

# Semantic Web Technologies II

## SS 2008

09.06.2008

## Semantic Web Services

**Dr. Peter Haase**

**PD Dr. Pascal Hitzler**

**Dr. Steffen Lamparter**

**Denny Vrandečić**

# Agenda

- Web Services – Foundations
- Semantic Web and Web Services
- Semantic Mash-ups
- Semantic Management of Web Services

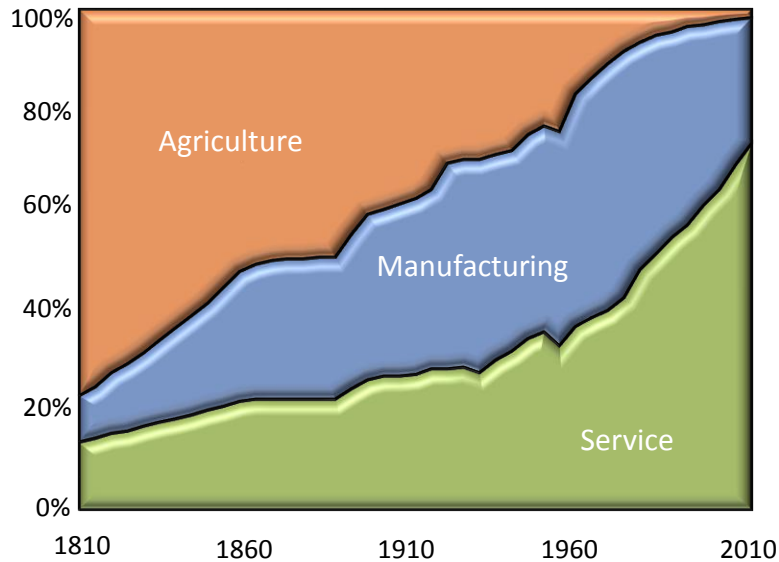
# WEB SERVICES - FOUNDATIONS

# Overview

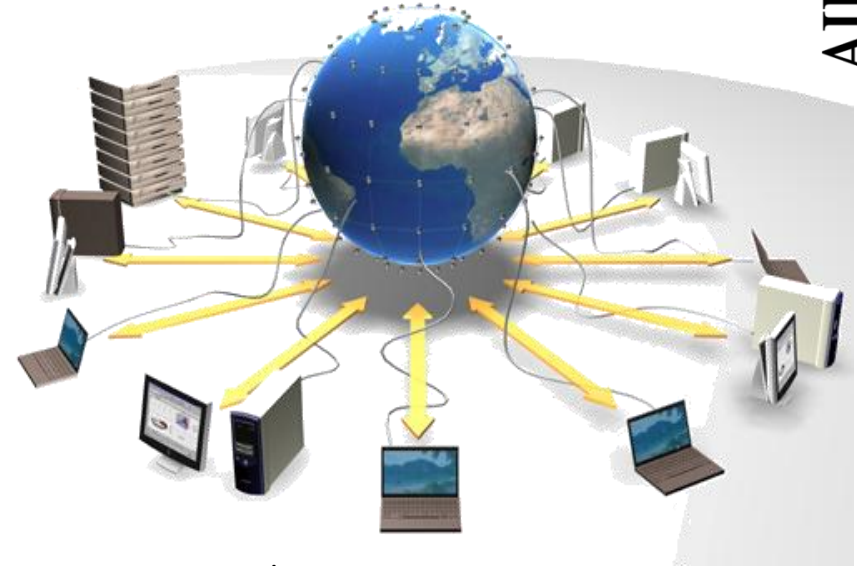
## → Services – Teaching and Research @ KIT

- Introduction to Web Services
- SOAP-based Web Services and WS\*-standards
- RESTful Web Services

# Why Service Research?



*Historical Development of GDP contribution of industrial sectors in Germany*

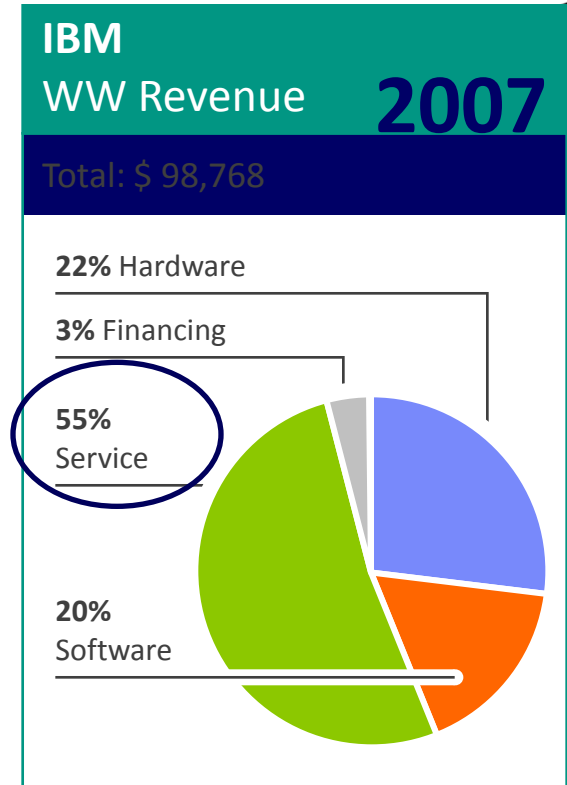
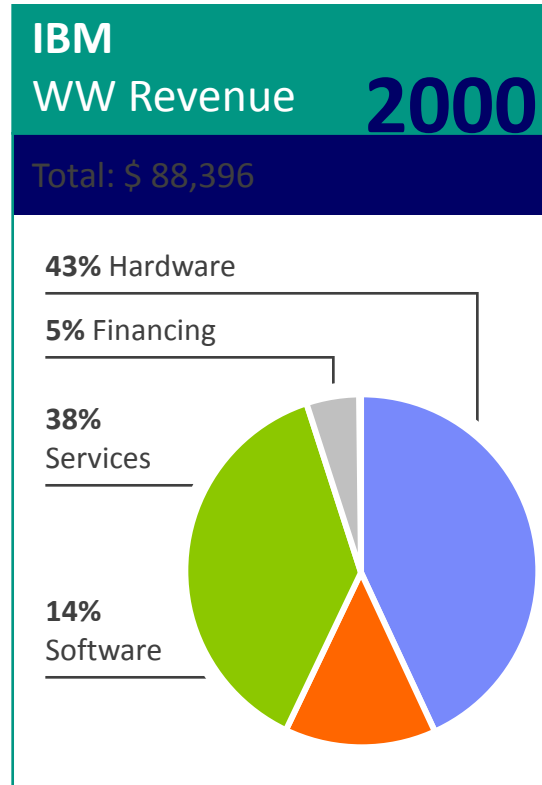
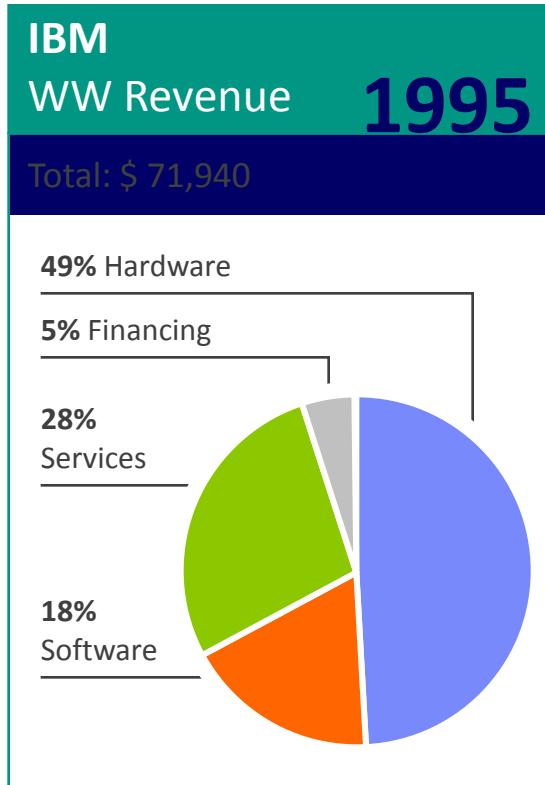


## Internet Evolution

- ICT drives de-/recomposition of value networks
- ICT drives location-independency of services

## SERVICE SCIENCES AND ENGINEERING

# “tertiarization” within companies



Source: IBM Annual Reports

## KIT founds “*Karlsruhe Service Research Institute*”

- Central hub for service research in Europe
- “Industry-on-Campus” concept as innovative Public Private Partnership model
- IBM first industry partner, open for additional partners
- Interdisciplinary approach to research and education
- Currently four research groups
  - Service Innovation & Management
  - eOrganisation
  - Knowledge Management
  - Information & Market Engineering



  **Press Release**  
No. 031 | ele | January 8, 2008

### Joint University, IBM Institute Partnership Strengthens Research and Training for Service Sector in Germany



**Dr. Elisabeth Zuber-Knost**  
Press Spokesperson  
Kaiserstrasse 12  
76131 Karlsruhe  
Tel.: +49 721 608-2089  
Fax: +49 721 608-3658

**Further Contacts:**  
University of Karlsruhe (TH)  
Institute of Applied Informatics and  
Formal Description Methods (AIFB)  
Steffen Lamparter  
Tel.: +49 721 608 7946  
E-mail: sla@aifb.uni-karlsruhe.de

IBM Deutschland GmbH  
Corporate Communications/  
University Relations  
Christian Schäfer  
Tel.: +49 711 785 4815  
E-mail: schaefer@de.ibm.com

**Partner:** Martin Jetter, CEO IBM Germany, and Horst Hippler, Head of Karlsruhe University, signing the contract. *Foto: Lilith Paul*

Today, the University of Karlsruhe and IBM Germany signed an agreement to set up a joint institute for service research. With the “Karlsruhe Service Research Institute”, the cooperation partners address the growing demand for academics who are specially trained for the service sector. They also plan to advance research in this field.

Provided the University’s governing body, the Senate, endorses the foundation in February, the Institute will commence work in the summer term of 2008. It will start by offering individual classes and seminars within the programs for Business and Information Engineering. In the medium term, the partners intend to offer a self-contained master program. “As a university of excellence we also want to lead in developing new programs and course offerings,” said Prof. Horst Hippler, Rector of Karlsruhe University, at the signing ceremony. The objective is to prepare students for key roles in industry, in research and in society. “The Institute will combine existing approaches at the University and strengthen them via closer linkage to business partners,” he said.

