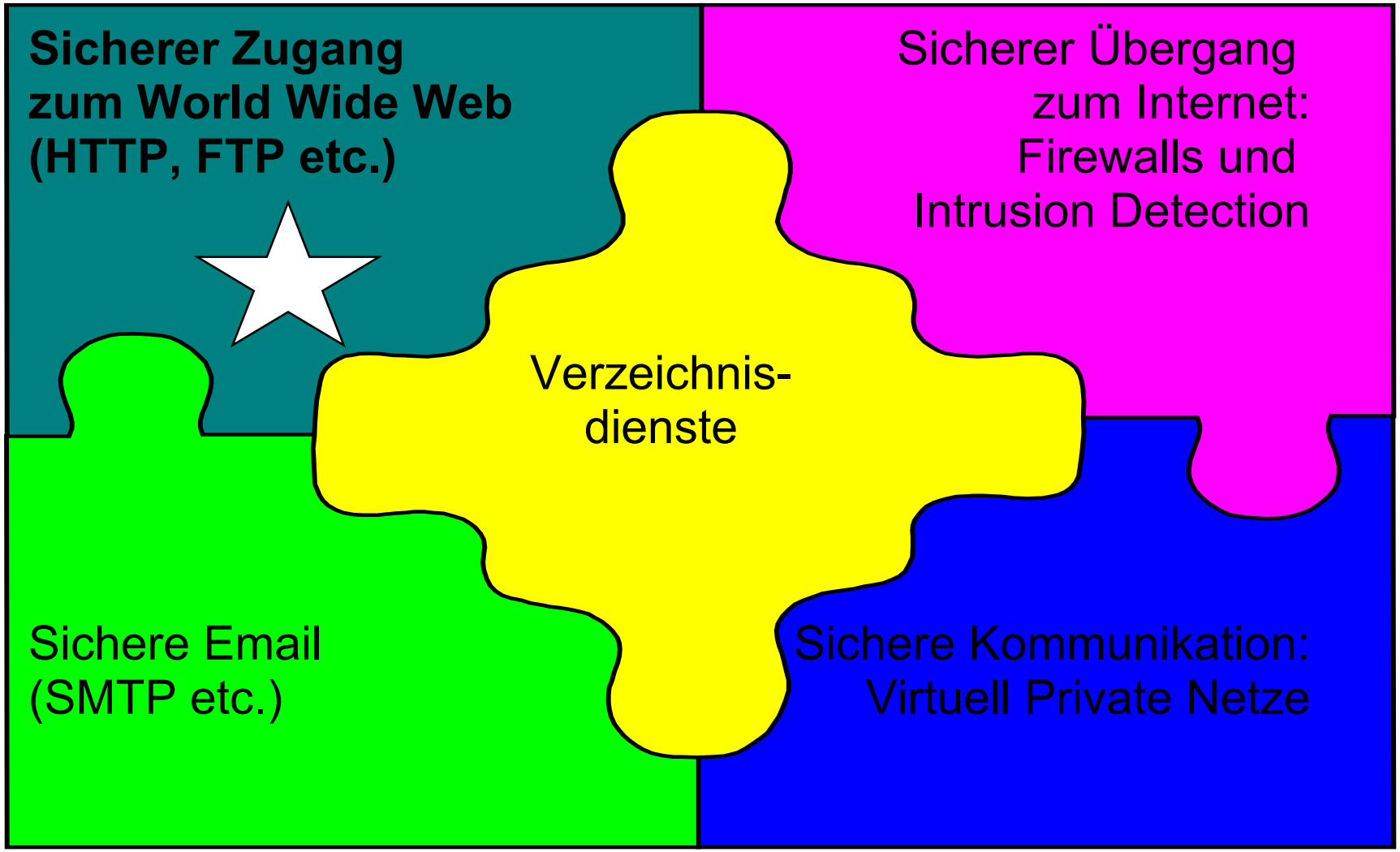
Integrierte IT-Service-Management-Lösungen anhand von Fallstudien

