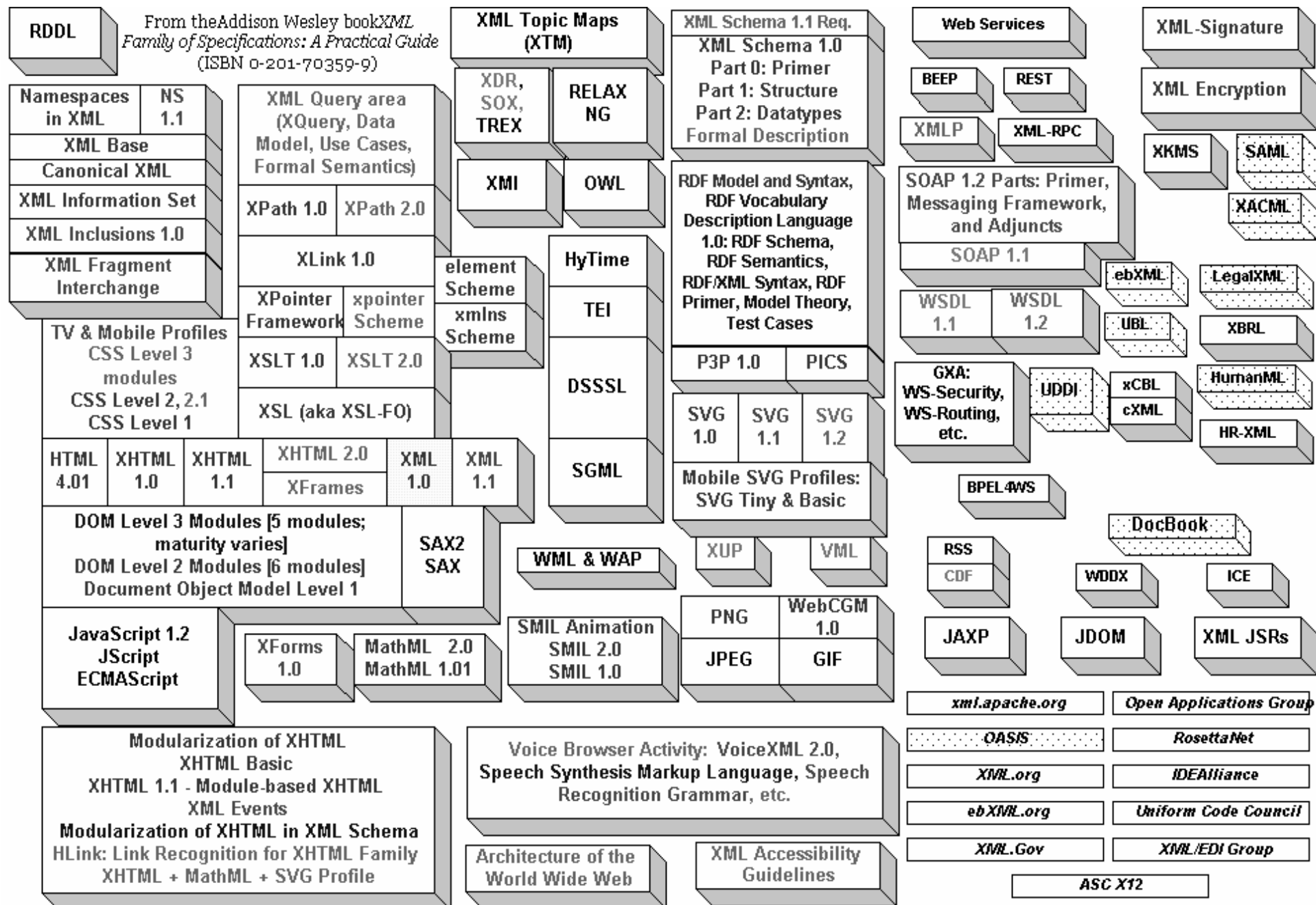


# Web Services

Mats Persson, FOI  
**matpe@foi.se**

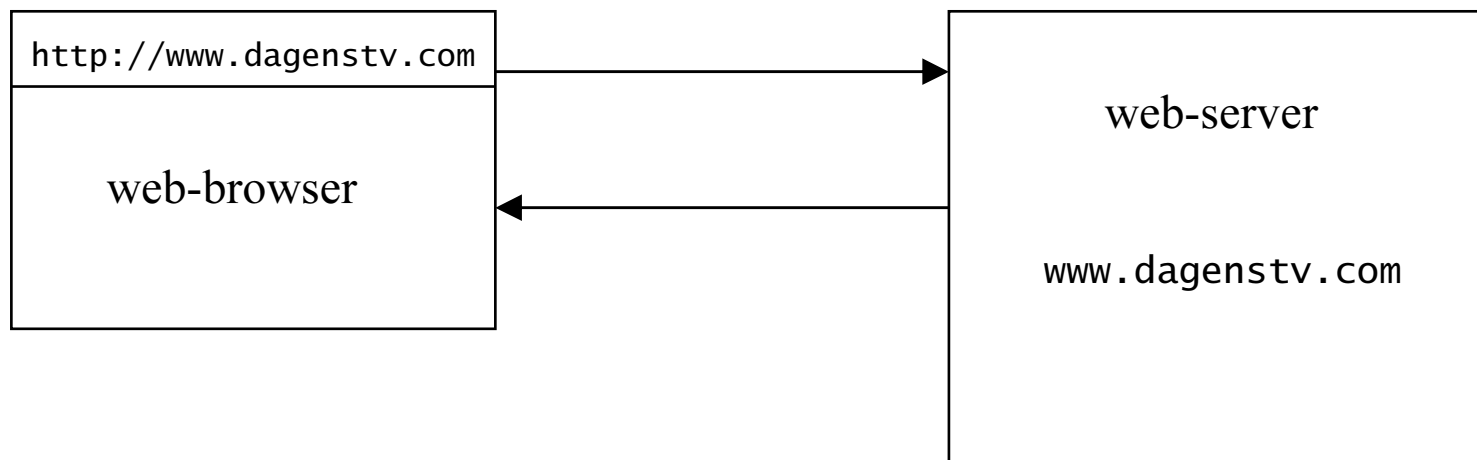