[www.kit.edu](http://www.kit.edu) Seite 1 / 2

KIT – die Kooperation von  
Forschungszentrum Karlsruhe GmbH  
und Universität Karlsruhe (TH)  Forschungszentrum Karlsruhe  
in der Helmholtz-Gemeinschaft  Universität Karlsruhe (TH)  
Forschungsuniversität • gegründet 1825

# Karlsruhe Service Research Institute

Service Innovation & Management

Knowledge Management

Information & Market Engineering

eOrganisation



Prof. Dr. Gerhard Satzger



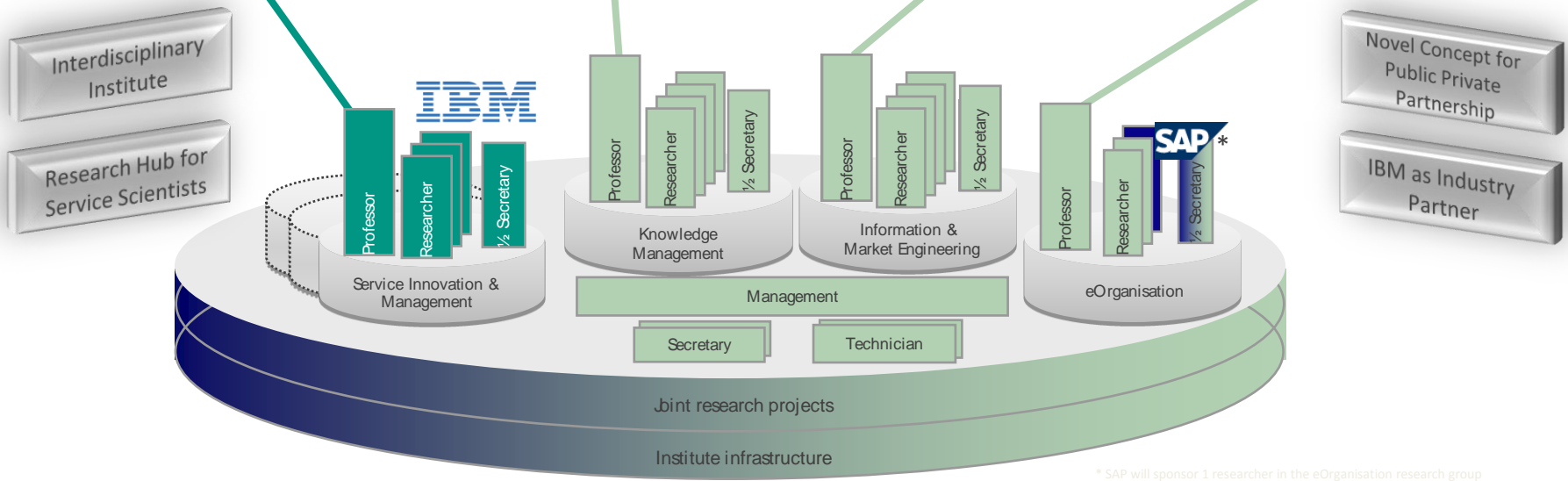
Prof. Dr. Rudi Studer



Prof. Dr. Christof Weinhardt



Prof. Dr. Stefan Tai



\* SAP will sponsor 1 researcher in the eOrganisation research group




In its research and teaching agenda, KSRI will focus on joint value creation and ICT enabled services


### Business view

Services := Deliverable (vs. Product)

### ICT\* view

Service := Software construct

means to „transport“  
value 

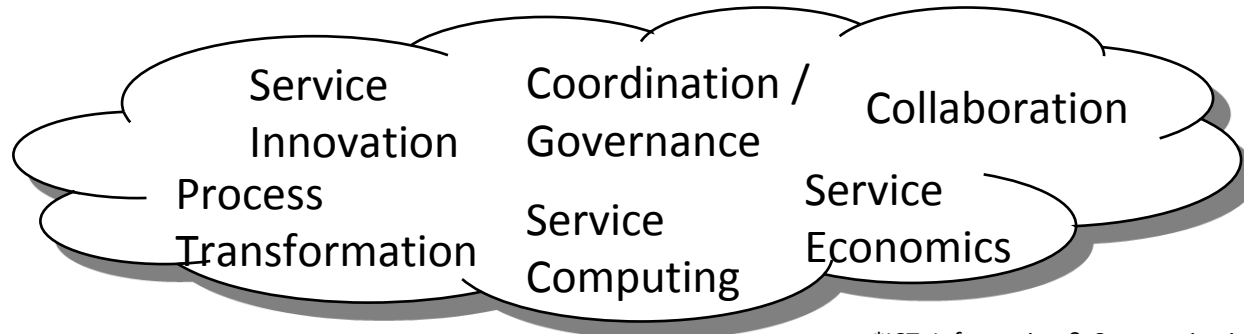
SOA, customer-provider  
interaction via Internet  


### Comprehensive view

Service := activity to co-create value



Joint value  
creation

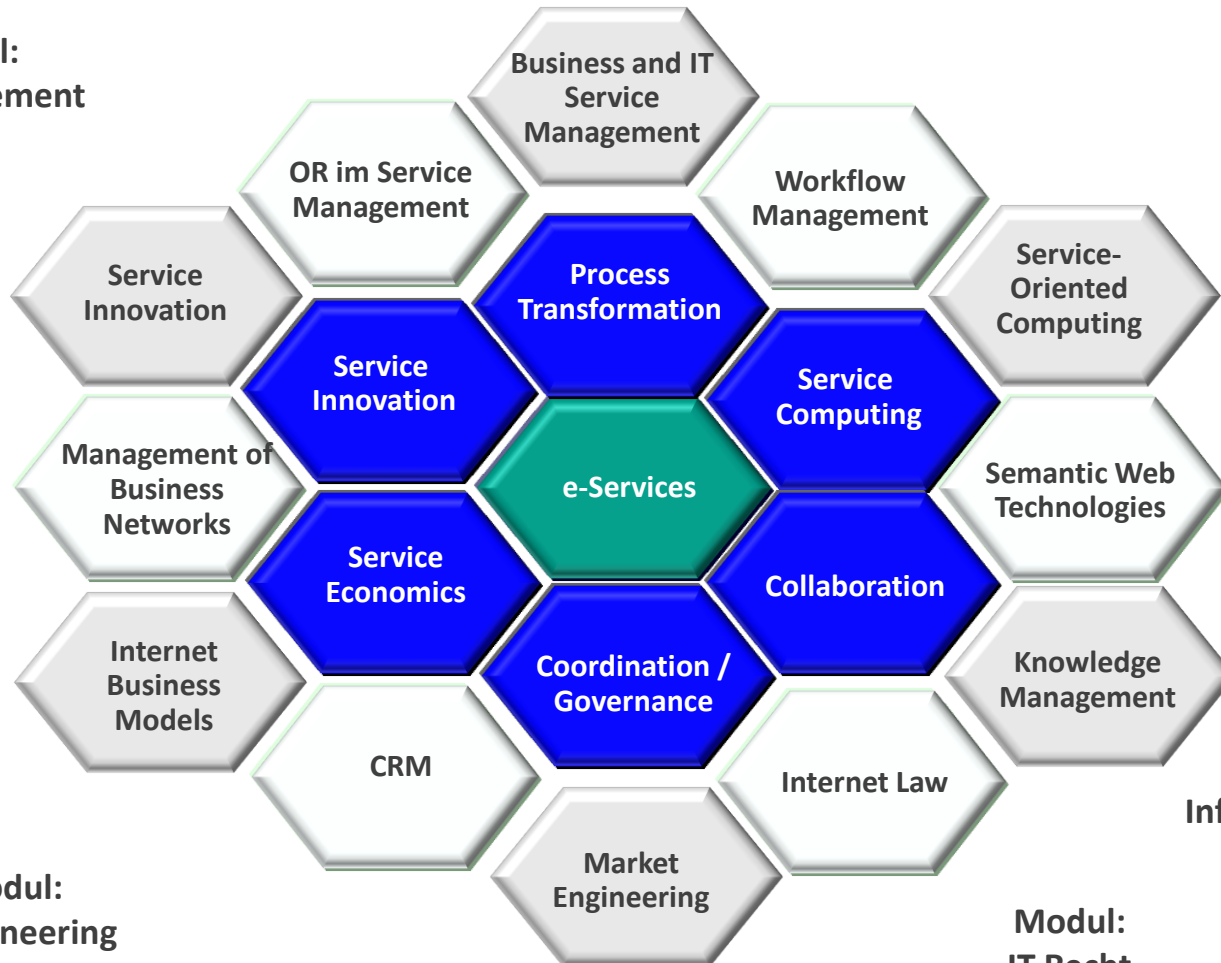


ICT  
enabled

\*ICT: Information & Communication Technology

# KSRI is driving a service-based curriculum based around six core topics

**BWL-Modul:  
Service Management**



# Overview

- Services – Teaching and Research @ KIT

## → Introduction to Web Services

- SOAP-based Web Services and WS\*-standards
- RESTful Web Services

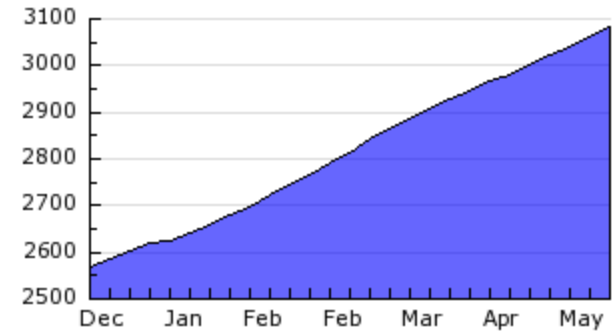
### Further readings:

- Wohlstadter, Tai: Web Services – Definition. 2008
- Michael P. Papazoglou: Web Services: Principles and Technology, Pearson Education Limited, 2008.
- Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju: Web Services: Concepts, Architectures and Applications, Springer, 2004.

