

# РАСПРЕДЕЛЕННЫЕ ОБЪЕКТНЫЕ ТЕХНОЛОГИИ

Веб-сервисы

# ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И СТАНДАРТЫ ПЕРВОГО ПОКОЛЕНИЯ

# XML WEB СЛУЖБЫ

- ◎ XML Web службы (Web-сервисы) – это программные компоненты, которые позволяют создавать независимые масштабируемые слабосвязанные приложения.
- ◎ В основе их функционирования лежит использование протоколов HTTP, XML, XSD, SOAP, WSDL.

# WSDL: WEB SERVICE DEFINITION LANGUAGE

4

- ◎ WSDL – это стандартный XML-документ, описывающий фундаментальные свойства Web службы, как то:
  - ◎ **Что это** – описание методов, предоставляемые Web службой;
  - ◎ **Как осуществляется доступ** – формат данных и протоколы;
  - ◎ **Где он расположен** – специальный сетевой адрес службы.

# SOAP (УЖЕ НЕ ТОЛЬКО SIMPLE OBJECT ACCESS PROTOCOL)

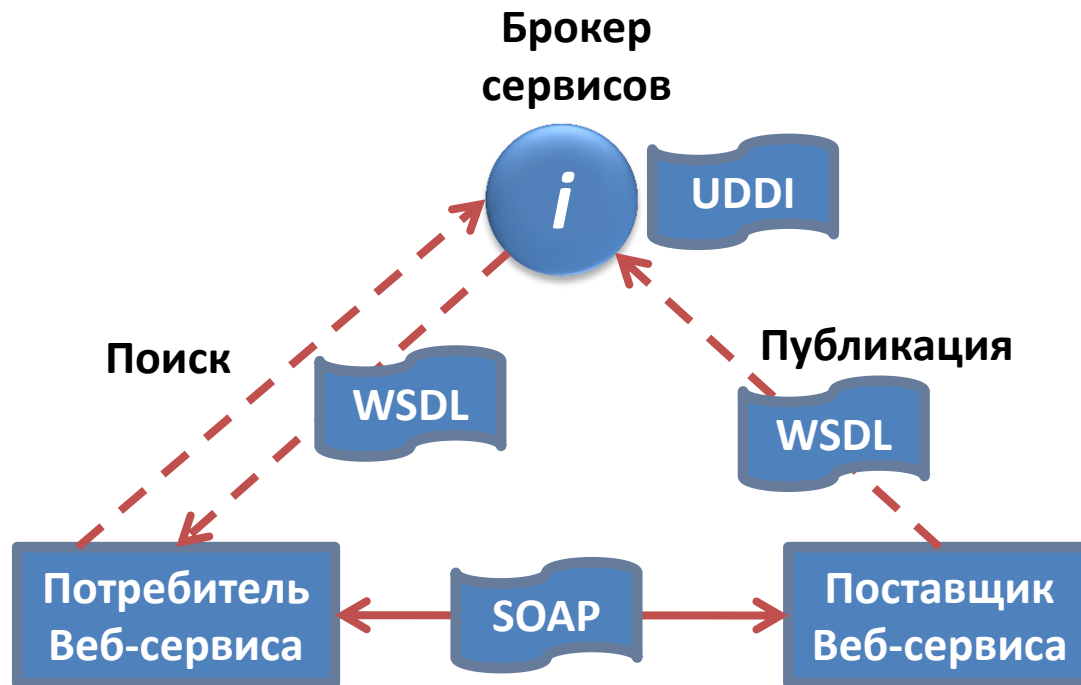
- ◎ SOAP – это протокол, основанный на обмене XML-документами.
- ◎ SOAP определяется следующим образом: «SOAP это основанный на XML протокол обмена информацией в децентрализованной распределенной среде.

# СТРУКТУРА SOAP

- ◎ SOAP состоит из 3-х частей:
  - ◎ обертка, определяющая среду для описания содержимого сообщения и как его обрабатывать;
  - ◎ набор правил кодирования для выражения сущностей типов данных, определенных в приложении;
  - ◎ конвенция осуществления удаленного запуска процедур и получения ответов.

