



Introduction to Web Services

Göttingen, 2005-02-28 – 2005-03-02

Content



- What are web services ?
- Why – Web services
- Web services architecture
- Web services stack
- SOAP
- WSDL
- UDDI
- Conclusion

Göttingen, 2005-02-28 – 2005-03-02

Definition



A simple definition:

“a Web Service is an application component accessible over internet”.

Göttingen, 2005-02-28 – 2005-03-02

Why Web services?



- Web services evolved from previous technologies that served the same purpose such as DCOM, CORBA , JAVA RMI.
- Web Services were intended to solve these problems:
 1. Interoperability
 2. Firewall traversal & Proprietary protocols
 3. Complexity

Göttingen, 2005-02-28 – 2005-03-02

Interoperability



- DCOM applications - Windows Operating system.
- RMI - Java programming language.

Göttingen, 2005-02-28 – 2005-03-02

Firewall and Proprietary protocols



- Distributed systems such as CORBA and DCOM used non-standard ports.
- CORBA uses IIOP which is not practical for inter-enterprise integration
- Web Services use HTTP as a transport protocol and most of the firewalls allow access though port 80 (HTTP).

Complexity



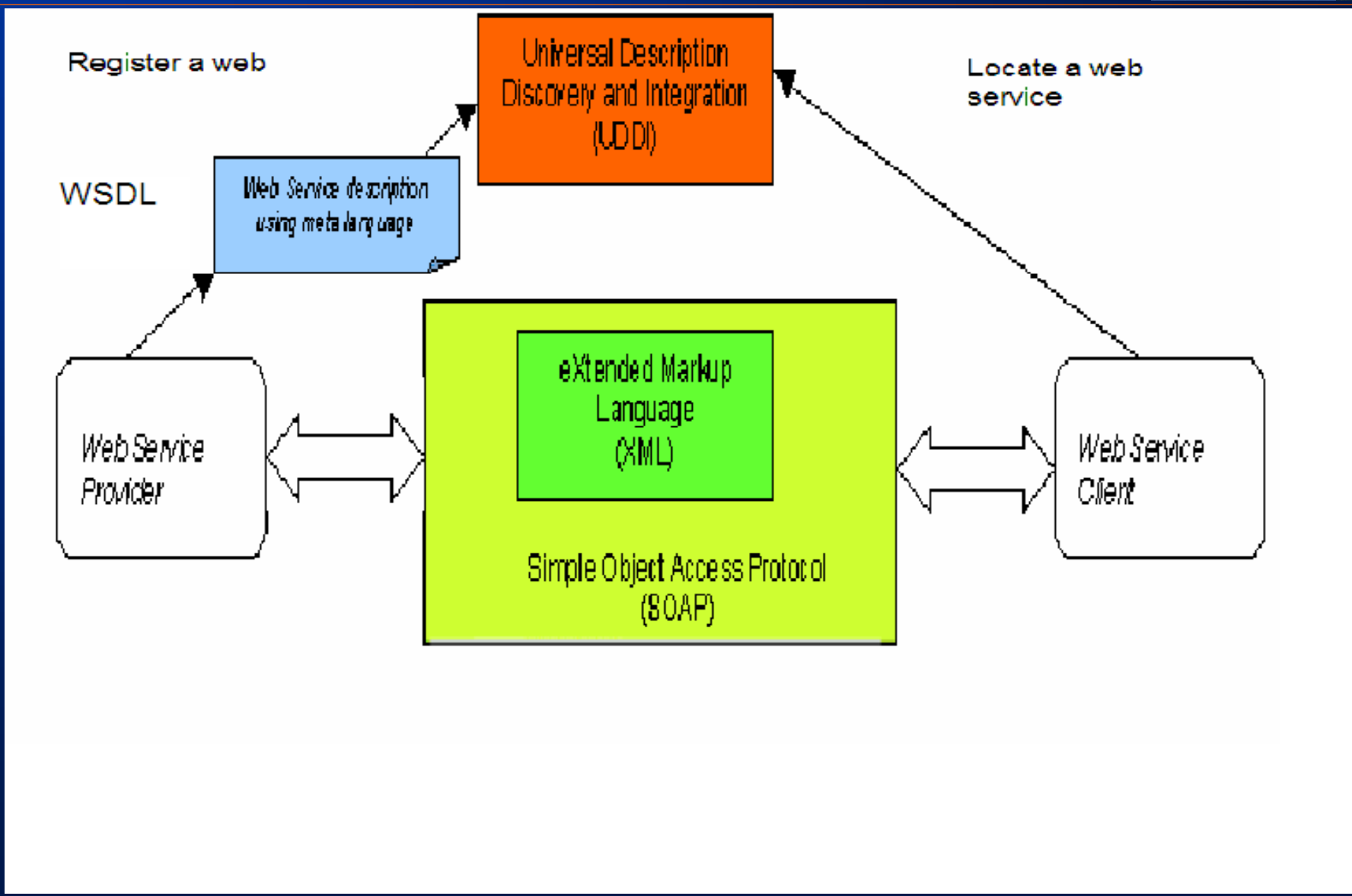
- Web Services are very easy to implement.
- No need to learn new technologies or languages
- We can implement the web service with our existing language and existing operating system.

