

# **Técnico em Desenvolvimento de Sistemas**

## **Programação WEB 3**

**Introdução às tecnologias Web Services  
SOA, SOAP, RESTful, WSDL e UDDI**

**Prof. Laércio Silva**

[laercio.silva31@etec.sp.gov.br](mailto:laercio.silva31@etec.sp.gov.br)

[Indsilva@Hotmail.com](mailto:Indsilva@Hotmail.com)

## História

- No ano de 2000, a W3C (World Wide Web Consortium) aceitou a submissão do **Simple Object Access Protocol (SOAP)**
- Este formato de mensagem **baseado em XML** estabeleceu uma estrutura de transmissão para comunicação entre aplicações (ou entre serviços) via HTTP.
- Sendo uma tecnologia não amarrada a fornecedor, o SOAP disponibilizou uma alternativa atrativa em relação aos protocolos proprietários tradicionais.

## Web Services e suas tecnologias

- No decorrer do ano seguinte, o W3C publicou a especificação **WSDL (Web Service Description Language)**.
- Uma nova implementação do XML.
- Este padrão forneceu uma linguagem para descrever a **interface dos web services**.
- Posteriormente suplementada pela especificação UDDI (Universal Description, Discovery and Integration).
- Que proporcionou um mecanismo padrão para a descoberta dinâmica (dynamic discovering) de descrições de serviço, a primeira geração da **plataforma de Web services** foi estabelecida.

# Web Services e suas tecnologias

## O relacionamento entre especificações de primeira geração

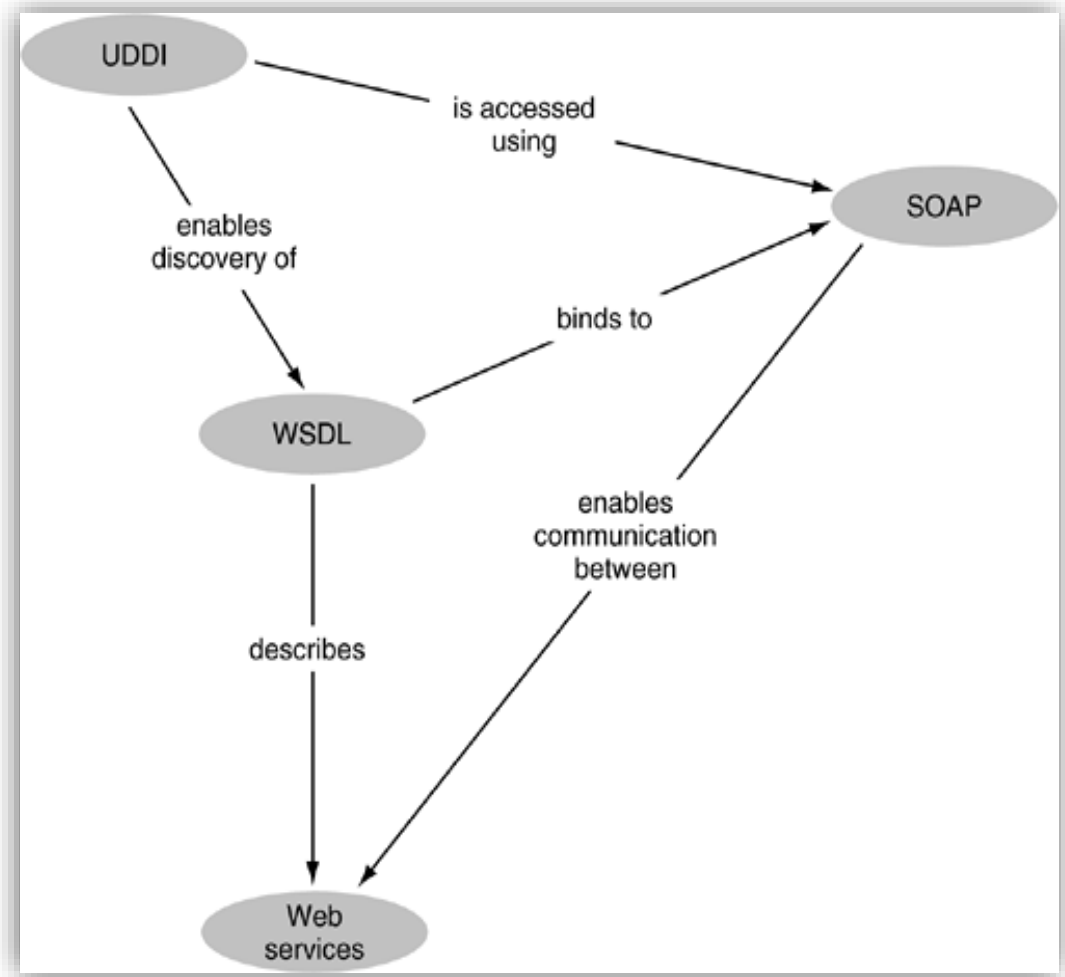
**Is accessed using:** é acessado utilizando;

