



- **Características gerais:**

- **Padrões de comunicação:**

- Web Services são projetados **para serem independentes** de qualquer paradigma de programação em particular
    - Em geral, os serviços Web usam um **padrão de comunicação de requisição-resposta síncrona** com seus clientes, ou se comunicam por meio de **mensagens assíncronas**
    - Para possibilitar o uso de uma variedade de padrões de comunicação, o protocolo SOAP é baseado no empacotamento de mensagens unidirecionais únicas.
    - Como tratar então uma interação requisição-resposta?

## Serviços Web (Web Services)

- **Características gerais:**
  - **Padrões de comunicação:**
    - Web Services são projetados **para serem independentes** de qualquer paradigma de programação em particular
    - Em geral, os serviços Web usam um **padrão de comunicação de requisição-resposta síncrona** com seus clientes, ou se comunicam por meio de **mensagens assíncronas**
    - Para possibilitar o uso de uma variedade de padrões de comunicação, o protocolo SOAP é baseado no empacotamento de mensagens unidirecionais únicas.
    - Como tratar então uma interação requisição-resposta?
      - Utilizando **pares de mensagens únicas** e especificando como irá **representar as operações**, seus argumentos e resultados.



## Serviços Web (Web Services)

- **Características gerais:**
  - **Baixo acoplamento:**
    - se refere a **minimizar as dependências** entre os serviços para se ter uma arquitetura subjacente flexível (reduzindo o risco de uma alteração em um serviço causar uma **reação em cadeia** em outros serviços)
    - Melhoramento através de:
      - Programação com interfaces
      - Tendência em usar interfaces simples e genéricas em sistemas distribuídos
      - Serviços Web podem ser usados com uma variedade de **paradigmas de comunicação**, incluindo **comunicação por requisição-resposta** ou **troca de mensagens assíncrona**

- **Características gerais:**
  - **Representação de Mensagens:**
    - Tanto o protocolo SOAP quanto os dados que ele transporta são representados em XML
      - Ocupam mais espaço do que as binárias e exigem mais tempo para seu processamento.
      - Qual a implicação disso?

## Serviços Web (Web Services)

- **Características gerais:**
  - **Referências de serviço:**
    - Cada serviço Web tem um URI, o qual os clientes utilizam para se referirem a ele
  - **Transparência**
    - Uma tarefa importante de **muitas plataformas de middleware** é **proteger** o programador dos detalhes da representação e do empacotamento dos dados e, às vezes, fazer as invocações remotas parecerem ser locais.
    - Os detalhes do protocolo SOAP e da XML geralmente são ocultos por uma API local

# Agenda

1. Introdução
2. Serviços Web
  1. SOAP (Simple Object Access Protocol)
  2. Serviços Web x Objetos Distribuídos

## SOAP(Simple Object Access Protocol)



O que você sabe sobre o SOAP?

## SOAP(Simple Object Access Protocol)

- Projetado para permitir tanto **interação cliente-servidor como assíncrona pela Internet**.
- Ele define um esquema para uso da XML para representar o conteúdo de mensagens de requisição-resposta
- Inicialmente, era baseado apenas em HTTP, mas a versão atual é projetada para usar uma variedade de protocolos de transporte, incluindo SMTP, TCP ou UDP

## SOAP(Simple Object Access Protocol)

- A especificação do SOAP declara:
  - como a XML deve ser usada para **representar o conteúdo de mensagens individuais**;
  - como duas mensagens podem ser combinadas para produzir um **padrão de requisição-resposta**;
  - as regras sobre como os destinatários das mensagens devem processar os elementos XML que elas contêm;
  - como HTTP e SMTP devem ser usados para comunicar mensagens SOAP

## SOAP(Simple Object Access Protocol)

### ● Mensagens SOAP:

- Uma mensagem SOAP é transportada em um envelope.
  - Dentro do envelope existe um **cabeçalho opcional e um corpo**
  - Os cabeçalhos das mensagens podem ser usados para **estabelecer o contexto necessário** para um serviço ou para **manter um log ou uma auditoria** das operações
- Os elementos XML envelope, cabeçalho e corpo, juntos a outros atributos e elementos de mensagens SOAP, são definidos como um **esquema no espaço de nomes XML do protocolo SOAP**.
- Os remetentes de mensagens usam as **descrições do serviço** para gerar o corpo e para garantir que ele possua o conteúdo correto, e os destinatários das mensagens as utilizam para **analisar e verificar a validade do conteúdo**.



