

Service Oriented Architectures (SOA)

Marcelo Faro (mfal@cin.ufpe.br); Carlos Ferraz (cagf@cin.ufpe.br)

Sistemas Distribuídos – CIn/UFPE

22 de janeiro de 2004

O que é uma Arquitetura Orientada a Serviços (SOA)

- ❑ Uma Arquitetura de Software descreve os componentes de um sistema e a maneira como eles interagem (conectam)
 - "A arquitetura de software de um programa ou sistema de computação é a estrutura ou estruturas do sistema, o que compreende componentes de software, propriedades externas visíveis desses componentes e o relacionamento entre eles."(Bass, Clements e Kazman 1997)
- ❑ Service Oriented Architecture
 - modelo conceitual de arquitetura de software distribuído onde entidades (serviços, registros e contratos, por exemplo) são estruturadas para maximizar o baixo acoplamento e reuso
- ❑ SOA “quebra” aplicações monolíticas em serviços que podem estar distribuídos dentro de uma organização

Por que construir sistemas distribuídos?

- ❑ Limitações das arquiteturas monolíticas em atender a sistemas complexos ou de grande porte
- ❑ Principais motivações para distribuir sistemas
 - complexidade (tamanho) das novas aplicações
 - volume (demanda) que as novas aplicações precisam atender
 - integração com aplicações existentes
 - Dados e/ou usuários distribuídos
- ❑ Leis de More, Gilder e Metcalf
 - Poder de processamento, largura de banda e utilidade/valor da rede
- ❑ “Dividir para conquistar”
 - SOA deve ser visto como uma forma de estruturar “as partes divididas” para que possam ser compostas em novos sistemas

Mas o que são serviços...

- ❑ Serviços são funções de negócio bem definidas e auto-contidas que recebem requisições de clientes (consumidores)
 - Exemplos: controle de acesso, cotação de ações, conversão monetária, etc.
- ❑ Serviços são implantados uma vez e podem ser acessados por diversas aplicações, enquanto componentes precisam ser instalados dentro do contexto de cada aplicação
- ❑ Evolução do modelo de componentes

Por que usar SOA?

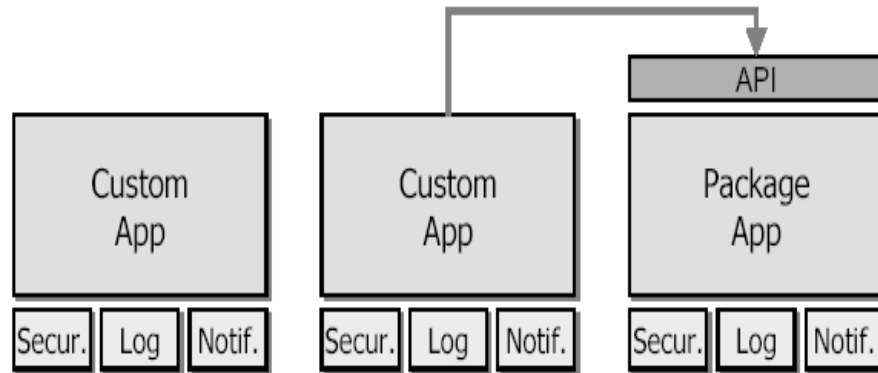
- ❑ Utilizando SOA é possível modelar a organização como um coleção de serviços disponíveis através da organização
- ❑ Aplicações de negócio monolíticas, como ERPs (Enterprise Resource Planning), podem ser decompostos em serviços auto-contidos que executam funções específicas
- ❑ Funções específicas podem ser compartilhadas entre aplicações permitindo que novas aplicações sejam desenvolvidas a partir da composição de serviços já existentes ao invés de começar do zero
 - Reuso (\$\$\$)
- ❑ SOA favorece a integração de aplicações
 - adoção de padrões
- ❑ Novos modelos de negócio
 - e.g., Service-outsourcing

Mudança de Paradigma

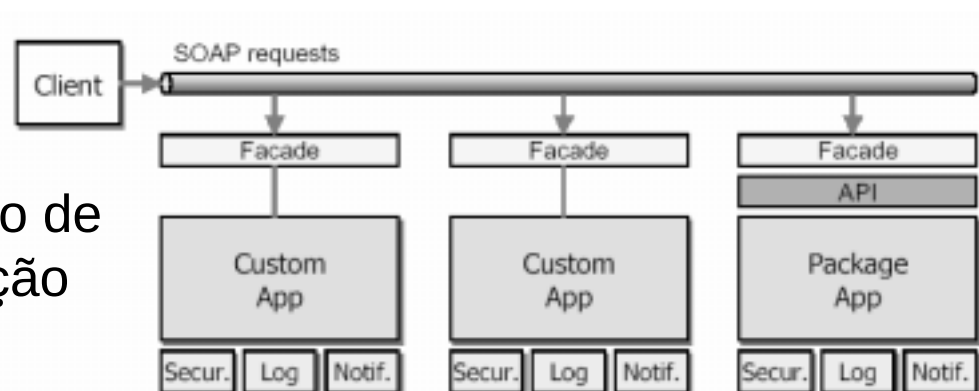
- ❑ Evolução ...
 - Modelo de desenvolvimento de aplicações baseado em **Componentes**
 - Component-based development (CBD)
- ❑ Para ...
 - Modelo de desenvolvimento de aplicações baseado em **Serviços**
 - Service-based development (SBD)
- ❑ Serviços podem (“devem”) ser desenvolvidos baseados em componentes