„Web-Zugang und Internet Sicherheit“

Dr. Stephen Heilbronner et al.
Prof. Dr. Heinz-Gerd Hegering

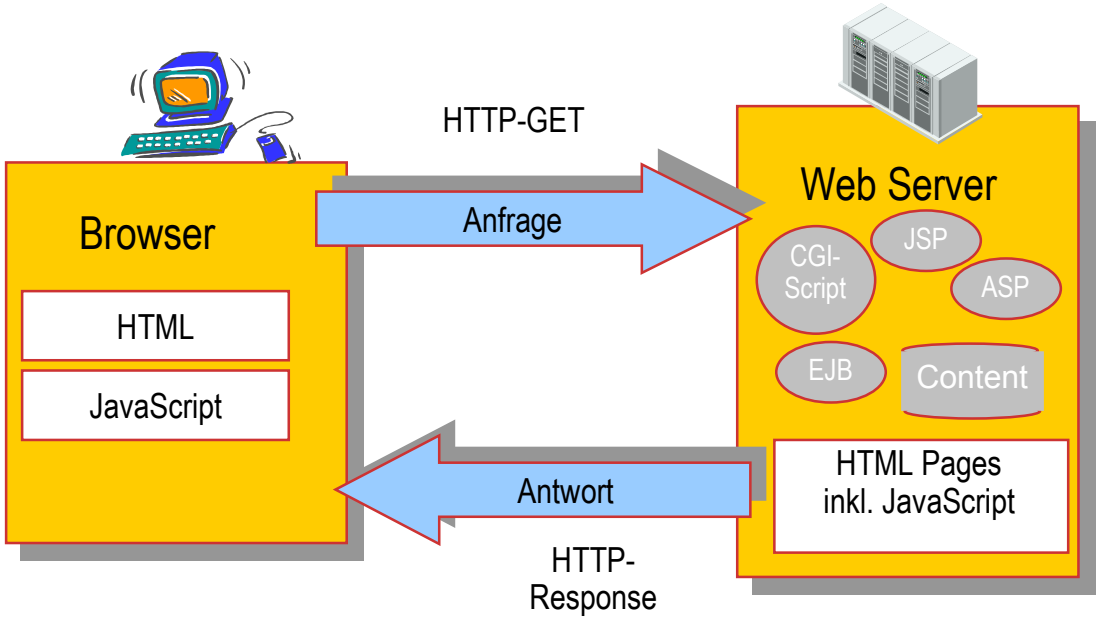
SoSe 2007

Sicherheitsdienste großer IT-Infrastrukturen => Überblick



Web-Zugang
Grundprinzip

- Zugriff auf WWW-Server durch WWW-Clients:
  - 1. Browser
  - 2. andere, „automatische“ Programme



- Formate:
  - Nicht nur HTML !
  - Auch: WML, XML, beliebige Dateiformate

Web-Zugriff

Veränderte Nutzuung des HTTP-Protokolls

- Ursprüngliche Verwendung von HTTP:
  - Übertragung statischer HTML-Seiten bzw. Dateien
  - keine Unterscheidung zw. Anfragenden
- Heutige Web-Zugriffe sind....
  - ... nicht nur mehr vom Typ „Request/Response“.
  - finden meist in einem längerdauerndem Kontext („Session“) statt, z.B. zur
    - Individualisierung der ansonsten anonymem Anfragenden bzgl.
      - Spracheinstellungen
      - „Theming“ der Webseite (Layout-Einstellungen)
    - Identifizierung / Authentisierung
      - „Zuordnung langlebiger Merkmale/Ressourcen“

Web-Zugang
Entwicklung des HTTP Protokolls

- „Hypertext Transfer Protocol“
  Protokoll-Typ: „Request/Response“
- HTTP 1.0 nur gedacht nur für „kurze“ Verbindungen:
  ⇒ 3-Way TCP-Handshake beim Verbindungsaufbau aufwendig
- HTTP 1.1 lässt die TCP-Verbindung nach Übertragung persistent bestehen:
  ⇒ Keine Handshakes beim Abbau/Wiederaufbau
  - Effizientere und schnellere Übertragung kleiner Informationsmengen (⇒ weniger Verzögerungen (latency))
- Integration mit HTTP Proxies:
  - Zieladresse nicht mehr direkt adressiert mit TCP/IP
  - Alle Adressierungsinfo daher nur im HTTP-Header

Weitere Literatur:
http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol
http://en.wikipedia.org/wiki/HTTP_cookie

Web-Zugang

„Web Sniffer“

For more information on HTTP see RFC 2616

HTTP(S)-URL: http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol Submit (IDN allowed)

HTTP version: ◉ HTTP/1.1 ○ HTTP/1.0 (with Host header) ○ HTTP/1.0 (without Host header)

☐ Raw HTML view ☑ Accept-Encoding: gzip • Request type: ◉ GET ○ POST ○ HEAD ○ TRACE

User agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1) Gecko/20060128 SeaMonkey/1.0 Web-Sniffer/1.0.24

HTTP Request Header

```
Connect to 145.97.39.155 on port 80 ... ok

GET /wiki/Hypertext_Transfer_Protocol HTTP/1.1[CRLF]
Host: de.wikipedia.org[CRLF]
Connection: close[CRLF]
Accept-Encoding: gzip[CRLF]
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5[CRLF]
Accept-Language: de,en-us;q=0.7,en;q=0.3[CRLF]
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7[CRLF]
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1) Gecko/20060128 SeaMonkey/1.0 Web-Sniffer/1.0.24[CRLF]
Referer: http://web-sniffer.net/[CRLF]
[CRLF]
```

