

# DAIMLERCHRYSLER

## **XML-basierter Datentransport SOAP und der W3C-standardisierte Nachfolger**

Mario Jeckle  
DaimlerChrysler Forschungszentrum Ulm  
mario.jeckle@daimlerchrysler.com  
mario@jeckle.de  
www.jeckle.de

## Gliederung

I Herausforderung:

Kommunikation in heterogenen Umgebungen

II Lösungsansatz:

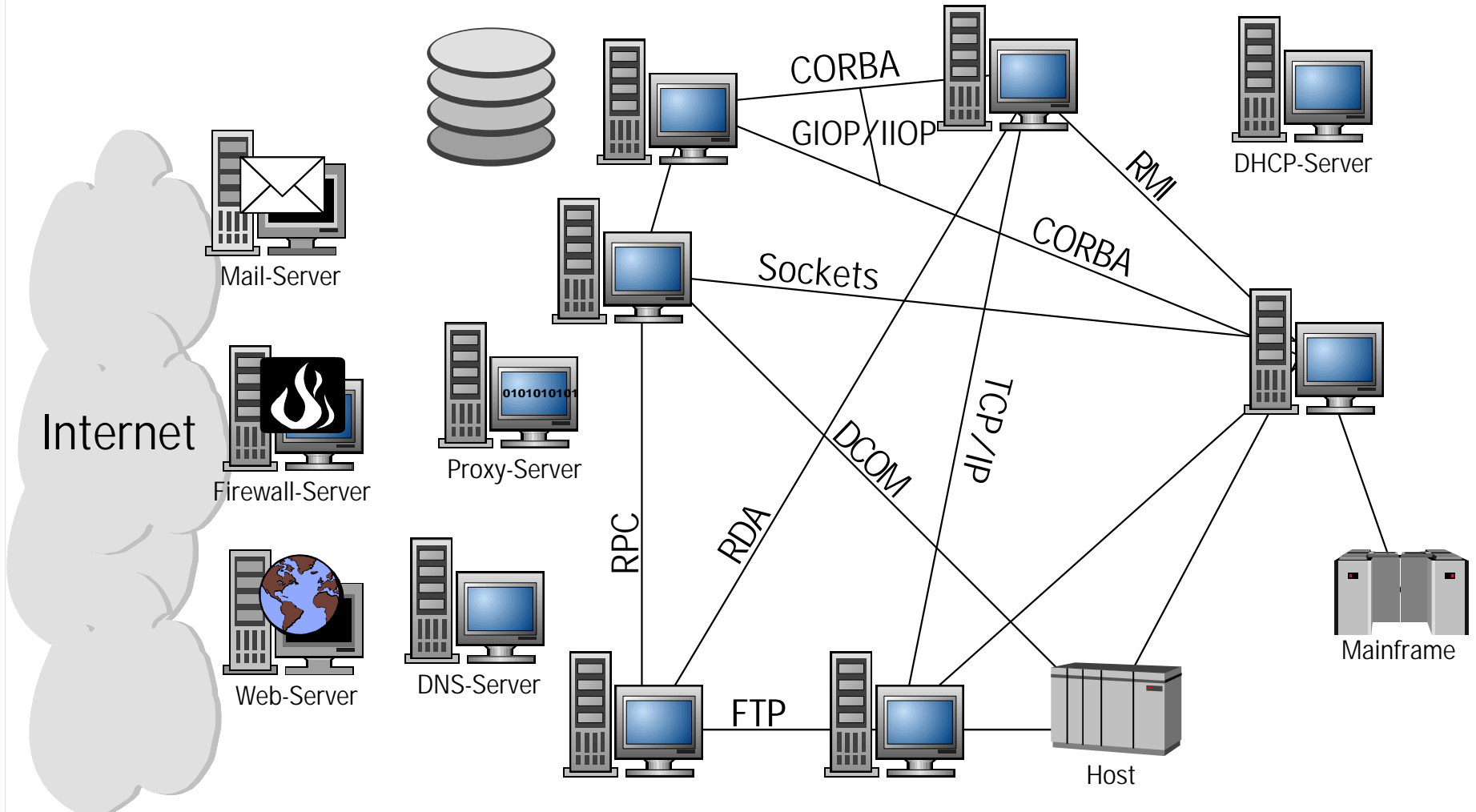
*XML basierter Remote Procedure Call*

Implementierung: *Simple Object Access Protocol (SOAP)*

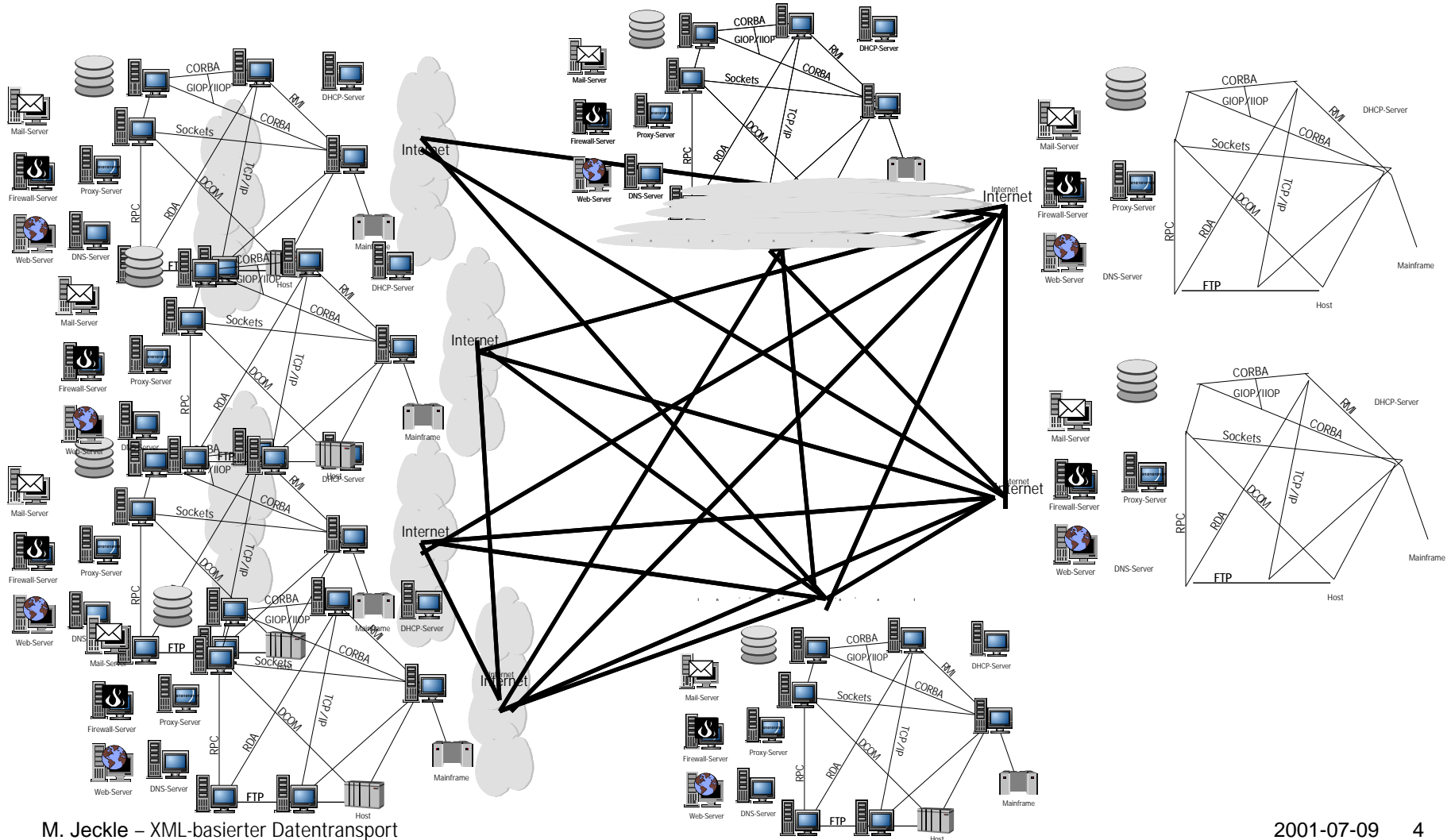
III Ausblick auf dem standardisierten SOAP-Nachfolger:

W3C's XML Protocols

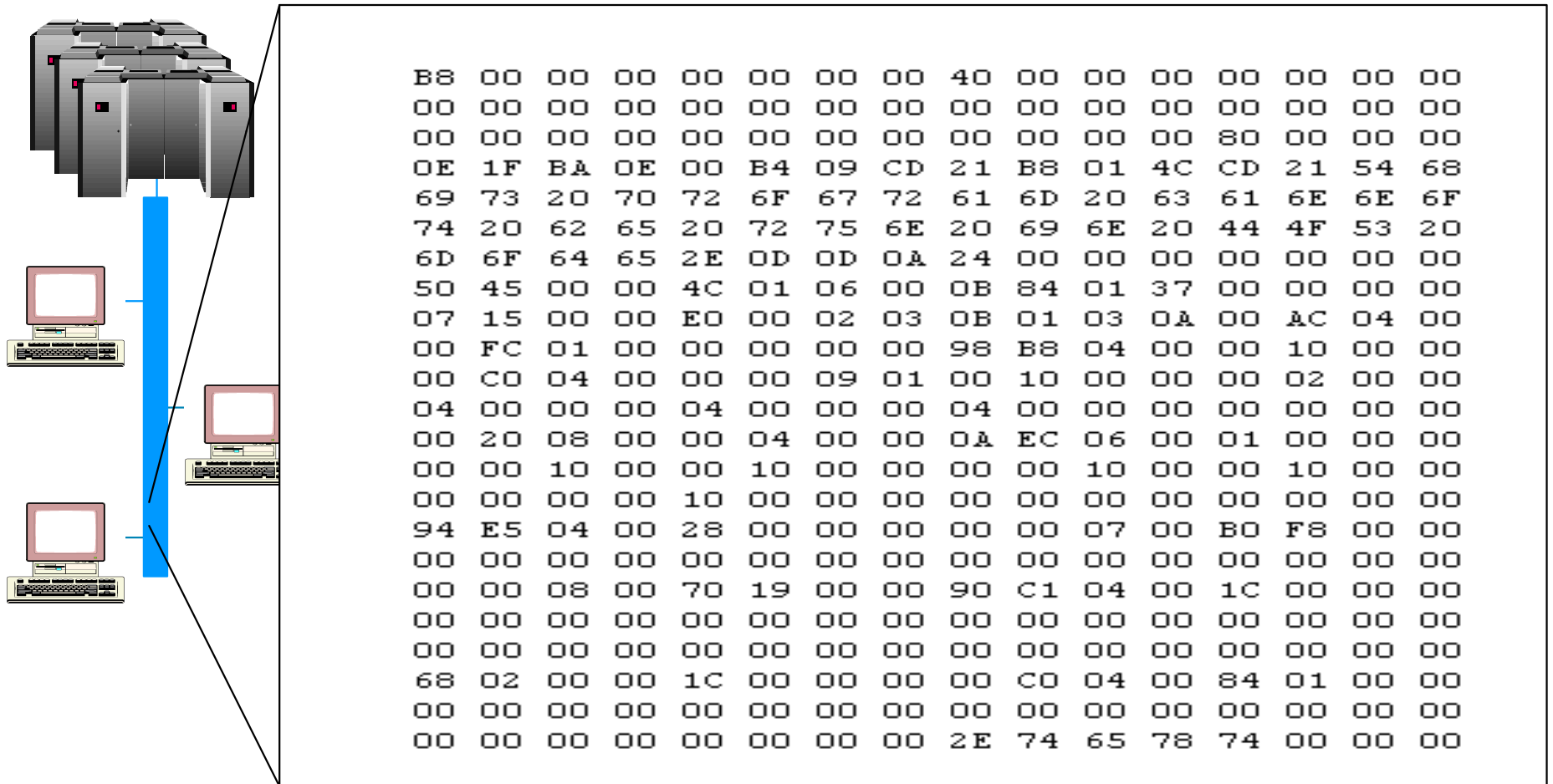
# Die Herausforderung ...



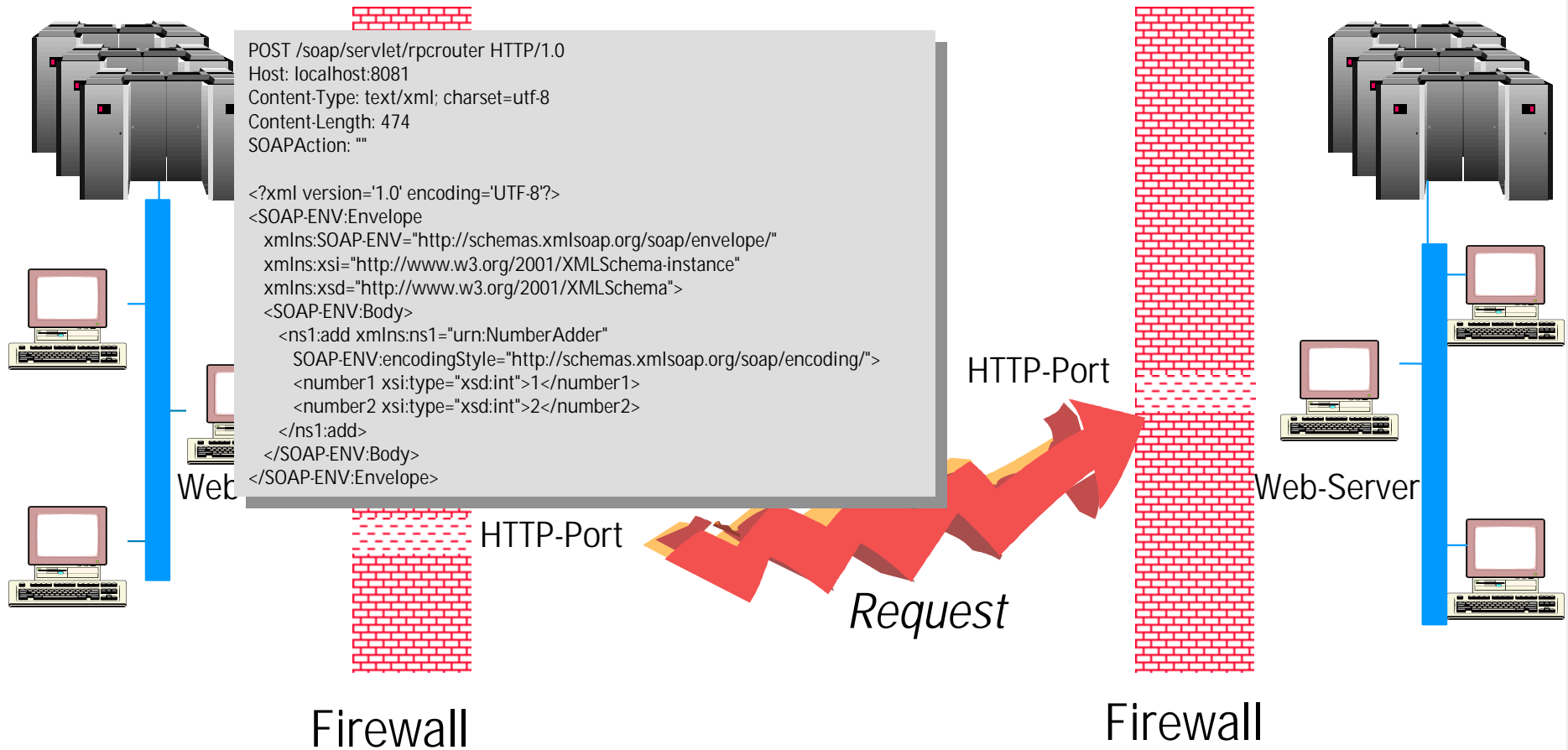
# Die Herausforderung ...