Mudança de Paradigma

- Modelo atual

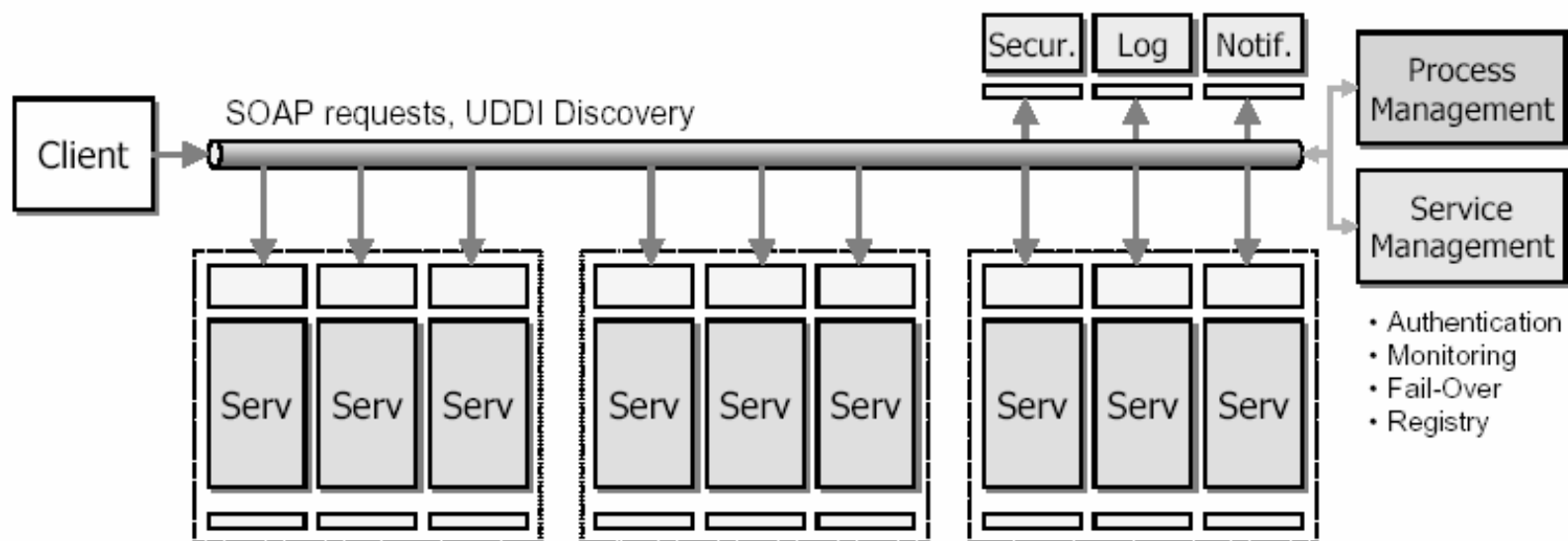


- Modelo de transição



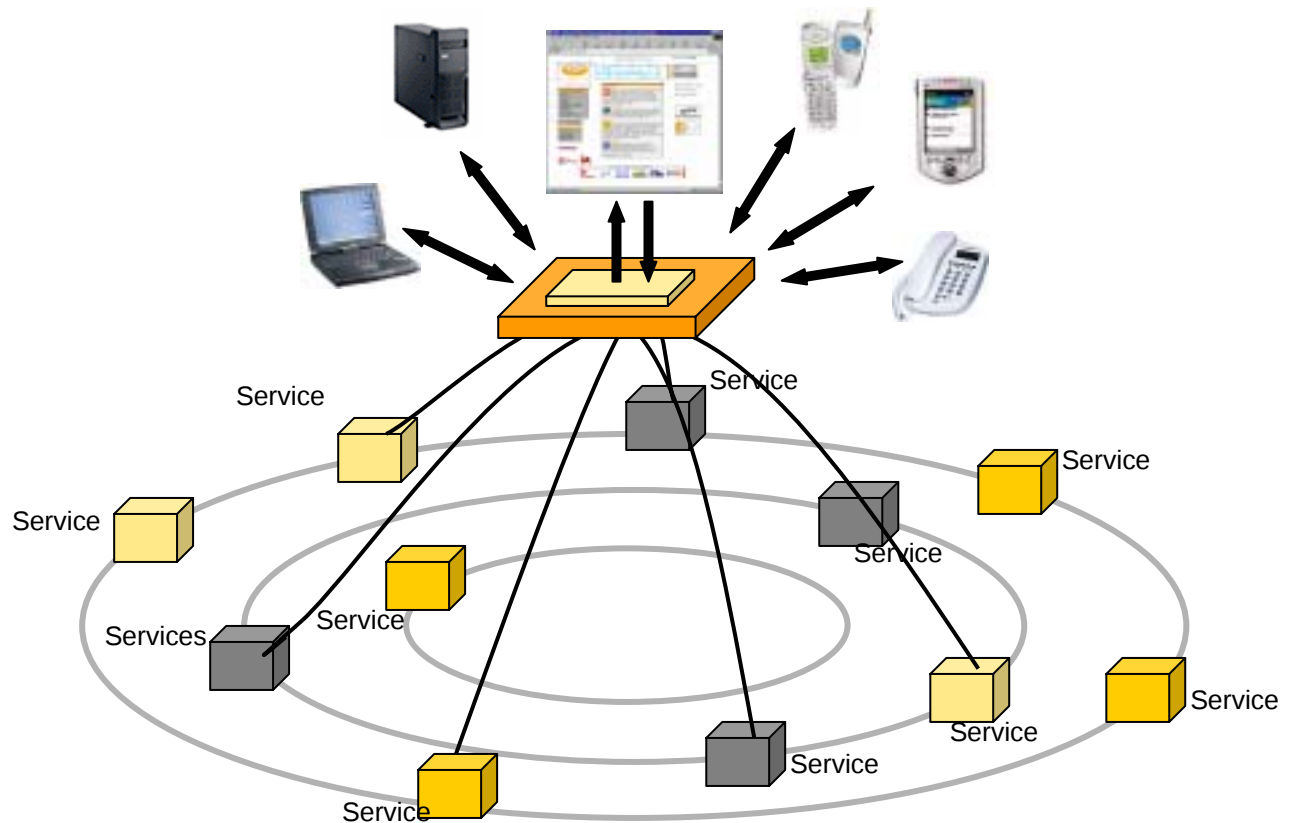
Mudança de Paradigma

SOA



Por que usar SOA?

Novas aplicações serão construídas compondo serviços



Implementando SOA

- ❑ Serviços devem ser
