

# Component-based Software Development

Lecture No. 9

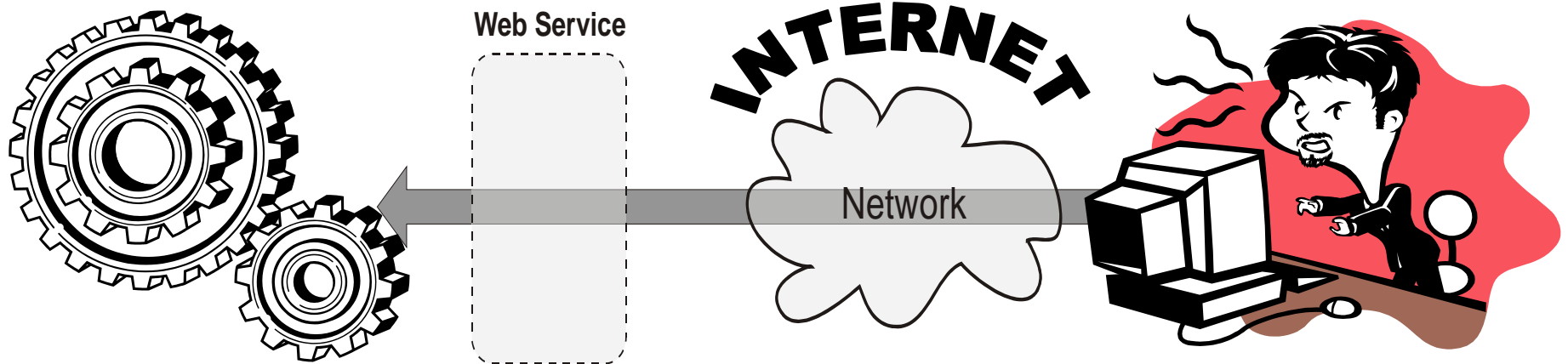
Web Services

# Outline of the Lecture

- Web Services
  - Concepts
  - Basic Architecture
  - Technologies
- SOAP
- WSDL
- UDDI
- Goal of the Lecture
  - Web Services in CBD
- 2<sup>nd</sup> part: discussion; 3rd homework (practical)

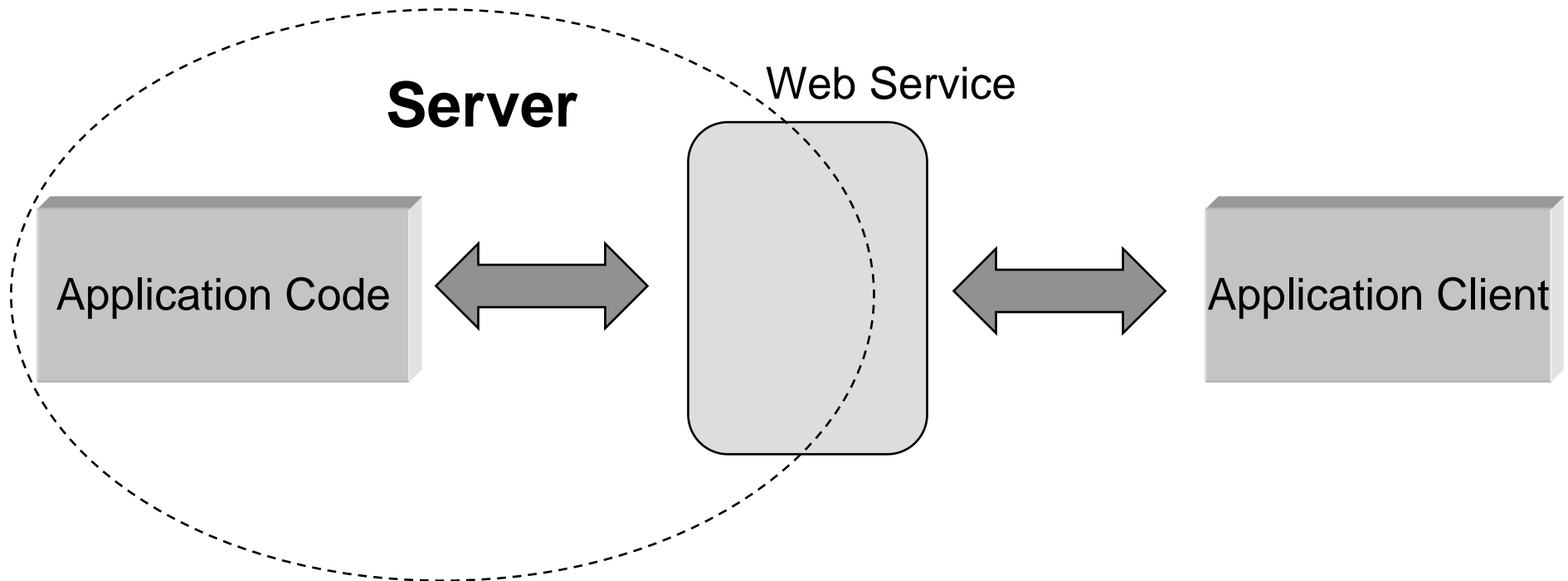
# Web Services

”A *Web Service* is a network accessible interface to application functionality, built using standard Internet Technologies”