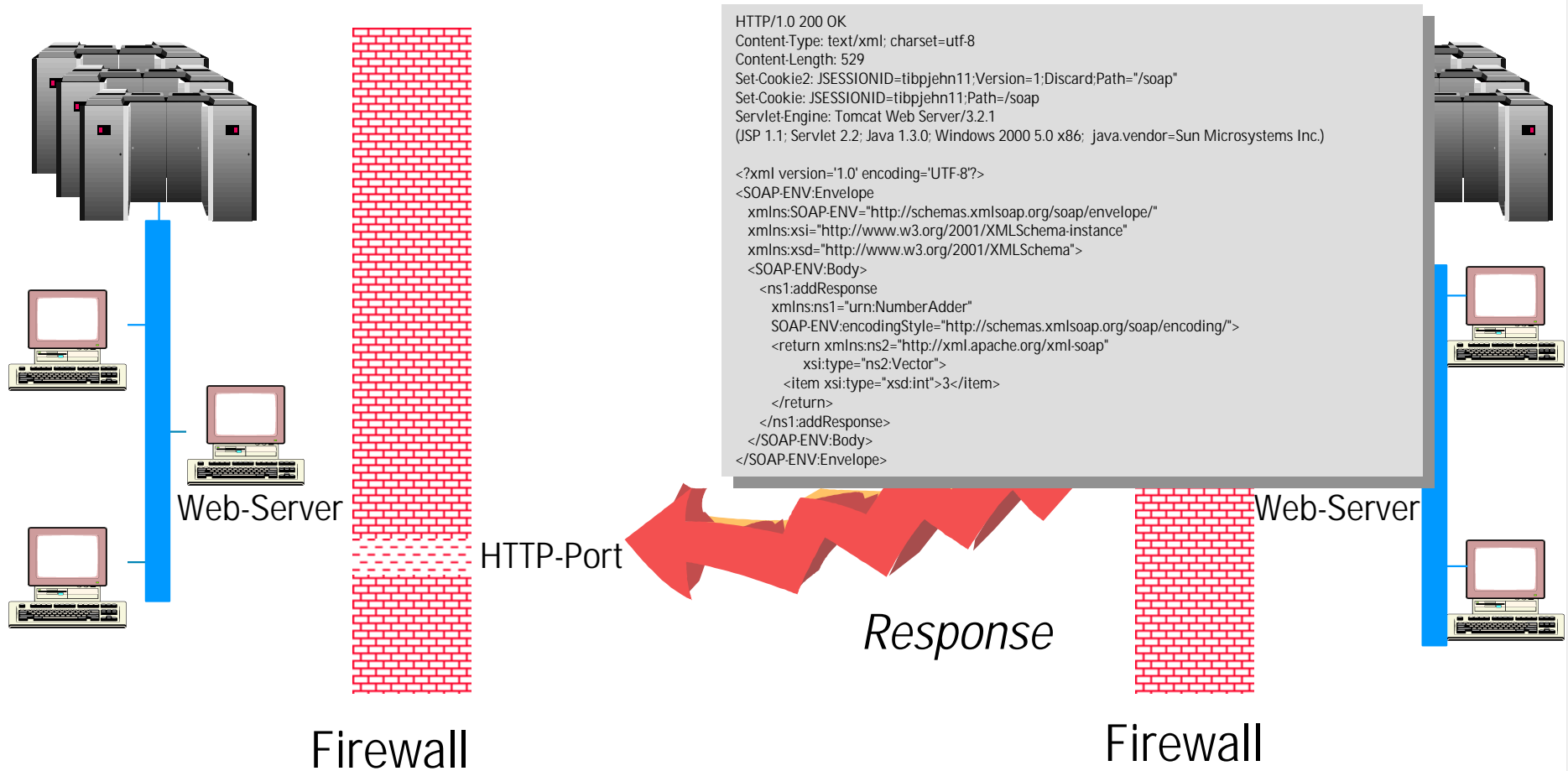
# Lösungsidee ... Internettechnologie



# Internettechnologie ... HTTP und XML

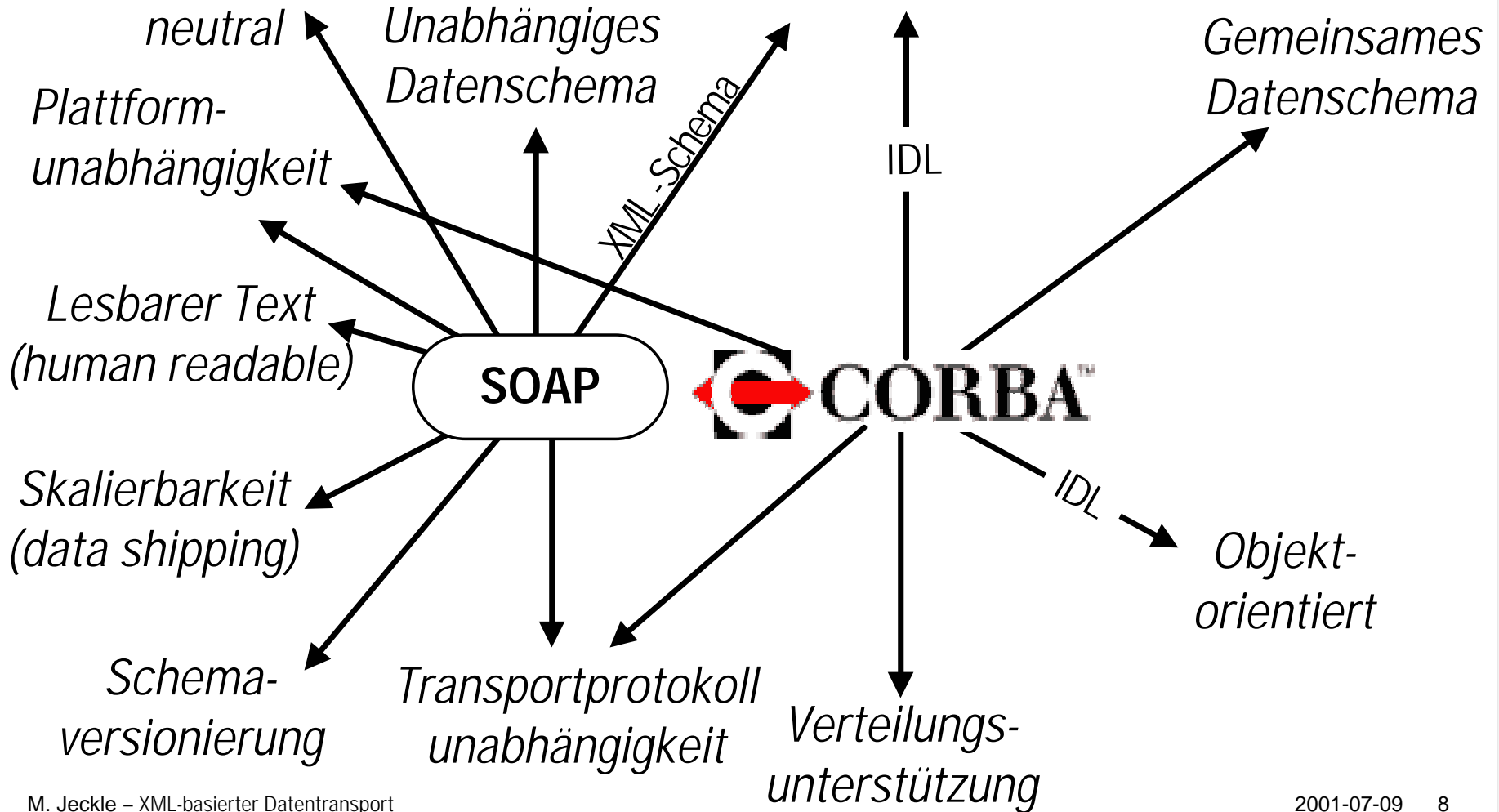


# Internettechnologie ... HTTP und XML



# CORBA vs. SOAP







Programmiermodell-



## CORBA vs. SOAP

<i>Kriterium</i>	<i>CORBA</i>	<i>SOAP</i>
<p><b>Plattformunabhängigkeit:</b>                      Realisierung von Client und Server in (nahezu) beliebiger Programmiersprache auf beliebigem Betriebssystem</p>	✓	✓
<p><b>Lesbarer Inhalt:</b>                      Ausgetauschte Dateninhalte mit Low-End Werkzeugen (wie Texteditoren) verarbeitbar</p>	⊘	✓
<p><b>Datentypen:</b>                      Verarbeitung von programmiersprachlichen Datentypen                      Definition eigener Inhaltstypen</p>	✓	✓

## CORBA vs. SOAP

<i>Kriterium</i>	<i>CORBA</i>	<i>SOAP</i>
<p><b>Verarbeitung großer Datenmengen:</b>                      Können Datenbestände größer als der zur Verfügung stehende Hauptspeicher verarbeitet werden</p>		
<p><b>Schemaversionierung:</b>                      Verarbeitung von Instanzen verschiedener Schema-versionen durch die Applikation</p>		
<p><b>Verteilungsunterstützung:</b></p> <ul style="list-style-type: none"> <li>• Verteilungstransparenz</li> <li>• Lookup-Funktionalität</li> <li>• Authentifizierung und Sicherheit</li> <li>• Transaktionsunterstützung</li> </ul>		

## CORBA vs. SOAP

### *Kriterium*

### *CORBA*

### *SOAP*

#### **Nachrichtenübermittlung**

Informationstransfer im Stile von Datagrammen  
(keine RPC-Semantik)



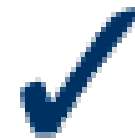
#### **Programmiermodellunabhängigkeit**