 - projetados (definição da interface)
 - implementados
 - implantados (deployed)
 - e publicados
- ❑ Clientes localizam e consomem os serviços publicados utilizando protocolos e formatos de dados padrões
- ❑ Paradigma "find, bind and execute"

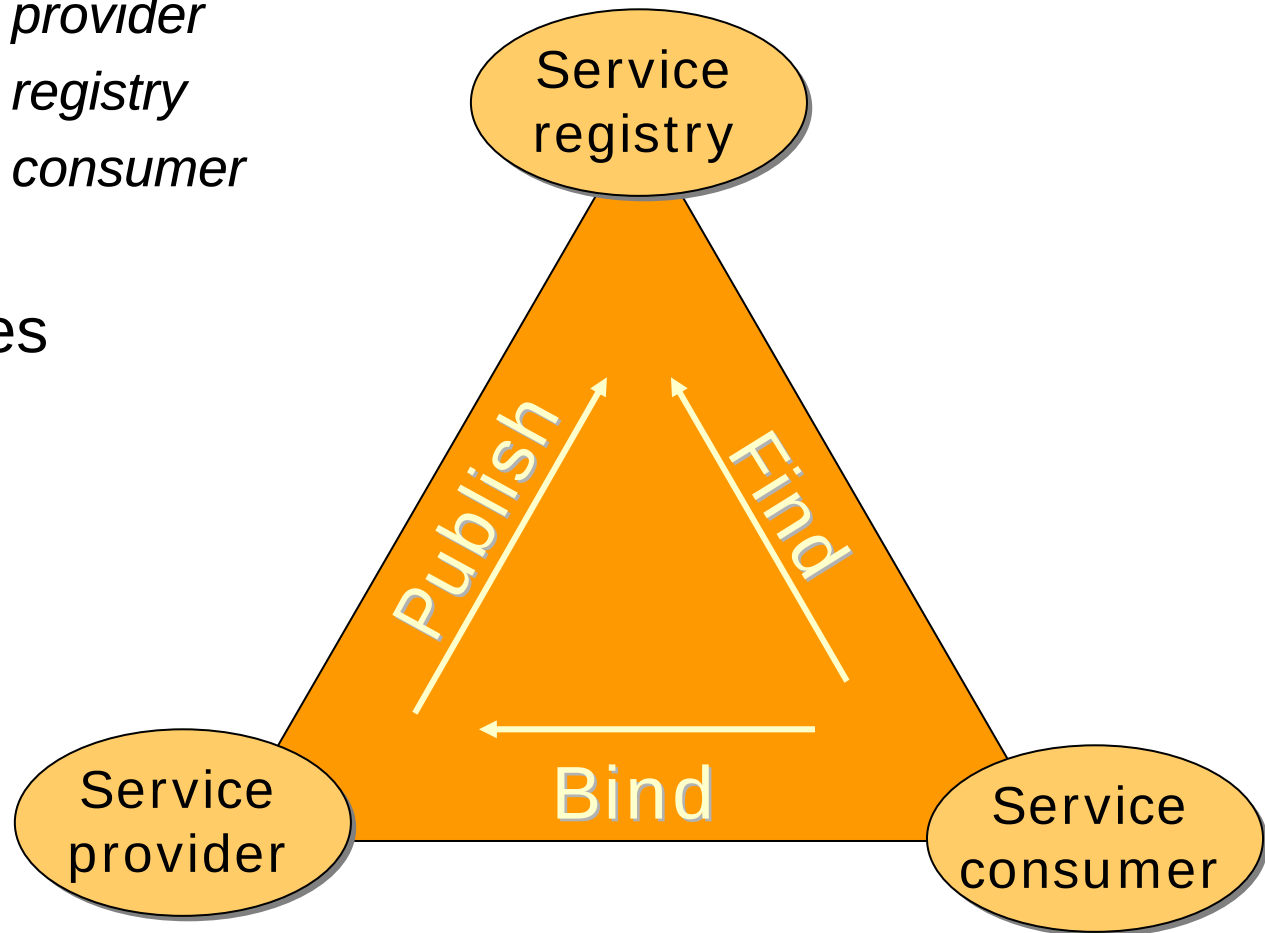
Paradigma "find, bind and execute"