- ◎ Предоставляет механизмы для:
  - ◎ определения коммуникационной единицы – сообщения SOAP;
  - ◎ обработки ошибок;
  - ◎ репрезентации данных;
  - ◎ удаленного вызова процедур;
  - ◎ связи с HTTP.

# «ТРЕУГОЛЬНИК ВЕБ-СЕРВИСОВ»

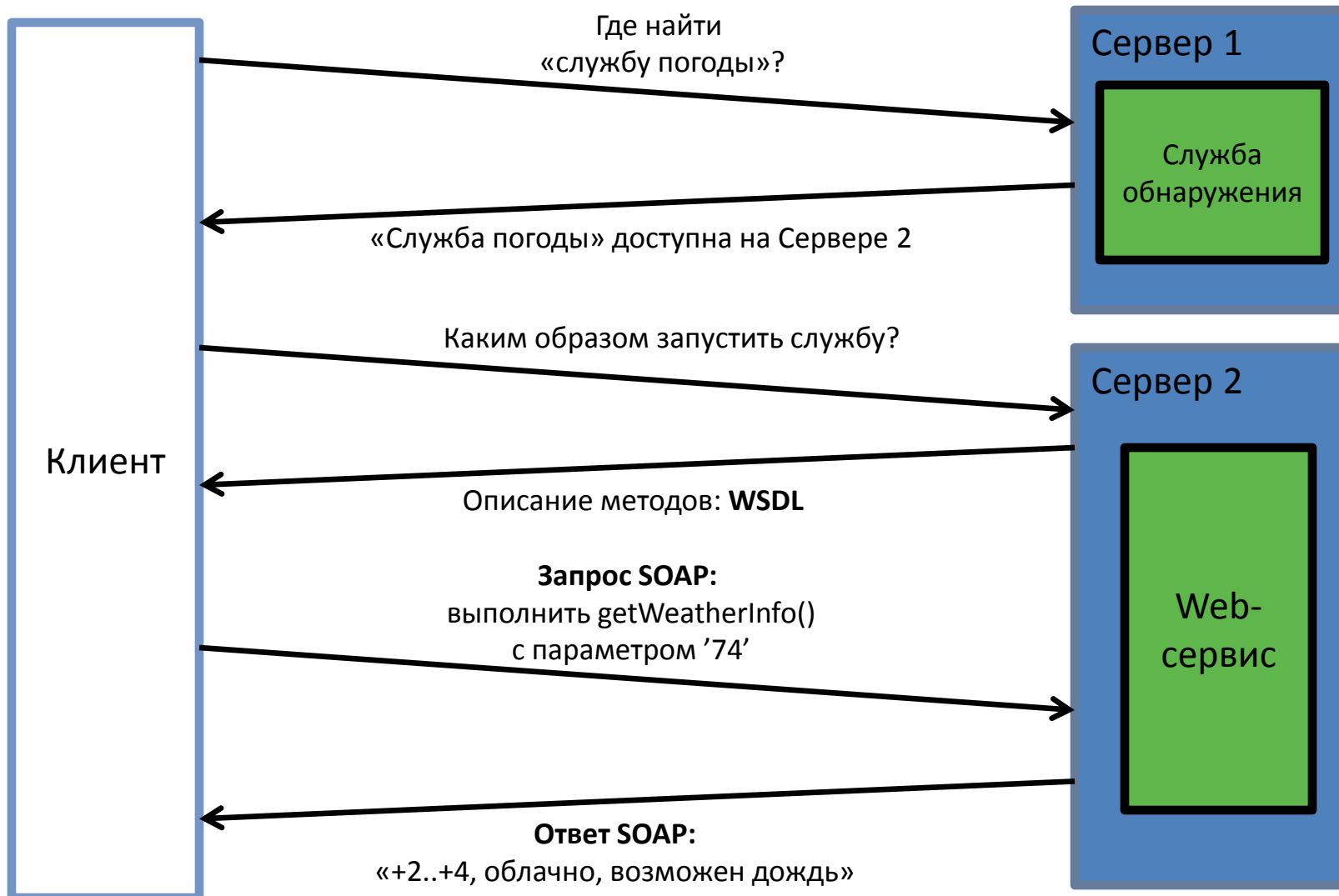