**Enables discovery of:** permite a descoberta de;

**Describes:** descreve;

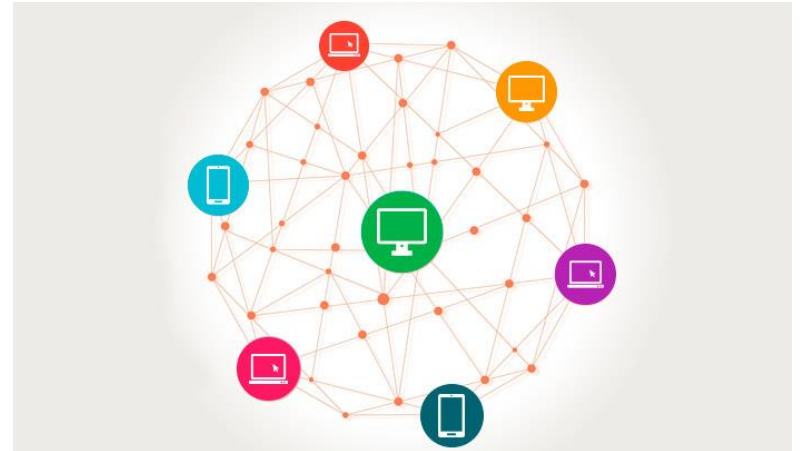
**Enables communication between:** permite a comunicação entre;

**Binds to:** ligação para.



# Web Services e suas tecnologias

- Desde então, os web services foram adotados por vendedores e fabricantes num ritmo considerável.
- Suporte amplo da indústria seguiu-se à popularidade e importância desta plataforma e de *princípios de projeto orientados a serviço*.
- Isto levou à criação de uma segunda geração de **especificação de Web services**.



## Entendendo os Serviços

- O conceito de serviços em uma aplicação existe faz algum tempo.
- Serviços, assim como componentes, são considerados blocos de construção independentes, os quais coletivamente representam um ambiente de aplicação.
- No entanto, diferente de componentes tradicionais, serviços têm algumas características únicas que lhes permitem participar como parte de uma *arquitetura orientada a serviços* – **SOA**.

## Web Services e a arquitetura voltada a SOAP

- Uma destas características é a completa autonomia em relação a outros serviços.
- Isto significa que cada serviço é responsável por seu próprio domínio, o que tipicamente significa limitar seu alcance para uma função de negócio específica (ou um grupo de funções relacionadas).

## Web Services e a arquitetura voltada a SOAP

- Este enfoque de projeto resulta na criação de unidades isoladas de funcionalidades de negócio ligadas fracamente entre si.
- Isto é possível por causa da definição de uma estrutura padrão de comunicação.
- Devido à independência que esses serviços desfrutam dentro desta estrutura, a lógica de programação que encapsulam não tem necessidade de obedecer a nenhuma outra plataforma ou conjunto de tecnologias.



## XML Web Services

- O tipo de serviço mais largamente aceito e bem sucedido é o **XML Web service**.
- Que será daqui em diante chamado apenas de *web service*, ou simplesmente *service*.
- Este tipo de serviço possui dois requisitos fundamentais:
  - Comunica-se via protocolos internet (normalmente HTTP);
  - Envia e recebe dados formatados como documentos XML.