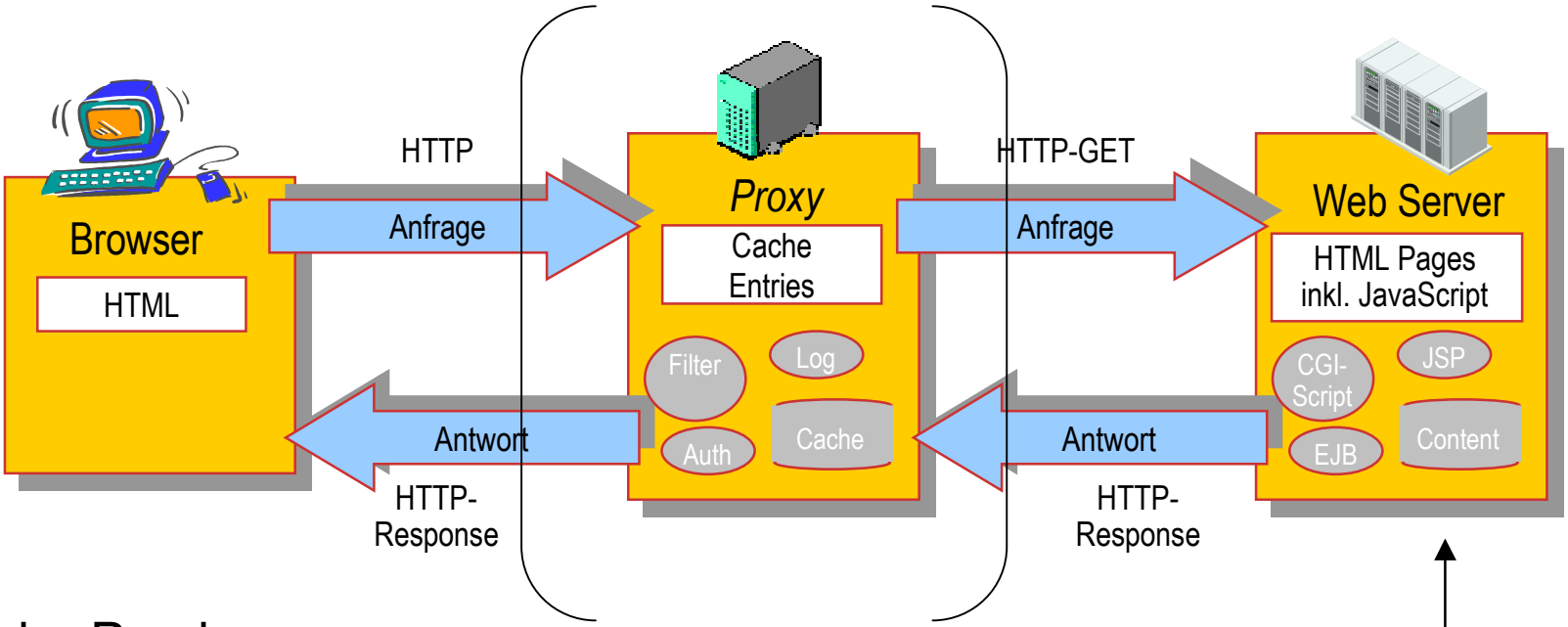
HTTP Response Header

Name	Value	Delim
HTTP Status Code: HTTP/1.0 200 OK		
Date:	Mon, 01 May 2006 20:02:18 GMT	CRLF
Server:	Apache	CRLF
X-Powered-By:	PHP/5.1.2	CRLF
Content-Language:	de	CRLF
ETag:	W/"dewiki:pcache:idhash:2179-0!1!0!0!!ldel2--20060428101018"	CRLF
Vary:	Accept-Encoding,Cookie	CRLF
Cache-Control:	private, s-maxage=0, max-age=0, must-revalidate	CRLF
Last-Modified:	Fri, 28 Apr 2006 10:10:18 GMT	CRLF
Content-Encoding:	gzip	CRLF
Content-Type:	text/html; charset=utf-8	CRLF
X-Cache:	MISS from srv7.wikimedia.org	CRLF
X-Cache-Lookup:	MISS from srv7.wikimedia.org:80	CRLF
X-Cache:	MISS from fuchsia.knams.wikimedia.org	CRLF
X-Cache-Lookup:	MISS from fuchsia.knams.wikimedia.org:80	CRLF
Connection:	close	CRLF

