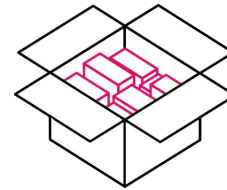


HTTP – Die Kerntechnologie des Webs

Hoai Viet Nguyen – TH Köln

Technology
Arts Sciences
TH Köln



Agenda

- **Protokolle**
- **HTTP**
- **HTTP-Verbindungspersistenz**
- **Cookies**
- **Zusammenfassung und Lernzielkontrolle**

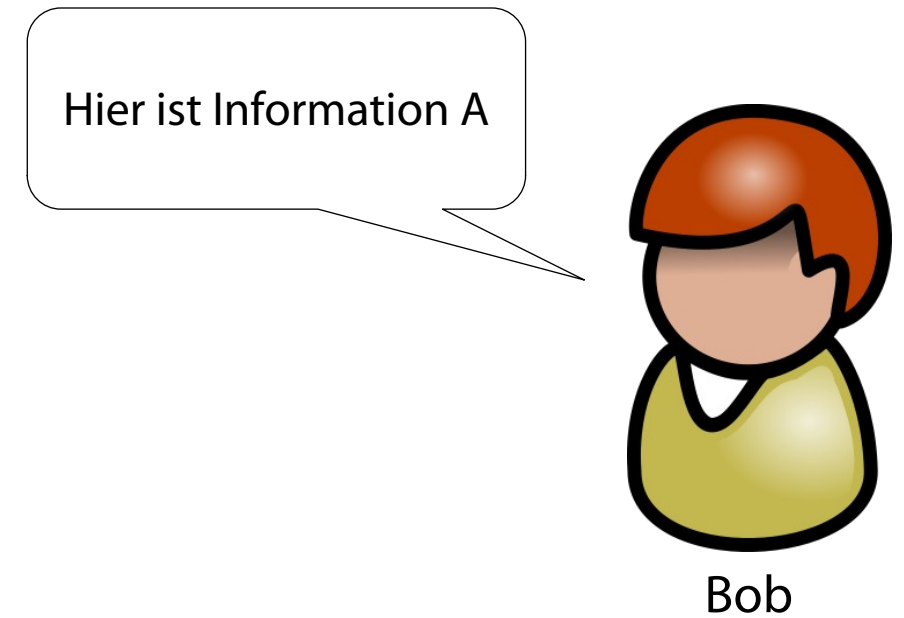
Agenda

- **Protokolle**
 - **HTTP**
 - **HTTP-Verbindungspersistenz**
 - **Cookies**
 - **Zusammenfassung und Lernzielkontrolle**

Warum brauchen wir Protokolle in der Informatik?



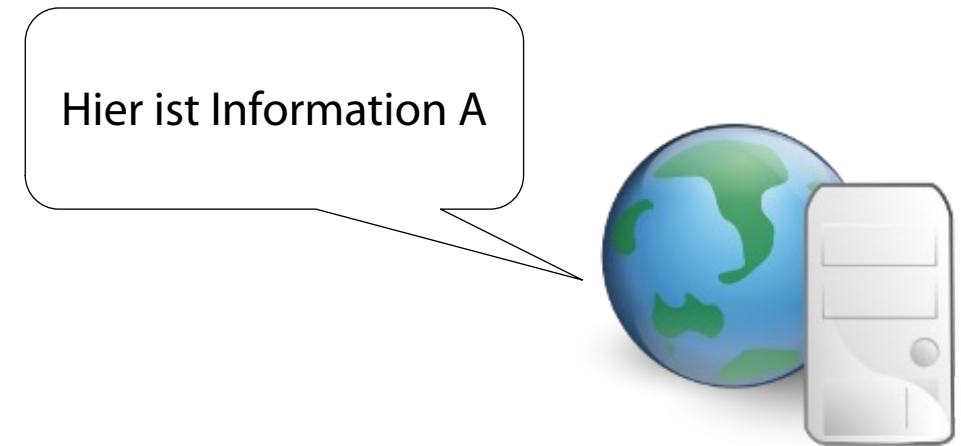
Kommunikation zwischen zwei Personen



Kommunikation zwischen zwei Rechnern

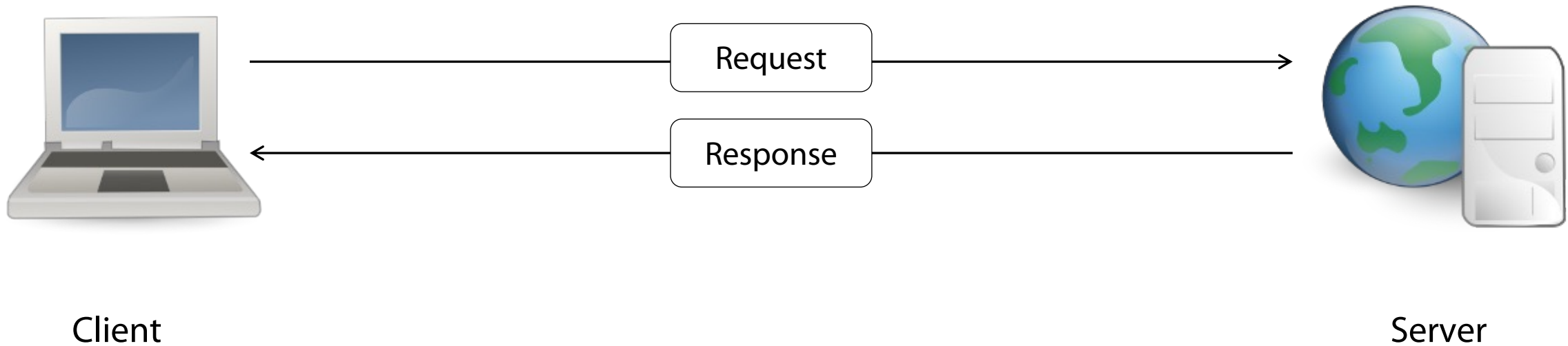


Rechner A



Rechner B

Kommunikation zwischen zwei Rechnern über Protokolle (Client-Server Modell)