Einsetzbarkeit in verschiedenen Programmiersprachen;  
Paradigmenneutralität

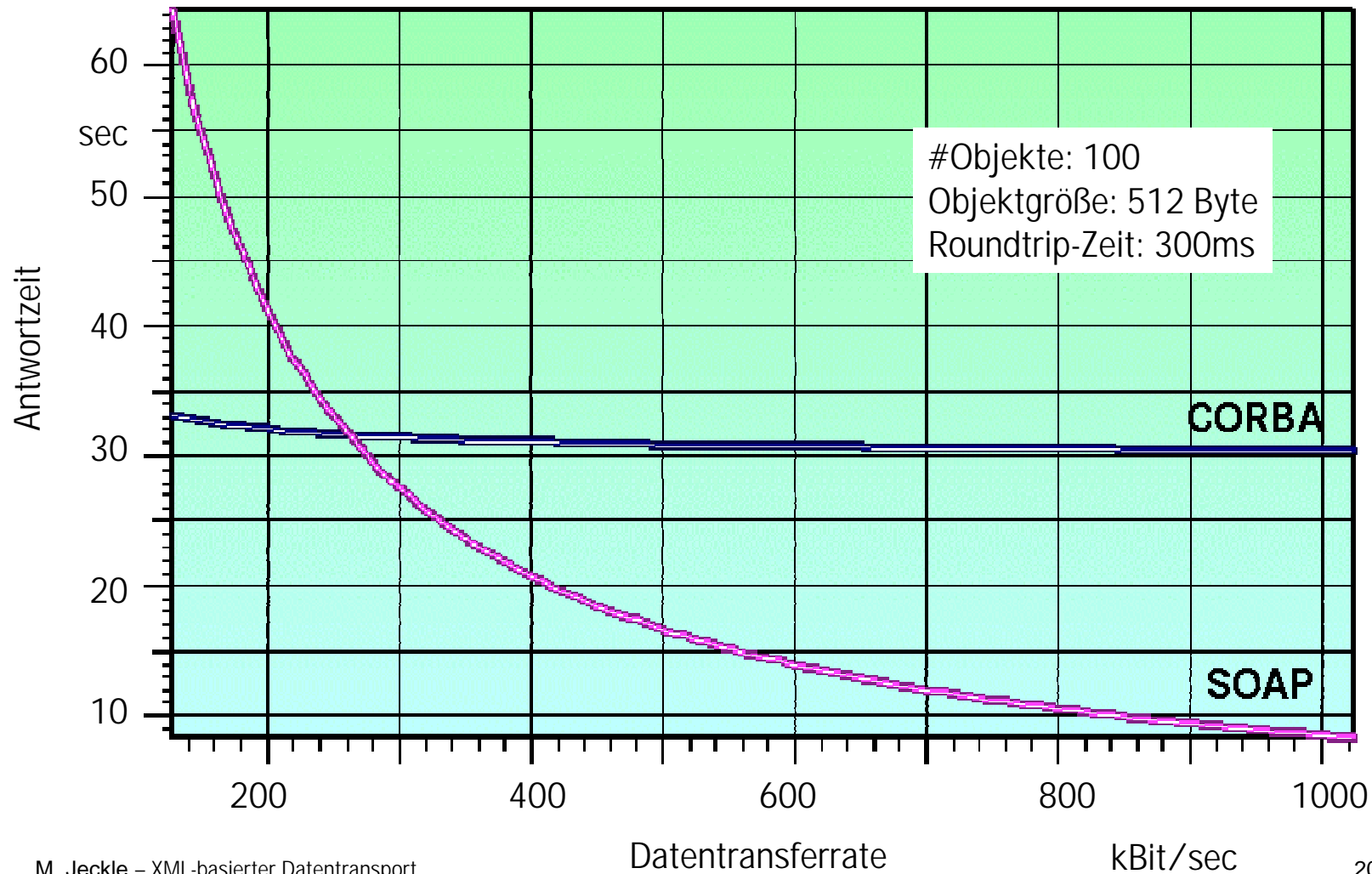


#### **Transportprotokollunabhängigkeit**

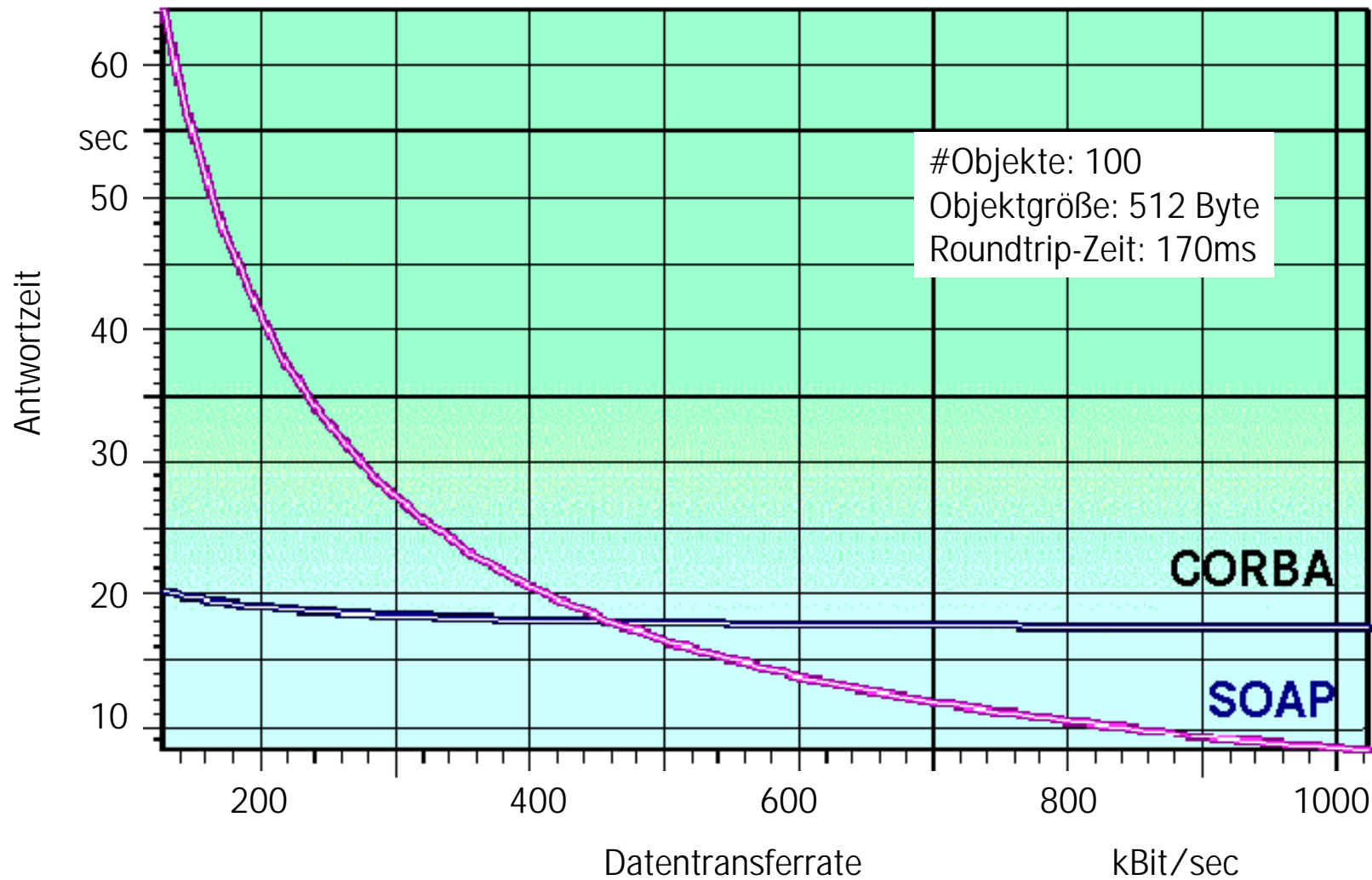
Transparente Verwendung verschiedener  
(Transport- oder Sitzungs-)Protokolle



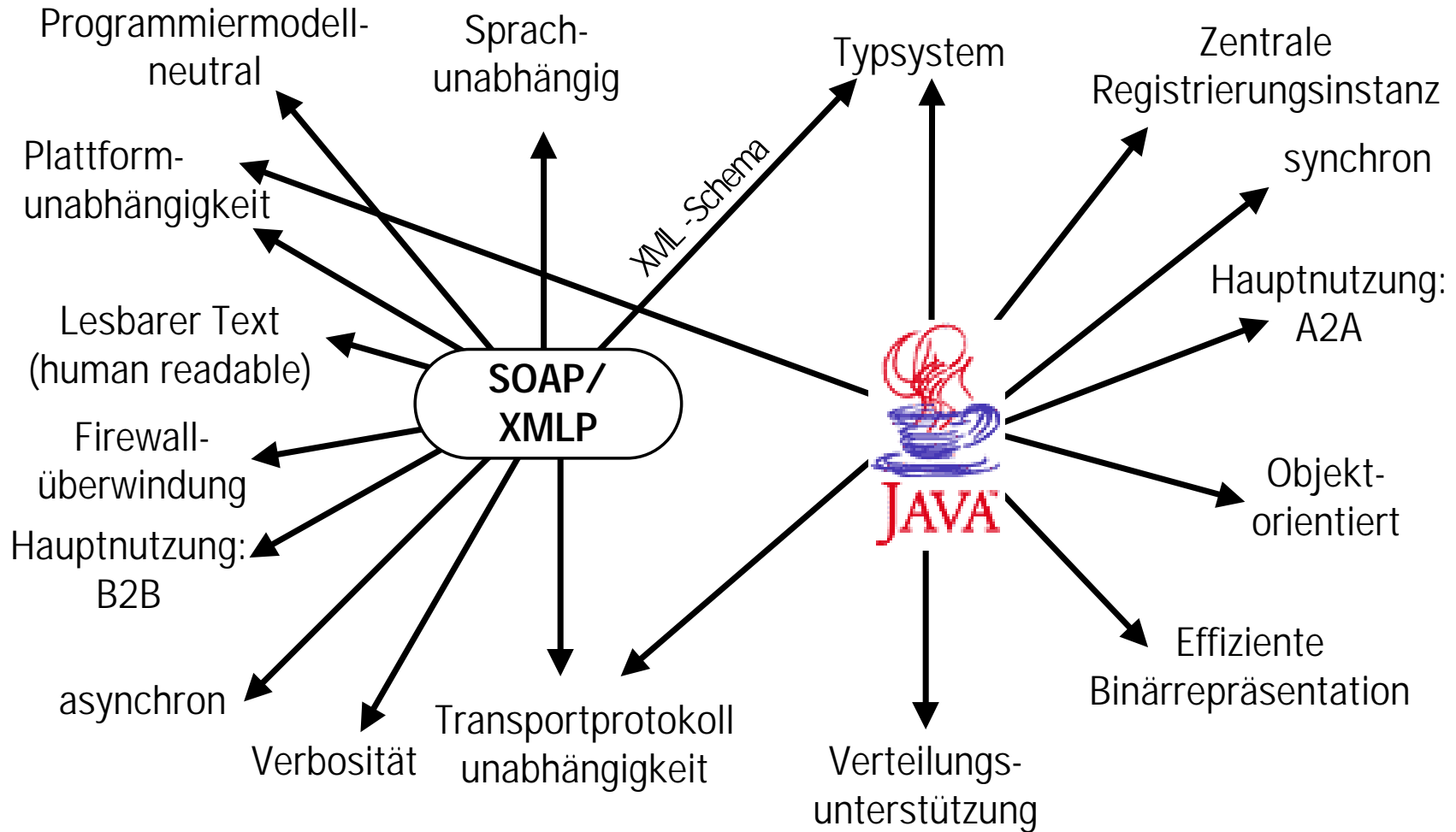
# SOAP vs. CORBA – Antwortzeitverhalten (WAN)