## Web Services e a arquitetura voltada a SOAP

- A ampla aceitação do web service resultou no surgimento de um conjunto de tecnologias suplementares que se tornaram um padrão de fato.
- Assim ao **desenvolver nossos web services** devemos considerar o uso de tecnologias que:
  - Forneça uma descrição de serviço que, no mínimo, consista de um documento WSDL (Web Service Description Language);
  - Seja capaz de transportar documentos XML utilizando SOAP sobre HTTP.

## Web Services e a arquitetura voltada a SOAP

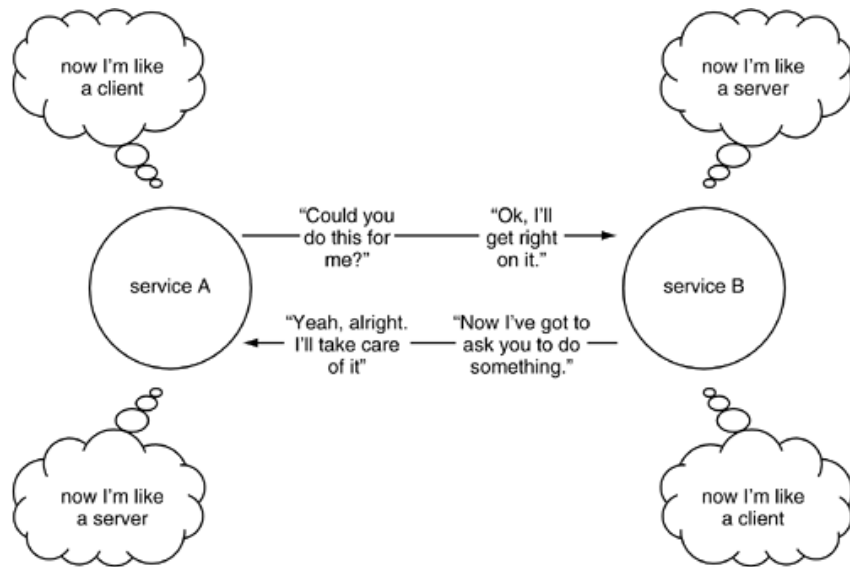
- Estas tecnologias não modificam a funcionalidade do núcleo de um serviço web, tanto como o faz sua habilidade para se representar e comunicar num modo padrão.
- Muitas das convenções de arquitetura assumem que SOAP (Simple Object Access Protocol) e WSDL (Web Service Description Language) fazem parte da estrutura de web services descrita.
- Além disto, é normal que um Web service seja:
  - Capaz de agir como o solicitante e o provedor de um serviço;
  - Registrado com um discovery agent através do qual possam ser localizados.

## Web Services e a arquitetura voltada a SOAP

- Numa conversa  o t  pica com um web service, o cliente iniciador do pedido   um web service tamb  m.
- Qualquer interface exposta por este client service tamb  m o qualifica como um servi  o a partir do qual outros servi  os podem solicitar informa  o.
- Web services n  o se encaixam no modelo cl  ssico de cliente-servidor.
- Eles tendem a estabelecer um sistema ponto-a-ponto, onde cada servi  o pode atuar como cliente ou servidor.

# Web Services e a arquitetura voltada a SOAP

Troca de papéis do Web Services durante uma conversação



Now I'm like a client: agora sou um cliente;

Now I'm like a server: agora sou um servidor;

Service: serviços;

Could you do this for me: poderia fazer isso por mim?;

Ok, I'll get right on it: Ok;

Now I've got to ask you to do something: agora eu gostaria de lhe pedir algo;

Yeah, alright. I'll take care of it: Tudo bem, é só falar.