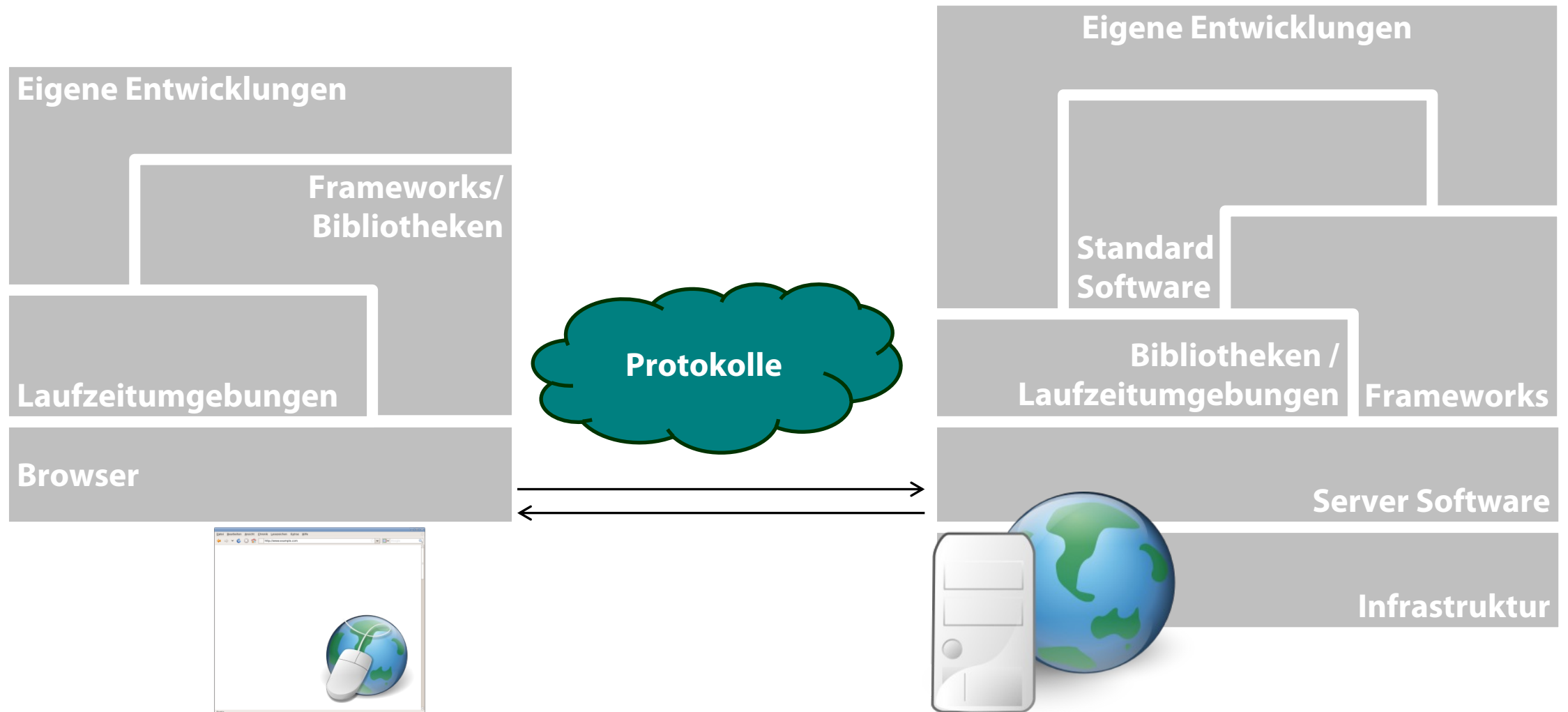


Definition Protokoll laut Duden

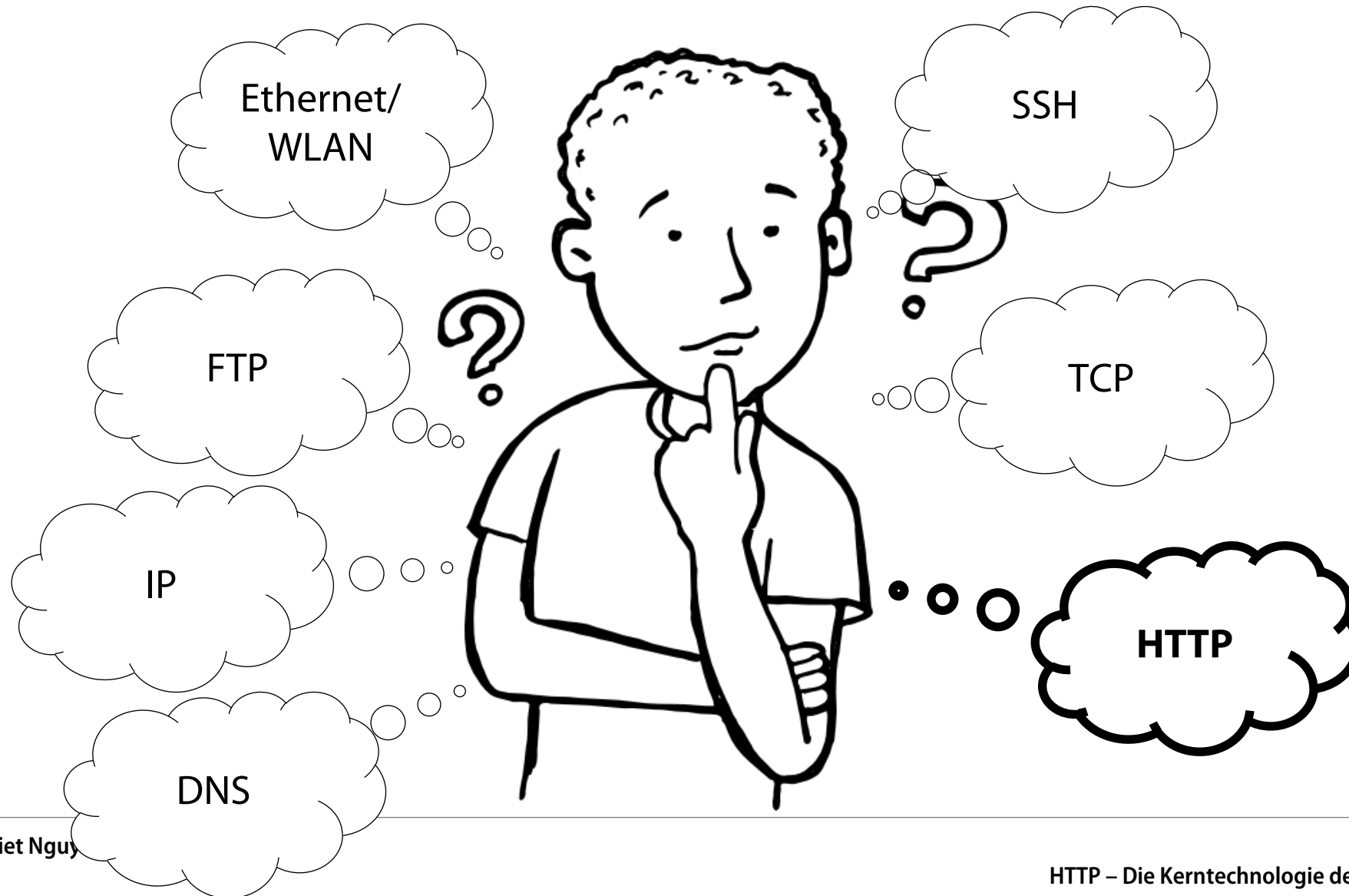
Festlegung von Standards und Konventionen für eine reibungslose Datenübertragung zwischen Computern.