## SOAP vs. CORBA – Antwortzeitverhalten (LAN)



## XML-basierte RPC-Mechanismen: SOAP/XMLP vs. RMI



## Was ist SOAP?

"The Simple Object Access Protocol is a protocol specification for invoking methods on servers, services, components and objects. SOAP codifies the existing practice of using XML and HTTP as a method invocation mechanism.

The SOAP specification mandates a small number of HTTP headers that facilitate firewall/proxy filtering.

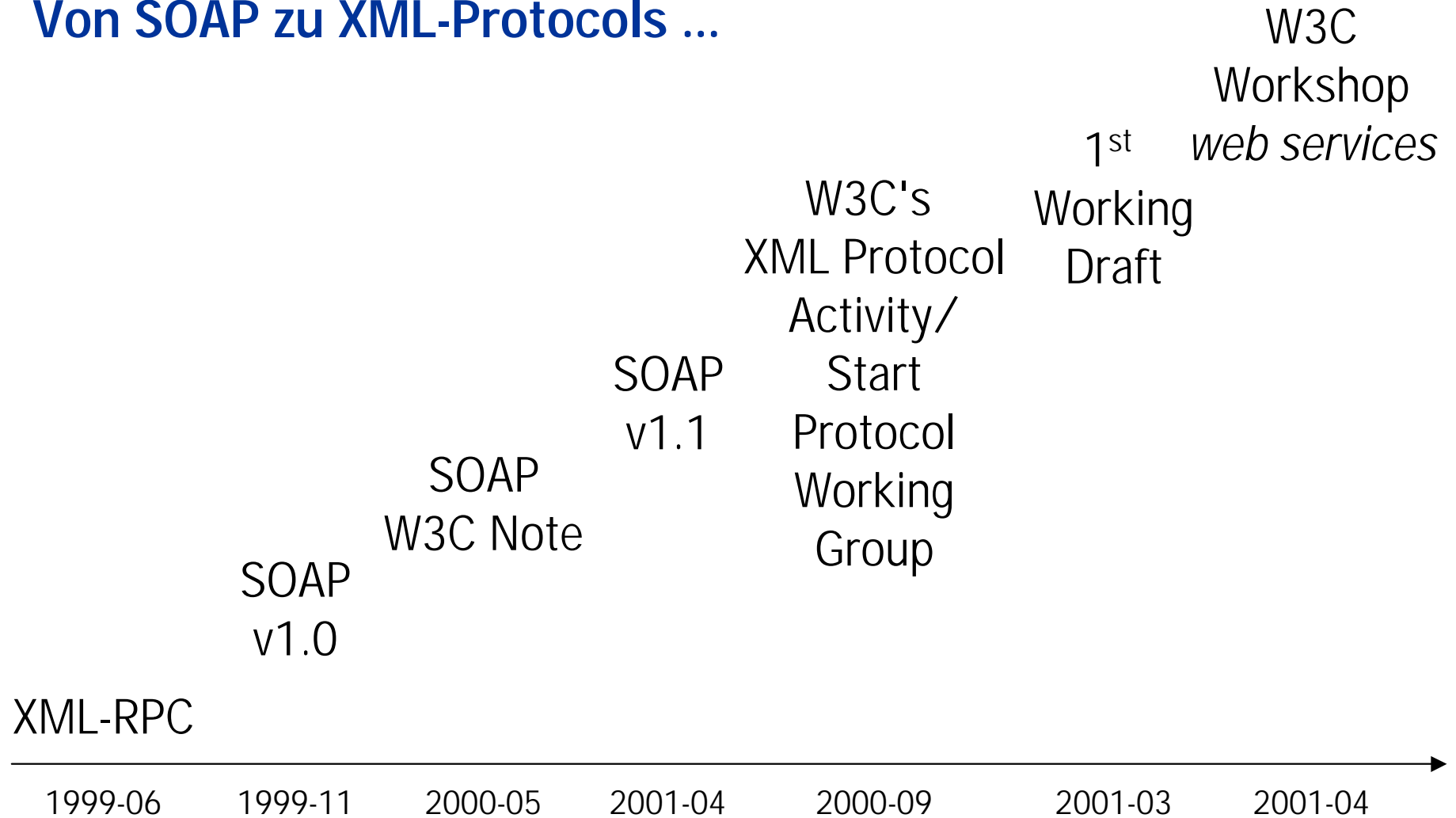
The SOAP specification also mandates an XML vocabulary that is used for representing method parameters, return values, and exceptions."

(DevelopMentor)

Ziele:

- einfacher offener Kommunikationsmechanismus auf Basis von Internet-Standards
- Erweiterbares XML-Format

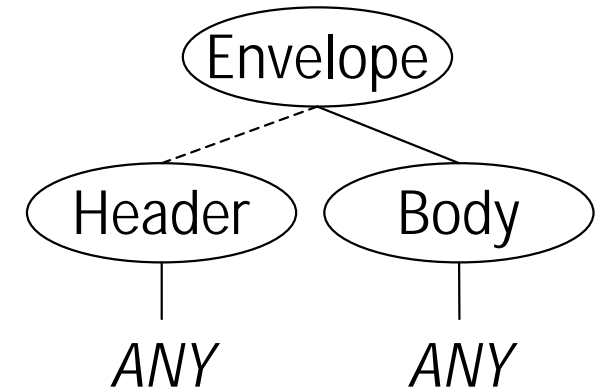
# Von SOAP zu XML-Protocols ...



## SOAP-Architektur

### SOAP-Envelope

Beschreibung einer XML-codierten Nachricht  
(XML-Namensräume, etc.)



### SOAP-Codierungsrichtlinien

Mechanismus zur Überführung der Applikationsdatenstrukturen  
in XML

### SOAP-Nachrichtendarstellung

Richtlinien für die Darstellung aufgerufener Dienste

## SOAP und XML

Transportprotokoll (z.B. HTTP)

