

Ansätze für eine Web-basierte Initiierung qualitätsbasierter Kommunikationsdienste

Mark Doll

Institut für Telematik

Prof. Dr. Martina Zitterbart

Universität Karlsruhe (TH)

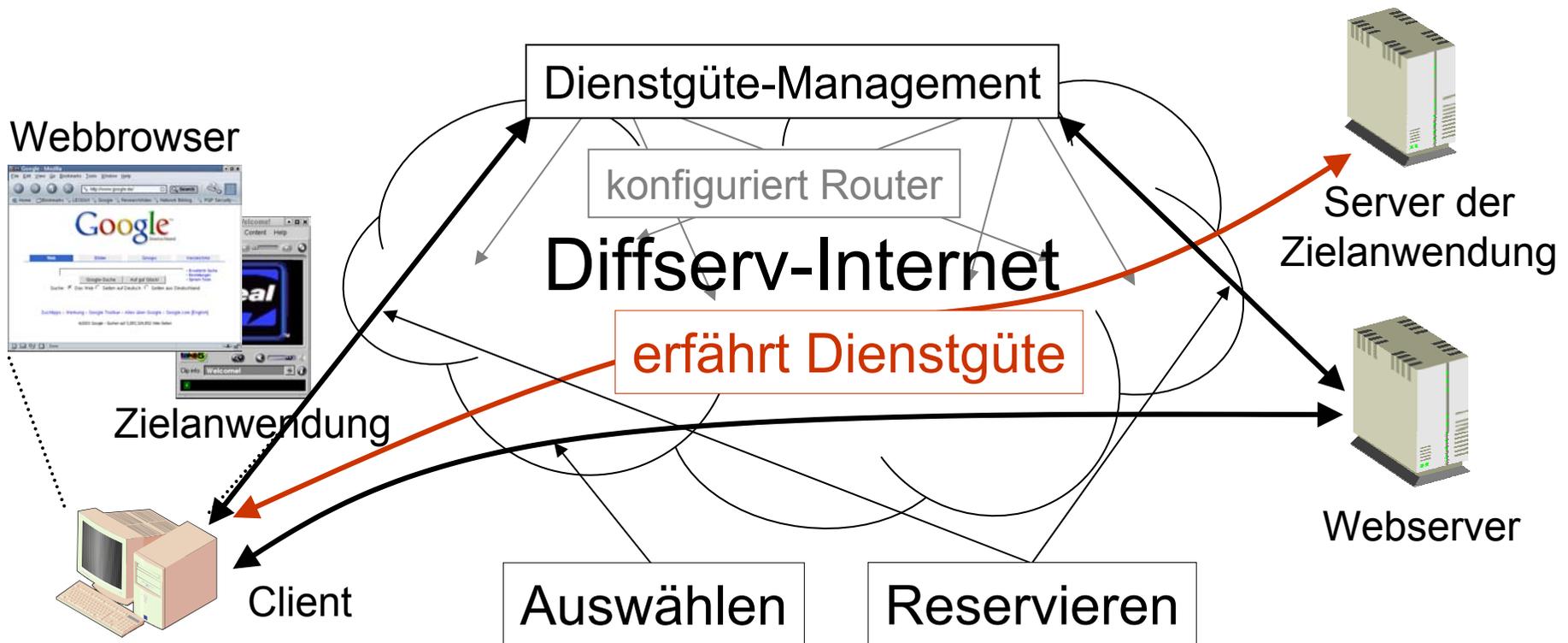
1. Überblick
2. Grundlagen
3. Varianten
4. Probleme
5. Vergleich



1. Überblick



- ❑ Dienstgüte initiieren = Dienstgüte auswählen & Reservierung aufbauen
- ❑ für möglichst viele Anwendungen – ohne diese zu ändern!
 - z. B. Live-Streaming, Online-Spiele, Konferenz-Systeme
- ❑ eine Anwendung modifizieren, über die Anwendungen gestartet werden
 - World Wide Web
 - aber kein QoS für das Web selbst!