□ Papéis

- *Service provider*
- *Service registry*
- *Service consumer*

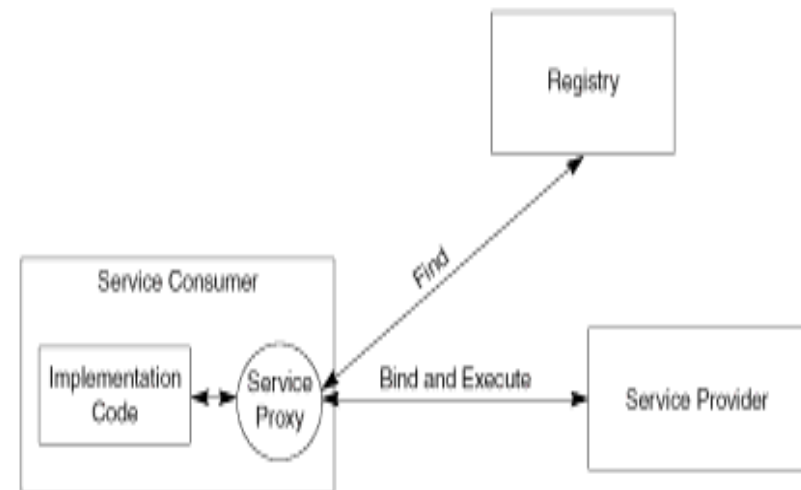
□ Operações

- *Publish*
- *Find*
- *Bind*



Conceitos

- ❑ Service contract
 - Especificação de como o cliente do serviço (consumer) vai interagir com o provedor (provider): interfaces
 - Pode incluir pré e pós condições para a execução do serviço
- ❑ Service provider
 - Componente de software responsável pela implementação do serviço
- ❑ Service consumer
 - Cliente que consome (requisita) o serviço
- ❑ Service proxy (opcional)
 - É um objeto fornecido pelo provedor para facilitar a interação do cliente
 - Implementa as interfaces e delega o serviço para o provedor
- ❑ Service registry
 - responsável pelo registro e consultas a serviços publicados. Funciona como um diretório ou página amarela
- ❑ Service lease
 - Corresponde ao direito de uso do serviço dado ao cliente por um determinado tempo



Características de SOAs

- ❑ Serviços podem ser descobertos (find) e executados (bind) dinamicamente
- ❑ Serviços são auto-contidos e modulares
- ❑ Serviços favorecem a interoperabilidade
- ❑ Serviços possuem baixo acoplamento
- ❑ Serviços possuem uma interface acessível por um endereço na rede (network-addressable)
- ❑ Serviços possuem transparência de localização
- ❑ Serviços podem ser compostos
- ❑ Serviços “podem” ser tarifados (service-outsourcing)
- ❑ SOA suporta self-healing