# SOAP(Simple Object Access Protocol)

## ● Mensagens SOAP:

*env:envelope* xmlns:env = URI do espaço de nomes para envelopes SOAP

*env:corpo*

*m:exchange*

xmlns:m = URI do espaço de nomes da descrição do serviço

*m:arg1*

Hello

*m:arg2*

World

**REQUISIÇÃO**

**RESPOSTA**

*env:envelope* xmlns:env = URI do espaço de nomes para envelopes SOAP

*env:corpo*

*m:exchangeResponse*

xmlns:m = URI do espaço de nomes da descrição do serviço

*m:res1*

World

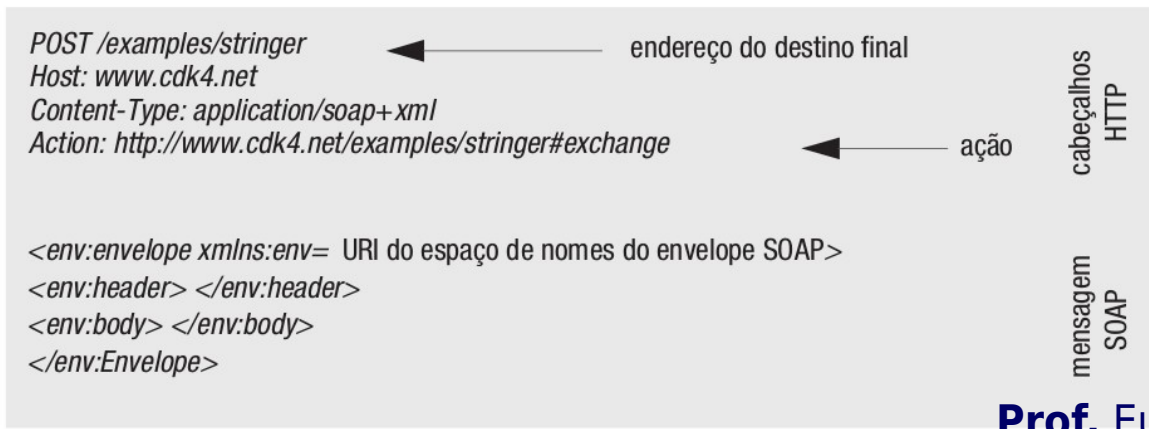
*m:res2*

Hello

## SOAP(Simple Object Access Protocol)

### ● Transporte de mensagens SOAP:

- Um protocolo de transporte é exigido para enviar uma mensagem SOAP para seu destino.
- As **mensagens SOAP são independentes do tipo de transporte** usado – seus envelopes não contém nenhuma referência ao endereço de destino.
- Fica para o protocolo HTTP (ou qualquer protocolo usado para transporte de uma mensagem SOAP) especificar o endereço do destino.



## SOAP(Simple Object Access Protocol)

- **Transporte de mensagens SOAP:**

- A separação da definição do envelope SOAP das informações sobre como e para onde elas devem ser enviadas **torna possível usar uma variedade de protocolos subjacentes diferentes**
  - Qual o problema disso?

## SOAP(Simple Object Access Protocol)

### ● Transporte de mensagens SOAP:

- A separação da definição do envelope SOAP das informações sobre como e para onde elas devem ser enviadas **torna possível usar uma variedade de protocolos subjacentes diferentes**
  - Qual o problema disso?
    - Ela implica que o desenvolvedor deve estar **envolvido nos detalhes do protocolo de transporte específico** escolhido
    - Torna difícil usar **diferentes protocolos para diferentes partes da rota** seguida por uma mensagem em particular

## SOAP(Simple Object Access Protocol)

- WS-ReliableMessaging: comunicação confiável
  - O protocolo normal do SOAP, HTTP, é executado sobre TCP
    - não garante o envio de mensagens em face de todas as dificuldades
      - Quando atinge um tempo limite na espera por confirmações, ele declara que a **conexão está desfeita**
      - Os processos que estão se comunicando ficam sem saber se as mensagens que enviaram recentemente **foram recebidas ou não**.

