

DAIMLERCHRYSLER

Der zukünftige SOAP-Standard des W3C

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

mario.jeckle@daimlerchrysler.com

mario@jeckle.de

www.jeckle.de

Gliederung

I Herausforderung:

Kommunikation in heterogenen Umgebungen

II Lösungsansatz:

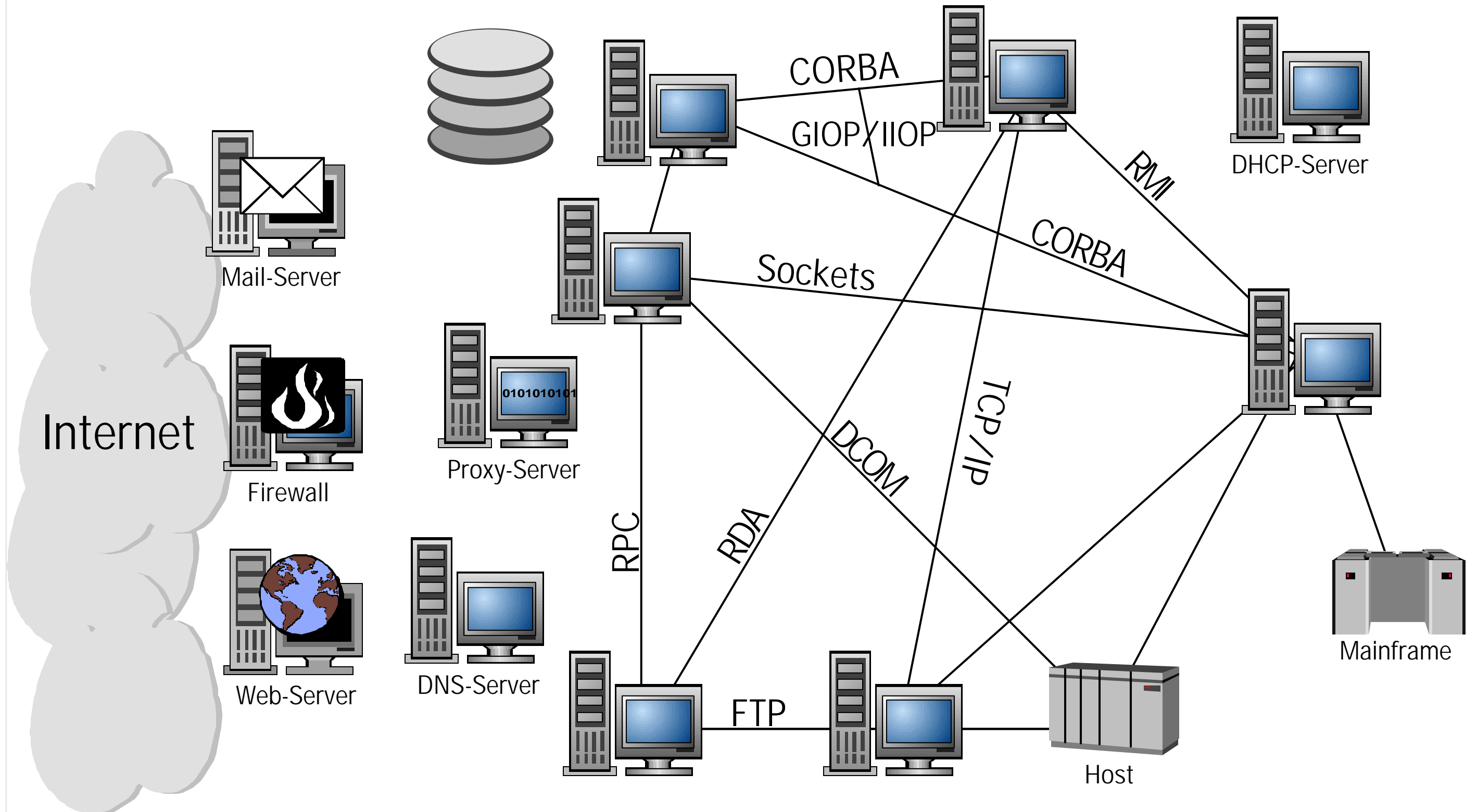
XML basierter Remote Procedure Call

Implementierung: *Simple Object Access Protocol (SOAP)*

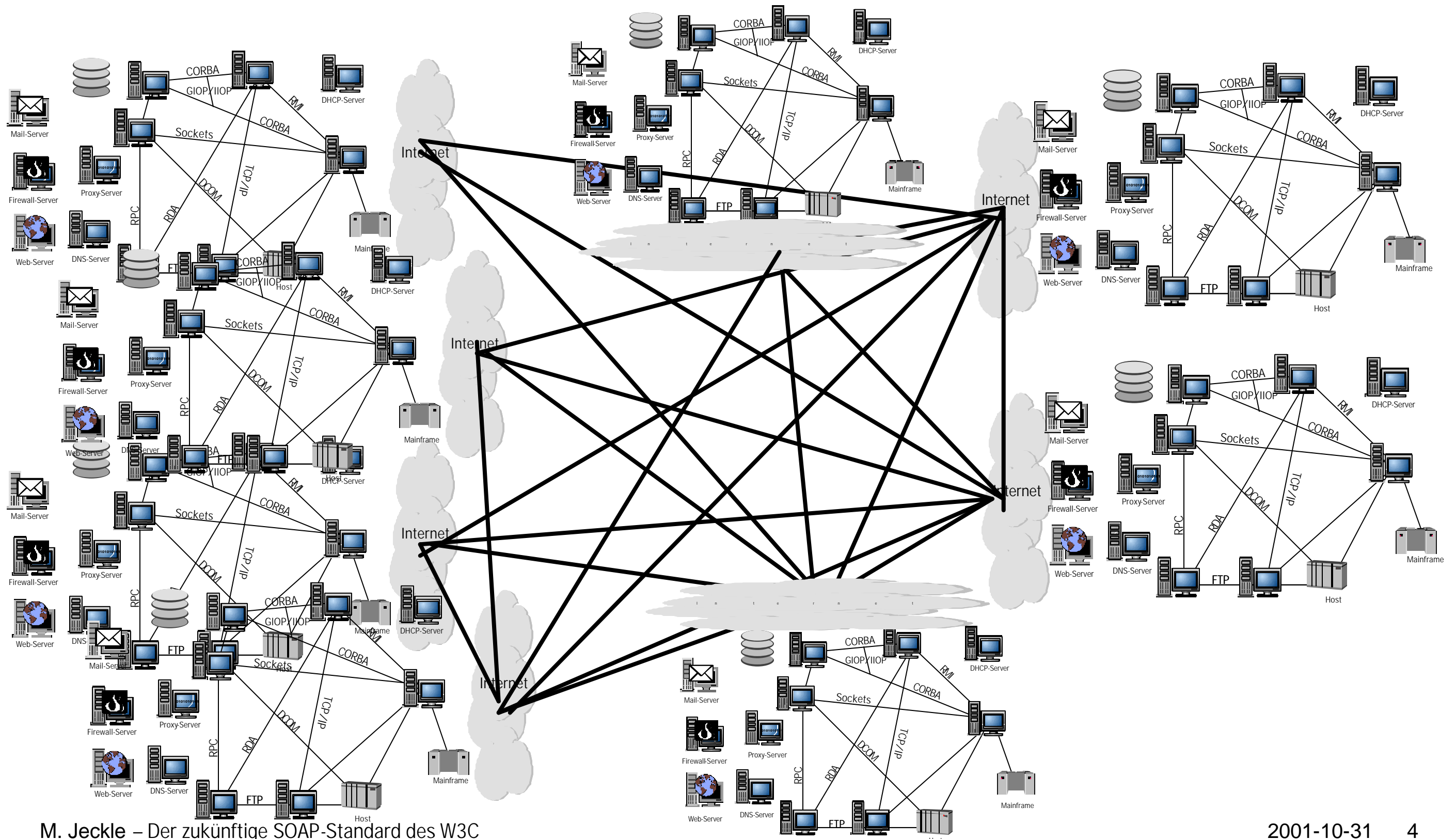
III Ausblick auf den standardisierten SOAP-Nachfolger:

W3C's SOAP v1.2

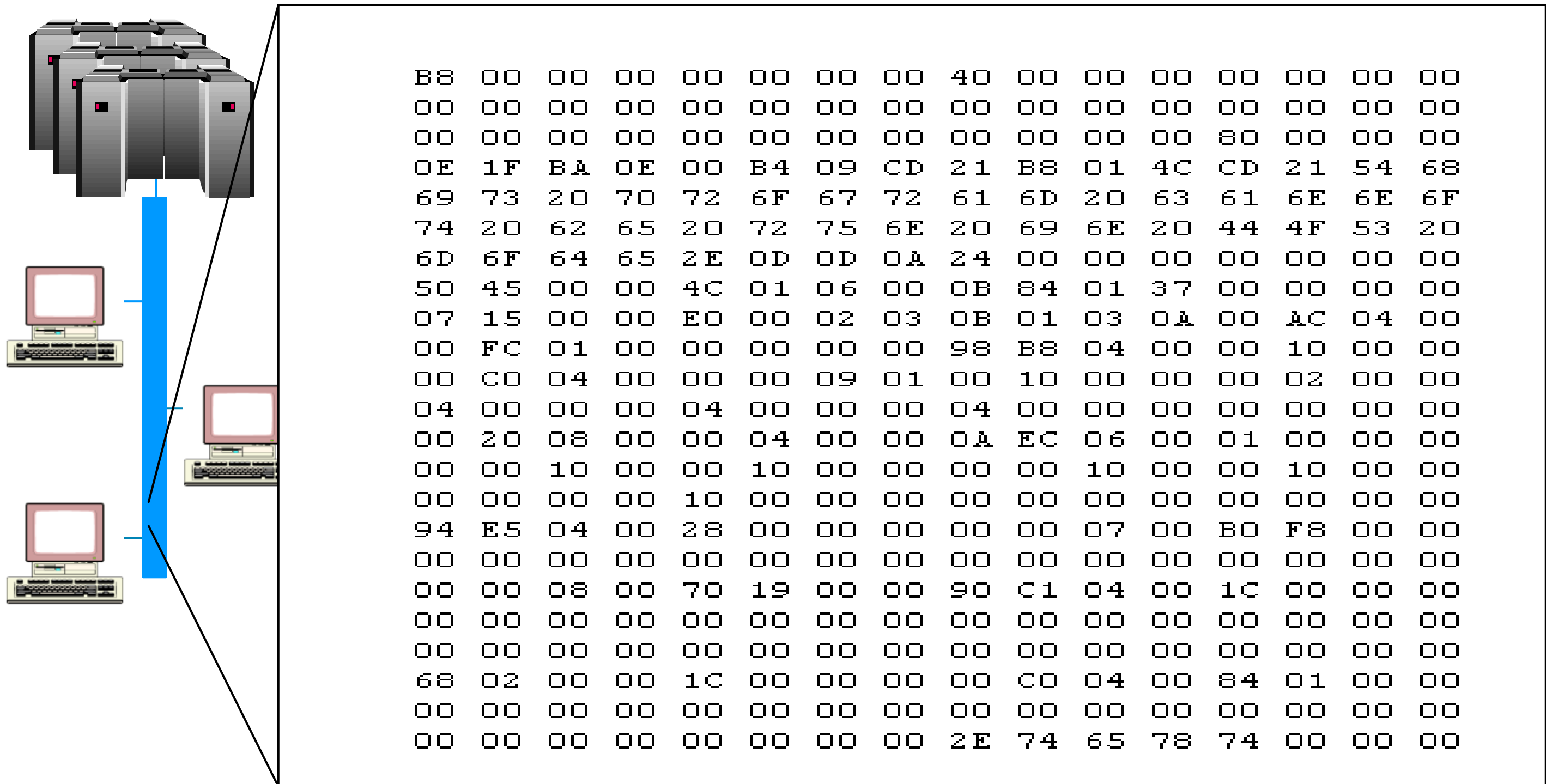
Die Herausforderung ...



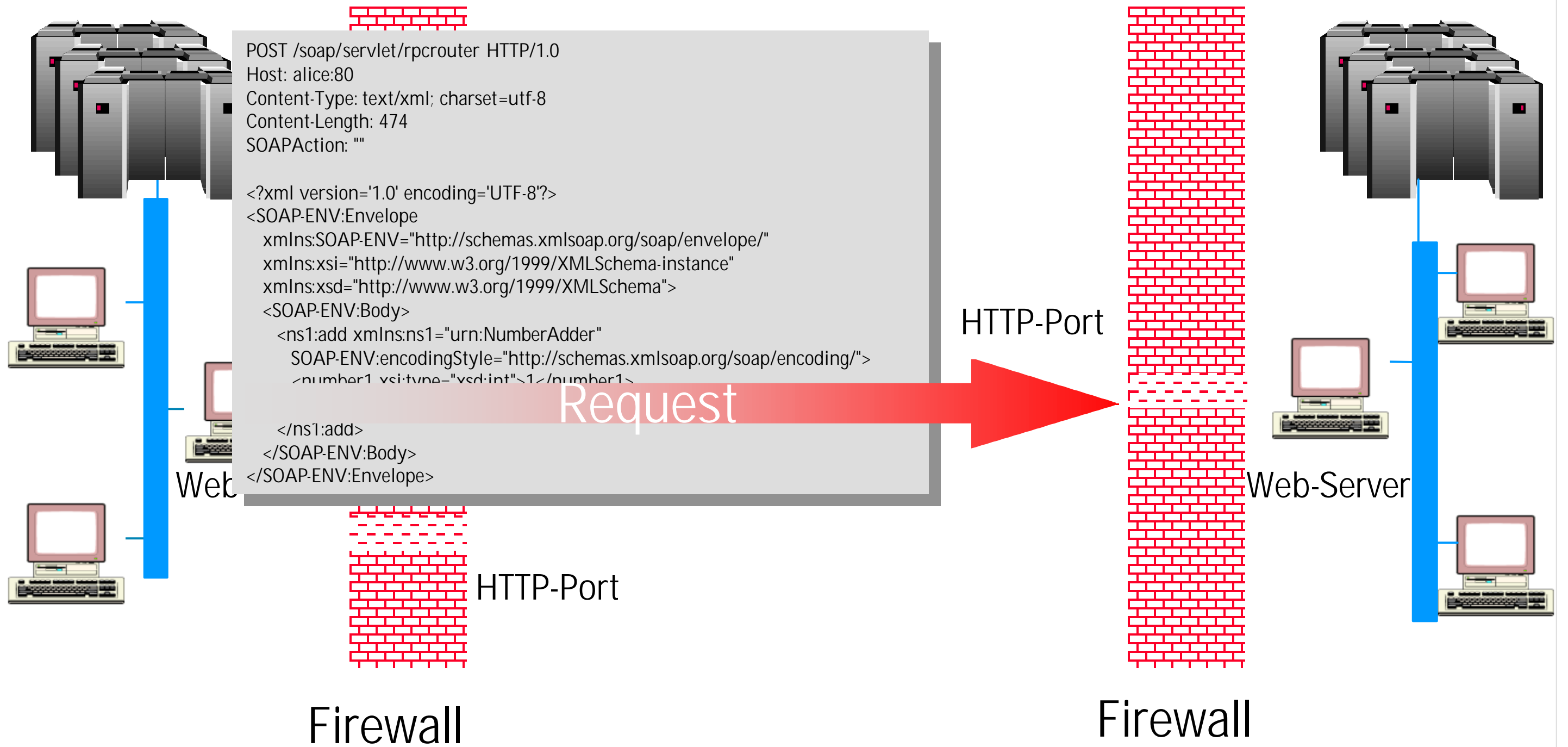
Die Herausforderung ...



