

Pertemuan 7

# **Web Services**

# Pengantar Web Service

- Web service menyediakan antarmuka layanan yang memungkinkan klien untuk berinteraksi dengan server dengan cara yang lebih umum daripada web browser lakukan.
- Klien mengakses operasi-operasi dalam antarmuka web service dengan cara meminta dan balasan diformat dalam **XML (eXtensible Markup Language)** dan biasanya ditransmisikan melalui HTTP.

# Pengantar Web Service

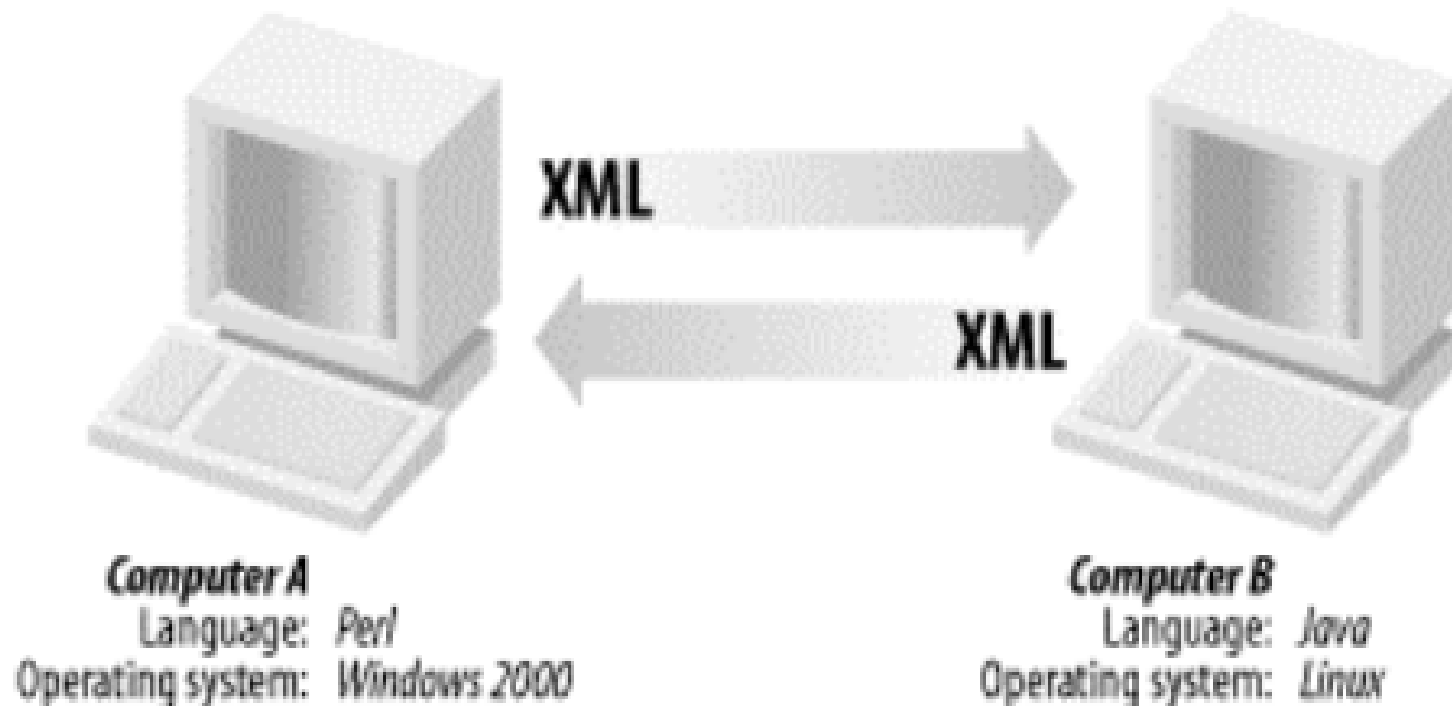
- Web services semakin penting dalam sistem terdistribusi:
  - Mendukung interoperabilitas di Internet global
  - Merupakan kunci penting dari integrasi bisnis-ke-bisnis
  - Memunculkan budaya 'mashup' yang memungkinkan pengembang pihak ketiga untuk mengembangkan perangkat lunak yang inovatif kreatif di atas layanan dasar yang sudah ada.
  - Menyediakan middleware yang mendasari Grid computing dan Cloud computing

