

Web Services

Markus Schulz

Seminar: XML für Fortgeschrittene

30.06.2003

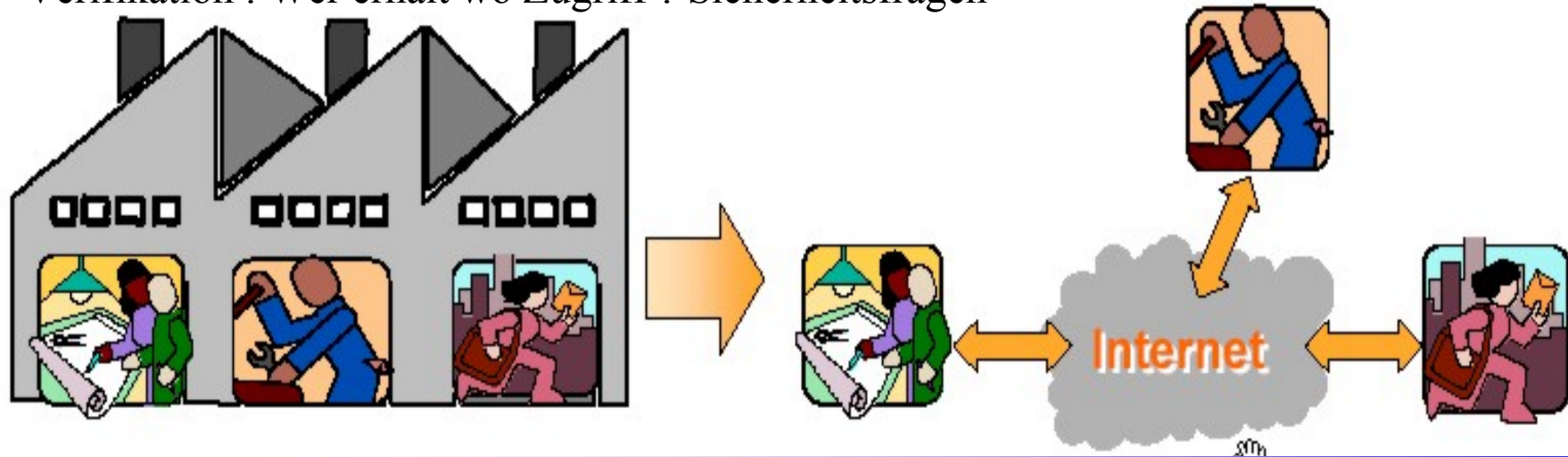
Vortragsgliederung

- 1. Motivation
- 2.-8. WS : Definition, Ansatz, Architektur, ...
- 9.x. SOAP : Definition, Geschichte, ...
- 10.x.x. WSDL : siehe oben...
- 11.x. UDDI Überblick
- 12.-14. Quo vadis WS ?

1. Motivation

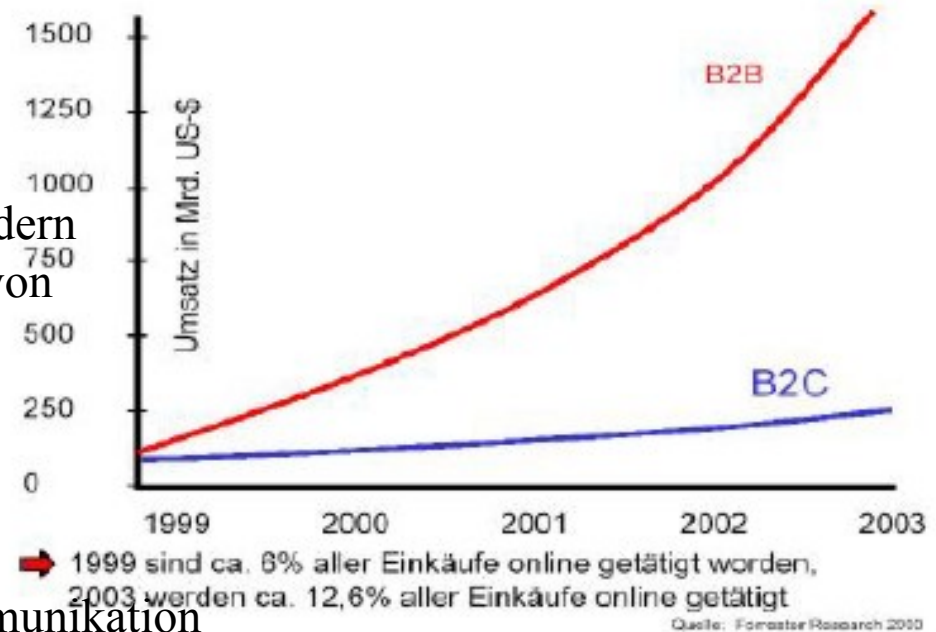
In der heutigen Zeit sind folgende wirtschaftliche Aspekte zu betrachten:

- Dezentralisierung : Verteilung, Auslagerung bestimmter Unternehmensteile
- Modularisierung : Wiederverwendbarkeit erhöht wirtschaftl. Nutzen
- Standardisierung : schafft einheitliche Rahmenbedingungen
- Modellierung : strukturierte Herangehensweise
- Komposition : fasst bestehende Module zusammen
- Verifikation : Wer erhält wo Zugriff ? Sicherheitsfragen



2. WS-Ansatz

- Business to Business (B2B) wird immer wichtiger (siehe Abbildung rechts)
 - verteilte Unternehmen sind effizient
 - Kostendruck steigt
- B2B ist dynamisch
 - Anpassung an Kundenwünsche
 - Rahmenbedingungen können sich ändern
 - ➔ Erfordern z.B. schnellen Wechsel von Geschäftspartnern
- B2B ist aufwendig
 - heterogene Systeme üblich
 - Welche Standards existieren ?
- B2B bedeutet meist (automatische) Kommunikation zwischen zwei oder mehreren Softwaresystemen