# Web Services



[http://findnearby.net/new\\_york\\_ny/ipod/100](http://findnearby.net/new_york_ny/ipod/100)



ProgrammableWeb.com 06/07/08

# Web Services - Definition

- **Web services** provide the middleware that enables component communication over standard Web protocols
- Web services are particularly useful in a compositional approach to application development, where
  - Some features are provided externally through one or more remote systems
  - Existing legacy applications are wrapped using Web services technology as a more convenient format for interoperability

# Web Services - Definition

- Fundamentally, a Web service represents a **business function or business process** that is available over the Web as a service
- Software components (application logic) accessible via standard Web protocols
  - “Programming the Web”
    - “**remote procedure calls over the Web**”
  - Web sites with no user interface
- Available to any client that speaks the necessary Web protocols (XML, SOAP)
  - Platform independent components

# What are the benefits?

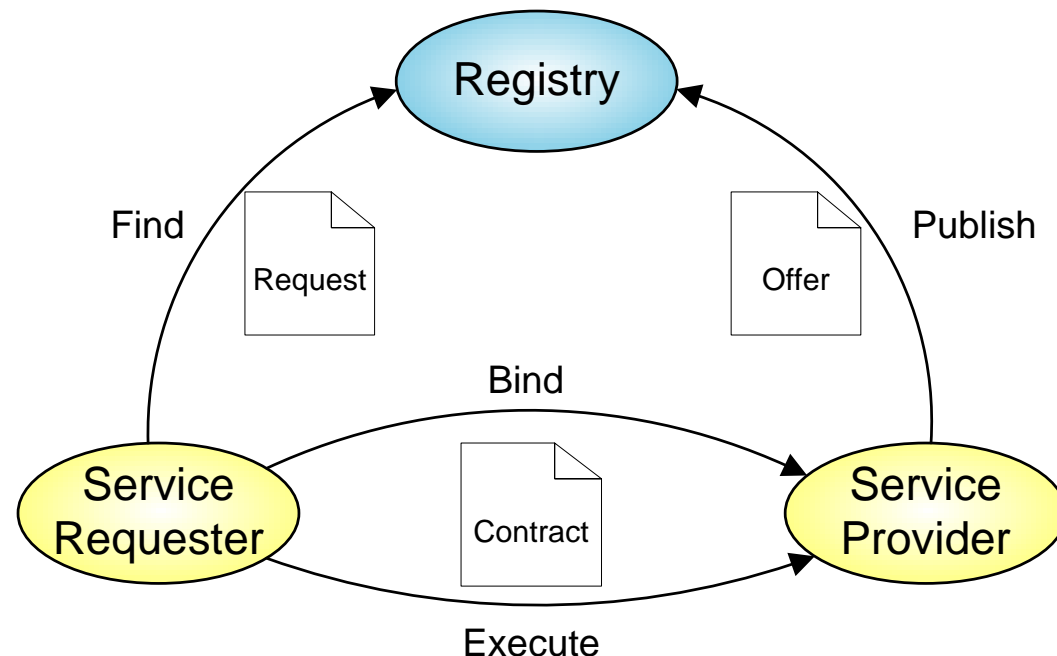
- Just-in-time integration of applications
- Reduce complexity by encapsulation
- Enable interoperability of legacy applications
- Minimize requirements for shared understanding

# How can I use a Web Service?

Clients need answers to following questions

- What services are available?
- How do I communicate with this particular service?
- Let's talk (give me some data)

**Publish – Find –  
Bind – Execute –  
Paradigm**





# Why are Web Services interesting?

## For Developers...

- Access to a “Web-wide library of software components”
- Smart development tools can...
  - Help you locate useful Web Services
  - Download Web Service descriptions (WSDL)
  - Automatically **generate code** to talk to the Web Service
  - On the server side, automatically generate service description for a service you write

# Use the Web Service in our program...

```
using System;  
using System.IO;
```

```
public class Quote {  
    public static void Main(String[] args) {  
        StockQuoteService service = new StockQuoteService();  
        float msftPrice = service.getQuote("MSFT");  
        Console.WriteLine(msftPrice);  
    }  
}
```

# Why are Web Services interesting?

## For Businesses...

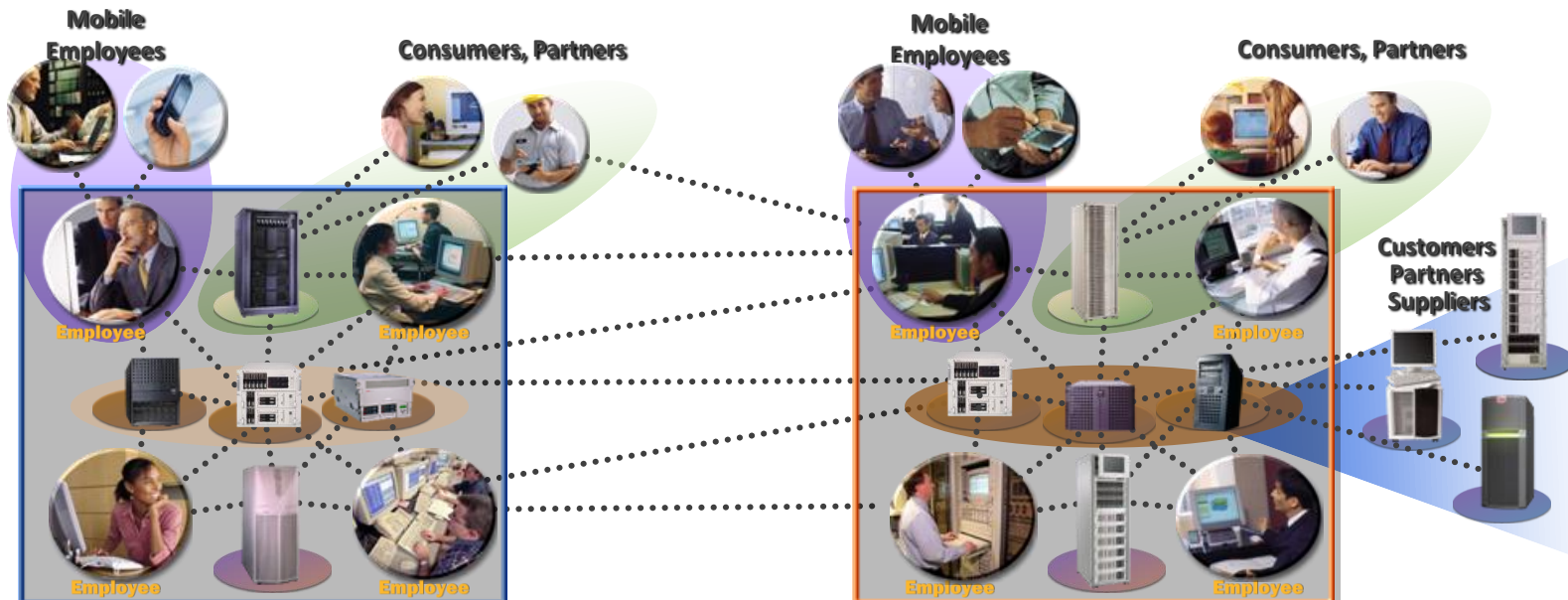
- Three keys to next generation applications:
  - “Any-to-Any” integration
    - Must tie together “islands of data, devices, OS, businesses, people”
  - Intelligent devices
    - Many types, with varying capabilities, but all speak common protocols
    - Anytime, anywhere access
    - Access and action
  - Open and accessible to all
    - Open, internet based standards
    - Broad accessibility

# New Applications

- Shift to decentralized/distributed architectures
- Span multiple clients, servers, services
- Federate across organizations
- Build systems that play in larger solutions

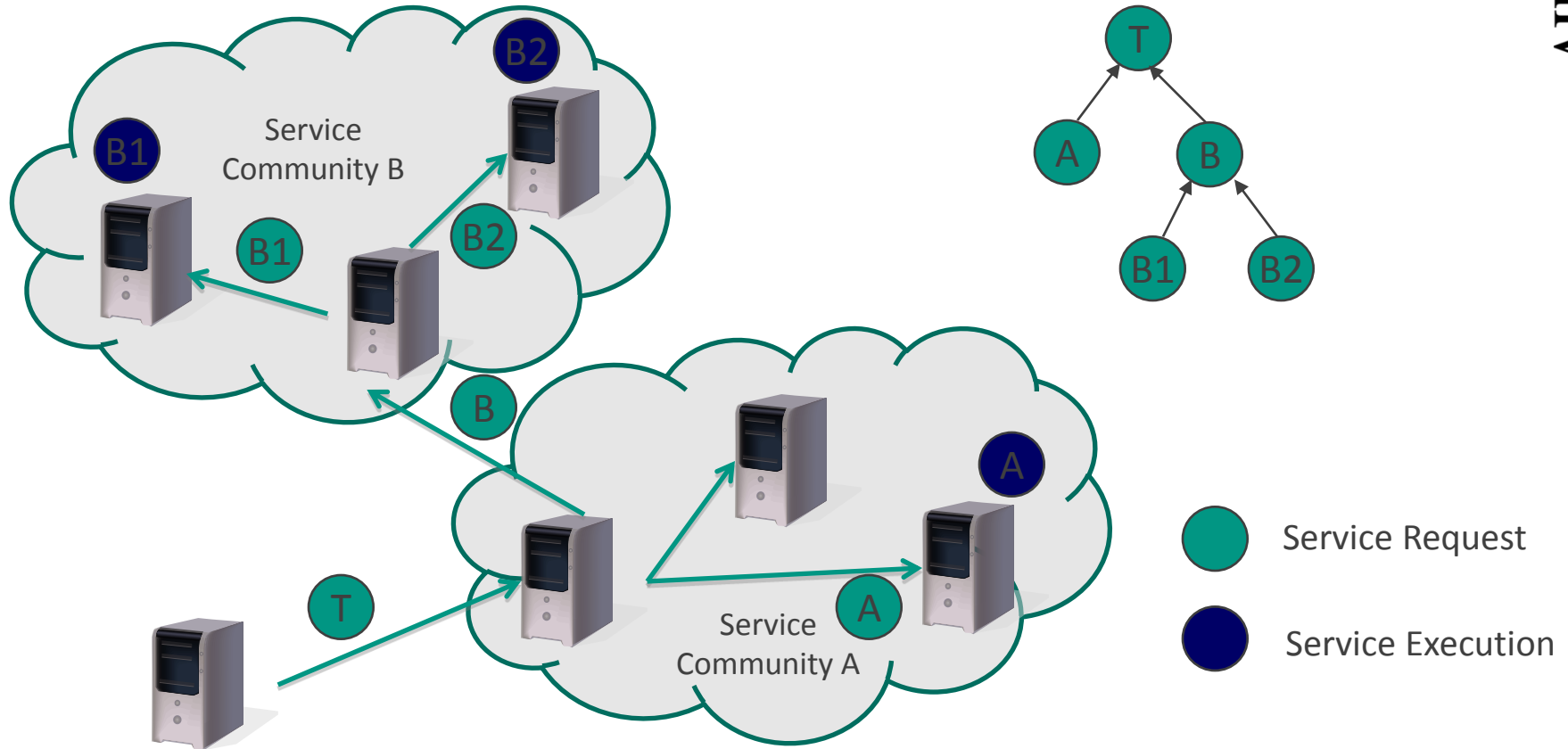
## Company A

## Company B



# Service Value Networks

## Service Composition Structure



- Providers can decompose Service Requests
- Local Routing Table for Request Forwarding