- Warum kein QoS für das Web?
 - HTTP ist transaktionsbasiert → burstartiger Verkehr
 - Diffserv bietet (noch?) keine geeigneten Dienste (PHBs/PDBs)

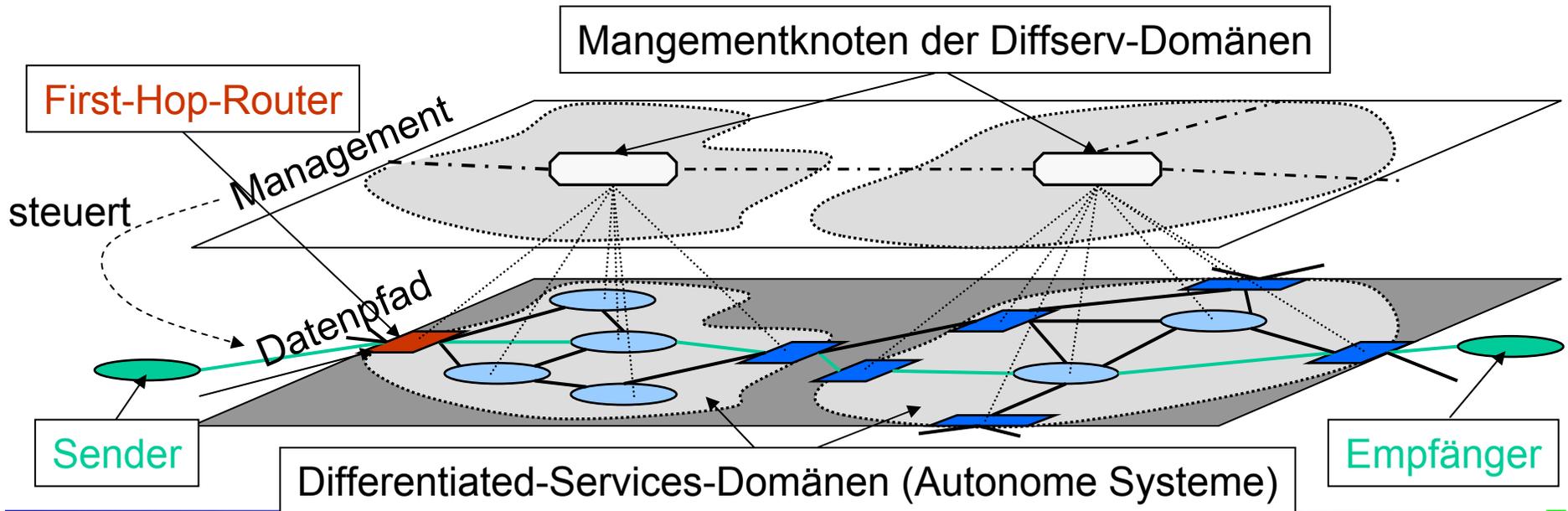
- Warum nicht die Ziellanwendung selbst QoS-fähig machen?
 - langfristig sicher sinnvoll
 - Web-basierte Initiierung ist eine Übergangslösung
 - kurzfristig kaum sinnvoll
 - Dienstgüte-Management nicht standardisiert
 - Dienste nicht standardisiert
 - stark divergierende Vorstellungen in der Internet Community
 - Änderung von Management(schnittstelle) → Änderung jeder QoS-Anwendung



2. Grundlagen



- Differentiated Services im Datenpfad
- Ressourcenmanagement
 - Ende-zu-Ende
 - gesamter Datenpfad **Sender–Empfänger**
 - pro Datenstrom
 - Abgrenzung gegen andere Datenströme (**First-Hop-Router**)
 - garantierte Dienstgüte



- Webseite mit Link (a-Tag mit href-Attribut) auf „Vorschaltdatei“

```
<html>  
<a href="http://www/stream.rm"  
</html>
```

Browser-basierte
Variante

- Browser reserviert entsprechend QoS-Attribut

- Browser fordert Datei vom Webserver an

```
GET http://www/stream.rm HTTP/1.1
```

- Server reserviert entsprechend lokal gespeicherter QoS-Informationen (je Vorschaltdatei ein Datensatz)

- Server antwortet mit Dateiinhalt

```
HTTP/1.1 200 OK  
Content-Type: application/vnd.rn-realmedia  
[...]  
rtsp://streamserver/cam3/front
```