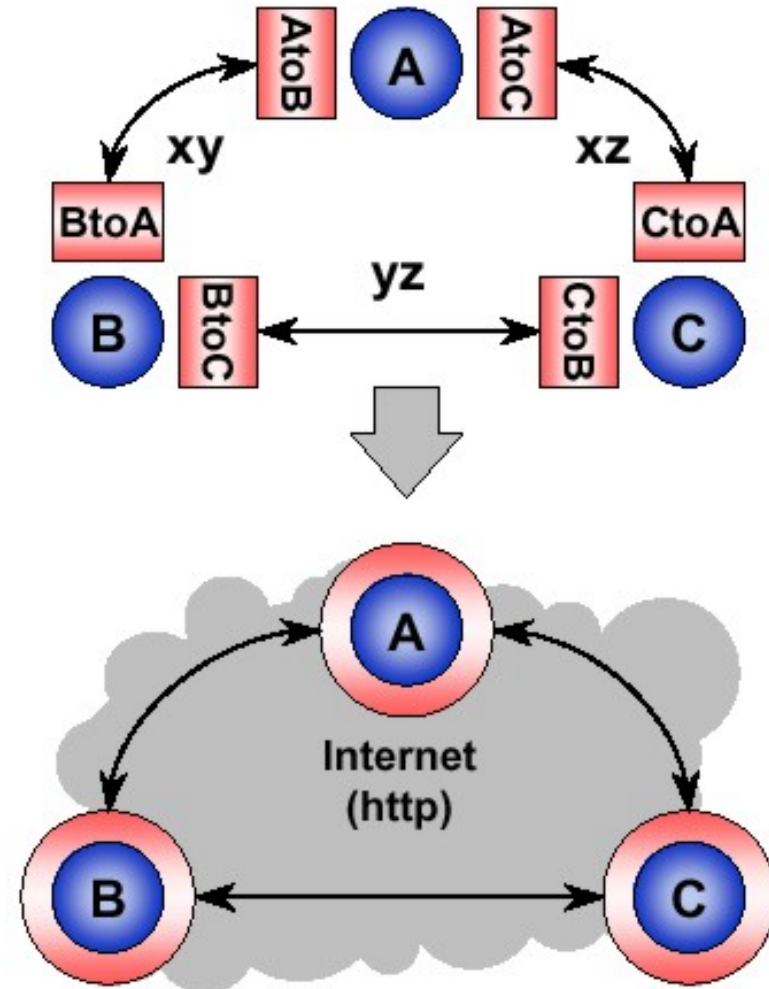
3. WS-Definition

- Web Service Definition (W3C):

A Web service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.

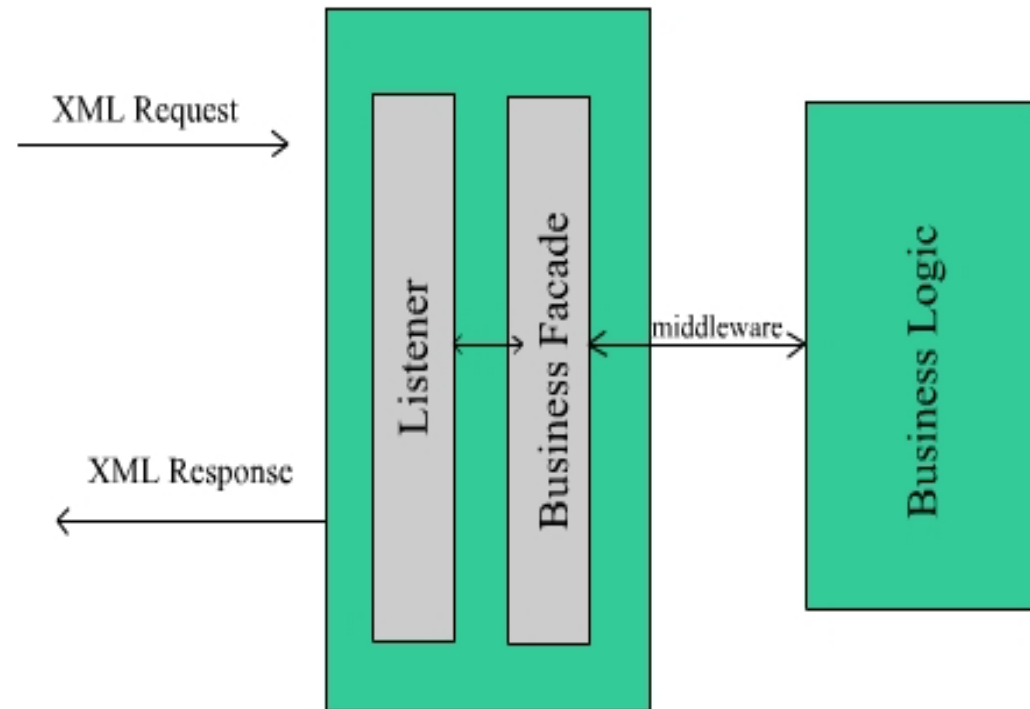
4. WS-Funktionalität

- Web Service Definition :
 - WS kapseln Funktionalität einer Anwendung und bringen diese standardisiert ins Internet
 - hüllen existierende Systeme ein
- ➔ Plattform-, Protokoll-, Sprachen-unabhängig (vs. CORBA, DCOM ,etc.)
- ➔ Vorteile des Zugang übers Internet (Verfügbarkeit, etc.)
- ➔ modularer Aufbau erhöht Wiederverwendbarkeit