# СТЕК ПРОТОКОЛОВ WEB СЛУЖБ



# Типичный пример вызова Web службы



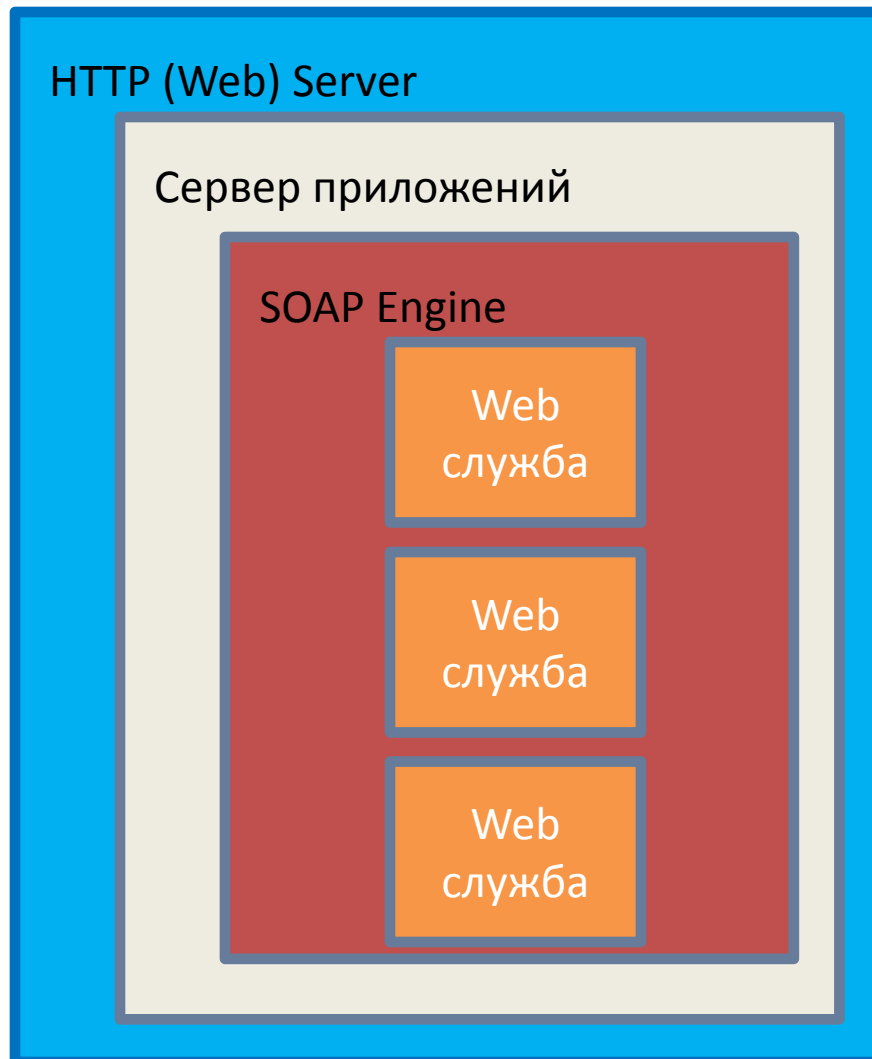
# АДРЕСАЦИЯ WEB СЛУЖБ

- ⦿ Адрес Web службы – это стандартный URI (Uniform Resource Identifier).

`http://live.capescience.com/ccx/GlobalWeather`

- ⦿ BTW: URL (Uniform Resource Locator) является одним из видов URI, и является прародителем URI.