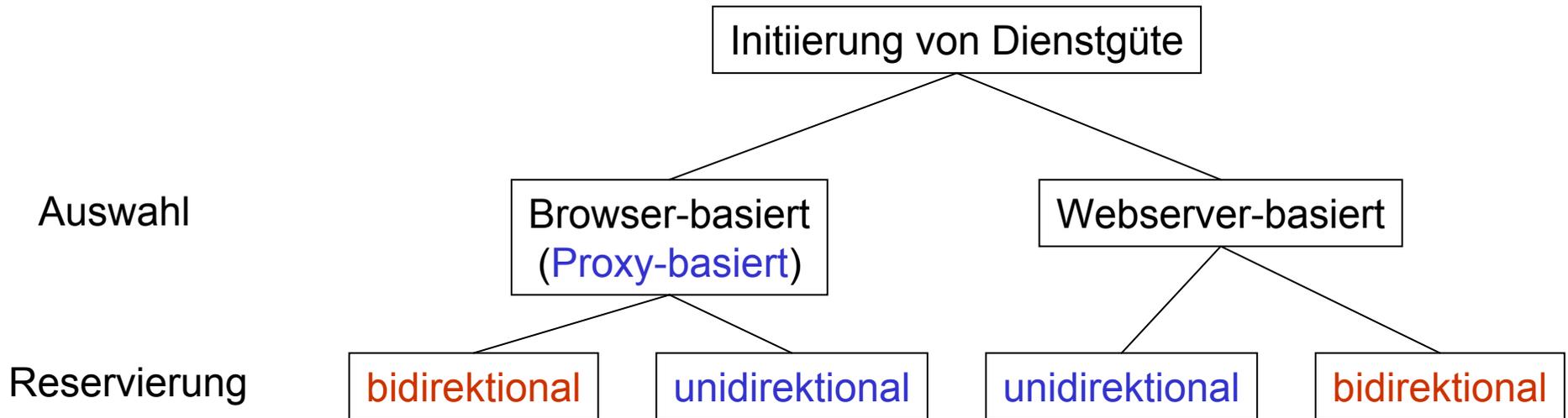
Webserver-basierte
Variante

- bestimmt durch Mime-Type im Content-Type:-Header startet Browser Zielanwendung



3. Varianten





- Rolle des Browsers durch (clientseitigen) HTTP-Proxy ersetzbar
 - Dienstgüteunterstützung Strecke Proxy–Client anderweitig; vgl. [Tagungsband](#)
- unidirektional: jeder kann nur für die eigene Senderichtung reservieren
 - Webserver Downstream (Daten), Browser Upstream (ACKs); vgl. [Tagungsband](#)
- bidirektional: beide Richtungen (durch Browser *oder* Webserver)
 - vereinfacht Initiierung deutlich
 - Dienstgüte-Management muss bidirektionale Reservierungen unterstützen