Quelle: <https://www.duden.de/rechtschreibung/Protokoll>

Wo befinden wir uns gerade?



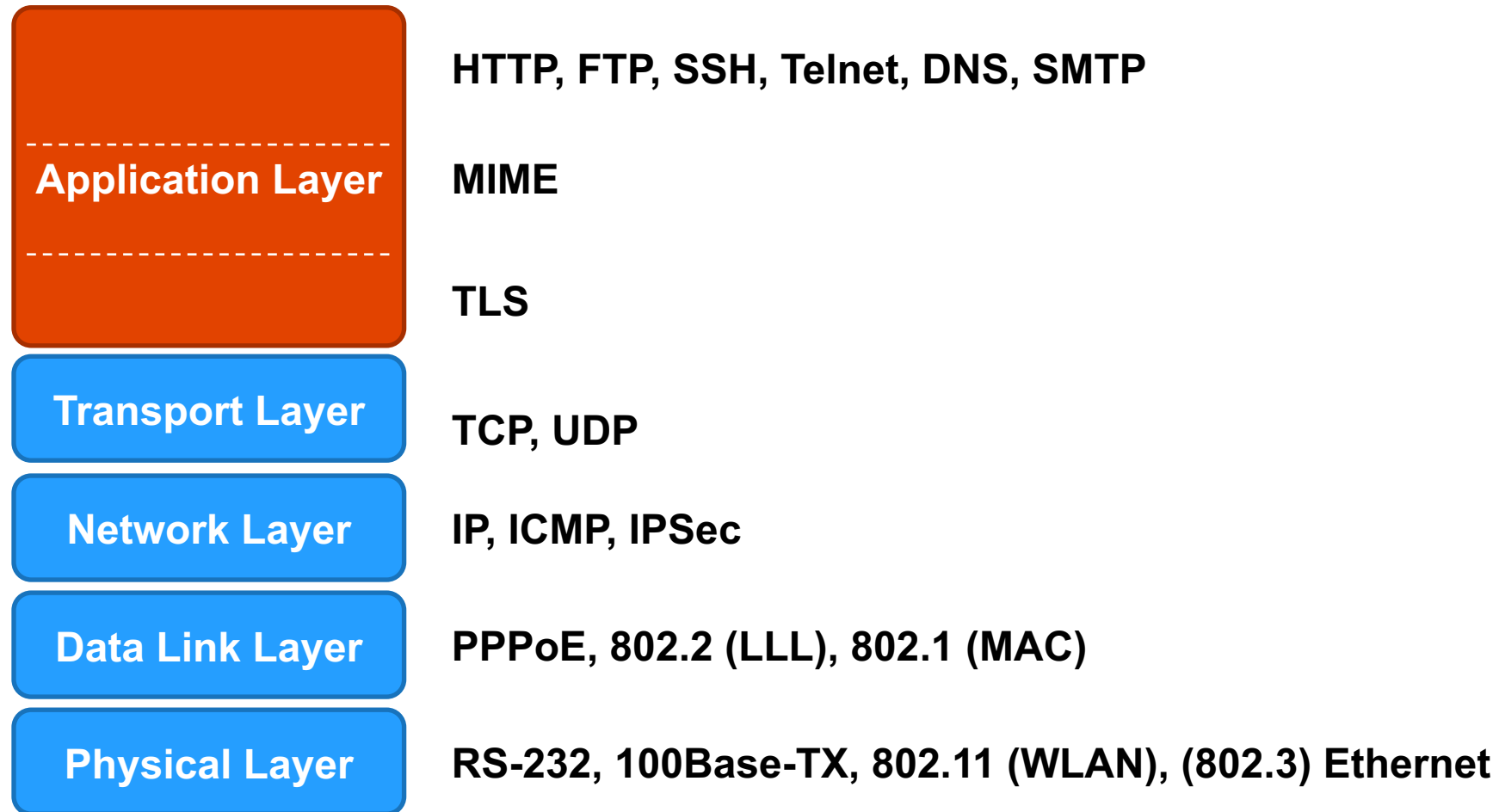
Welche Protokolle haben wir im letzten Video verwendet?