# Web Service Intermediaries

- Public UDDIs have not been adopted by industry
- Other Web service market platforms
  - Often domain dependent
  - Provide some added value on top
  - Examples:
    - Salesforce.com
    - Amazon.com
    - Strikelron.com
    - ProgrammableWeb.com

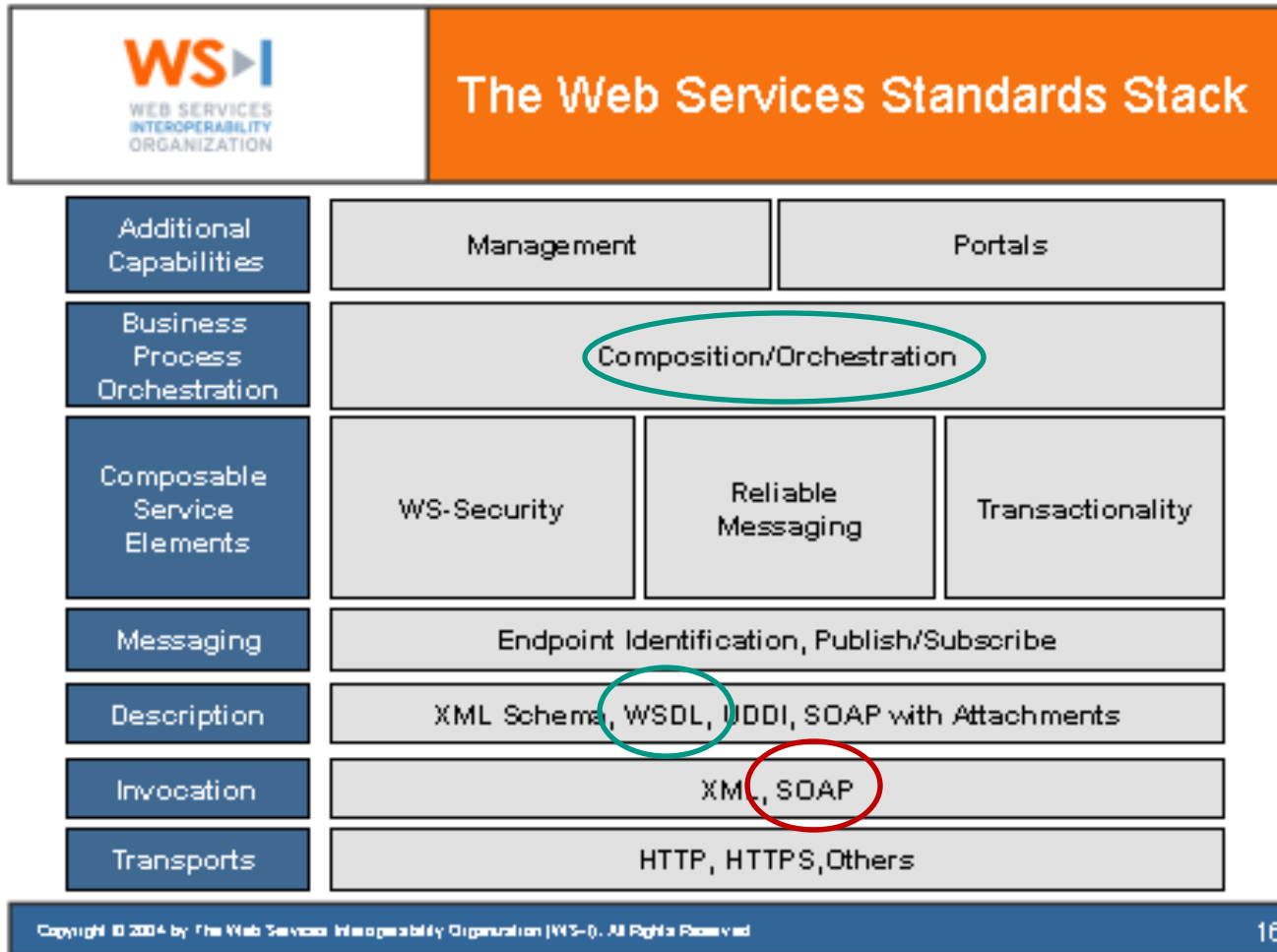
# Overview

- Services – Teaching and Research @ KIT
- Introduction to Web Services
- ➔ SOAP-based Web Services and WS\*-standards
- RESTful Web Services

## Further readings:

- Michael P. Papazoglou: *Web Services: Principles and Technology*, Pearson Education Limited, 2008.
- Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju: *Web Services: Concepts, Architectures and Applications*, Springer, 2004.

# WS-\*: Composeable Architecture





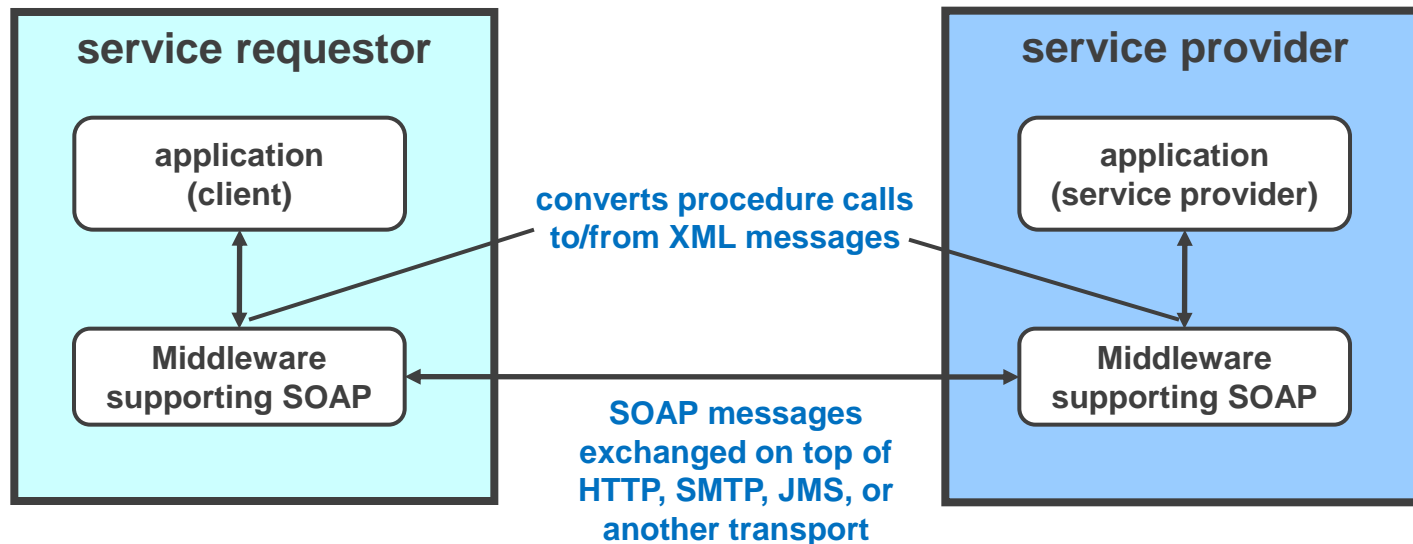
# SOAP

- SOAP is an XML-based standard for **Web services messaging**
  - SOAP is one of the two most popular standard technologies used for formatting application messages and for processing messages
- Reference material:
  - <http://www.w3.org/TR/soap/>

# SOAP as simple protocol

- SOAP does not include anything about:
  - reliability
  - complex message exchanges
  - transactions
  - security
  - ...
- As such, it is not adequate by itself to implement industrial strength applications that incorporate typical middleware features such as transactions or reliable delivery of messages
- SOAP does not prevent such features from being implemented but they need to be standardized to be useful in practice:
  - WS-Security
  - WS-Coordination
  - WS-Transactions
  - ...
- A wealth of additional standards are being proposed to add the missing functionality

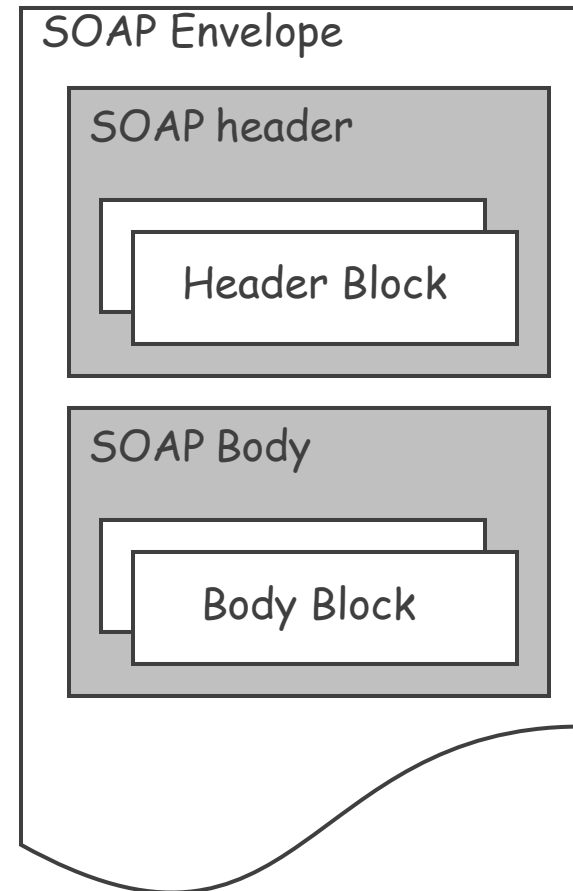
# Communication using SOAP



- SOAP mandates XML as the encoding scheme for messages sent over a variety of transports
- SOAP-over-HTML is a common model for Web services messaging