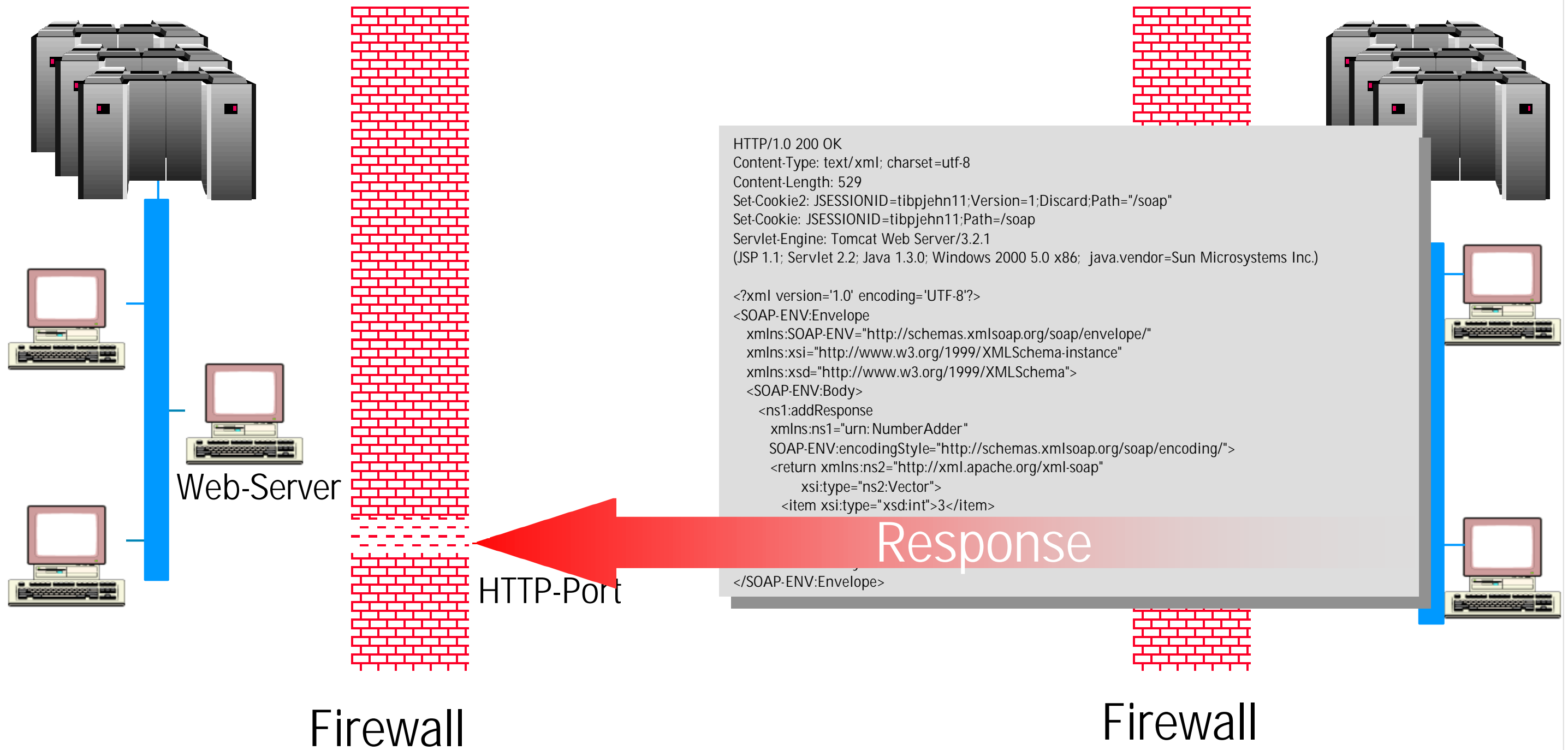
Lösungsidee ... Internettechnologie



SOAP-Grundidee

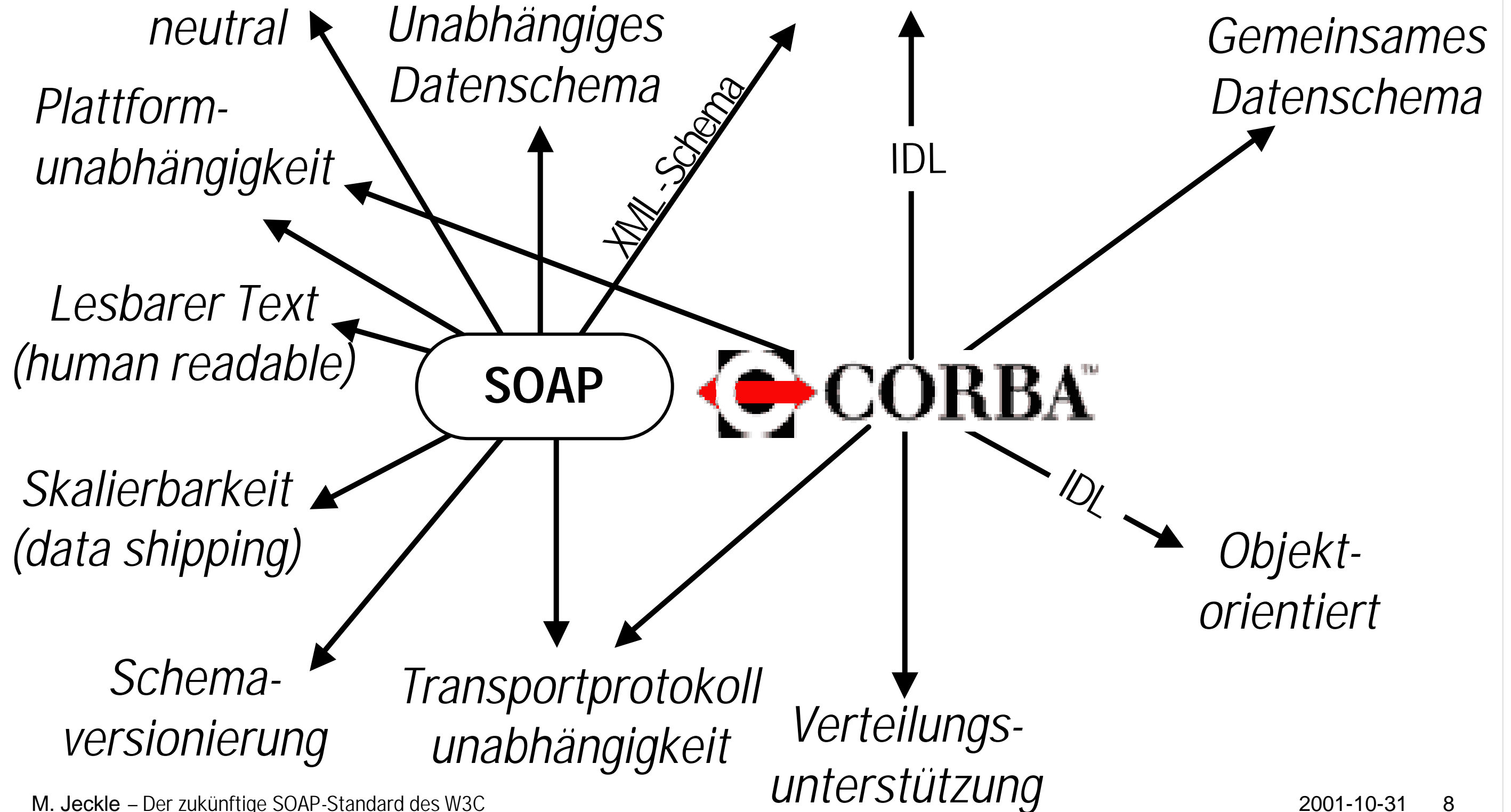


SOAP-Grundidee



CORBA vs. SOAP

Programmiermodell-



CORBA vs. SOAP

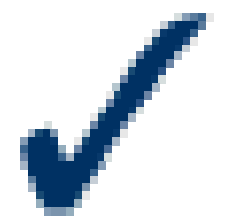
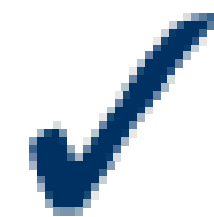
Kriterium

CORBA

SOAP

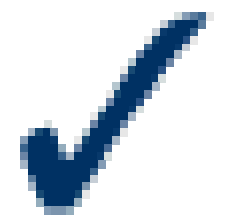
Plattformunabhängigkeit:

Realisierung von Client und Server in (nahezu) beliebiger Programmiersprache auf beliebigem Betriebssystem



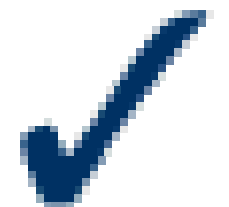
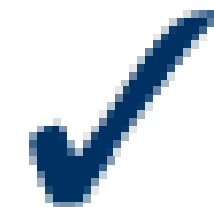
Lesbarer Inhalt:

Ausgetauschte Dateninhalte mit Low-End Werkzeugen (wie Texteditoren) verarbeitbar



Datentypen:

Verarbeitung von programmiersprachlichen Datentypen
Definition eigener Inhaltstypen



CORBA vs. SOAP

Kriterium

Verarbeitung großer Datenmengen:

Datenbestände größer als der zur Verfügung stehende Hauptspeicher können verarbeitet werden

Schemaversionierung:

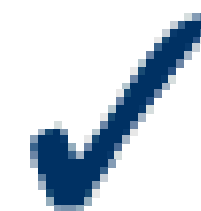
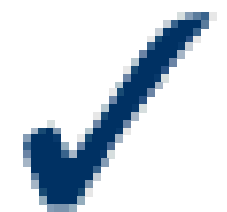
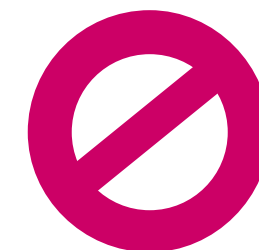
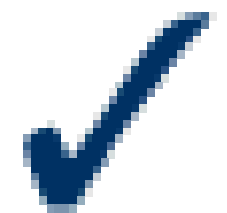
Verarbeitung von Instanzen verschiedener Schema-versionen durch die Applikation