# Web server & Web service

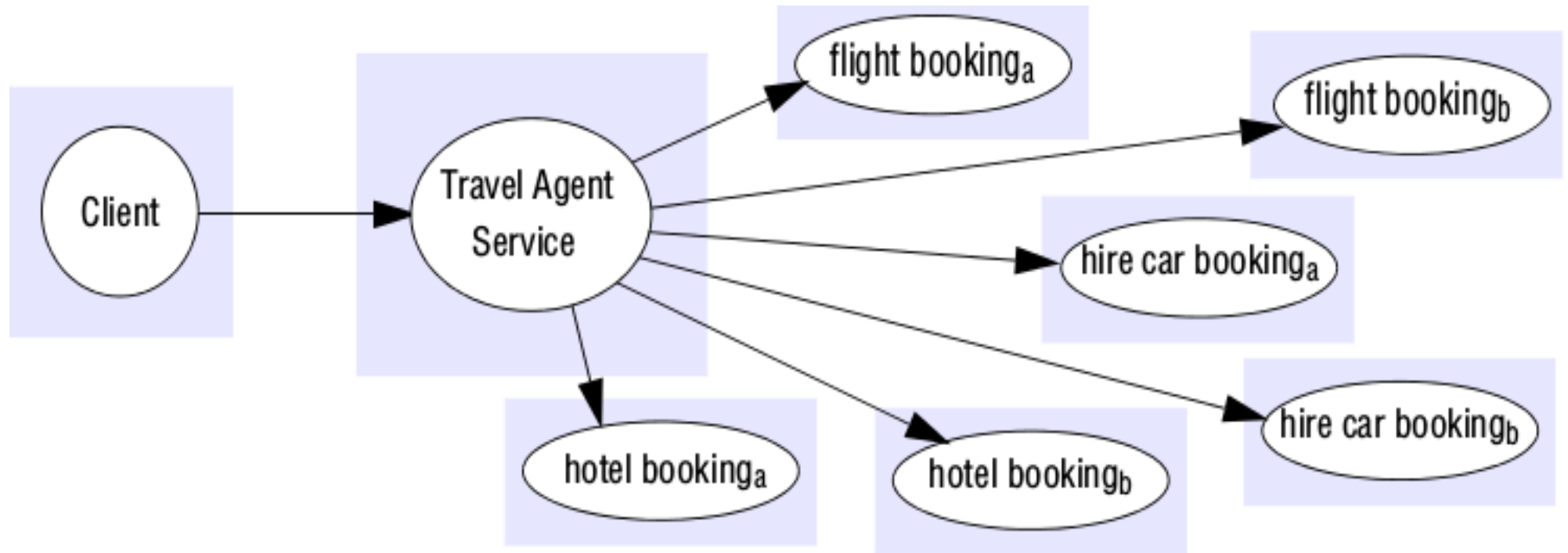
- Web service merupakan perpanjangan dari Web dan dapat disediakan oleh web server. Namun sebuah server web service tidak harus berupa sebuah web server yang umum
- 'Web server' dan 'Web service' ! .
  - Web server menyediakan layanan HTTP dasar, sedangkan web service menyediakan layanan berbasis pada operasi-operasi yang didefinisikan dalam interfacenya
- Operasi-operasi dalam web service dapat diberikan oleh berbagai sumber yang berbeda, misalnya, program-program, object atau database. Sebuah layanan web dapat dikelola oleh web server bersama dengan halaman web; atau mungkin layanan yang benar-benar terpisah

# Web service

- Sebuah web service adalah suatu layanan yang tersedia melalui jaringan / Internet, menggunakan sistem pesan XML standar, dan tidak terikat pada satu sistem operasi atau bahasa pemrograman.