# SOAP messages

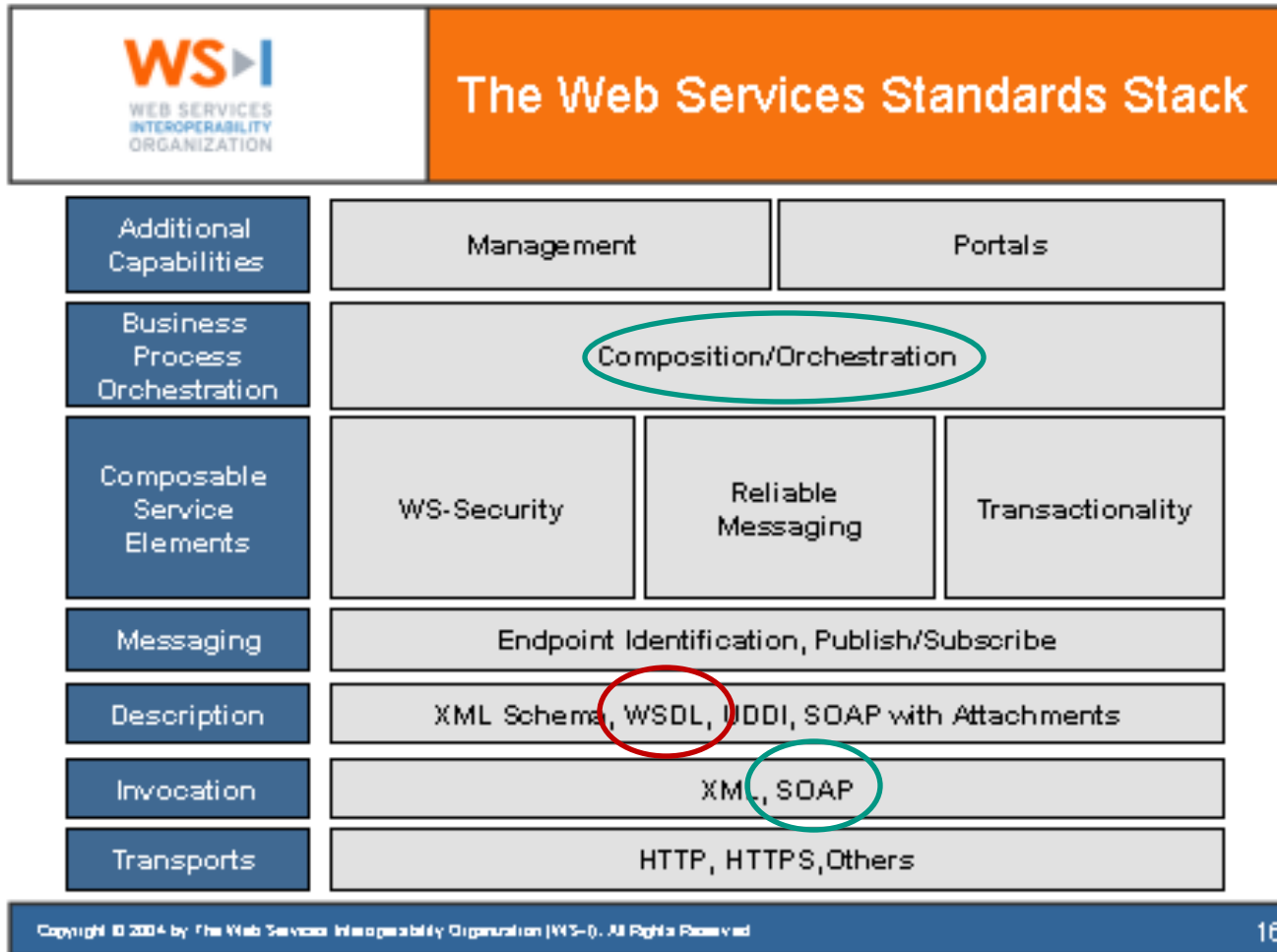
- SOAP messages are seen as **envelops** where the application encloses the data to be sent
- A message has two main parts:
  - **Header**: which can be divided into blocks
  - **Body**: which can be divided into blocks
- SOAP does not say what to do with the header and the body, it only states that the header is optional and the body is mandatory
- Use of header and body, however, is implicit. The body is for application level data. The header is for infrastructure level data.



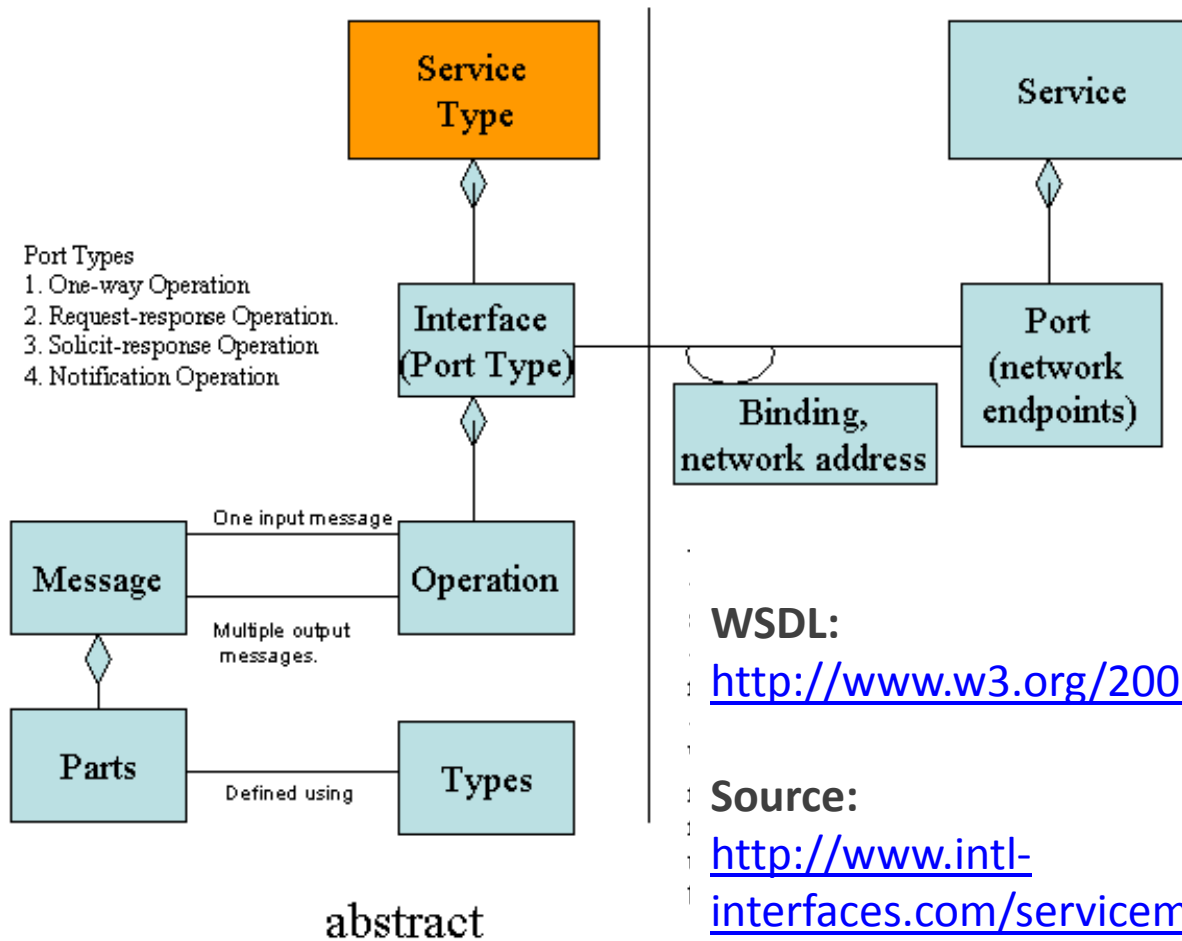
# SOAP Message

```
<env:Envelope xmlns:env=http://www.w3.org/2003/05/soap-envelope"
  xmlns:w=http://www.widget.inc/shop" xmlns:n="http://notaries.example.org">
  <env:Header> <w:ticket>54B42CF401A</w:ticket>
    <n:token>
      <n:value>32158546</n:value>
      <n:issuer>"http://notarypublic.example.com"</n:issuer>
    </n:token>
  </env:Header>
  <env:Body>
    <w:buy>
      <w:product>light gadget</w:product>
      <w:amount>430</w:amount>
    </w:buy>
  </env:Body>
</env:Envelope>
```

# WS-\*: Composeable Architecture



# Web Service Description Language (WSDL)



WSDL:

<http://www.w3.org/2002/ws/desc/>

Source:

[http://www.intl-interfaces.com/servicemodel/2001-06-21.GP-WSDL\\_object\\_diagram.png](http://www.intl-interfaces.com/servicemodel/2001-06-21.GP-WSDL_object_diagram.png)

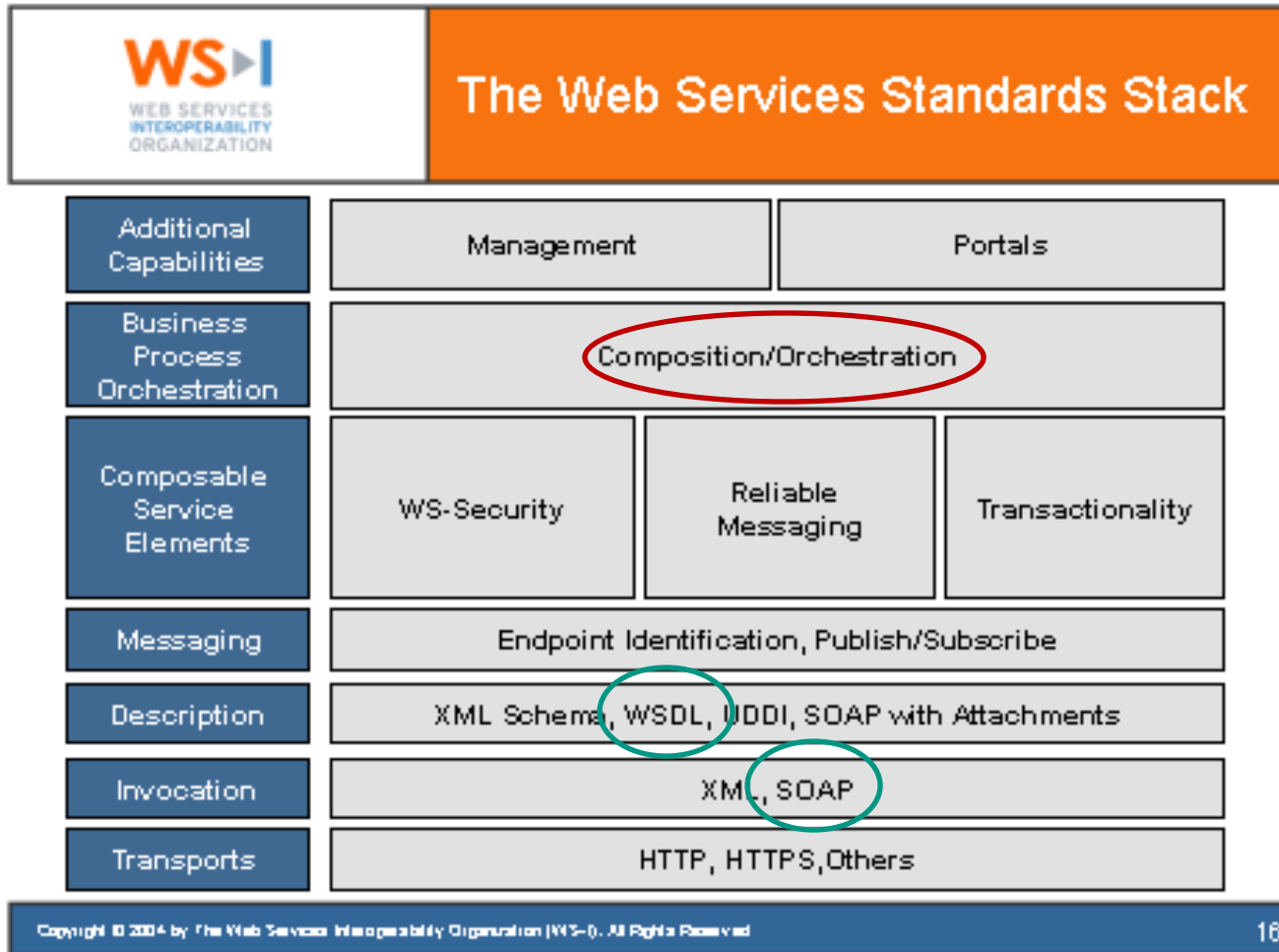
# Web Service Description Language (WSDL)

```
<description xmlns=http://www.w3.org/2004/08/wsdl targetNamespace="..." ...>
  <types>
    <!--XML Schema description of types being used in messages--> ...
  </types>
  <interface name="...">
    <!--list of operations and their input and output--> ...
  </interface>
  <binding name="..." interface="..." type="...">
    <!--message encodings and communication protocols--> ...
  </binding>
  <service name="..." interface="...">
    <!--combination of an interface, a binding, and a service location--> ...
  </service>
</description>
```