Quais os benefícios do uso de SOA

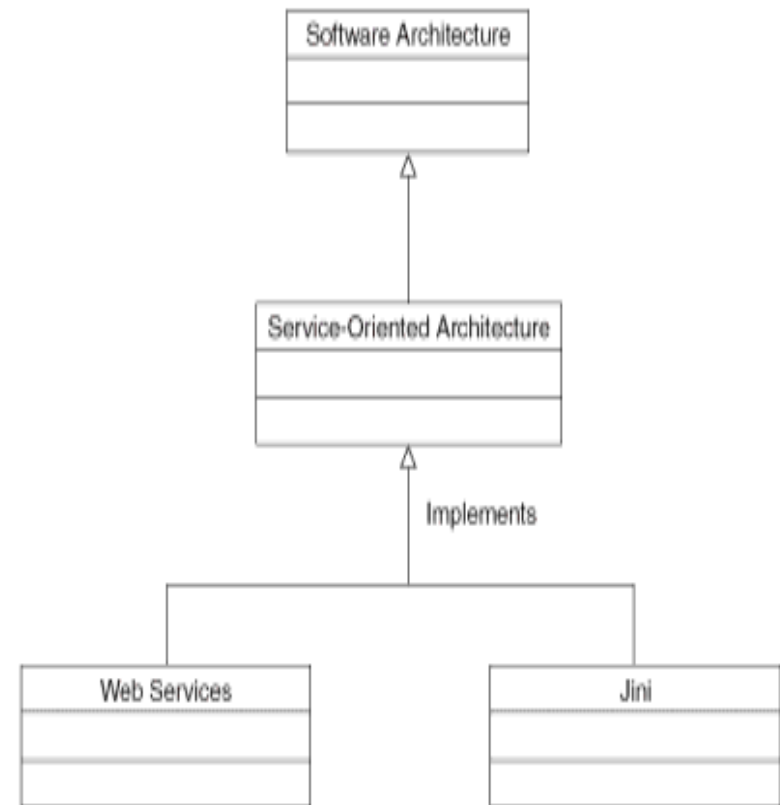
- ❑ Favorece a integração de aplicações
 - Adoção de padrões para comunicação e formato das mensagens
- ❑ Promove reuso
 - Componentes de serviço podem ser compostos para gerar novas aplicações
- ❑ Promove o baixo acoplamento
 - Interfaces, formatos de dados, transparência de localização
 - Facilidade em trocar de provedor de serviço
- ❑ Maior manutenibilidade
 - SOA promove a separação entre a interface e a implementação do serviço ("o quê" do "como")
- ❑ Maior extensibilidade
 - Um único serviço pode suportar diversas interfaces

Algumas dificuldades para implementar...

- ❑ Definição de protocolos e formatos padrões
 - Comunicação, segurança, transações, confiabilidade, etc.
- ❑ Suporte por Middleware
- ❑ Definição das interfaces

Tecnologias para implementar SOA

- ❑ Basicamente qualquer tecnologia que suporte o desenvolvimento de sistemas distribuídos
 - Jini, CORBA, Web Services, etc.
- ❑ Diferentes tecnologias oferecem suporte diferenciado com algumas limitações
- ❑ Web Services vêm promovendo a utilização de SOA
 - Padrão adotado pela indústria



Jini

- ❑ Jini é uma plataforma de distribuição que oferece uma estrutura de software robusta e flexível para o desenvolvimento de sistemas distribuídos
 - concebida inicialmente como uma solução de conectividade entre dispositivos eletrônicos
- ❑ Componentes eletrônicos poderiam ser conectados automaticamente a medida que aparecem na rede (Plug and Play)
- ❑ Visão
 - Você pode andar com um aparelho habilitado com tecnologia Jini - seja uma câmera digital, uma impressora, um PDA ou um telefone celular - conectá-lo a uma rede TCP/IP, e automaticamente ver e usar os serviços dos outros dispositivos Jini-habilitados na sua vizinhança

Web Services

- ❑ Web Services constituem uma **arquitetura** computacional **distribuída** baseada em **serviços auto-descritivos** que podem ser publicados, localizados e executados através da Web
 - utilizam XML e um conjunto de padrões e protocolos para integrar aplicações e sistemas heterogêneos através da Web
- ❑ Qualquer tecnologia com paradigma orientado a serviço
 - é basicamente XML sobre HTTP
 - XML: porque o mercado concorda
 - HTTP: livre acesso através de *firewalls*
- ❑ Padrões propostos
 - SOAP: transporte de dados (XML/HTTP)
 - WSDL: descrição dos serviços
 - UDDI: registro e busca de serviços

