Web services

- Um web service é qualquer software que está disponível através da Internet através de uma interface XML.

- XML é utilizado para codificar toda a comunicação de/para um web service.

- Web services são as peças essenciais para se construir as aplicações distribuídas do futuro.

Vantagens do XML

- Uma aplicação cliente invoca um web service enviando-lhe uma mensagem XML.

- Depois espera por uma resposta que também vem codificada em XML.

- Comunicação feita em XML implica que os Web Services sejam independentes do sistema operativo e da linguagem de aplicação em que são feitos.

- Podemos ter programas Linux/UNIX feitos em Python a comunicar com programas Windows feitos em Visual Basic.

Exemplos de aplicações

- website de viagens
  - acede a um web service de uma rent-a-car
  - acede a web services de agência hoteleiras
  - acede a web services de companhias aéreas
- cheapCamaras.com
  - acede a web services de várias revendedores de máquinas fotográficas.
  - consegue ver em cada momento, o preço mais barato para cada modelo.

WSDL (Web Service Description Language)

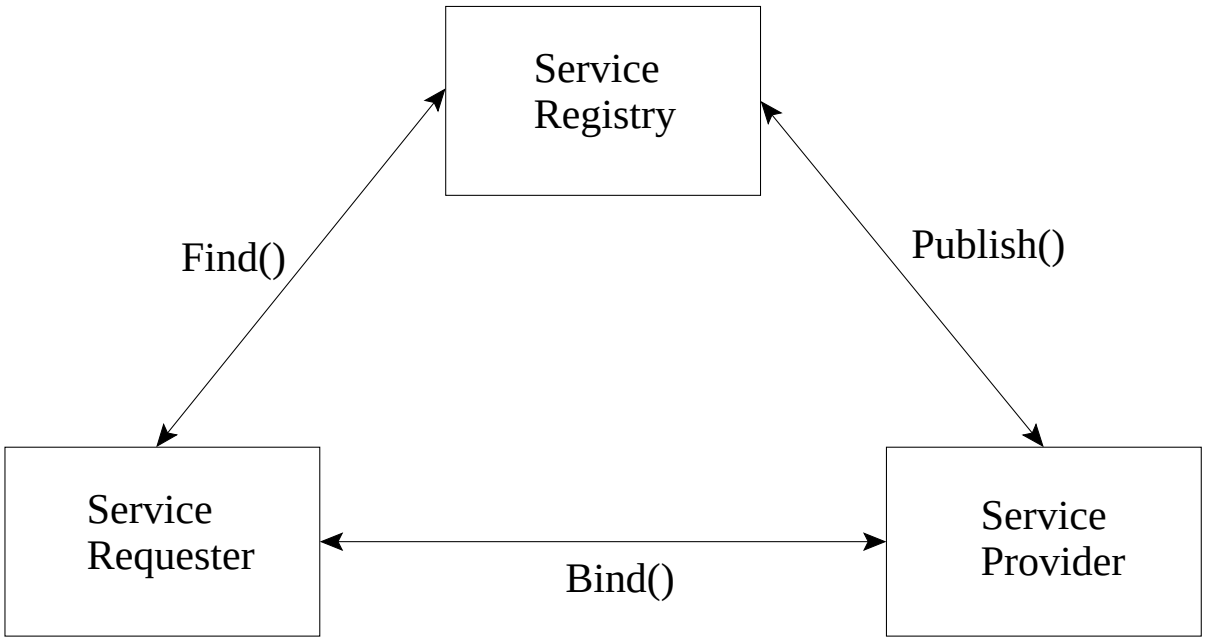
- Web Service têm uma interface que é pública.
- A interface também é definida em XML e especifica todos os métodos (e parâmetros) disponíveis para os programas cliente.
- A interface é especificada em WSDL.

UDDI (Universal Description, Discovery, and Integration)

- Ao fazer um Web Service, deve haver um mecanismo simples de publicá-lo algures de modo a que potenciais programas clientes saibam da sua existência.

- Isso é feito através de UDDI.

Service-oriented Architecture (SOA)



XML-RPC

- RPC (Remote Procedure Call)

- XML-RPC é um protocolo que utiliza XML para fazer chamadas de procedimentos remotos.

