"Everything you always wanted to know about Web services*

*but were afraid to ask"

Antonio Villegas Niño

avillegas@essi.upc.edu

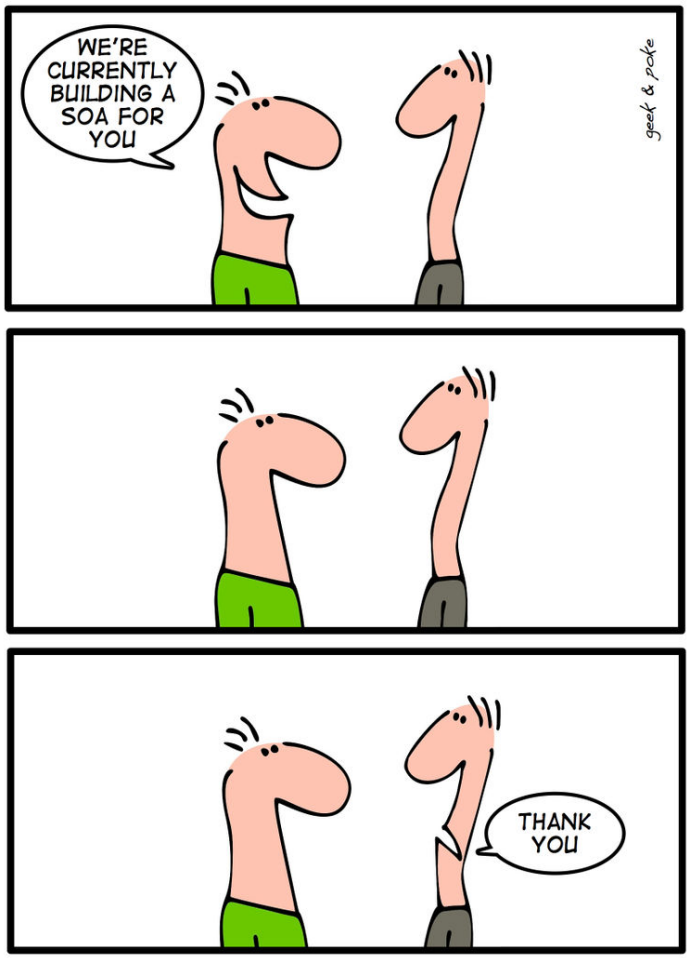
@avillegasn

Services and Information Systems Engineering Department
Universitat Politècnica de Catalunya

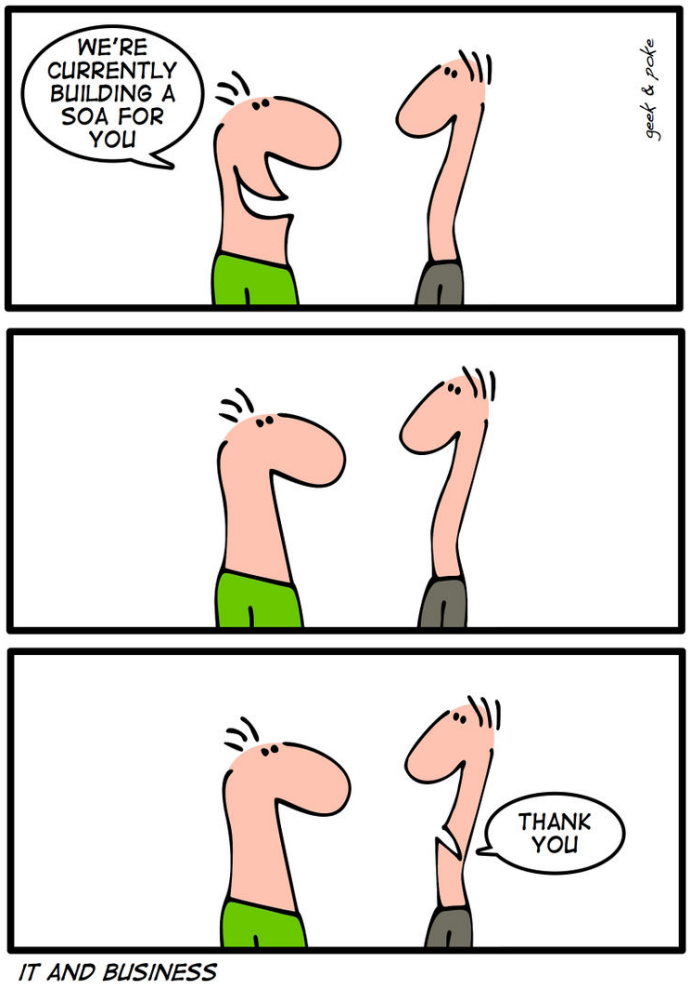
October 20, 2010

Outline

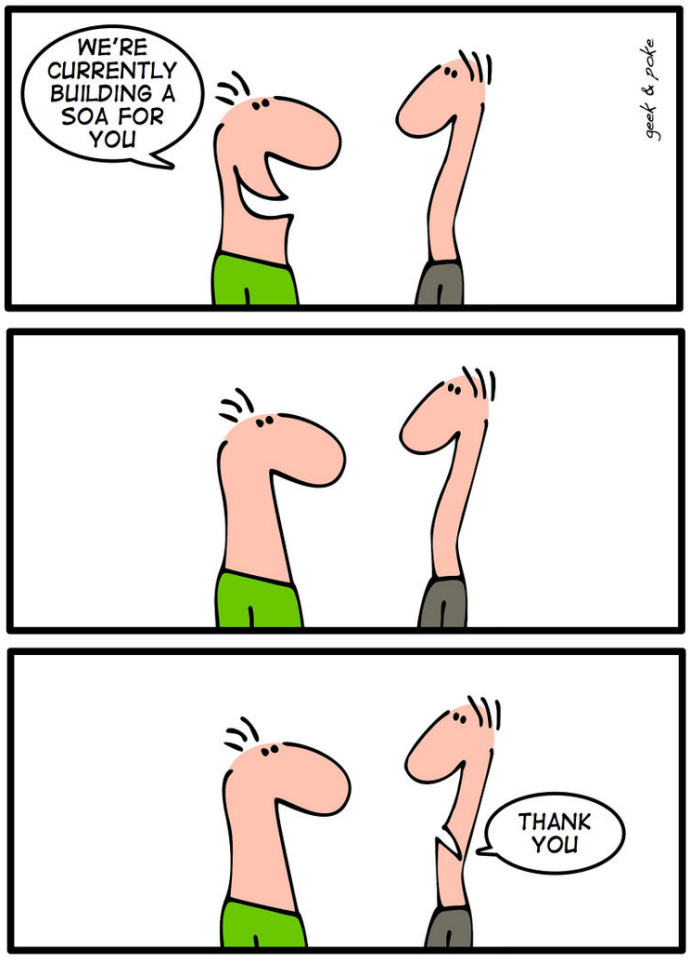
SOA


WE'RE CURRENTLY BUILDING A SOA FOR YOU
geek & poke
THANK YOU
IT AND BUSINESS

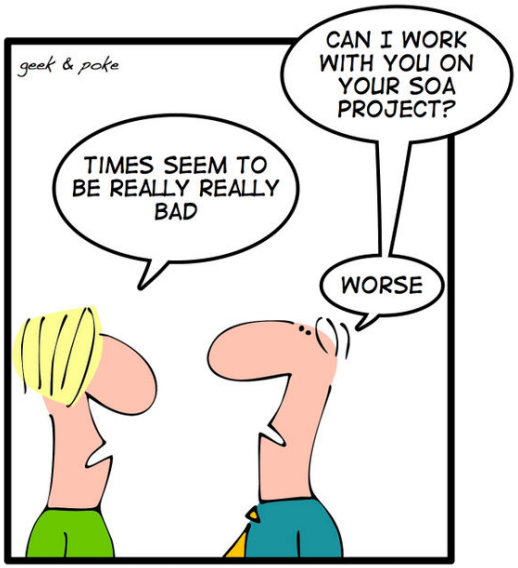
SOA


WE'RE CURRENTLY BUILDING A SOA FOR YOU
geek & poke
THANK YOU
IT AND BUSINESS

SOA

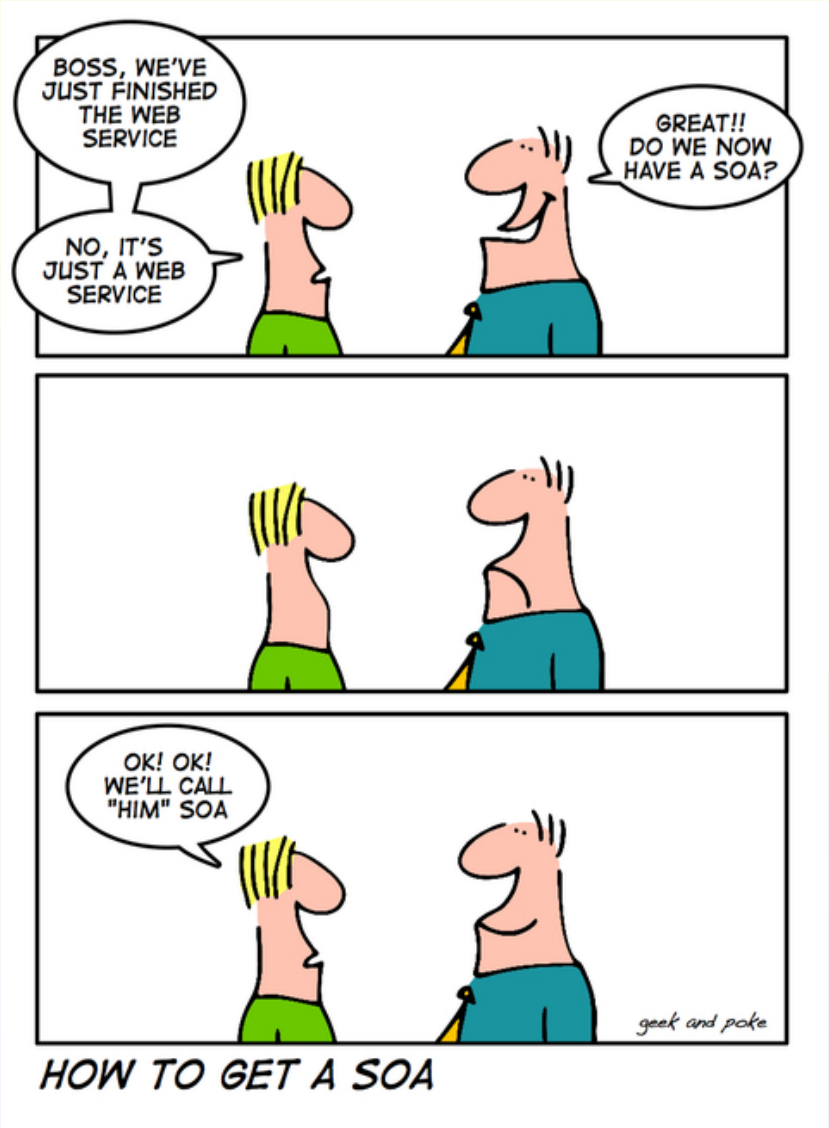


IT AND BUSINESS



BUSINESS IS MORE AND MORE INTERESTED IN SOA

SOA ≠ Web services



Web Services

This presentation focuses on Web services.

What are Web services?

- In essence, a **technology** for application **integration** and **interoperability** based on open standards.
- Application **functionality packaged** as a single unit and **exposed** to the network.
- The **W3C** defines a Web service as a **software system** designed to support **interoperable** machine to machine **interaction** over a network.

Web Services

This presentation focuses on Web services.

What are Web services?

- In essence, a **technology** for application **integration** and **interoperability** based on open standards.
- Application **functionality packaged** as a single unit and **exposed** to the network.
- The **W3C** defines a Web service as a **software system** designed to support **interoperable** machine to machine **interaction** over a network.

Web Services

This presentation focuses on Web services.

What are Web services?

- In essence, a **technology** for application **integration** and **interoperability** based on open standards.
- Application **functionality packaged** as a single unit and **exposed** to the network.
- The **W3C** defines a Web service as a **software system** designed to support **interoperable** machine to machine **interaction** over a network.

Web Services

This presentation focuses on Web services.

What are Web services?

- In essence, a **technology** for application **integration** and **interoperability** based on open standards.
- Application **functionality packaged** as a single unit and **exposed** to the network.
- The **W3C** defines a Web service as a **software system** designed to support **interoperable** machine to machine **interaction** over a network.

Web Services

This presentation focuses on Web services.

What are Web services?

- In essence, a **technology** for application **integration** and **interoperability** based on open standards.
- Application **functionality packaged** as a single unit and **exposed** to the network.
- The **W3C** defines a Web service as a **software system** designed to support **interoperable** machine to machine **interaction** over a network.