- **WS-ReliableMessaging: comunicação confiável**
  - O WS-ReliableMessaging ([www.oasis.org](http://www.oasis.org)) é uma recomendação permitindo que uma **mensagem SOAP seja enviada pelo menos uma vez, no máximo uma vez ou exatamente uma vez**, com a seguinte semântica:
    - **Pelo menos uma vez:** a mensagem é entregue pelo menos uma vez, mas se não puder ser entregue, um erro será relatado.
    - **No máximo uma vez:** a mensagem é entregue no máximo uma vez, mas se não puder ser entregue, nenhum erro é relatado.
    - **Exatamente uma vez:** a mensagem é entregue exatamente uma vez, mas se não puder ser entregue, um erro será relatado.

## SOAP(Simple Object Access Protocol)



Qual a implicação do uso do SOAP nos firewalls?

## Exercício Rápido (10 min)

- Faça uma comparação entre os serviços Web e o modelo de objeto distribuído estudado anteriormente.



# Agenda

1. Introdução
2. Serviços Web
  1. SOAP (Simple Object Access Protocol)
  2. Serviços Web x Objetos Distribuídos

## Serviços Web x Objeto Distribuído

- Um **serviço Web tem uma interface** que pode fornecer operações para acessar e atualizar os recursos de dados que gerencia.
  - De forma superficial, a interação entre cliente e servidor é muito parecida com RMI, em que um **cliente usa uma referência de objeto remoto para invocar uma operação em um objeto remoto**.
  - Para um serviço Web, o cliente usa um URI para invocar uma operação no recurso nomeado por esse URI.

- **Comparação:**

- **Referência de Objeto Remoto x URLs:**

- O URI de um serviço Web pode ser comparado com a referência de objeto remoto de um único objeto.
      - Entretanto, no modelo de objeto distribuído, **os objetos podem criar objetos remotos dinamicamente e retornar referências remotas** para eles.
    - Nada parecido com isso pode ser feito com serviços Web, que **não podem criar instâncias de objetos remotos**; com efeito, um serviço Web consiste em um único objeto remoto;

- **Comparação:**
  - **Modelo de serviços Web:**
    - As ferramentas de desenvolvimento devem modelar seus programas que usam serviços Web **considerando o fato de que não estão usando invocação remota transparente de Java para Java**, mas sim **usando o modelo de serviços Web**, no qual objetos remotos não podem ser instanciados.

Obrigado!!!

**Dúvidas?**



## Exercícios

1. Sobre as características dos Web Services, o que significa dizer que Web Services possuem transparência de linguagem? Exemplifique.
2. Sobre as características dos Web Services, o que significa dizer que Web Services possuem design modular? Exemplifique.
3. Por que os Web Services podem ser úteis na integração de software legado. Exemplifique.
4. Realize uma comparação entre a abordagem de um protocolo de requisição-resposta com a implementação de comunicação cliente-servidor no SOAP. Cite dois motivos pelos quais o uso de mensagens assíncronas pelo SOAP é mais apropriado para uso na Internet. Até que ponto o uso de HTTP pelo SOAP reduz a diferença entre as duas estratégias?

### ● **Web Service:**

- é a parte de uma lógica de negócio, localizada em algum lugar na Internet, que é acessível através de protocolos da Internet, tal como HTTP ou SMTP. O Uso de um Web Service pode ser tão simples como logar em um site ou tão complexo como uma negociação multiorganizacional. (David Chappel, Java Web Services)
- Quais as características do Web Service?

### ● **Web Service: Características**

- **Baseado em XML:** garante a interoperabilidade
- **Baixo acoplamento:** alterações no Web Service não ocasionaram a necessidade de alteração da aplicação cliente.
- **Síncrono/Assíncrono:** os dois tipos de transmissão estão disponíveis
- **Suporte a chamadas RPC**
- **Suporte a diferentes tipos de documentos**



