02267: Software Development of Web Services

Week 1

Hubert Baumeister

huba@dtu.dk

Department of Applied Mathematics and Computer Science
Technical University of Denmark

Fall 2015

Contents

What are Web services?

Service Oriented Architecture

Demo

Learning Objectives

Practical Information

What is this Course About?

- Web Services & Service Oriented Architecture
- 5 ECTS points
- Complementary courses
  - 02221/02222 on distributed systems

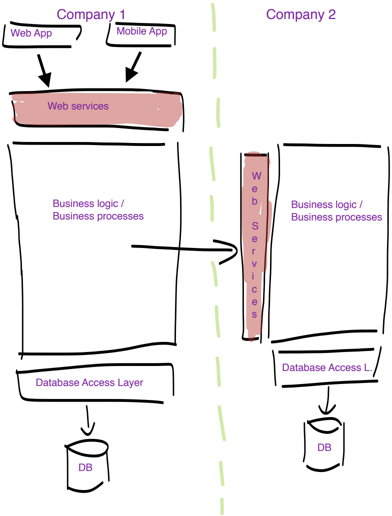
What are Web services?

- A technique such that one computing device offers services to another computing device using standard internet protocols (i.e. HTTP, SOAP, XML, . . .)
- Not to be confused with Web sites/Web applications, though they may use Web services

Uses of Web services

- Provides business logic of Web applications
  - Google, Twitter, . . . provide public Web services
- Provide the connection to the server for mobile applications
  - Request and store data from the server
  - Request computations (like route calculations, image manipulations, . . .)
- Offers business services and automates buisness processes
  - Within a company / across companies (Business to Business B2B)

Web service architecture I



Example: Purchase Order

The customer wants to purchase some goods (via the Internet)

1. Customer contacts the supplier and orders the goods
2. Sales department check with credit card department if credit is okay
3. Sales department check with inventory department if the goods are on stock
4. Sales department informs the billing department to bill the customer
5. Sales department informs the shipment department to send out the goods
6. Shipment department sends the goods to customer
7. Shipment department informs the billing department to send the invoice
8. Billing department sends the invoice to the customer

Example: Purchase Order II

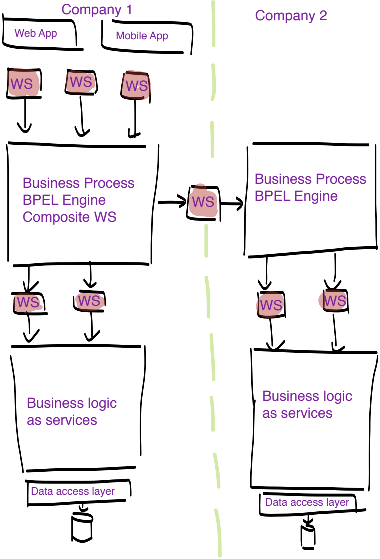
The customer wants to purchase some goods (via the Internet)

1. Customer contacts the supplier’s Web site and orders the goods
2. The Web site uses the process sales service to actually order the goods
3. The process sales service contacts the credit card service to check the customer’s credit
4. The process sales service calls the check inventory service to check if the goods are on stock
    - …

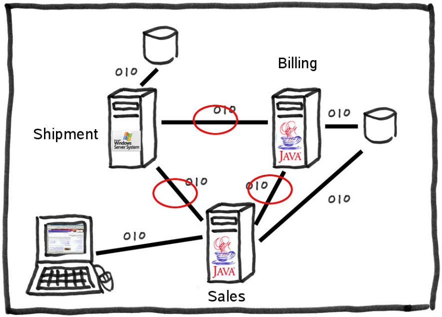
Example: Purchase Order III

- The IT systems of each department offer services
  - → simple (Web) services
- The purchase order business process is itself a service offered to the customer which uses the services of the other IT systems
  - → composite (Web) service

Web service architecture II



Why Web services?



- Problems
  - Different technologies CORBA, EJB, .NET, COM/DCOM
  - No well defined interfaces
  - Ad-hoc combination

Web services to the rescue

- They reuse existing infrastructure for Web applications
  - → Web servers
  - → HTTP, CGI-bin, Servlet architecture, . . .
- Exchange format is simple text messages (SOAP, JSON , . . .)
  - Don’t return HTML but SOAP, XML, JSON, . . .
  - → Easy to parse and construct (reuse XML parsers)
- Crosses company boundaries easy: port 80 with standard Web servers

Success of Web services …

- SOAP is based on XML
- SOAP messages have a simple structure:

```
<soapenv:Envelope>
  <soapenv:Body>
    <ns1:srrequest>
      <ns1:action>register</ns1:action>
      <ns1:student>
         <ns1:name>Søren Helmersen</ns1:name>
      </ns1:student>
    </ns1:srrequest>
  </soapenv:Body>
</soapenv:Envelope>
```

- SOAP messages are transported by HTTP or SMTP

. . . Consequences . . .

→ It is easy to generate SOAP messages

→ Only an XML parser is needed for parsing the messages

→ Web server technology can be reused for Web services (e.g. cgi-bins and servlets)

→ Easy to provide Web services in any programming language

→ Easy to use Web services in any programming language and operating system

. . . But

- Message contents / data encoding is not standardised
  - WSDL and XML Schema help with that
- Simple messaging model: Support needed, e.g., for
  - Security
  - Meta information (Authorization, Authentification, Dialogs, . . .)
  - . . .