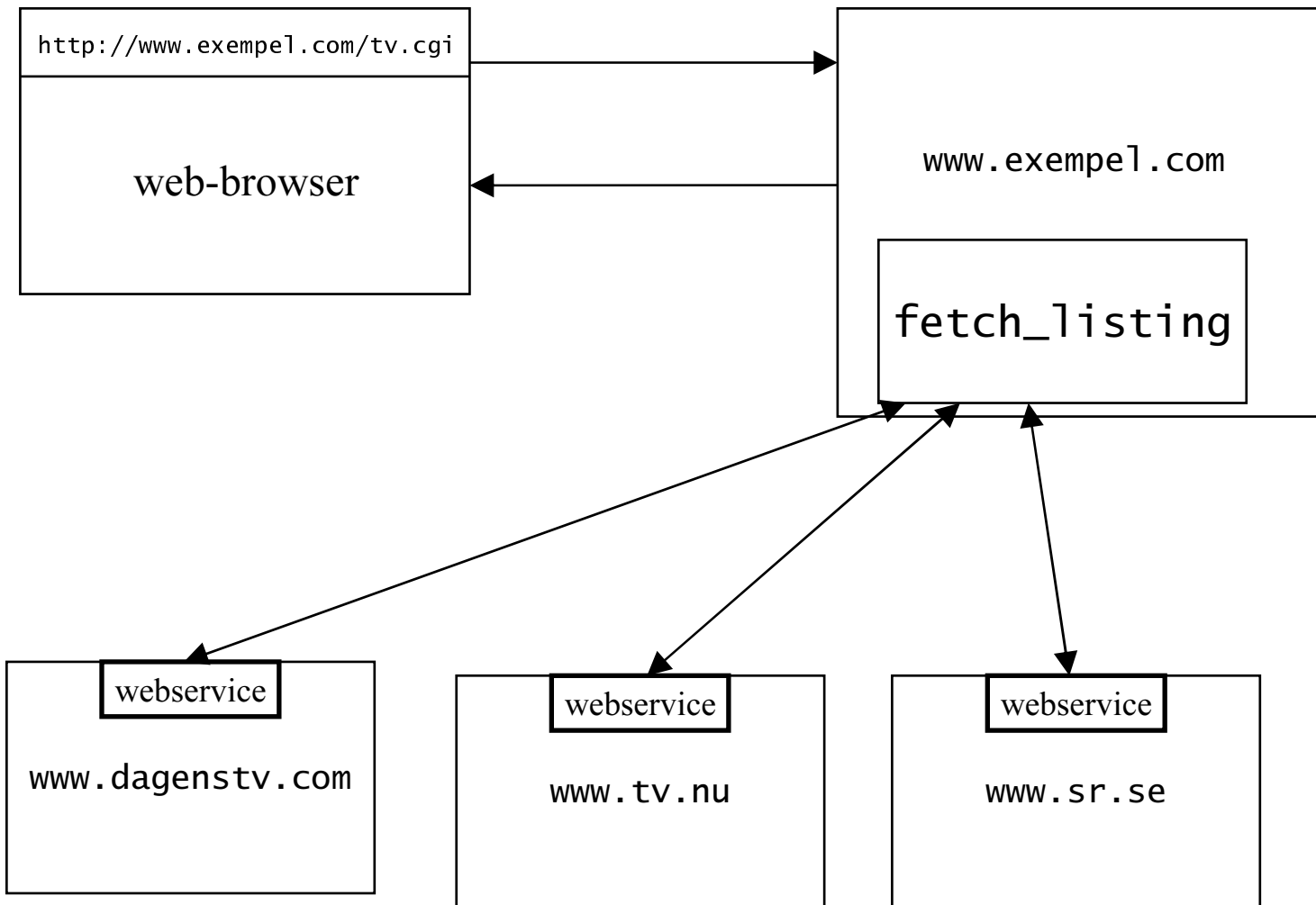
**The XML Family of Specifications: The Big Picture**