Content (encoded: 13.68 KiB / decoded: 48.04 KiB)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de" dir="ltr">
  <head>
```

Web-Zugang
Web-Architektur mit Proxies (HTTP)



In der Praxis:

- Kette oft mehrfach wiederholt: „Proxy Chaining“
- Großzügige Proxy-Auslegung wichtig für:
  - Multimedia-Streaming in Echtzeit
  - „Pre-pushed content“

Mehr zu dessen Architektur im Juni

Web-Zugang
Proxies (für HTTP)

- HTTP ist
  - entweder anonym, oder
  - Authentisierungs-Information im Header

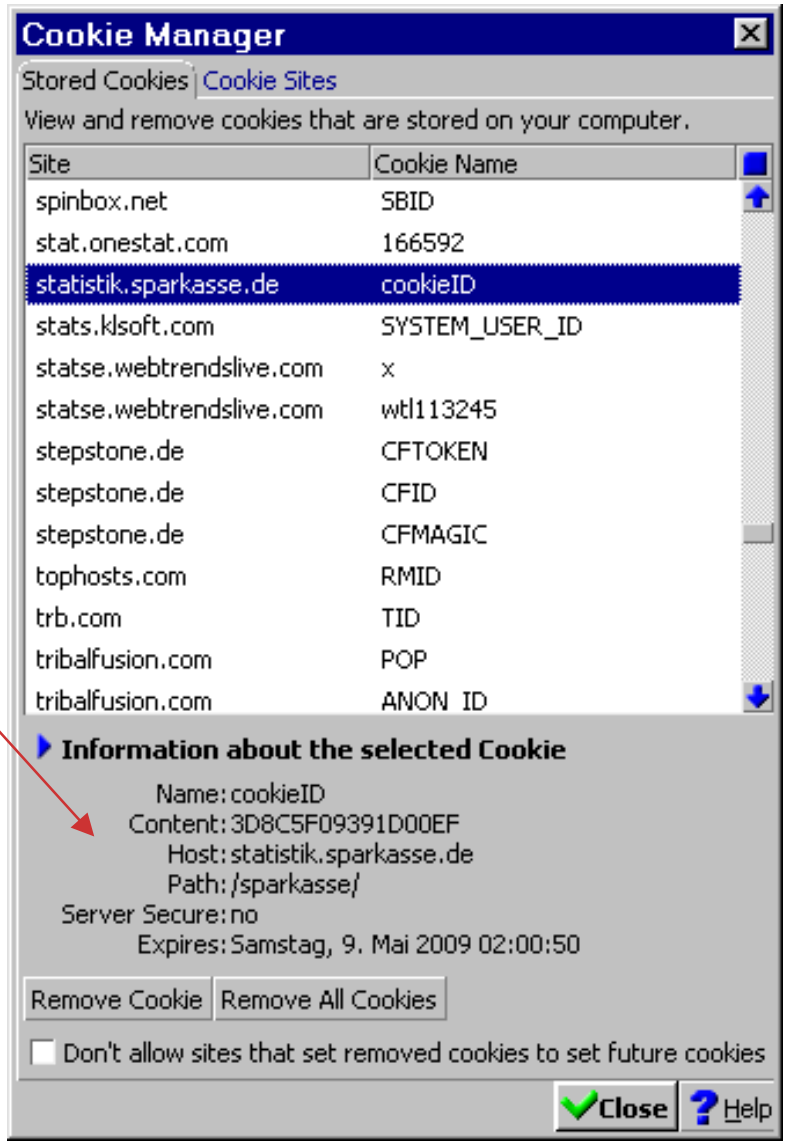
- Autorisierende Proxies unterbrechen den FIuß zum Server
  - erscheinen dem Client genauso wie geschützte Server
  - verlangen Authentisierung vom Benutzer
  - filtern diese vor Weitergabe wieder heraus
    - echter Server benötigt dann eventuell weitere Autorisierung

Web-Zugang Exkurs: AAA

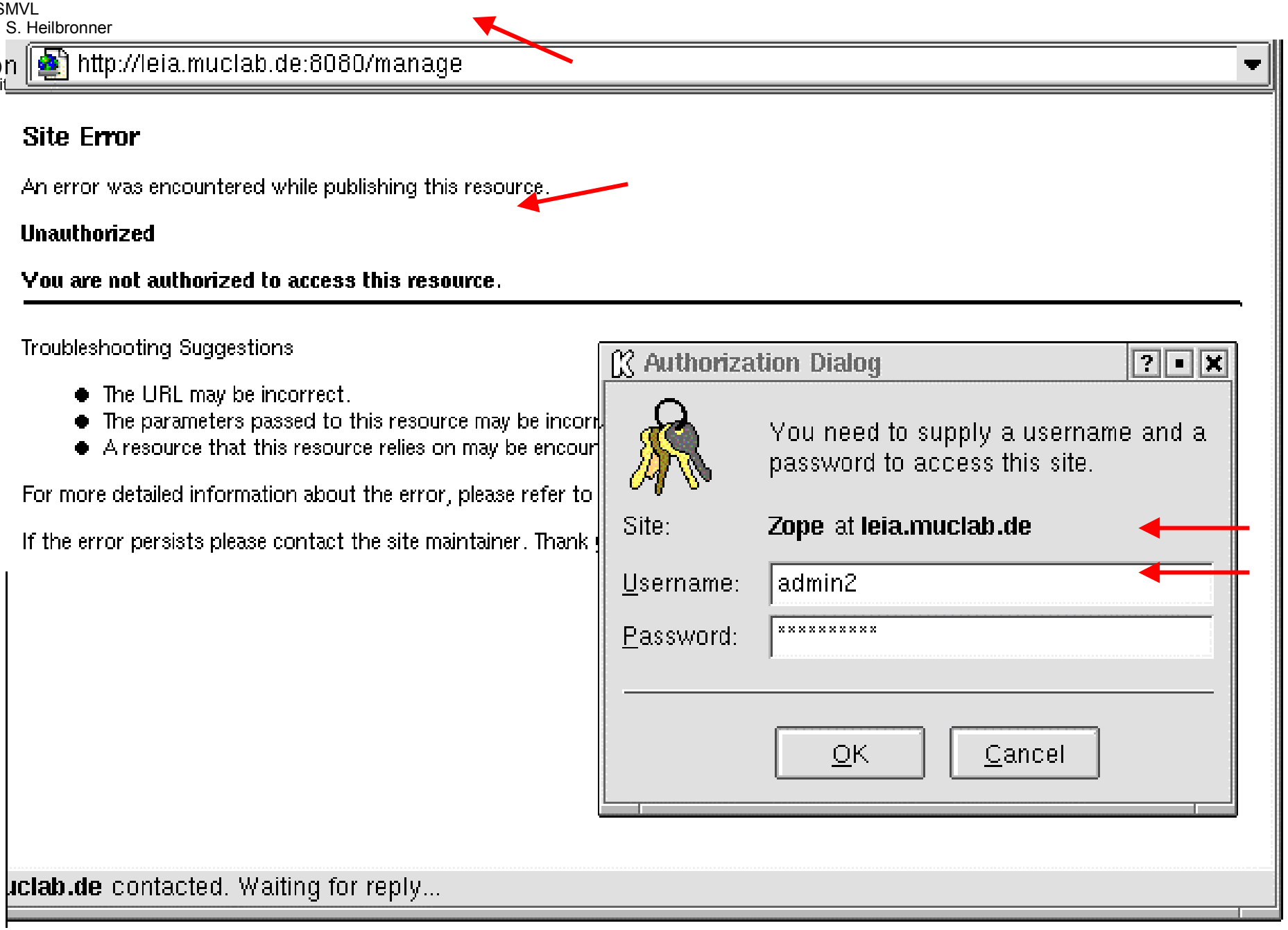
- **A**uthentisierung
  - Feststellung der Identität
  - Implementierung:
    - Abfrage Benutzername/Passwort
    - Keycard
- **A**utorisierung
  - Festlegung der Nutzungsrechte
  - hier: Welche weiteren Zugriffe sind erlaubt ?
- **A**ccounting
  - Aufzeichnung verrechnungsrelevanter Nutzungsdaten
  - Aggregation der Daten
  - Ziel: Verrechnung der Nutzen

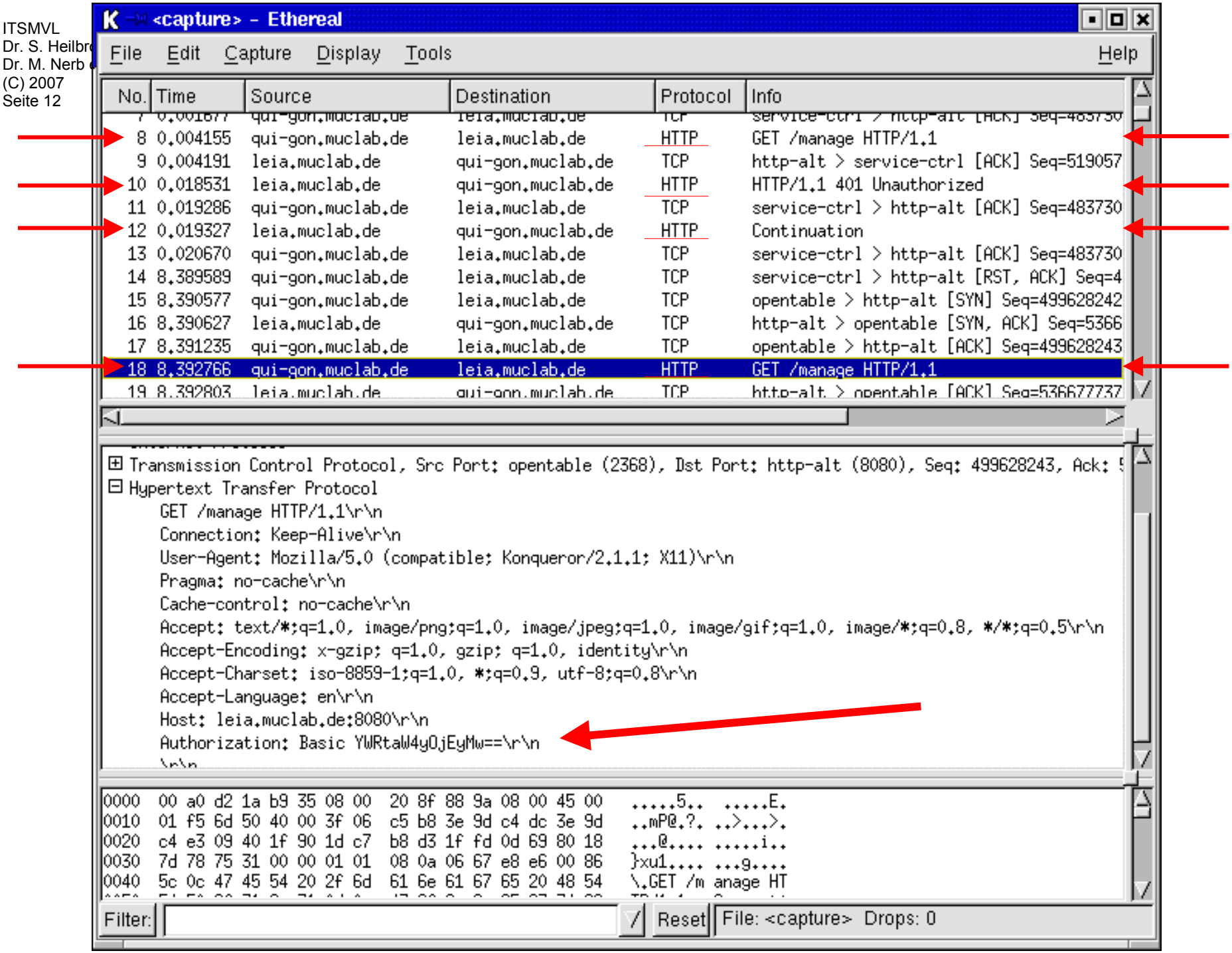
AAA: Übertragung von Autorisierungs-Information

- Cookies
  - Kleine „Stücke von Information“
  - Inhalte vom Server festgelegt
  - Browser stellt sie bestimmten Servern zur Verfügung
  - lange Lebensdauer
- HTTP-Basic/Digest
  - Authentisierunginformation im HTTP-Header
- Oder: Im URL kodiert
  - „Session-ID‘s“



Wie sieht so etwas aus ? ...


http://leia.muclab.de:8080/manage
Site Error
An error was encountered while publishing this resource.
Unauthorized
You are not authorized to access this resource.
Troubleshooting Suggestions
The URL may be incorrect.
The parameters passed to this resource may be incorr