- Pedidos são codificados XML e enviados através de um HTTP POST.

- Respostas são enviadas no corpo de uma resposta HTTP.

- SOAP é parecido com XML-RPC, mas é mais complexo.

Exemplo de um pedido em XML-RPC

- chamada do método getWeather com um parâmetro (localidade)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodCall>
  <methodName>weather.getWeather</methodName>
  <params>
     <param><value>Faro</value></param>
  </params>
</methodCall>
```

Exemplo da resposta em XML-RPC

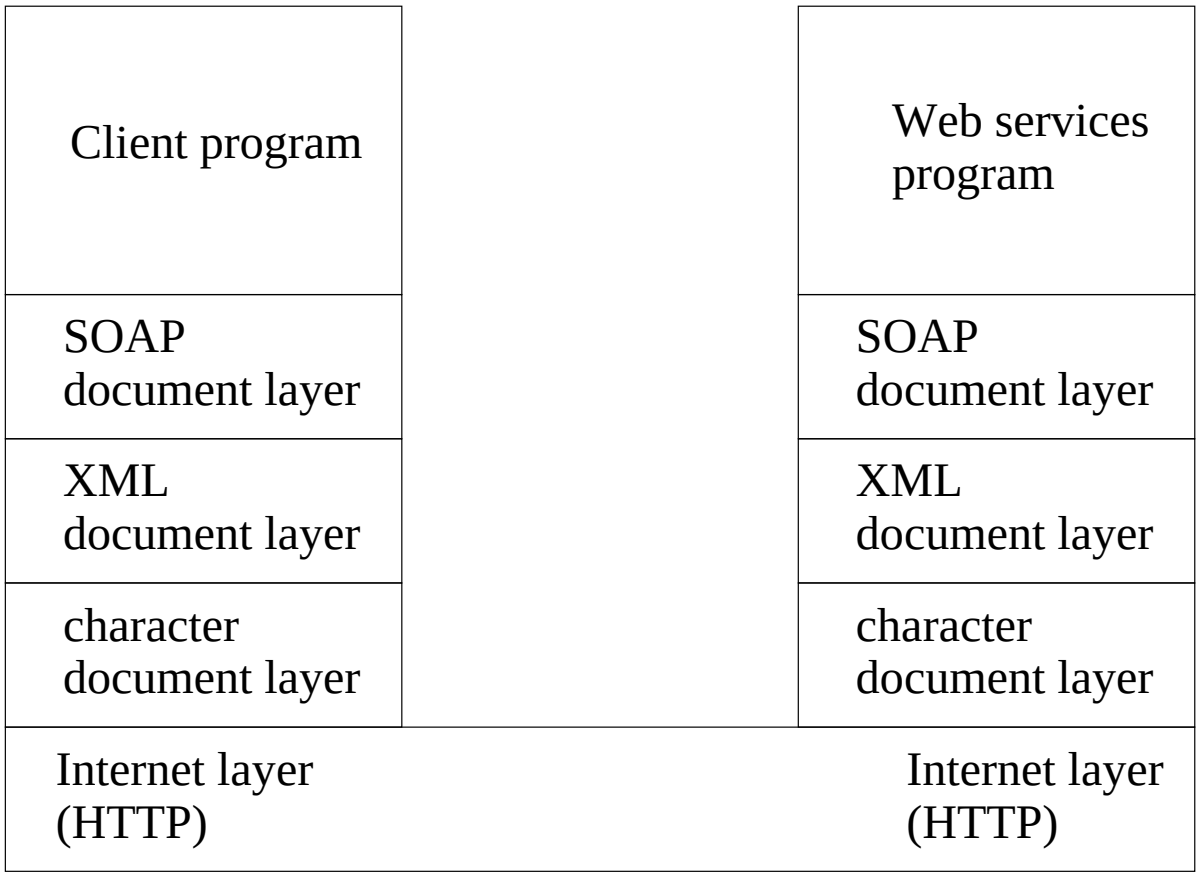
- A resposta consiste num elemento <methodResponse> que especifica o resultado do pedido (neste caso, a temperatura em Faro).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<methodResponse>
  <methodName>weather.getWeather</methodName>
  <params>
     <param>
        <value><int>24</int></value>
     </param>
  </params>
</methodResponse>
```