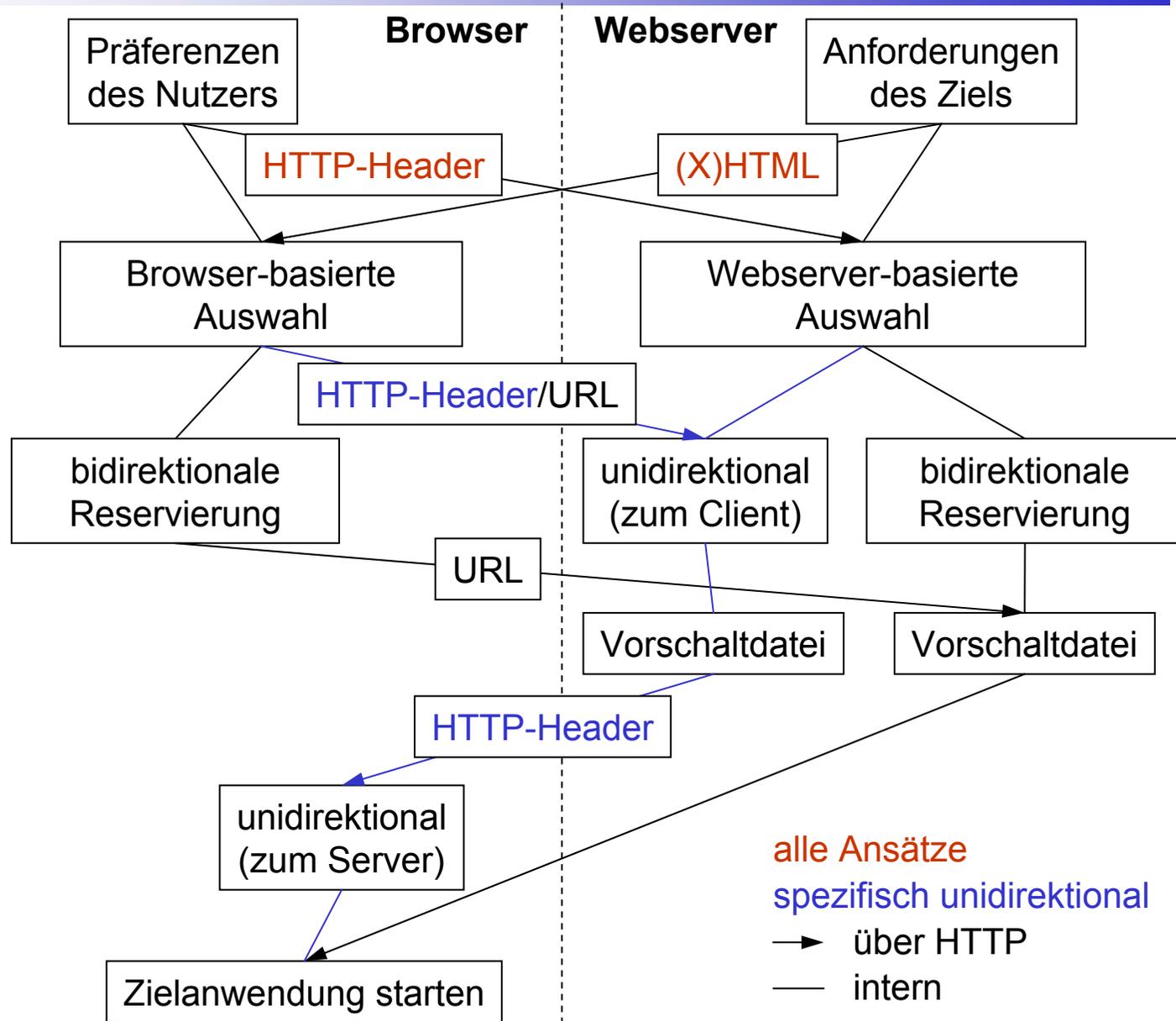
Transport von

Anforderungen
Präferenzen

Ergebnis
Auswahl

Ergebnis
Auswahl

Ergebnis
Reservierung



Neue Attribute für das a-Tag `<a href="http://tm.uka.de/cam.rm" ...`

□ Topologische Information

- Zieladresse gleich Browseradresse
- Quelladresse aus href-Tag bekannt
- optional Attribut mit expliziter Quelladresse (oder Hostname)

`... qossrc="tm.uka.de" ...`

□ Dienstspezifikation

- Attribut Dienst, Bandbreite, Verzögerung, Wahrscheinlichkeit
- Syntax wie Mime-Type mit Parametern

`... qosservice="qos/af;r=2000000-8000000;d=.3;p=.99" ...`

hier: Assured Forwarding, 2-8 MBit/s, max. 300ms One-way Delay für 99% aller Pakete



- Attribut mit ID zur Gruppierung von a-Tags
 - Semantik: alle Links mit derselben ID führen zum selben Inhalt (aber mit unterschiedlicher Qualität)
 - ... `qosid="group23"` ...
 - Dienstbeschreibungen aller Links in einer Reservierungsanfrage zusammenfassen

- Attribut mit Substitutionsvorschrift zum Überschreiben des href-Attributs
 - erweiterter regulärer Ausdruck nach POSIX
 - Angewendet auf Antwort (in Mime-Type Syntax) vom Dienstgüte-Management
 - neuer Link mit Substring Backreferences \1 bis \9 (entlehnt von sed)
 - ... `qoshref="@ (r=[0-9]*) @http://tm.uka.de/cam.rm?\1@"` ...

z. B. 5 MBit/s vom Management reserviert, Antwort: qos/af;r=5000000;d=.42;p=.98 enthält Substring „r=5000000“ → \1; neuer Link: http://tm.uka.de/cam.rm?r=5000000

- Hintergrund
 - `qosid`-Attribut ermöglicht Liste von Bandbreiten
 - z. B. für einen mit verschiedenen Raten kodierter Videostream
 - `qoshref`-Attribut ermöglicht kontinuierlichen Bandbreitenbereich



- Behandlung unbekannter Elemente/Tags und Attribute
 - HTML 4.01: um Experimentieren und Interoperabilität zu erleichtern, wird **empfohlen**, dass User-Agents...
 - XHTML 1.0: konforme User-Agents **müssen**...

