

DAIMLERCHRYSLER

Systemintegration und Web-Services mit SOAP

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

mario.jeckle@daimlerchrysler.com

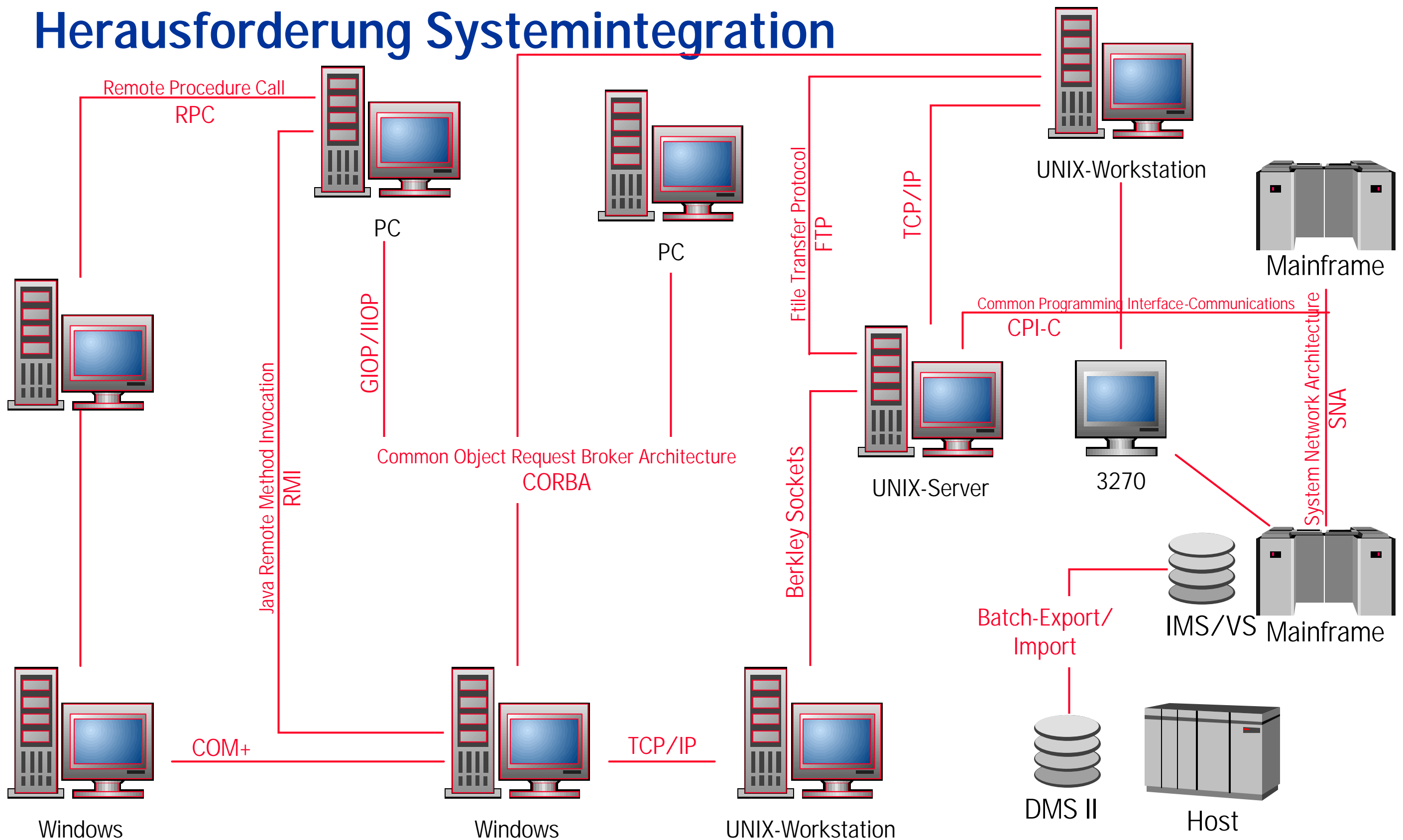
mario@jeckle.de

www.jeckle.de

Gliederung

- **Systemintegration**
 - Herausforderungen
 - Lösungsansätze
- **SOAP**
 - Grundidee
 - Entwicklung und Einsatz
- **WSDL**
 - Grundidee
 - Beispiel
- **UDDI**
 - Grundidee
 - Zusammenwirken mit WSDL

