Web services em Sistemas Distribuídos

Ricardo Costa – rcosta@estg.ipp.pt

Web Services: Motivação

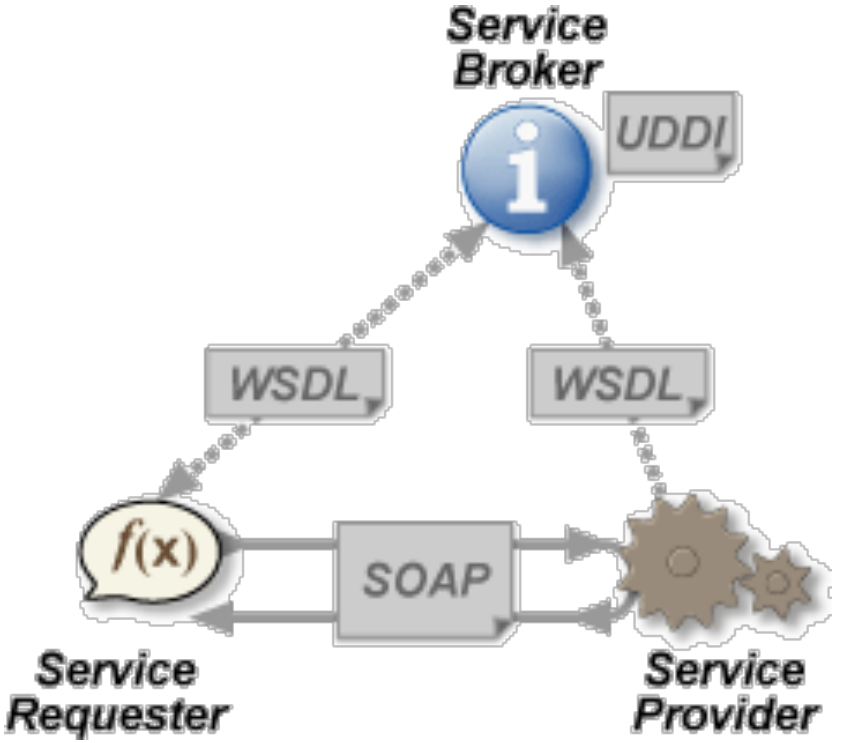
- Modelo para acesso a servidores: invocação remota
- Desenhado para garantir inter-operabilidade

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using ***HTTP*** *with an* ***XML serialization*** *in conjunction with other* ***Web-related standards***

- Protocolo: HTTP e HTTPS (eventualmente SMTP)
- Referências: URL/URI
- Representação dos dados: XML

Componentes Básicos

- SOAP: protocolo de invocação remota
- WSDL: linguagem de especificação de serviços
- UDDI: serviço de registo



SOAP

- Protocolo para trocar mensagens XML
  - Inclui definição do formato das mensagens a trocar
  - Inclui um mecanismo de ligação das mensagens SOAP com o protocolo de transporte usado: HTTP ou HTTPS (ou SMTP, …)
  - Inclui mecanismo para tratar falhas

SOAP: interações

- **Oneway:** mensagem unidirecional do cliente para o servidor
- **Pedido-resposta:** interação cliente-servidor-cliente
- **Notificação:** interação unidirecional servidor-cliente
  - E.g. callback, notificação
- **Notificação-resposta:** interação servidor-cliente-servidor

SOAP: segurança

- É possível efetuar pedidos SOAP usando canais seguros (HTTPS)
  - Permite autentica o servidor (i.e., que o cliente tenha a certeza de estar a falar com o servidor)
  - Autenticação do cliente ao nível do canal seguro é opcional