## Web Services e a arquitetura voltada a SOAP

Para resumir, um serviço web completo é, portanto, qualquer serviço que:

- Está disponível na Internet ou em redes privadas (intranet)
- Usa um sistema de mensagens XML padronizado
- Não está vinculado a nenhum sistema operacional ou linguagem de programação
- É auto descritivo por meio de uma gramática XML comum
- É detectável por meio de um mecanismo de localização simples

## Componentes de serviços da Web

- A plataforma básica de serviços da web é XML + HTTP. Todos os serviços da web padrão funcionam usando os seguintes componentes:
  - SOAP (protocolo de acesso a objetos simples)
  - UDDI (descrição universal, descoberta e integração)
  - WSDL (Web Services Description Language)

### Por que você precisa de um serviço da Web?

- Os aplicativos de negócios modernos usam uma variedade de plataformas de programação para desenvolver aplicativos baseados na web. Alguns aplicativos podem ser desenvolvidos em Java, outros em .Net, enquanto outros em Angular JS, Node.js, etc.
- Na maioria das vezes, esses aplicativos heterogêneos precisam de algum tipo de comunicação entre eles. Por serem construídos em diferentes linguagens de desenvolvimento, torna-se realmente difícil garantir uma comunicação precisa entre os aplicativos.



## Web Services e a arquitetura voltada a SOAP

- É aqui que entram os serviços da Web. Os serviços da Web fornecem uma plataforma comum que permite que vários aplicativos criados em várias linguagens de programação tenham a capacidade de se comunicarem entre si.

## Tipo de serviço da web

- Existem basicamente dois tipos de serviços da web.
  1. Serviços da web SOAP.
  2. Serviços da Web RESTful.
- Para que um serviço da web seja totalmente funcional, certos componentes precisam estar no lugar. Esses componentes precisam estar presentes independentemente de qualquer linguagem de desenvolvimento usada para programar o serviço da web.

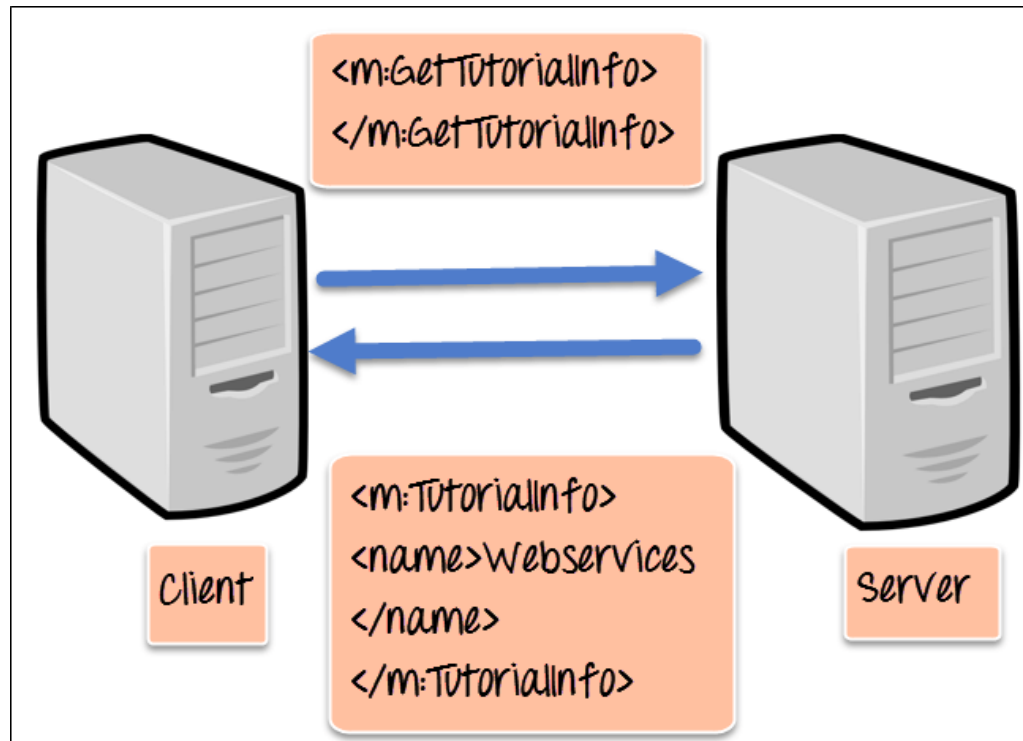
## Service Oriented Architecture (SOA)

- Adicionar uma aplicação com uns poucos web services não é nenhum problema.
- Esta integração limitada pode ser apropriada para uma experiência de aprendizado, ou para complementar a arquitetura de uma aplicação existente com uma peça de funcionalidade baseada em serviços que atende a um requisito específico do projeto.
- No entanto, isto não estabelece uma arquitetura orientada a serviço.