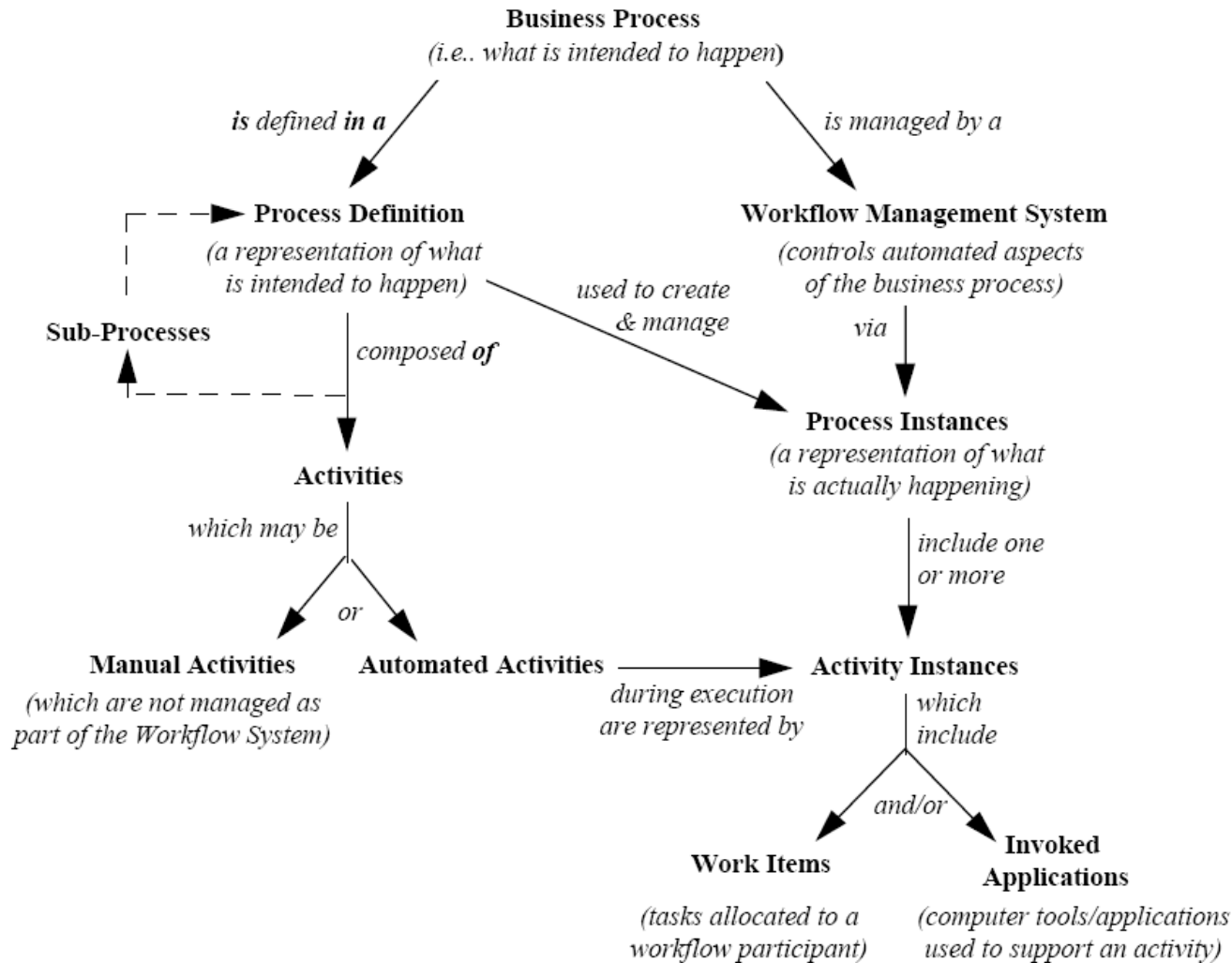
Example: <http://api.google.com/GoogleSearch.wsdl>



# WS-\*: Composeable Architecture



# Business Process Orchestration



[http://www.wfmc.org/standards/docs/TC-1011\\_term\\_glossary\\_v3.pdf](http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf)

# Composition/Orchestration

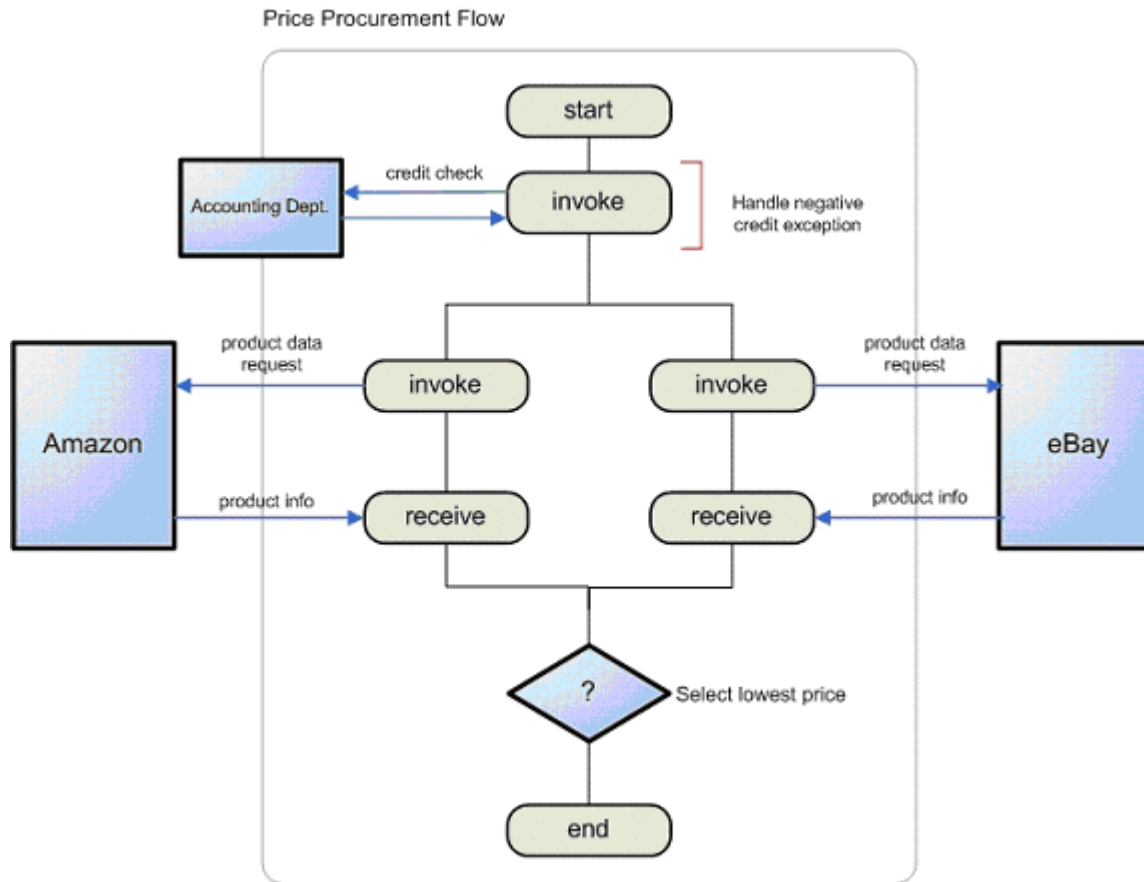
- Business Process:
  - "A set of one or more **linked procedures or activities** which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships."  
(Workflow Management Coalition (WfMC))
- Web Service Composition:
  - "A web service is a composition of other web service, when all its operations are implemented by invoking operations from other web services."
- Web Service Orchestration:
  - Generalization of "Composition"; an orchestration does not need to expose its functionality again as Web service
- Web Service Choreography
  - "A choreography is the observed sequence of message exchanged by peer services when performing a unit of work."

<http://www.ebpml.org/soa.htm>

## Business Process Execution Language (BPEL)

- An orchestration language provides a means to formally specify business processes and interaction protocols
- extends the Web Services interaction model and enables it to support business transactions
- describes complex work flows of business processes

<http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>



- BPEL provides a XML serialization of the process definition which can be executed by standard BPEL engines (Oracle, ActiveBPEL, etc.)

<http://www.devx.com/assets/vendshow/11625.gif>

# Web Services Standards Overview

## Interoperability Issues

- Basic Profile WS-I Basic Profile 1.1**
  - WS-I Basic Profile 1.1 consists of a set of standards that address interoperability issues in the context of an enterprise.
- Basic Profile WS-I Basic Profile 1.2**
  - WS-I Basic Profile 1.2 is a set of standards that address interoperability issues in the context of an enterprise.
- Attachments Profile WS-I Attachments Profile 1.1**
  - WS-I Attachments Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- Single SOAP Binding Profile WS-I Single SOAP Binding Profile 1.1**
  - WS-I Single SOAP Binding Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- Basic Security Profile WS-I Basic Security Profile 1.1**
  - WS-I Basic Security Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- REL Token Profile WS-I REL Token Profile 1.1**
  - WS-I REL Token Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- SAML Token Profile WS-I SAML Token Profile 1.1**
  - WS-I SAML Token Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- Confederation Chain Attachment Mechanism WS-I Confederation Chain Attachment Mechanism 1.1**
  - WS-I Confederation Chain Attachment Mechanism 1.1 is a set of standards that address interoperability issues in the context of an enterprise.
- Reliable Asynchronous Messaging Profile WS-I Reliable Asynchronous Messaging Profile 1.1**
  - WS-I Reliable Asynchronous Messaging Profile 1.1 is a set of standards that address interoperability issues in the context of an enterprise.