Web Services
Some Examples

Vancouver

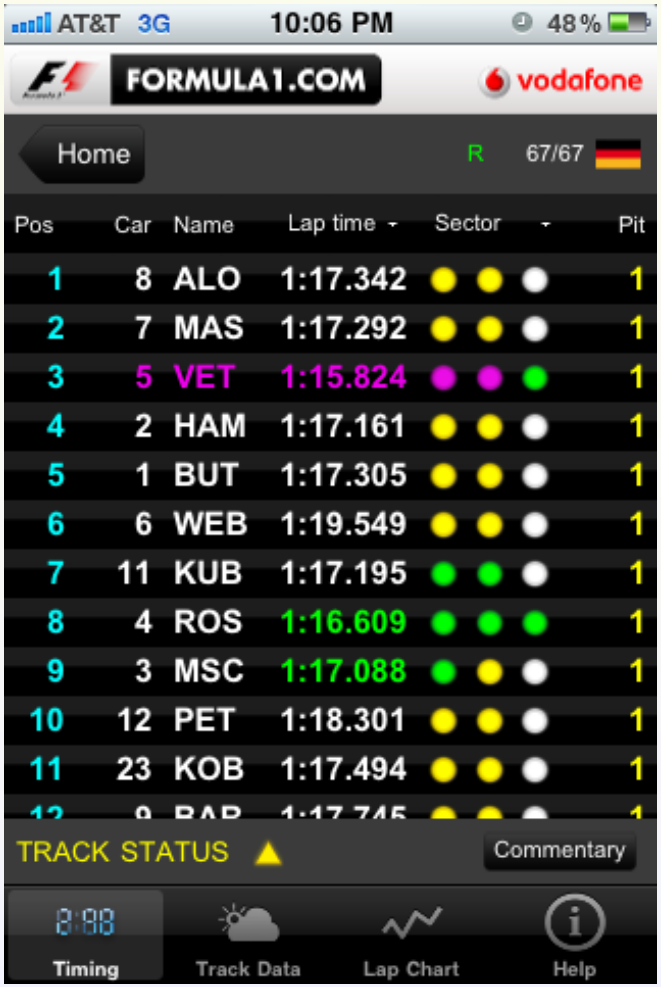
Current Conditions More info +

11 °C	Observed at:	Vancouver Int'l Airport		
	Date:	9:00 AM PDT Monday 27 April 2009		
	Condition:	Sunny	Temperature:	11.0°C
	Pressure:	101.8 kPa	Dewpoint:	4.8°C
	Tendency:	falling	Humidity:	66 %
	Visibility:	48 km	Wind:	NW 21 km/h
	Air Quality Health Index:	3		

Forecast More info +

Today	Tue	Wed	Thu	Fri	Sat	Sun
16°C	17°C 9°C	16°C 9°C	16°C 9°C	16°C 7°C	16°C 9°C 60%	16°C 10°C 60%

Web Services
Some Examples

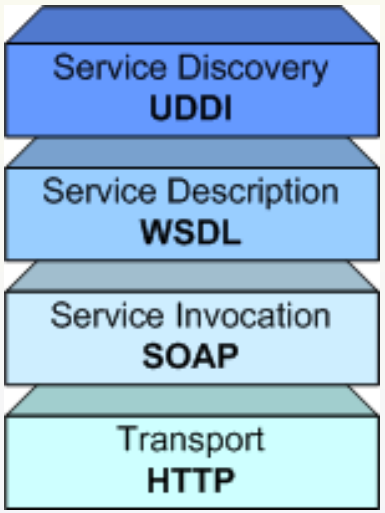


Web Services
Some Examples

Web Services
Internals

The **Web services framework** is divided into three areas:

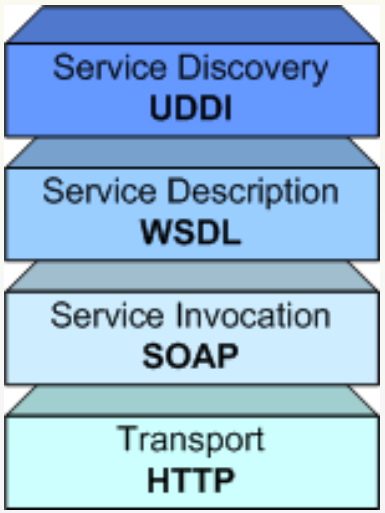
- **Invocation:** the simple object access protocol (SOAP) which enables communication among Web services.
- **Description:** the Web Services Description Language (WSDL) which provides a formal, computer-readable description of Web services.
- **Discovery:** the Universal Description, Discovery, and Integration (UDDI) directory which is a registry of Web services descriptions.



Web Services
Internals

The **Web services framework** is divided into three areas:

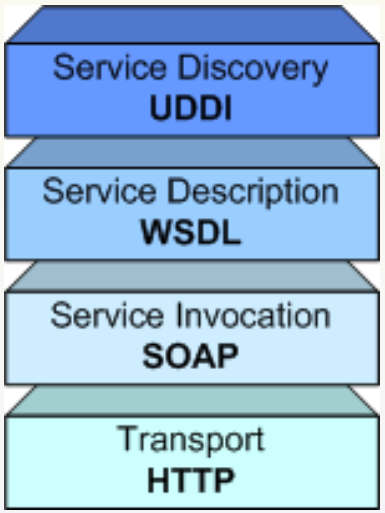
- **Invocation:** the simple object access protocol (SOAP) which enables communication among Web services.
- **Description:** the Web Services Description Language (WSDL) which provides a formal, computer-readable description of Web services.
- **Discovery:** the Universal Description, Discovery, and Integration (UDDI) directory which is a registry of Web services descriptions.



Web Services
Internals

The **Web services framework** is divided into three areas:

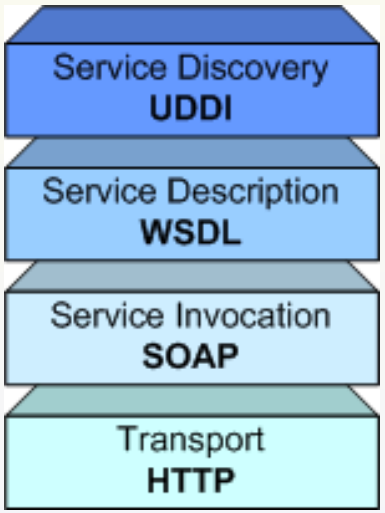
- **Invocation:** the simple object access protocol (SOAP) which enables communication among Web services.
- **Description:** the Web Services Description Language (WSDL) which provides a formal, computer-readable description of Web services.
- **Discovery:** the Universal Description, Discovery, and Integration (UDDI) directory which is a registry of Web services descriptions.



Web Services
Internals

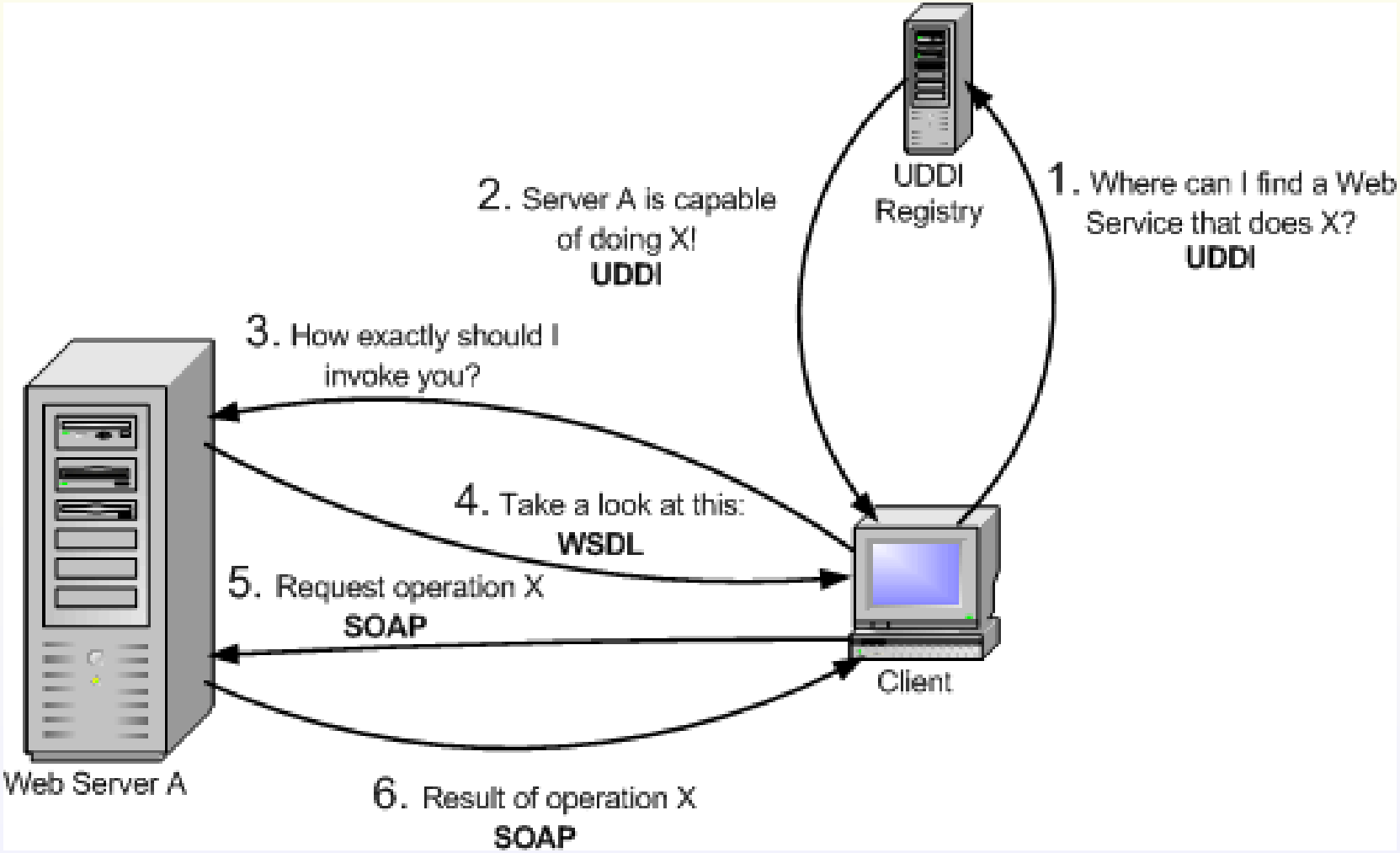
The **Web services framework** is divided into three areas:

- **Invocation:** the simple object access protocol (SOAP) which enables communication among Web services.
- **Description:** the Web Services Description Language (WSDL) which provides a formal, computer-readable description of Web services.
- **Discovery:** the Universal Description, Discovery, and Integration (UDDI) directory which is a registry of Web services descriptions.



How does it work?

UDDI+WSDL+SOAP



Simple Object Access Protocol

Communication Mecanisms:

- Platform-independent
- International
- Secure
- As lightweight as possible

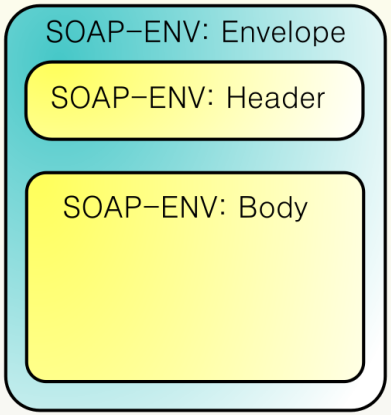
Simple Object Access Protocol

SOAP is an XML-based protocol for messaging and remote procedure calls. Rather than define a new transport protocol, SOAP works on existing transports, such as HTTP, SMTP, etc.

Simple Object Access Protocol

Communication Mecanisms:

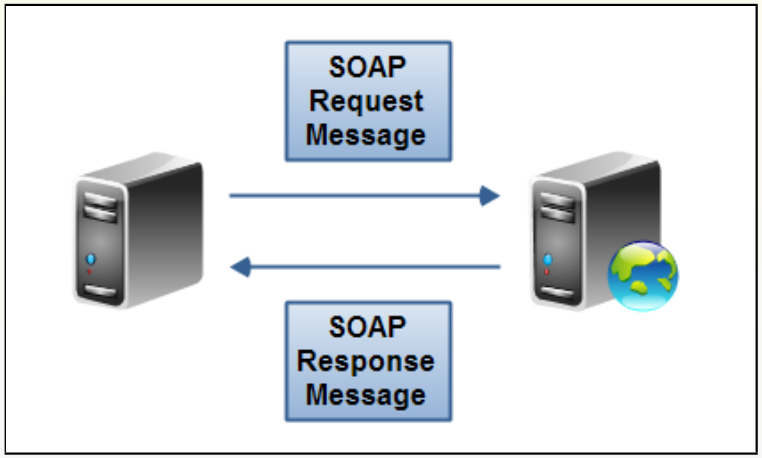
- Platform-independent
- International
- Secure
- As lightweight as possible



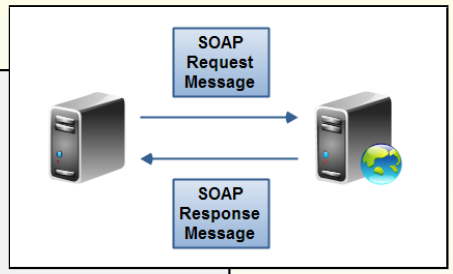
Simple Object Access Protocol

SOAP is an XML-based protocol for messaging and remote procedure calls. Rather than define a new transport protocol, SOAP works on existing transports, such as HTTP, SMTP, etc.

SOAP Message Interchange

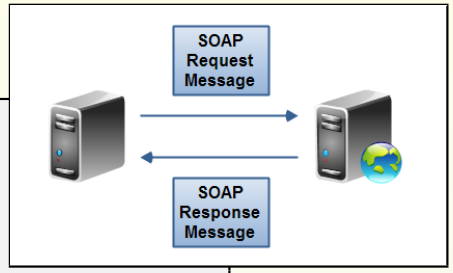


SOAP Request Message




```
POST /travelservice
SOAPAction: "http://www.acme-travel.com/flightinfo"
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">

 <SOAP:Header>
 </SOAP:Header>

 <SOAP:Body>
  <m:GetFlightInfo
     xmlns:m="http://www.acme-travel.com/flightinfo"
     SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

     <airlineName xsi:type="xsd:string">OA</airlineName>
     <flightNumber xsi:type="xsd:int">815</flightNumber>

  </m:GetFlightInfo>
 </SOAP:Body>

</SOAP:Envelope>
```

SOAP Response Message




```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">

 <SOAP:Body>
  <m:GetFlightInfoResponse
   xmlns:m="http://www.acme-travel.com/flightinfo"
   SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

   <flightInfo>
    <gate xsi:type="xsd:int">10</gate>
    <status xsi:type="xsd:string">ON TIME</status>
   </flightInfo>

  </m:GetFlightInfoResponse>
 </SOAP:Body>

</SOAP:Envelope>
```

Web Services Description Language

Web Services Description Language

WSDL is an **XML** format developed by IBM and Microsoft to **describe Web services** as collections of communication end points that can exchange certain messages.

A WSDL document describes a Web service's interface and provides users with a point of contact about **what messages must be exchanged** to successfully interact with a service.