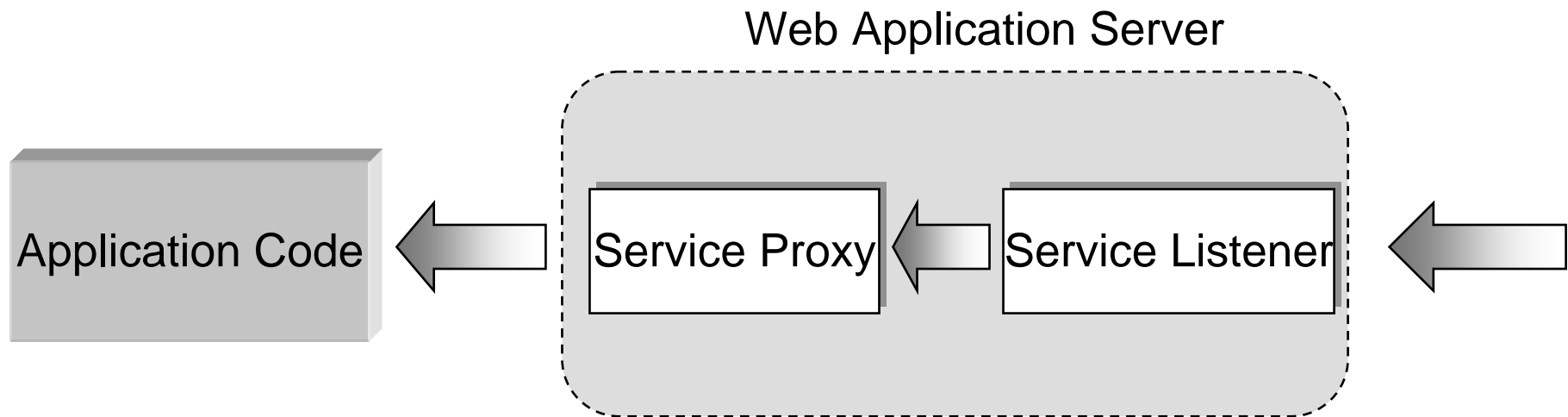
# Web Services

- Basic concept



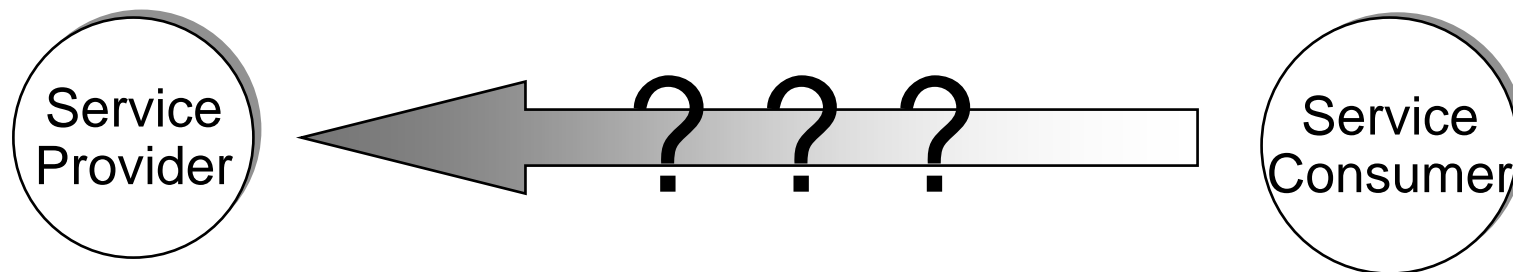
# Components of a Web Service

- **Service Listener:** waits for incoming requests
- **Service Proxy:** evaluates the request and sends it to the appropriate Application