SOAP

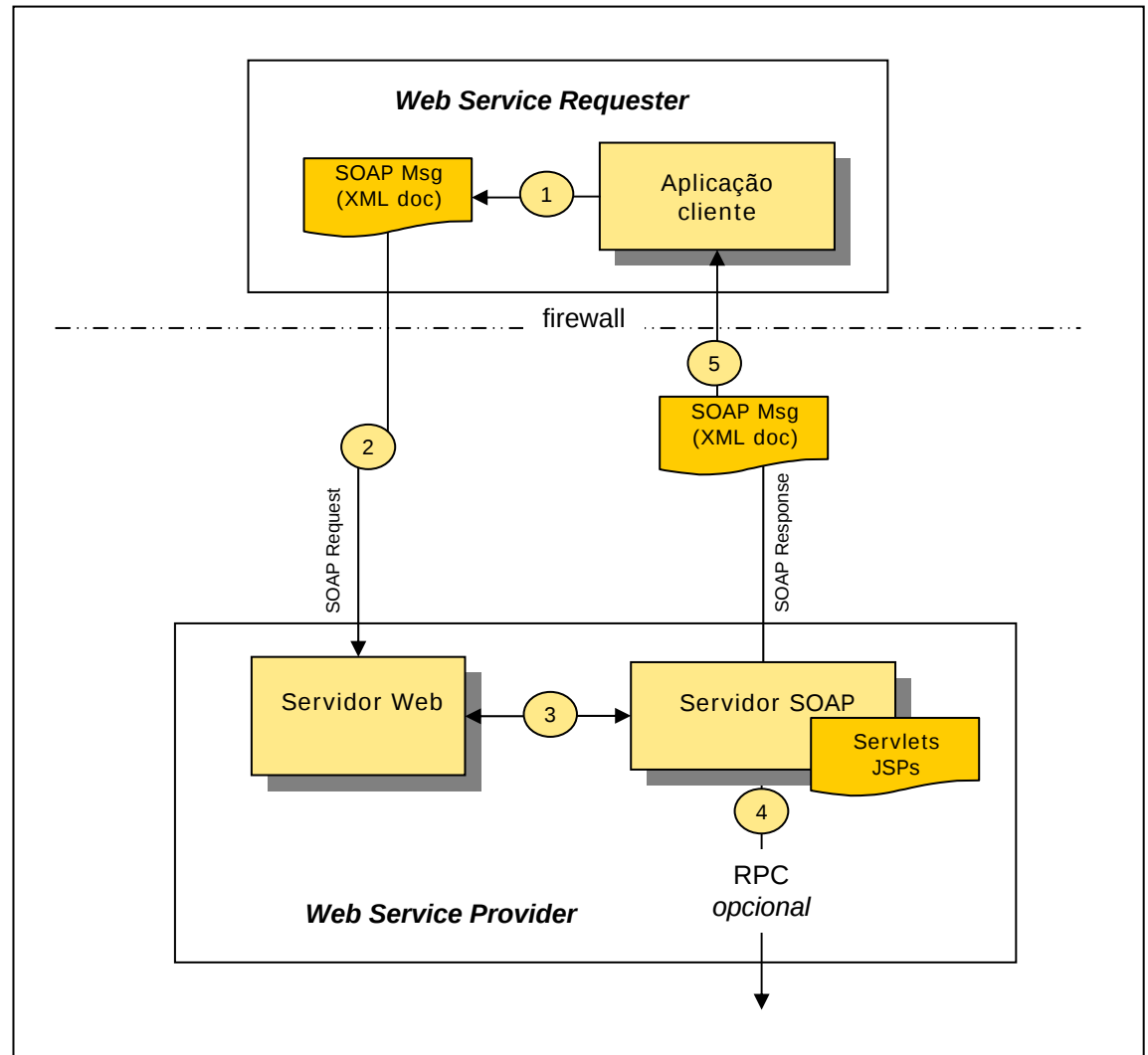
- ❑ Simple Object Access Protocol
 - protocolo baseado em XML para o envio de mensagens e execução de chamadas remotas em ambientes distribuídos
- ❑ Independente de linguagem ou plataforma
 - implementações em mais de 60 linguagens e 20 plataformas
- ❑ Usado como referência pelo W3C para a definição do XML Protocol (XMLP)
 - <http://www.w3.org/TR/SOAP>

SOAP

- ❑ Modelo de mensagens independente do protocolo de transporte
 - suporte para HTTP
- ❑ Mensagens são modeladas como um envelope SOAP
 - *header e body*
- ❑ Mensagens *one-way* e RPC (*Remote Procedure Call*)
- ❑ Define regras de codificação para expressar tipos de dados definidos na aplicação
 - exemplo: dados XML para objetos Java, e vice-versa

SOAP

Exemplo



SOAP

- Exemplo de envelope SOAP
 - *request*

```
<SOAP-ENV: Envelope
  xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV: encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <t:Transaction xmlns:t="some-URI" SOAP-ENV:mustUnderstand="1" >
      5
    </t:Transaction>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="some-URI">
      <symbol>DEF</Symbol>
    </m: GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-Envelope>
```

SOAP

- ❑ Exemplo de envelope SOAP
 - *response*

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

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

<SOAP-ENV:Header>

<t:Transaction xmlns:t="some-URI" xsi:type="xsd:int" mustUnderstand="1">
5

</t:Transaction>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<m:GetLastTradePriceResponse xmlns:m="Some-URI">