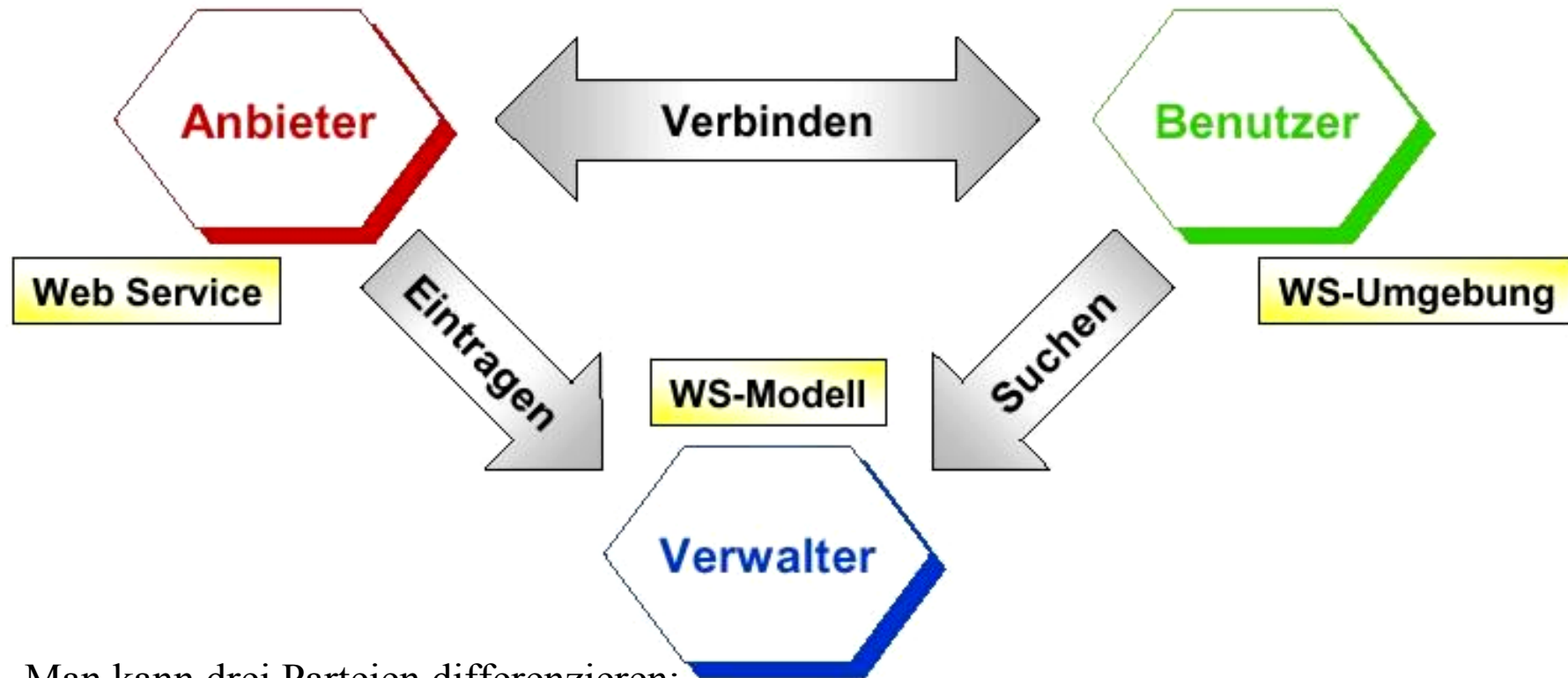


5. WS-Architektur

- MS, SAP, SUN, HP, IBM, Oracle, Intel treiben die Entwicklung von WS voran
- ein WS besteht aus mehreren XML-Dokumenten
- Es ist ein standard. Web-Interface, das die darunter liegende Geschäftslogik kaschiert
- ein typisches WS-Modul wartet auf eine XML-Anfrage, um eventuell darauf zu reagieren



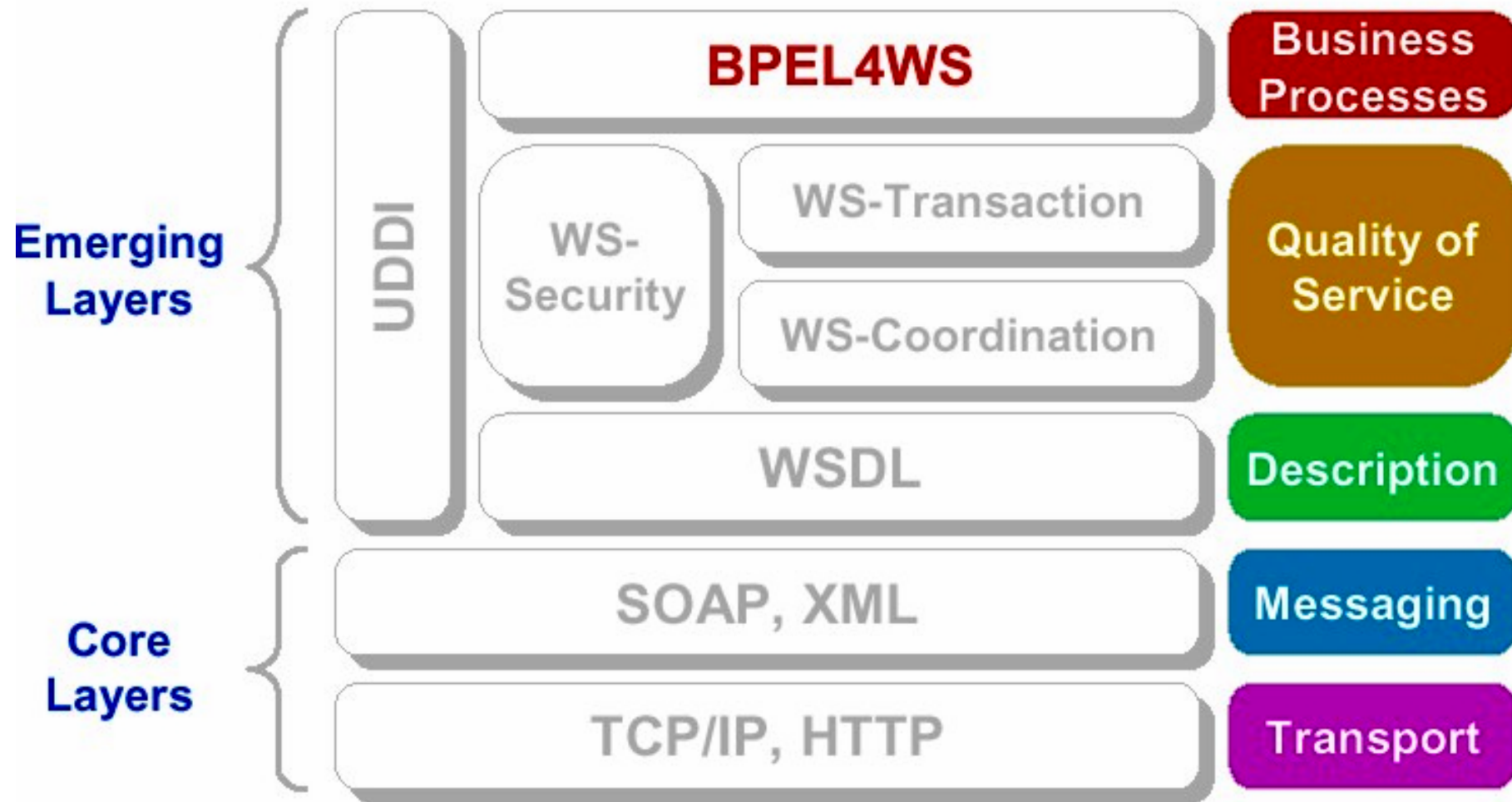
6. WS-Integration (Service-orientierte Architektur (SOA))



Man kann drei Parteien differenzieren:

- **Anbieter** : will einen Dienst über das Netz anbieten und meldet diesen an
 - **Verwalter**: stellt in einer großen DB verschiedene WS vor (Nutzen, techn. Aspekte, etc.)
 - **Benutzer** : will einen Dienst nutzen und informiert sich deshalb beim Verwalter
- !!! Idealmodell; Anbieter und Benutzer können auch ohne Verwalter agieren !!!