Types of Web services

- SOAP-based Web services
  - Use HTTP protocol to exchanged SOAP messages (special type of XML)
  - SOAP messages, however, are independent of HTTP (one possible transport protocol)
  - Based on the concept of services as functions
  - → Used with B2B applications
- RESTfull
  - Use the concepts behind HTTP
    - → Resources, URL's identifying resources
    - → Representations defined by Mime-Types: XML, JSON, Text, HTML, . . .
    - → HTTP Verbs: GET, POST, PUT, and DELETE are functions on resources
  - → Used with mobile applications and Web applications

Contents

What are Web services?

Service Oriented Architecture

Demo

Learning Objectives

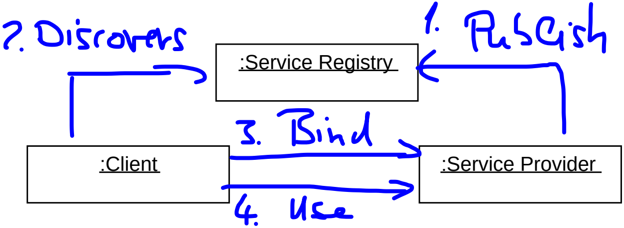
Practical Information

Service-Oriented Architecture (SOA)

- A set of principles for organising the software
- Not restricted to the use of Web services
  - Web services
  - OSGi services
  - Grid services
  - Cloud services
  - . . .

SOA Principles (I)

- Loose coupling
  - Services represent self contained units of logic (one function or a set of functions) which are relatively independent
  - → Resusability
- Discoverablity



SOA Principles (II)

- Abstract service description (independent of implementation)
- Encapsulation (autonomy and abstraction)
- Compositionality
- And additional for Web services
  - Based on open standards
  - Vendor neutral / vendor diversity

Service invocation vs. function call

Function call

- Within the same process
- Function is always available
- Takes almost no time
- Focus on single calls

→ Fine grained

→ Tight coupling

→ Simple data as parameters

Service invocation

- Across processes, computers, networks
- Takes time (several magnitudes more than a function call)
- May fail
- Several service invocation may form a dialog

→ Coarse grained

→ Loosely coupled

→ Complex data, documents, as parameters

Contents

What are Web services?

Service Oriented Architecture

Demo

Learning Objectives

Practical Information

Demo: Order Service Web Service

1. Creating a simple OrderService Web service in Java
2. Deploying the Web service on a Web server (GlassFish)
3. Calling the Web service from Java
4. Calling the Web service from Ruby

Contents

What are Web services?

Service Oriented Architecture

Demo

Learning Objectives

Practical Information

Learning objectives

- Creating and calling simple SOAP-based Web services
  - Bottom Up: Service class first
  - Top Down: Service description in the Web Service Description Language (WSDL) first
  - Using XMLSchema to represent complex datatypes
- Automating Business Processes as Web services
  - Composition and coordination of Web services
  - Business Process Execution Language (BPEL)
- Addressing specific problems of SOAP-based Web services
  - e.g. Reliability of message exchange, security (privacy and authenticity), . . .
- Implementing and calling RESTful Web services
  - Simple Web services
  - How to model business processes with RESTful services
  - What is the difference between SOAP-based and RESTful Web services?
- More detailed on

  `http://www.compute.dtu.dk/courses/02267`

Contents

What are Web services?

Service Oriented Architecture

Demo

Learning Objectives

Practical Information

Course Prerequisites

- Programming language: Java
- Operating system knowledge
  - Examples and exercises use Unix (Linux / FreeBSD), Mac, or Windows
  - Software needs to be installed
  - Shell commands; Shell-scripts need to be written or adapted
- Basic knowledge of Internet technologies: XML, HTML, Sockets, TCP, ...

Practical Information I

- Workload: 5ECTS = 9 hours / week
  - 105 min lecture
  - 105 min tutoring sessions
  - 5 hours self study (homework)
  - Tutoring sessions are used to check your homework and to get help if you have problems with them
- Lectures 10:15 – 12:00 on Monday
- Tutoring sessions 8:15 – 10:00 in E-Databar: Rooms 003, 015, and 019 in Building 341
- Homework is not mandatory
  - but help prepare for the exam project
  - test your understanding of the topic
  - if you run into problems send an e-mail to `huba@dtu.dk` or the TA's

Practical Information II

- How to reach me
  - E-Mail: `huba@dtu.dk`
  - Room: 303B/058
- Course Web Page
  - `http://www.imm.dtu.dk/courses/02267`
- Software
  - Java 7/8 (JDK), OpenESB 3 (Netbeans (IDE) plus Glassfish 4 (Application Server))
  - `http://www.open-esb.org` or from CampusNet (see course Web page)

Grading

1. Project work
   - 5 weeks (week 44—49)
   - Teams of 4—6 (Team building week 44 (participation is mandatory))
   - Implementing Web services (simple and composite / SOAP and RESTful)
   - Writing a report
2. Project presentation by the project teams
   - Project presentation ($\approx$ 10min) + questions: total 45 min
   - Dates week 51: Tu 15 — Fr 18.12

Literature

- Michael P. Papazoglou, Web services: Principles and Technolgoy, Pearson Education Limited, second edition, 2012
- plus additional literature throughout the course