A resource that this resource relies on may be encour
For more detailed information about the error, please refer to
If the error persists please contact the site maintainer. Thank
Authorization Dialog
You need to supply a username and a password to access this site.
Site:
Zope at leia.muclab.de
Username:
admin2
Password:
OK
Cancel
uclab.de contacted. Waiting for reply...


<capture> - Ethereal
File Edit Capture Display Tools Help
No. Time Source Destination Protocol Info
8 0.004155 qui-gon.muclab.de leia.muclab.de HTTP GET /manage HTTP/1.1
9 0.004191 leia.muclab.de qui-gon.muclab.de TCP http-alt > service-ctrl [ACK] Seq=519057
10 0.018531 leia.muclab.de qui-gon.muclab.de HTTP HTTP/1.1 401 Unauthorized
11 0.019286 qui-gon.muclab.de leia.muclab.de TCP service-ctrl > http-alt [ACK] Seq=483730
12 0.019327 leia.muclab.de qui-gon.muclab.de HTTP Continuation
13 0.020670 qui-gon.muclab.de leia.muclab.de TCP service-ctrl > http-alt [ACK] Seq=483730
14 8.389589 qui-gon.muclab.de leia.muclab.de TCP service-ctrl > http-alt [RST, ACK] Seq=4
15 8.390577 qui-gon.muclab.de leia.muclab.de TCP opentable > http-alt [SYN] Seq=499628242
16 8.390627 leia.muclab.de qui-gon.muclab.de TCP http-alt > opentable [SYN, ACK] Seq=5366
17 8.391235 qui-gon.muclab.de leia.muclab.de TCP opentable > http-alt [ACK] Seq=499628243
18 8.392766 qui-gon.muclab.de leia.muclab.de HTTP GET /manage HTTP/1.1
Transmission Control Protocol, Src Port: opentable (2368), Dst Port: http-alt (8080), Seq: 499628243, Ack:
Hypertext Transfer Protocol
GET /manage HTTP/1.1\r\n
Connection: Keep-Alive\r\n
User-Agent: Mozilla/5.0 (compatible; Konqueror/2.1.1; X11)\r\n
Pragma: no-cache\r\n
Cache-control: no-cache\r\n
Accept: text/*;q=1.0, image/png;q=1.0, image/jpeg;q=1.0, image/gif;q=1.0, image/*;q=0.8, */*;q=0.5\r\n
Accept-Encoding: x-gzip; q=1.0, gzip; q=1.0, identity\r\n
Accept-Charset: iso-8859-1;q=1.0, *;q=0.9, utf-8;q=0.8\r\n
Accept-Language: en\r\n