### Standards Bodies

- OASIS** The Organization for the Advancement of Structured Information Standards (OASIS) is a not-for-profit organization that develops and promotes standards for the information technology industry.
- W3C** The World Wide Web Consortium (W3C) is an international community that develops and maintains standards for the World Wide Web.
- ISO** The International Organization for Standardization (ISO) is an international organization that develops and publishes standards for a wide range of products, services, and systems.

## Business Process Specifications

- Business Process Execution Language for Web Services 1.1 WS-BPEL**
  - WS-BPEL is a language for describing business processes that can be executed by a Web Services engine.
- WS-Choreography Model Overview WS-CD**
  - WS-CD is a model for describing business processes that involve multiple participants.
- Web Service Choreography Interface WS-SCI**
  - WS-SCI is a language for describing business processes that involve multiple participants.
- Web Service Choreography Description Language WS-CDL**
  - WS-CDL is a language for describing business processes that involve multiple participants.
- Business Process Execution Language for Web Services 2.0 WS-BPEL20**
  - WS-BPEL20 is a language for describing business processes that can be executed by a Web Services engine.
- Business Process Management WS-BPM**
  - WS-BPM is a language for describing business processes that can be managed by a Web Services engine.
- XML Process Definition Language WS-XPL**
  - WS-XPL is a language for describing business processes that can be executed by a Web Services engine.

## Management Specifications

- Management Using Web Services WS-M**
  - WS-M is a language for describing management operations that use Web Services.
- Management Of Web Services WS-MOM**
  - WS-MOM is a language for describing management operations that use Web Services.
- WS-Management WS-MAN**
  - WS-MAN is a language for describing management operations that use Web Services.
- Service Modeling Language WS-SM**
  - WS-SM is a language for describing management operations that use Web Services.
- Service Binding Language WS-SBL**
  - WS-SBL is a language for describing management operations that use Web Services.

## Presentation Specifications

- WS-Addressing WS-Addressing 1.0**
  - WS-Addressing 1.0 is a language for describing presentation operations that use Web Services.
- WS-Reliability WS-Reliability 1.0**
  - WS-Reliability 1.0 is a language for describing presentation operations that use Web Services.

## Metadata Specifications

- WS-Policy WS-P**
  - WS-P is a language for describing metadata specifications that use Web Services.
- WS-PolicyAttachments WS-PA**
  - WS-PA is a language for describing metadata specifications that use Web Services.
- WS-PolicyAssertions WS-PA**
  - WS-PA is a language for describing metadata specifications that use Web Services.
- WS-Discovery WS-DISCOVERY**
  - WS-DISCOVERY is a language for describing metadata specifications that use Web Services.
- WS-Discovery Metadata WS-DM**
  - WS-DM is a language for describing metadata specifications that use Web Services.
- WS-MetadataExchange WS-MEX**
  - WS-MEX is a language for describing metadata specifications that use Web Services.
- Web Service Description Language 2.0 WS-SDL**
  - WS-SDL 2.0 is a language for describing metadata specifications that use Web Services.
- Web Service Description Language 2.0 Core WS-SDL**
  - WS-SDL 2.0 Core is a language for describing metadata specifications that use Web Services.
- Web Service Description Language 1.1 WS-SDL**
  - WS-SDL 1.1 is a language for describing metadata specifications that use Web Services.

## Reliability Specifications

- WS-ReliableMessaging WS-RM**
  - WS-RM is a language for describing reliability specifications that use Web Services.
- WS-Reliable Messaging Policy Assertion WS-RM-PA**
  - WS-RM-PA is a language for describing reliability specifications that use Web Services.
- WS-Reliability WS-RELIABILITY**
  - WS-RELIABILITY is a language for describing reliability specifications that use Web Services.

## Security Specifications

- WS-Security WS-S**
  - WS-S is a language for describing security specifications that use Web Services.
- WS-SecurityPolicy WS-SP**
  - WS-SP is a language for describing security specifications that use Web Services.
- WS-Security: SOAP Security WS-SS**
  - WS-SS is a language for describing security specifications that use Web Services.
- WS-Security: Username Token Profile WS-ST**
  - WS-ST is a language for describing security specifications that use Web Services.
- WS-Security: Kerberos Binding WS-KB**
  - WS-KB is a language for describing security specifications that use Web Services.
- WS-Federation WS-FED**
  - WS-FED is a language for describing security specifications that use Web Services.
- WS-Security: SAML Token Profile WS-ST**
  - WS-ST is a language for describing security specifications that use Web Services.
- WS-Security: X.509 Certificate Token Profile WS-ST**
  - WS-ST is a language for describing security specifications that use Web Services.
- WS-Security: X.509 Certificate Token Profile WS-ST**
  - WS-ST is a language for describing security specifications that use Web Services.
- WS-Security: X.509 Certificate Token Profile WS-ST**
  - WS-ST is a language for describing security specifications that use Web Services.

## Transaction Specifications

- WS-Coordination WS-COORD**
  - WS-COORD is a language for describing transaction specifications that use Web Services.
- WS-Business Activity WS-BA**
  - WS-BA is a language for describing transaction specifications that use Web Services.
- WS-Atomic Transaction WS-AT**
  - WS-AT is a language for describing transaction specifications that use Web Services.
- WS-Composite Application Framework WS-CAF**
  - WS-CAF is a language for describing transaction specifications that use Web Services.
- WS-Coordination Framework WS-COORD**
  - WS-COORD is a language for describing transaction specifications that use Web Services.
- WS-Transaction Management WS-TM**
  - WS-TM is a language for describing transaction specifications that use Web Services.
- WS-Transaction Management WS-TM**
  - WS-TM is a language for describing transaction specifications that use Web Services.

## Resource Specifications

- Web Services Resource Framework WS-RF**
  - WS-RF is a language for describing resource specifications that use Web Services.
- WS-Base-URL WS-BU**
  - WS-BU is a language for describing resource specifications that use Web Services.
- WS-ServiceGroup WS-SG**
  - WS-SG is a language for describing resource specifications that use Web Services.
- WS-ResourceProperties WS-RP**
  - WS-RP is a language for describing resource specifications that use Web Services.
- WS-ResourceLocation WS-RL**
  - WS-RL is a language for describing resource specifications that use Web Services.
- WS-ResourceTransfer WS-RT**
  - WS-RT is a language for describing resource specifications that use Web Services.
- Resource Representation WS-RR**
  - WS-RR is a language for describing resource specifications that use Web Services.

## Messaging Specifications

- WS-Notification WS-N**
  - WS-N is a language for describing messaging specifications that use Web Services.
- WS-BrokerNotification WS-BN**
  - WS-BN is a language for describing messaging specifications that use Web Services.
- WS-BaseNotification WS-BN**
  - WS-BN is a language for describing messaging specifications that use Web Services.
- WS-Eventing WS-E**
  - WS-E is a language for describing messaging specifications that use Web Services.
- WS-Addressing - Core WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.
- WS-Addressing - SOAP WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.
- WS-Addressing - SOAP WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.
- WS-Addressing - SOAP WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.
- WS-Addressing - SOAP WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.
- WS-Addressing - SOAP WS-Addressing**
  - WS-Addressing is a language for describing messaging specifications that use Web Services.

## SOAP

- SOAP WS-SOAP**
  - WS-SOAP is a language for describing messaging specifications that use Web Services.
- SOAP Message Transmission Optimization Mechanism WS-MTOM**
  - WS-MTOM is a language for describing messaging specifications that use Web Services.

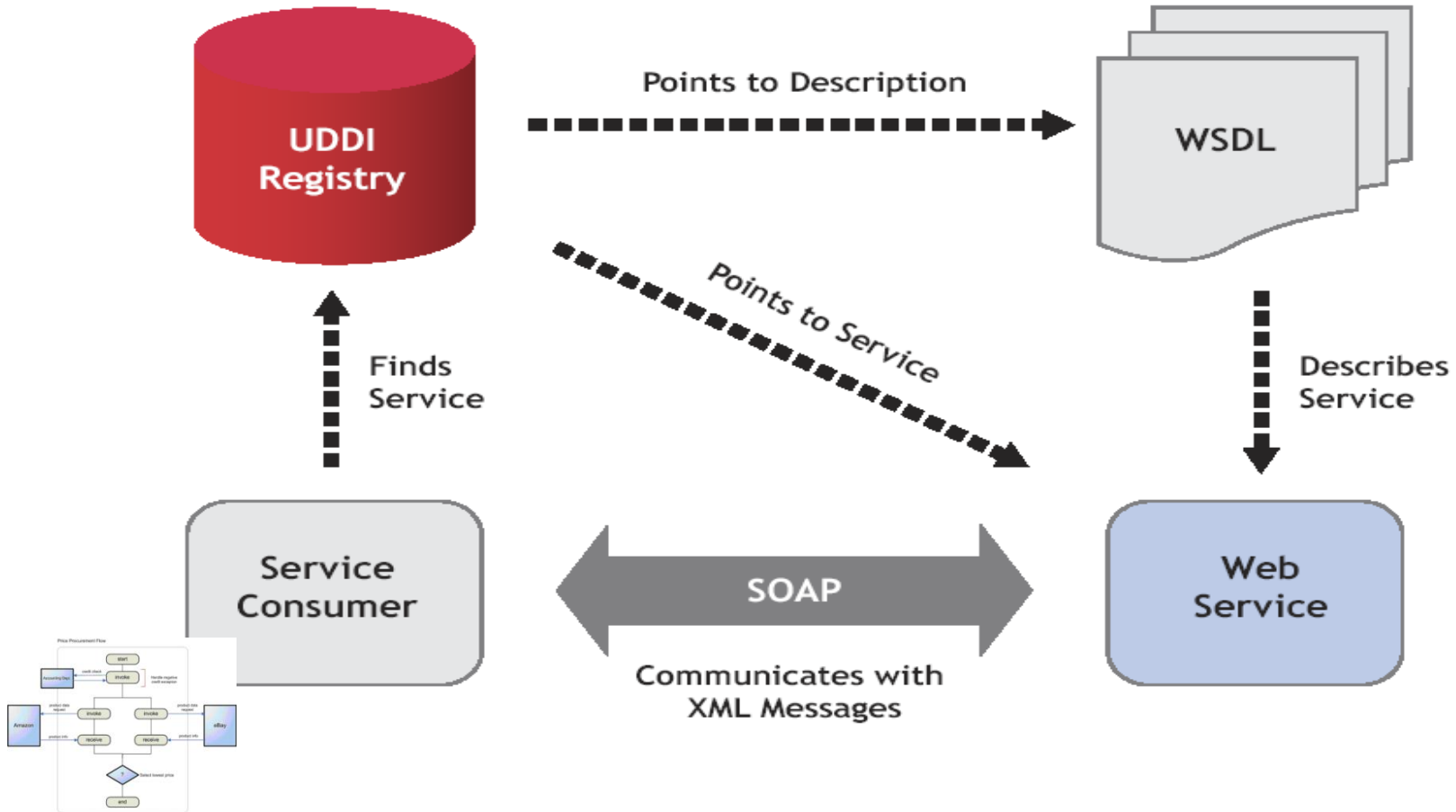
## XML Specifications

- XML 1.1 WS-XM11**
  - WS-XM11 is a language for describing XML specifications that use Web Services.
- XML 1.0 WS-XM10**
  - WS-XM10 is a language for describing XML specifications that use Web Services.
- Namespaces in XML WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Information Set WS-XMIS**
  - WS-XMIS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.
- XML Schema WS-XMNS**
  - WS-XMNS is a language for describing XML specifications that use Web Services.