Agenda

- ✓ **Protokolle**
- **HTTP**
 - **HTTP-Verbindungspersistenz**
 - **Cookies**
 - **Zusammenfassung und Lernzielkontrolle**

HTTP im ISO/OSI-Modell



Hypertext Transfer Protocol (HTTP)

- Basiert auf dem Client-Server Modell
- Zustandsloses Protokoll: Request unabhängig voneinander
- Standardport: 80 und 443
- Vier Versionen:
 - HTTP/1.0 (text-basiert über TCP)
 - HTTP/1.1 (text-basiert über TCP)
 - HTTP/2 (binär über TCP)
 - HTTP/3 (binär über UDP)

Protokollformat

■ Request

```
[Method] [URL] [HTTP Version]<CR><LF>
[Header Field Name]: [Value]<CR><LF>
...
[Header Field Name]: [Value]<CR><LF>
<CR><LF>
[Body]
```

■ Response

```
[HTTP Version] [Status Code] [Reason]<CR><LF>
[Header Field Name]: [Value]<CR><LF>
...
[Header Field Name]: [Value]<CR><LF>
<CR><LF>
[Body]
```

Metadaten und Nutzdaten in HTTP-Nachrichten

■ Request

```
[Method] [URL] [HTTP Version]<CR><LF>
[Header Field Name]: [Value]<CR><LF>
...
[Header Field Name]: [Value]<CR><LF>
<CR><LF>
[Body]
```

Header
(Metadaten)

Body
(Nutzdaten)

■ Response

```
[HTTP Version] [Status Code] [Reason]<CR><LF>
[Header Field Name]: [Value]<CR><LF>
...
[Header Field Name]: [Value]<CR><LF>
<CR><LF>
[Body]
```

Header
(Metadaten)

Body
(Nutzdaten)

Metadaten: Methoden

- POST (C)reate
- GET (R)ead
- PUT/PATCH (U)pdate
- DELETE (D)elete
- HEAD (S)tatus
- OPTIONS (A)ctions

Metadaten: Header Fields (Auszug)

- **Request**
 - **Accept:** Gewünschter Content-Types
 - **Cookie:** Ein HTTP-Cookie
 - **Host:** Hostname für angefragte Ressource
- **Request & Response**
 - **Content-Type:** MIME-Type des Inhalts im Body
 - **Content-Length:** Länge des Inhalts im Body
- **Response**
 - **Set-Cookie:** Ein HTTP-Cookie setzen
 - **Location:** URL der Weiterleitung

Metadaten: Status Codes (Auszug)

- **1xx Intermediate Status**
 - 100 Continue
 - 101 Switching Protocols
- **2xx Successful Response**
 - 200 OK
 - 201 Created
 - 204 No Content
- **3xx Redirects**
 - 301 Moved Permanently
 - 304 Not Modified
- **4xx Request Errors**
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed
- **5xx Server Errors**
 - 500 Internal Server Error
 - 501 Not Implemented
 - 504 Gateway Time-out

Nutzdaten (Auszug)

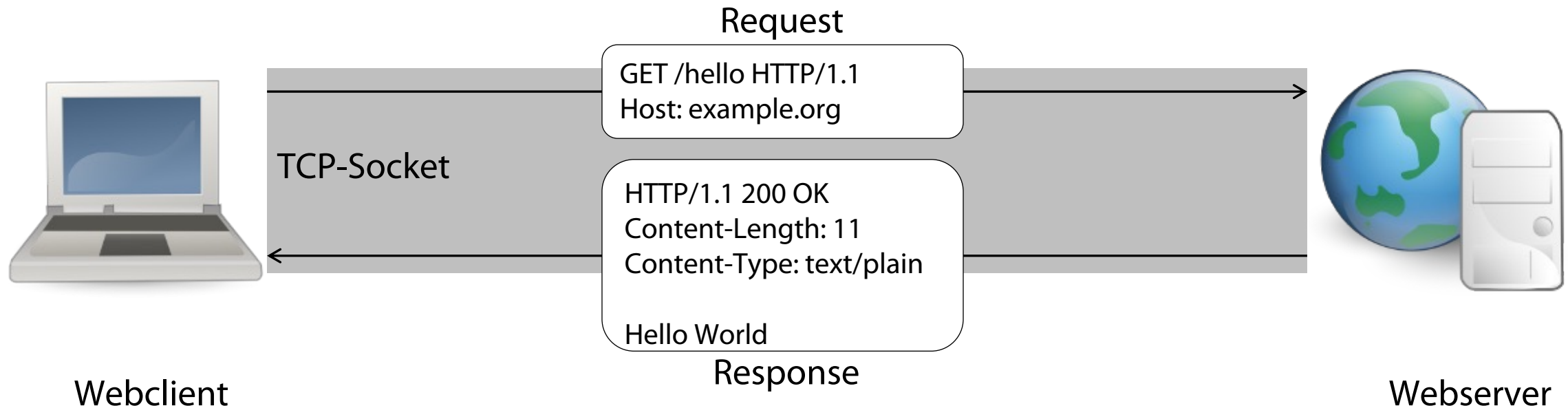
- **Text-basierte Daten:**

- Einfacher Text
- HTML
- XML
- JSON

- **Binärdaten:**

- Bilder
- Video
- Audio
- PDF

Kommunikation zwischen Webclient und Webserver über HTTP



Webclients/HTTP-Clients

- Webclients erstellen HTTP-Requests und verarbeitet die zurückgelieferte Antwort
- Webbrowser sind Webclients mit umfangreichen Funktionalitäten
 - Darstellung HTTP-Responses als UI
 - Berechnung von Animation
 - Zwischenzuspeichern von Daten
 - ...
- „Einfachere“ Webclients für Entwickler sind z.B.:
 - Postman
 - curl
 - wget

Wiederholung und Zusammenfassung: Webserver

- Webserver überträgt Daten als HTTP-Response an anfragende Clients
- Nutzdaten können
 - statische Dateien oder
 - dynamische erzeugte Daten sein

HTTP-Nachrichtenaustausch im Rohformat am Beispiel mit GET

■ Request

```
GET /cities/gummersbach HTTP/1.1<CR><LF>
Host: weather.org<CR><LF>
Accept: text/html<CR><LF>
Accept-Language: de<CR><LF>
<CR><LF>
```

```
GET /index.html HTTP/1.1<CR><LF>
Host: example.org<CR><LF>
Accept: application/json<CR><LF>
<CR><LF>
```