7. WS-Technology Stack



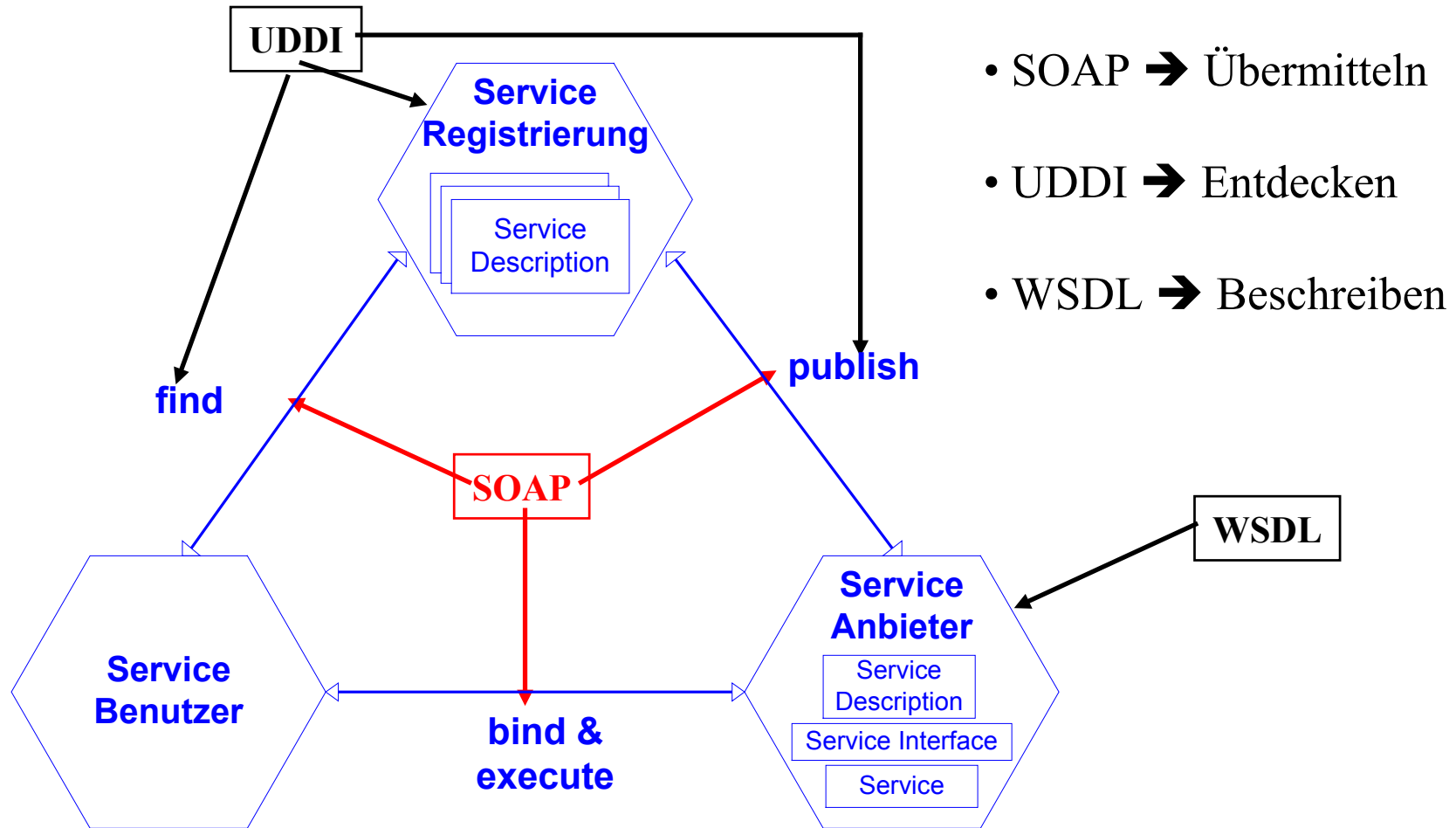
8. WSAH - als Standard???

- TCP / IP – Transfer Control Protocol / Internet Protocol
- HTTP – Hypertext Transfer Protocol
- XML – Extensible Markup Language
- SOAP – Simple Object Access Protocol
- WSDL – Web Service Description Language
- UDDI – Universal Description Discovery and Integration Protocol
- WS..... – Web Service... (alles mögliche)
- BPEL4WS – Business Process Execution Language for Web Services

... und das war erst der Anfang

➔ WSAH – „*Web Service Acronym Hell*“

Überblick



9.1. SOAP - Definition

- Definition (W3C): *„einfacher [...] Mechanismus für den Austausch strukturierter und typisierter Informationen zwischen Rechnern in einer dezentralen verteilten Umgebung auf Basis von XML“*
- alt. Definition: *„ein XML-/HTTP-basiertes Protokoll für den [plattformunabhängigen] Zugriff auf Dienste, Objekte und Server“*

9.2. SOAP - Geschichte

- 1998: Entwicklung von SOAP durch Microsoft u.a.
- November 1999: eine erste Version von SOAP von MS über die Internet Engineering Task Force (IETF) veröffentlicht
- Mai 2000: SOAP Spezifikation in der Version 1.1 beim W3C eingereicht; (Status einer „W3C Note“ → kein Standard, sondern Diskussionspapier)
- September 2000: beim W3C formiert sich eine Arbeitsgruppe, die ein XML-basiertes Protokoll zur Kopplung von Applikationen über das Web definieren soll → SOAP 1.2
- Juni 2003: SOAP Version 1.2 wird Empfehlung des W3C → SOAP ist jetzt ein Eigenname