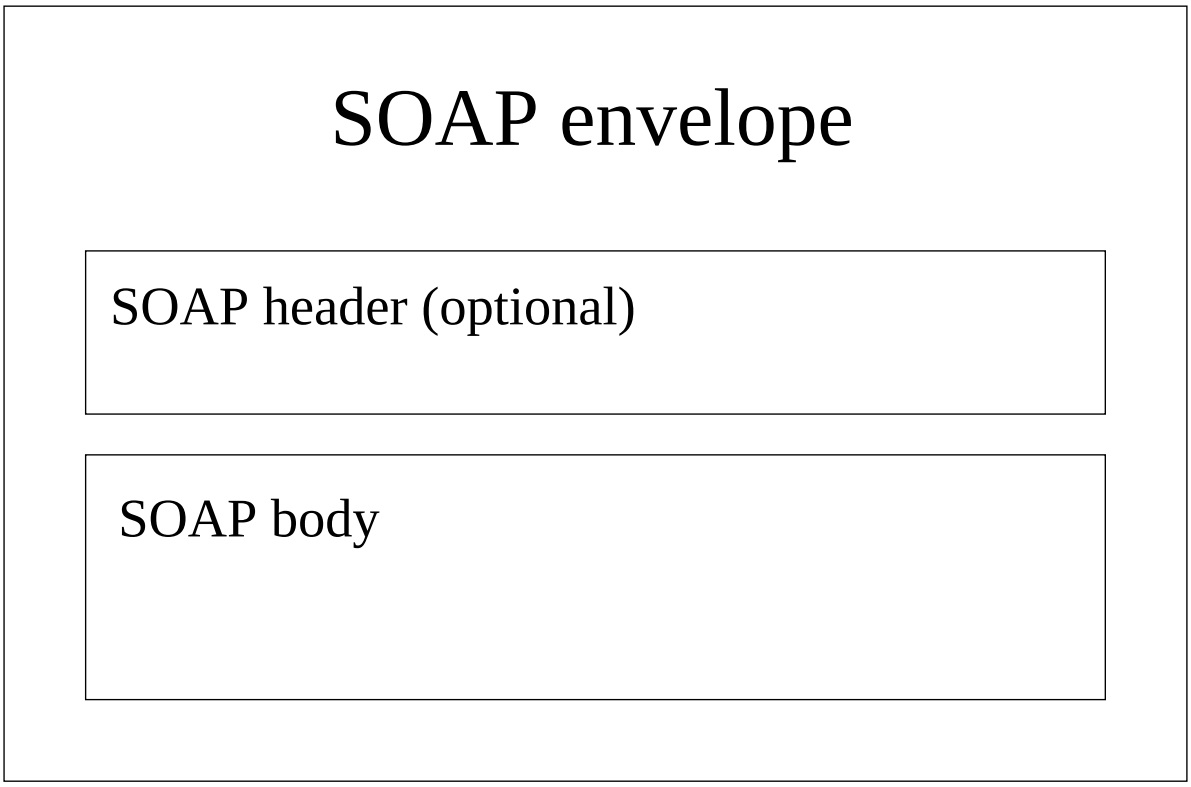
SOAP

- SOAP (Simple Object Access Protocol).
- SOAP é uma recomendação do W3C.
- SOAP é um protocolo de comunicação.
- Serve para fazer comunicação entre aplicações.
- É um formato para enviar mensagens baseado em XML
- É independente de SO e de linguagens de programação.