<Price>34.5</Price>

</m:GetLastTradePriceResponse>

</SOAP-ENV:Body>

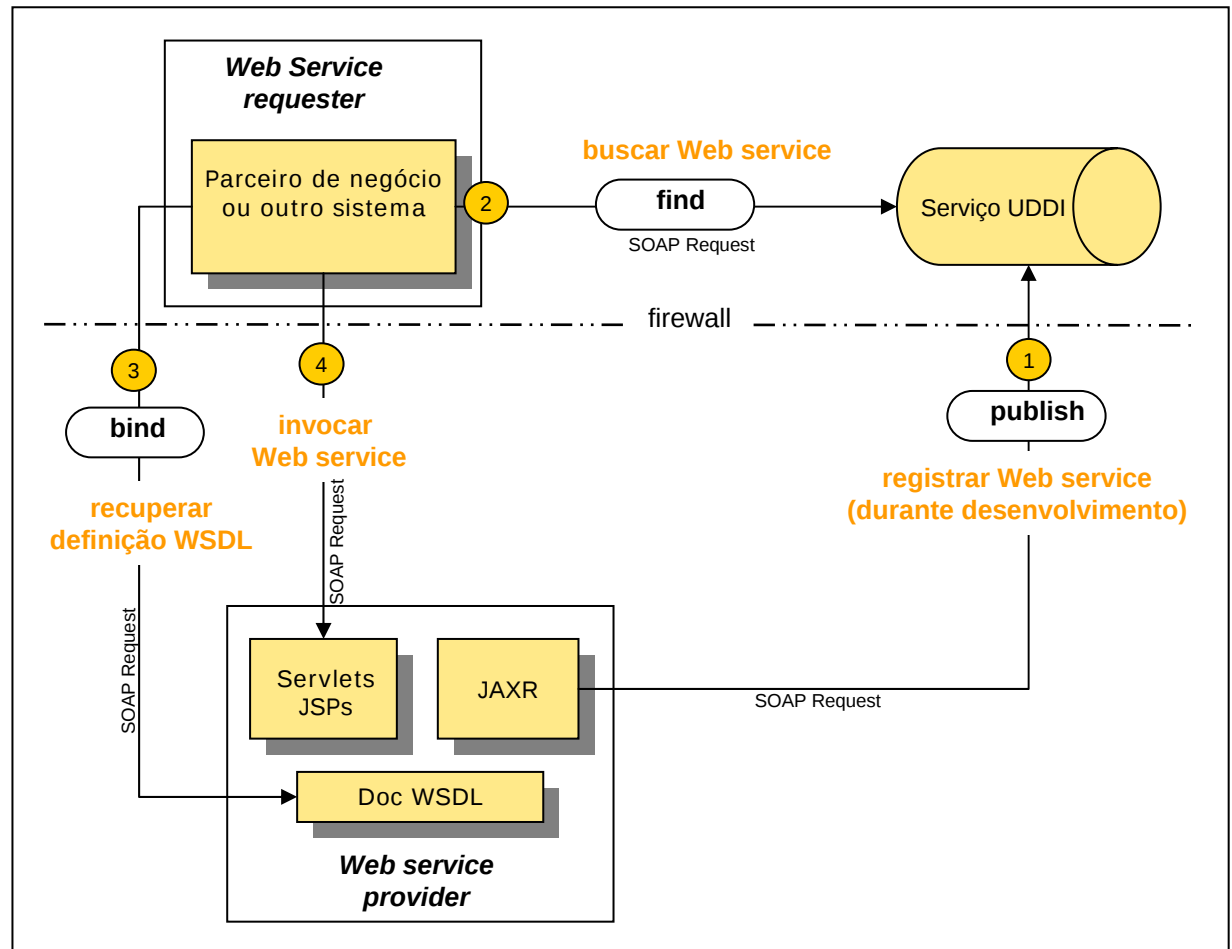
</SOAP-ENV:Envelope>

WSDL

- ❑ *Web Services Definition Language*
 - linguagem baseada em XML para a descrição de serviços disponíveis em redes de computadores (*Machine-readable description*)
- ❑ Usada para descobrir a localização dos serviços, suas chamadas de funções e como acessá-los
 - idealmente deve prover todas as informações necessárias para acessar o serviço (programaticamente)
- ❑ Define
 - interfaces (i.e. operações, tipos de dados, etc.)
 - protocolos de acesso (i.e. SOAP sobre HTTP)
 - localização dos serviços (i.e URL do Web Service)
- ❑ Padronização proposta ao W3C
 - <http://www.w3.org/TR/wsdl>