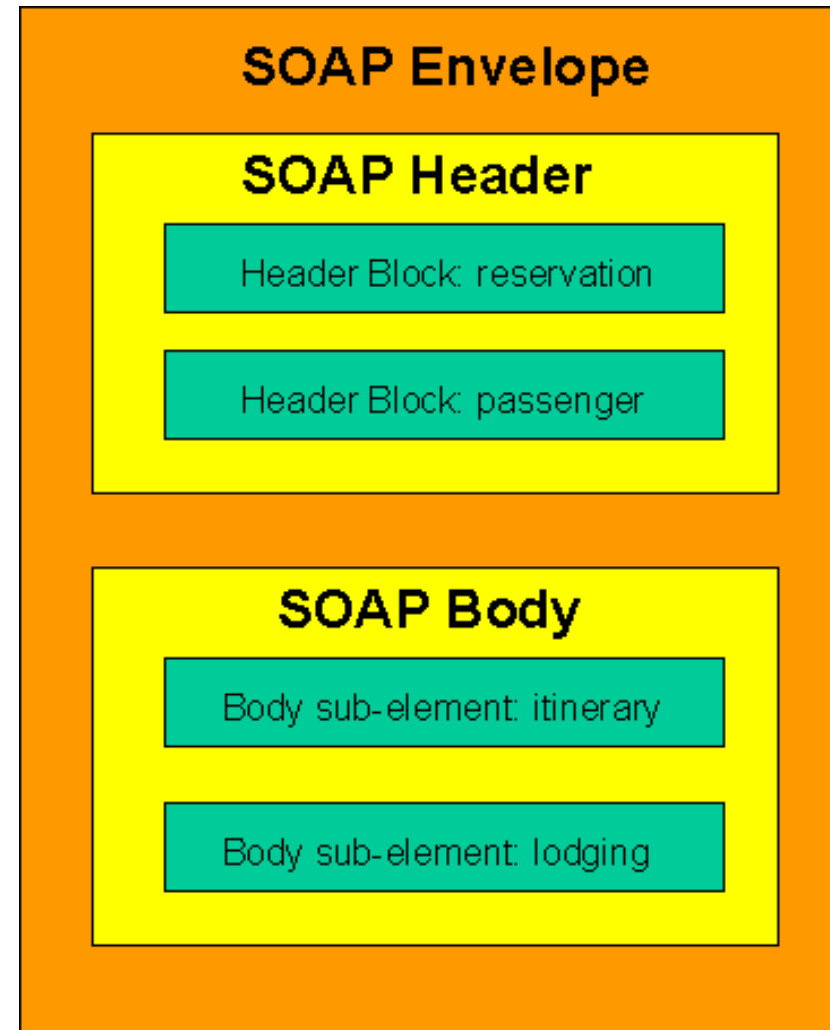
9.3. SOAP - allg. Aufbau

- die SOAP Spezifikation definiert eine XML Grammatik, mit der strukturierte und typisierte Informationen zu einer Nachricht zusammengefasst werden können
- eine SOAP Nachricht ist ein XML-Dokument, auch Umschlag („Envelope“) genannt, das einen optionalen Kopfinformations-Teil („Header“) und den eigentlichen Inhaltsteil („Body“) enthält
- eine SOAP Nachricht ist unidirektionale Übermittlung von einem Sender zu einem Empfänger (auch über mehrere Instanzen)
- mehrere Nachrichten können aber über mehrere Instanzen zu komplexen Interaktionsmustern komponiert werden
- das folgende Beispiel zeigt eine SOAP-Nachricht bei einer Reisebuchung

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger xmlns:n="http://mycompany.example.com/employees"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:name>Åke Jógvan Øyvind</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itinerary
      xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>New York</p:departing>
        <p:arriving>Los Angeles</p:arriving>
        <p:departureDate>2001-12-14</p:departureDate>
        <p:departureTime>late afternoon</p:departureTime>
        <p:seatPreference>aisle</p:seatPreference>
      </p:departure>
      <p:return>
        <p:departing>Los Angeles</p:departing>
        <p:arriving>New York</p:arriving>
        <p:departureDate>2001-12-20</p:departureDate>
        <p:departureTime>mid-morning</p:departureTime>
        <p:seatPreference/>
      </p:return>
    </p:itinerary>
    <q:lodging
      xmlns:q="http://travelcompany.example.org/reservation/hotels">
      <q:preference>none</q:preference>
    </q:lodging>
  </env:Body>
</env:Envelope>
```

9.4. SOAP - Beispiel 1

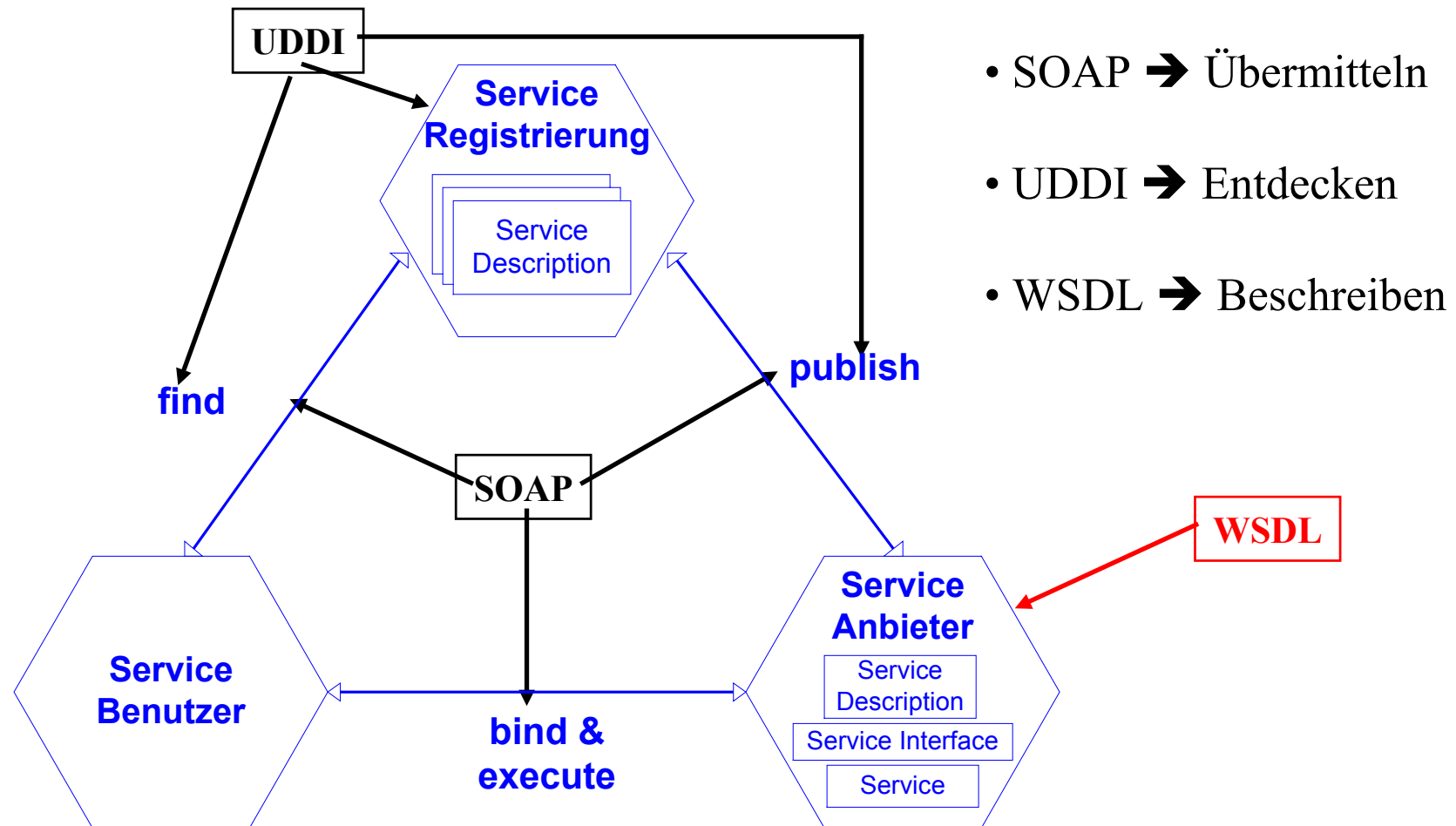
- der Header ist optional – enthält z.B. Verarbeitungsinformationen, die von eventuellen Zwischenstationen bearbeitet werden müssen
- der Body enthält die zu übermittelnde Information die schließlich vom ultimativen Empfänger verwertet werden
- die Rollen und die Reaktionen der einzelnen Akteure muss noch genauer spezifiziert werden – durch Entwickler
- Fehlerquellen müssen genau bedacht werden



9.5. SOAP - Fazit

- ebenso einfach kann Empfänger jetzt antworten
- Remote Procedures Call's möglich, d.h. Funktionsaufrufe für das Zielsystems können mit ins SOAP-Dokument aufgenommen wird
- Fehlerbehandlung möglich, d.h. es gibt standard. Fehlermeldungen
- Nachrichtenpfad über mehrere Zwischenstationen üblich
- neben HTTP kann SOAP auch über TCP, SMTP oder UDP betrieben werden
- gibt nur an, wie eine Transmission strukturiert werden soll
- gibt nicht deren Semantik an, bzw. deren Verarbeitungsprozess
- sicherheitsrelevante Aspekte wurden separat entwickelt

Überblick



10.1. WSDL - Definition

- Definition: „*WSDL beschreibt auf Basis von XML, wie ein Web Service technisch funktioniert*“
- WSDL gibt z.B. die Internetadresse, Port(s), Methodennamen, Datentypen, Bindings u.s.w. an
- WSDL ist noch kein verabschiedeter Standard (de-facto), hat Status eines „Working Drafts“

10.2. WSDL - Aufbau

- typischer Aufbau eines WSDL-Dokuments:

```
<definitions>  
  <dokumentation></dokumentation>  
  <import> </import>  
  <types> </types>  
  <message> </message>  
  <portType> </portType>  
  <serviceType> </serviceType>  
  <binding> </binding>  
  
  <service> </service>  
</definitions>
```

10.3. WSDL - Aufbau

- <definitions>
 - Wurzelement
 - definiert Namespaces, die im Dokument gebraucht werden

- <dokumentation>
 - optional
 - enthält menschen-lesbare Beschreibung des WS

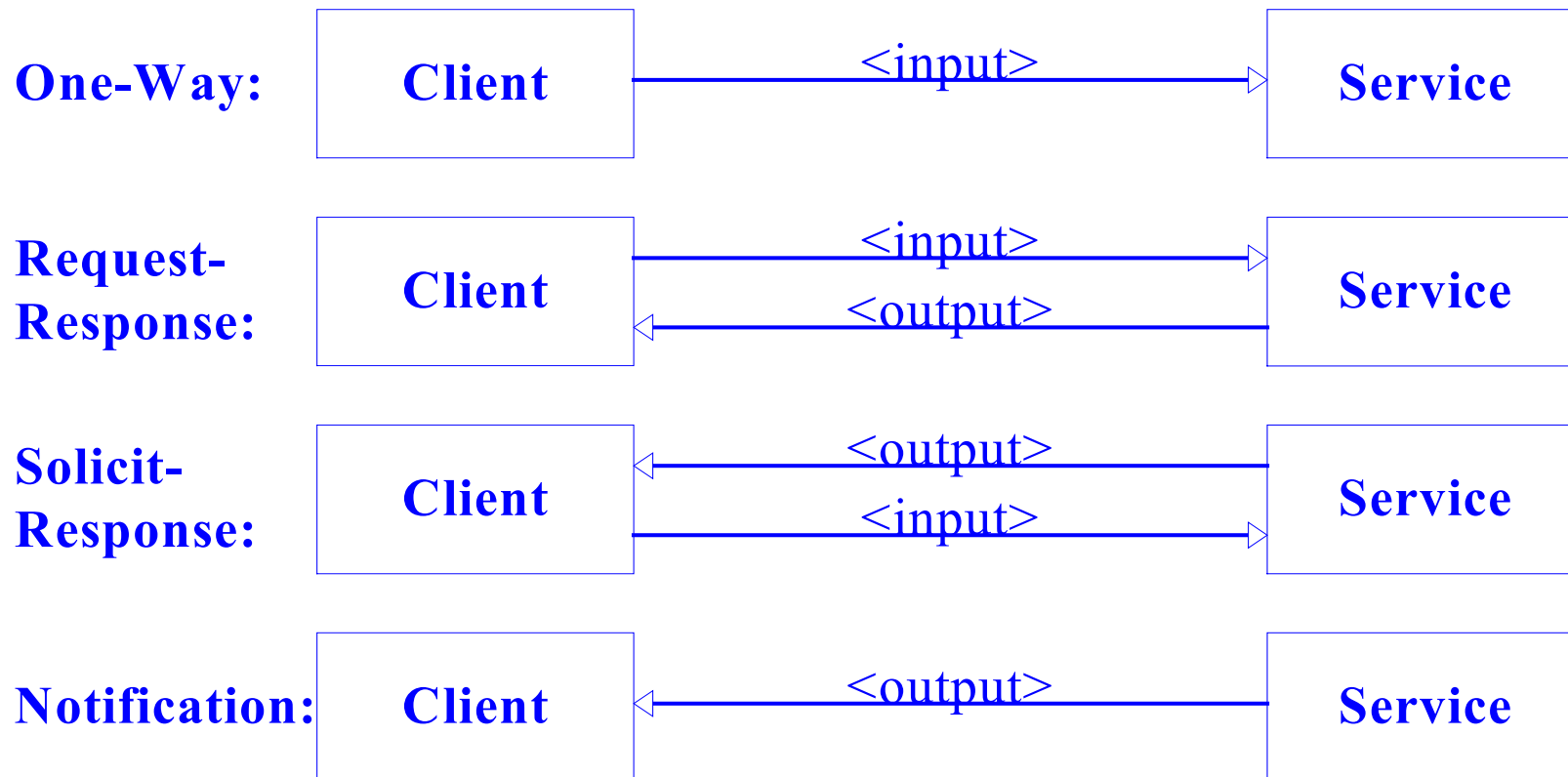
- <import>
 - optional
 - gibt importierbare(s) Dokument(e) an (XML Schema Def., WSDL-Dokument)

10.4. WSDL - Aufbau

- <types>
 - optional (bei Verwendung einfacher Datentypen)
 - beschreibt komplexe Datentypen
 - einfache Datentypen werden ja bereits von XML Schema offeriert (z.B.: string, integer, float, double, date,...)
- <message>
 - beschreibt abstrakt die Nachrichten, die zwischen Client und Service ausgetauscht werden könnten
 - besteht aus verschiedenen Parts (Name und Typ [type])
- <portType>
 - definiert „Interfaces“ des WS

10.4.1. portType Arten

- jede Kommunikationsart hat spezifische Fehlermeldungen



10.5. WSDL - Aufbau

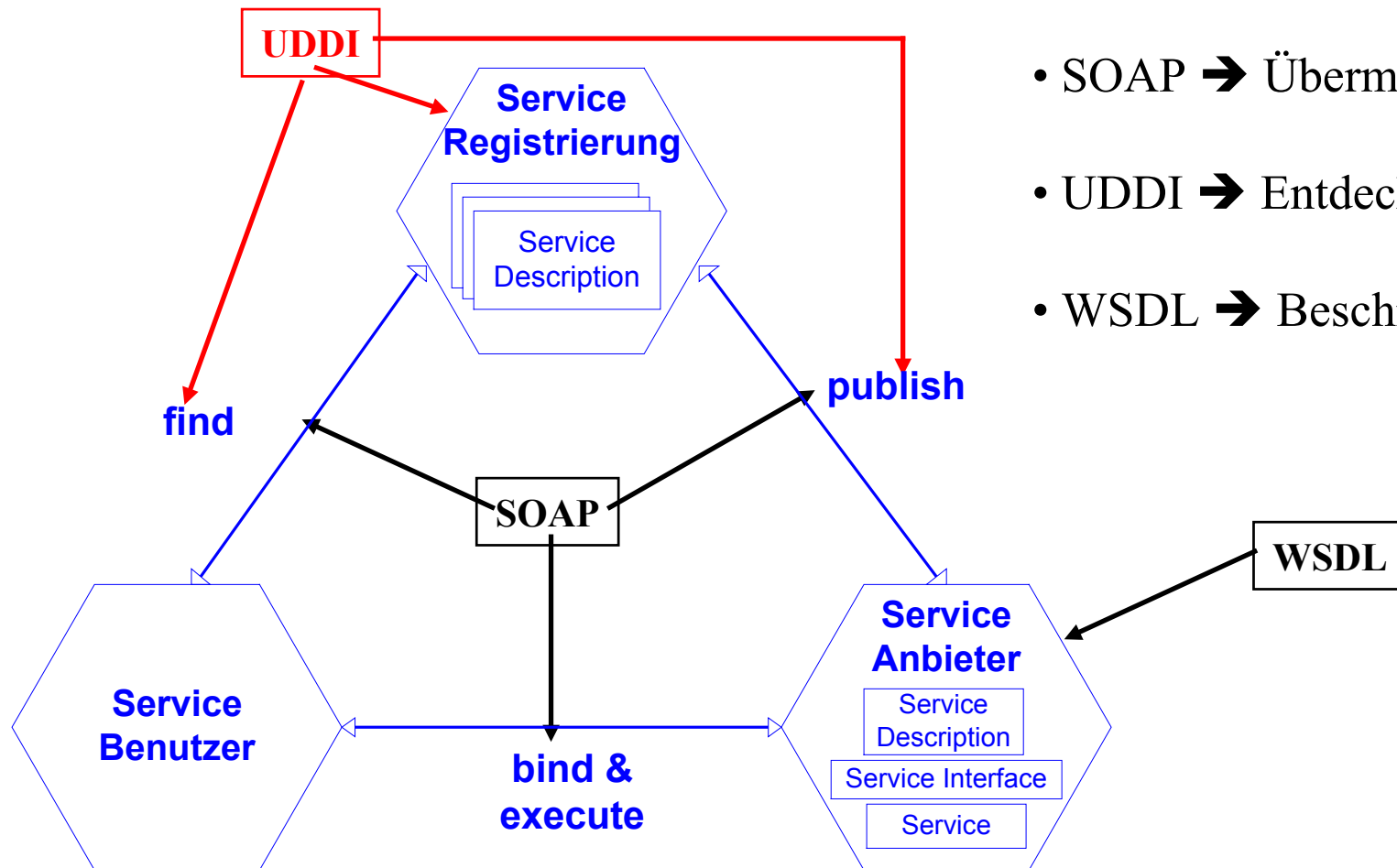
- <binding>
 - definiert Nachrichtenformat und Übertragungsprotokoll(e) (SOAP 1.2, HTTP 1.1 GET/ POST, MIME,...)
 - für Nachrichten und Operationen über einen bestimmten portType

- <service>