Herausforderung Systemintegration

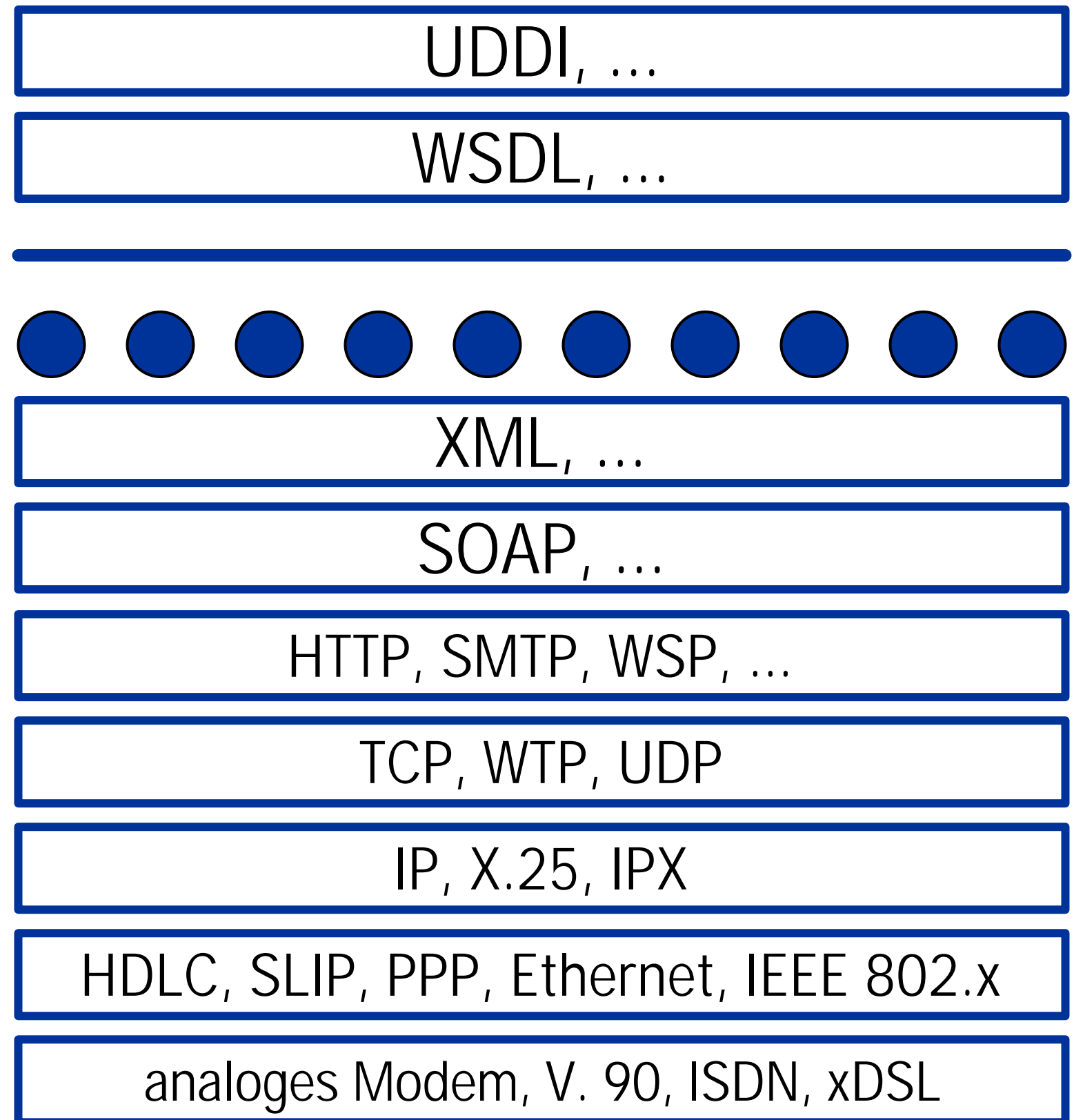


Systemintegration

- **Problem:** Heterogenität und (mangelnde) Konnektivität
 - Applikationsobjekte und -schemata
 - Netzwerkprotokolle und -schnittstellen
 - Paketierung und Verteilung
- **Hintergrund:** Inhalte und Protokolle
 - Datenhaltung (relational, objektorientiert, dateibasiert, ...)
 - Datenrepräsentation (ASCII, CDR, ASN.1, ...)
 - Datentransport (TCP/IP, Sockets, Messaging, ...)
- **Lösungsmöglichkeit:** Standards und Generizität
 - Schnittstellen und entfernte Funktionsaufrufe
 - Gemeinsame Schemata und verteilte Objekte
 - Web-Techniken und XML

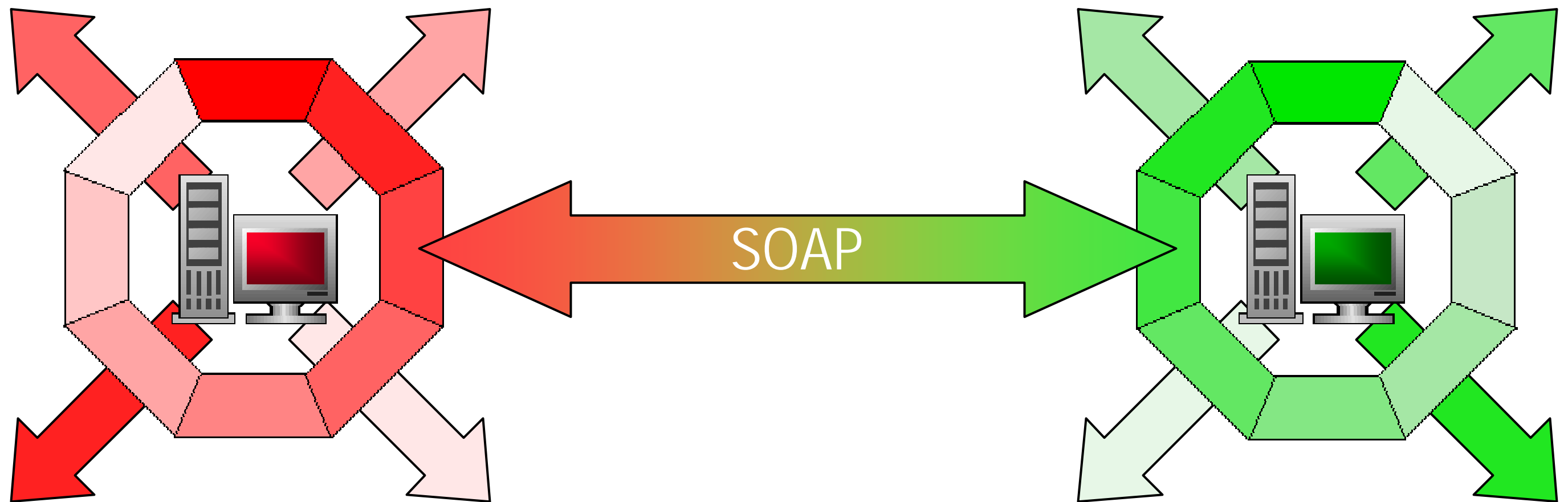
Systemintegration

- Verzeichnisschicht (*Directory Layer*)
- Schnittstellenschicht (*Interface Layer*)
- Applikationsschicht (*Application Layer*)
- Dienstschicht (*Service Layer*)
- Applikationsschicht (*Application Layer*)
- Darstellungsschicht (*Presentation Layer*)
- Sitzungsschicht (*Session Layer*)
- Transportschicht (*Transport Layer*)
- Netzschicht (*Network Layer*)
- Sicherungsschicht (*Data Link Layer*)
- Bitübertragungsschicht (*Physical Layer*)



SOAP als Lösungsansatz in der Systemintegration

- Überbrückung der Heterogenität durch Plattform- und Programmiersprachenunabhängigen Kommunikationsmechanismus



Warum SOAP?

- RPCs (UNIX) erfordern binärkompatible Implementierung der Kommunikationspartner
- CORBA erfordert (komplexe) Infrastruktur (ORB!)
- RMI benötigt in Java implementierte Kommunikationspartner
- DCOM benötigt Windows als Plattform
- SOAP
 - system- und plattformneutral
 - paradigmenneutral
 - programmiersprachenneutral

Von der Systemintegration zu Web-Services

● Systemintegration

- Kopplung technischer Systeme
- (zumeist) Programmierparadigmenspezifisch
- Technische Realisierung der Kommunikation sekundär
- Langfristige Kooperation

● Web-Service

- Dienstleistung
- Paradigmenunabhängigkeit
- Kommunikation von primärem Interesse
- Ad hoc Kollaboration