# Contoh – Travel Agents Service



# XML

- XML singkatan dari EXtensible Markup Language.
- XML dirancang untuk mendeskripsikan data, bukan menampilkan data
- Contoh pesan dengan format XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<note>
```

```
  <to>Budi</to>
```

```
  <from>Joko</from>
```

```
  <heading>Reminder</heading>
```

```
  <body>Jangan lupa meeting jumat!</body>
```

```
</note>
```

# XML Basics

- Start Tag

- End Tag

Text Content

- Attribute

- Self Closing Tag

```
<person>
  <name>Joko</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```



# Protokol akses Web service

- Protokol untuk mengakses web service :
  - SOAP (Simple Object Access Protocol)
  - XML-RPC
  - REST (Representational State Transfer)
- Format pesan atau representasi data pada web service :
  - XML → SOAP, XML-RPC
  - JSON (JavaScript Object Notation)

# SOAP

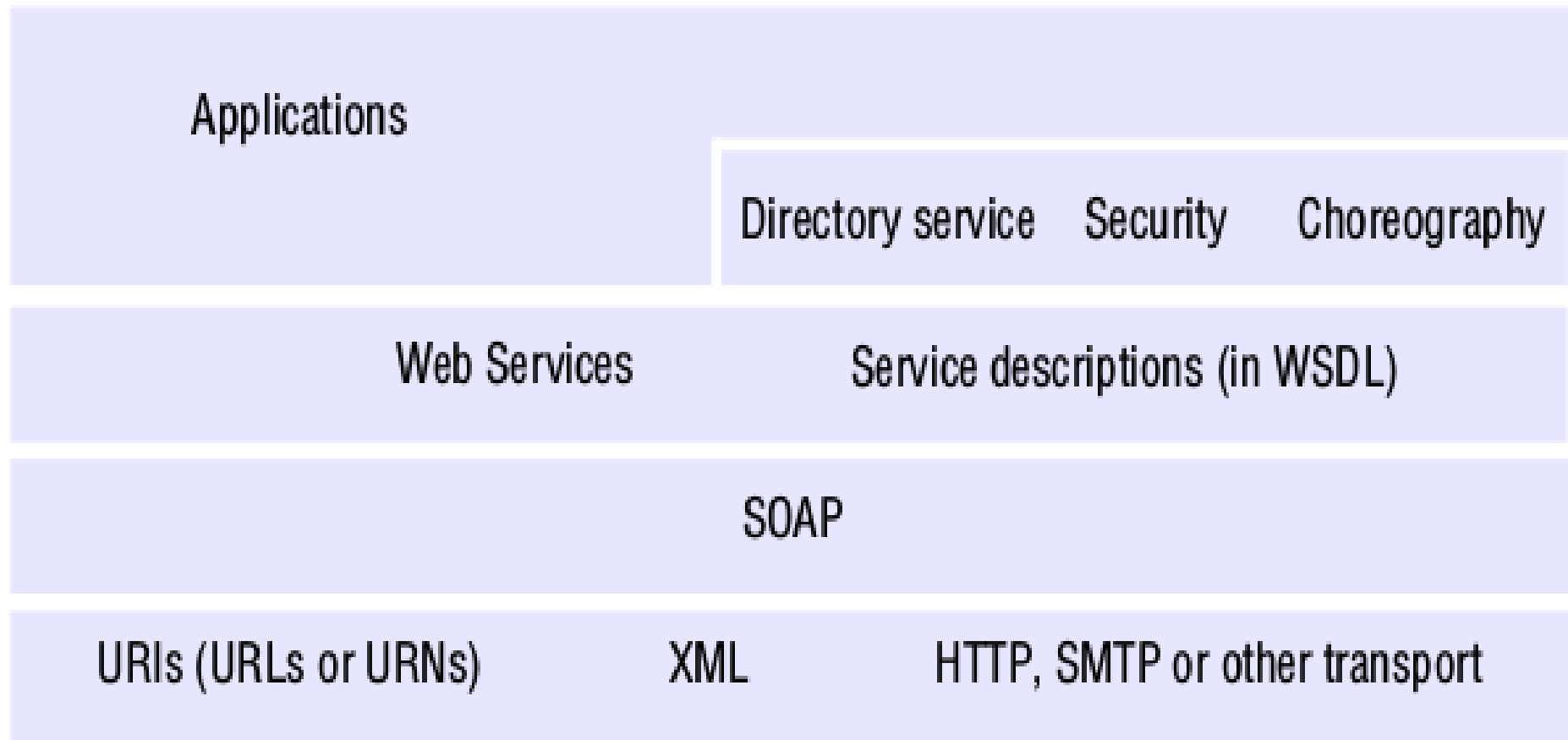
- SOAP adalah protokol berbasis XML untuk bertukar informasi antar komputer. Meskipun SOAP dapat digunakan dalam berbagai sistem pesan dan dapat disampaikan melalui berbagai protokol transport, fokus awal SOAP adalah pemanggilan remote prosedur yang dibawa melalui HTTP