Web Services Description Language

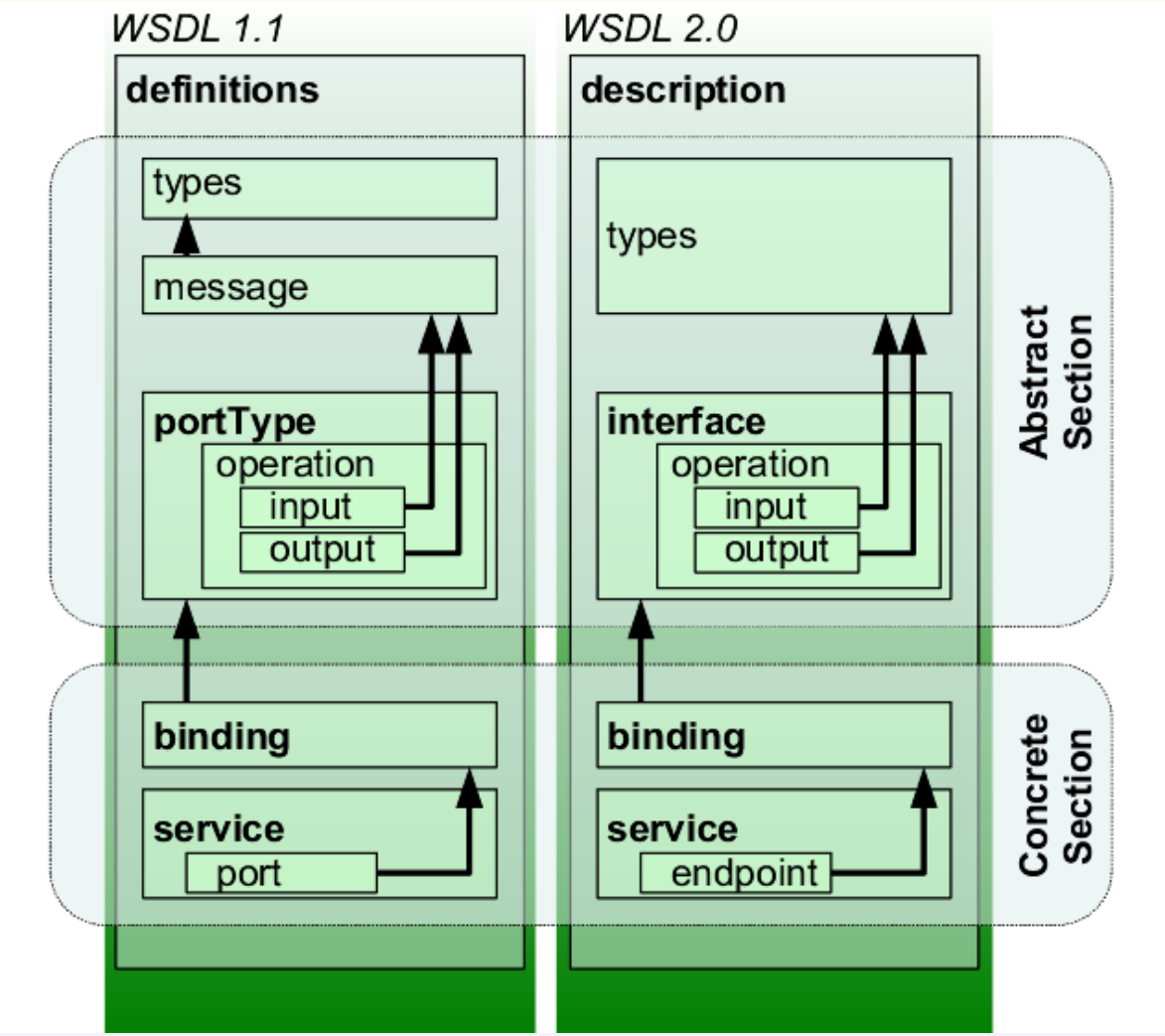
Web Services Description Language

WSDL is an **XML** format developed by IBM and Microsoft to **describe Web services** as collections of communication end points that can exchange certain messages.

A WSDL document describes a Web service's interface and provides users with a point of contact about **what messages must be exchanged** to successfully interact with a service.

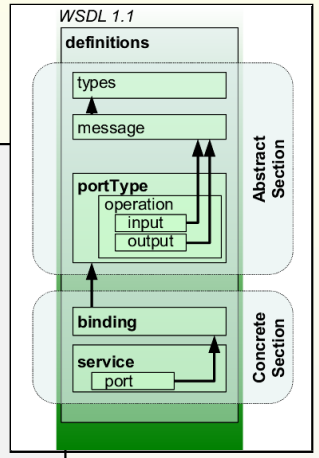
Web Services Description Language

XML Structure



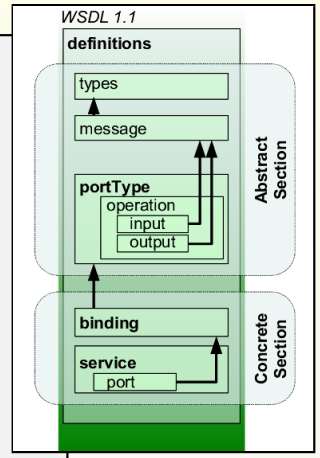
WSDL: Abstract Section

```
<wsdl:types>
   <!-- FlightInfoType definition (gate:int, status:string) -->
</wsdl:types>

<wsdl:message name="GetFlightInfoInput">
   <wsdl:part name="airlineName" type="xsd:string"/>
   <wsdl:part name="flightNumber" type="xsd:int"/>
</wsdl:message>

<wsdl:message name="GetFlightInfoOutput">
   <wsdl:part name="flightInfo" type="xsd:FlightInfoType"/>
</wsdl:message>

<wsdl:portType name="AirportServicePortType">
   <wsdl:operation name="GetFlightInfo">
      <wsdl:input message="tns:GetFlightInfoInput"/>
      <wsdl:output message="tns:GetFlightInfoOutput"/>
   </wsdl:operation>
</wsdl:portType>
```



WSDL: Concrete Section

```
<binding name="AirportServiceSoapBinding"
         type="tns:AirportServicePortType">
 <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>

 <operation name="GetFlightInfo">
    <soap:operation style="rpc" soapAction="http://acme-travel/
         flightinfo"/>
    <input>
       <soap:body use="encoded"
         namespace="http://acme-travel.com/flightinfo"
         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
       <soap:body use="encoded"
         namespace="http://acme-travel.com/flightinfo"
         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
 </operation>
</binding>

<service name="travelservice">
  <port name="travelservicePort"
        binding="tns:AirportServiceSoapBinding">
    <soap:address location="http://acmetravel.com/travelservice"/>
  </port>
</service>
```



Universal Description, Discovery, and Integration

Universal Description, Discovery, and Integration

The **UDDI** specification offers users a unified and systematic way to **find service providers** through a centralized registry of services that is roughly equivalent to an automated online **phone directory of Web services.**

Registry access is accomplished using a standard **SOAP API** for both querying and updating.

Universal Description, Discovery, and Integration

Universal Description, Discovery, and Integration

The **UDDI** specification offers users a unified and systematic way to **find service providers** through a centralized registry of services that is roughly equivalent to an automated online **phone directory of Web services.**

Registry access is accomplished using a standard **SOAP API** for both querying and updating.

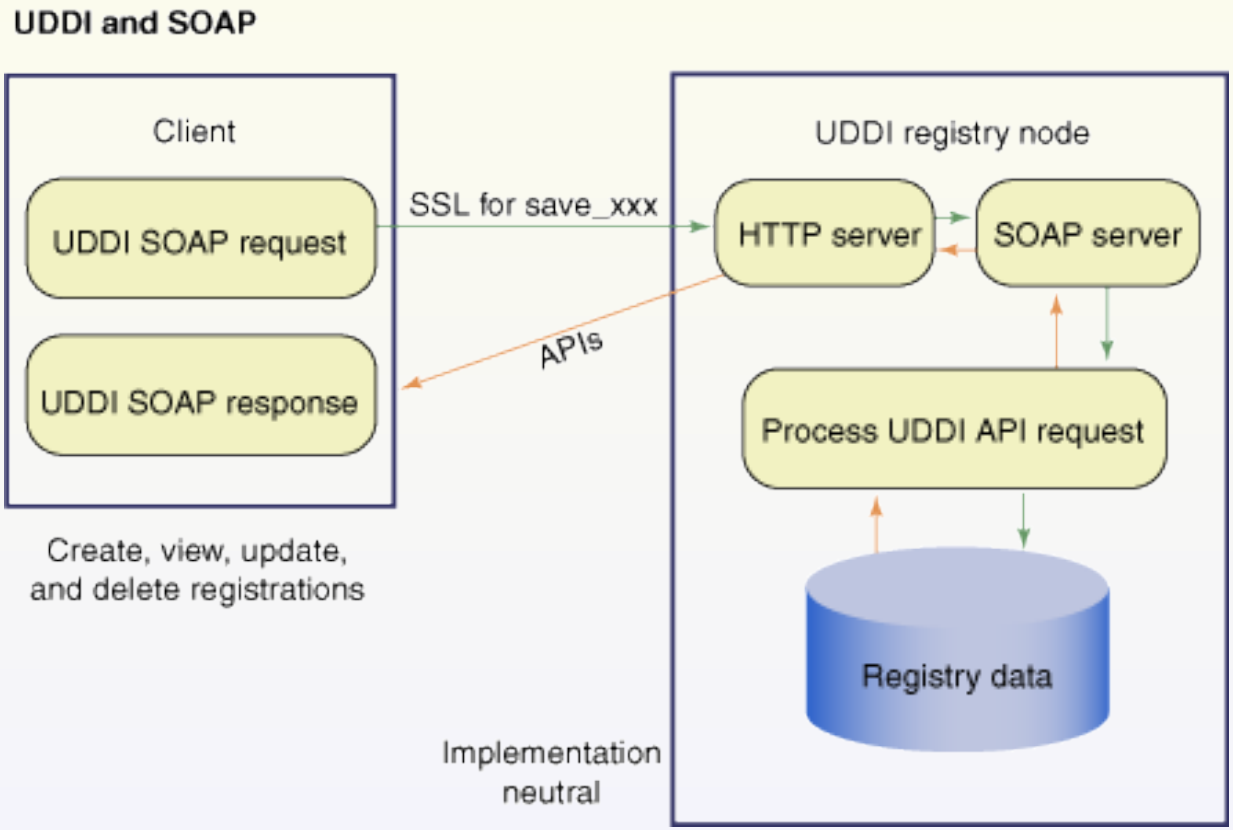
UDDI
UDDI API

```
Inquiry Operations:
Find
     find_business
     find_service
     find_binding
     find_tModel
Get details
     get_businessDetail
     get_serviceDetail
     get_bindingDetail
     get_tModelDetail
     get_registeredInfo
```

```
Publishing Operations:
 Save
      save_business
      save_service
      save_binding
      save_tModel
 Delete
      delete_business
      delete_service
      delete_binding
      delete_tModel
      get_registeredInfo
 Security
      get_authToken
      discard_authToken
```