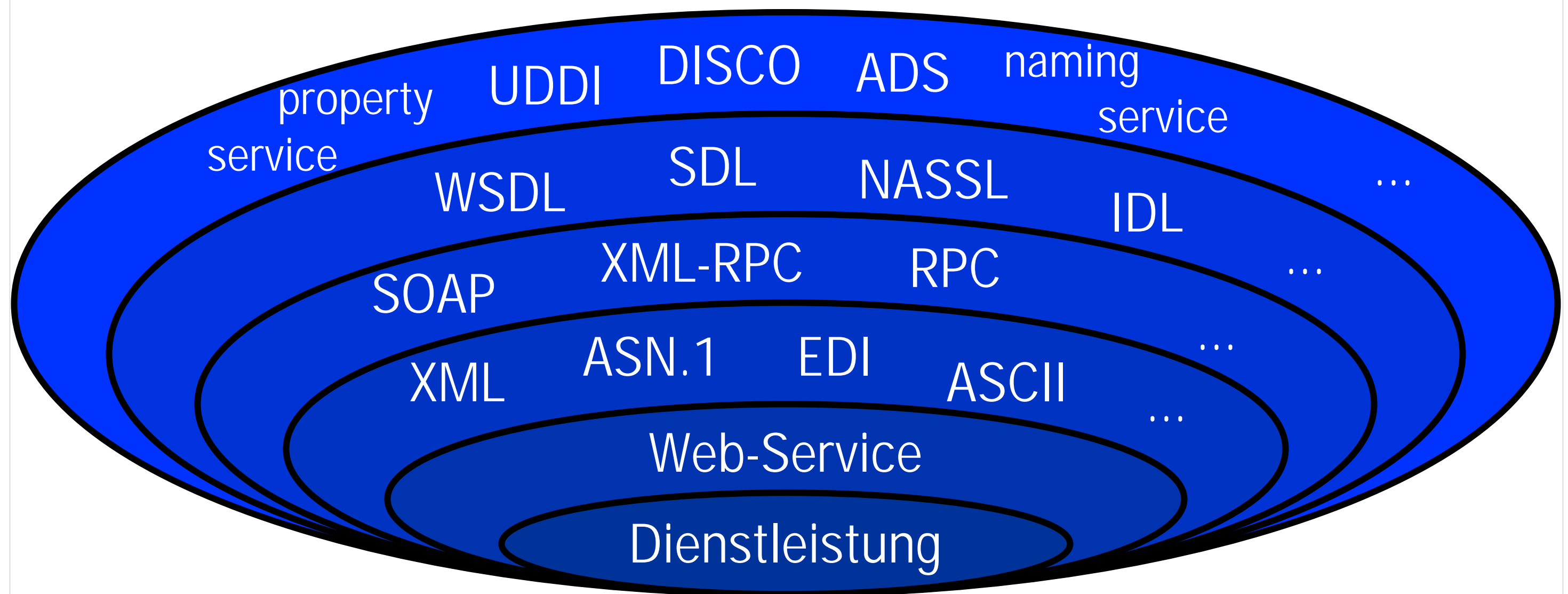
Definition: Web-Services -- techno-ökonomisch

- Charakteristische Merkmale:
 - Immaterialität (keine Produktion physischer Güter)
 - Ephemer (nicht lagerfähig)
 - Standortabhängigkeit (nicht transportfähig)
 - Produktions-Konsumptions-Synchronität (keine Entkopplung zwischen Produzent und Konsument)



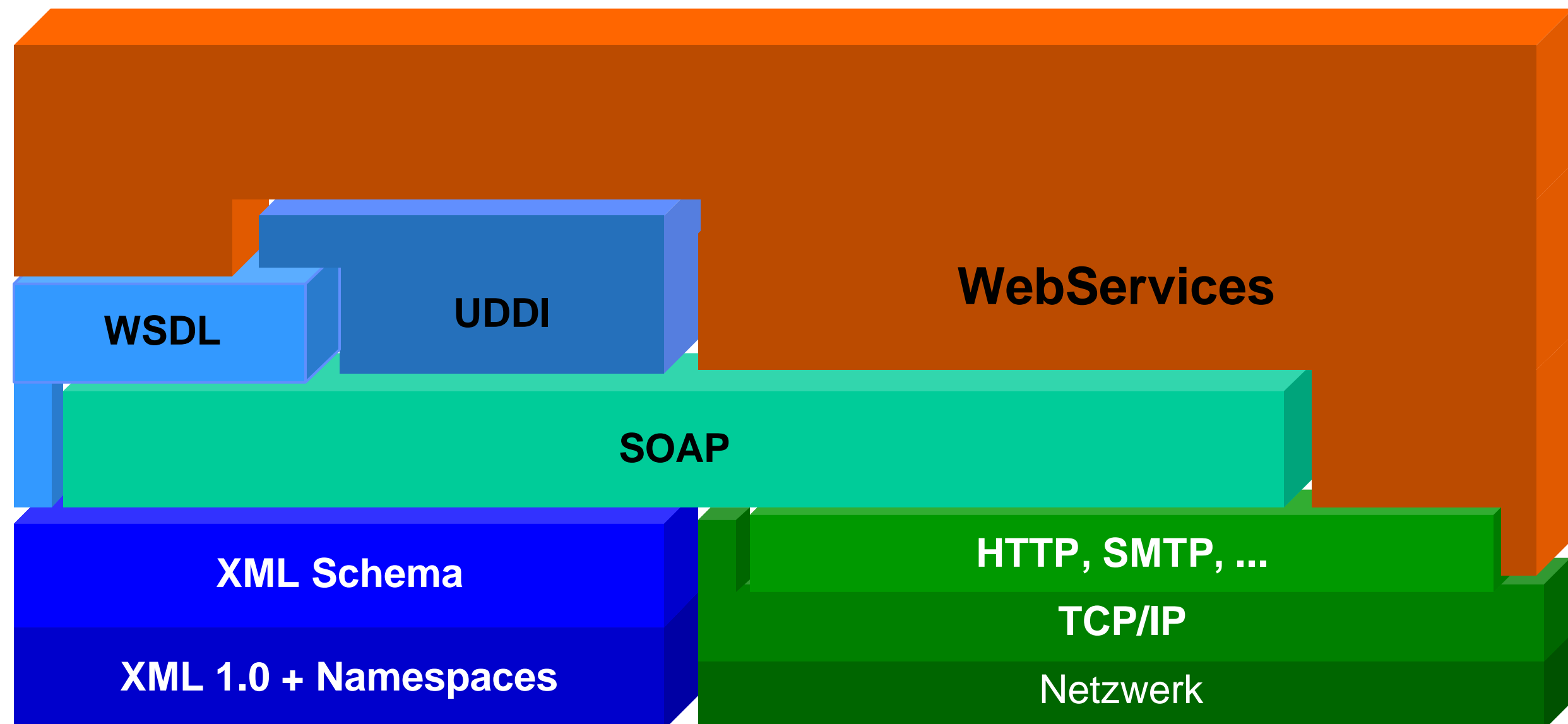
Definition: Web-Services -- techno-ökonomisch

- Web-Service = Dienstleistung über Web



Definition: Web-Services -- technisch

- Web Service setzt auf Internet-Protokollen auf
- Web Service *kann* auf XML-Techniken aufsetzen