# СЕРВЕРНАЯ ЧАСТЬ WEB СЛУЖБ



# ПРОГРАММНЫЕ СРЕДЫ ДЛЯ РАЗРАБОТКИ WEB-СЕРВИСОВ

- ⊙ Apache Axis, на базе, например, Apache Tomcat
- ⊙ IBM Websphere
- ⊙ Microsoft .NET
- ⊙ J2EE (Java 2 Enterprise Edition) server container

# ПРИМЕРЫ WEB СЛУЖБ

Назначение Web-службы	Адрес
Web-службы для добавления в базу данных заявок на перевозку груза и предложений транспорта. В виде Web-служб также реализован доступ к транспортным новостям сайта, тематика которых охватывает все виды перевозок и сопутствующие области — таможенную, страховую и т.д.	<a href="http://www.parevozki.ru/">http://www.parevozki.ru/</a>
<p>Web-службы для получения оперативной информации об актуальном расписании, прилете/вылете самолетов, состоянии рейсов. Сервис «Справка о рейсах/Табло аэропортов» предоставляет следующий набор функций:</p> <ul style="list-style-type: none"> <li>- AirportList — список аэропортов;</li> <li>- AirportInfo — информация об аэропорте по его коду;</li> <li>- DateList — список дат, по которым имеется информация;</li> <li>- Arrival/Departure — информация о прибытии/отправлении рейсов;</li> <li>- FlightSearch — поиск рейса по номеру;</li> <li>- FlightInfo — подробная информация о текущем состоянии рейса (по ключу, полученному в результате вызова Arrival/Departure/FlightSearch).</li> </ul> <p>Сервис «Расписание рейсов» предоставляет следующий набор функций:</p> <ul style="list-style-type: none"> <li>- AirportAlphabetList — список букв, на которые начинаются названия аэропортов;</li> <li>- AirportList — список аэропортов на указанную букву;</li> <li>- AirportInfo — информация об аэропорте;</li> <li>- Search — поиск рейсов;</li> <li>- Calendar — информация о других датах, в которые летает указанный рейс</li> </ul>	<a href="http://webservices.aeroflot.ru/">http://webservices.aeroflot.ru/</a>
Использование Web-служб для извлечения информации из поисковой системы Google позволяет разработчикам выполнять запросы более чем к 8 млрд. Web-страниц. (Справедливости ради отметим, что Yandex предоставляет похожую услугу, но базирующуюся на XML-запросах, посылаемых методом POST или методом GET (протокол HTTP) без использования SOAP-запроса.)	<a href="http://www.google.com/apis/">http://www.google.com/apis/</a>
<p>Создание набора Web-служб — Amazon Web Services (AWS) — превратило онлайн-магазин Amazon в полноценную платформу для электронной коммерции. В настоящее время в рамках Amazon Web Services доступны следующие службы:</p> <ul style="list-style-type: none"> <li>- Amazon E-Commerce Service (ECS);</li> <li>- Amazon Simple Queue Service;</li> <li>- Alexa Web InformationService</li> </ul>	<a href="http://www.amazon.com/">http://www.amazon.com/</a>

# СТАНДАРТ WSDL

# ПРИМЕР WEB СЛУЖБЫ

- ◎ Простой пример Web службы (Java)

```
public class MyMath
{
    public int squared(int x)
    {
        return x * x;
    }
}
```