Soap Protocol Stack



Mensagem Soap



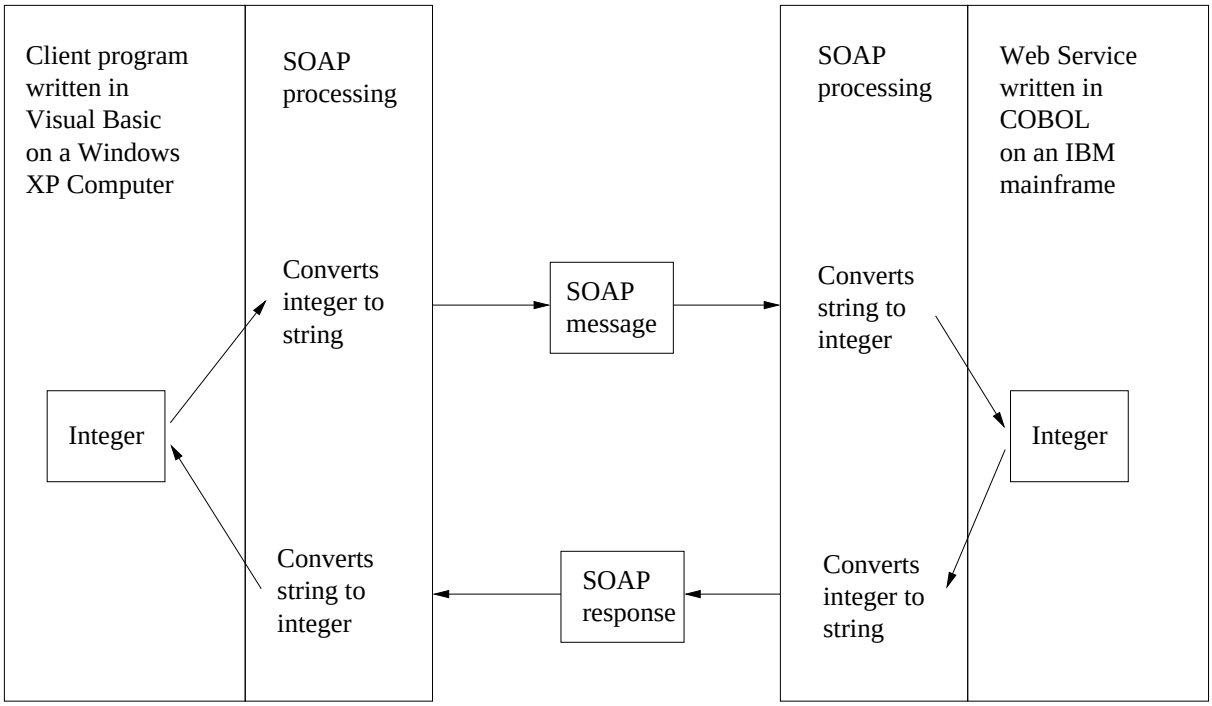
Exemplo de um pedido SOAP

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <SOAP-ENV:Body>
     <ns1:getWeather
        xmlns:ns1="urn:examples:weatherservice"
        SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/so
        <localidade xsi:type="xsd:string">Faro</localidade>
     </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Exemplo de uma resposta SOAP

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <SOAP-ENV:Body>
     <ns1:getWeatherResponse
        xmlns:ns1="urn:examples:weatherservice"
        SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/so
        <return xsi:type="xsd:int">24</return>
     </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Conversão de tipos de dados



WSDL

- WSDL é uma gramática XML que permite especificar uma interface pública para um Web Service.

- A interface inclui:
  - o nome dos métodos que o Web Service aceita.
  - os parâmetros que os métodos aceitam.
  - formato da respostas que devolve.
  - o protocolo de transporte utilizado (tipicamente é HTTP).
  - quais os URLs para o serviço.

Desvantagens de Web Services

- Para aplicações críticas, WS podem ser uma má opção porque (sites podem estar em baixo).

- A interface não deve mudar. Não há maneira de avisar os utilizadores do WS.

- tecnologia nova $\rightarrow$ muita coisa por descobrir/melhorar (ex: transacções).

Outras tecnologias para aplicações distribuídas

- CORBA: Common Object Request Broker Architecture
- RMI: Remote Method Invocation
- DCOM: Distributed Common Object Model

Outras tecnologias para aplicações distribuídas (cont.)

- CORBA: independente da linguagem de programação, mas bastante complexo.

- RMI: só para Java.

- DCOM: só funciona com aplicações Microsoft.

Web Services é o futuro?

- Será que os web services vão ser o futuro para as aplicações distribuídas?

- Muita gente diz que sim.

- Pelo menos há mais consenso do que com as tecnologias anteriores.

- O facto de ter o aval do W3C é muito positivo.

- Os grandes fabricantes mundiais (Microsoft, IBM, etc.) apoiam a iniciativa.