WSDL

Exemplo



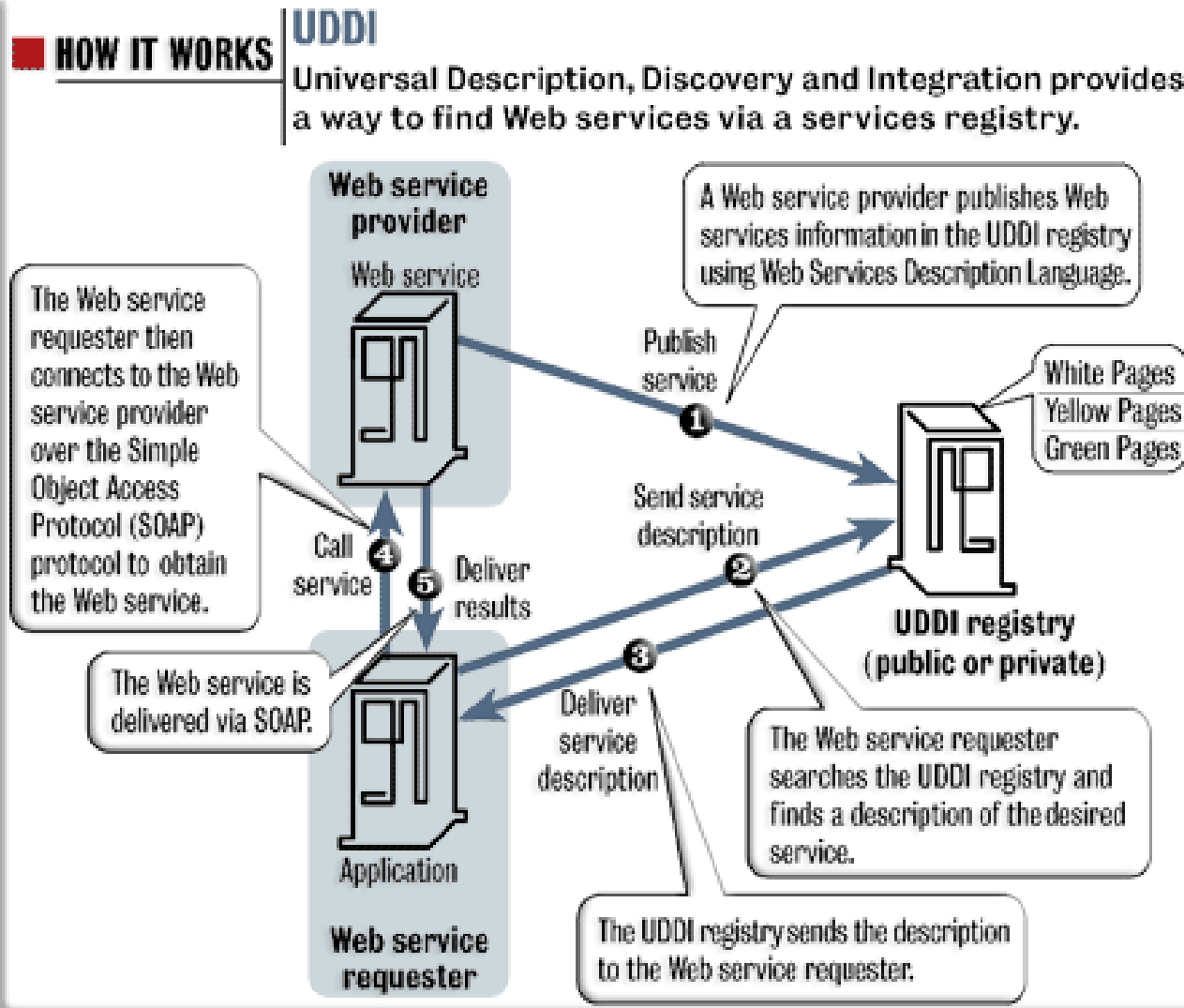
UDDI

- ❑ *Universal Description, Discovery and Integration*
 - especificação com definição dos mecanismos para publicação e busca de Web Services em repositórios de serviços
 - baseado em XML e SOAP
 - XML: estruturas de dados que definem os Web Services em um UDDI Registry
 - SOAP: API para comunicação com um UDDI Registry
 - a especificação UDDI ainda não foi submetida a organizações de padronização
 - UDDI Project www.uddi.org

UDDI

- ❑ Organização de registros UDDI
 - Páginas brancas
 - informações de contato (nome, endereço, URLs, etc.)
 -
 - Páginas amarelas
 - categorização dos serviços baseados em taxonomias padrões: NAICS, UM/SPC e GSC
 - registra informações como códigos de indústrias, códigos de produtos, códigos geográficos, etc.
 - Páginas verdes
 - informações técnicas sobre o serviço publicado, como especificações do Web Service e localização
- ❑ Descoberta dos serviços
 - busca automática
 - UDDI APIs
 - busca manual
 - engenho de busca no site do UDDI Registry

UDDI



Soluções e ferramentas

- ❑ Plataformas para implementação de SOA usando Web Services
 - J2EE (Java Enterprise Edition)
 - MS .Net

- ❑ Existem no mercado diversos servidores e ferramentas para desenvolver e publicar serviços na forma de Web Services
 - IBM Websphere, Sun One, Borland JBuilder e BES, JBoss, Axis, JWSDP, Glue, etc.

- ❑ Exemplos de serviços na Web
 - MS Passport
 - Amazon
 - XMethods (www.xmethods.com)

Infravida/CENAS

□ CENAS

- Centro de Excelência no Desenvolvimento de Novas Aplicações
- Centro Mundial de Excelência na Arquitetura Sun One
 - Parceria entre a Sun e o CESAR
- P&D com SOA em Java utilizando Web Services em Plataforma Sun One



□ Projeto InfraVida

- Infra-estrutura de Vídeo Digital para Aplicações de Tele-medicina
- Parceria CENAS, CIn e outras instituições de ensino e pesquisa



InfraVIDA