# Dagens UppLYSning

1. Tre alternativa förklaringar på vad ”Web Services” är
2. XML
3. SOAP och WSDL
4. Plattformar och Java

# Web-exempel: TV-program



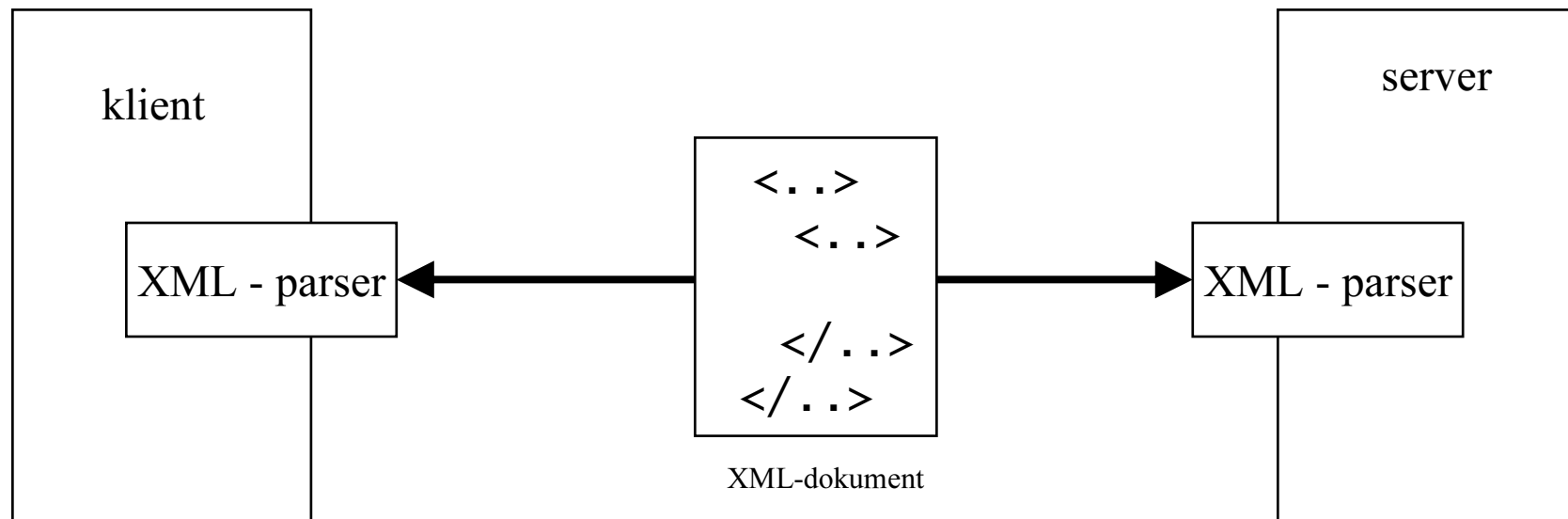


# En web av XML - parsers

Ett annat sätt att beskriva webbtjänster är att de består av tre komponenter:

- Ett XML-dokument som skickas
- En ”service” som kan parse ett XML-dokument
- En adress och port där ”servicen” finns

# Web service "core"



# Protokollstack

”Service” (SOAP/WSDL)
”Transfer/ Access” (HTTP)
Transport (TCP)
Network (IP)
Data link (Ethernet)
Physical



# XML

```
<root>  
  <child>  
    <subchild>  
      . . . . .  
    </subchild>  
  </child>  
</root>
```

# XML namespaces

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<myns:root xmlns:myns="http://www.example.com">
  <myns:child>
    <subns:child xmlns:subns="http://www.sub.nu">
      .....
    </subns:child>
  </myns:child>
  <child xmlns="http://www.defaultns.com/" />
</myns:root>
```

# XML Schema

- Definierar tillåtna element och attribut för ett XML-dokument
- Definierar struktur på dokumentet
- Definierar antal och ordning på element
- Definierar typer och default-värden för element och attribut

# XML Schema - exempel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.com">
  <element name="root">
    <complexType>
      <element name="subchild" type="string">
    </complexType>
  </element>
</schema>
```

# SOAP

- Simple Object Access Protocol
- Baserat på XML
- Ett kommunikationsprotokoll som specificerar formatet på meddelanden
- Plattforms- och språkoberoende
- Utökningsbart

## SOAP message

### Envelope

Header (kan utelämnas)

(innehåller routing, säkerhet...)

Body (måste finnas)

(innehåller meddelande, fråga...)

Fault

# SOAP exempel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope">

  <soap:header>
    <m:authheader xmlns:m="http://www.example.com/">
      <m:name> Mats </m:name>
      <m:password> abcd </m:password>
    </m:authheader>
  </soap:header>

  <soap:body xmlns:b="http://www.example.com/temp">
    <b:queryTemperature/>
  </soap:body>