UDDI

Flow of UDDI messages between Client and Registry

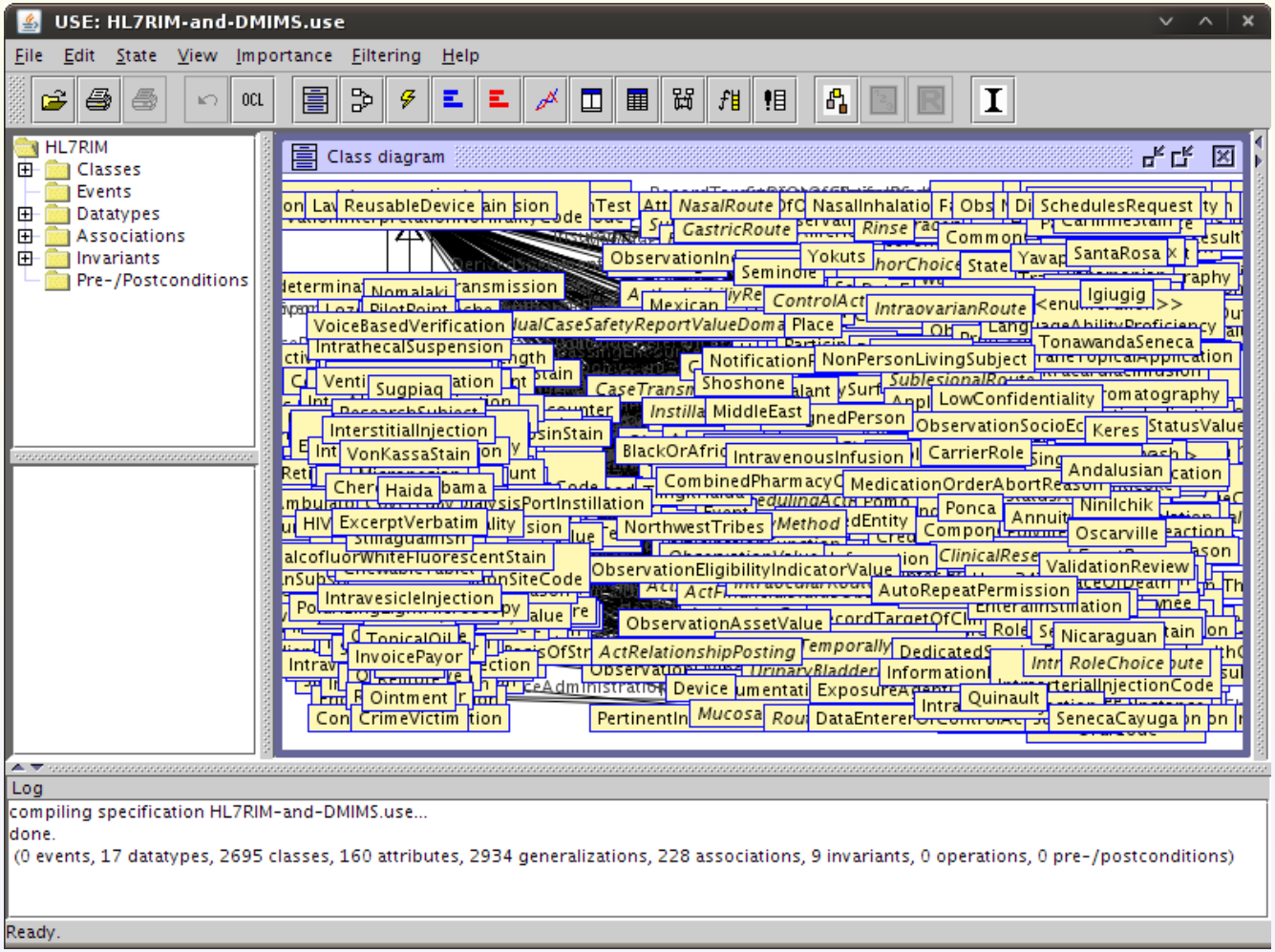


From Desktop to Web

Example of Web service migration from Java Desktop App

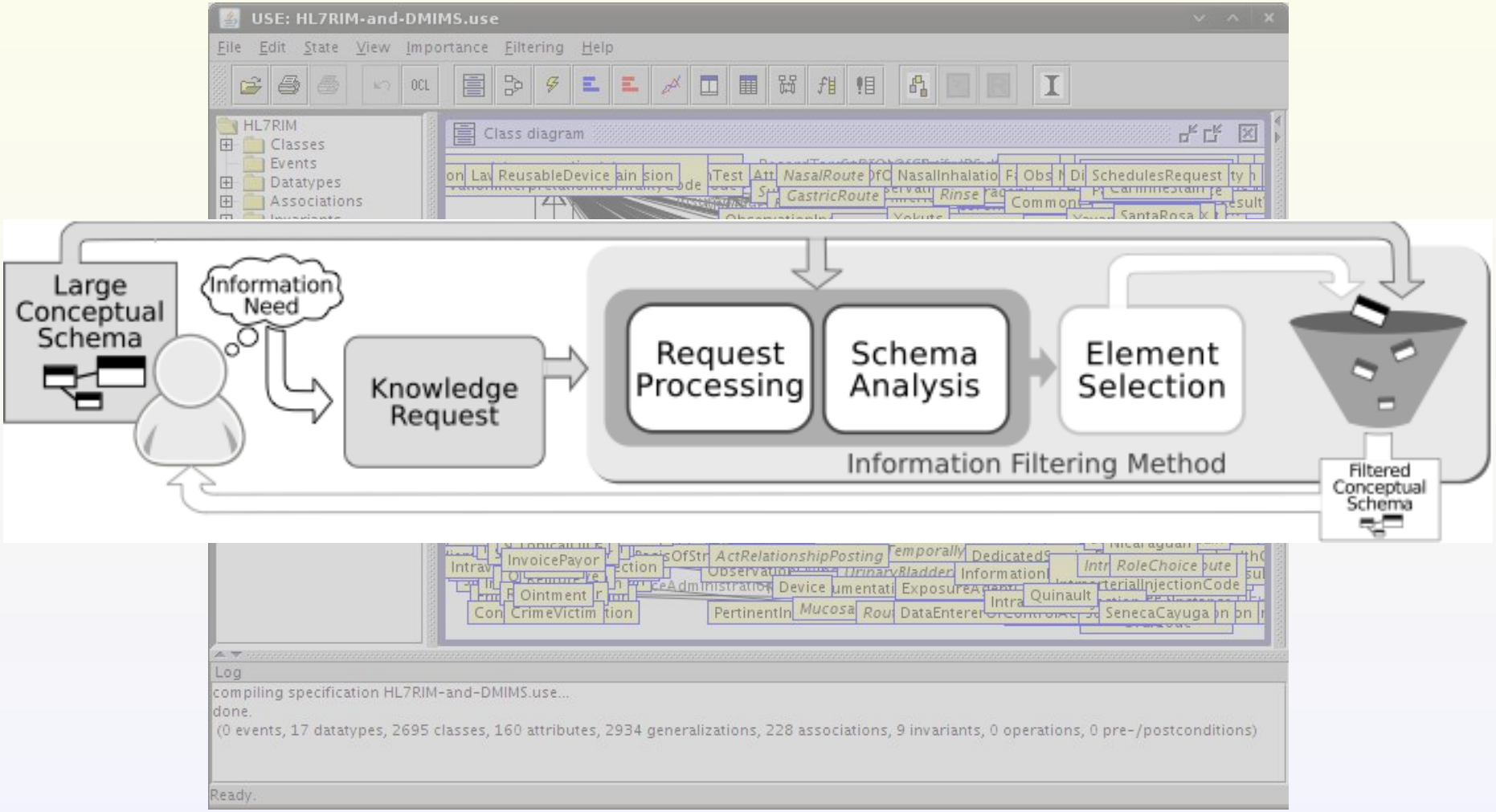
From Desktop to Web

Example of Web service migration from Java Desktop App



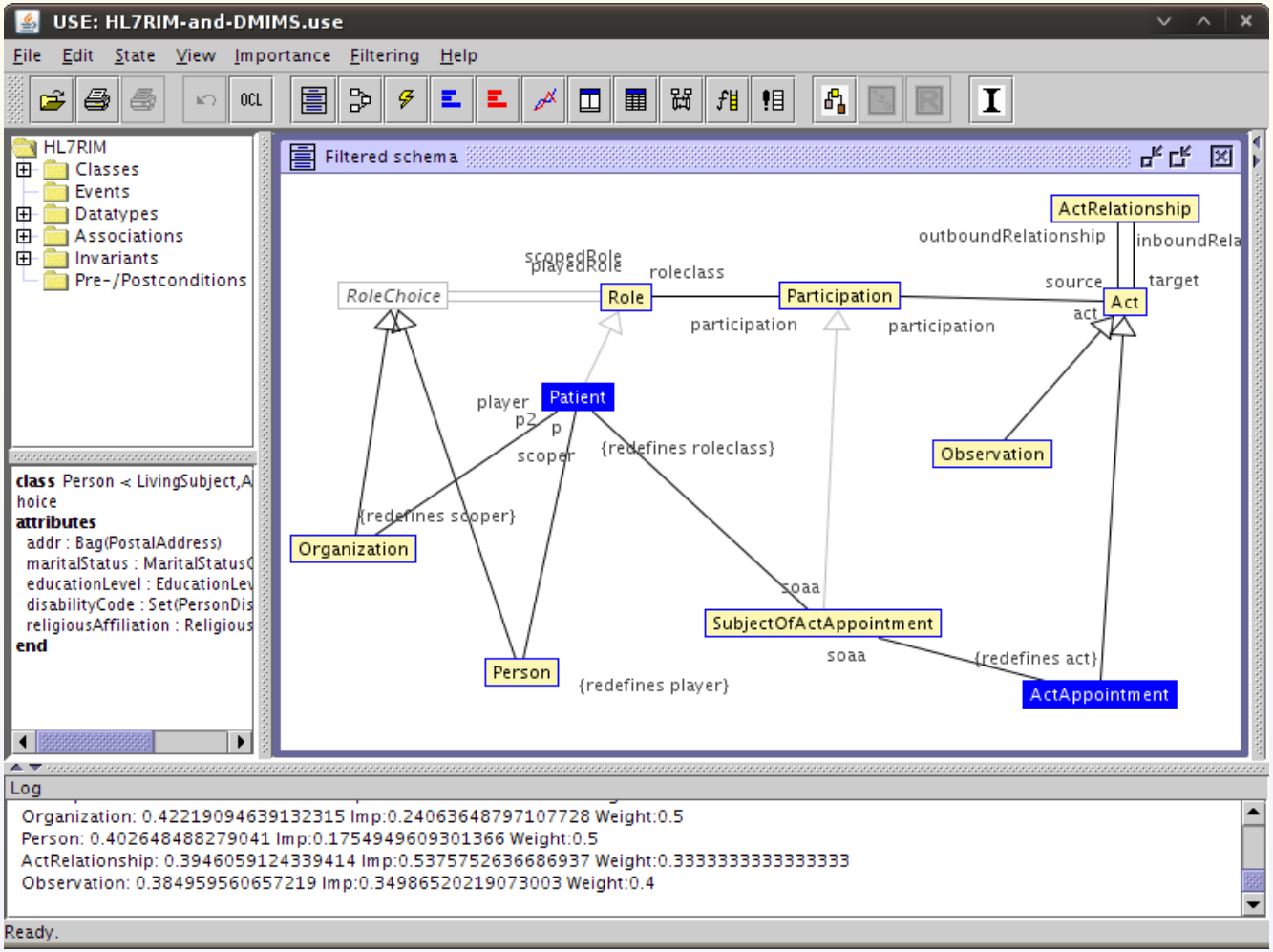
From Desktop to Web

Example of Web service migration from Java Desktop App



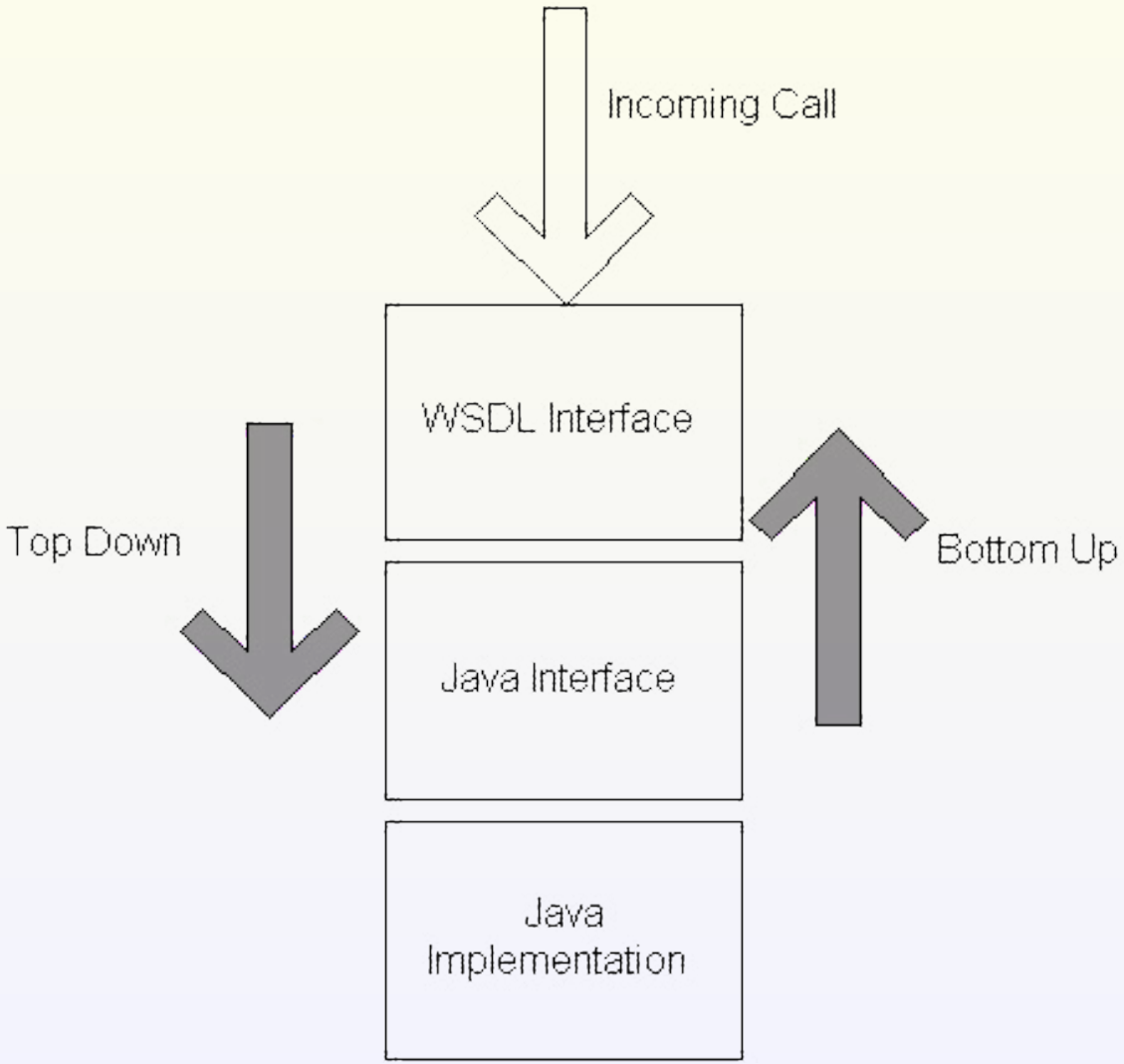
From Desktop to Web

Example of Web service migration from Java Desktop App



From Desktop to Web

Bottom-up approach



First problem

SOAP

WSDL **UDDI**

Do we have to master these technologies?

First problem

SOAP

WSDL **UDDI**

Do we have to master these technologies?

Not at all!

Eclipse WTP + AXIS2 + Tomcat = 1-click Web service generator

Setting up the environment
Eclipse WTP + Axis2 + Tomcat

Eclipse WTP

- Provides functionality to develop web functionality with Java
- Install via Eclipse Update Manager

Axis 2

- Axis2 is a Web Services / SOAP / WSDL engine
- Currently Axis2 1.5 is not supported. Make sure you download Axis 1.4.1
- Extract the Axis2 zip and point Eclipse WTP on it (through Eclipse Preferences)

Tomcat

- Open source software implementation of the Java Servlet and JavaServer Pages technologies.
- Download, extract zip and point Eclipse WTP on it (through Eclipse Preferences)

Setting up the environment
Eclipse WTP + Axis2 + Tomcat

Eclipse WTP

- Provides functionality to develop web functionality with Java
- Install via Eclipse Update Manager

Axis 2

- Axis2 is a Web Services / SOAP / WSDL engine
- Currently Axis2 1.5 is not supported. Make sure you download Axis 1.4.1
- Extract the Axis2 zip and point Eclipse WTP on it (through Eclipse Preferences)

Tomcat

- Open source software implementation of the Java Servlet and JavaServer Pages technologies.
- Download, extract zip and point Eclipse WTP on it (through Eclipse Preferences)

Setting up the environment

Eclipse WTP + Axis2 + Tomcat

Eclipse WTP

- Provides functionality to develop web functionality with Java
- Install via Eclipse Update Manager

Axis 2

- Axis2 is a Web Services / SOAP / WSDL engine
- Currently Axis2 1.5 is not supported. Make sure you download Axis 1.4.1
- Extract the Axis2 zip and point Eclipse WTP on it (through Eclipse Preferences)

Tomcat

- Open source software implementation of the Java Servlet and JavaServer Pages technologies.
- Download, extract zip and point Eclipse WTP on it (through Eclipse Preferences)

Setting up the environment

Eclipse WTP + Axis2 + Tomcat

Eclipse WTP

- Provides functionality to develop web functionality with Java
- Install via Eclipse Update Manager

Axis 2

- Axis2 is a Web Services / SOAP / WSDL engine
- Currently Axis2 1.5 is not supported. Make sure you download Axis 1.4.1
- Extract the Axis2 zip and point Eclipse WTP on it (through Eclipse Preferences)

Tomcat

- Open source software implementation of the Java Servlet and JavaServer Pages technologies.
- Download, extract zip and point Eclipse WTP on it (through Eclipse Preferences)