# SOAP

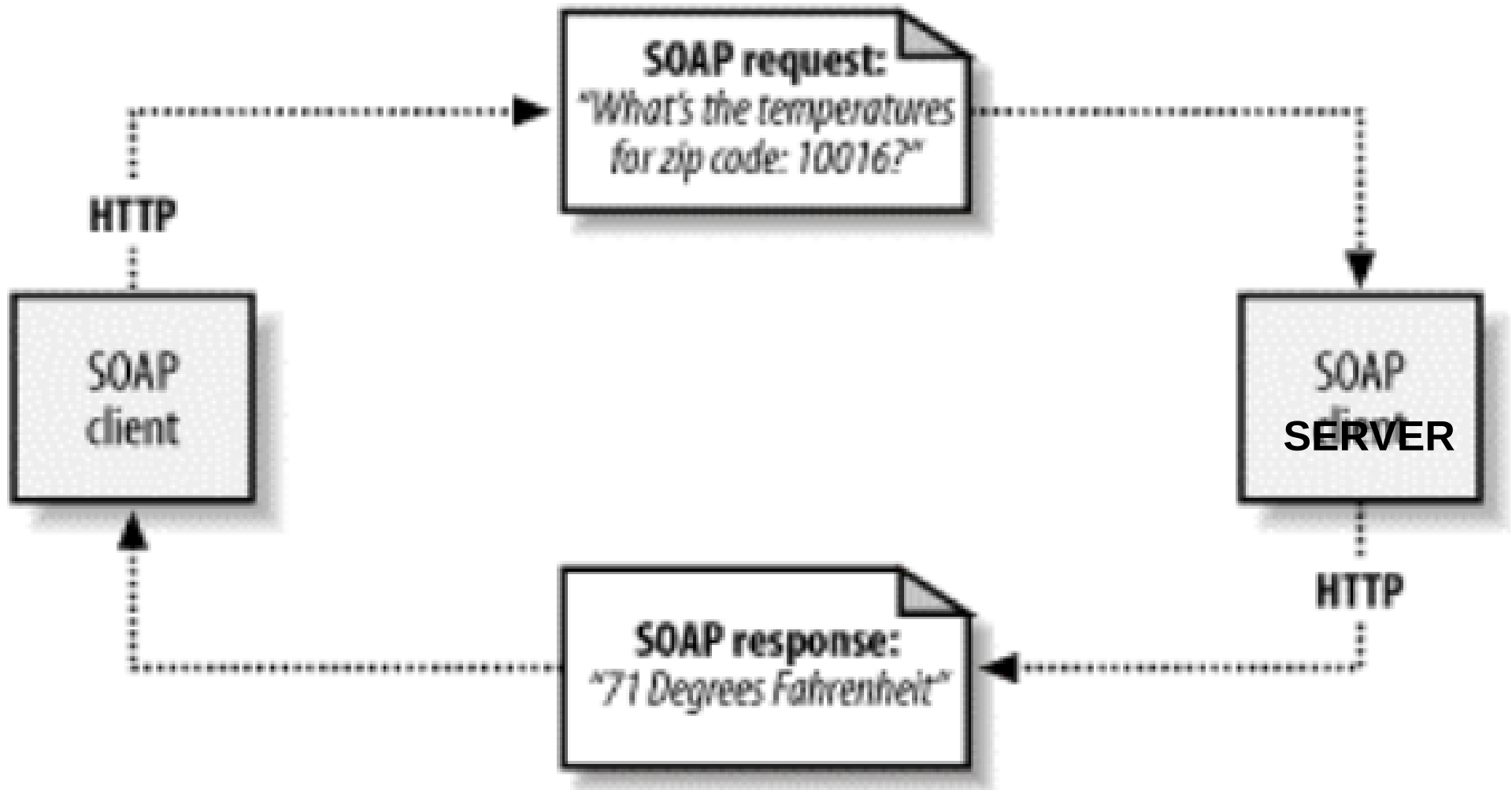
- Metode lainnya, termasuk CORBA dan RMI, menyediakan fungsionalitas mirip dengan SOAP, tetapi pesan SOAP yang ditulis seluruhnya dalam XML dan karenanya independen terhadap platform dan bahasa pemrograman.