# Web Services e a arquitetura voltada a SOAP

## Service Oriented Architecture (SOA)

- Existe uma clara diferença entre
  - Uma aplicação que usa web service;
  - Uma aplicação baseada numa arquitetura orientada a serviços.



## Web Services e a arquitetura voltada a SOAP

- Uma **SOA (Serviço Orientado a Arquitetura)** é um modelo de projeto com um conceito profundamente amarrado à questão do encapsulamento de aplicação.
- A arquitetura resultante estabelece essencialmente um paradigma de projeto, no qual web services são os blocos de construção chave.
- Isto quer dizer que ao migrar a arquitetura da sua aplicação para uma SOA, estabelece-se um compromisso com os princípios de projeto de web services e a tecnologia correspondente, como partes fundamentais do seu ambiente técnico.

## Web Services e a arquitetura voltada a SOAP

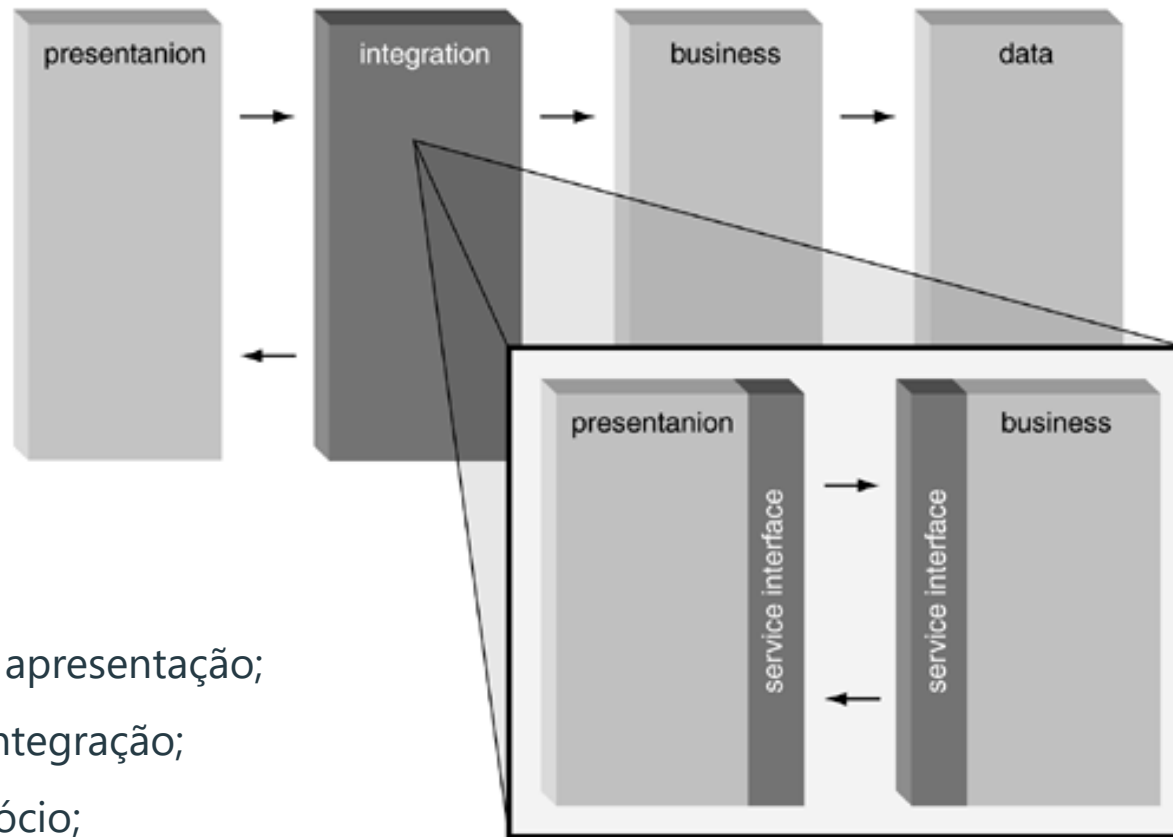
- Uma SOA baseada em XML web service é construída sobre camadas de tecnologia XML estabelecidas, focada em expor a lógica de aplicação existente como um serviço fracamente acoplado.
- Para apoiar este modelo, uma SOA promove o uso de um mecanismo de discovery por serviços via um service broker ou discovery agent.