## Dependencies



# Summary – SOAP-based Web Services



# Overview

- Services – Teaching and Research @ KIT
  - Introduction to Web Services
  - SOAP-based Web Services and WS\*-standards
- ➔ RESTful Web Services

## Further readings:

- Roy Fielding, *Architectural Styles and the Design of Network-based Software Architectures, PhD Dissertation, 1994* (<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>)
- Richardson & Ruby, *RESTful Web Services*, O'Reilly, 2007



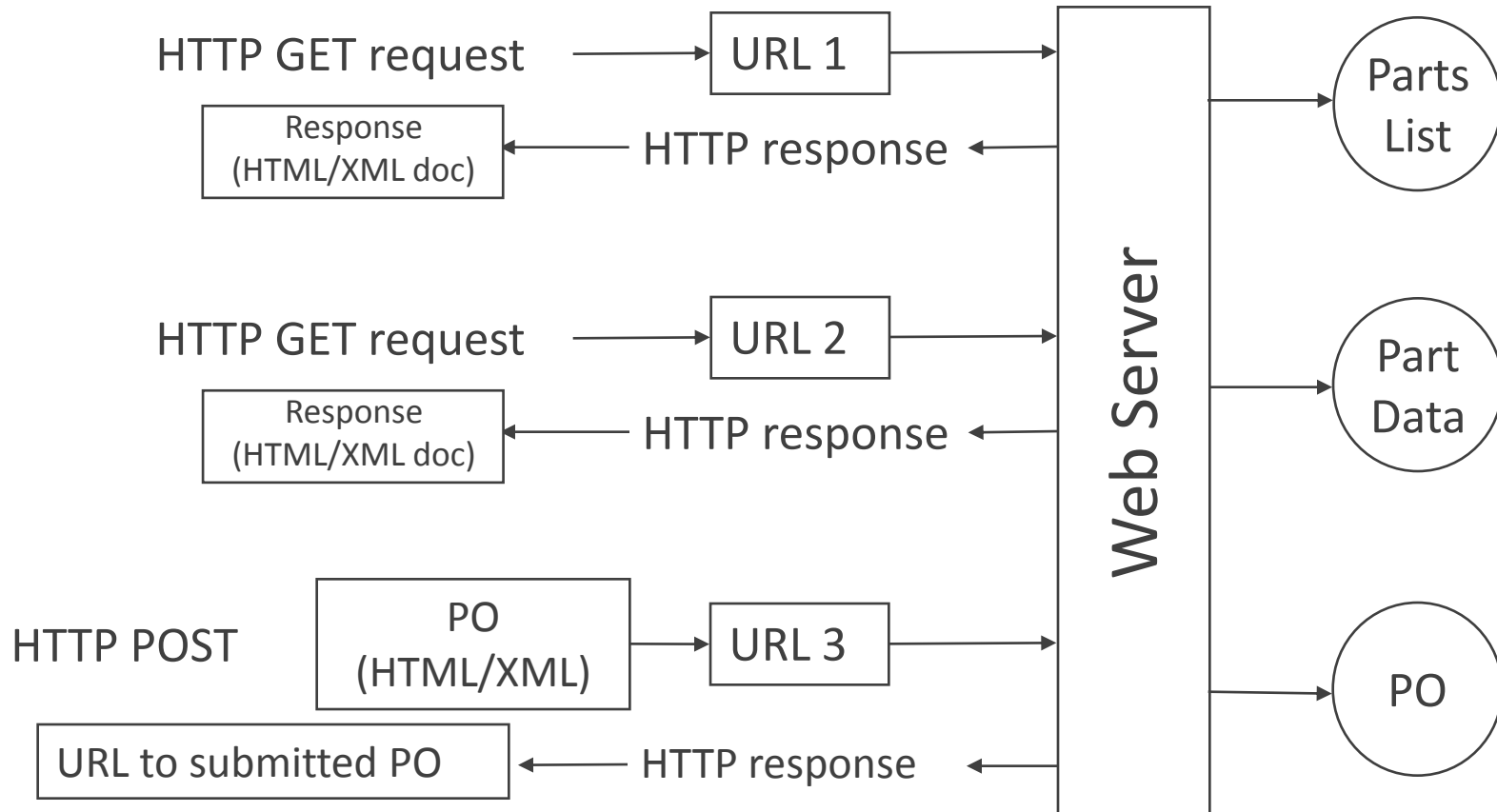
# REST – REpresentational State Transfer

- REST is the architecture of the Web; its principles have been used to explain why the HTTP protocol scales so well (*Roy Fielding, PhD Dissertation, 1994, HTTP 1.0 1989*)
  - Application state and functionality are divided into **resources**
  - Every resource is uniquely addressable using a **universal syntax** for use in hypermedia links
  - All resources share a **uniform interface** for the transfer of state between client and resource, consisting of
    - A constrained set of **well-defined operations**
    - A constrained set of **content types**, optionally supporting code on demand

# The four key principles of REST

1. **Resource Identification** through URI
2. **Uniform Interface** for all resources:
  - GET (Query the state, idempotent, can be cached)
  - POST (Modify, transfer the state)
  - PUT (Create a resource)
  - DELETE (Delete a resource)
3. **“Self-Descriptive”** Messages through Meta-Data
4. **Hyperlinks** to define the application state

# Example: RESTful services to get a list of parts, get info about a part, and to submit a Purchase Order (PO)



# Data Returned – Parts List and Parts Data

```
<?xml version="1.0"?>
<Parts>
  <Part id="00345" href="http://www.parts-depot.com/parts/00345"/>
  <Part id="00346" href="http://www.parts-depot.com/parts/00346"/>
  <Part id="00347" href="http://www.parts-depot.com/parts/00347"/>
  <Part id="00348" href="http://www.parts-depot.com/parts/00348"/>
</Parts>
```

- Note that the parts list has links to get detailed info about each part.
- The client transfers from one state to the next by examining and choosing from among the alternative URLs in the response document.

```
<?xml version="1.0"?>
<Part>
  <Part-ID>00345</Part-ID>
  <Name>Widget-A</Name>
  <Description>This part is used within the frap assembly</Description>
  <Specification href="http://www.parts-depot.com/parts/00345/specification"/>
  <UnitCost currency="USD">0.10</UnitCost>
  <Quantity>10</Quantity>
</Part>
```

# REST Strengths

- Simplicity
  - Uniform interface is **immutable** (no problem of breaking clients)
- HTTP/POX is **ubiquitous** (goes through firewalls)
- Stateless/synchronous interaction
- Proven **scalability**
  - “After all the Web works”, caching, clustered server farms for QoS
- Perceived ease of adoption (**light infrastructure**)
  - Just need a browser to get started - no need to buy WS-\* middleware
- Grassroots approach
- Leveraged by all major Web 2.0 applications
  - 85% clients prefer Amazon RESTful API  
(<http://www.oreilynet.com/pub/wlg/3005>)

# REST Weaknesses

- Mapping REST-style synchronous semantics on top of back end systems creates **design mismatches** (when they are based on asynchronous messaging or event driven interaction)
- Cannot deliver enterprise-style transport mechanisms beyond HTTP/SSL
- Challenging to identify and locate resources appropriately in all applications
- Apparent lack of standards (other than URI, HTTP, XML, MIME, HTML)
- Semantics/Syntax description very informal (user/human oriented)

# From REST to WS-\*

- Not everything is about HTTP, XML, XML Schema
- Not all interactions are synchronous request/response
- Full security, reliability etc. not needed all the time, but some times: feature composeability
  
- ... the WS-\* platform architecture defines a set of specifications, where each specification addresses one specific concern and has value in its own right. All the specifications are designed to work in conjunction with each other, building on the basic Web services specifications WSDL and SOAP.

# Tutorials on the Web

- Web services
  - <http://www.roseindia.net/webservices/>
- REST
  - <http://learn-rest.blogspot.com/2008/02/what-is-rest.html>
- BPEL
  - [http://www.eclipse.org/tptp/platform/documents/design/choreography\\_html/tutorials/wsbpel\\_tut.html](http://www.eclipse.org/tptp/platform/documents/design/choreography_html/tutorials/wsbpel_tut.html)
  - <http://developer.capeclear.com/?q=bpeltraining>
- ... and may more...



# Acknowledgements

- Stefan Tai, Service-oriented Computing, SS08, Universität Karlsruhe (TH)
  - [https://ilias.rz.uni-karlsruhe.de/goto\\_rz-uka\\_crs\\_3782.html](https://ilias.rz.uni-karlsruhe.de/goto_rz-uka_crs_3782.html)
- Brian LaMachhia, CS155, Spring 03, Yale University
  - <http://zoo.cs.yale.edu/classes/cs155/spr03/lectures.html>
- An Introduction to XML and Web Technologies - Web Services, Anders Mueller & Michael I. Schwartzbach 2006, Addison-Wesley

# Next lectures

- Next lecture on „Semantic Web Services“
  - Wednesday (June 11, 2008)
  - Topics:
    - Semantic Web and Web Services
    - Semantic Mash-ups
    - Semantic Management of Web Services
- Invited talk: York Sure (SAP)
  - Topic: Internet of Services
  - Wednesday (June 18, 2008)