Inhalt unbekannter *Elemente*
darstellen (HTML) bzw. *auswerten* (XHTML)

unbekannte *Attributspezifikationen*
(d. h. Attribut und Wert) *ignorieren*

`<element attribut="Wert">Inhalt des Elements</element>`

Element: `<a ...>...`, Start-Tag: `<a ...>`, End-Tag: ``

- Quellen
 - <http://www.w3.org/TR/1999/REC-html401-19991224/appendix/notes.html#h-B.1>
 - <http://www.w3.org/TR/2002/REC-xhtml1-20020801/#uaconf>



- ❑ Warum nicht einfach erweitern wie bei HTML?
 - wäre wohlgeformt (well-formed) aber nicht mehr gültig (valid)
 - eigene DTD (Document Type Definition) für QoS-Erweiterungen
- ❑ User-Agents (Browser)
 - validierende User Agents lehnen erweitertes XHTML ohne DTD ab
 - aber wie behandeln nicht-validierende User-Agents Namespaces?
 - aufgrund Performance Mehrzahl (alle?) Browser nicht-validierend

Alternativen

- ❑ neue Attribute für a-Tag in eigenen Namespace (für Attribute)

```
<html xmlns="http://www.w3.org/1999/xhtml
      xmlns:qos="http://diffserv.org/tr/2003/4xhtml1">
  <a href="..." qos:src="..." qos:service="..."
    qos:href="...">...</a>
</html>
```

- ❑ neues a-Tag und Default-Namespace (für Elemente) wechseln

```
<a xmlns=http://diffserv.org/tr/2003/4xhtml1
  href="..." qossrc="..." qosservice="..." qosid="...">...</a>
```



4. Probleme



- Abgrenzung gegen andere Datenströme
 - erfordert Kenntnis des vollständigen Sockets (=5-Tupel Quell-/Ziel-IP & -Port, Protokoll)
 - Client der Zielanwendung verwendet „ephemeral Port“
 - erst bekannt nachdem Anwendung Socket öffnet
 - Client der Zielanwendung öffnet mehrere Sockets
 - für welchen soll die Dienstgüte gelten?
- Abbau der Reservierung

Lösung:

- Zielanwendung muss überwacht werden
 - durch Browser bzw. Webserver Client- bzw. Serverseite
 - zum Einschränken der Reservierung, nachdem ephemeral Port bekannt
 - muss vom Dienstgüte-Management unterstützt werden
 - eine Reservierung für alle Ports
 - muss vom Dienstgüte-Management unterstützt werden
 - zum Abbauen der Reservierung
 - nachdem Zielanwendung (alle) Socket(s) geschlossen hat
 - nach einem Timeout, wenn Verkehr ausbleibt