■ Response

```
HTTP/1.1 200 OK<CR><LF>
Content-Length: 33<CR><LF>
Content-Type: text/html<CR><LF>
Cache-Control: max-age=300<CR><LF>
<CR><LF>
<html><body><p>15 Grad</p></html>
```

```
HTTP/1.1 200 OK<CR><LF>
Content-Length: 27<CR><LF>
Content-Type: application/json<CR><LF>
Cache-Control: max-age=3600<CR><LF>
<CR><LF>
{"degree in celsius": "15"}
```

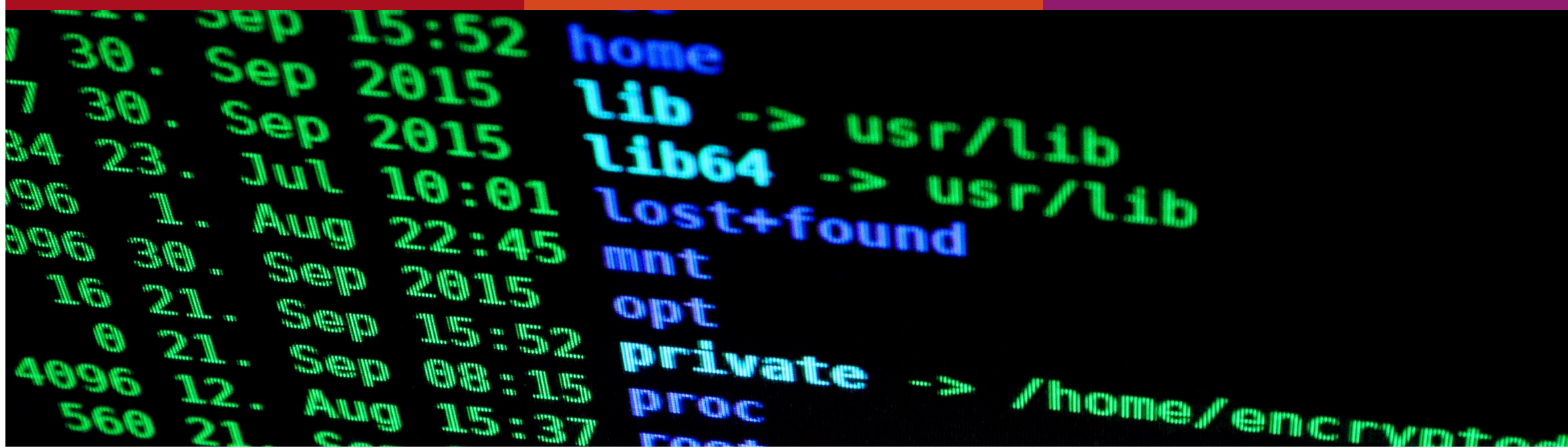
HTTP-Nachrichtenaustausch im Rohformat am Beispiel mit POST

■ Request

```
POST /upload HTTP/1.1<CR><LF>
Host: example.org<CR><LF>
Content-Type: multipart/form-data<CR><LF>
Content-Length: 42<CR><LF>
<CR><LF>
data data data data...
```

■ Response

```
HTTP/1.1 201 Created<CR><LF>
Location: /images/eXwS7g<CR><LF>
<CR><LF>
```



Demo Webbrowser, Postman, curl und netcat

Agenda

- ✓ Protokolle
- ✓ HTTP
- HTTP-Verbindungspersistenz
- Cookies
- Zusammenfassung und Lernzielkontrolle

Abruf von HTML-Dateien



```
<HTML>
...
<script src="script.js">
...
<link rel="stylesheet" href="style.css">
...

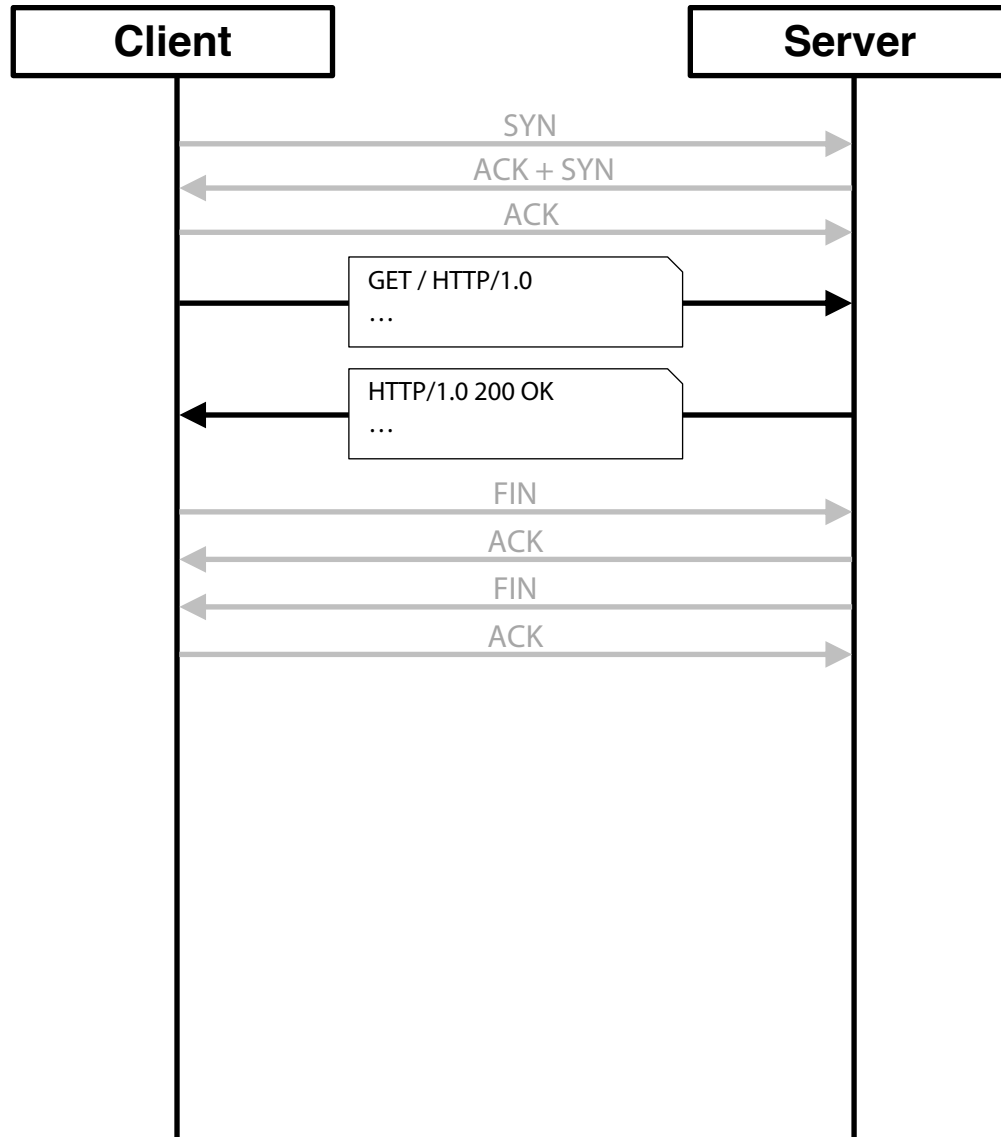
...
</HTML>
```