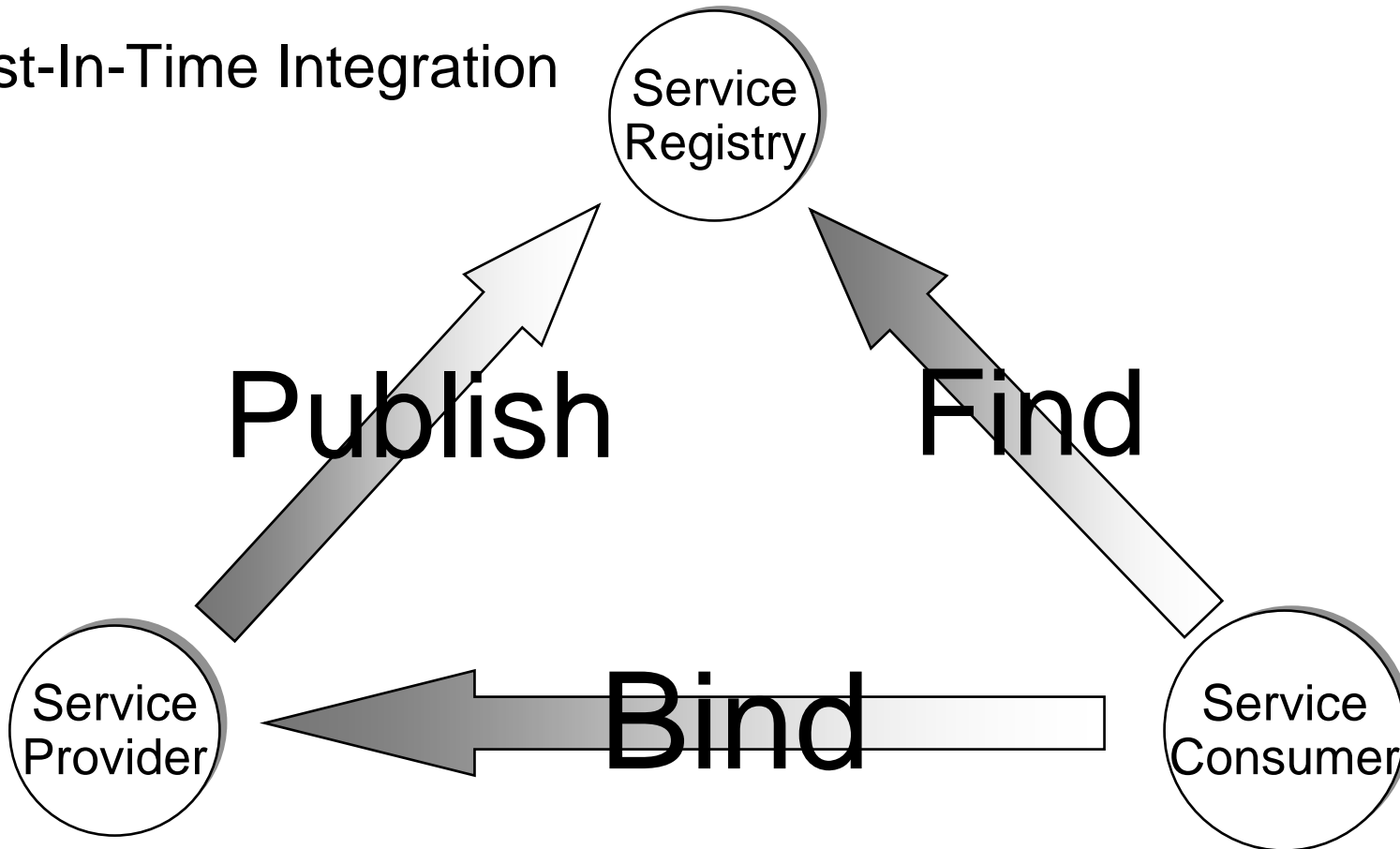
- **Application:** contains the actual business logic

# Web Service Architecture



# Web Service Architecture

- Just-In-Time Integration



# Web Services Technologies

- Network
  - LAN, WAN, 3G, ...
- Transport
  - HTTP, SMTP, TCP/IP, IM, Jabber, ...
- Packaging (messaging)
  - SOAP (XML Protocol), XML-RPC, ...
- Description
  - WSDL, Resource Description Framework (WRDF),  
OWL-S (DARPA Agent Markup Language (DAML)), ...
- Discovery
  - Universal Description and Deployment Interface (UDDI),  
Web Services Inspection Language (WSIL), ...