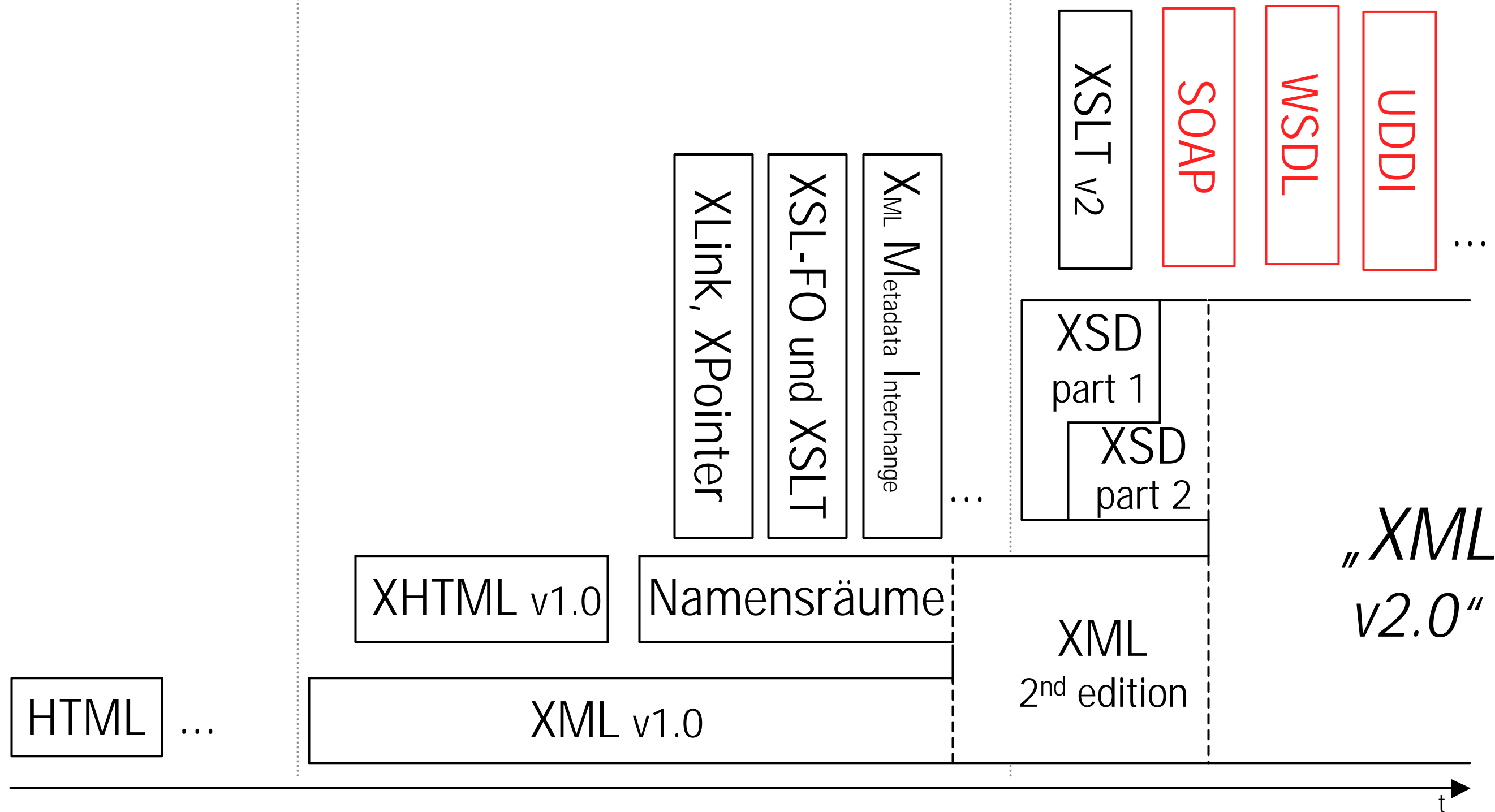


Web-Services: Die Technik hinter den Kulissen -- XML

Web-Prähistorie

1. Generation

2. Generation



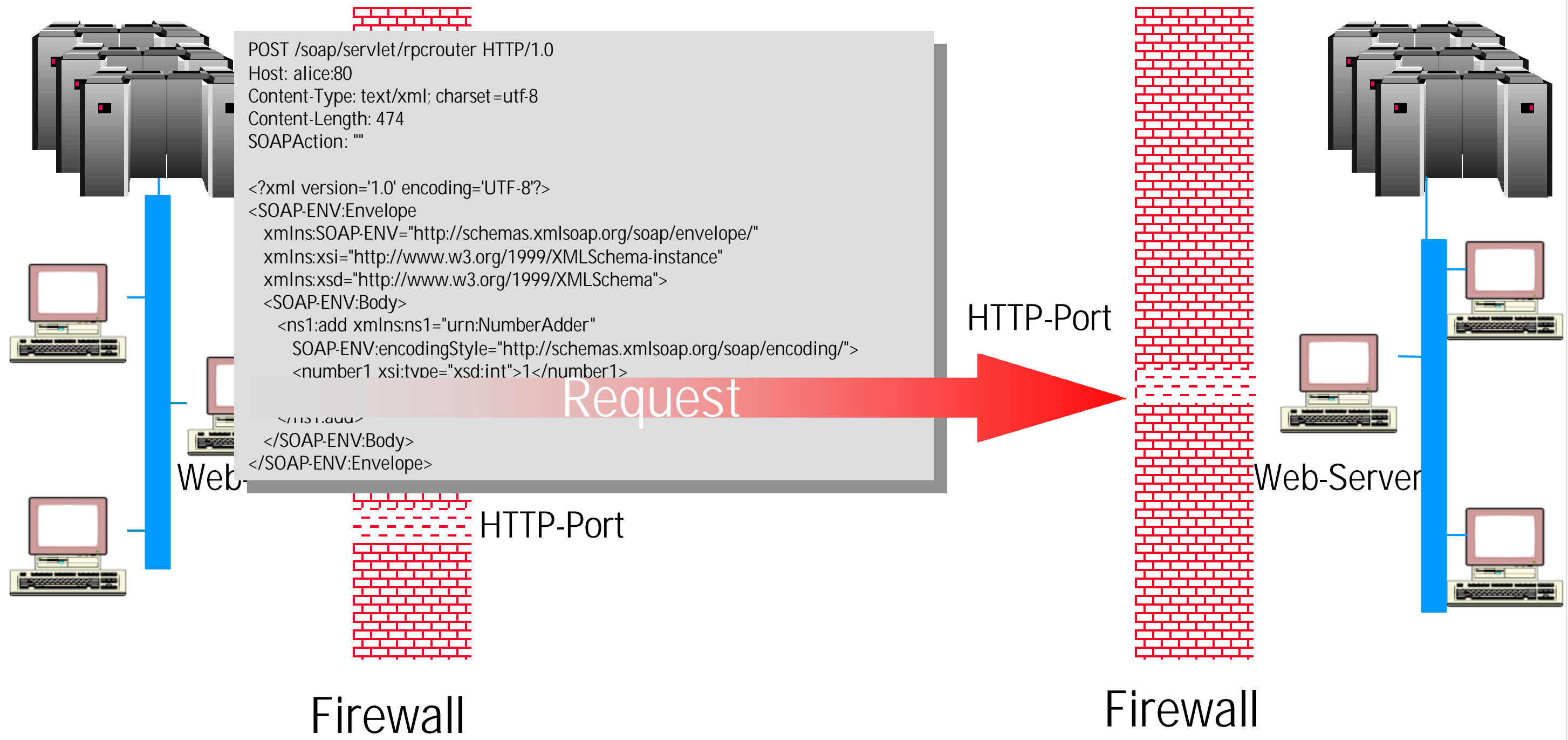
Web-Services: Die Technik hinter den Kulissen -- XML