## Revisando conceitos dos Web Services



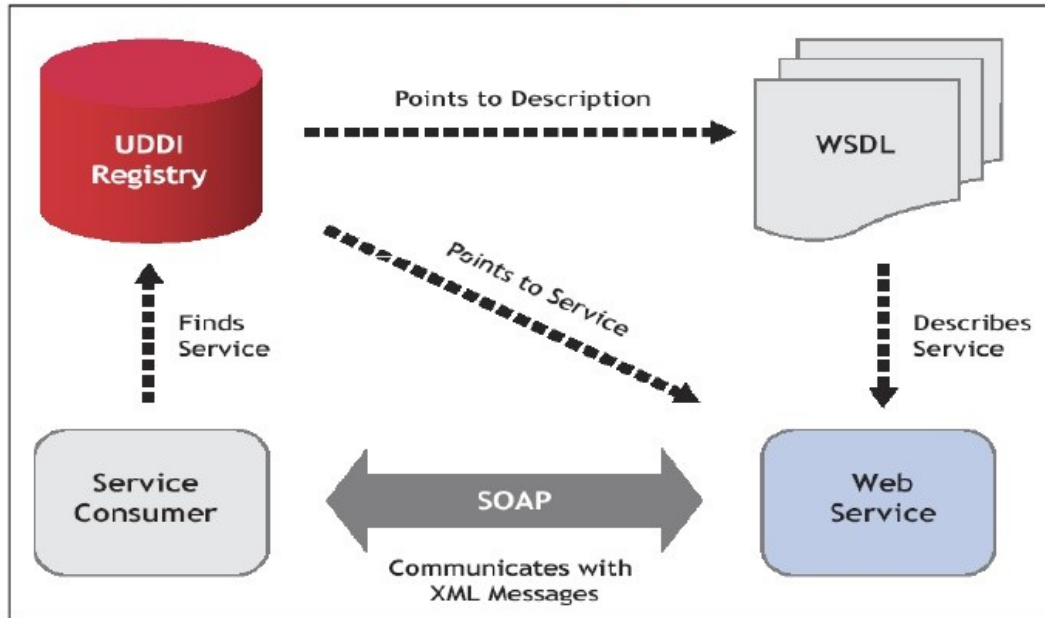
Quais tecnologias são utilizadas?

### ● Web Service: Principais Tecnologias

- **SOAP: estrutura de empacotamento padrão** para transportar documentos XML através dos protocolos de Internet (HTTP, SMTP, FTP...)
- **WSDL**: Tecnologia que padroniza a descrição da interface de um web service. Descreve propriedades como **parâmetros de entrada/saída** para uma **chamada externa, estrutura de uma função e o protocolo** que é utilizado no mesmo.
- **UDDI (Universal Discovery Description and Integration)**: Repositório mundial de Web Services. Útil para publicar e/ou descobrir serviços que já foram criados e utilizar os já existentes.

## Revisando conceitos dos Web Services

- Funcionamento básico:

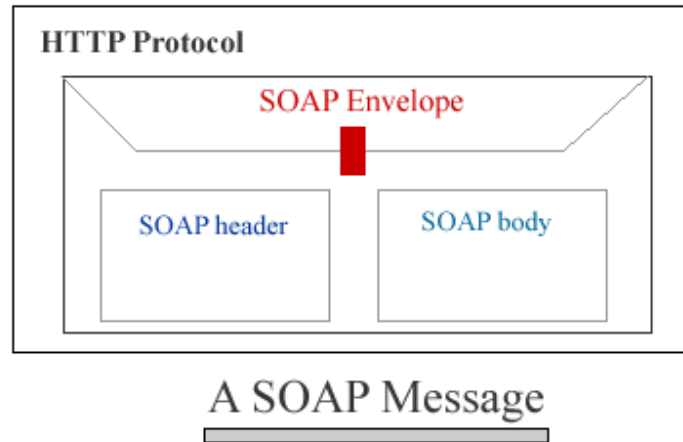


## Revisando conceitos dos Web Services



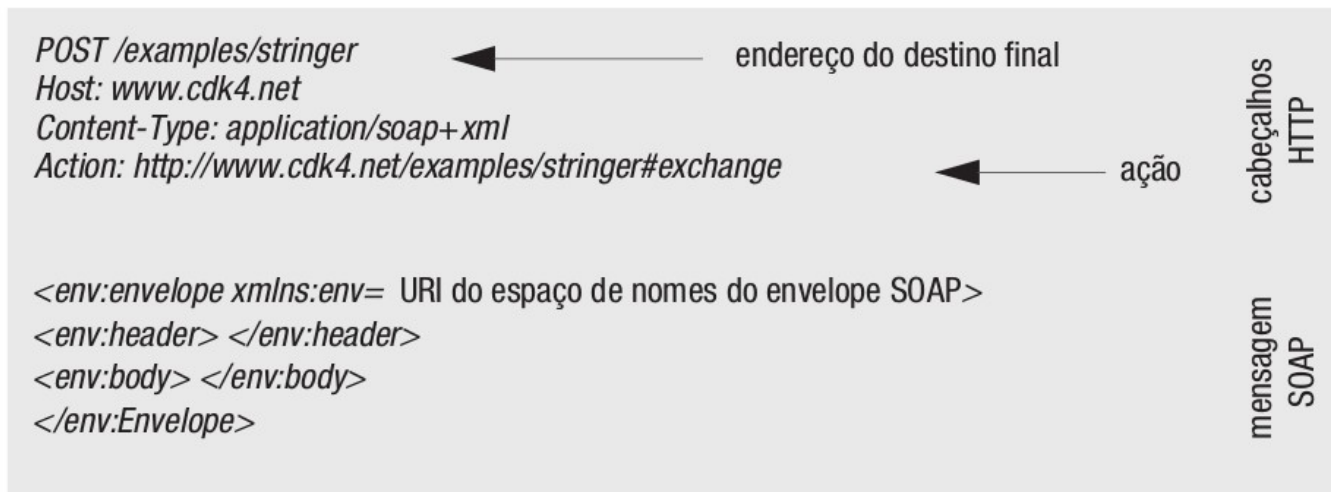
O que você lembra sobre o SOAP?

- **SOAP (Simple Object Access Protocol):**
  - Protocolo para **trocar informações** pela rede em um ambiente distribuído
  - **Codificação de objetos** para transporte pela rede, criando um envelope
  - **Implementação simples**



- **SOAP (Simple Object Access Protocol):**

- Exemplo:





Qual a necessidade das definições de interfaces?

## Descrições de Serviços

- As definições de interface são necessárias para permitir que os clientes se comuniquem com os serviços.
- Para serviços Web, as **definições de interface** são fornecidas como **parte de uma descrição de serviço** mais geral, que especifica duas outras características adicionais:
  - como as mensagens devem ser comunicadas (por exemplo, por SOAP com HTTP)
  - URI do serviço
- Escritas em XML.
  - Por que?



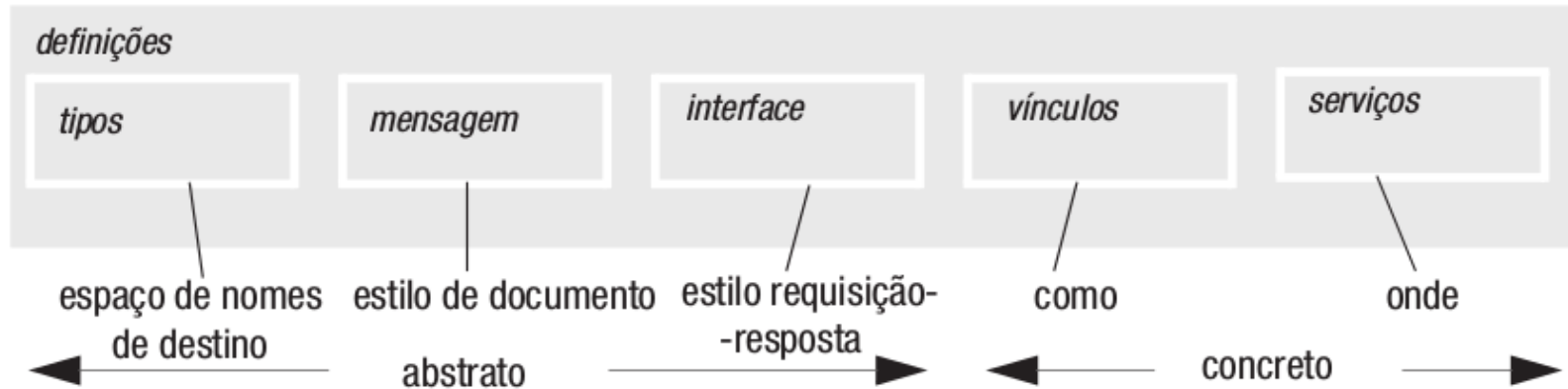
## Descrições de Serviços



O que você lembra sobre o WSDL?