- Ressourcen, die in einer HTML-Seite häufig vorkommen/verlinkt sind
 - `http://example.com/index.html`
 - `http://example.com/script.js`
 - `http://example.com/style.css`
 - `http://example.com/logo.gif`
- Beispiel: `http://heise.de` benötigt um 100 Ressourcen

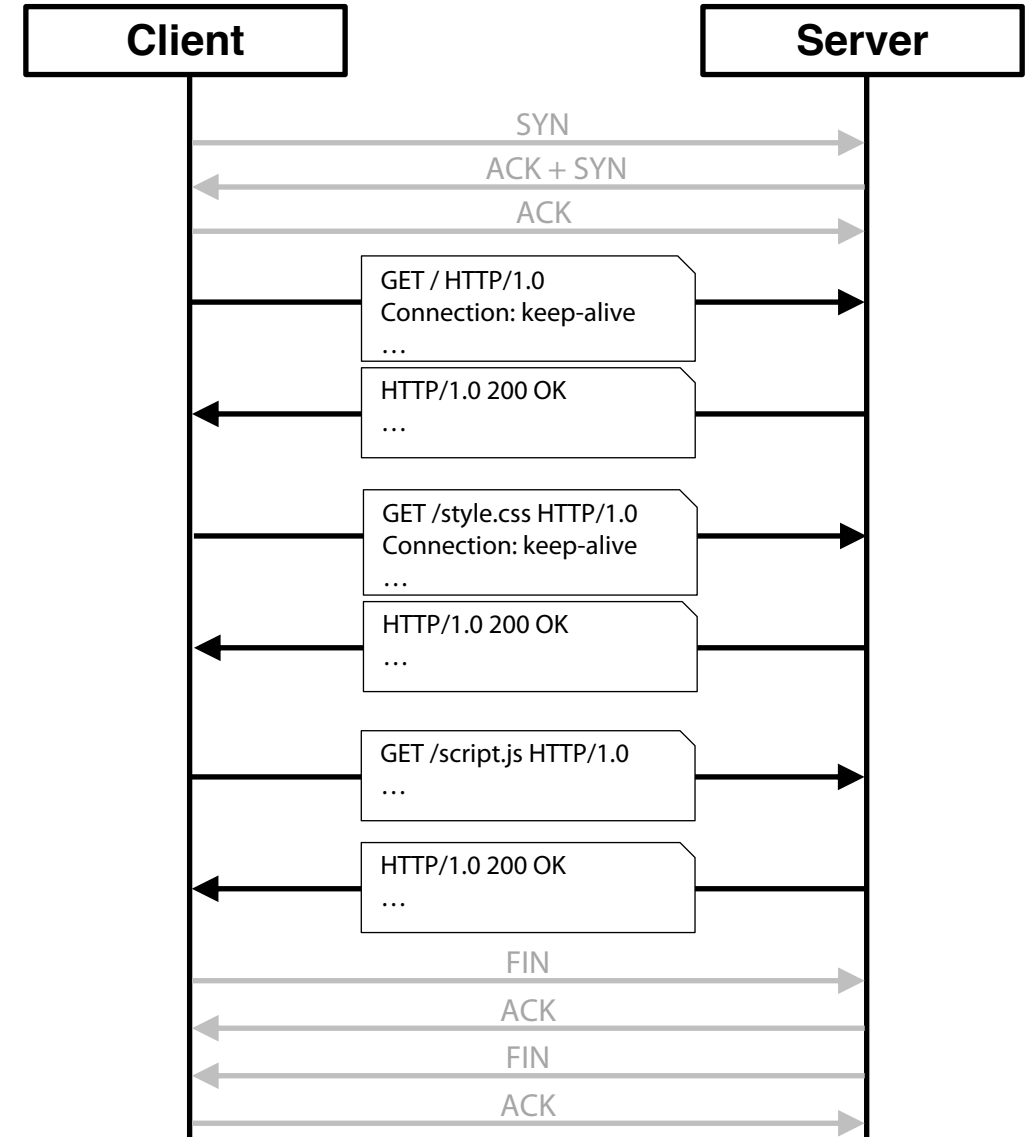
HTTP/1.0

- Autoren: Tim Berners-Lee, Roy Fielding und H. Frystyk
- Text-basiertes Application Layer Protokoll
- Standardmäßig Nicht-Persistente TCP-Verbindung
- Persistenz durch einen entsprechenden Header:
 - Connection: keep-alive