- XML ist eine Methode, um strukturierte Daten in einer Textdatei darzustellen
- XML ist Text, aber nicht zum Lesen
- XML ist ausführlich, was aber kein Problem darstellt
- XML sieht fast aus wie HTML, ist aber kein HTML
- XML ist eine Familie von Techniken
- XML ist eine Untermenge des ISO-Standards SGML
- XML ist international und unterstützt beliebige Alphabete (Unicode)
- XML ist neu, aber nicht so neu
- XML ist lizenzfrei, plattform- und herstellerunabhängig, und gut unterstützt
- XML ist inzwischen weit verbreitet

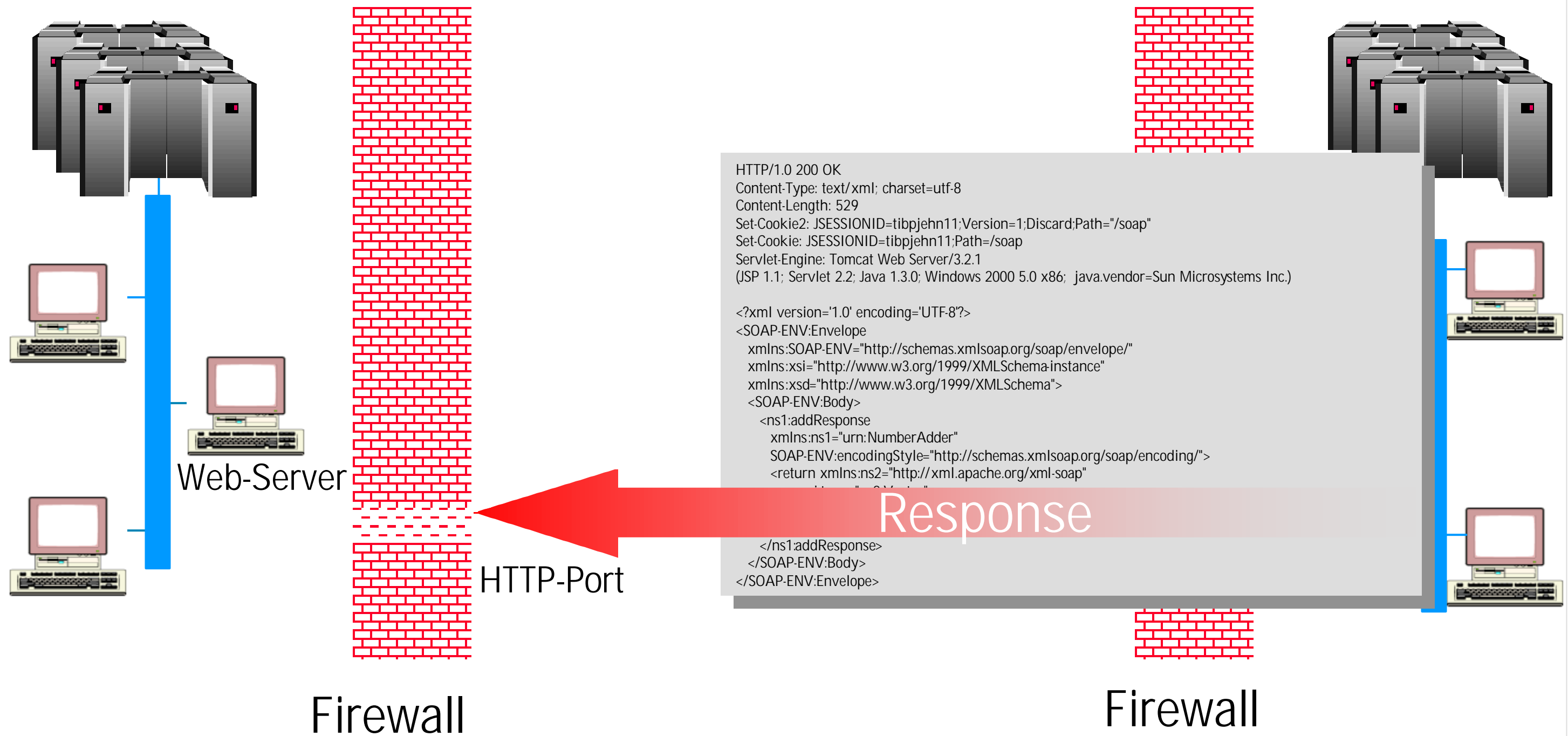
Web-Services: Die Technik hinter den Kulissen – SOAP

- Grundidee: Entfernte Methodenaufrufe und Nachrichtenaustausch
- XML-Darstellung beliebiger Inhalte
- Trennung zwischen Nutzinformation und Metadaten
- Ursprüngliche Entwicklung als *XML-RPC* (HTTP-basiert, asynchron)
- Weiterentwicklung durch DevelopMentor, IBM, Microsoft, Lotus und UserLand Software als *Simple Object Access Protocol (SOAP)*
- Seit *SOAP/1.1* synchron und (Transport-)Protokollunabhängig
- Als SOAP v1.2 derzeit durch eine W3C-Arbeitsgruppe im Standardisierungsprozeß

SOAP-Grundidee



SOAP-Grundidee



SOAP-Technik

Transportprotokoll (z.B. HTTP, SMTP, TCP, UDP, ...)