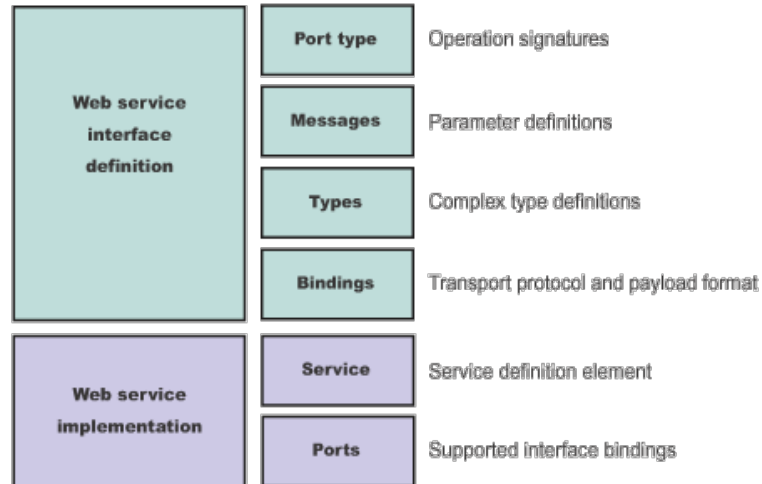
## Descrições de Serviços

- **WSDL (Web Services Description Language):**
  - Ela define um esquema XML para **representar os componentes de uma descrição de serviço**, os quais incluem, por exemplo, os nomes de elemento definições, tipos, mensagens, interface, vínculos e serviços



## Revisando conceitos dos Web Services

- **WSDL (Web Services Description Language):**
  - Descreve a interface para alguém que utilizará o serviço
  - Operações de entrada/saída
  - Acesso (Protocolo de Comunicação)
  - Localização do serviço



## Descrições de Serviços



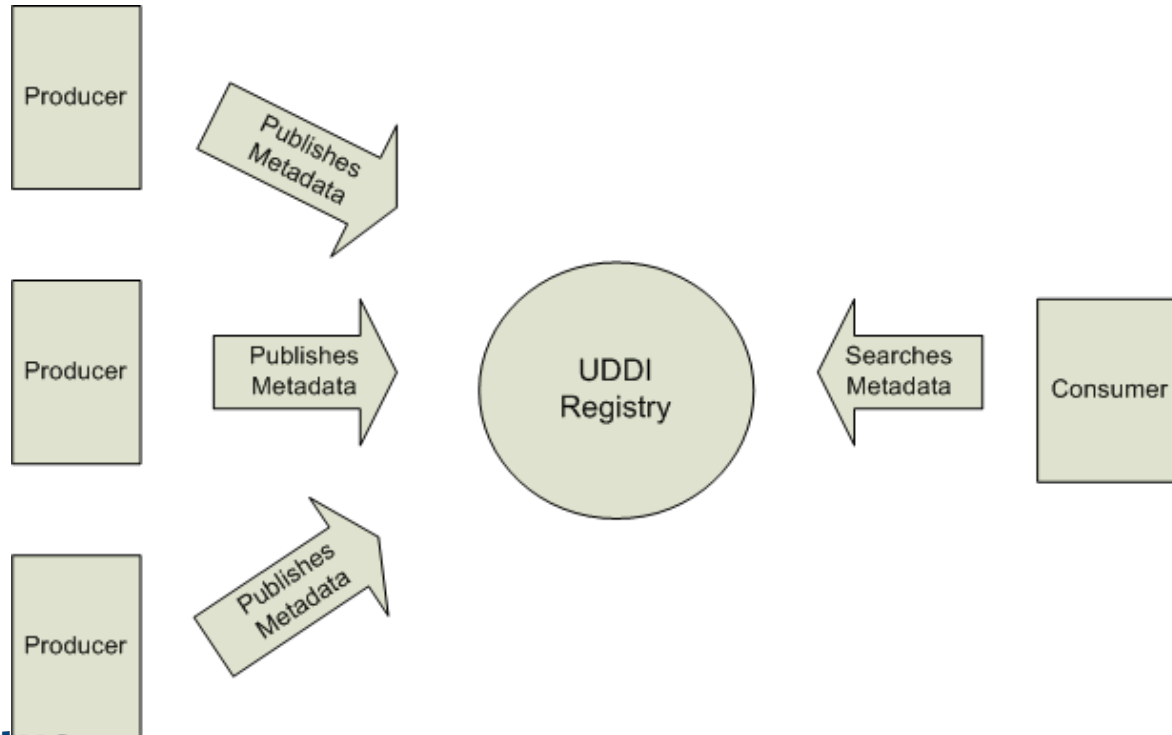
Como obter as descrições de serviço de um Web Service?