HTTP/1.0 Verbindungspersistenz



Hoai Viet Nguyen



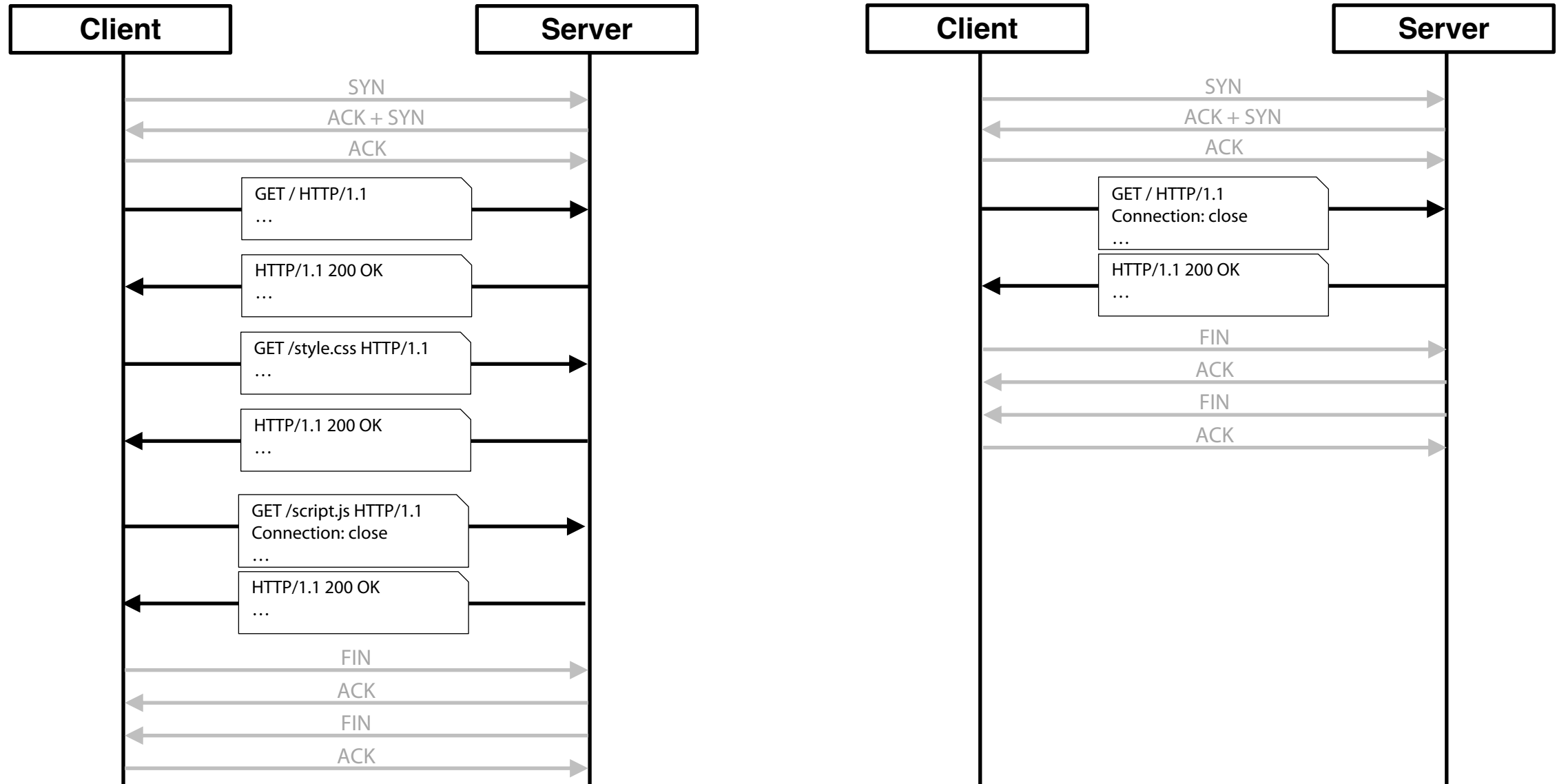
HTTP – Die Kerntechnologie des Webs

Technology
Arts Sciences
TH Köln

HTTP/1.1

- RFC 9112
- Text-basiert
- Standardmäßig persistente TCP-Verbindung
- Nicht-Persistenz kann durch entsprechenden Header eingestellt werden: `Connection: close`