Discovery

Description

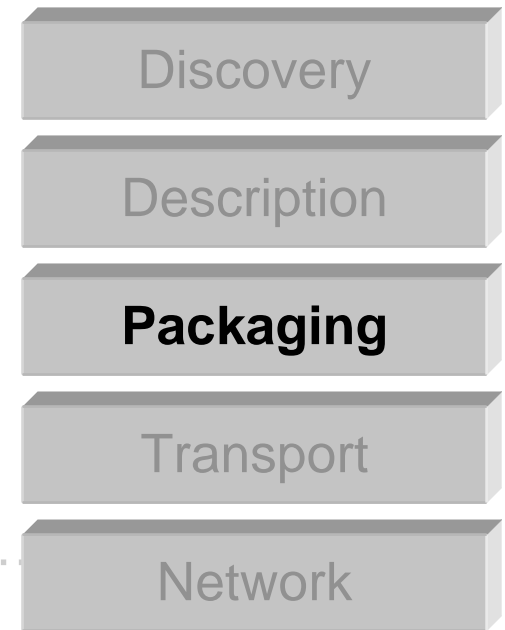
Packaging

Transport

Network

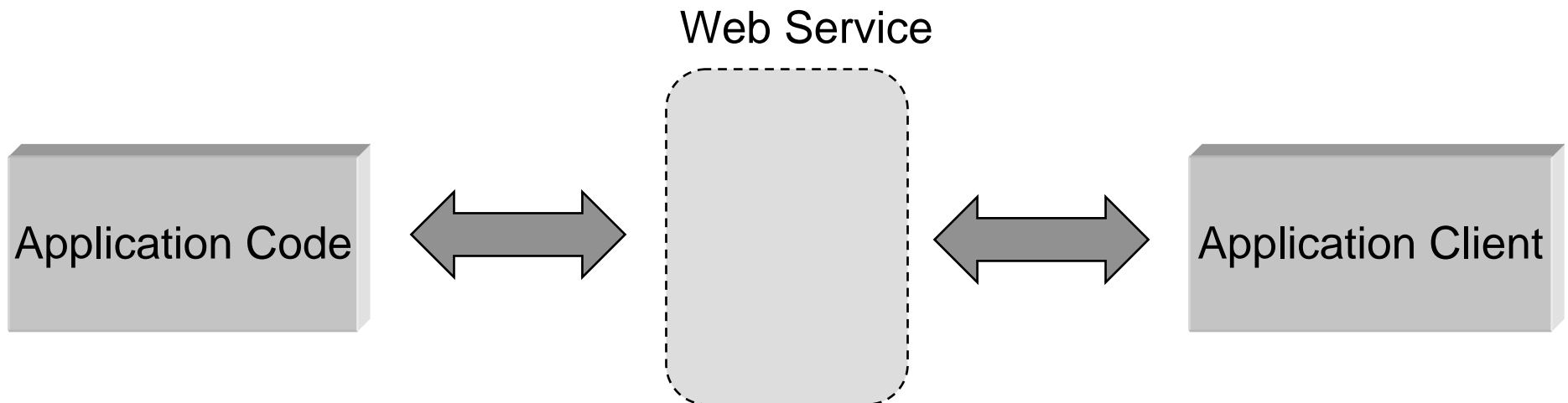
# Simple Object Access Protocol

- Network
  - LAN, WAN, 3G, ...
- Transport
  - HTTP, SMTP, TCP/IP, IM, Jabber, ...
- **Packaging (messaging)**
  - SOAP (XML Protocol), XML-RPC, ...
- Description
  - WSDL, Resource Description Framework (WRDF),  
OWL-S (DARPA Agent Markup Language (DAML)), ...
- Discovery
  - Universal Description and Deployment Interface (UDDI),  
Web Services Inspection Language (WSIL), ...



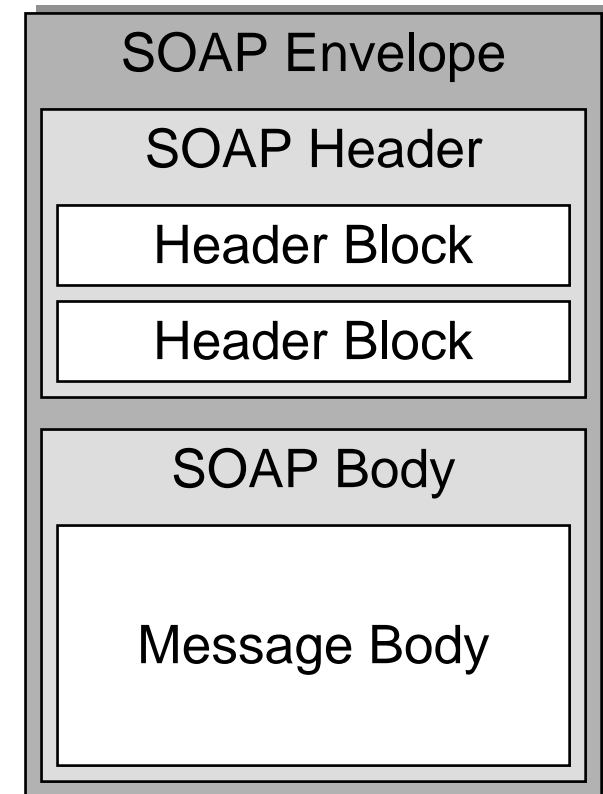
# Simple Object Access Protocol

- Information Exchange using XML Documents
- SOAP Styles
  - Different Body encoding
  - Electronic Document Interchange (EDI)
    - Document literal encoding (ASP.NET)
    - XML Schema to define document format
  - Remote Procedure Call (RPC)
    - RPC literal encoding
    - All parameters in single element



# Simple Object Access Protocol

- SOAP specification (Section 5)
  - <http://www.w3.org/TR/SOAP/>
- SOAP envelope = message
- A SOAP envelope
  - SOAP Header
    - Set of header Blocks
    - Processing, routing, ... instructions
  - SOAP Body
    - Message Body



# Simple Object Access Protocol

- Message Body
  - Differentiation into Accessor and Value (`<firstname>Michael</firstname>`)
    - Structs (Accessor differentiated by Name)
    - Arrays (Differentiated by **ordinal position**)
    - Compound data types
  - Requests / Replies
- Atomic Datatypes specified by “XML Schema data types”
  - `<value xsi:type="xsd:int">36</value>`
  - Anonymus Accessor
    - `<SOAP-ENC:int>36</SOAP-ENC:int>`

# Web Services Description Language

- Network
  - LAN, WAN, 3G, ...
- Transport
  - HTTP, SMTP, TCP/IP, IM, Jabber, ...
- Packaging (messaging)
  - SOAP (XML Protocol), XML-RPC, ...
- **Description**
  - WSDL, Resource Description Framework (WRDF), OWL-S (DARPA Agent Markup Language (DAML)), ...
- Discovery
  - Universal Description and Deployment Interface (UDDI), Web Services Inspection Language (WSIL), ...