## Serviço de Diretório

- Forma conveniente:
  - usar um serviço de diretório para tornar esses serviços disponíveis para os clientes.
- Esse é o objetivo do serviço UDDI (Universal Description, Discovery and Integration)
  - Fornece um serviço de nome e um serviço de diretório.
  - Isto é, as descrições de serviço WSDL podem ser pesquisadas pelo nome (um serviço de catálogo telefônico) ou pelo atributo (um serviço de páginas amarelas).

- **UDDI (Universal Description, Discovery and Integration):**
  - [uddi.xml.org](http://uddi.xml.org)
  - Protocolo avançado para publicação e descoberta de serviços Web
  - Padrão proposto pela fundação OASIS, que visa a padronização de alguns elementos Web
  - Conceito de “Páginas Amarelas”, similar às listas telefônicas
  - Define registros para Web Services:
    - Provedor
    - Informações sobre o serviço
    - Acesso técnico

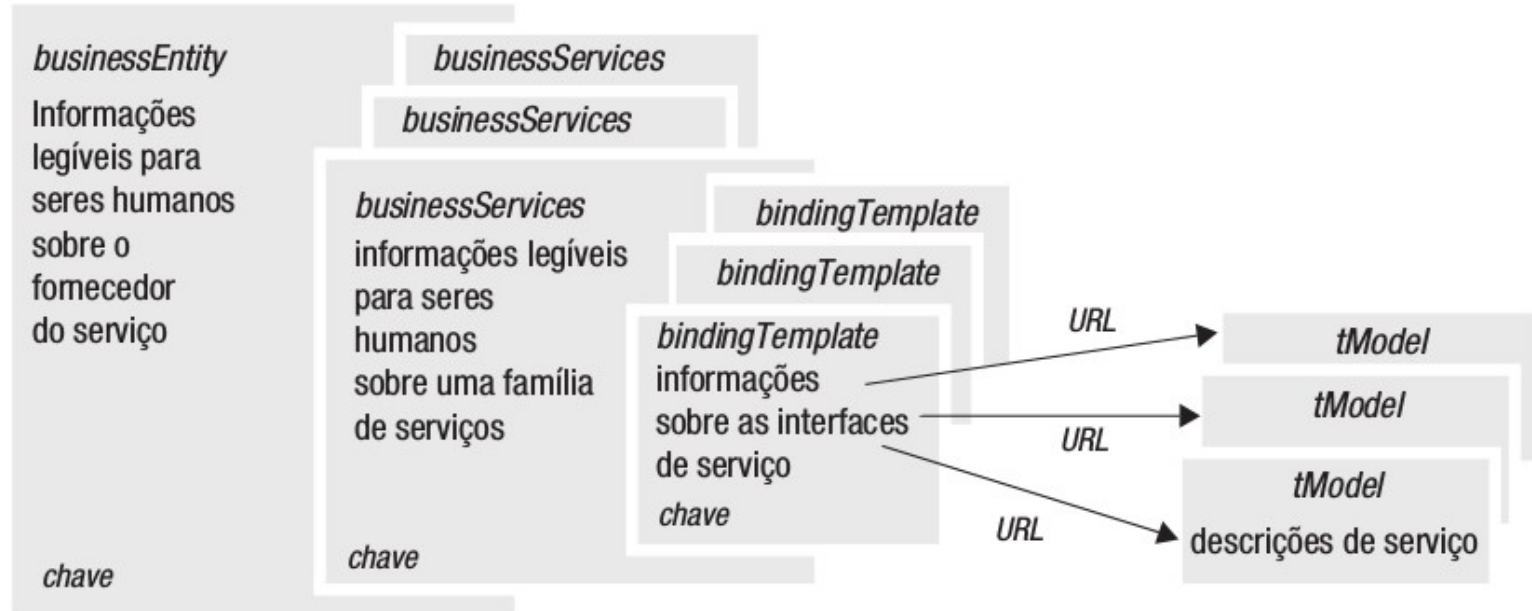
- **UDDI (Universal Description, Discovery and Integration):**