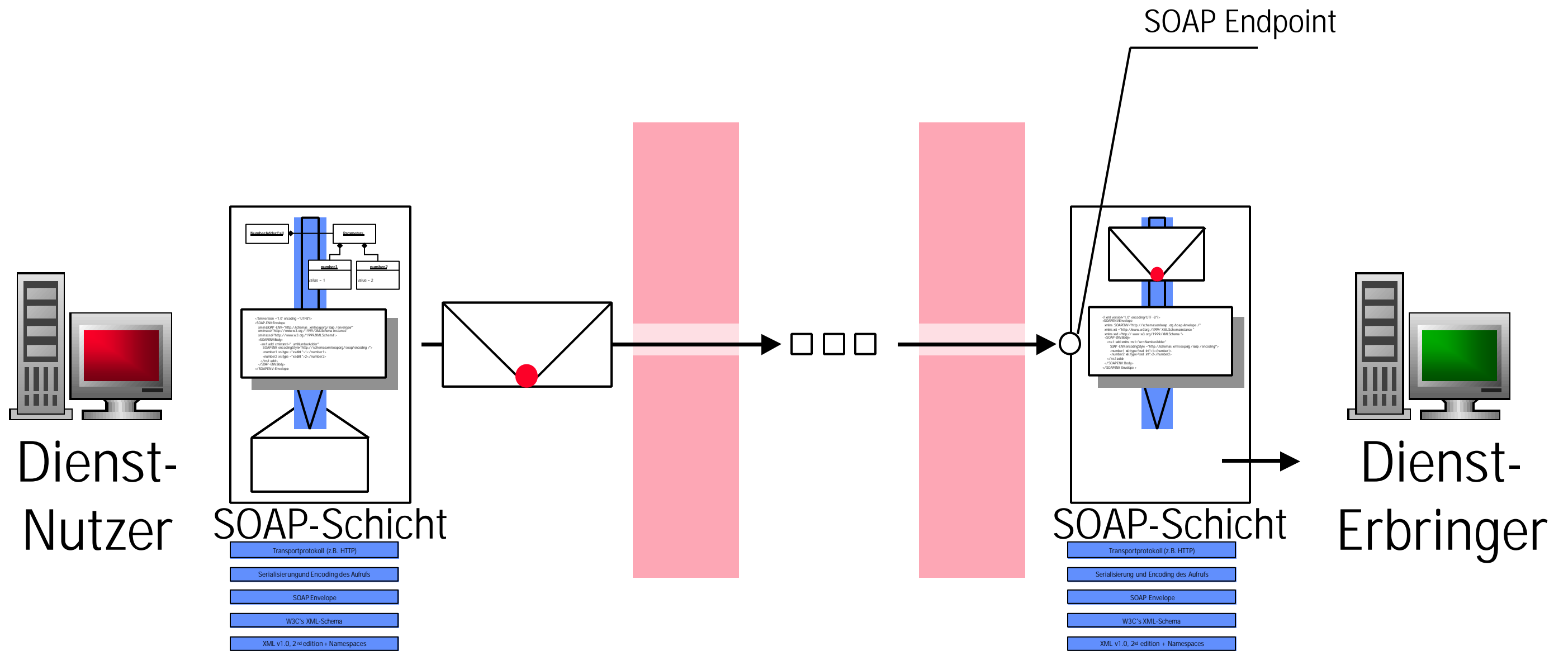
Serialisierung und Encoding des Aufrufs

SOAP Envelope

W3C's XML-Schema

XML v1.0, 2nd edition + Namespaces

Ablauf eines SOAP-Aufrufs



SOAP-Einsatzszenarien

- *Fire-and-forget*
(Unbestätigtes Versenden einer Nachricht)
 - z.B. Newsservice, Börsenticker, Wetterdaten, ...
- *Request-Response*
(Bestätigtes Versenden einer Nachricht mit Warten auf Antwort)
 - z.B. Synchroner Nachrichtenübermittlung
- *Remote Procedure Call*
(Synchroner Aufruf einer entfernten Methode)
 - z.B. „klassische“ Verteilungsszenarien
- *Caching*
(Transparente Zwischenspeicherung)
 - z.B. Vorhalten von Ergebnissen

Ausblick: SOAP v1.2

- Breit (über 30 Arbeitsgruppenmitglieder) getragener Standard
- Frühe Umsetzung
- (Teilweise deutlich) präzisere Spezifikation
 - SOAP-Action Header
 - MIME-Type bei HTTP-Binding
 - mustUnderstand
 - Status- und Fehlercodes
 - XML-Namensräume
 - ...
- Engere Integration in W3C-XML-Standards (z.B. InfoSet)

Web-Services: Die Technik hinter den Kulissen -- WSDL

- WSDL = Web Service Description Language
 - Abstrakte Beschreibung von Web-Services
 - Definition des Nachrichtenverkehrs
 - Definition der verfügbaren Operationen
 - Bindung an konkrete Netzprotokolle



Web-Services: Die Technik hinter den Kulissen -- WSDL

```
<message name="...Request">  
  <port name="..." type="..."/>  
</message>
```

```
<message name="...Response">  
  <port name="..." type="..."/>  
</message>
```

```
<portType name="...Service">  
  <operation name="...">  
    <input message="..Request"/>  
    <output message="...Response"/>  
  </operation>  
</portType>
```

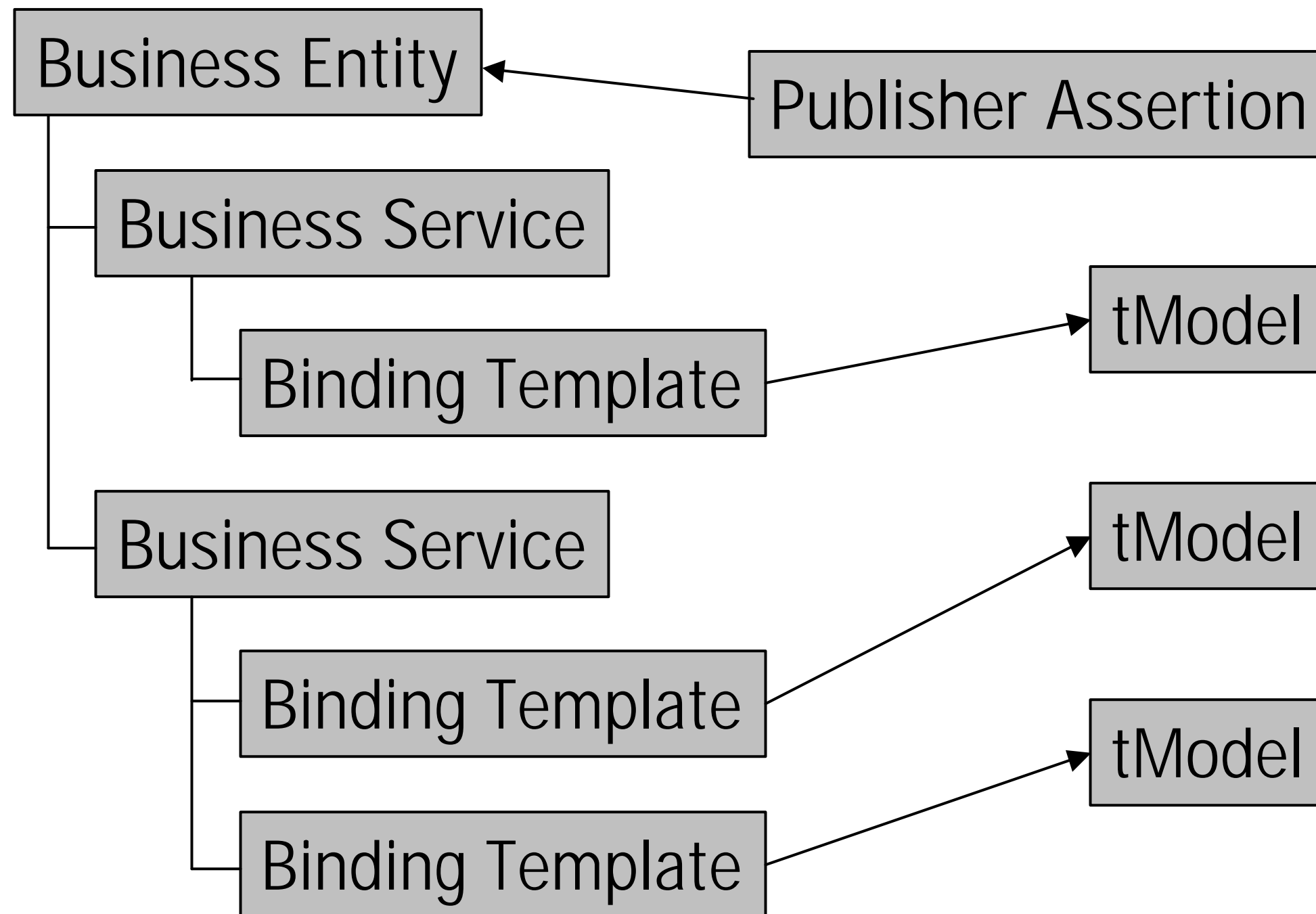
```
<portType name="...Service">  
  <operation name="...">  
    <input message="..Request"/>  
    <output message="...Response"/>  
  </operation>  
</portType>
```

```
<binding name="...Service" type="...">  
  <soap:binding style="rpc" .../>  
  <operation name="...">  
    <soap:operation soapAction="..."/>  
    <input>...</input>  
    <output>...</output>
```

