Распределенные вычислительные системы

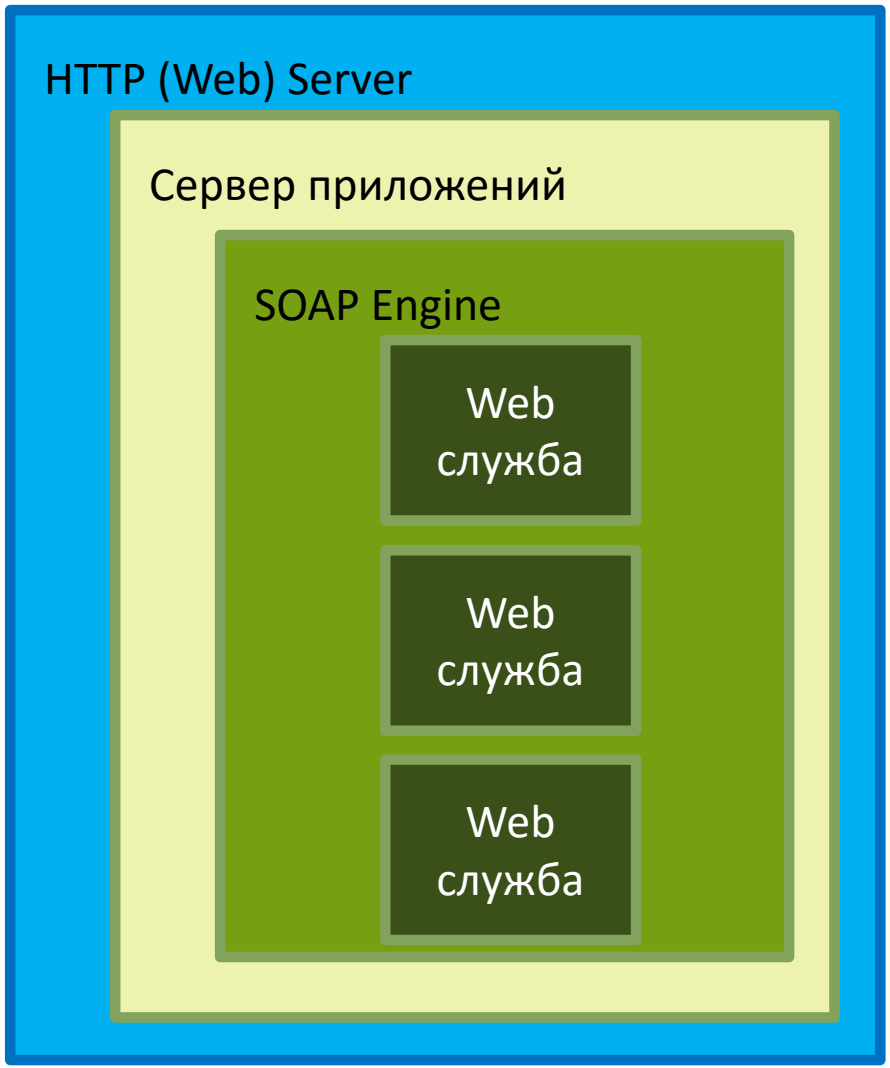
Сервис-ориентированная архитектура. XML Веб-сервисы.

XML (SOAP) Веб-сервисы

XML Веб-сервисы

- XML Веб-сервисы – это основанная на языке XML платформо-независимая технология, поддерживающая разработку распределенных приложений.
- XML Веб-сервисы обеспечивает создание независимых, масштабируемых, слабосвязанных приложений.
- Они основаны на протоколах HTTP, XML, XSD, SOAP, WSDL.
- Реализации SOAP/ WSDL:
  - The Java API for XML Web Services (JAX-WS)
  - Apache CXF
  - Microsoft’s Windows Communication Foundation (WCF)
  - …