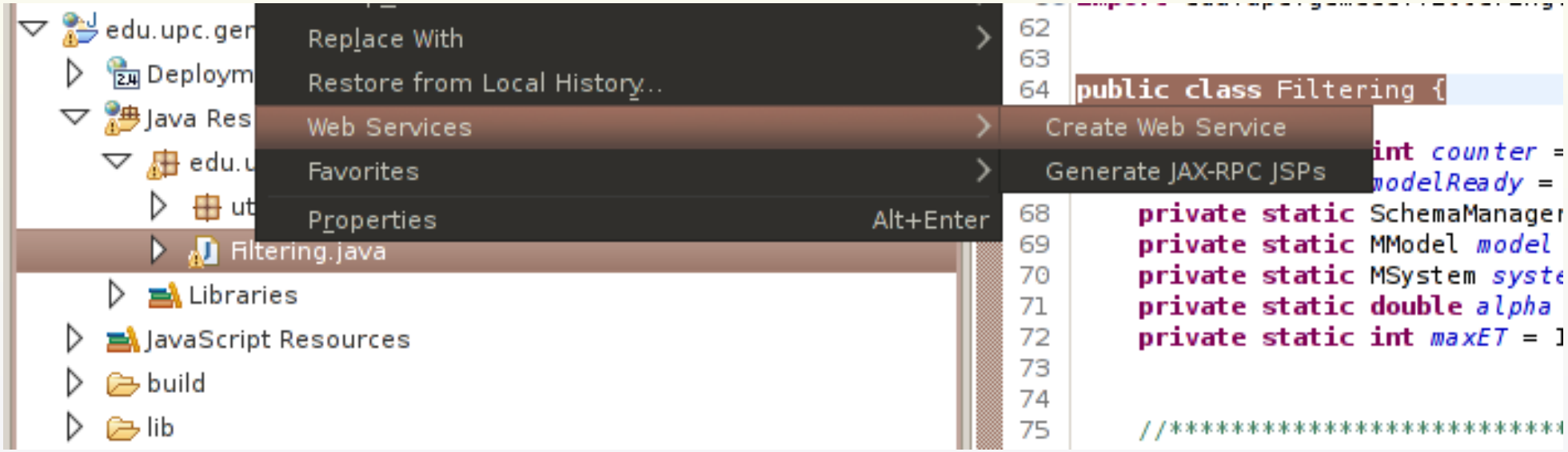
Create a Web service from Java

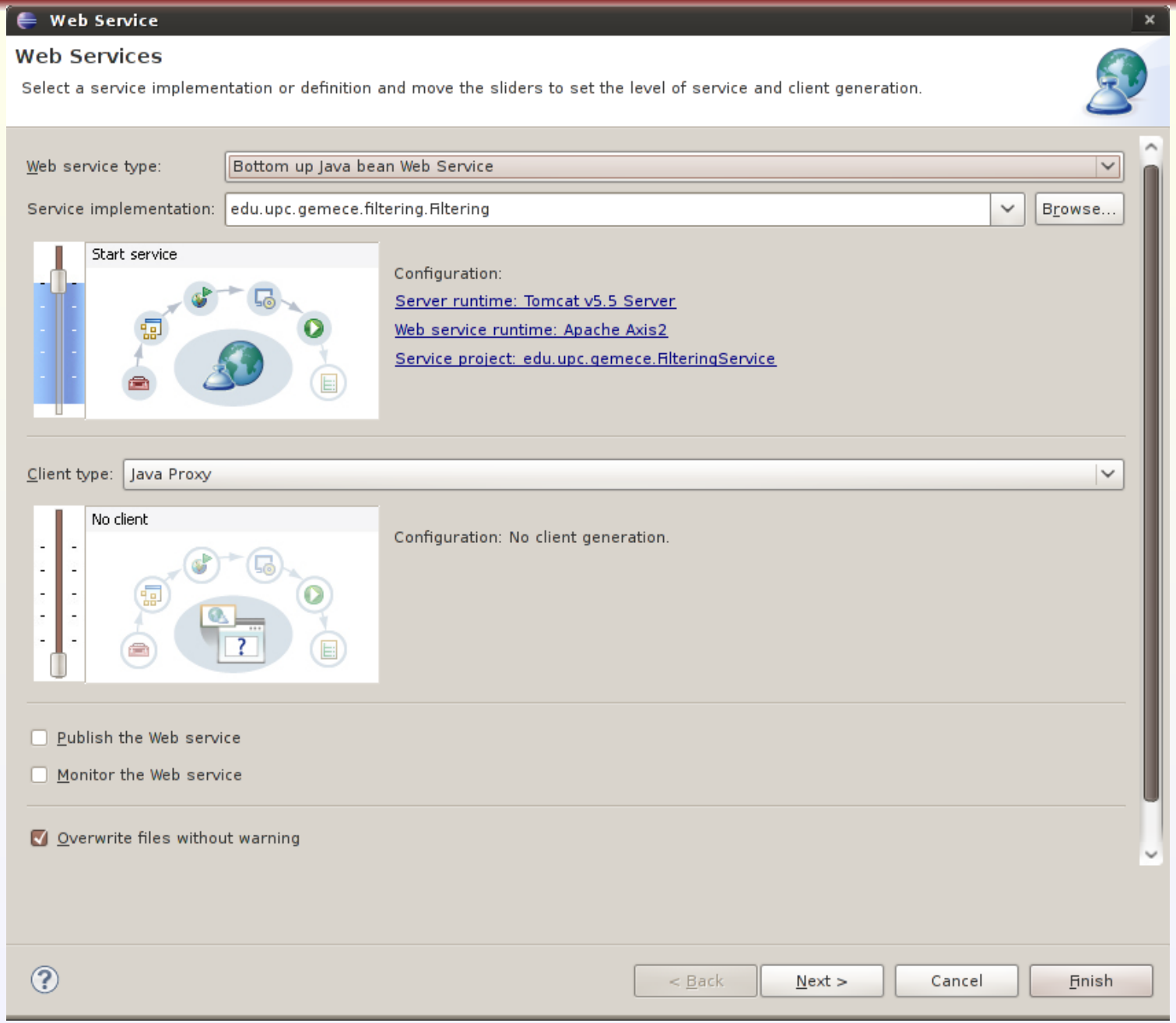
Java Project

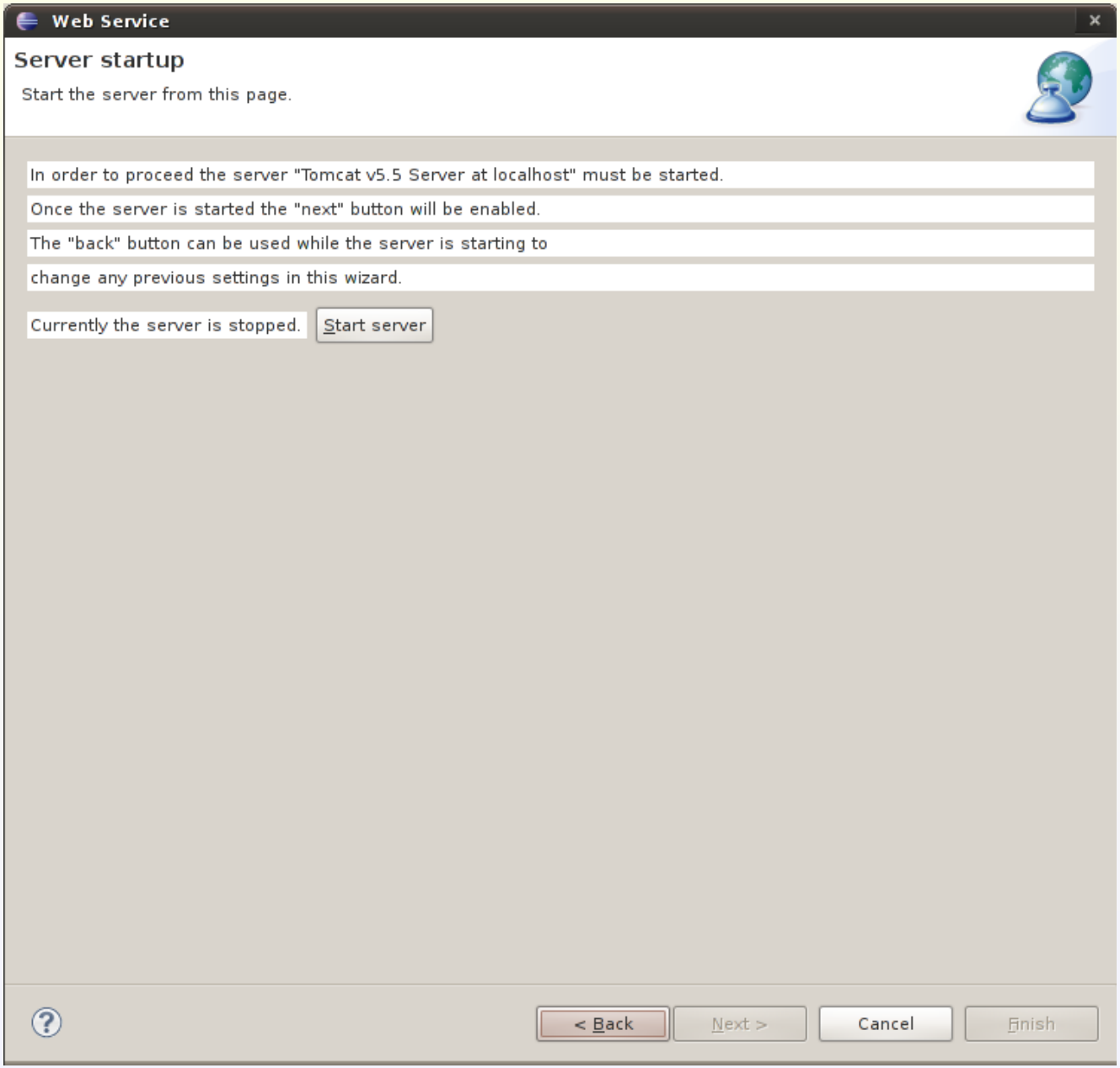
- Create a new Dynamic Web project
- Create a Java class with the operations of the Web service

```
public class Filtering {

    // keep the model up between messages
    private static MModel model;

    public static void start() {
        // load schema
    }
    public static String filter(String[] focusSet) {
        // compute a filtered conceptual schema
        // same code as in USE Environment
        // return a XMI representation
    }
}
```

Create a Web service from Java



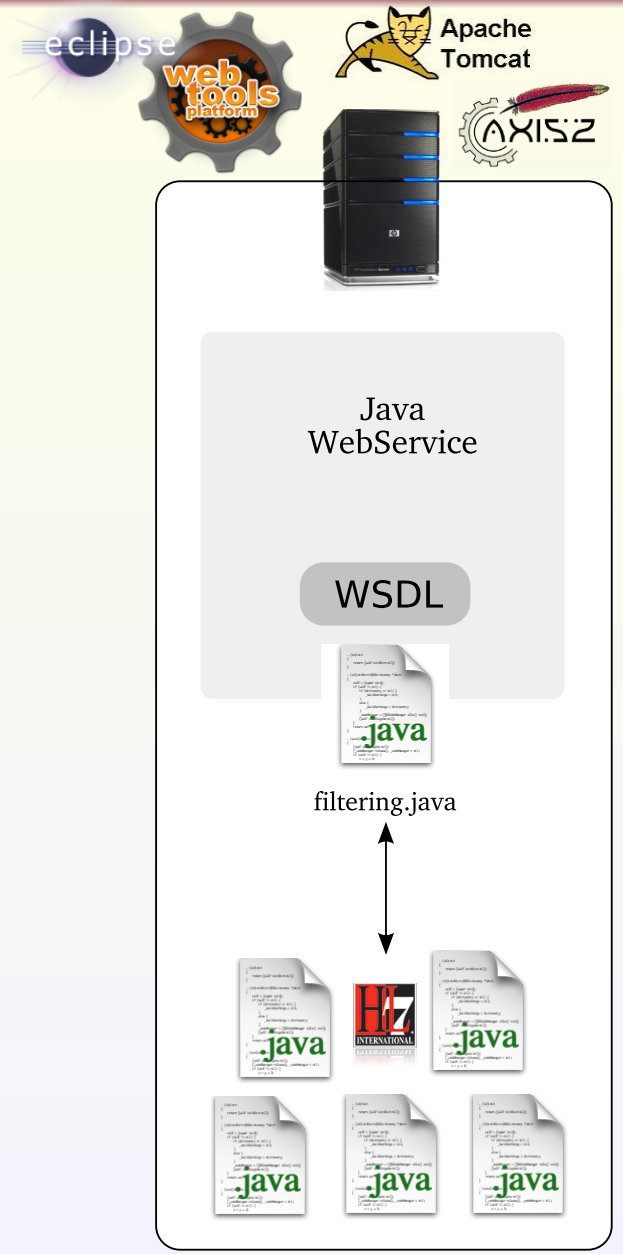

Web Service
Web Services
Select a service implementation or definition and move the sliders to set the level of service and client generation.
Web service type:
Bottom up Java bean Web Service
Service implementation:
edu.upc.gemece.filtering.Filtering
Browse...
Start service
Configuration:
Server runtime: Tomcat v5.5 Server
Web service runtime: Apache Axis2
Service project: edu.upc.gemece.FilteringService
Client type:
Java Proxy
No client
Configuration: No client generation.
Publish the Web service
Monitor the Web service
Overwrite files without warning
< Back
Next >
Cancel
Finish


Web Service
Server startup
Start the server from this page.
In order to proceed the server "Tomcat v5.5 Server at localhost" must be started.
Once the server is started the "next" button will be enabled.
The "back" button can be used while the server is starting to
change any previous settings in this wizard.
Currently the server is stopped.
Start server
< Back
Next >
Cancel
Finish

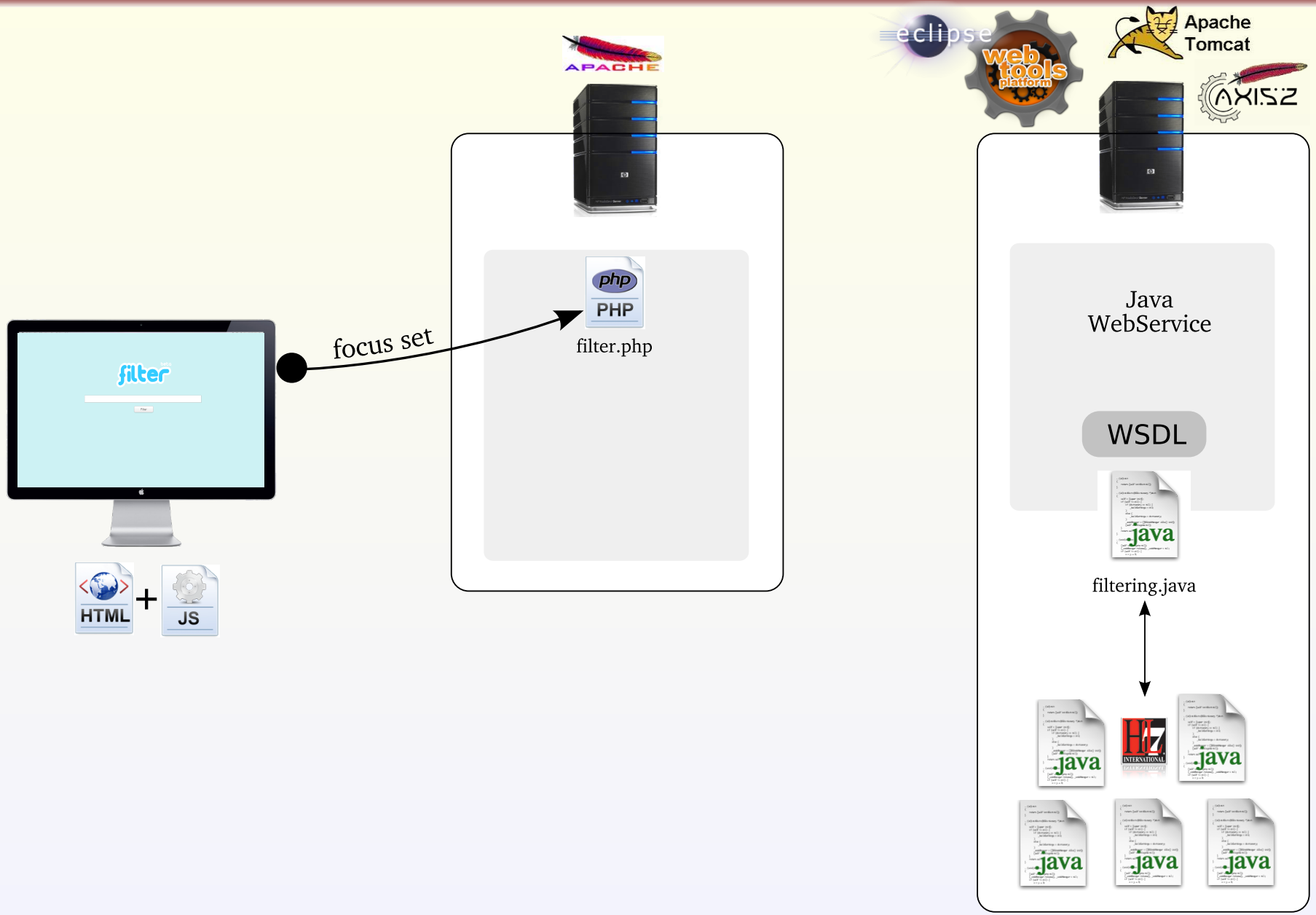
The Web service is running!