Göttingen, 2005-02-28 – 2005-03-02



nestor





Steps in web services development



Service provider:

- Define the services that will be provided
- Implement the functionality behind the services
- Deploy the service provider application
- Publish the Web services with a directory service
- Wait for processing client requests

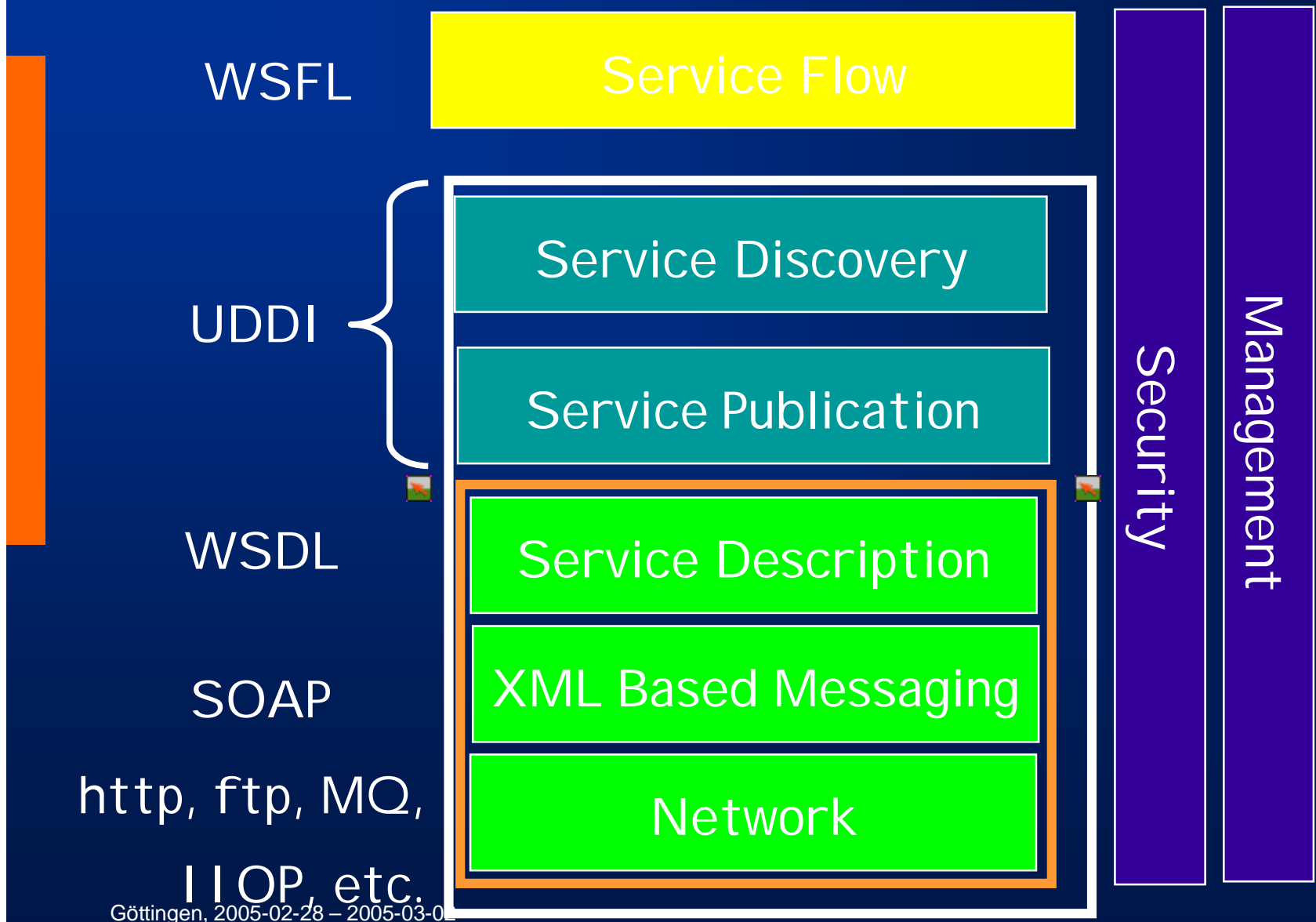
Steps in web services development



Service user:

- Identify the services that will be required
- Locate the Web service by querying a directory service
- Send the request to the service
- Receive the response from the service

Web services stack



Göttingen, 2005-02-28 – 2005-03-01

SOAP



- SOAP originally stood for "Simple Object Access Protocol" .
- An XML envelope for XML messaging,
Header + body.
- Soap is an XML vocabulary standard to enable programs on separate computers to interact across any network.
- Soap is not bound to any particular protocol.

Advantages of SOAP



SOAP has three major characteristics:

- Extensibility – XML message , we can create our own elements.
- Neutrality - SOAP can be used over any transport protocol such as HTTP, SMTP or even TCP.
- Independent - SOAP allows for any programming model .

Structure of SOAP



A SOAP message is an ordinary XML document containing the following elements:

- A required Envelope element that identifies the XML document as a SOAP message.
- An optional Header element that contains header information.
- A required Body element that contains call and response information.
- An optional Fault element that provides information about errors that occurred while processing the message.

Example SOAP request



```
<?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:Body xmlns:m="http://www.weather1.org/wr">
    <m:GetWeather>
      <m:Postalcode>49088</m:Postalcode>
    </m:GetWeather>
  </soap:Body>
</soap:Envelope>
```

Example SOAP response



```
<?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:Body xmlns:m=" http://www.weather1.org/wr">
    <m:GetWeatherResponse>
      <m:Degree>-5</m:Degree>