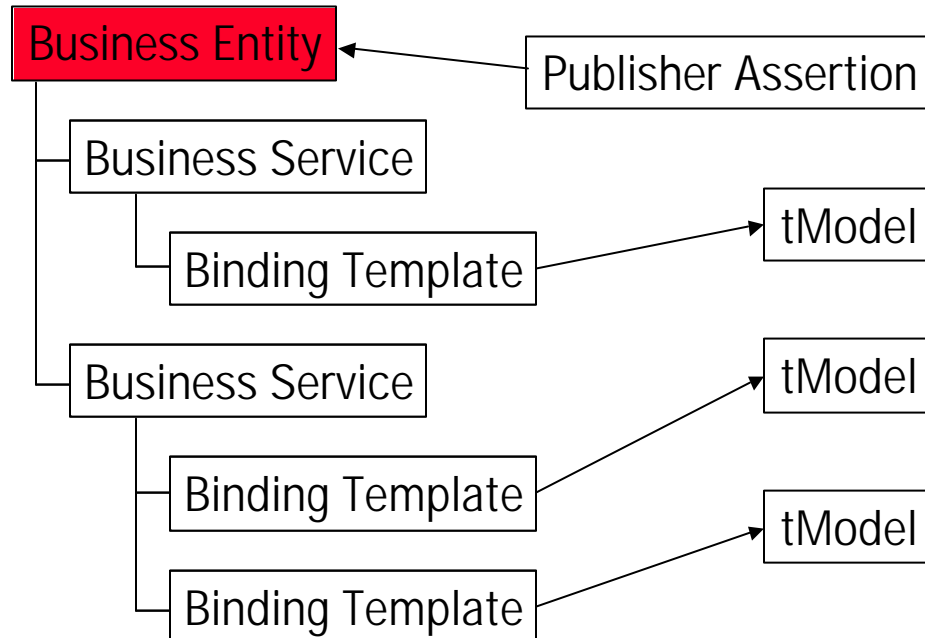
...

Web-Services: Die Technik hinter den Kulissen -- UDDI

- UDDI = Universal Description, Discovery and Integration



Web-Services: Die Technik hinter den Kulissen -- UDDI

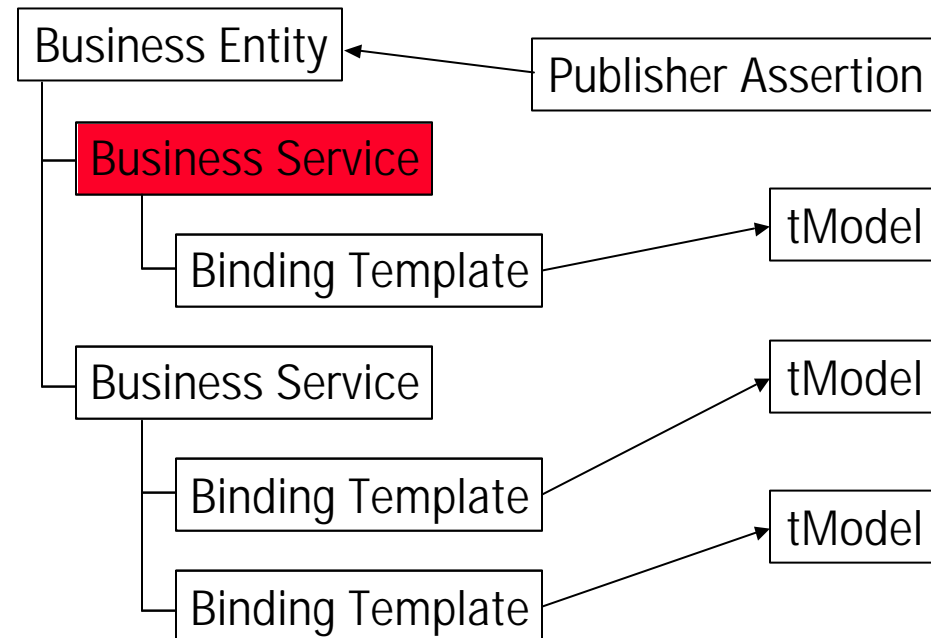


<u>Business Entity</u>
businessKey = 6a0adcbe-58b9-46b3-99b3-dd44b81c6a93 name = Carl Inc. categoryBag = {(NAICS,811192)}

● Business Entity

- *BusinessKey* – eindeutige Identifikation des Dienstanbieters
- *Name* – Lesbare Klartextbenennung (evtl. mehrsprachig)
- *Description* – Lesbare Klartextbeschreibung (evtl. mehrsprachig)
- *Contacts* – Lesbare Kontaktdetails (Telephon, Mail, etc.)
- *Category Bag* – Namen-Wert-Paare zur Einordnung des Dienstanbieters