Host: leia.muclab.de:8080\r\n
Authorization: Basic YWRtaW4yOjEyMw==\r\n
\r\n
0000 00 a0 d2 1a b9 35 08 00 20 8f 88 9a 08 00 45 00 .....5.. .....E.
0010 01 f5 6d 50 40 00 3f 06 c5 b8 3e 9d c4 dc 3e 9d ..mP@.?. ..>...>.
0020 c4 e3 09 40 1f 90 1d c7 b8 d3 1f fd 0d 69 80 18 ...@.... .....i..
0030 7d 78 75 31 00 00 01 01 08 0a 06 67 e8 e6 00 86 }xu1.... ...g....
0040 5c 0c 47 45 54 20 2f 6d 61 6e 61 67 65 20 48 54 \.GET /m anage HT
Filter: Reset File: <capture> Drops: 0

Web-Proxies

Server Backends: Anforderungen

- Abfrage von Informationen für Authentisierung und Autorisierung
  - Benutzer
  - Jeweilige Rechte
  - Nutzungszeiten
  - Sonstige Vorbelegungen (GUI)
- Aufzeichnung der Nutzungsdaten
  - Accounting
  - Leistungsmanagement
- Prüfung von Inhalten
  - Angefragte URLs
  - Empfangene Daten

Web-Proxies
Server-Backends: Implementierungen

- Datenbank
  - Vorzuhaltende Information
    - Autorisierung: Name/Passwort
    - Autorisierung: Welche Bereiche dürfen erreicht werden?
    - .....
  - Zugriffsprotokolle
    - ODBC für SQL-Datenbank
    - RADIUS (Remote Access and DialIn User Service
    - LDAP (Lightweight Directory Access Protocol)
- Logging
  - Logdatei aus Performanz-Gründen (keine DB!)
- Weitere Dienste
  - Spezielle Protokolle

Typisches Nutzerverhalten 2001 bis 2004: Ziele aus großen IT-Infrastrukturen

2004

Top Websites Yesterday:

destination	request	%	Byte	%	hit-%
*.ebay.de	3032	2.44	99970K	13.43	0.66
*.uni-kl.de	143	0.11	91385968	11.99	0.00
*.	133	0.11	27819751	3.65	97.74
*.ebaystatic.com	25216	20.26	24799285	3.25	66.86
*.hp.com	66	0.05	22314733	2.93	34.85
*.ebayimg.com	2382	1.91	21111267	2.77	25.65
*.berkeley.edu	114	0.09	20421144	2.68	0.00
*.smc.com	349	0.28	17355549	2.28	59.03
*.comdirect.de	4300	3.45	17172862	2.25	1.00
*.t-online.de	3260	2.62	13278385	1.74	32.82
<error>	6481	5.21	12263399	1.61	13.56
*.gmx.net	1899	1.53	11962863	1.57	66.72
*.praline.de	1325	1.06	10141176	1.33	50.79