Let's see all the architecture...

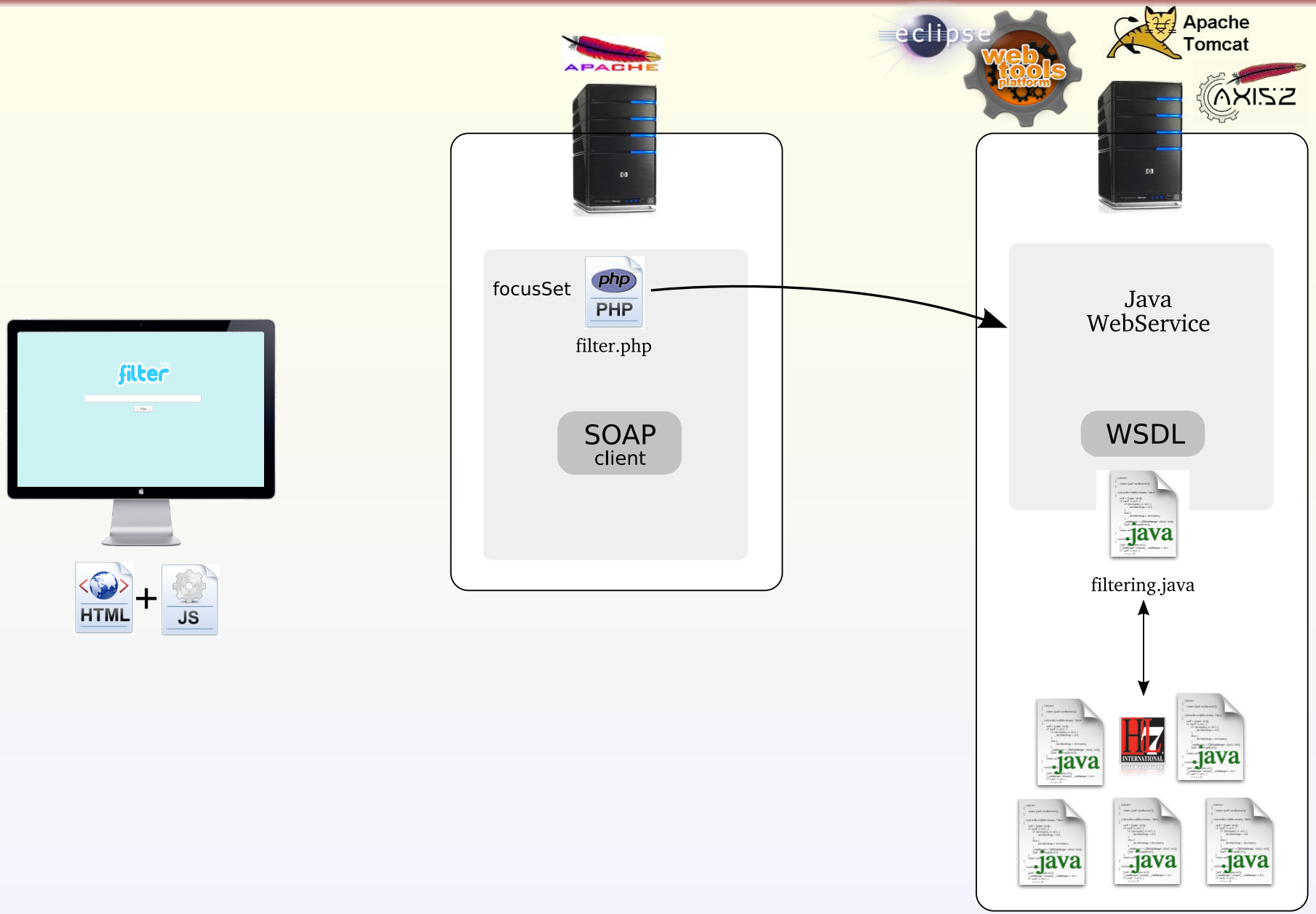
Web service architecture



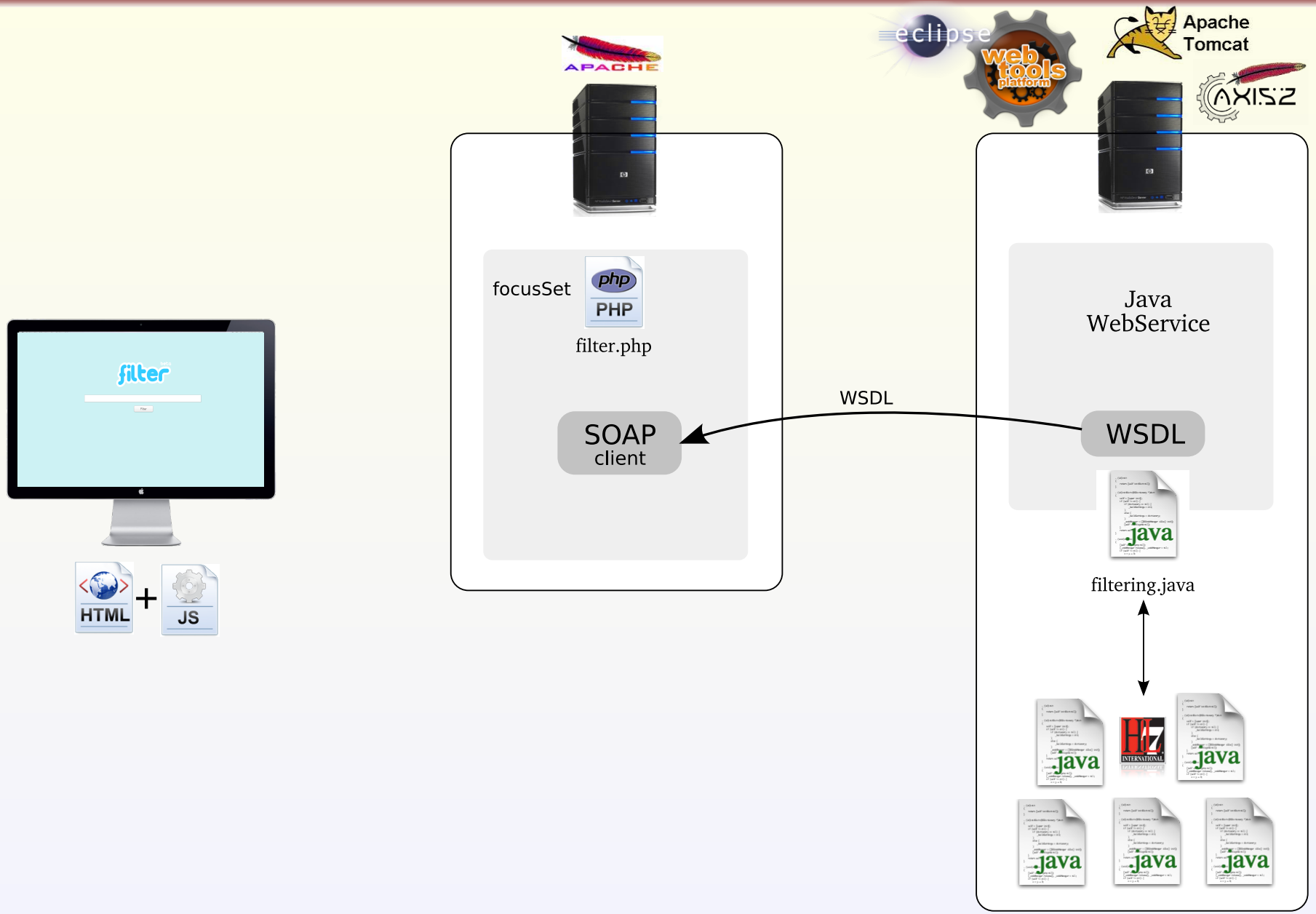
Web service architecture



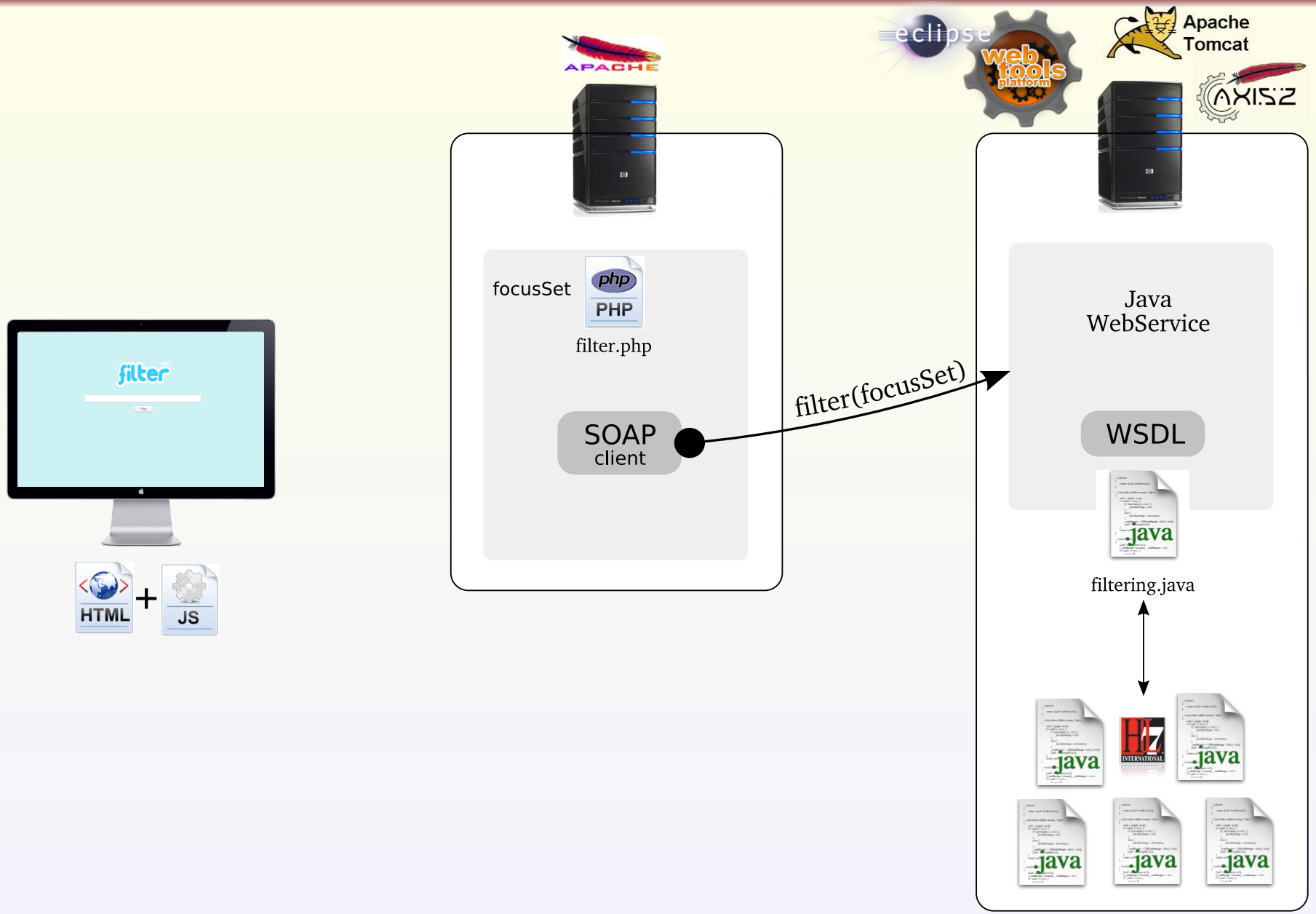
Web service architecture



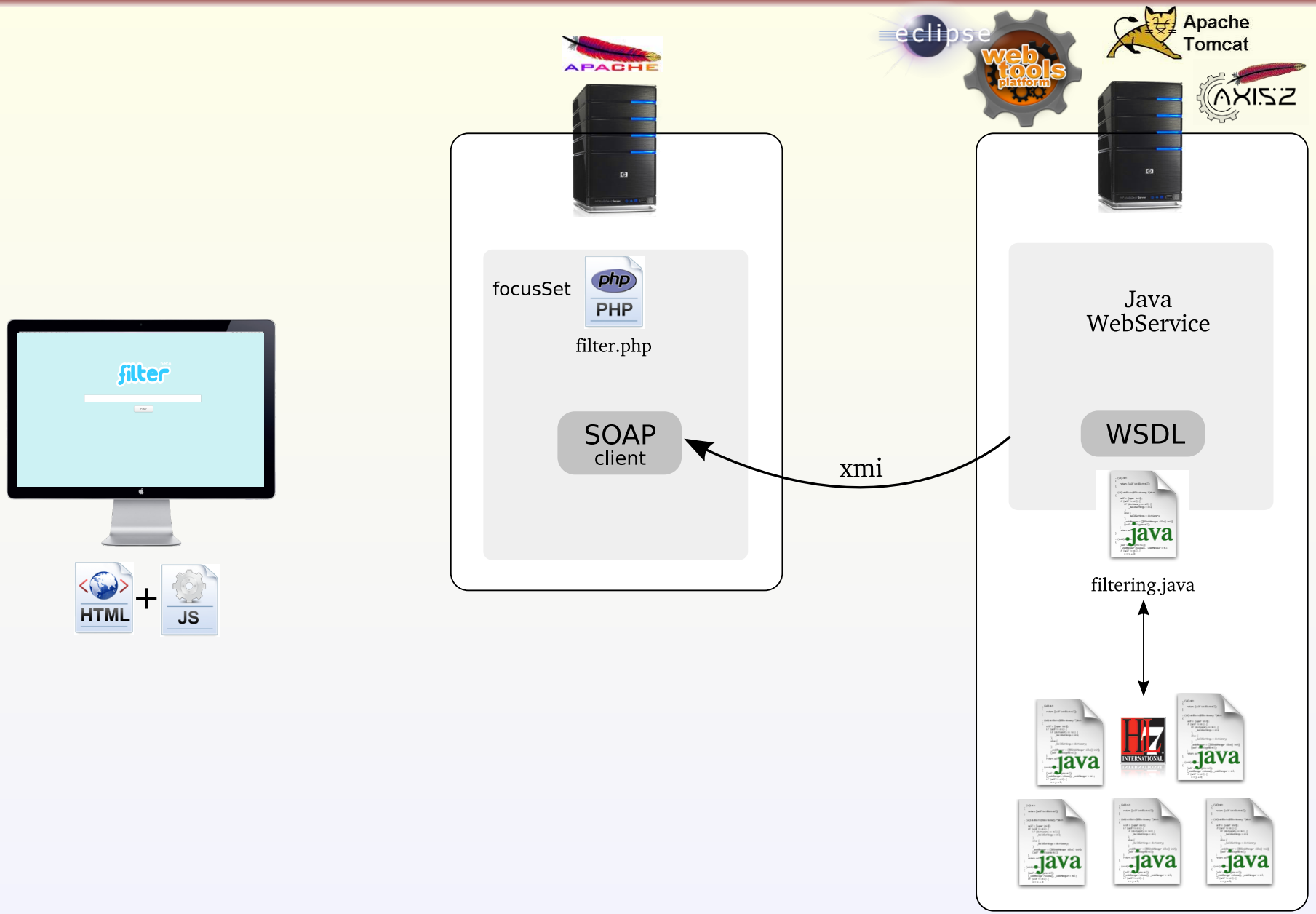
Web service architecture

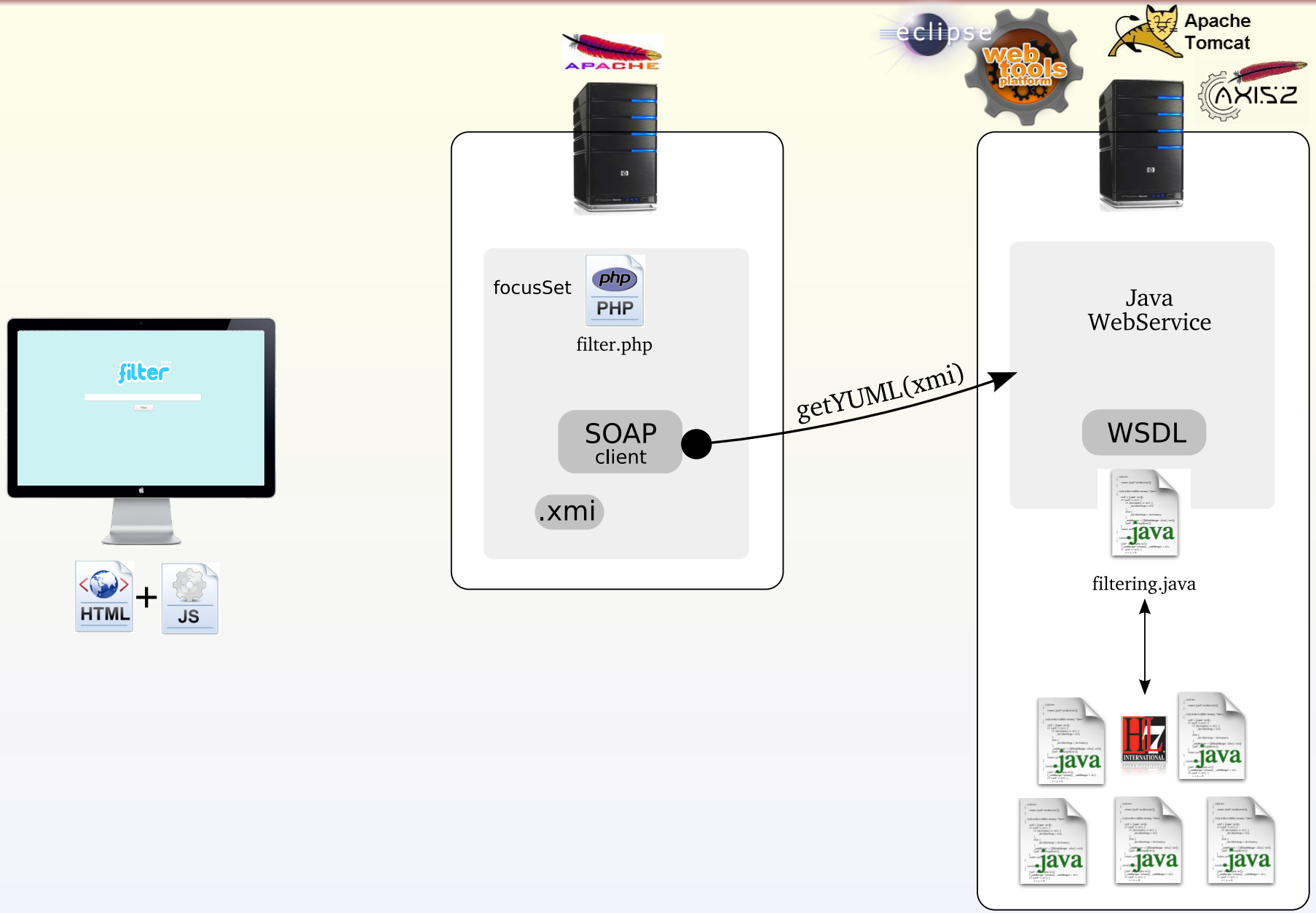


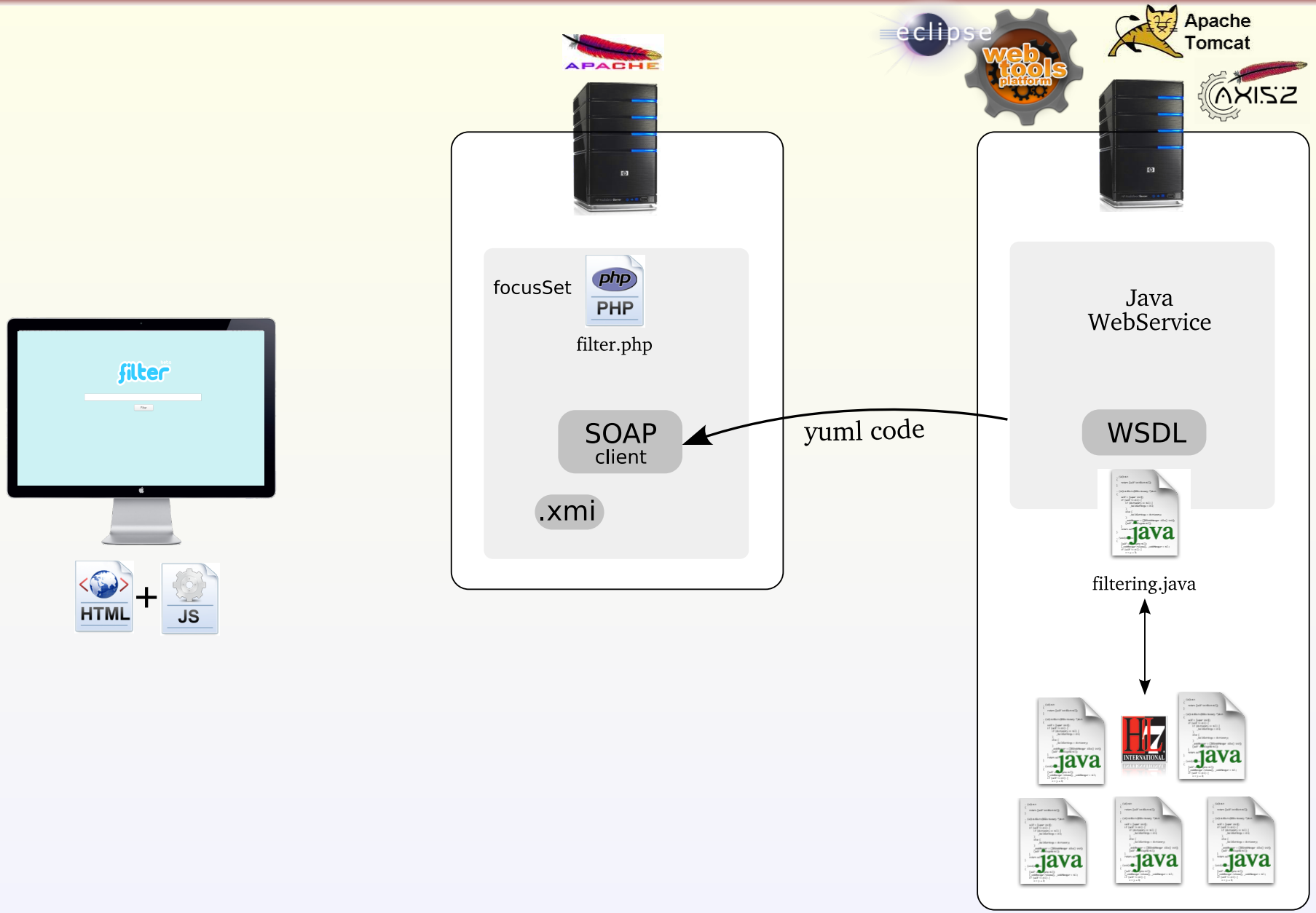
Web service architecture