WSDL – IDL para Web Services

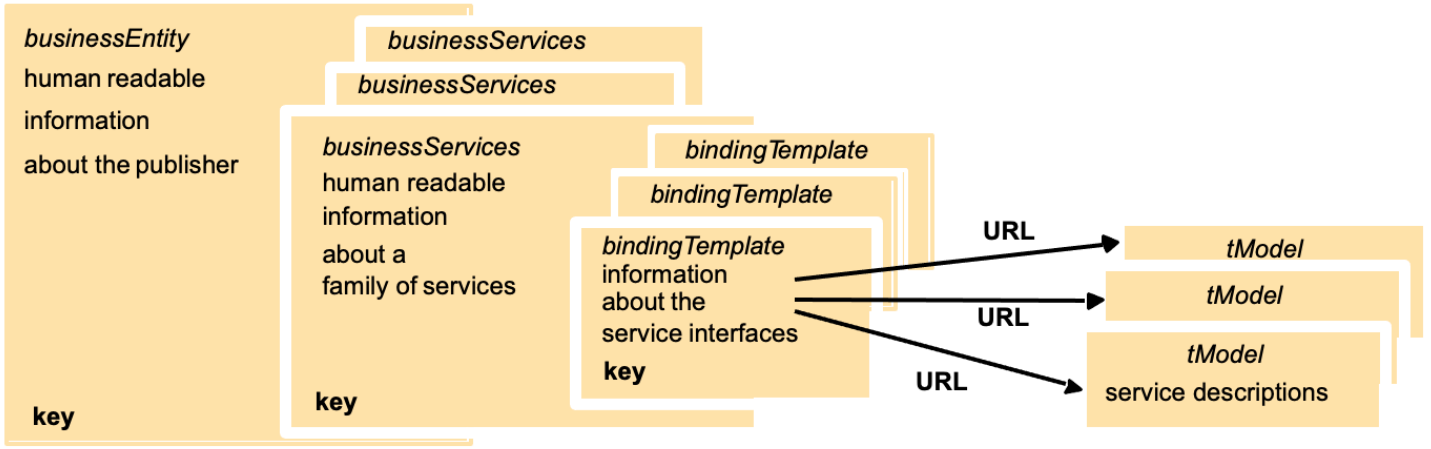
- Define a interface do serviço, indicando quais as operações disponíveis
- Define as mensagens trocadas na interação (e.g. na invocação duma operação, quais as mensagens trocadas)
- Permite definir a forma de representação dos dados e a forma de aceder ao serviço
- Especificação WSDL bastante verbosa – normalmente criada a partir de interface ou código do servidor
  - Em Java e .NET existem ferramentas para criar especificação a partir de interfaces Java
  - Sistemas de desenvolvimento possuem wizards que simplificam tarefa

WSDL – Elementos

- No seu conjunto, um documento WSDL comporta os seguintes elementos:
  - **Types** - definições de tipos de dados, num dado sistema (e.g., XSD)
  - **Messages** - definição abstracta dos dados trocados nas operações
  - **Operation** - definição abstracta das acções suportadas pelo serviço
  - **Port Type** - conjunto abstracto das operações suportadas por um ou mais endpoints
  - **Binding** - uma especificação concreta do protocolo e formato de dados usados por um port type.
  - **Port** - um endpoint concreto que combina um endereço e um binding particulares.
  - **Serviço** - um conjunto de endpoints/ports relacionados.

UDDI

- Protocolo de ligação ou binding entre o cliente e o servidor
- Os fornecedores dos serviços publicam a respetiva interface
- O protocolo de inspeção permite verificar se um dado serviço existe baseado na sua identificação
- O UDDI permite encontrar o serviço baseado na sua definição – capability lookup