Серверная часть Web служб

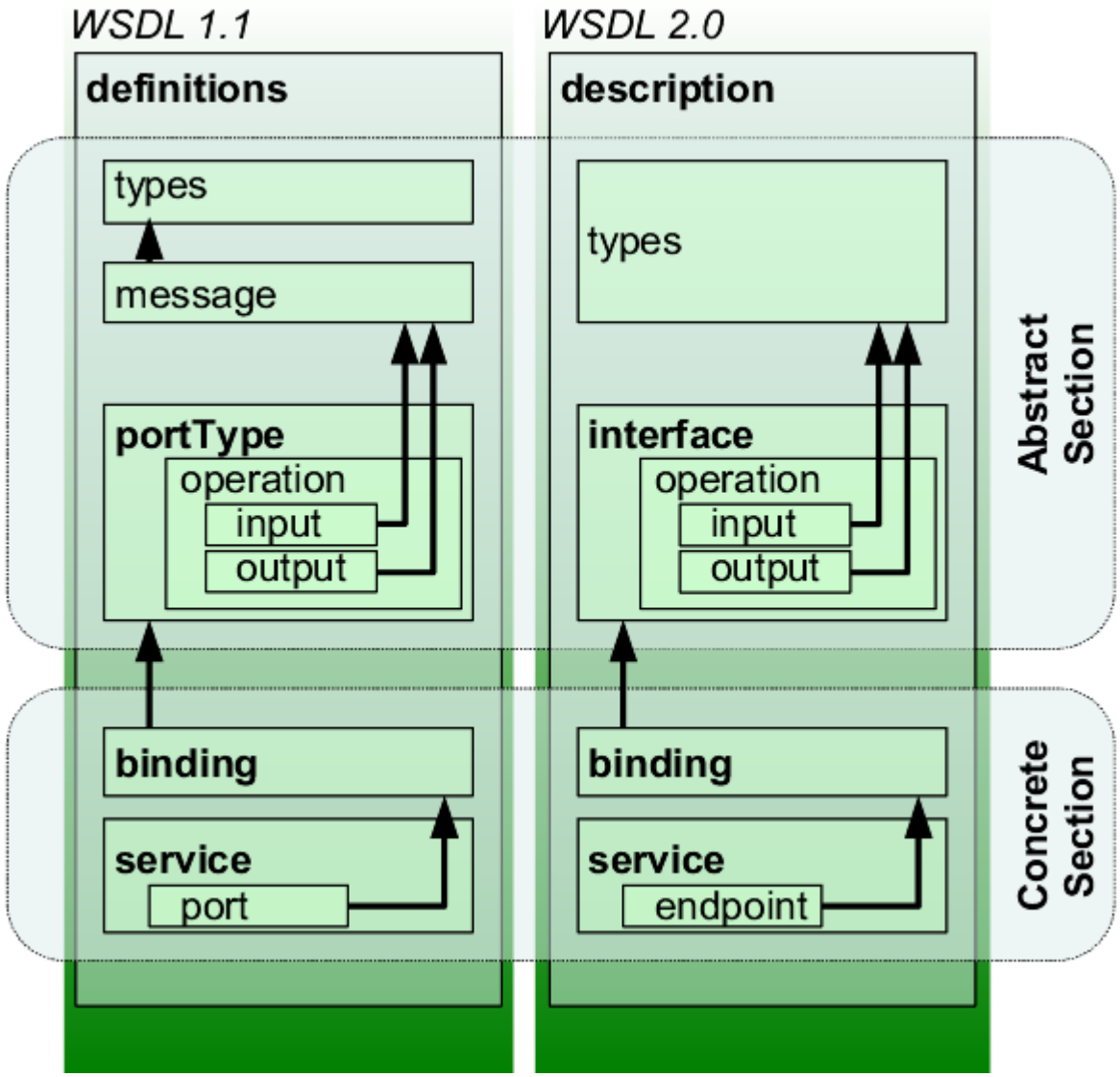


Borja Sotomayor . The Globus Toolkit 4 Programmer's Tutorial. University of Chicago Department of Computer Science

WSDL: Web Service Definition Language

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

WSDL 1.1 VS WSDL 2.0



Пример Web службы

- Простой пример Web службы (Java)