Verteilungsunterstützung:

- Verteilungstransparenz
- Lookup-Funktionalität
- Authentifizierung und Sicherheit
- Transaktionsunterstützung

CORBA

SOAP



CORBA vs. SOAP

Kriterium

Nachrichtenübermittlung

Informationstransfer im Stile von Datagrammen
(keine RPC-Semantik)

Programmiermodellunabhängigkeit

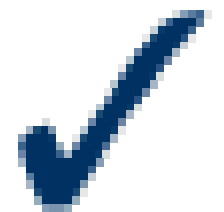
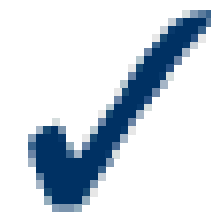
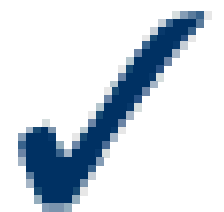
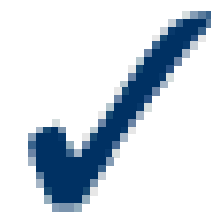
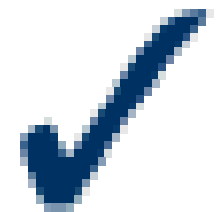
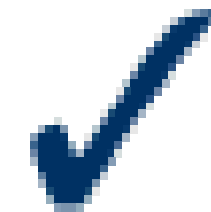
Einsetzbarkeit in verschiedenen Programmiersprachen;
Paradigmenneutralität

Transportprotokollunabhängigkeit

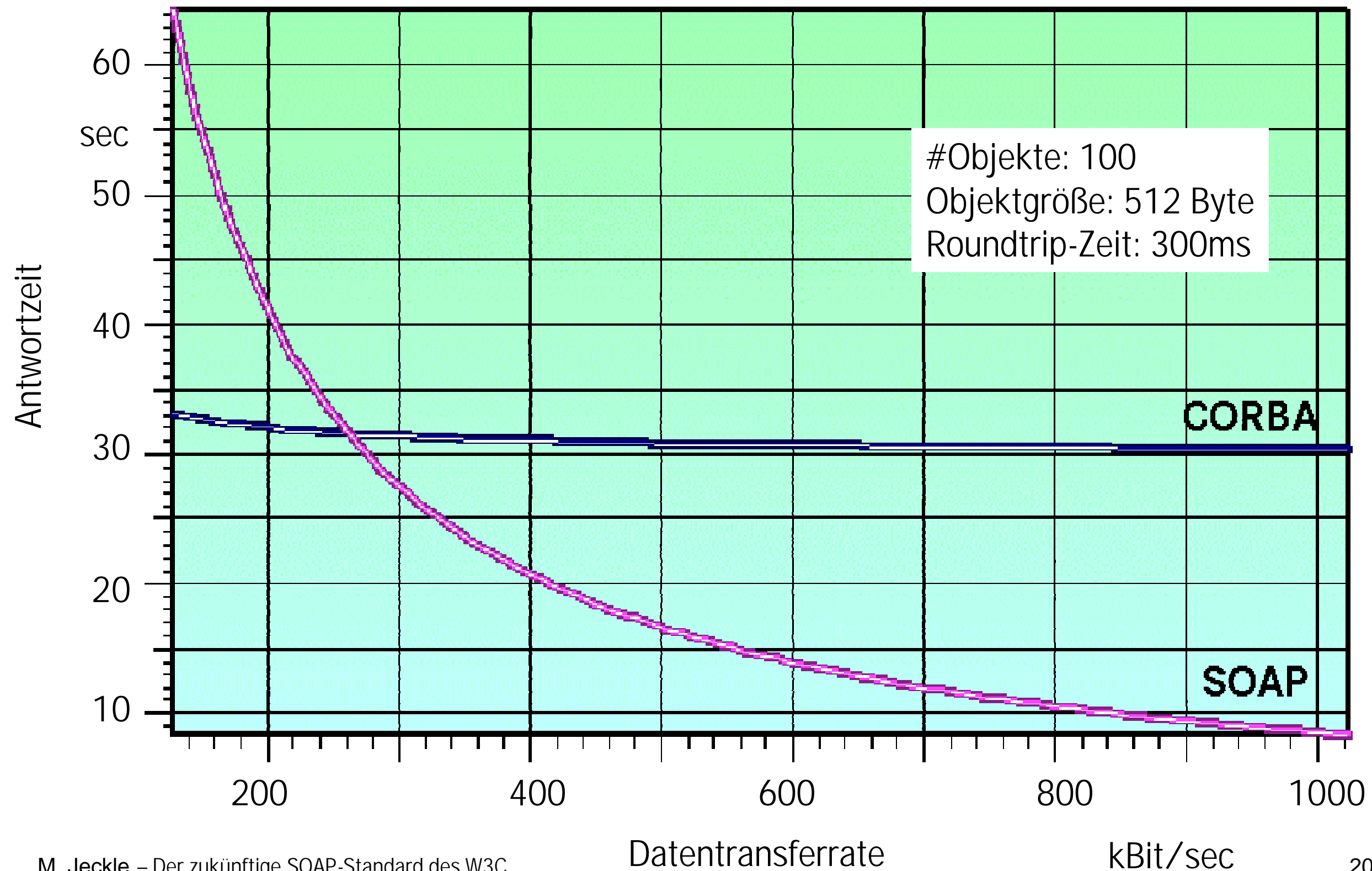
Transparente Verwendung verschiedener
(Transport- oder Sitzungs-)Protokolle