Discovery

**Description**

Packaging

Transport

Network

# Web Services Description Language

- SOAP Specification does not address the description
  - De-facto Standard WSDL (W3C)
  - Web Services Inspection Language (MS, IBM)
- A Web Service Description
  - Abstract interface through which a Service Consumer communicates with a Service Provider
  - Details on how a specific web server has implemented that interface
- A Service is a Description of Ports
  - Abstract definition
  - Concrete definition

# Web Services Description Language

- Abstract Definition (Port Type)
  - Interface name
  - Collections of Operations
  - Defines messages
- Concrete Definition (Binding)
  - Transport Protocols
  - Ports
  - Packaging Protocol
- Messages
  - Mappings of Data Values of particular Types (Schema)

# Web Services Description Language

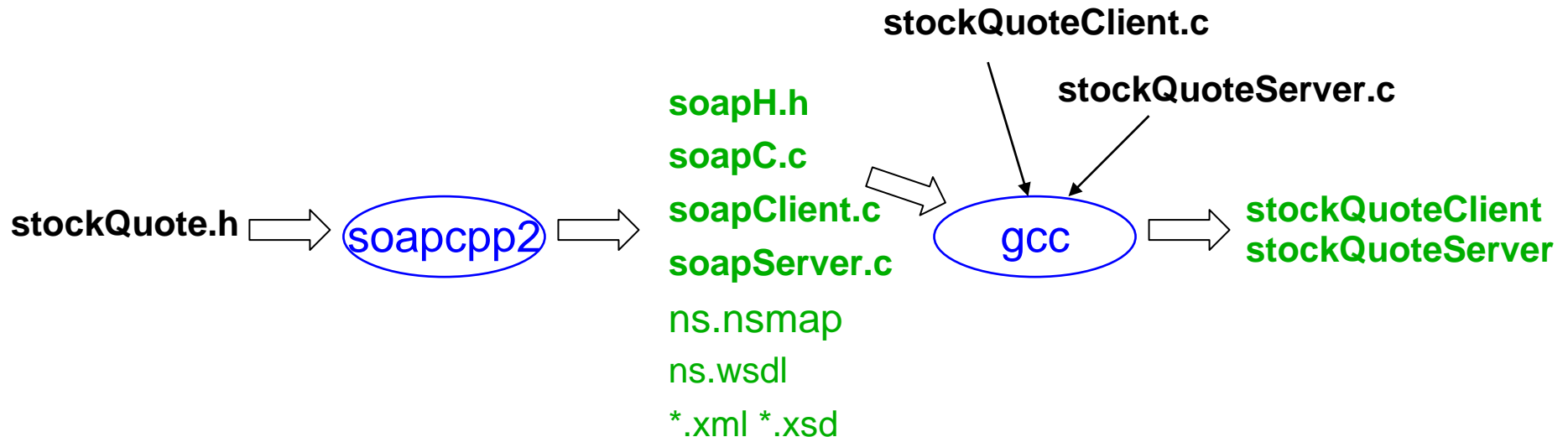
```
<?xml version="1.0"?>
<definitions name="StockQuote" ... >
  <types>
    <schema targetNamespace=http://example.com/stockquote.xsd xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        ...
      </element>
    </schema>
  </types>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input> <soap:body use="literal"/> </input>
      <output> <soap:body use="literal"/> </output>
    </operation>
  </binding>
  <service name="StockQuoteService">
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

# Alternative Web Service Descriptions

- Alternatives to WSDL descriptions
  - Apache Axis: Java2WSDL
    - Java interface translation to WSDL
  - gSOAP: soapstd2 (SOAP stub generator)
    - C/C++ header file translation to WSDL

# A SOAP example with gSOAP

- The stockQuote service



# stockQuote interface

- stockQuote.h

```
//gsoap ns service name:  
stockQuote  
  
struct ns__dateType {  
    long    day;  
    long    month;  
    long    year;  
};  
  
int ns__stockQuote (char *symbol,  
    struct ns__dateType date,  
    double *retValue );
```

- stockQuote.idl

```
module stockQuoteService {  
  
    struct dateType {  
        long    day;  
        long    month;  
        long    year;  
    };  
  
    interface stockQuote {  
        double getQuote(  
            in string symb,  
            in dateType date);  
    };  
};
```

# stockQuote server

```
#include "soapH.h"
#include "stockQuote.nsmap"

int main(int argc, char **argv)
{
    int m, s;                /* master and slave sockets */
    struct soap soap;

    soap_init(&soap);
    m = soap_bind(&soap, NULL, atoi(argv[1]), 100);
    while (1) {
        s = soap_accept(&soap);
        soap_serve(&soap);
        soap_end(&soap);
    }
    return 0;
}

int ns__stockQuote (struct soap *soap, char symbol,
                    struct ns__dateType date, double *retValue )
{
    (*retValue) = 16.83;
    return SOAP_OK;
}
```

# stockQuote client