*.ebay.com	3103	2.49	8927071	1.17	58.07
*.mobile.de	978	0.79	8708235	1.14	35.99
*.web.de	1539	1.24	8012137	1.05	15.85
*.	1355	1.09	6998993	0.92	79.63
*.sportbilder.de	297	0.24	6690092	0.88	43.77

2001

Destination	Request	%	Bytes	%	hit-%
<error>	33683	12.84	39721881	3.41	5.38
*.t-online.de	8765	3.34	19083028	1.64	66.77
*.bild.de	8282	3.16	55732135	4.79	46.61
*.doubleclick.net	5061	1.93	7087115	0.61	9.27
*.web.de	4917	1.87	24289001	2.09	35.67
*.akamai.net	4884	1.86	7574273	0.65	87.24
*.xxxxxxxxxxxxxxxxx.de	4098	1.56	14681246	1.26	68.64
*.sueddeutsche.de	3768	1.44	13554644	1.16	38.96
*.consors.de	3133	1.19	6986643	0.60	69.04
*.boerse.de	2831	1.08	11167450	0.96	72.45
*.lycos.de	2796	1.07	22830497	1.96	65.24
*.microsoft.com	1898	0.72	14627786	1.26	54.74
*.br-online.de	1761	0.67	3545059	0.30	73.25
*.ebay.com	1623	0.62	2222164	0.19	91.44
*.yyyyyyyyyyyyyyyy.de	1599	0.61	3464164	0.30	49.47
*.gmx.net	1483	0.57	11860845	1.02	0.20
other: 2691 2nd-level-domains	160337	61.13	854876K	75.20	48.87
Sum	262293	100.00	1136733K	100.00	43.35

Web-Proxies

Ein paar Gedanken zu Optimierungspotentialen....

- Hit/Miss-Rate:
  - Anzahl: ca. 1/3 Treffer
  - Größe: ca. 1/4 Treffer
  - 2/3 aller Anfragen werden verlangsamt
- Nutzung über Tageszeit
  - Mittags NICHT weniger :-)

Top Websites Yesterday:

destination	request	%	Byte	%	hit-%
*.ebay.de	3032	2.44	99970K	13.43	0.66
*.uni-kl.de	143	0.11	91385968	11.99	0.00
*	133	0.11	27819751	3.65	97.74
*.ebaystatic.com	25216	20.26	24799285	3.25	66.86
*.hp.com	66	0.05	22314733	2.93	34.85
*.ebayimg.com	2382	1.91	21111267	2.77	25.65
*.berkeley.edu	114	0.09	20421144	2.68	0.00
*.smc.com	349	0.28	17355549	2.28	59.03
*.comdirect.de	4300	3.45	17172862	2.25	1.00
*.t-online.de	3260	2.62	13278385	1.74	32.82
<error>	6481	5.21	12263399	1.61	13.56
*.gmx.net	1899	1.53	11962863	1.57	66.72
*.praline.de	1325	1.06	10141176	1.33	50.79
*.ebay.com	3103	2.49	8927071	1.17	58.07