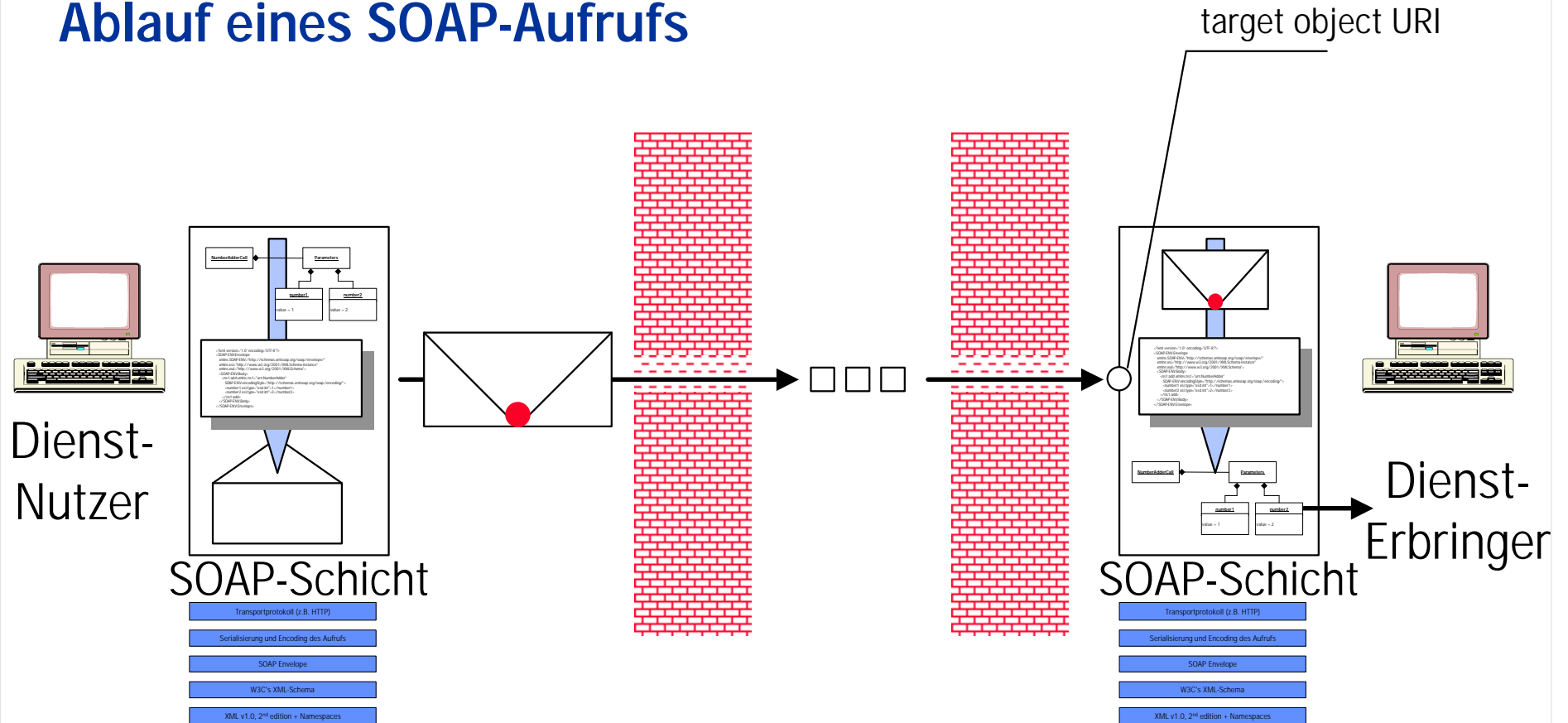
Serialisierung und Encoding des Aufrufs

SOAP Envelope

W3C's XML-Schema

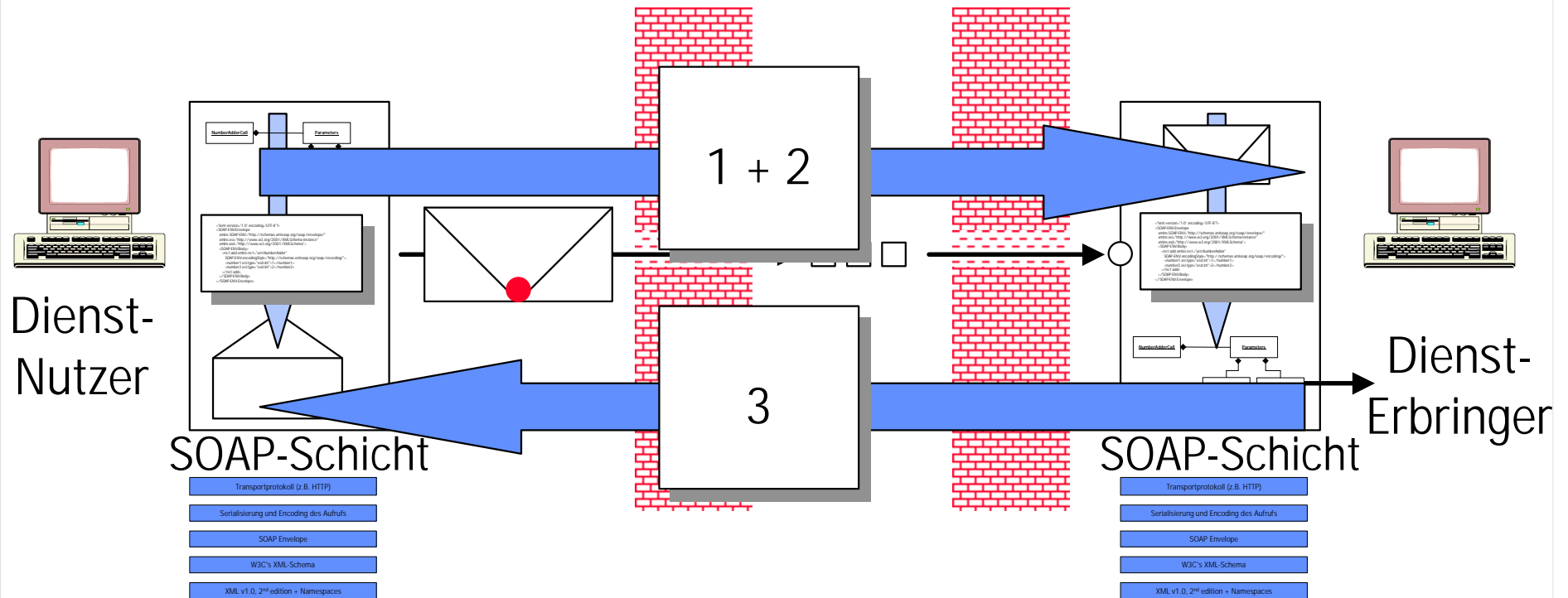
XML v1.0, 2<sup>nd</sup> edition + Namespaces

# Ablauf eines SOAP-Aufrufs



# SOAP am Beispiel ...

## ... ein einfacher Rechner



# Anforderungen an XML Protocols (XMLP)

## Anforderungen

berücksichtigte

unberücksichtigte

- Allgemeines
- Einfachheit und Stabilität
- Datenkapselung und Erweiterbarkeit
- RPC-Vermittler
- Datenrepräsentation
- Protokollbindung
- RPC-Konvention

- Direkte Behandlung binärer Daten
- Kompaktifizierung und Kompression
- Transport- und Vermittlungsspezifika
- Applikationssemantik (Transaktionen, Sicherheit)
- Dienst-Metadaten (Service Directory)

## Anforderungen an XMLP: Allgemeines

- Keine Festlegung auf
  - Programmiermodell (Sprache oder Paradigma)
  - Kommunikationsmechanismus (*protocol binding*)
  - Kommunikationsszenarien (*one-way, request-response, publish-subscribe, ...*)
- Leichtgewichtige Spezifikation
  - minimale Anzahl zwingender Eigenschaften
  - Orthogonalität optionaler Eigenschaften
- Kommunikation zwischen Partnern ohne vorheriges Wissen  
übereinander

## Anforderungen an XMLP: Einfachheit und Stabilität

- XML-Schemabeschreibung eines Umschlages (*envelope*) zur Kapselung der transportierten Daten
- Definition eines Verarbeitungsmodells für den Umschlag;  
Berücksichtigung möglicher Fehlersituationen
- Einbettungsszenarien für Applikationsinformation in den Umschlag:
  - Umschlag enthält Applikationsinformation physisch
  - Applikationsinformation wird extern gehalten und durch den Inhalt des Umschlages referenziert
  - Umschlag enthält weitere Umschläge
  - Referenz auf externe Umschläge innerhalb eines Umschlages

## Anforderungen an XMLP: Datenkapselung und Erweiterbarkeit

- Einsetzbarkeit über Organisationsgrenzen hinweg
- Modularität und Erweiterbarkeit durch Schichtenbildung als Basis einer langlebigen und zukunftssicheren Spezifikation
- Formulierung der Spezifikation so, daß Konformitätstests einfach durchzuführen sind; somit für Implementierer einfache Möglichkeit zur Gewährleistung der Konformität
- Dezentrale unabgestimmte Erweiterungsmöglichkeit
- Einfach Einsetzbarkeit in Systemen die bereits XML (insbesondere XML-Schema und XML-Namensräume) unterstützen
- Einsetzbarkeit auf Endgeräten mit beschränkten Ressourcen