CORBA

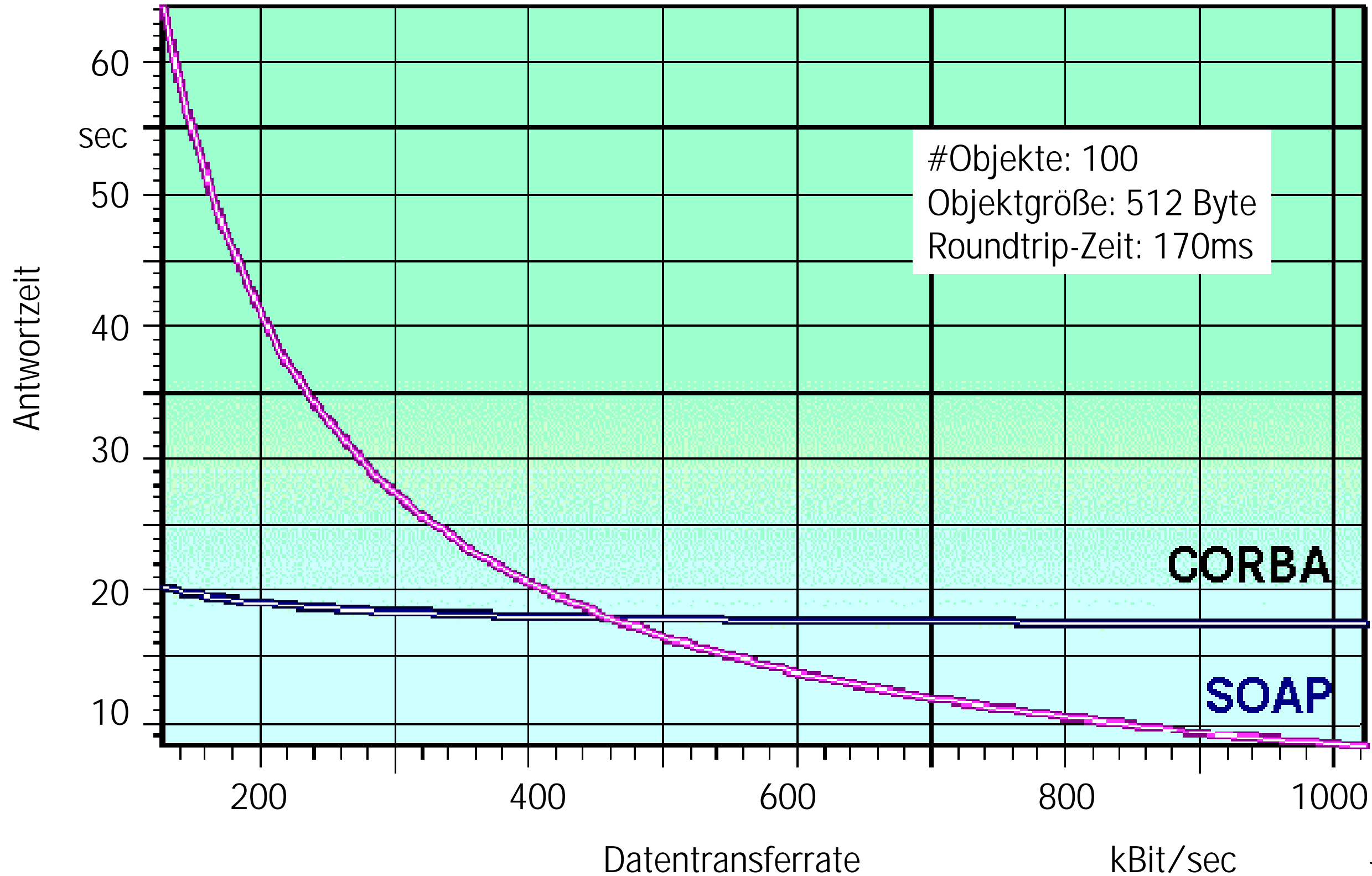
SOAP



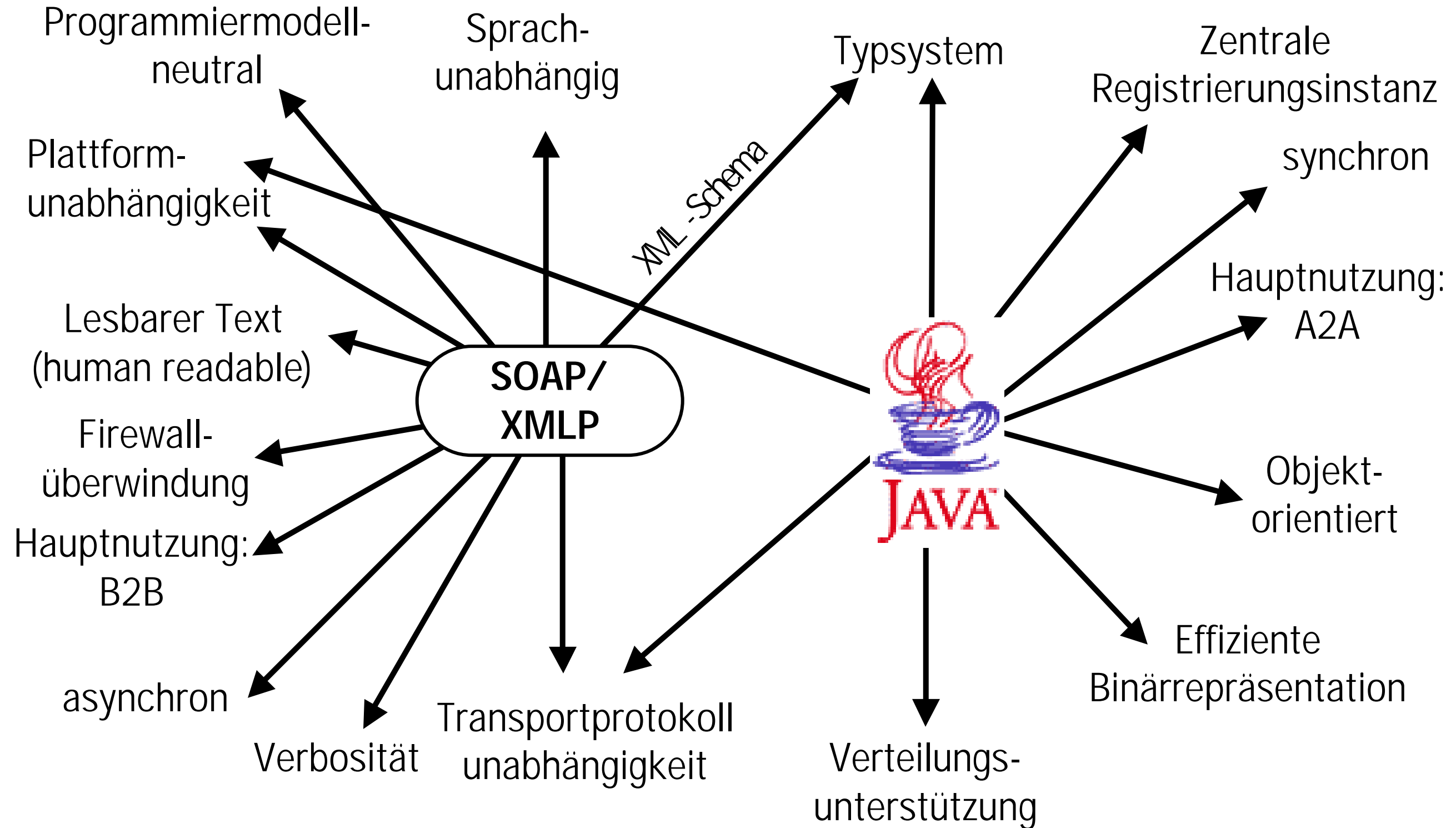
SOAP vs. CORBA – Antwortzeitverhalten (WAN)



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.

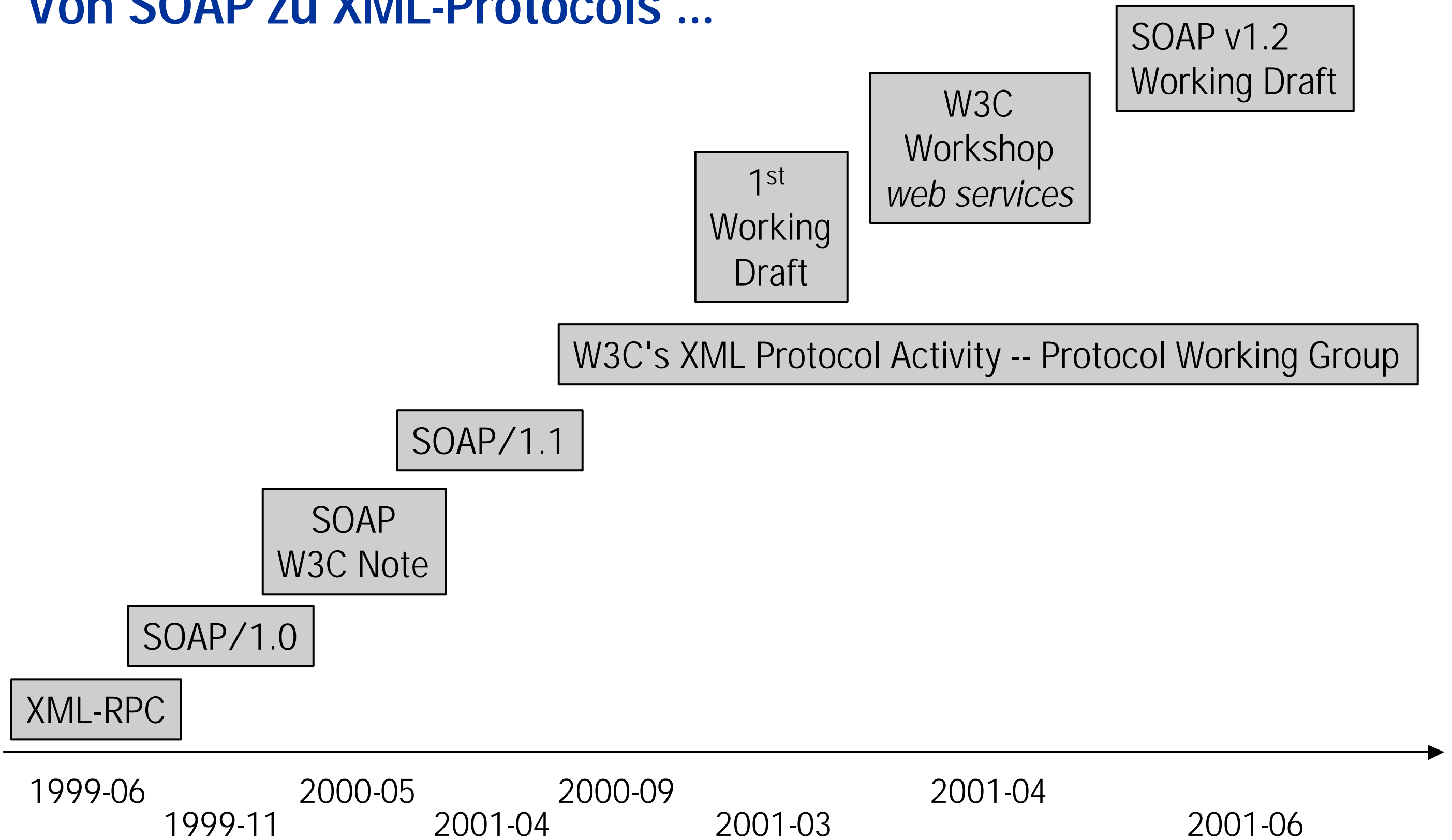
The SOAP specification also mandates an XML vocabulary that is used for representing method parameters, return values, and exceptions."