## Web Services e a arquitetura voltada a SOAP

- SOA altera a arquitetura multicamada existente, ao introduzir uma camada lógica que, através do uso de interfaces programáticas padrão (providas pelo web services), estabelece um ponto comum de integração.
- Esta camada de integração de serviços constitui a base para um novo modelo que pode se estender além do escopo de uma única aplicação, unificando plataformas legadas díspares em um ambiente aberto.
- Quando web services são utilizados para integração cruzada de aplicações, elas se estabelecem como parte da infraestrutura do sistema.

# Web Services e a arquitetura voltada a SOAP

## Uma representação lógica de uma arquitetura orientada a serviços



**Presentation:** apresentação;

**Integration:** integração;

**Business:** negócio;

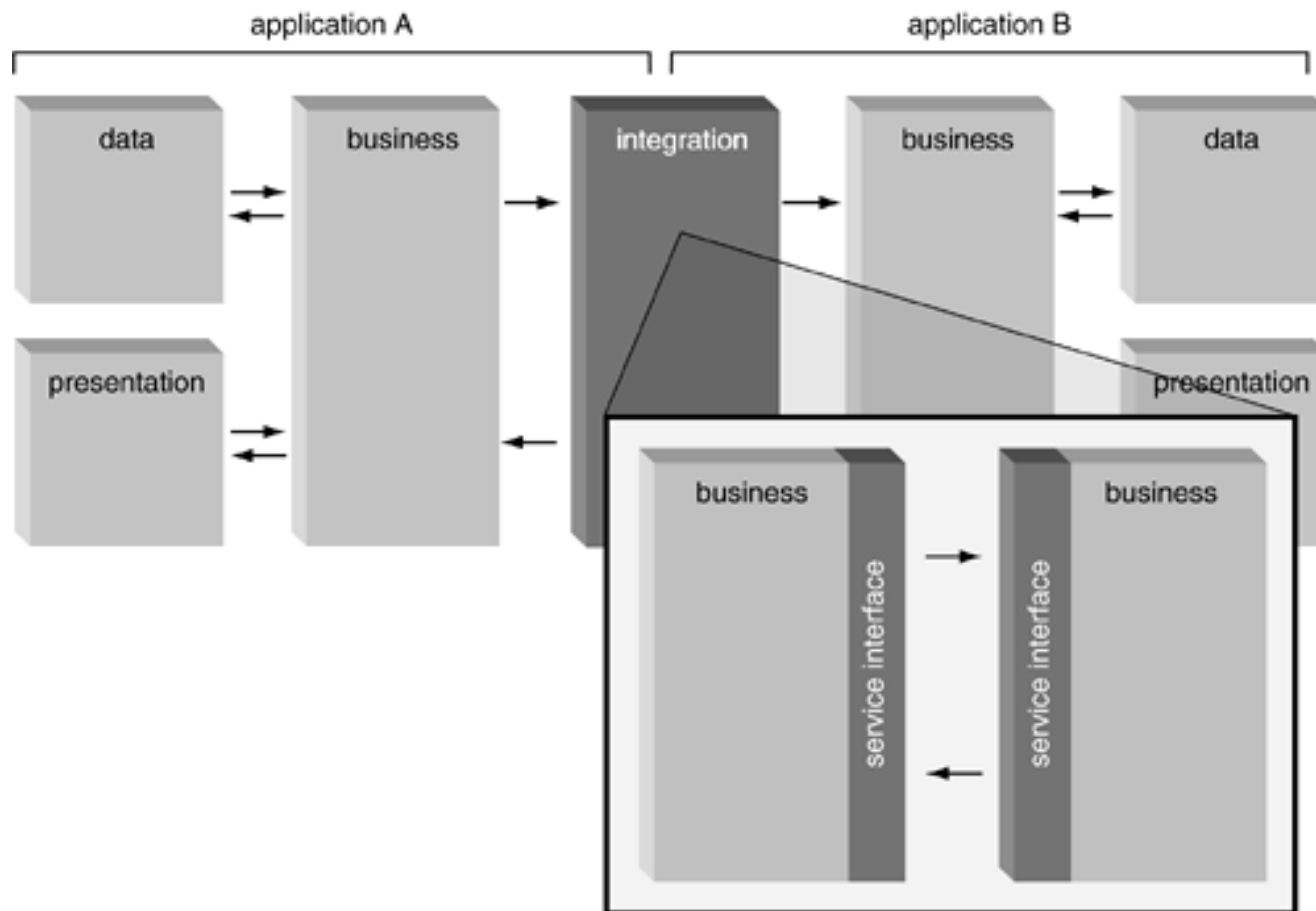
**Data:** dado;