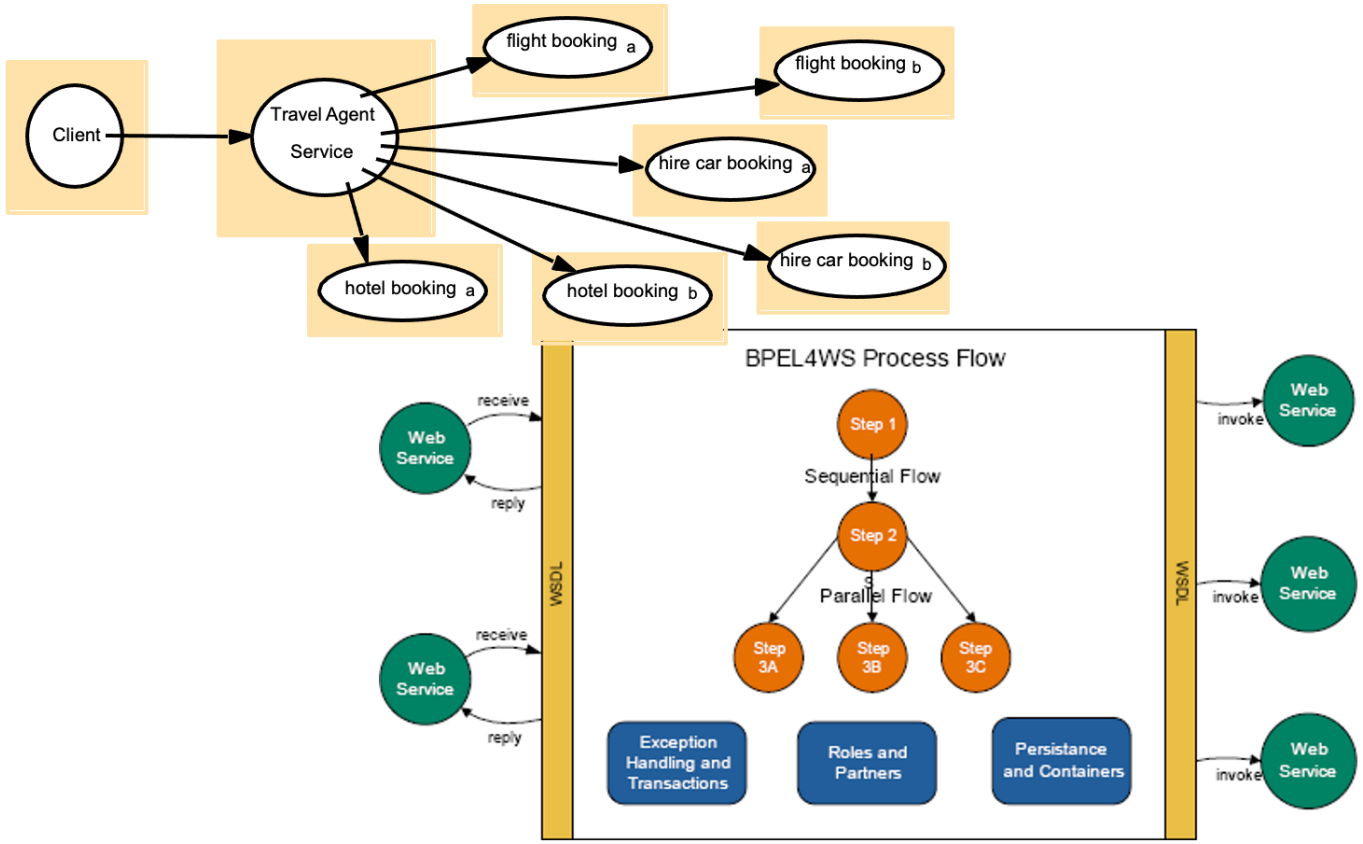
Clientes

- O cliente pode ter um static proxy criado antes da execução (static stub) a partir do WSDL
- O cliente pode também usar um dynamic proxy uma classe que é criada durante a execução a partir do WSDL

WS-*

- **WS-Reliability**: especificação que permite introduzir fiabilidade nas invocações remotas (com as diferentes semânticas)
  - WS-ReliableMessaging
- **WS-Security**: define como efetuar invocações seguras
- **WS-Coordination**: fornece enquadramento para coordenar as ações de aplicações distribuídas (coreografia de web services)
- **WS-AtomicTransaction**: fornece coordenação transacional (distribuída) entre múltiplos web services
  - E.g.: BPEL4WS, WSBPEL linguagem de orquestração