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

WSDL-документ

```
<?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

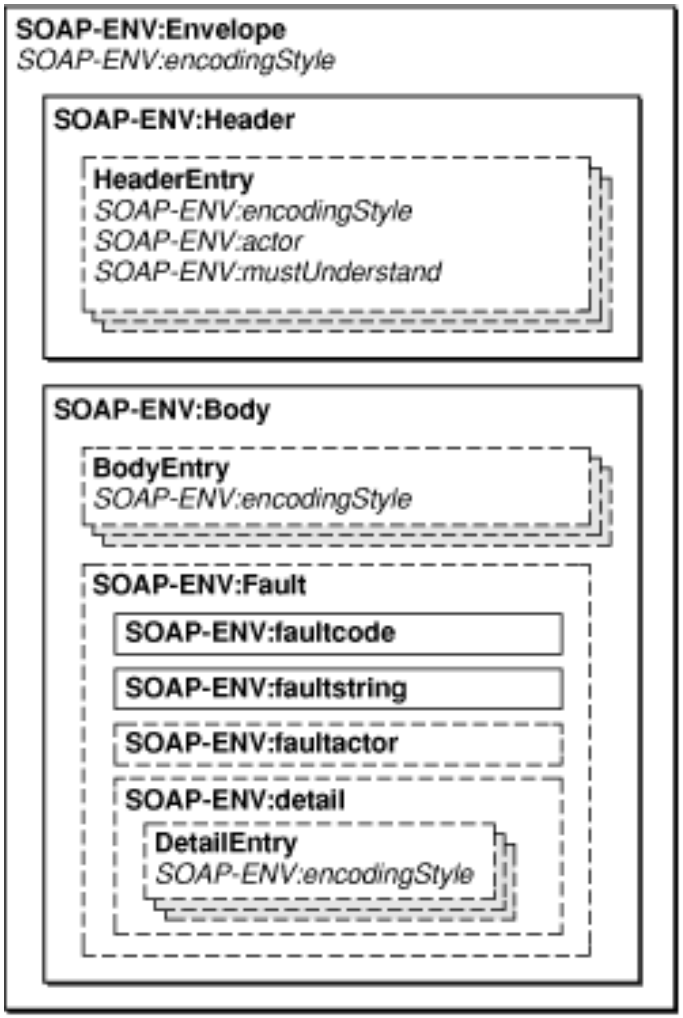
Определение службы

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.

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>
```

W3C schools website (http://www.w3schools.com)

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 OK

```
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
…
```

Советы по проектированию RPC сервисов

RPC: «Плоский API»

- «Плоский API»: создание интерфейсов, которые выглядят как методы классов

```
@WebMethod(operationName = "ReserveRentalCar")
public RentalOptions ReserveRentalCar (
  @WebParam( name = "RentalCity") String RentalCity,
  @WebParam( name = "PickupMonth") int PickupMonth,
  @WebParam( name = "PickupDay") int PickupDay,
  @WebParam( name = "PickupYear") int PickupYear,
  @WebParam( name = "ReturnMonth") int ReturnMonth,
  @WebParam( name = "ReturnDay") int ReturnDay,
  @WebParam( name = "ReturnYear") int ReturnYear,
  @WebParam (name = "RentalType") String RentalType )
{ // implementation would appear here }
```

- Такой API очень не гибкий, и очень хрупкий, т.к. в нем невозможно даже изменить последовательность параметров без необходимости замены кода во всех клиенах.
- Возможное решение: использование одного аргумента-сообщения.

Daigneau, Robert (2011-10-25). Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. Pearson Education. Kindle Edition.

RPC: Посредники

- Не смотря на то, что можно формировать RPC-сообщения вручную, на клиентской стороне чаще всего используются посредники (proxy), которые инкапсулируют процесс сетевого взаимодействия. В этом случае, с клиентской точки зрения, процесс вызова удаленного метода не отличается от вызова локального метода.
- Для генерации посредников используются специальные программные генераторы кода, которые на основе описания *клиентского интерфейса* (например, WSDL-документа) создают код вызова удаленной процедуры.
- При изменении интерфейса сервиса, все клиенты должны перегенерировать свои proxy-методы

Daigneau, Robert (2011-10-25). Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. Pearson Education. Kindle Edition.

RPC: Адресация

- Для обеспечения прозрачной адресации, необходимо чтобы клиенты не имели информации о конкретном расположении сервиса.
- В этом случае его можно перенести или реплицировать (при необходимости)
- Для реализации такой возможности необходимо использовать промежуточный «Соединитель сервисов» (Service Connector), который обеспечивает управление соединениями клиентов с сервисом, перенаправляя запросы от клиентов на реальный адрес используемого сервиса.

Daigneau, Robert (2011-10-25). Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. Pearson Education. Kindle Edition.