Sebagai contoh, klien Java SOAP berjalan pada Linux atau klien Perl berjalan pada Solaris dapat terhubung ke server Microsoft SOAP yang berjalan pada Windows 2000.

# Web service infratructure & Components



# SOAP in Action



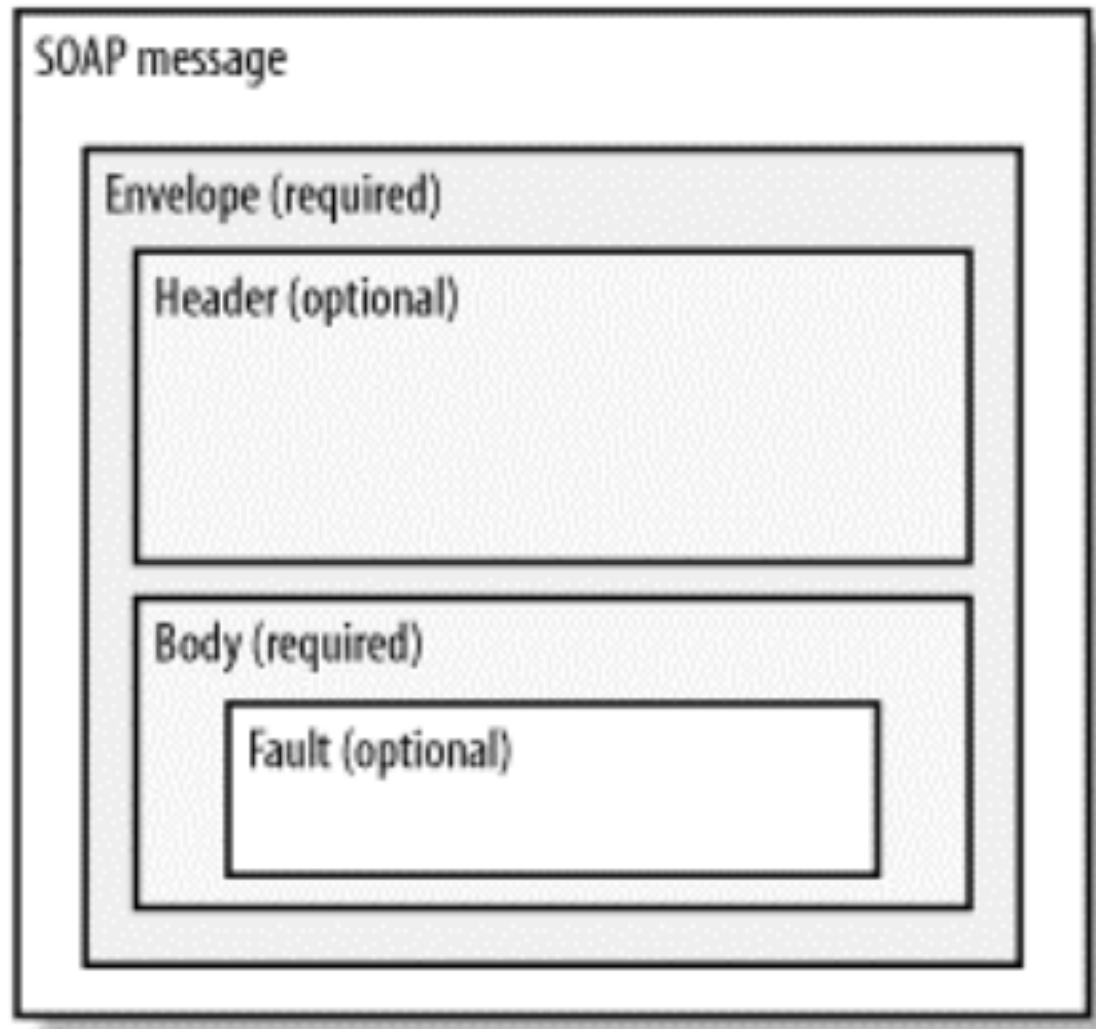
# SOAP Request

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getTemp
      xmlns:ns1="urn:xmethods-Temperature"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getTemp>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# SOAP Response