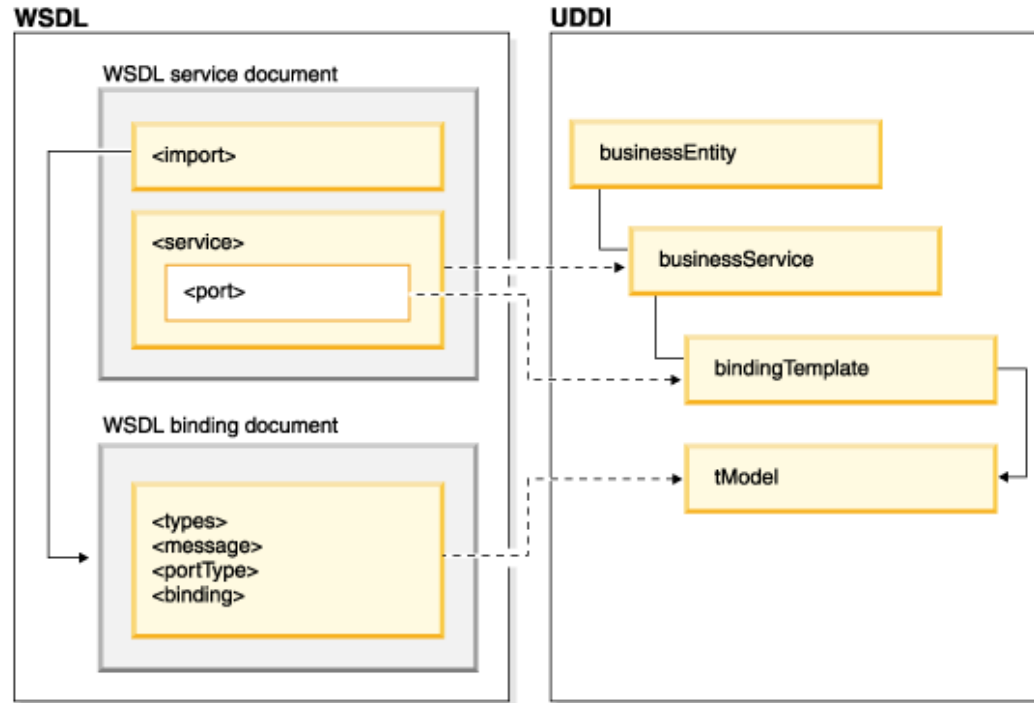
- **UDDI (Universal Description, Discovery and Integration):**
  - Estrutura de dados utilizada:
    - Cada elementos pode ser acessado a partir de uma chave:
      - **businessEntity**: descreve a organização que fornece esses serviços Web, dando seu nome, endereço, atividades etc.;
      - **businessServices**: armazena informações sobre um **conjunto de instâncias de um serviço Web**, como seu nome e uma descrição de seu propósito; por exemplo, agente de viagens ou livraria;
      - **bindingTemplate**: contém o **endereço** de uma instância de serviço Web e **referências para descrições do serviço**;
      - **tModel**: contém descrições de serviço, normalmente documentos WSDL, armazenadas fora do banco de dados e acessadas por meio de URLs.



- **UDDI (Universal Description, Discovery and Integration):**



- UDDI (Universal Description, Discovery and Integration):



## Exercícios

- Simular o web service de um sistema acadêmico onde deverá ser possível executar as seguintes operações:
  - Consulta de dados dos alunos (dados pessoais)
  - Dados do histórico de disciplinas do aluno (com as notas)
  - Média dos alunos
- Implemente uma aplicação que consuma o web service dos correios onde deve ser possível:
  - Rastrear uma encomenda
  - Verificar o preço e prazo de uma entrega

## Exercícios

- Desafio:
  - Realizar a implementação de um web service que deverá depender de autenticação para a consulta de dados.

## Referências

- Sistemas Distribuídos - Conceitos e Projeto - 5ª Ed. 2013 - George Coulouris, Tim Kindberg, Jean Dollimore
- Andrew S. Tanenbaum, Sistemas Distribuídos,. Princípios e Paradigmas 2ª edição