(DevelopMentor)

Ziele:

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

Von SOAP zu XML-Protocols ...



SOAP v1.2 Spezifikation

Part 0 -- Primer

Einführung und Anwendungsbeispiele (nicht normativ)

Part 1 – Messaging Framework

SOAP envelope und Bindung an Transportprotokoll

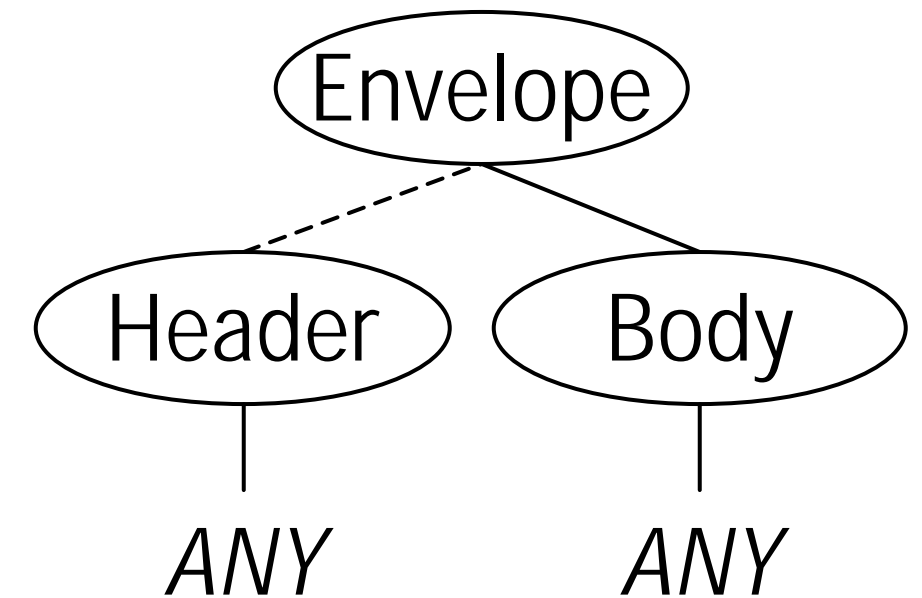
Part 2 – Adjuncts

SOAP encoding, RPC-Konventionen und HTTP-Binding

SOAP-Architektur

SOAP-Envelope

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



SOAP-Transport (*Binding Framework*)

Richtlinien zur Nutzung von Transportprotokollen
(Protokollunabhängigkeit)

SOAP-Codierungsrichtlinien

Mechanismus zur Überführung der Applikationsdatenstrukturen
in XML

SOAP-Nachrichtendarstellung

Richtlinien für die Darstellung aufgerufener Dienste

SOAP und XML

Transportprotokoll (z.B. HTTP)

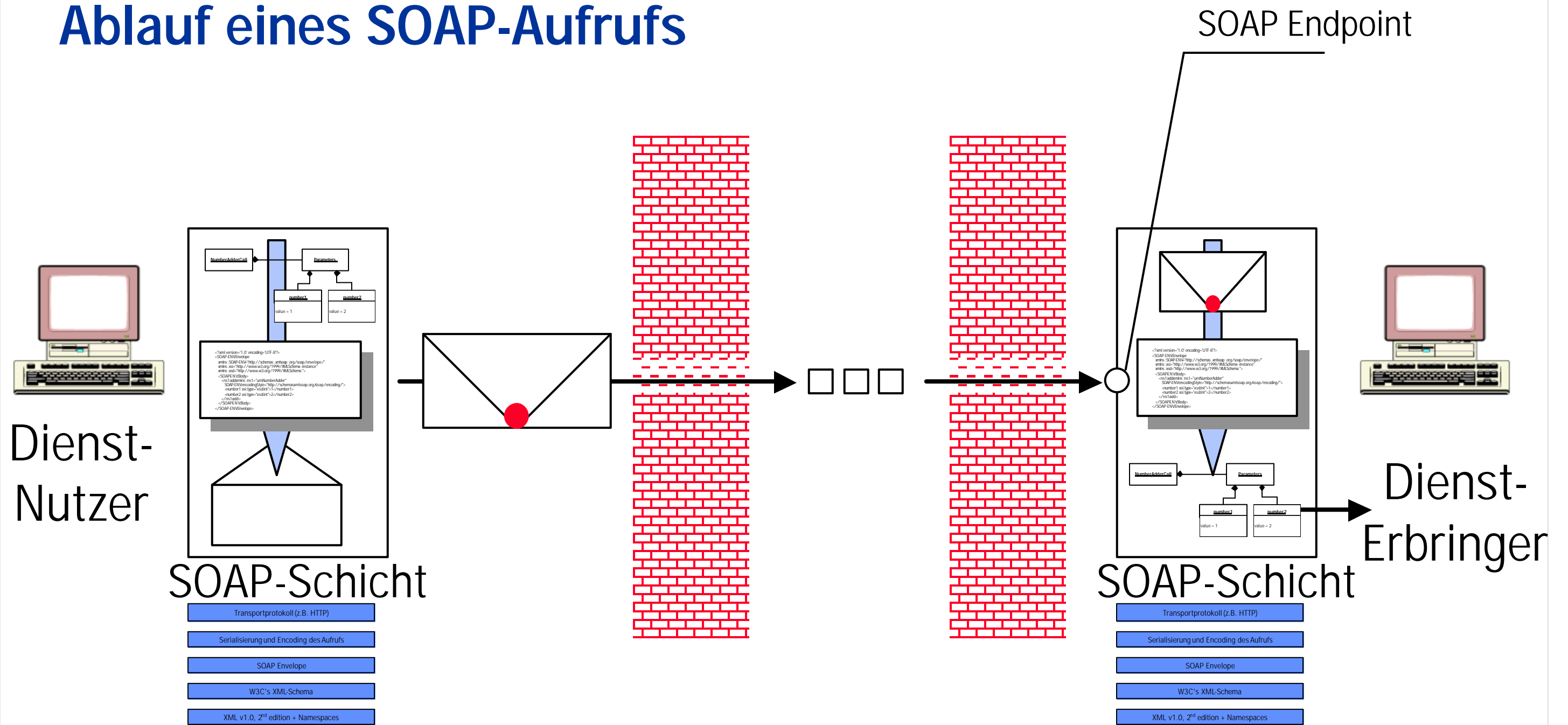
Serialisierung und Encoding des Aufrufs

SOAP Envelope

W3C's XML-Schema

XML v1.0, 2nd edition + Namespaces

Ablauf eines SOAP-Aufrufs



Anforderungen an XML Protocols (SOAP v1.2)

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 XML Protocols: Allgemeines

- Keine Festlegung auf
 - Programmiermodell (Sprache oder Paradigma)
 - Kommunikationsmechanismus (*protocol binding*)
 - Kommunikationsszenarien (*one-way, request-response, RPC, ...*)
- 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 kann mehrere Verarbeitungsanweisungen enthalten
 - Umschlag kann mehrere Nutzinformationsblöcke enthalten

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
- Einfache Einsetzbarkeit in Systemen die bereits XML (insbesondere XML-Schema und XML-Namensräume) unterstützen
- Einsetzbarkeit auf Endgeräten mit beschränkten Ressourcen