WS-Coordination



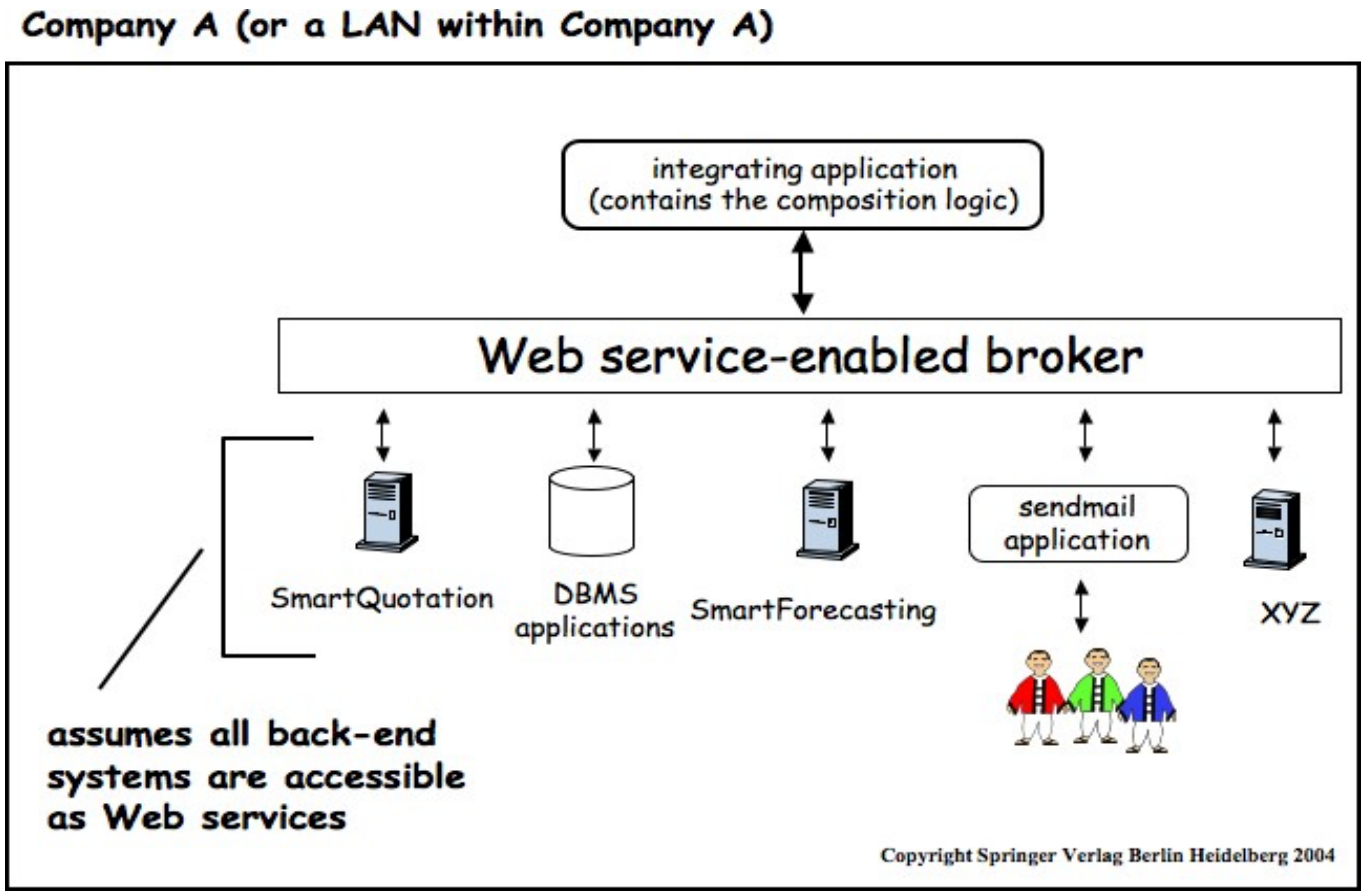
WS : Resumo

- Standard na indústria porque apresenta uma solução para integrar diferentes aplicações
- Permite:
  - Utilização de standards
    - HTTP, XML
  - Reutilização de serviços (arquiteturas orientadas para os serviços)
    - WS-Coordination
  - Modificação parcial/evolução independente dos serviços
    - WSDL
  - Disponibilidade, assincronismo
    - WS-Eventing

WS : Utilizações Possíveis

- Interoperação entre instituições
  - Utilização tende a ser limitada
  - Promessas de todos os serviços disponíveis para todos acabaram por não se concretizar
- Interoperação entre sistemas duma mesma instituição
  - Utilização com forte implantação