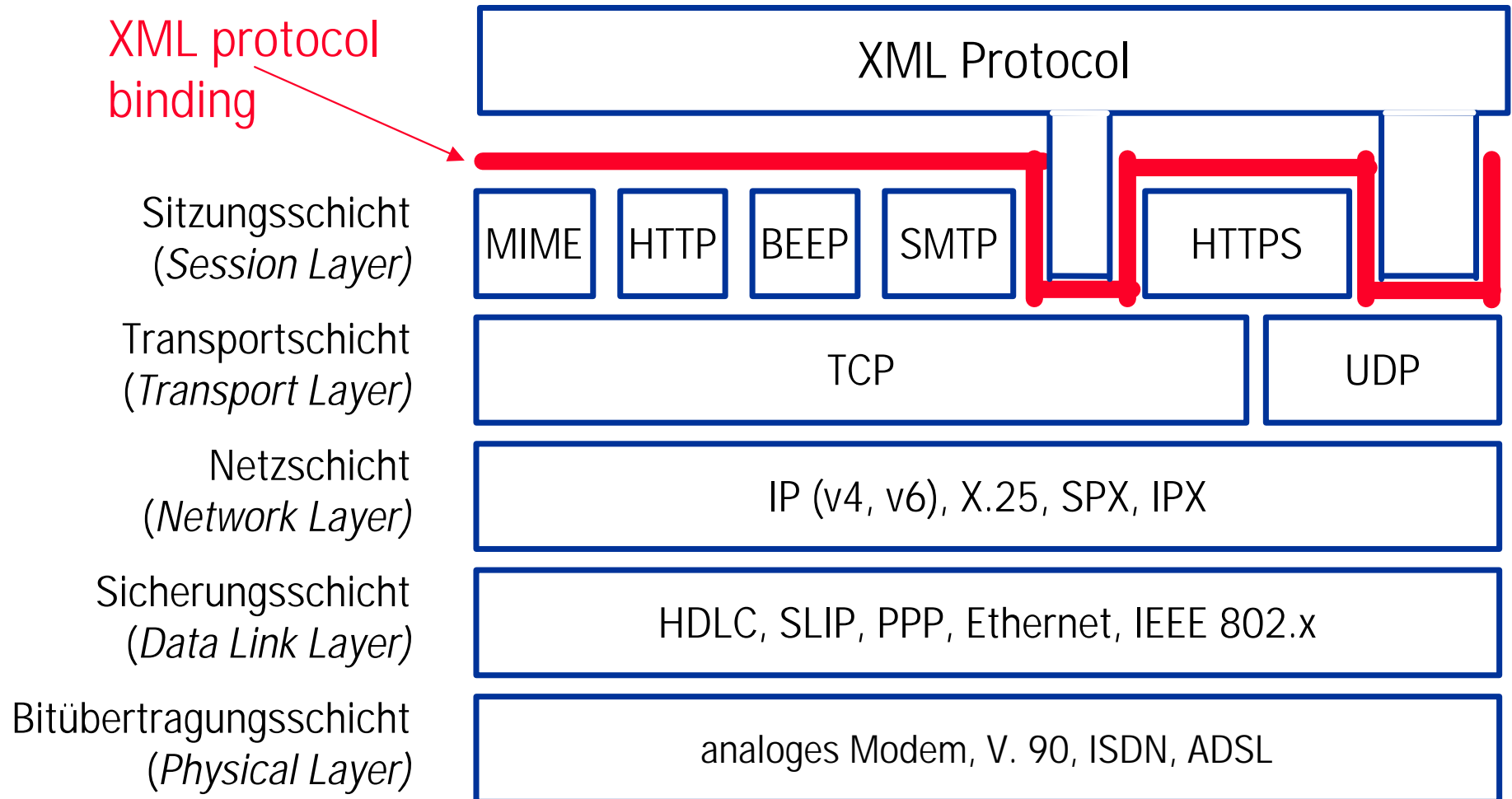
## Anforderungen an XMLP: RPC-Vermittler

- Vermittler (*intermediaries*) können verschiedene Aufgaben auf dem Weg der XMLP-Nachricht zwischen Sender und dem Empfänger erfüllen:
  - Proxy (Empfang obwohl nicht entgeltlicher Empfänger)
  - Cache (Zwischenspeicherung zur Geschwindigkeitssteigerung)
  - Store-and-forward (Zwischenspeicherung gesamter XMLP-Nachrichten vor Weitersendung)
  - Protokollumsetzer (Gateways)
- Transportvermittler (*transport intermediary*)  
Keine Verarbeitung der übermittelten Nachricht, sondern Operation als Teil der Bindung an das Transportprotokoll
- Verarbeitender Vermittler (*processing intermediary*)  
Vollständige XMLP-Prozessoren mit der Möglichkeit die XMLP-Nachricht zu verarbeiten

## Anforderungen an XMLP: Datenrepräsentation

- Orthogonalität zwischen Datendarstellung (*data representation*) und Dateneinbettung (*data encapsulation*) in eine XMLP-Nachricht
- Unterstützung der XML-Schematypen *simple-types* und *complex-types* für die Datenrepräsentation
- Datenrepräsentation muß Instanzen von nicht XML-Schema-basierenden Modellen (z.B.: Objektgraphen, gerichtete Graphen)
- Referenzierung von nicht-serialisierten Daten durch Uniform Resource Identifier (URI)
- Darstellung geschachtelter Arrays

# Anforderungen an XMLP: Protokoll-Bindung



## Anforderungen an XMLP: Protokoll-Bindung

- HTTP-Bindung wird durch XMLP-Arbeitsgruppe erarbeitet
- Beliebige andere Protokollbindungen sind jedoch ohne weiteres möglich
- Mapping auf existierende *Applikationsschichtprotokolle* (z.B.: HTTP, BEEP) kann zu Problemen hinsichtlich Sicherheit, Skalierbarkeit oder Semantik führen
- XMLP-Nachricht kann durch Vermittlungsinstanzen über verschiedene Protokolle transportiert werden
- Nutzung von gesicherten Protokollen (z.B.: *secure HTTP, SSL, S/MIME*) explizit möglich

## Anforderungen an XMLP: RPC-Konventionen

- Vollständige und Eindeutige Identifikation des aufgerufenen Dienstes, Objektes oder Methode durch eine URI
- Unterstützung des Zusammenfügens von Aufruf und Antwort, falls dies nicht durch das gebundene Protokoll erledigt wird
- Parameterübergabe und Transport
- Protokollunabhängiger Mechanismus zur Fehlerrückgabe in Ergebnismnachricht