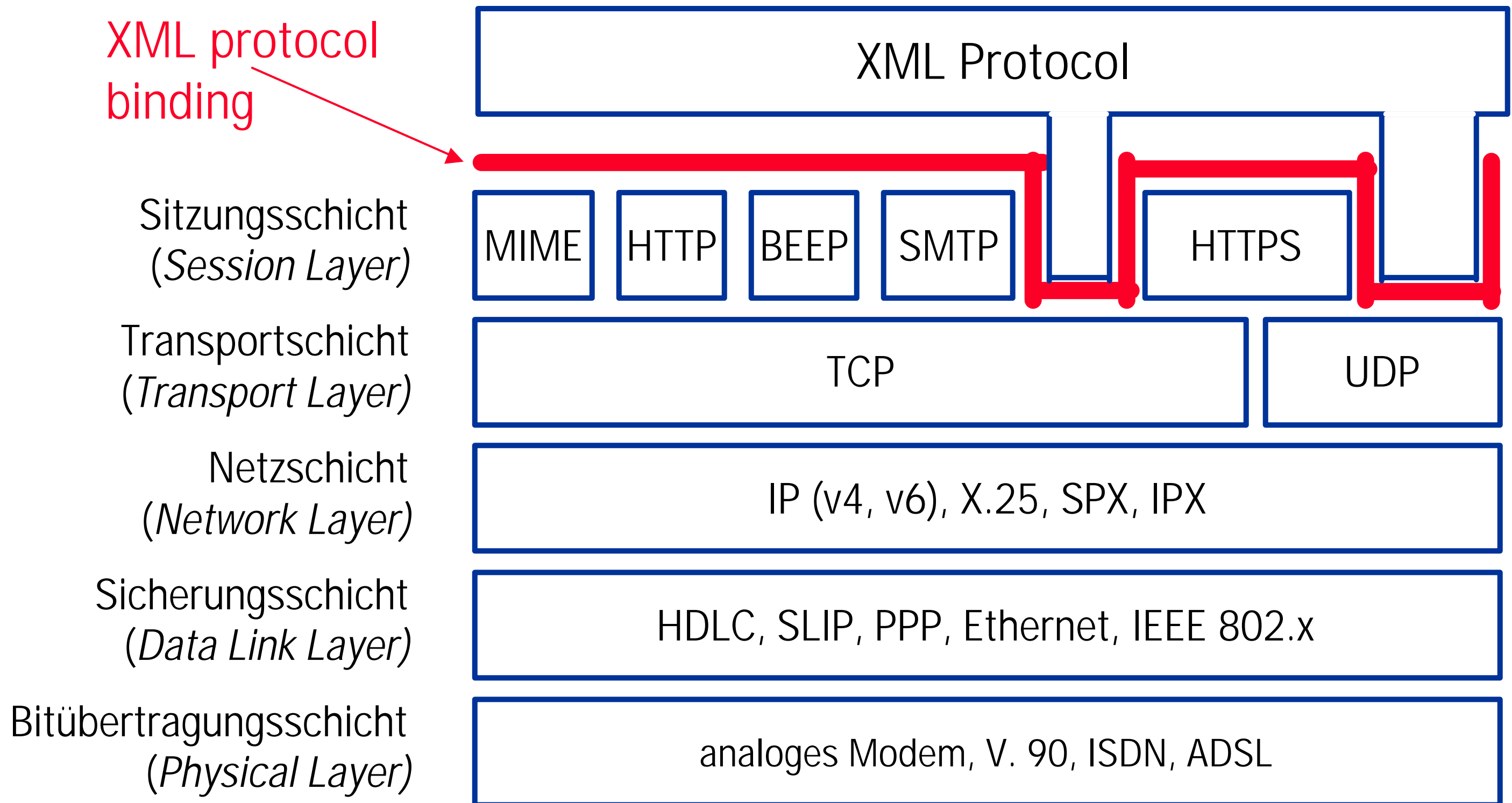
Anforderungen an XMLP: RPC-Vermittler

- Vermittler (*intermediaries / actors*) können verschiedene Rollen auf dem Weg der SOAP-Nachricht zwischen Sender und Empfänger erfüllen
- Standardisierte Rollen:
 - **none**
keine Verarbeitung des Inhalts durch den Prozessor, außer eine andere schon durch ihn verarbeitete Nachricht verweist auf diese
Transportvermittler (*transport intermediary*), operiert als Teil der Bindung an das Transportprotokoll
 - **next**
Verarbeitung des Inhalts durch jeden Prozessor (zwingend)
Verarbeitender Vermittler (*processing intermediary*), vollständiger SOAP-Prozessor
 - **anonymous**
Verarbeitung des Inhalts durch den Ziel-Prozessor

Anforderungen an XMLP: Datenrepräsentation

- Orthogonalität zwischen Datendarstellung (*data representation*) und Dateneinbettung (*data encapsulation*) in eine SOAP-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) encodieren können
- Referenzierung von nicht-serialisierten Daten durch Uniform Resource Identifier (URI)
- Darstellung geschachtelter Arrays

Anforderungen an XMLP, Verwirklichung in SOAP v1.2



Anforderungen an XMLP: Protokoll-Bindung

- HTTP-Binding wird durch XMLP-Arbeitsgruppe erarbeitet
- Beliebige andere Protokollbindungen sind jedoch ohne weiteres möglich
- Mapping auf existierende *Leitungsprotokolle* (z.B.: HTTP, BEEP) kann zu Problemen hinsichtlich Sicherheit, Skalierbarkeit oder Semantik führen
- SOAP-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/1.1-Nachricht

SOAP-Nachricht

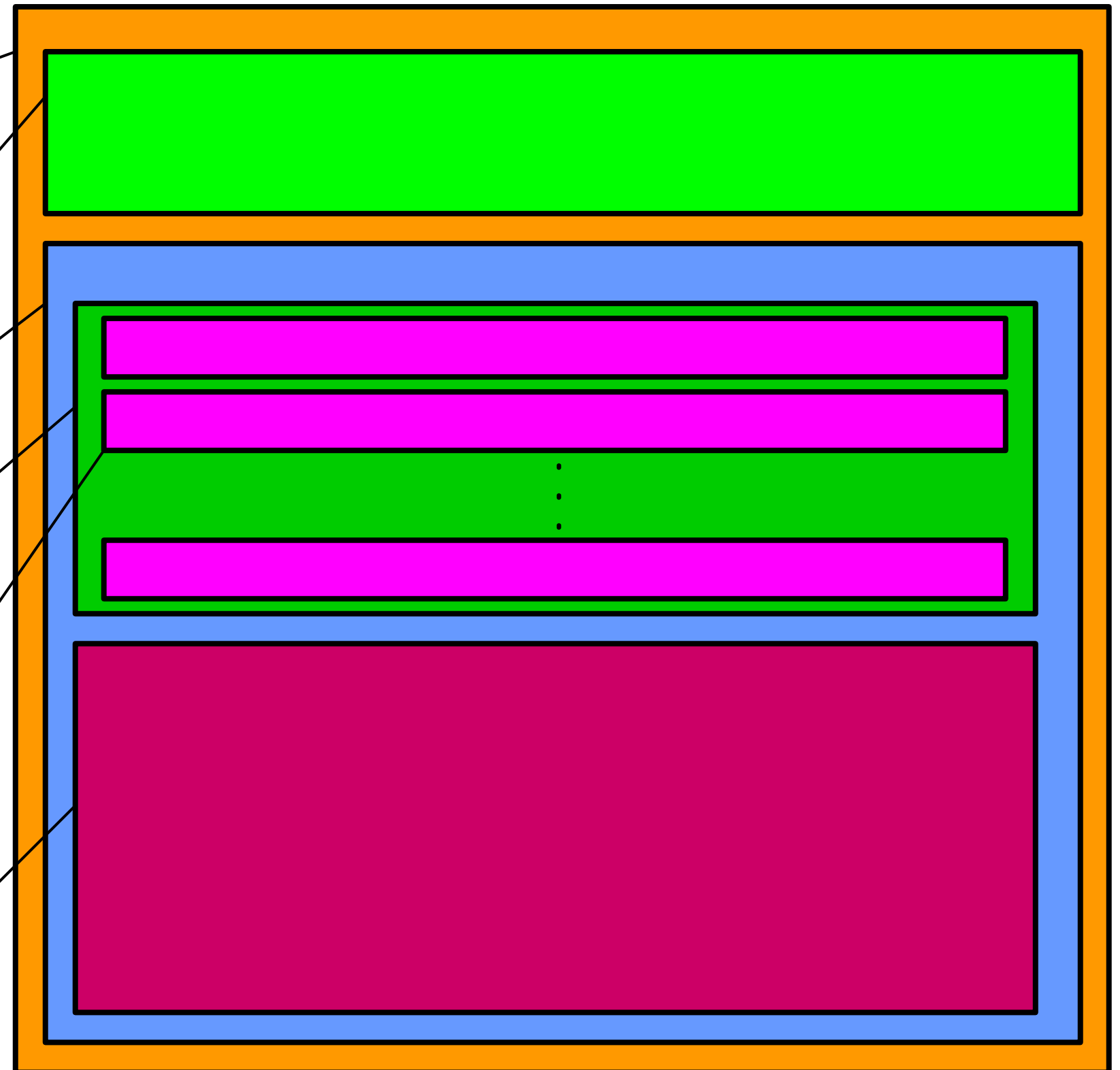
Protokoll-spezifische Information

SOAP-Envelope

SOAP-Header

Individuelle SOAP-Header

SOAP-Body (XML-Nachricht)