    </m:GetWeatherResponse>
  </soap:Body>
</soap:Envelope>
```


WSDL example



```
<message name="GetWeatherRequest">
  <part name="Postalcode" type="xs:string"/>
</message>
<message name="GetWeatherResponse">
  <part name="Degree" type="xs:string"/>
</message>

<portType name="Weather">
  <operation name="GetWeather">
    <input message="GetWeatherRequest"/>
    <output message="GetWeatherResponse"/>
  </operation>
</portType>
```

Göttingen, 2005-02-28 – 2005-03-02



UDDI



- Universal Description, Discovery, & Integration
- A project to encourage interoperability and adoption of web services
- Similar to universal phone book
- Partnership among industry & business leaders
 - Initiated by Ariba, IBM, and Microsoft
- UDDI Operators (IBM, Microsoft)

Göttingen, 2005-02-28 – 2005-03-02

UDDI



- Specification to publish and discover web services in the web
- Shared Distributed registry on the web
- Enables three basic functions
 - Publish function (how a web service registers)
 - Find function (how a client finds a Web Service)
 - Bind function (how the client connects and interacts with a Web Service)

Göttingen, 2005-02-28 – 2005-03-02



- Now we come to our new definition to describe the web services as
 - an application component that:
 - Communicates via open protocols (HTTP, SMTP, etc.)
 - Processes XML messages framed using SOAP
 - Describes its messages using XML Schema
 - Describes itself using WSDL
 - Find and bound using UDDI

Summary



- “Web services” is a growing technology
- Brings interoperability at unprecedented level of web applications
- It’s been accepted by broad industries
- Hope it will bring a big revolution in web.

Göttingen, 2005-02-28 – 2005-03-02



References:

- <http://www.w3c.org/TR/soap>
- <http://www.w3c.org/TR/wsdl>
- <http://www.uddi.org/>
- <http://msdn.microsoft.com/webservices/understanding/webservicebasics/default.aspx>

I am : Dhanapalan

Email : dkulanda@uni-osnabrueck.de

Göttingen, 2005-02-28 – 2005-03-02



Thank you for your attention J

Göttingen, 2005-02-28 – 2005-03-02