# ЭЛЕМЕНТЫ WSDL- ДОКУМЕНТА

17

- ⊙ *Блок types* – типы данных, используемые Web-службой
- ⊙ *Блок message* – сообщения, используемые Web-службой
- ⊙ *Блок portType* – методы, предоставляемые Web-службой
- ⊙ *Блок binding* – протоколы связи, используемые Web-службой

```
18 <?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://DefaultNamespace"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://DefaultNamespace"
  xmlns:intf="http://DefaultNamespace"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="squaredRequest">
    <wsdl:part name="in0" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="squaredResponse">
    <wsdl:part name="squaredReturn" type="xsd:int"/>
  </wsdl:message>
  <wsdl:portType name="MyMath">
    <wsdl:operation name="squared" parameterOrder="in0">
      <wsdl:input message="impl:squaredRequest" name="squaredRequest"/>
      <wsdl:output message="impl:squaredResponse" name="squaredResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="MyMathSoapBinding" type="impl:MyMath">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="squared">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="squaredRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://DefaultNamespace" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="squaredResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://DefaultNamespace" use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="MyMathService">
    <wsdl:port binding="impl:MyMathSoapBinding" name="MyMath">
      <wsdlsoap:address location="http://localhost:8080/axis/testaccount/MyMath"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Namespaces

messages - сообщения

portTypes - методы

binding - связи

Определение службы

# ПОРТЫ WSDL <PORTTYPE>

- ⊙ Элемент <portType> является наиболее важным элементом WSDL.
- ⊙ Он определяет саму Web-службу, предоставляемые ей операции и используемые сообщения.
- ⊙ Можно сравнить с библиотекой функций, в которой указаны входные параметры и результаты работы функции.

# ПРИМЕР БЛОКА <PORTTYPE>

```
<wsdl:portType name="MyMath">
  <wsdl:operation name="squared"
parameterOrder="in0">
    <wsdl:input message="impl:squaredRequest"
      name="squaredRequest" />
    <wsdl:output message="impl:squaredResponse"
      name="squaredResponse" />
  </wsdl:operation>
</wsdl:portType>
```

# СООБЩЕНИЯ WSDL

## <MESSAGE>

- ⊙ Элемент <message> определяет элементы данных операции.
- ⊙ Каждое сообщение может содержать одну или несколько частей. Эти части можно сравнить с параметрами вызываемых функций в традиционных языках программирования.

# ПРИМЕР БЛОКА <MESSAGE>

```
<wsdl:message name="squaredRequest">  
  <wsdl:part name="in0" type="xsd:int"/>  
</wsdl:message>  
  
<wsdl:message name="squaredResponse">  
  <wsdl:part name="squaredReturn"  
type="xsd:int"/>  
</wsdl:message>
```

# СВЯЗИ WSDL <BINDING>

- ⊙ Элемент <binding> определяет формат сообщения и детали протокола для каждого порта.
- ⊙ Отвечает за то, каким образом элементы абстрактного интерфейса в блоке <portType> преобразуются в массивы информации в формате протоколов взаимодействия, например SOAP.

# ПРИМЕР БЛОКА <BINDING>

```
<wsdl:binding name="MyMathSoapBinding" type="impl:MyMath">
  <wsdlsoap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="squared">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="squaredRequest">
      <wsdlsoap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="squaredResponse">
      <wsdlsoap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://DefaultNamespace" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```