*.mobile.de	978	0.79	8708235	1.14	35.99
*.web.de	1539	1.24	8012137	1.05	15.85
*	1355	1.09	6998993	0.92	79.63
*.sportbilder.de	297	0.24	6690092	0.88	43.77

- Server-seitige Optimierung der Übertragung?
  - Header „LAST-MODIFIED“ mitschicken
  - Explizite Informationen zu „EXPIRES“ (z.B. in 10 Minuten)
  - Grafiken/Inhalte browser-spezifisch aufbereiten
  - Inhalte komprimieren (GZ)

Caching Proxies
Statische Auslegung

- Plattenplatzbedarf:
  - Statistische Fragen
    - Wie häufig wird auf welche Seiten zugegriffen?
    - Wie schnell veralten welche Seiten?
  - Nutzen vs. Verwaltungsaufwand berücksichtigen
  - Typische, sinnvolle Größe ??

- Welche Schlüsse zieht man aus der Beobachtung:
  - 80 % der Seiten im Cache veralten innerhalb eines Tages...
  - Bringt ein großer Cache wirklich so viel ..... (nein, aber ....)

Caching Proxies
Dynamische Auslegung

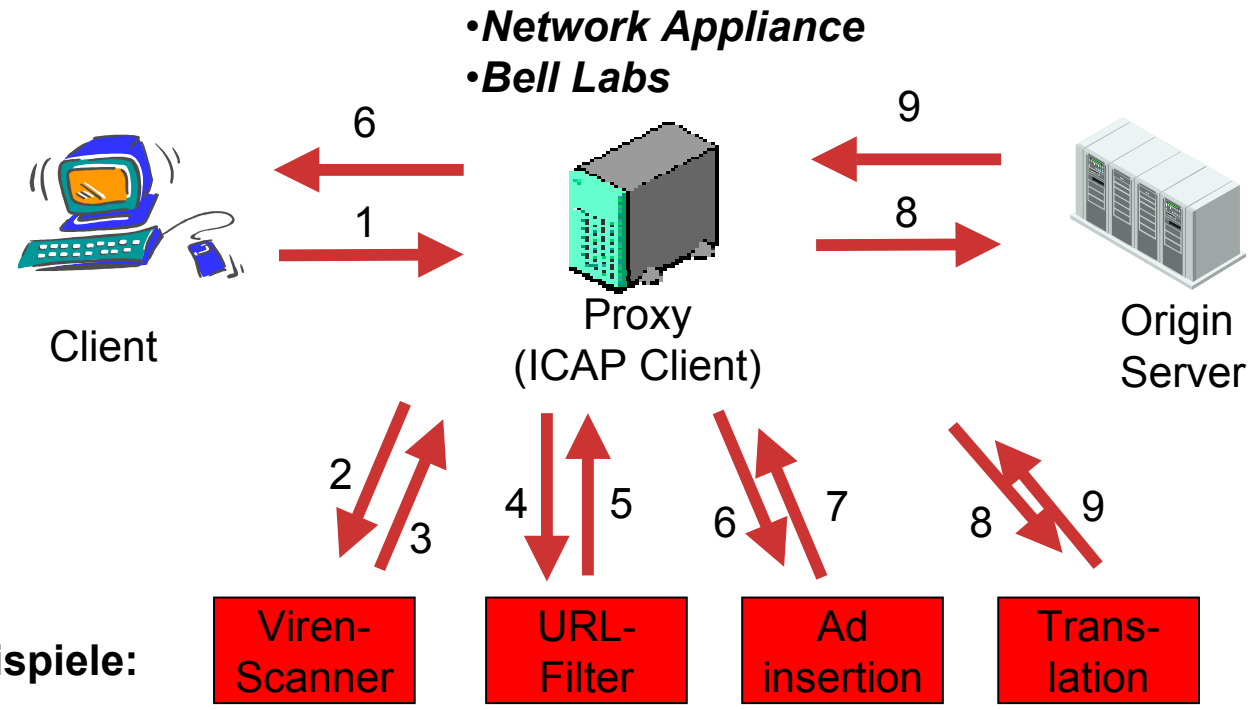
- Anzahl Prozessoren
  - Anzahl parallel laufender Zugriffe (HTTP 1.1 vs 1.0)
  - Wieviel kann ein Prozessor davon abwickeln ?
    - Wie ist das Verhalten bei Überlast ?
    - Toleranz der Benutzer ?
  - Entscheidend sind Zusatzdienste:
    - Virenscanning für HTTP/FTP => hohe Prozessorlast
    - Multimedia-Streaming
- Weitere limitierende Faktoren
  - Zugangsbandbreiten eingehend
  - Zugangsbandbreiten abgehend
  - Zugriffscharakteristik für Hintergrundspeicher

Virenscanning
Implementierung

- Scanner im HTTP-Strom
  - unterbricht Zugriff bei
    - Erkennung Virenmuster
    - Zugriff auf bestimmte Seiten
  - Problem:
    - Gesamter Strom muß gefiltert werden (auch HTML)
- besser wäre:
  Proxy „präsentiert“ dem Scanner nur Wichtiges....

Exkurs
Internet Content Adaptation Protocol

- ICAP-Server nur für „bestimmten“ Content registriert
  - somit: Kein Durchschleusen des gesamten HTTP-Stroms



Weitere Literatur:
http://www.i-cap.org/docs/icap_whitepaper_v1-01.pdf
http://de.wikipedia.org/wiki/ICAP