## Anatomie einer SOAP-Nachricht

SOAP-Nachricht

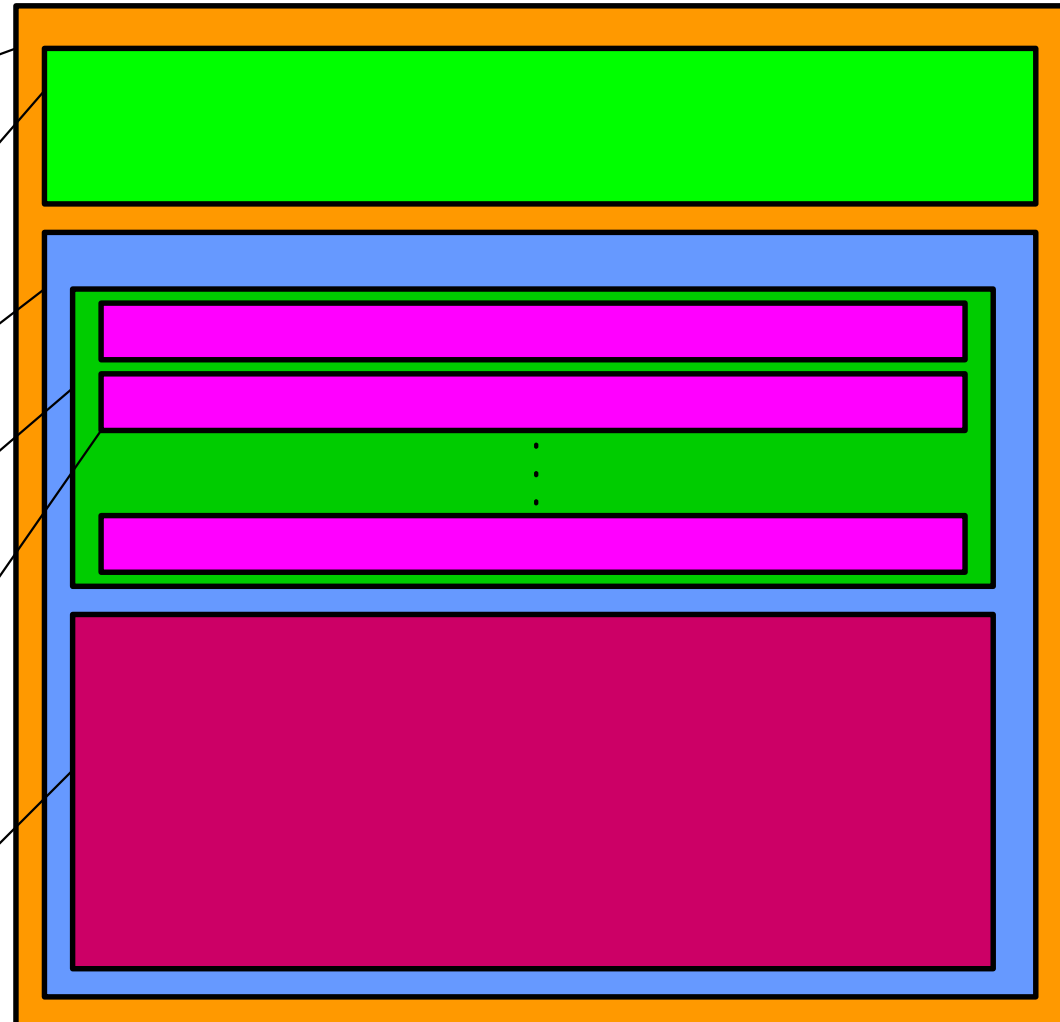
Protokoll-spezifische  
Information

SOAP-Envelope

SOAP-Header

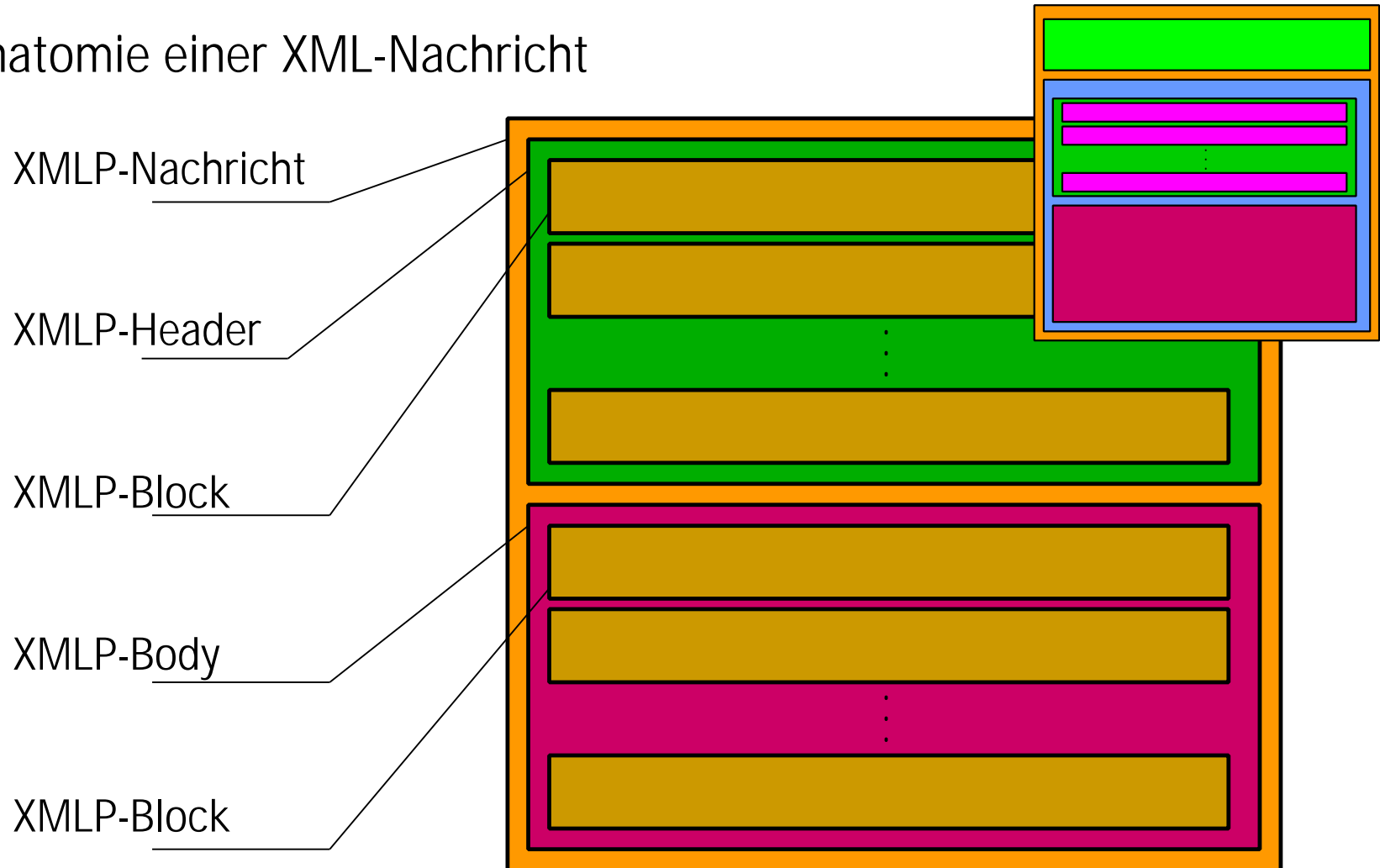
Individuelle  
SOAP-Header

SOAP-Body  
(XML-Nachricht)

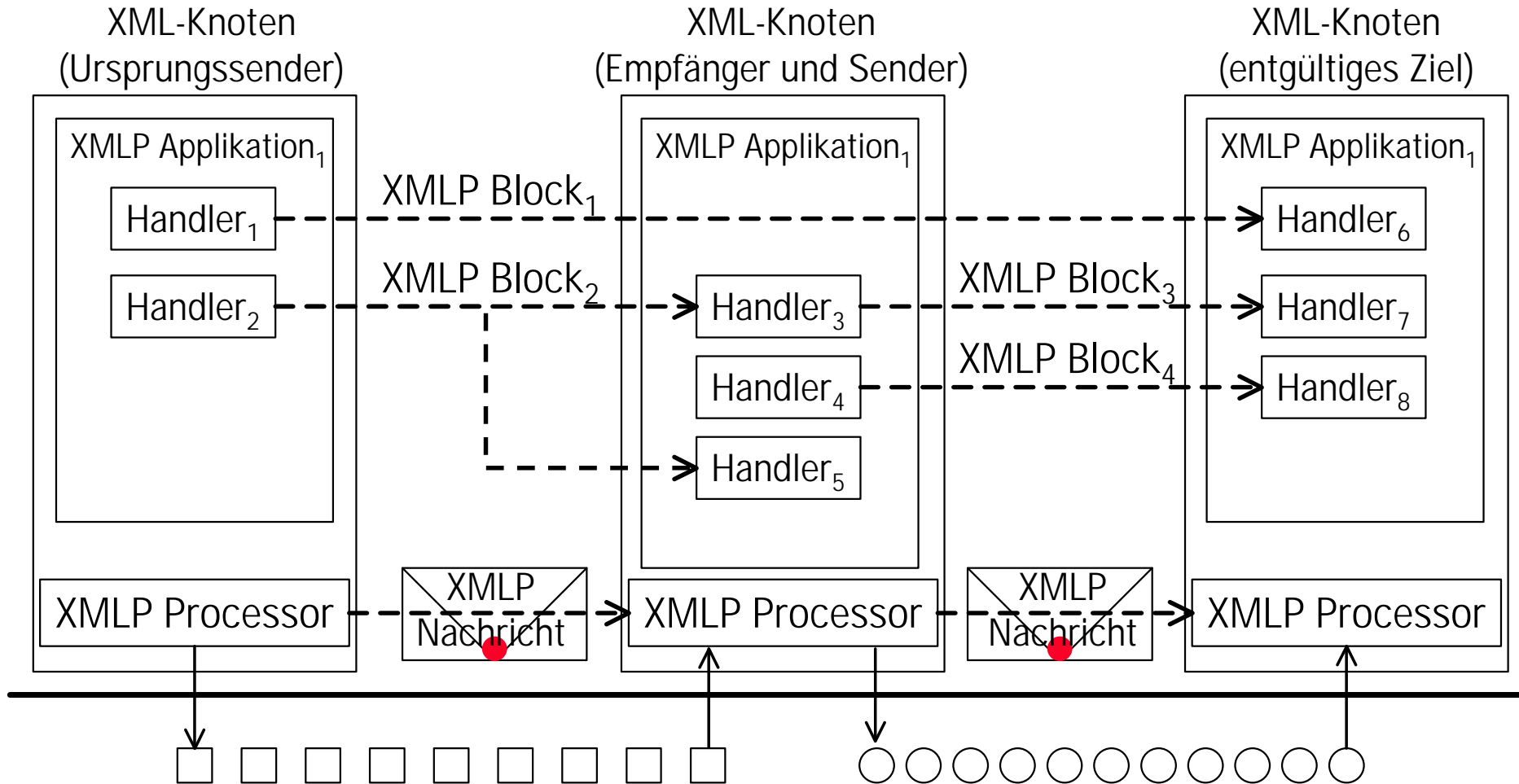


# Ausblick auf die SOAP-Standardisierung: W3C's XML Protocols

## Anatomie einer XML-Nachricht



# Abstraktes Modell der XML Protocols (XMLP)



## Kritik am Ansatz

- XML zur Inhaltsdarstellung
  - zusätzlicher Parsingaufwand beim Empfänger
  - Informationsdarstellung platzraubender als optimierte Binärformate
- HTTP als Transportprotokoll
  - Probleme bei Transaktionsverarbeitung durch fehlende Zustandsinformation  
Lösung: anderes Transportprotokoll verwenden (ab SOAP v1.1 möglich)
- Keine explizite Berücksichtigung von Sicherheit
  - ... nicht im durch SOAP angesprochenen Problembereich
  - Lösung: Nutzung von HTTPS möglich

## Referenzen

### Allgemeines:

- <http://xml.coverpages.org/soap.html>
- [http://www.sun.com/software/xml/developers/xp/;\\$sessionid\\$XIBMMEQAAE1XRAMTA1LU5YQ](http://www.sun.com/software/xml/developers/xp/;$sessionid$XIBMMEQAAE1XRAMTA1LU5YQ)
- <http://www.w3.org/TR/SOAP/>
- <http://www.w3.org/2000/03/29-XML-protocol-matrix>

### Artikel:

- SOAP – die Lösung aller Interoperabilitätsprobleme?  
OBJEKTspektrum, Nr. 6, Nov./Dez. 2000.
- Kommunikation mit dem Simple Object Access Protocol  
Java Spektrum, Nr. 2, Mrz./Apr. 2001.

### SOAP-Implementierungen:

- <http://xml.apache.org/soap/index.html>
- <http://xml.apache.org/axis/index.html>
- <http://msdn.microsoft.com/xml/general/soaptemplate.asp>
- <http://msdn.microsoft.com/net/>
- <http://alphaworks.ibm.com>
- Vergleich MS SOAP vs. IBM/Apache SOAP: [http://windows.oreilly.com/news/soap\\_0800.html](http://windows.oreilly.com/news/soap_0800.html)