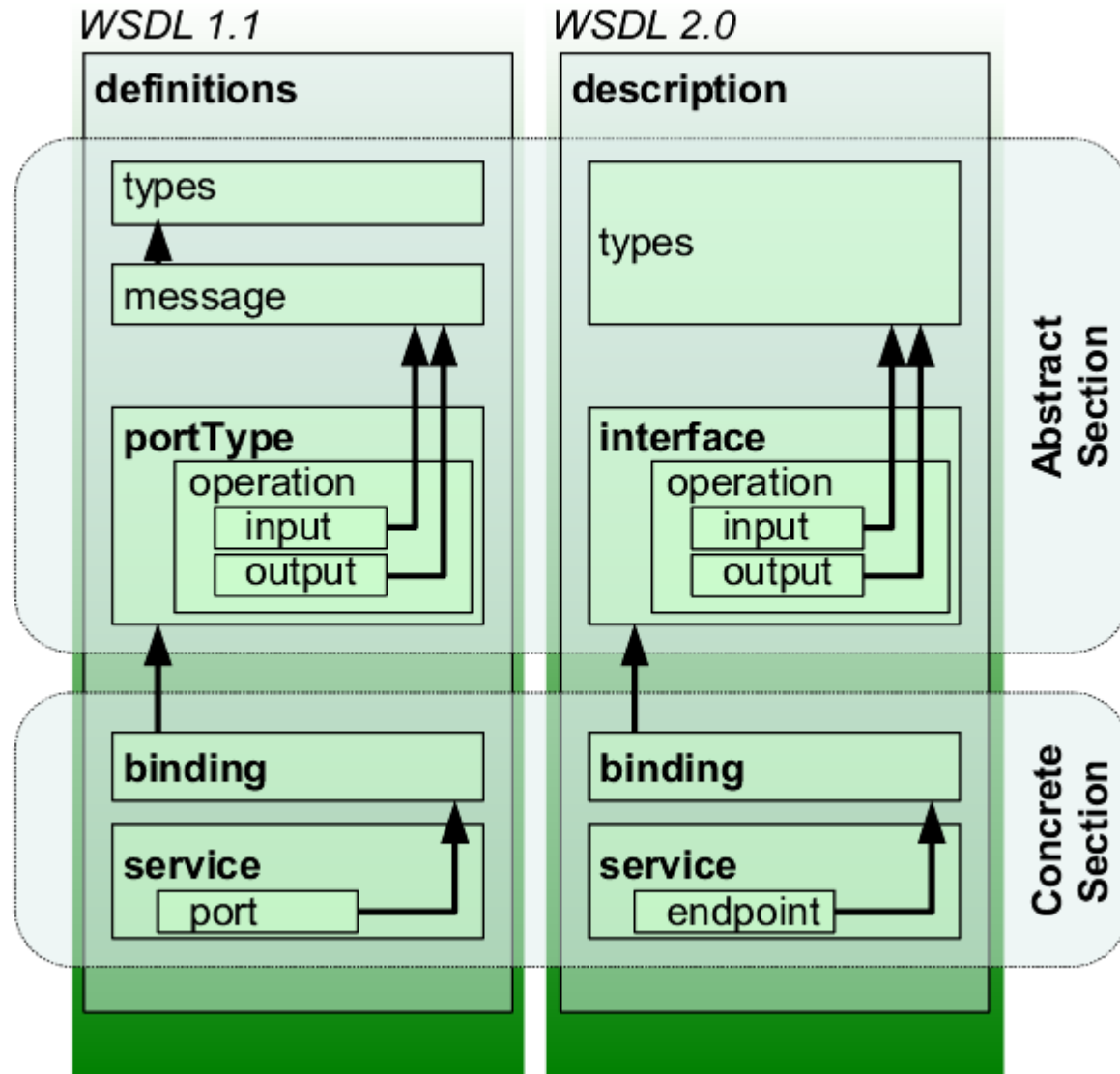
# БЛОКИ <PORT> И <SERVICE>

- ⊙ Данные блоки определяют, где находится служба.
- ⊙ `port` – описывает расположение и способ доступа к конечной точке
- ⊙ `service` – именованная коллекция портов

# ПРИМЕР БЛОКА <SERVICE>

```
<wsdl:service name="MyMathService">  
  <wsdl:port binding="impl:MyMathSoapBinding"  
    name="MyMath">  
    <wsdlsoap:address  
      location="http://localhost:8080/axis/myaccount/MyM  
ath"/>  
    </wsdl:port>  
  </wsdl:service>  
</wsdl:definitions>
```