 - fügt mehrere Ports zusammen

10.6. WSDL - Fazit

- beschreibt nur Syntax, nicht Semantik eines WS
- kann nicht mehrere WS zusammenfassen
- kann keine komplexeren Interaktionen außer Request/Response beschreiben
- ist Programmiersprachen-, Plattform-unabhängig
- ist in guten Tools bereits integriert
- Erweiterungen → WSCI, WSCL, WSFL, XLANG

Überblick



- SOAP → Übermitteln
- UDDI → Entdecken
- WSDL → Beschreiben

11.1. UDDI - Überblick

- universelles Webverzeichnis (Businessverzeichnis)
- soll Verbindung zwischen Käufer und Verkäufer herstellen
- Initiative von MS, IBM, (Ariba) – mittlerweile auch Dell, SUN, HP, Intel, usw.
- soll beschreibender sein als andere WV → Firmen können sich besser repräsentieren
→ Suche ist effizienter
- UDDI besteht aus drei Hauptbestandteilen: der UDDI Business Registry, dem UDDI-Framework und SOAP-Nachrichten für RPC-Aufrufe von Web-Services
- alle Bestandteile basieren auf Standards (XML, HTTP, DNS)

11.2. UDDI – Business Registry

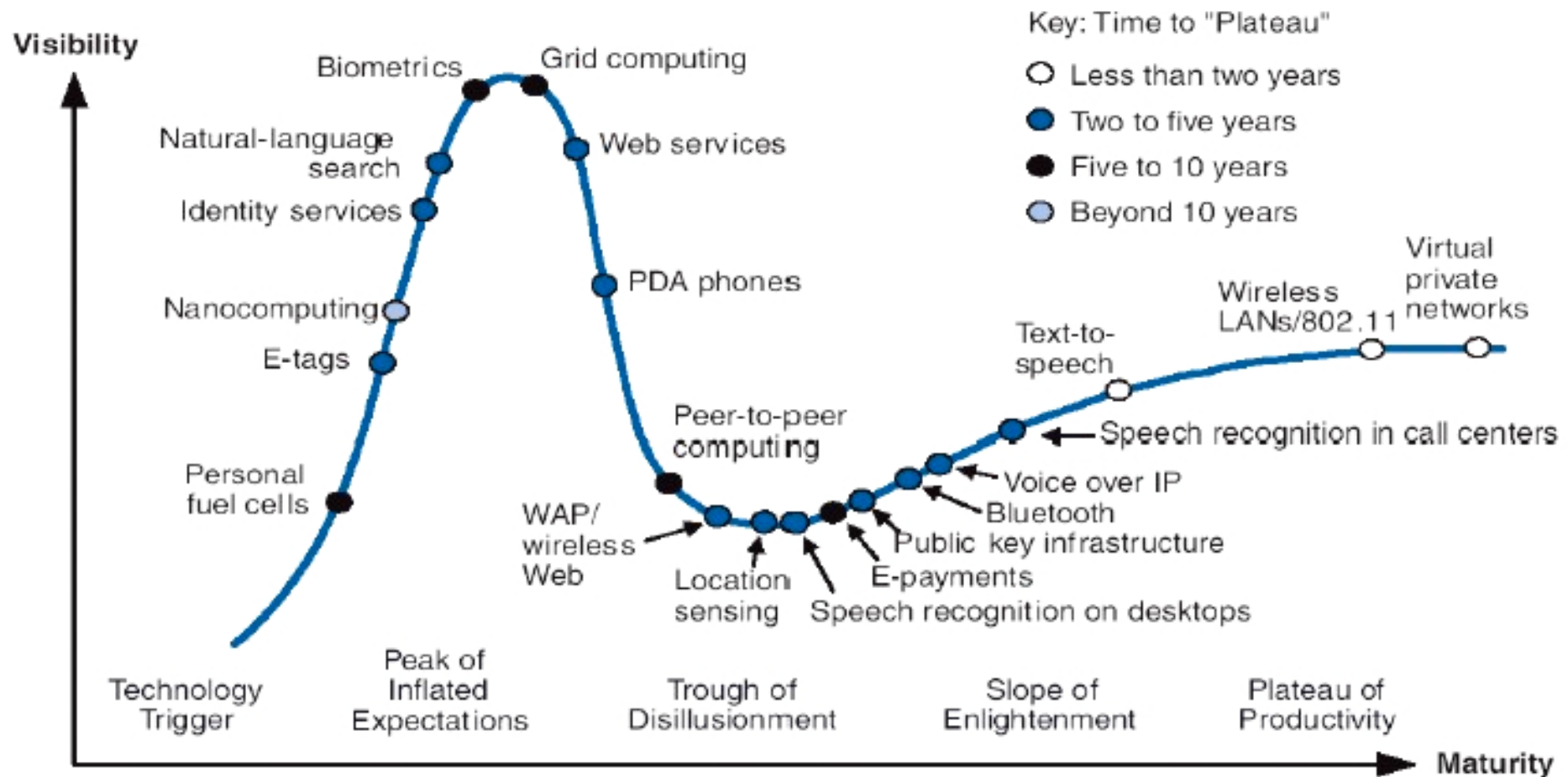
- ist ein zentraler Bestandteil der UDDI-Architektur
- eine UDDI Business Registry ist ein Service-Registrierungsdienst:
 - White Pages: Adressen und Kontaktinformationen der Firma
 - Yellow Pages: Kategorisierungen der Dienstleister.
 - Green Pages: technische Beschreibungen der zur Verfügung gestellten Dienste über Web und Referenzen auf Dateien oder URLs.