**Service interface:** interface de serviço.



# Web Services e a arquitetura voltada a SOAP

Uma representação lógica de uma arquitetura de integração orientada a serviço



## Web Services e a arquitetura voltada a SOAP

- É importante se conscientizar quanto ao acréscimo de complexidade de projeto introduzido pelo SOA.
- Mais ainda do que em um ambiente n-camada, projetistas de aplicação devem considerar de uma forma completa como a introdução de serviços vai afetar dados existentes e modelos de negócio.

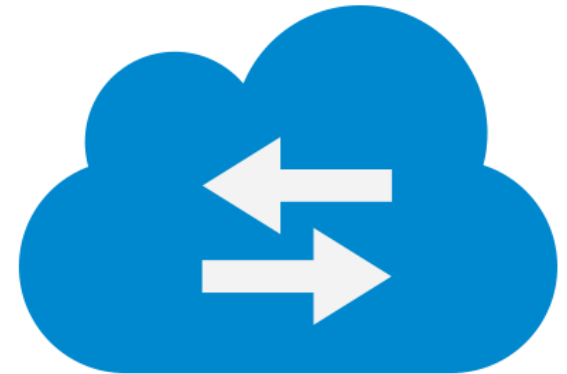


## Web Services e a arquitetura voltada a SOAP

- Na medida em que a utilização de serviços se diversifica, o significado dos requisitos de segurança e escalabilidade são amplificados.
- Ambientes orientados a serviço bem projetados tentarão vencer estes desafios com infraestrutura adequada, ao invés de utilizar soluções sob medida, específicas de aplicação.

## Funções de um web service

- Serviços podem assumir diferentes funções quando envolvidos em diversos cenários de interação. Dependendo do contexto pelo qual é visualizado, assim como o estado da tarefa rodando no momento, o mesmo web service pode trocar de papéis ou ser designado para múltiplos papéis simultâneos:



## Provedor de serviços

- Agindo como um provedor de serviços, um web service expõe uma interface pública através da qual pode ser chamado por solicitantes do serviço.
- Um provedor de serviços disponibiliza esta interface publicando uma descrição do serviço. Num modelo cliente-servidor, o provedor de serviço pode ser comparado ao servidor.
- O termo “*provedor de serviço*” pode também ser usado para descrever a organização ou ambiente que hospeda (provê) o web service.

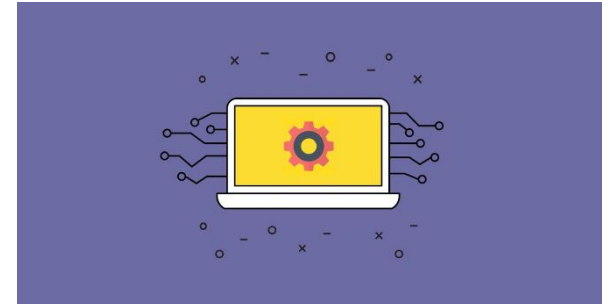
## Web Services e a arquitetura voltada a SOAP

- Um provedor de serviço pode também agir como um solicitante de serviço.
- Por exemplo, um web service pode atuar como um provedor de serviço quando um solicitante de serviço lhe pede para executar uma função.
- Pode então atuar como um solicitante de serviço quando mais tarde contata o solicitante de serviço original (agora agindo como um provedor de serviço) para solicitar informação de status.