# WSDL 1.1 VS WSDL 2.0

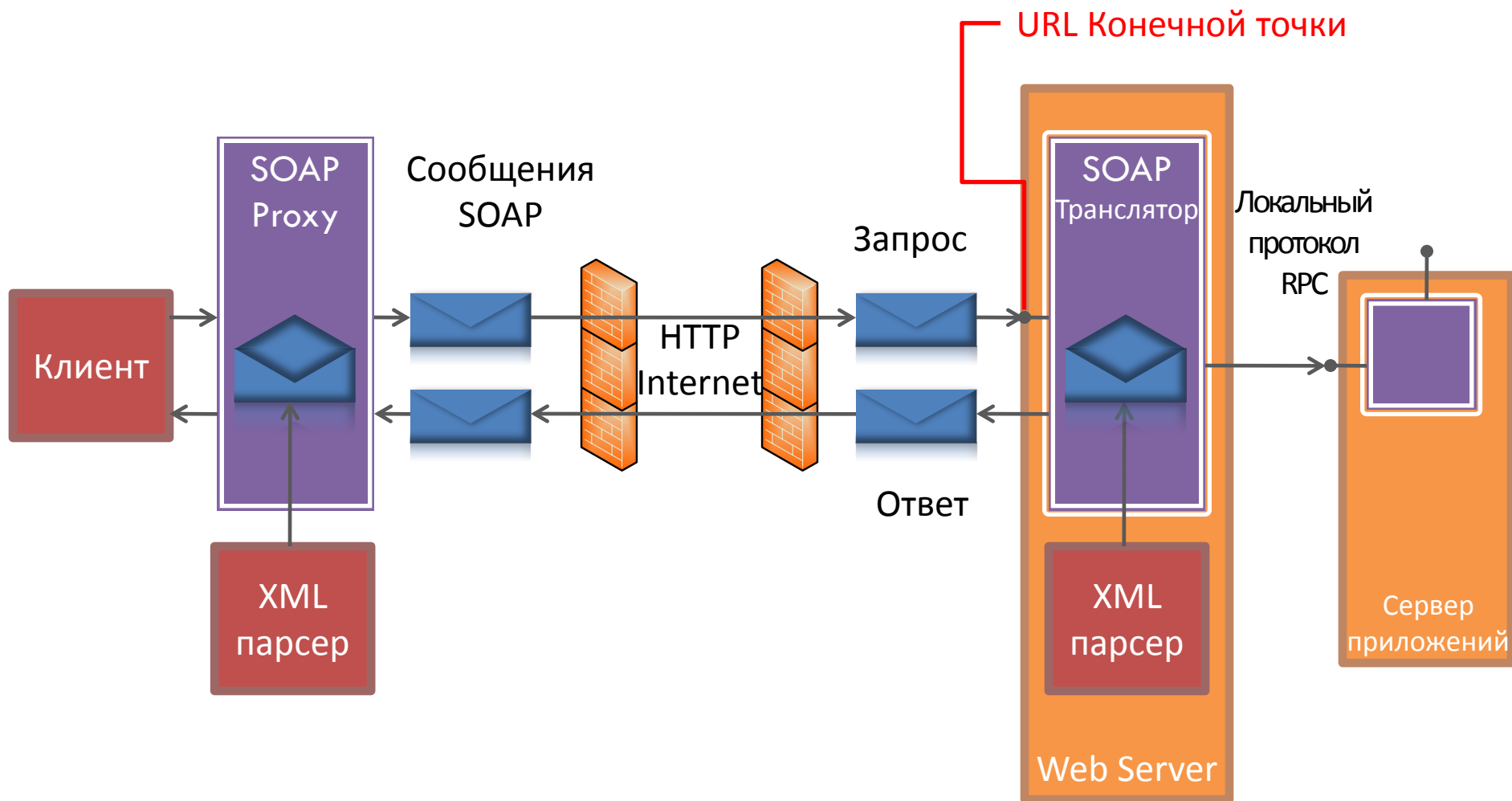


# СТАНДАРТ SOAP

# Сообщение SOAP

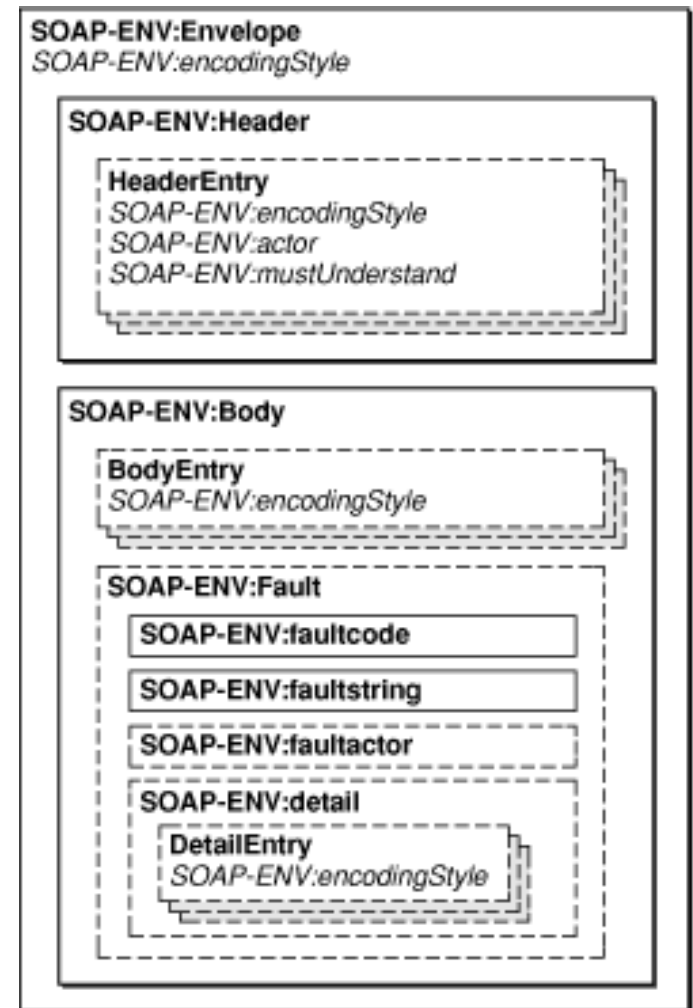
- ◎ Сообщением SOAP называют одностороннюю передачу информации между узлами SOAP: от источника к приемнику.
- ◎ Сообщения SOAP являются фундаментальным строительным блоком, обеспечивающим возможности более сложных шаблонов взаимодействия:
  - Запрос/ответ
  - «диалоговый» режим
  - и т.п.

# АРХИТЕКТУРА SOAP



# ЭЛЕМЕНТЫ СООБЩЕНИЯ SOAP

- ⊙ Envelope (конверт) – корневой элемент сообщения.
- ⊙ Header (заголовок) – не обязательный элемент сообщения. Может содержать дополнительную информацию для приложения, обрабатывающего запрос.
- ⊙ Body (Тело) – обязательный элемент сообщения. Содержит вызовы необходимых методов и передаваемые параметры.



# ШАБЛОН СООБЩЕНИЯ SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap=http://www.w3.org/2001/12/soap-envelope
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
  ...
  ...
</soap:Header>
<soap:Body>
  ...
  ...
  <soap:Fault>
    ...
    ...
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```



# ПРИМЕР ЗАГОЛОВКА SOAP

- ⦿ В заголовке мы можем ввести новый элемент, не предусмотренный стандартом SOAP. Например, номер транзакции.
- ⦿ Атрибуты:
  - ⦿ `mustUnderstand` – получатель обязан обрабатывать этот элемент;
  - ⦿ `actor` – указывает конкретное приложение-получатель при обработке сообщения в цепочке.

```
<soap:Header>
```

```
<trans:Transaction  
  xmlns:trans="http:  
  //www.host.com/namespaces/space/"  
  soap:mustUnderstand="1">
```

```
12
```

```
</trans:Transaction>
```

```
</soap:Header>
```

# ТЕЛО СООБЩЕНИЯ SOAP

## Запрос

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productID>12345</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

## Ответ

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productID>12345</productID>
        <productName>Стакан граненый</productName>
        <description>Стакан граненый. 200 мл.</description>
        <price>9.95</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

# СВЯЗЫВАНИЕ SOAP И HTTP

- ⊙ Передача SOAP сообщений происходит поверх HTTP – протокола посредством запроса POST (начиная со стандарта SOAP 1.2 возможно применение GET)
- ⊙ Стандартный протокол связи:
  - ⊙ Клиент посылает запрос
  - ⊙ Сервер отвечает ОК

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type:
application/soap+xml;
charset=utf-8
Content-Length: nnn
...
```

```
HTTP/1.1 200 OK
Content-Type:
application/soap; charset=utf-
8
Content-Length: nnn
...
```