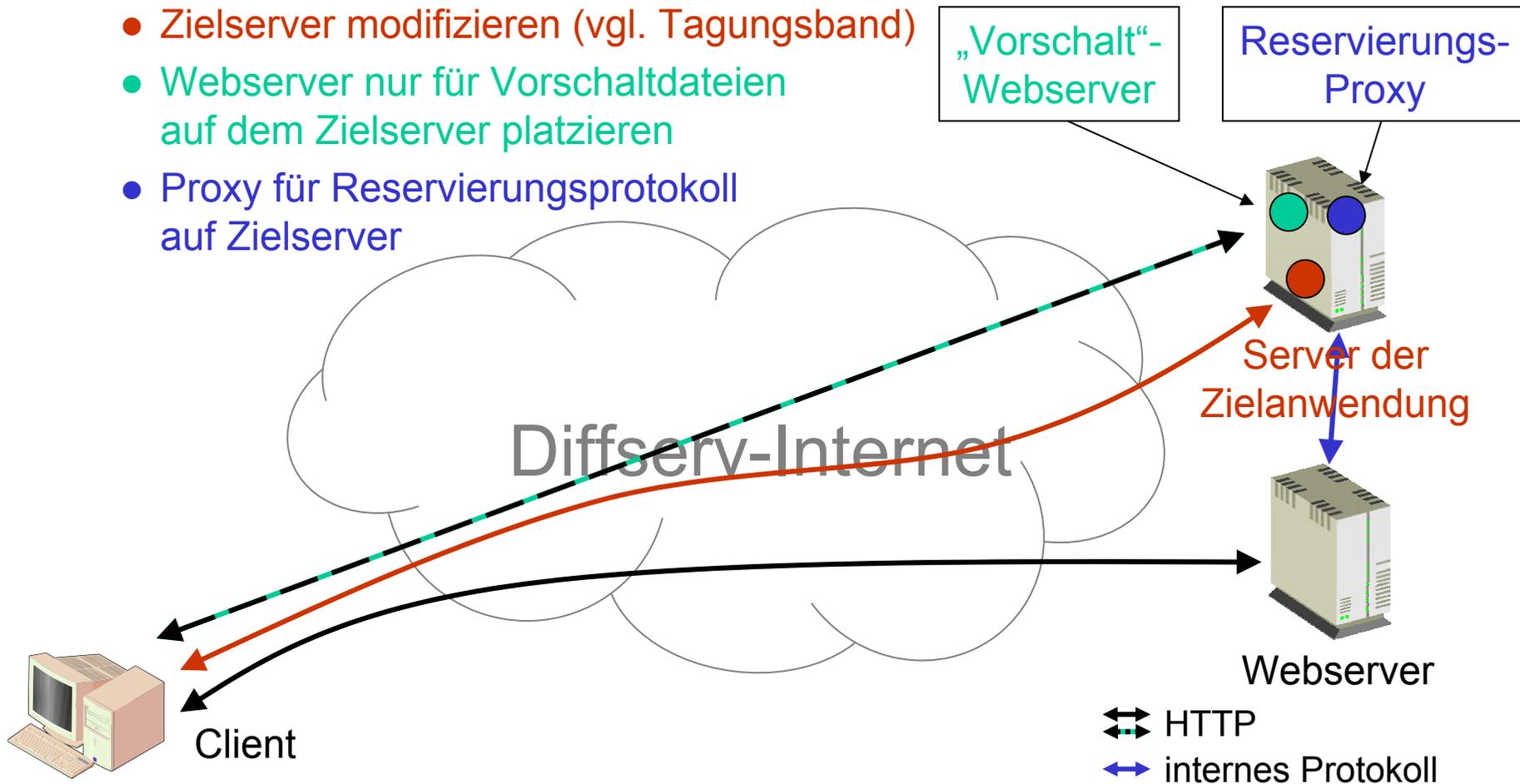


□ Problem

- Webserver und Server der Zielanwendung getrennt
→ kein Zugriff auf Datenverbindung der Zielanwendung

□ Lösungsalternativen

- Zielserver modifizieren (vgl. Tagungsband)
- Webserver nur für Vorschaltdateien auf dem Zielserver platzieren
- Proxy für Reservierungsprotokoll auf Zielserver



- Browser benutzt HTTP-Proxy(s)
 - erkennbar an Via:-Header
 - Webserver benötigt Clientadresse zur Reservierung
- Nutzer möchte Einfluss nehmen
 - Min. und/oder Max. Rate
- Lösung
 - Browser fügt spezielle HTTP-Header ein
 - erfordert modifizierten Browser
 - mit seinem Hostnamen oder IP-Adresse
 - Mit QoS-Präferenzen

```
GET http://tm.uka.de/chooser.html HTTP/1.1
QoS-Client: surfer.tm.uka.de
Accept-QoS: qos/af;r=0-2500000, qos/ef
[...]
```



5. Vergleich



□ Vergleich

Varianten	Eigenschaften		Überwachung	QoS-Daten auf Gegenstelle	Einschränkung für Nutzer
	Modifikationen Web-server	Browser			
Browser-basiert	+	-	○	-	+
Webserver- basiert	modifizierter Browser	○	-	○	○
	Legacy-Browser	-	+	-	-
unidirektional Webserver-/Browser-basiert	-	-	-	-	○/+

□ Interessanteste Varianten

**Browser-basiert
bidirektional**

- + keine Modifikation an (sensiblen) Servern
- abhängig von QoS-Daten in (X)HTML-Quellen auf Server

**Webserver-basiert
bidirektional, Legacy-Browser**

- + einzig Webserver verändert, QoS-Daten bleiben Webserver-intern
- keine HTTP-Proxys (oder Reservierung endet dort)



Beteiligte Studenten:

Marek Tomczyk

Modifikationen am Webserver Apache

Andreas Kunz

Modifikationen am Browser Mozilla

Vielen Dank!

Beitrag online: <http://tm.uka.de/doc/2003/doll-kivs2003-web-initiated-qos.pdf>

Folien online: <http://tm.uka.de/doc/2003/doll-kivs2003-web-initiated-qos-slides.pdf>