```
#include "soapH.h"
#include "stockQuote.nsmap"

const char server[] = "localhost:5678";

int main(int argc, char **argv)
{
    struct soap soap;
    double result;
    char symbol[]="NOK";
    struct ns__dateType date;

    date.day = 30;    date.month = 3;    date.year = 2006;
    soap_init(&soap);
    soap_call_ns__stockQuote(&soap, server, "", symbol, date,
        &result);
    printf("result = %g\n", result);
    return 0;
}
```

# Universal Description and Deployment Interface

- Network
  - LAN, WAN, 3G, ...
- Transport
  - HTTP, SMTP, TCP/IP, IM, Jabber, ...
- Packaging (messaging)
  - SOAP (XML Protocol), XML-RPC, ...
- Description
  - WSDL, Resource Description Framework (WRDF), ...  
OWL-S (DARPA Agent Markup Language (DAML))
- Discovery
  - Universal Description and Deployment Interface (UDDI), Web Services Inspection Language (WSIL), ...

**Discovery**

Description

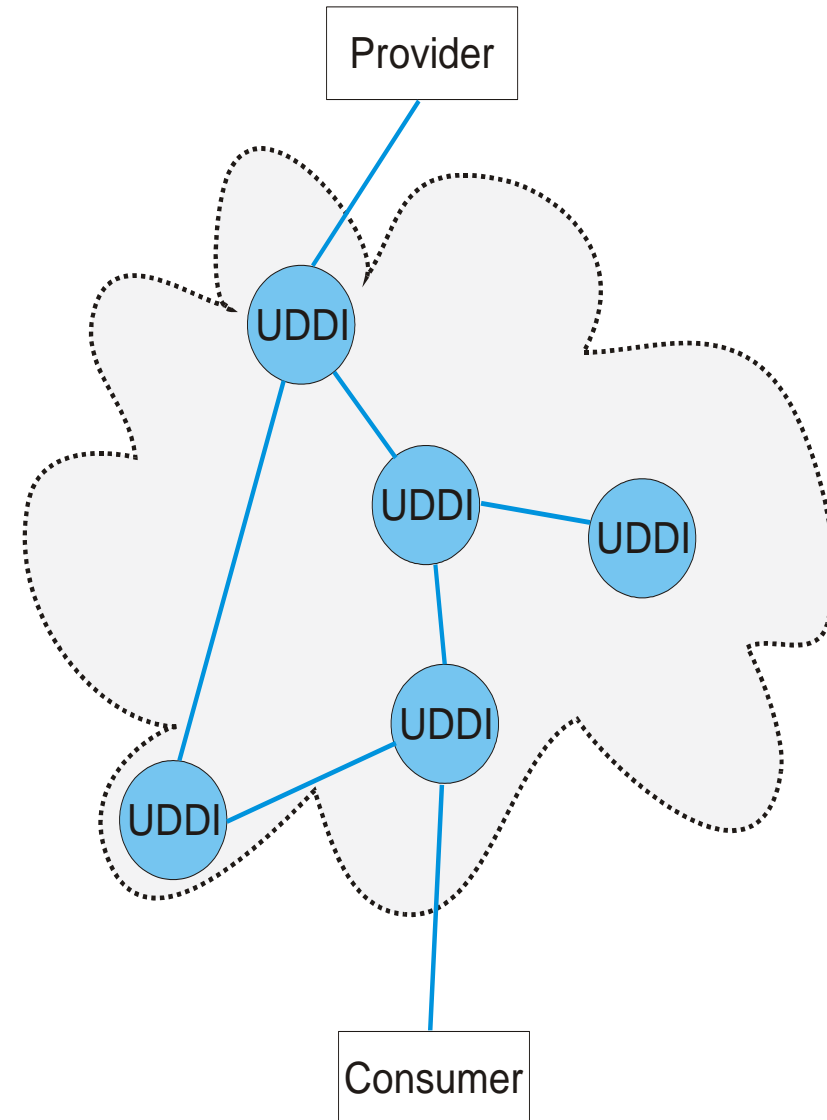
Packaging

Transport

Network

# Universal Description and Deployment Interface

- Automatic Discovery of Services
  - Searchable Registry of Services
  - Description of Services
- Distributed Search Capability
  - Federated Registries
  - Private Registries
- Specific SOAP Message for Registration



# UDDI Interfaces

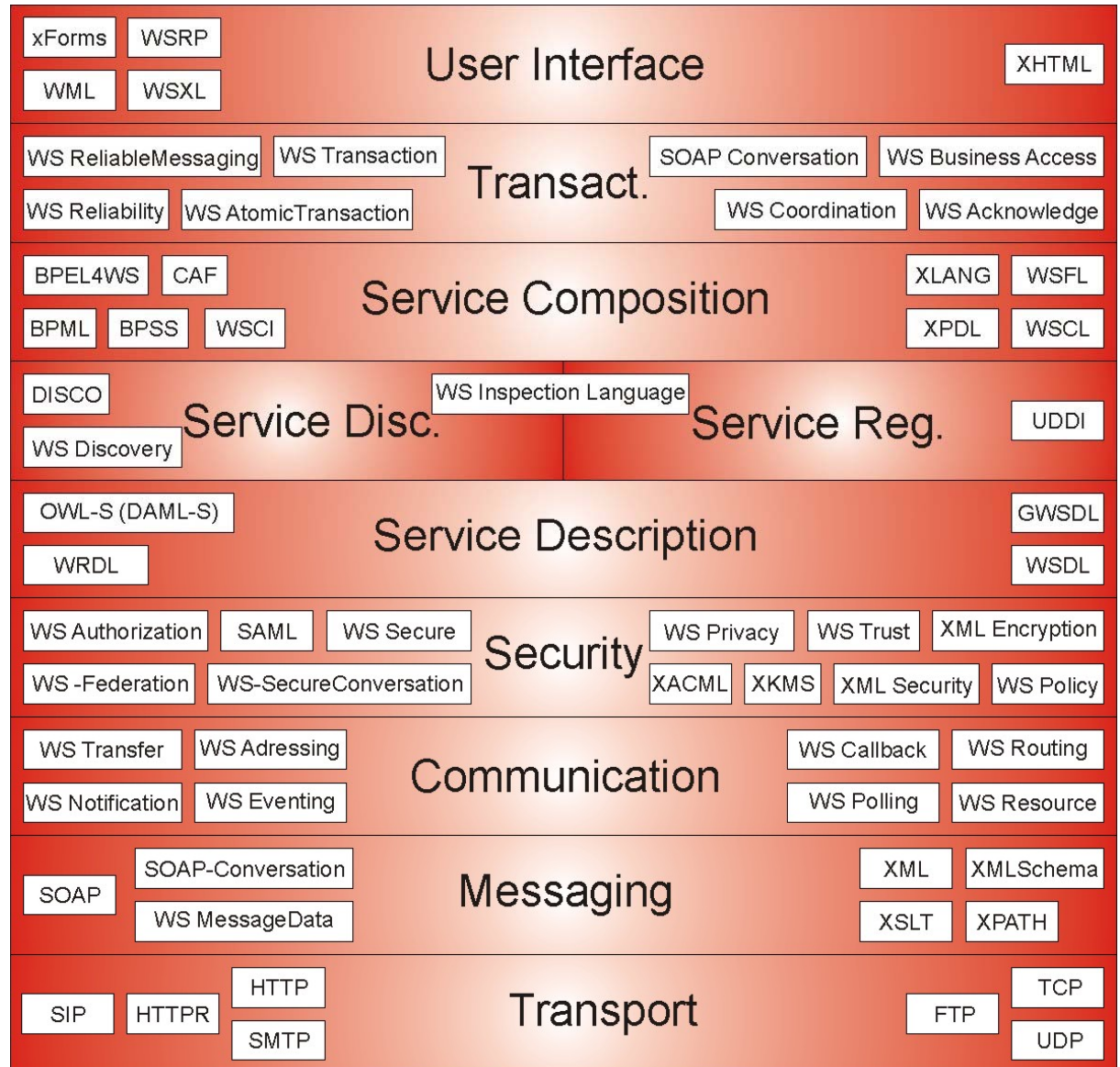
- Publisher Interface
  - Operations for the Service Provider
  - Authentication (Authentication Tokens)
  - Business information
  - Service information
  - Binding information
- Inquiry Interface
  - Operations for Search and Retrieval