- die Einträge einer UDDI Business Registry sind mittels WSDL beschrieben.
- alle eingetragenen Daten werden global zur Verfügung gestellt, da die einzelnen UDDI-Nodes eine automatische Replikation durchführen

12. WS-Welt

- Global Player
 - Microsoft
 - IBM
 - SUN
 - Oracle
 - Intel
 - HP
 - Bea
 - ...
- Allgemein akzeptiert sind untere Schichten:
 - XML, SOAP, WSDL, UDDI
- **Neu:** Web Service Interoperability Organization WS-I
- ?Standards?
 - XLang
 - WSFL
 - WSCI
 - BPML
 - BPSS
 - ebXML
 - bizTalk
 - BPEL4WS
 - ...
- Systeme
 - .NET
 - WebSphere
 - Sun ONE
 - ...
- Organisationen
 - W3C
 - OASIS
 - BPMI
 - Microsoft, IBM & BEA Systems
 - ...

13. Gegenwärtiger Stand

The 2002 Hype Cycle of Emerging Technologies



Source: Gartner Research

LAN Local-area network
IP Internet Protocol

PDA Personal digital assistant
WAP Wireless Application Protocol

14. Fazit

- **Pro:**

- flexibel, unabhängig
- „neuer Ansatz“ von Web-Diensten
- alle großen Firmen sind beteiligt (Allianzen)
- mächtig durch effiziente Middleware
- Komponentenkonzept
 - vollständig virtuelle Unternehmen
 - *The Web's next Revolution (Evolution)*

- **Kontra:**

- Unsaubere Standards
- WSAH
- Grabenkämpfe um Kleinigkeiten
- Ignoranz gegenüber theoretischen Erkenntnissen aus: Datenbanktheorie,
- Workflow-Theorie,.....
 - Wozu sind die Hochschulen da?

Noch Fragen ???