</soap:envelope>
```

# WSDL

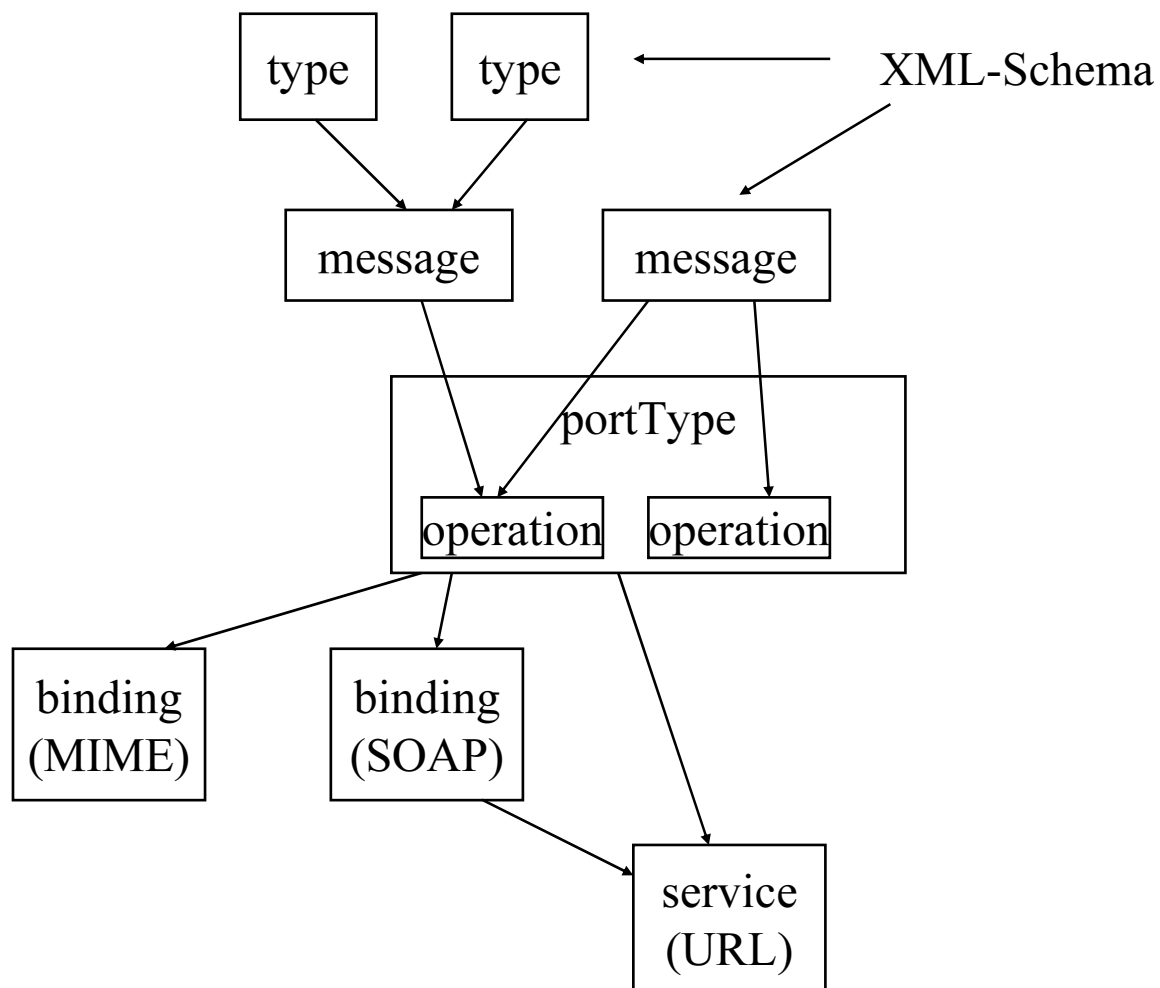
- Web Service Description Language
- Är ett XML-dokument  
(som i många implementationer kan hittas på  
`http://.../..wsdl` )
- Beskriver och lokaliserar webbtjänsten



# Delar i ett WSDL-dokument

- `<types>` definierar ingående typer med XML-Schema
- `<message>` definierar ingående meddelande-block (parametrar)
- `<portType>` definierar de portar som sedan kommer att innehålla operationer byggda av meddelande-block
- `<binding>` definierar operationer bunda till ett visst protokoll (tex SOAP)
- `<service>` portar och bindningar definierar en service

# WSDL-delarnas beroenden



# WSDL - exempel

```
<?xml version="1.0"?>
<wsdl:definitions name="Exempel"
  targetNamespace="http://www.example.com/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

  <wsdl:types>
    <xsd:schema>
      <xsd:complexType name="auth">
        <xsd:sequence>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="password" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>
</wsdl:definitions>
```

...

# Plattformer för Web Services

- IBM – WebSphere
- Microsoft - .NET
- Sun – Sun ONE Platform
- Systinet – WASP
- Apache – Axis
- Perl – SOAP::Lite

# Möjliga servicenivåer för Java

- Java-nivån – direkta metदानrop som ett ”remote procedure call” via SOAP
- XML/SOAP – ger direkt tillgång till SOAP-meddelande (som är parsat till ett DOM-träd) via SAAJ
- Raw – läser och skriver direkt på InputStream och OutputStream

# Java - exempel

```
//: HelloService.java  
  
public class HelloService {  
  
    public String hello(String message) {  
  
        return "Hello, " + message + "!";  
    }  
}
```

# Java – server - exempel

```
//: HelloServer.java
import org.systinet.wasp.Wasp;
import org.systinet.wasp.webservice.Registry;

public class HelloServer {

    public static void main(String[] args) throws Exception {
        // initialize WASP and start server like 'HTTP'
        Wasp.startServer("http://localhost:6060");

        // publish HelloService
        Registry.publish("/HelloService", new HelloService());
    }
}
```

# Java – klient - exempel

```
//: HelloClient.java
import org.systinet.wasp.webservice.Registry;

public class HelloClient {

    public static void main(String[] args) throws Exception {

        // lookup of HelloService
        HelloService helloService = (HelloService) Registry.lookup(
            "http://localhost:6060/HelloService/WSDL",
            HelloService.class);

        // call HelloService and print out a response message
        System.out.println(helloService.hello("world"));
    }
}
```



# Åsikter om web services

- Borde kanske heta ”internet services”
- Är inte distribuerade objekt, men däremot ett distribuerat system
- SOAP borde bytas till annan förkortning