  - List Services
  - Bindings
  - Services
  - Businesses
- Public UDDI registries: [uddi.org](http://uddi.org)

# A UDDI Alternative

- Web Services Inspection Language
  - Developed by IBM and Microsoft
  - More simple Description of
    - Simple Pointer to WSDL Document
    - Services URL Endpoint
  - Should be developed and placed on **well known** web server
- Difference
  - UDDI = Phonebook (Yellow Pages)
  - WSIL = Contact Database in YOUR Phone

# Further Web Services Efforts

- Security
  - Security Assertions Markup Language (SAML)
- QoS
- Sequencing of operations
- ...



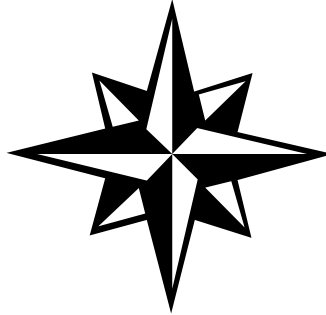
# Further Studies

- J. Snell, D. Tiwell, P. Kulchenko: “Programming Web Services with SOAP”
- <http://www.soapware.org>
- Web Services Toolkits
  - SUN: Java Webservices Developer Pack (2.0)
  - IBM: Alphaworks
  - Apache: Axis
  - Microsoft .NET developer center
  - Oracle: Web Services Technology Center
  - Mobile Web Services (MIDP, Symbian)
  - SOAP::Lite for Pearl
    - <http://www.soaplite.com>
- Selected Reading
  - R. Smith, B. Meyer, C. Szyperski, G. Pour: “Component-Based Development? Refining the Blueprint”

# Key points

- Web Services
  - Simple Object Access Protocol
    - Packaging / Messaging
    - Binding
  - Web Services Description Language
    - De-Facto Description mechanism
    - Publish / Find
  - Universal Discovery and Deployment Interface
    - Discovery, Search
    - Publish / Find
  - Many alternatives and additional, related efforts
    - WSIL