Ausblick auf die SOAP-Standardisierung: SOAP v1.2

Anatomie einer XML-Nachricht

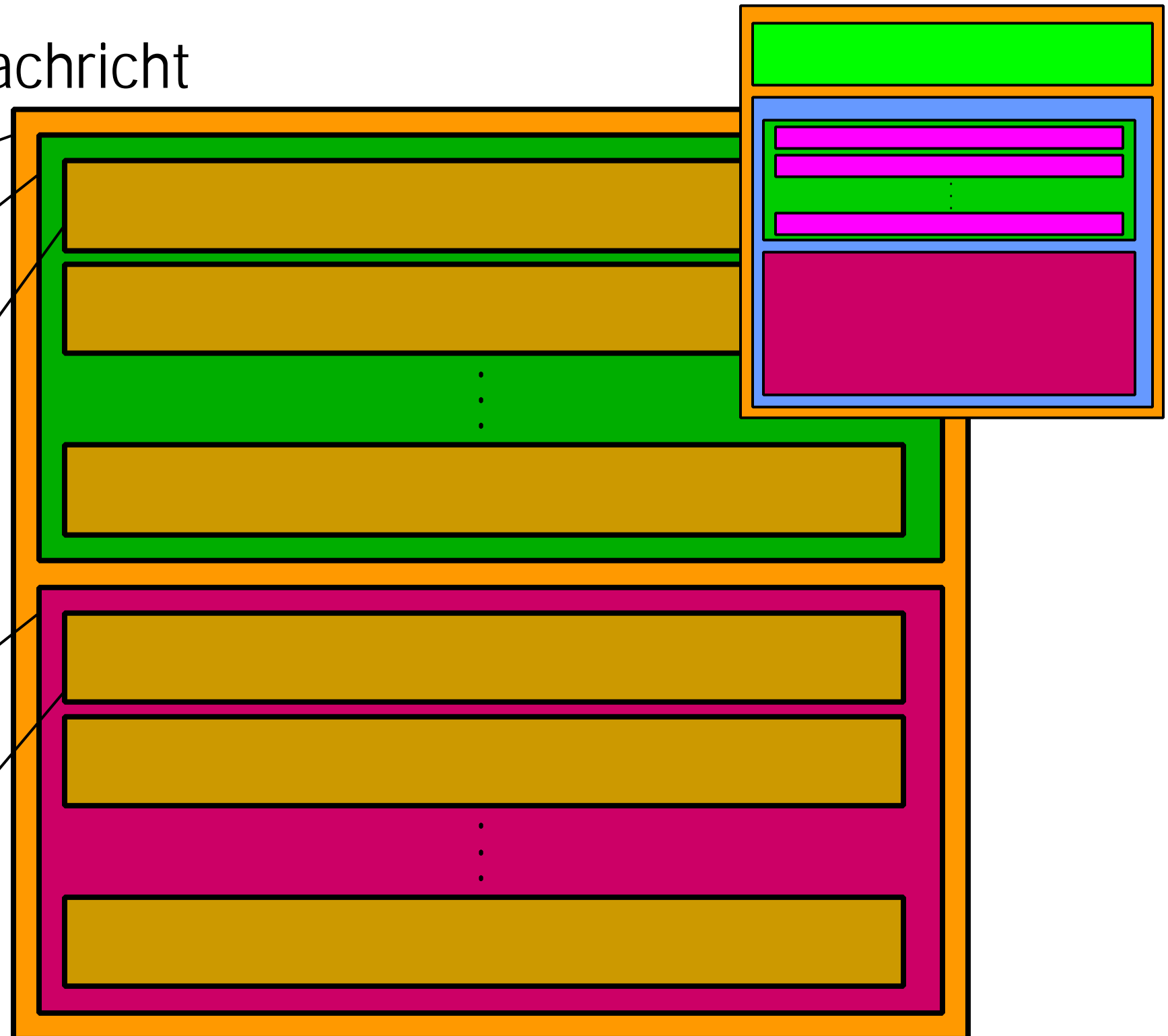
SOAP-Nachricht

SOAP-Header

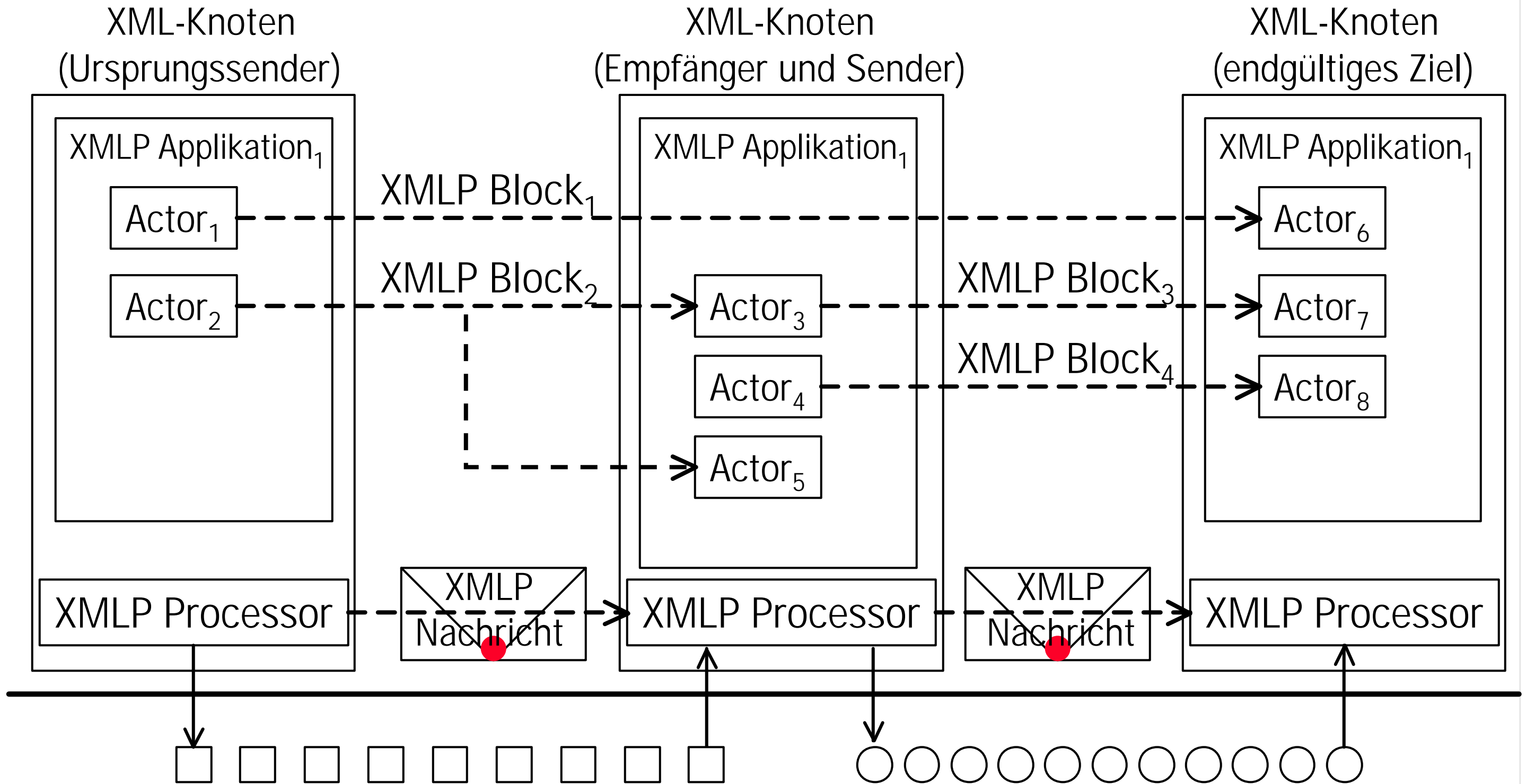
SOAP-Block

SOAP-Body

SOAP-Block



Abstraktes Modell der XML Protocols (XMLP)



Kritik am Ansatz

- XML zur Inhaltsdarstellung
 - zusätzlicher Verarbeitungsaufwand beim Empfänger (XML-Parsing)
 - Informationsdarstellung platzraubender als optimierte Binärformate
- HTTP als Transportprotokoll
 - Probleme bei Transaktionsverarbeitung durch fehlende Zustandsinformation
Lösung: anderes Transportprotokoll verwenden (seit SOAP/1.1 möglich)
- Keine explizite Berücksichtigung von Sicherheit
 - ... nicht im durch SOAP angesprochenen Problembereich
 - Lösung: Nutzung von HTTPS, S-HTTP 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