Web-Services: Die Technik hinter den Kulissen -- UDDI



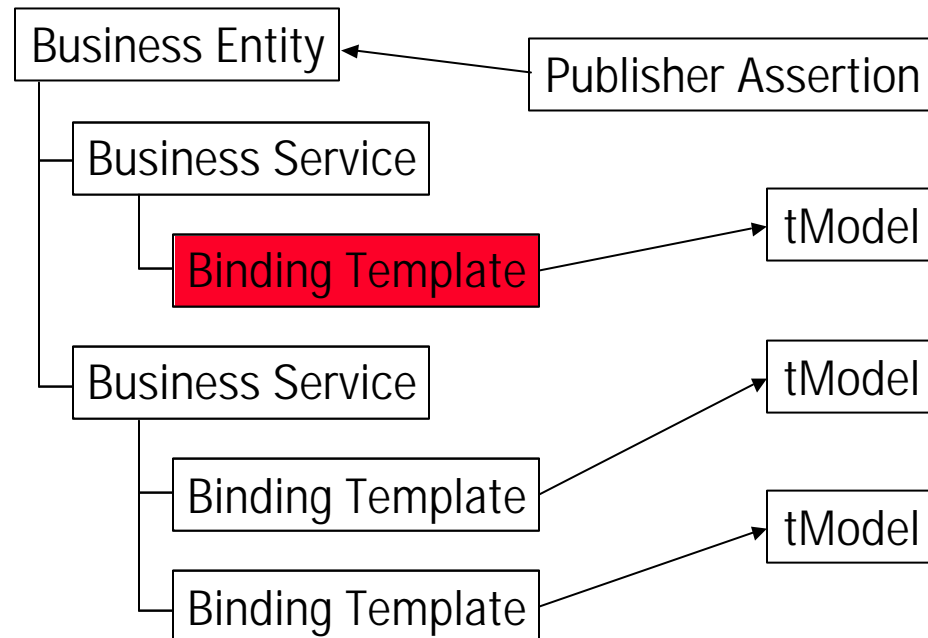
Business Service

```
serviceKey = 8adf1960-4bc3-49ec-bf4d-3db06cfab2f9  
name = washing and cleaning of various car
```

● Business Service

- *Service Key* – eindeutige Identifikation des Dienstes
- *Name* – Lesbare Klartextbenennung (evtl. mehrsprachig)
- *Description* – Lesbare Klartextbeschreibung (evtl. mehrsprachig)

Web-Services: Die Technik hinter den Kulissen -- UDDI



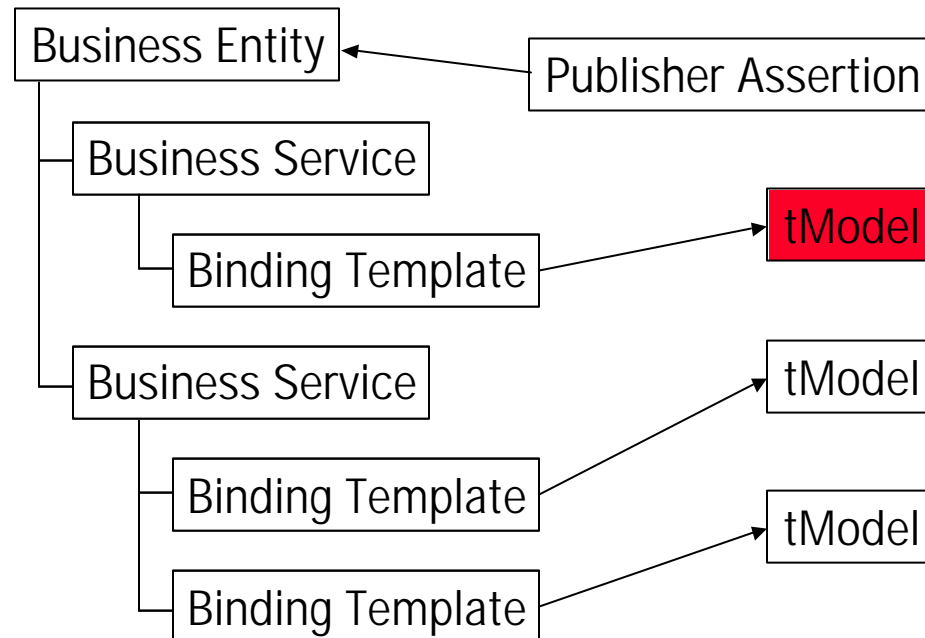
Binding Template

```
bindingKey = 469ae6a1-4884-48b9-b631-53be23adfaae  
accessPoint = http://example.com/carWash
```

● Binding Template

- *Binding Key* – eindeutige Identifikation der technischen Dienstrealisierung
- *accessPoint* – Aufrufpunkt des Dienstes (URI, E-Mail Adresse, Telefonnummer, ...)

Web-Services: Die Technik hinter den Kulissen -- UDDI



tModel

tModelKey = c210b702-f3bc-431b-8822-0257e820872f
name = Carl's Car Wash Service @ NY

● tModel

- *tModelKey* – eindeutige Identifikation der konkreten Dienstinkarnation
- *Name* – Lesbare Beschreibung des konkreten Dienstes (evtl. mehrsprachig)

Web-Services: Die Technik hinter den Kulissen –WSDL und UDDI

```

<definitions name="CarWashService"
  targetNamespace="http://...">
  <import namespace="..."
    location="...">
  <service name="CarWashService">
    <port name="CarWashNY"
      binding="ccw:CarWashBinding">
  ...

```

```

<definitions name="CCW-interface"
  targetNamespace="http://">
  <binding name="CarWashBinding"
    type="tns:CarWashService">
  ...

```

```

<businessEntity businessKey="..."
  <name>Carl, Inc.</name>
  <businessService serviceKey="..."
    <name>CarWashService</name>
  <bindingTemplates>
    <bindingTemplate bindingKey="...">
      <tModelInstanceInfo bindingKey="...">
      ...
    </bindingTemplate>
  </bindingTemplates>
  <overviewDoc>...</overviewDoc>
  ...
  <tModel tModelKey="..."
    <name>...</name>
  <overviewDoc>
    <overviewURL>...</overviewURL>
  </overviewDoc>
  ...

```

Referenzen

● XML

<http://www.w3.org>

● SOAP

<http://www.w3.org/TR/SOAP>

<http://www.w3.org/2000/xp/Group/1/08/29/soap12-part1.html>

<http://www.w3.org/2000/xp/Group/1/08/29/soap12-part2.html>

● WSDL

<http://www.w3.org/TR/wsdl>

● UDDI

<http://www.uddi.org>

<http://www-3.ibm.com/services/uddi/find>

<http://ibm.com/services/uddi>

<http://www-124.ibm.com/developerworks/projects/uddi4j/>

<http://ibm.com/websphere/developer/downloads/UDDIregistry.html>

<http://uddi.microsoft.com/search.aspx>

<http://test.uddi.microsoft.com>