    - Security, Composition, ...

# Questions?



# Next Time

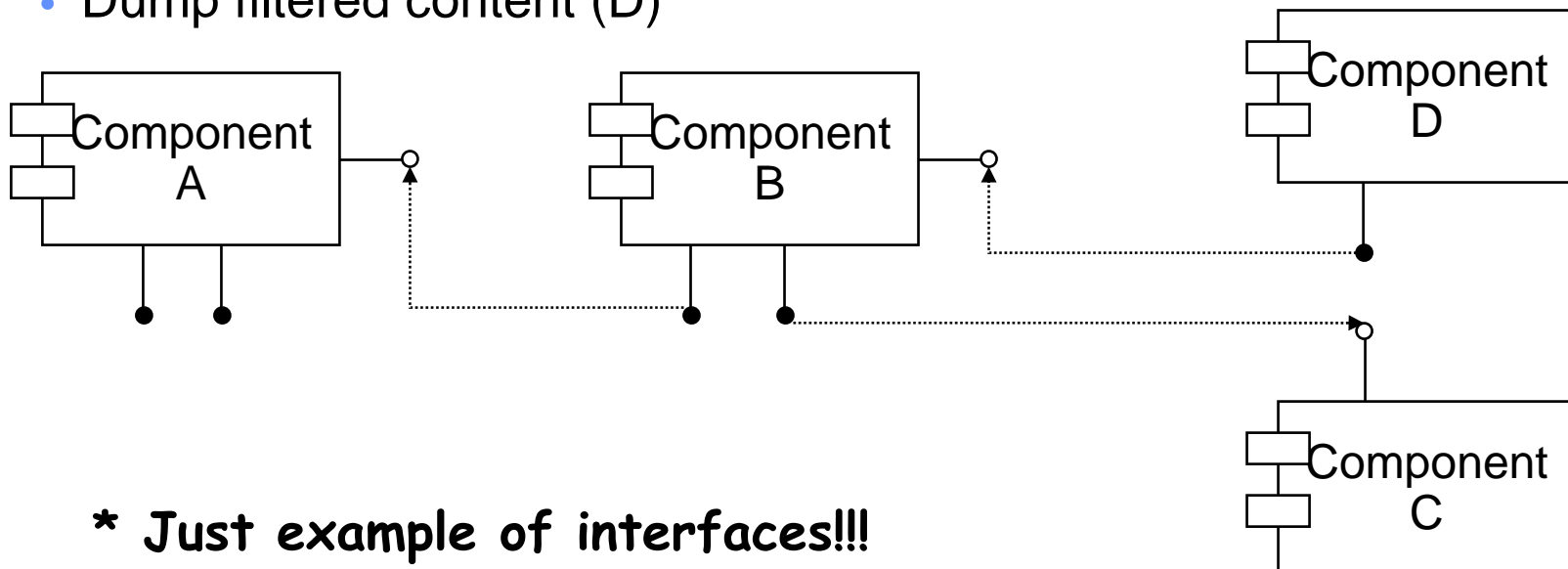
- 06.04.2006
  - The Catalysis Approach

# 3<sup>rd</sup> Homework (practical part)

- 10 people
  - 5 groups to two people
- Problems in 2<sup>nd</sup> Homework
  - Communication
  - <https://moodle.cs.helsinki.fi>
  - More details
  - Common structures

# 3<sup>rd</sup> Homework (practical part)

- Design one component of a system that, based on user input, allows to
  - Accept different sources in different types (A)
  - Filter content (B)
  - Learn different filter (C)
  - Dump filtered content (D)



**\* Just example of interfaces!!!**

# 3<sup>rd</sup> Homework (practical part)

- Retriever Component (previously component A)
  - Retrieve different types of content
  - Common functionality
  - Common external interfaces (e.g. data formats)
  - Meeting date?
- Specific Components
  - xhtml-4, PDF (to text?), Binaries (zip, tarballs, ...), pictures, ...
  - Specific interfaces
- EJB (department)
- Working EJB (sources + documentation)
- Simple test client

# Discussion

- Input: URI
- Output
  - Metadata interface
    - Type (MIME type), Size, modification / access date
  - Content interface
    - XML (according to schema; MIME type, modification dates, )
  - Interrogation interface
  - ?Event mechanism?
  
- Friday, 31.03.2006, 12:00 – 13:30
- Room: tbd (email this evening)
- Send me email if you're interested in the practical part
  - Group you're interested in

# Questions?

- EJB 3 in our jboss install?