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getTempResponse
      xmlns:ns1="urn:xmethods-Temperature"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:float">71.0</return>
    </ns1:getTempResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# SOAP Message





# Envelope

- Setiap pesan SOAP memiliki elemen root ***Envelope***. Berbeda dengan spesifikasi lainnya, seperti HTTP dan XML, SOAP tidak mendefinisikan model versi tradisional yang didasarkan pada nomor rilis mayor dan minor (misalnya, HTTP 1.0 vs HTTP 1.1). Sebaliknya, SOAP menggunakan XML namespaces untuk membedakan versi
- Versi harus dirujuk dalam elemen ***Envelope***. Sebagai contoh:

***<SOAP-ENV:Envelope***

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

The SOAP 1.1 namespace URI is <http://schemas.xmlsoap.org/soap/envelope/>, whereas the

SOAP 1.2 namespace URI is <http://www.w3.org/2001/09/soap-envelope>

# Header

- Elemen header opsional menawarkan kerangka kerja yang fleksibel untuk menentukan persyaratan tambahan tingkat aplikasi .
- Misalnya, elemen header dapat digunakan untuk menentukan tanda tangan digital untuk layanan yang dilindungi sandi; juga, dapat digunakan untuk menentukan nomor rekening untuk layanan SOAP yang dibayar per-digunakan.
- Banyak layanan SOAP saat ini tidak memanfaatkan unsur Header, tetapi pada layanan-layanan SOAP yang sudah matang, header dapat digunakan untuk menyediakan mekanisme terbuka untuk otentikasi, manajemen transaksi, dan otorisasi pembayaran

# Contoh Header

- Elemen Header yang menentukan rekening pembayaran, yang harus dipahami dan diproses oleh Server SOAP. Berikut ini adalah contoh Header:

```
<SOAP-ENV:Header>  
  <ns1:PaymentAccount xmlns:ns1="urn:ecerami" SOAP-ENV: mustUnderstand="true">  
    orsenigo473  
  </ns1:PaymentAccount >  
</SOAP-ENV:Header>
```

# Body - Fault

- Elemen Body adalah elemen wajib bagi semua pesan SOAP
- Dalam hal terjadi kesalahan, elemen Body akan mencakup elemen Fault.

```
<SOAP-ENV:Body>  
  <SOAP-ENV:Fault>  
    <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>  
    <faultstring xsi:type="xsd:string">  
      Failed to locate method (ValidateCreditCard) in class  
      (examplesCreditCard) at /usr/local/ActivePerl-5.6/lib/  
      site_perl/5.6.0/SOAP/Lite.pm line 1555.  
    </faultstring>  
  </SOAP-ENV:Fault>  
</SOAP-ENV:Body>
```

# SOAP Fault Subelements

Element name	Description
<i>faultCode</i>	A text code used to indicate a class of errors. See Table 3-2 for a listing of predefined fault codes.
<i>faultString</i>	A human-readable explanation of the error.
<i>faultActor</i>	A text string indicating who caused the fault. This is useful if the SOAP message travels through several nodes in the SOAP message path, and the client needs to know which node caused the error. A node that does not act as the ultimate destination must include a <i>faultActor</i> element.
<i>detail</i>	An element used to carry application-specific error messages. The <i>detail</i> element can contain child elements, called detail entries.

# SOAP Fault Codes

Name	Description
SOAP-ENV:VersionMismatch	Indicates that the SOAP <i>Envelope</i> element included an invalid namespace, signifying a version mismatch.
SOAP-ENV:MustUnderstand	Indicates that the recipient is unable to properly process a <i>Header</i> element with a <i>mustUnderstand</i> attribute set to <i>true</i> . This ensures that <i>mustUnderstand</i> elements are not silently ignored.
SOAP-ENV:Client	Indicates that the client request contained an error. For example, the client has specified a nonexistent method name, or has supplied the incorrect parameters to the method.
SOAP-ENV:Server	Indicates that the server is unable to process the client request. For example, a service providing product data may be unable to connect to the database.