Web Services



WS : Problemas

- Desempenho:
  - Complexidade do XML
  - Difícil fazer caching: noção de operação + estado
- Complexidade
  - Necessário utilização de ferramentas de suporte

REST

- Aproximação: vê uma aplicação como uma coleção de recursos
  - Um recuso é identificado por um URI/URL
  - Um URL devolve um documento com a representação do recurso Podem-se fazer referências a outros recursos usando ligações (links)
- Estilo arquitetural, não um sistema de desenvolvimento
- Aproximação proposta por Roy Fielding na sua tese de doutoramento
  - Não como uma alternativa aos web services, mas como uma forma de aceder a informação

Roy Fielding

UNIVERSITY OF CALIFORNIA,
IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

Dissertation Committee:
Professor Richard N. Taylor, Chair
Professor Mark S. Ackerman
Professor David S. Rosenblum

2000

- Author of HTTP specification
- Co-founder of Apache HTTP server
- Currently working at Adobe

REST: Princípios de Desenho

- Protocolo cliente/servidor stateless: cada pedido contém toda a informação necessária para ser processado – objetivo: tornar o sistema simples.
- Recursos – no sistema tudo são recursos, identificados por um URI/URL.
  - Recursos tipicamente armazenados num formato estruturado que suporta hiperligações (e.g. XML)
- Interface uniforme: todos os recursos são acedidos por um conjunto de operações bem-definidas. Em HTTP: POST, GET, PUT, DELETE (equiv. criar, ler, atualizar, remover).
- Cache: para melhorar desempenho, respostas podem ser etiquetadas como permitindo ou não caching.

REST vs. RPCs/Web Services

- Nos sistemas de RPCs/Web Services a ênfase é nas operações que podem ser invocadas
- Nos sistemas REST, a ênfase é nos recursos, na sua representação e em como estes são afetados por um conjunto de métodos standard

Codificação dos Dados: XML vs. JSON

- A informação transmitida nos pedidos e nas respostas é codificada tipicamente em:
  - XML
  - Json

Codificação dos Dados: XML vs. JSON

- A informação transmitida nos pedidos e nas respostas é codificada tipicamente em:
  - XML
  - Json

- Json tem ganho popularidade porque permite manipulação simples em Javascript

REST : Protocolos

- Baseado em protocolos standard Comunicação
  - HTTP (tipicamente)
- Identificação de recursos
  - URL/URI
- Representação dos recursos
  - Json, XML

REST : Notas Soltas

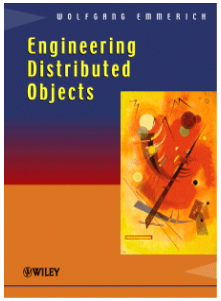
- Tem ganho popularidade
  - Muitos serviços web disponibilizados segundo este modelo Grandes vantagens: simplicidade e desempenho
- Serviços disponibilizados num modelo REST nem sempre aderem completamente aos princípios REST
  - Difícil disponibilizar número elevado de métodos

http://api.flickr.com/services/rest/?method=flickr.photos.search&appid=sdfjkshf

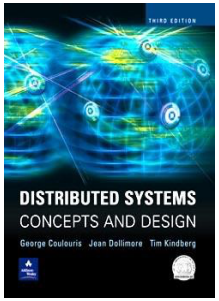
WebServices ou REST?

- Questão em aberto
- REST:
  - Mais simples
  - Mais eficiente
  - Complicado implementar serviços complexos
- Web services
  - Mais complexo
  - Grande suporte da indústria
- E.g.:
  - http://www.oreillynet.com/pub/wlg/3005

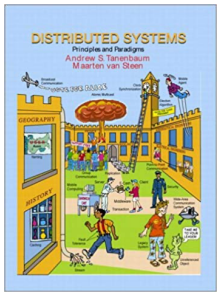
Bibliografia



From: Wolfgang Emmerich
Engineering Distributed Objects
John Wiley & Sons, Ltd 2000



From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design
Edition 4 © Addison-Wesley 2005



From: Andrew S., Tanembaum and Van Steen, Maarten
Distributed Systems: Principles and Paradigms
Edition 2 © Pearson 2013

Questões?