## Solicitante de serviço

- Um solicitante de serviço é o remetente de uma mensagem web service ou o programa de software solicitando um web service específico.
- O solicitante de serviço é comparável ao cliente dentro de um modelo cliente-servidor padrão.
- Solicitantes de serviços são às vezes chamados de consumidores de serviços.



## Intermediário

- O papel de intermediário é assumido pelo web service quando ele recebe a mensagem de um solicitante de serviço e a passa adiante para o provedor de serviço.
- Neste caso, ele pode também agir como um provedor de serviço (recebendo a mensagem) e como um solicitante de serviço (passando adiante a mensagem).



## Web Services e a arquitetura voltada a SOAP

- Intermediários podem existir em muitas formas diferentes. Alguns são passivos e simplesmente retransmitem ou roteiam as mensagens, enquanto outros processam ativamente uma mensagem antes de repassá-la.
- Tipicamente, aos intermediários só é permitido o processamento e modificação do cabeçalho da mensagem. Para preservar a integridade da mensagem, seus dados não devem ser alterados.

## Remetente inicial

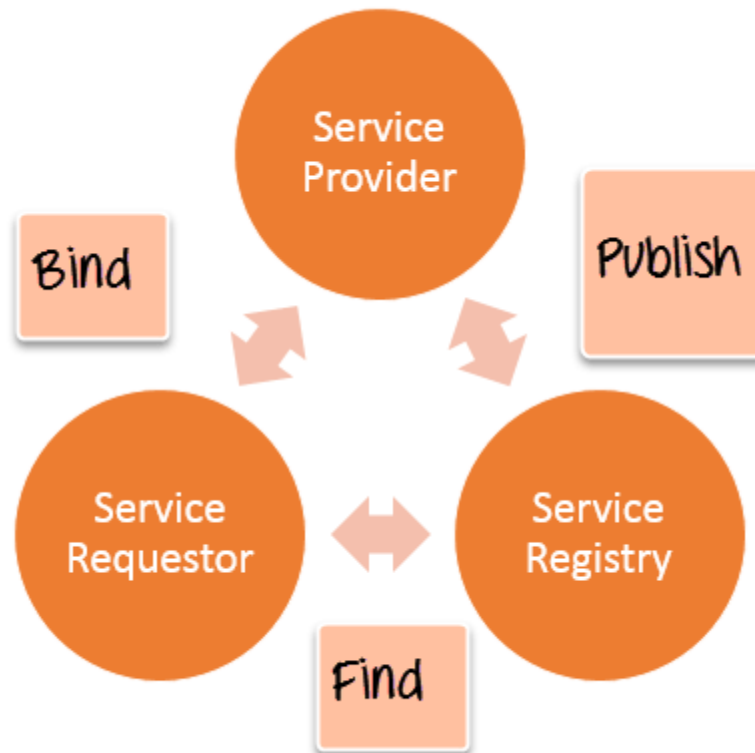
- Como o web service responsável por iniciar a transmissão da mensagem, remetentes iniciais também podem ser considerados solicitantes de serviço.
- Este termo existe para ajudar a diferenciar o primeiro web service que envia uma mensagem, dos intermediários também qualificados como solicitantes de serviço.

## Receptor final

- O último Web service a receber uma mensagem é o receptor final.
- Estes serviços representam o destino final de uma mensagem e também podem ser considerados provedores de serviço.

# Web Services e a arquitetura voltada a SOAP

## Arquitetura de Serviços da Web



## Atividade:

- Responda o questionário:

<https://forms.gle/2s5LfRkTPLizg3ZN6>

# **Muito Obrigado**

Até a Próxima Aula

Prof. Laércio Silva  
Email: [Indsilva@hotmail.com](mailto:Indsilva@hotmail.com)