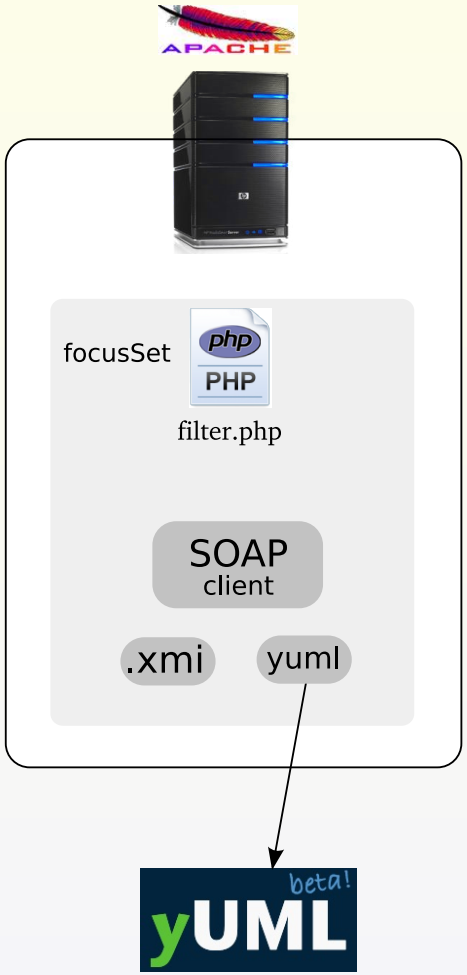
Web service architecture

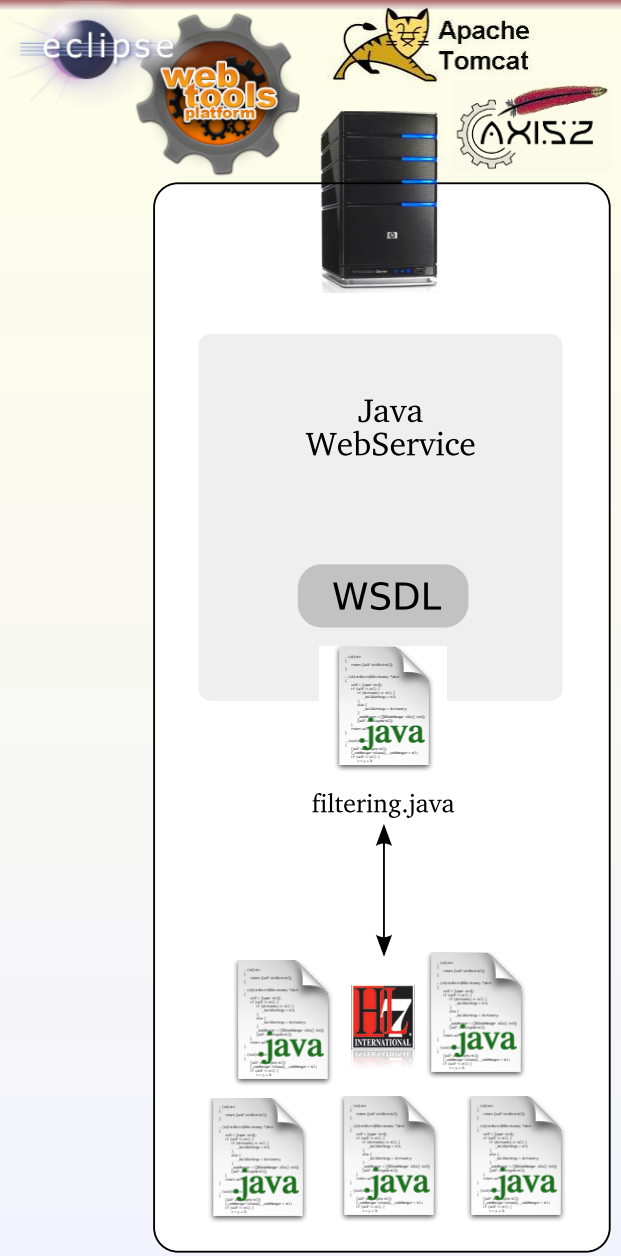



eclipse
web tools platform
Apache Tomcat
AXIS2
APACHE
focusSet
PHP
filter.php
SOAP client
.xmi
getYUML(xmi)
Java WebService
WSDL
filtering.java
HTML
JS

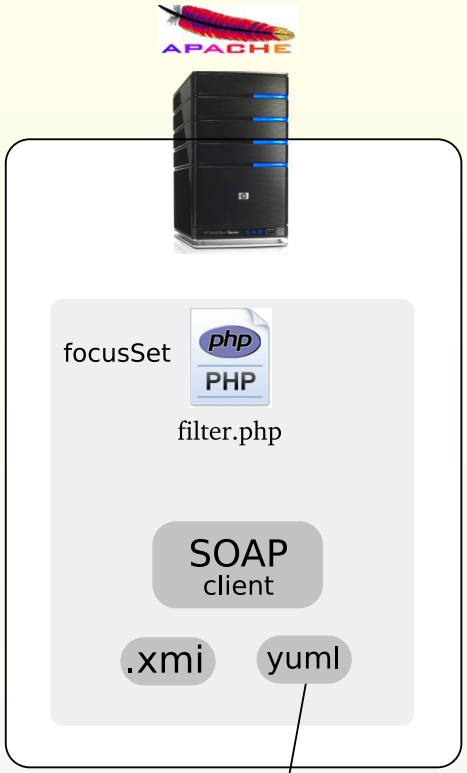

eclipse
web tools platform
Apache Tomcat
AXIS2
APACHE
focusSet
PHP
filter.php
SOAP client
.xmi
yuml code
Java WebService
WSDL
filtering.java
HTML
JS