# SOAP Encoding

- SOAP meliputi sekumpulan built-in aturan pengkodean tipe data. Hal ini memungkinkan pesan SOAP untuk menunjukkan tipe data tertentu, seperti integer, float, double, atau array.
- encoding style untuk sebuah pesan SOAP message diatur melalui atribut ***SOAP-ENV:encodingStyle***
- To SOAP 1.1 encoding, use the value ***<http://schemas.xmlsoap.org/soap/encoding/>*** .
- To use SOAP 1.2 encoding, use the value ***<http://www.w3.org/2001/09/soap-encoding>*** .

# SOAP Encoding

- Contoh respon SOAP dengan tipe data double:

```
<?xml version='1.0' encoding='UTF-8'?>
<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:getPriceResponse
      xmlns:ns1="urn:examples:priceservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding">
      <return xsi:type="xsd:double">54.99</return>
    </ns1:getPriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# SOAP via HTTP

- SOAP tidak terikat hanya untuk satu protokol transportasi saja. Bahkan, SOAP dapat dibawa melalui protokol SMTP, FTP, MQSeries IBM, atau Microsoft Message Queuing (MSMQ). Namun, Spesifikasi SOAP hanya merinci tentang protokol HTTP saja, dan HTTP tetap yang populer sebagai Protokol transport SOAP.

# SOAP via HTTP

- Permintaan SOAP dikirim melalui permintaan HTTP dan tanggapan SOAP dikembalikan dalam isi respon HTTP. Sementara permintaan SOAP dapat dikirim melalui HTTP GET, spesifikasi termasuk rincian tentang HTTP POST. (HTTP POST lebih disukai karena sebagian besar server menempatkan batas karakter pada permintaan GET.)
- Pada permintaan dan tanggapan HTTP perlu untuk menetapkan header content-type yaitu text/xml

# SOAP via HTTP

- klien harus menentukan header **SOAPAction**.
- Header SOAPAction adalah sebuah spesifikasi server URI digunakan untuk menunjukkan maksud dari permintaan.

# SOAPAction Header

```
POST /perl/soaplite.cgi HTTP/1.0
Host: services.xmethods.com
Content-Type: text/xml; charset=utf-8
Content-Length: 538
SOAPAction: "urn:xmethodsBabelFish#BabelFish"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:BabelFish
      xmlns:ns1="urn:xmethodsBabelFish"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <translationmode xsi:type="xsd:string">en_fr</translationmode>
      <sourcedata xsi:type="xsd:string">Hello, world!</sourcedata>
    </ns1:BabelFish>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# SOAP Response

*HTTP/1.1 200 OK*

*Date: Sat, 09 Jun 2001 15:01:55 GMT*

*Server: Apache/1.3.14 (Unix) tomcat/1.0 PHP/4.0.1pl2*

*SOAPServer: SOAP::Lite/Perl/0.50*

*Cache-Control: s-maxage=60, proxy-revalidate*

*Content-Length: 539*

*Content-Type: text/xml*

*<?xml version="1.0" encoding="UTF-8"?>*

*<SOAP-ENV:Envelope*

*xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"*

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

*xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"*

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

*xmlns:xsd="http://www.w3.org/1999/XMLSchema">*

*<SOAP-ENV:Body>*

*<namespace1:BabelFishResponse xmlns:namespace1="urn:xmethodsBabelFish">*

*<return xsi:type="xsd:string">Bonjour, monde!</return>*

# REST

- REST singkatan dari State Transfer Representasi, yang merupakan gaya arsitektur untuk aplikasi jaringan hypermedia, ini terutama digunakan untuk membangun Web service yang ringan, maintainable, dan scalable.
- Sebuah layanan berbasis REST disebut RESTful service.
- REST tidak tergantung pada protokol, tapi hampir setiap RESTful service menggunakan HTTP sebagai protokol yang mendasarinya.

# JSON

- Representasi data REST menggunakan format JSON
- JSON: JavaScript Object Notation
- JSON adalah sebuah sintak untuk menyimpan atau bertukar data
- JSON adalah alternatif yang mudah untuk digunakan dari pada XML.

# JSON example

```
{"employees": [  
  {"firstName": "John", "lastName": "Doe"},  
  {"firstName": "Anna", "lastName": "Smith"},  
  {"firstName": "Peter", "lastName": "Jones"}  
]}
```