HTTP/1.1 Verbindungspersistenz



Agenda

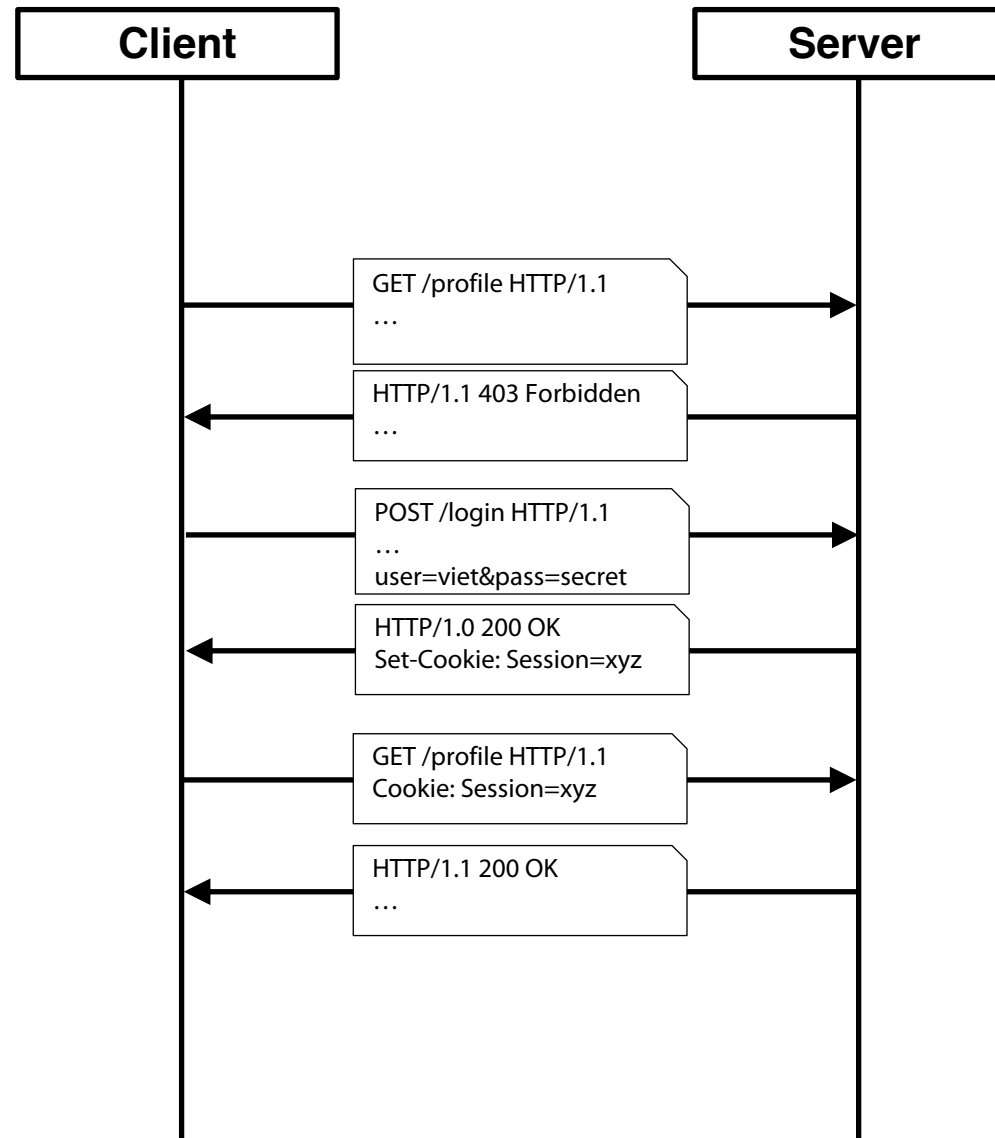
- ✓ **Protokolle**
- ✓ **HTTP**
- ✓ **HTTP-Verbindungspersistenz**
- **Cookies**
- **Zusammenfassung und Lernzielkontrolle**

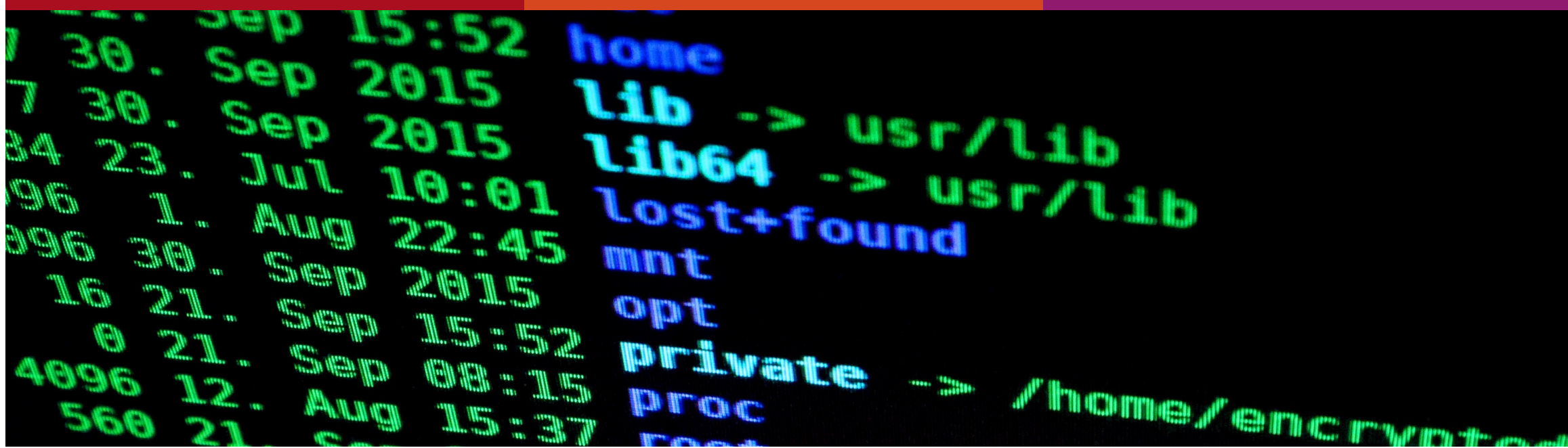
Cookies



- HTTP ist zustandslos
- Viele Szenarien benötigen Sessions (Web Shop, Online Banking, ...)
- Zu diesem Zweck wurden die sog. Cookies eingeführt (Netscape, 1994)
- Ein Cookie beinhaltet eine Menge von Name/Value-Paaren, die beliebige Werte annehmen können
- Typischer Einsatz: Zufallswerte, die eine Session eindeutig identifizieren

Cookies in HTTP





Demo Cookie

Agenda

- ✓ **Protokolle**
- ✓ **HTTP**
- ✓ **HTTP-Verbindungspersistenz**
- ✓ **Cookies**
- **Zusammenfassung und Lernzielkontrolle**

Zusammenfassung

- **Protokolle werden benötigt damit Systeme nach einer festgelegten Spezifikation miteinander kommunizieren können**
- **HTTP ist ein standardisiertes Protokoll und bildet die Kerntechnologie des Webs**
- **Webclients können HTTP-Request erstellen und HTTP-Response verarbeiten**
- **Da HTTP zustandslos ist, werden Cookies benötigt, um Zustände z.B. Sitzungen zu repräsentieren**

Diese Fragen sollten Sie nach der heutigen Sitzung beantworten können

- **Was ist HTTP?**
- **Wo ist HTTP im ISO/OSI-Modell?**
- **Wie ist eine HTTP-Nachricht aufgebaut?**
- **Welche HTTP-Methoden gibt es?**
- **Wie ist die Verbindungspersistenz bei HTTP/1.0 und HTTP/1.1?**
- **Was sind Statuscodes und was bedeuten sie?**
- **Was sind Cookies und warum werden sie benötigt?**

Danksagung

Die verwendeten Icons stehen unter der Flaticon Basic Lizenz (flaticon.com) und stammen von

- Freepik
- Atif
- juicy fish
- BZZRINCANTATION