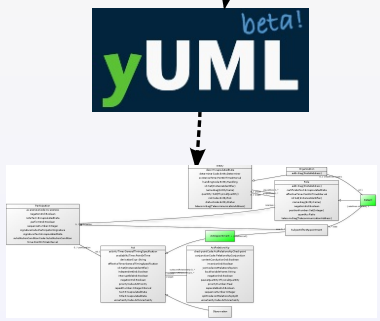

filter
HTML
JS

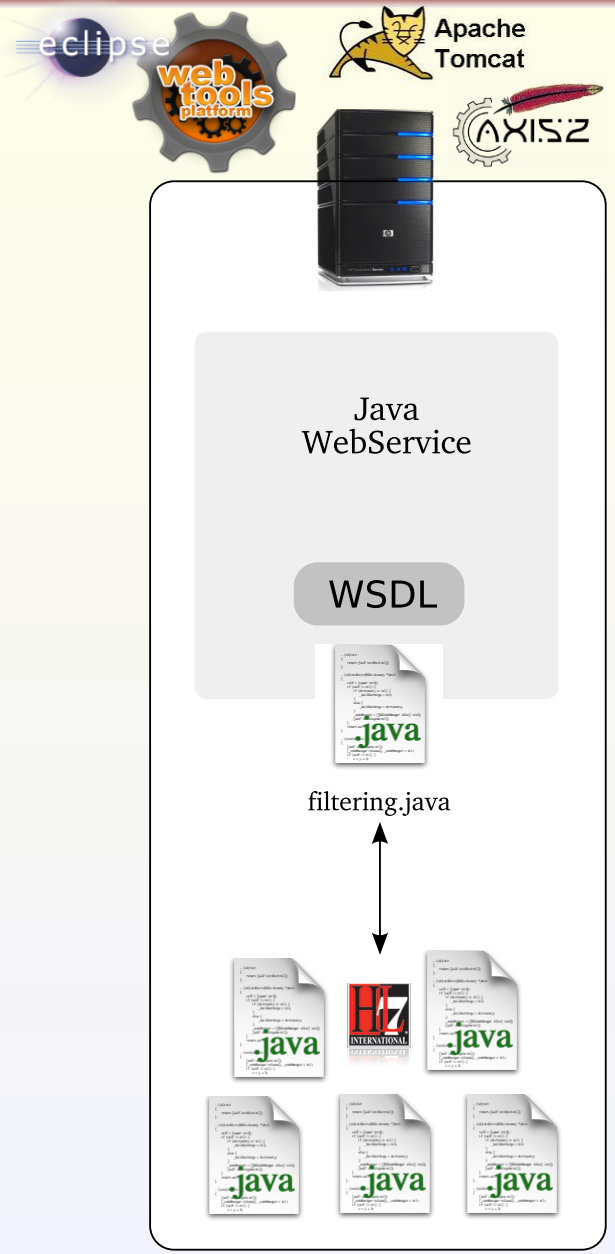

APACHE
focusSet
PHP
filter.php
SOAP
client
.xmi
yuml
beta!
yUML

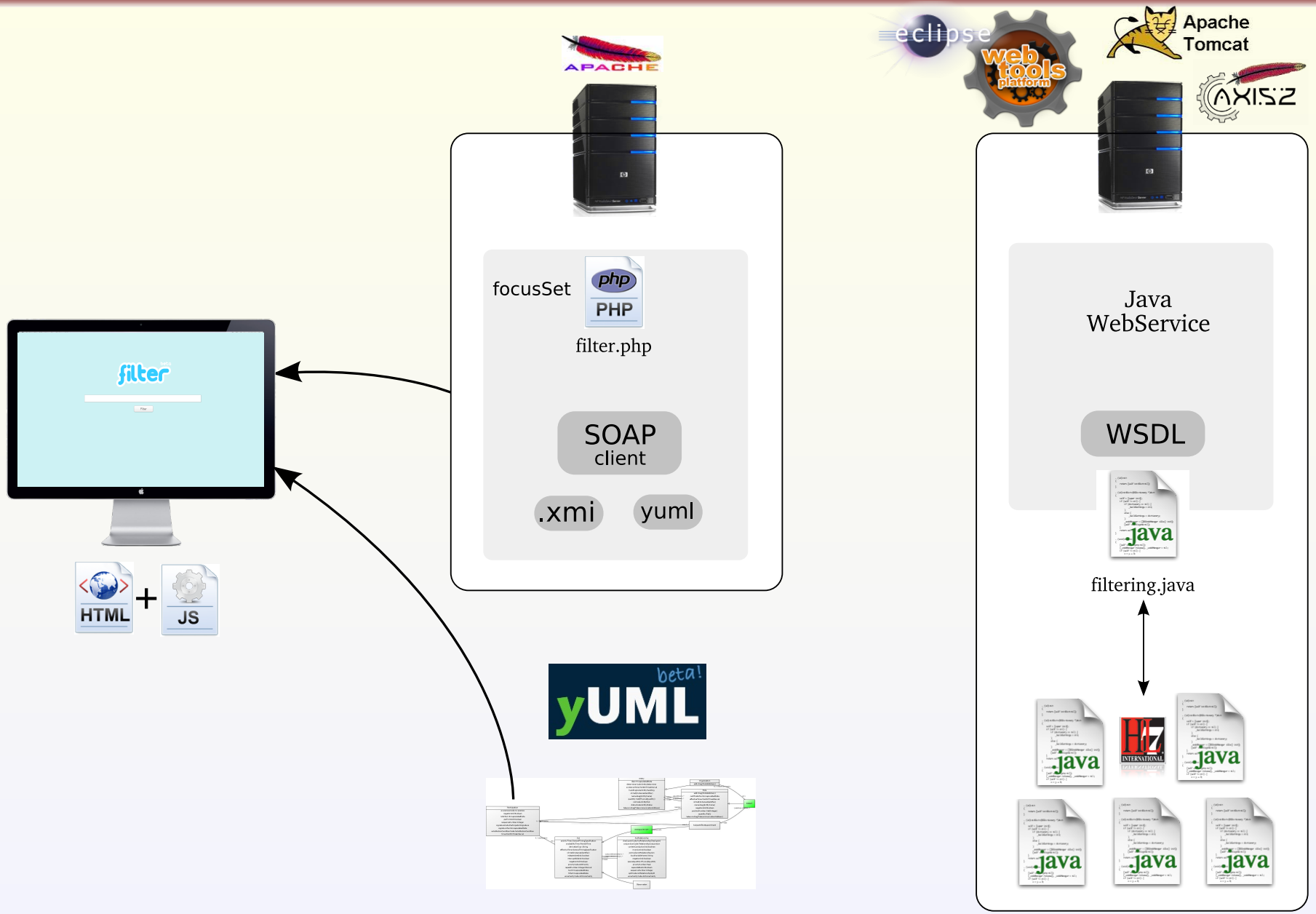

eclipse
web tools platform
Apache Tomcat
AXIS2
Java
WebService
WSDL
java
filtering.java
java
java
java
java

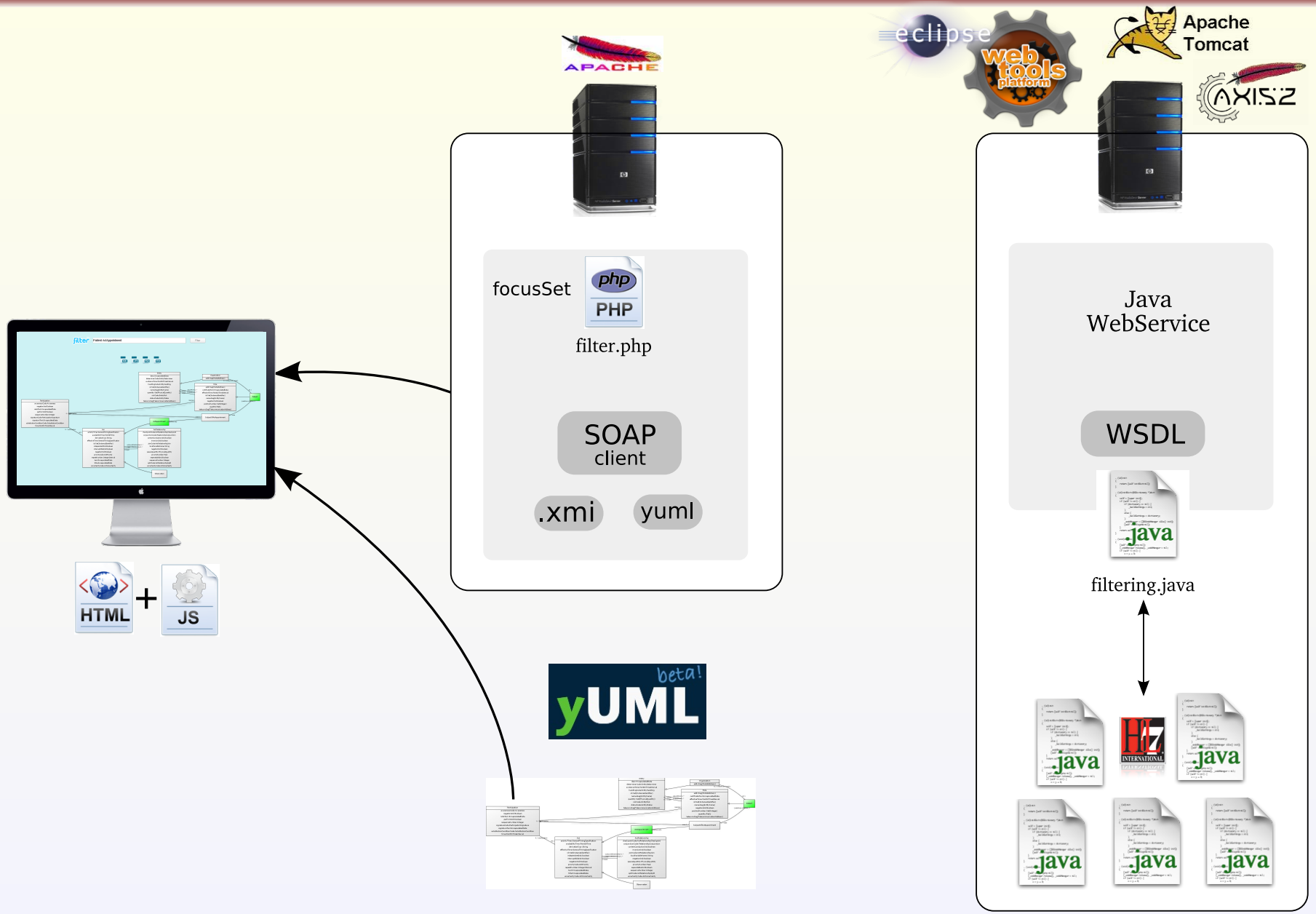

filter
HTML
JS


APACHE
focusSet
PHP
filter.php
SOAP
client
.xmi
yuml


beta!
yUML


eclipse
web tools platform
Apache Tomcat
AXIS2
Java
WebService
WSDL
java
filtering.java


APACHE
eclipse
web tools platform
Apache Tomcat
AXIS2
focusSet
PHP
filter.php
SOAP client
.xmi
yuml
Java WebService
WSDL
filtering.java
filter
HTML
JS
yUML
beta!


APACHE
eclipse
web tools platform
Apache Tomcat
AXIS2
focusSet
PHP
filter.php
SOAP client
.xmi
yuml
Java WebService
WSDL
filtering.java
HTML
JS
yUML
beta!

Demo Time

Conclusions

Integration

Web services are a good solution to **complete functionalities** of your software.

Technologies

Fortunately, they are **well-supported** and it makes their adoption and understandability **less complex** than it could seem at first.

Development cost

It is **easier to port** a desktop app into a Web service than to develop a new web app from scratch.

Conclusions

Integration

Web services are a good solution to **complete functionalities** of your software.

Technologies

Fortunately, they are **well-supported** and it makes their adoption and understandability **less complex** than it could seem at first.

Development cost

It is **easier to port** a desktop app into a Web service than to develop a new web app from scratch.

Conclusions

Integration

Web services are a good solution to **complete functionalities** of your software.

Technologies

Fortunately, they are **well-supported** and it makes their adoption and understandability **less complex** than it could seem at first.

Development cost

It is **easier to port** a desktop app into a Web service than to develop a new web app from scratch.

Conclusions

Integration

Web services are a good solution to **complete functionalities** of your software.

Technologies

Fortunately, they are **well-supported** and it makes their adoption and understandability **less complex** than it could seem at first.

Development cost

It is **easier to port** a desktop app into a Web service than to develop a new web app from scratch.

Credits

References

F. Curbera et al. **Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI**. *IEEE Internet Computing (2002), 86–93.*

E. Cerami. **Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL**. *O'Reilly Media (2002).*

M. Weiss. **Web Services**. *Carleton University.* `http://www.slideshare.net/mrw/web-services-84334`

T. Bellwood. **Understanding UDDI. Tracking the evolving specification**. *IBM developerWorks (2002).* `http://www.ibm.com/developerworks/webservices/library/ws-featuddi/`

T. Harris **yUML. Simple UML diagrams online.** `yuml.me`

Tutorial

L. Vogel. **Webservices with Axis2 and the Eclipse Web Tool Platform (WTP)**. `http://www.vogella.de/articles/Webservice/article.html`

Comics

Geek & Poke. `http://geekandpoke.typepad.com`

Credits

References

F. Curbera et al. **Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI**. *IEEE Internet Computing (2002), 86–93.*

E. Cerami. **Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL**. *O'Reilly Media (2002).*

M. Weiss. **Web Services**. *Carleton University.* `http://www.slideshare.net/mrw/web-services-84334`

T. Bellwood. **Understanding UDDI. Tracking the evolving specification**. *IBM developerWorks (2002).* `http://www.ibm.com/developerworks/webservices/library/ws-featuddi/`

T. Harris **yUML. Simple UML diagrams online.** `yuml.me`

Tutorial

L. Vogel. **Webservices with Axis2 and the Eclipse Web Tool Platform (WTP)**. `http://www.vogella.de/articles/Webservice/article.html`

Comics

Geek & Poke. `http://geekandpoke.typepad.com`