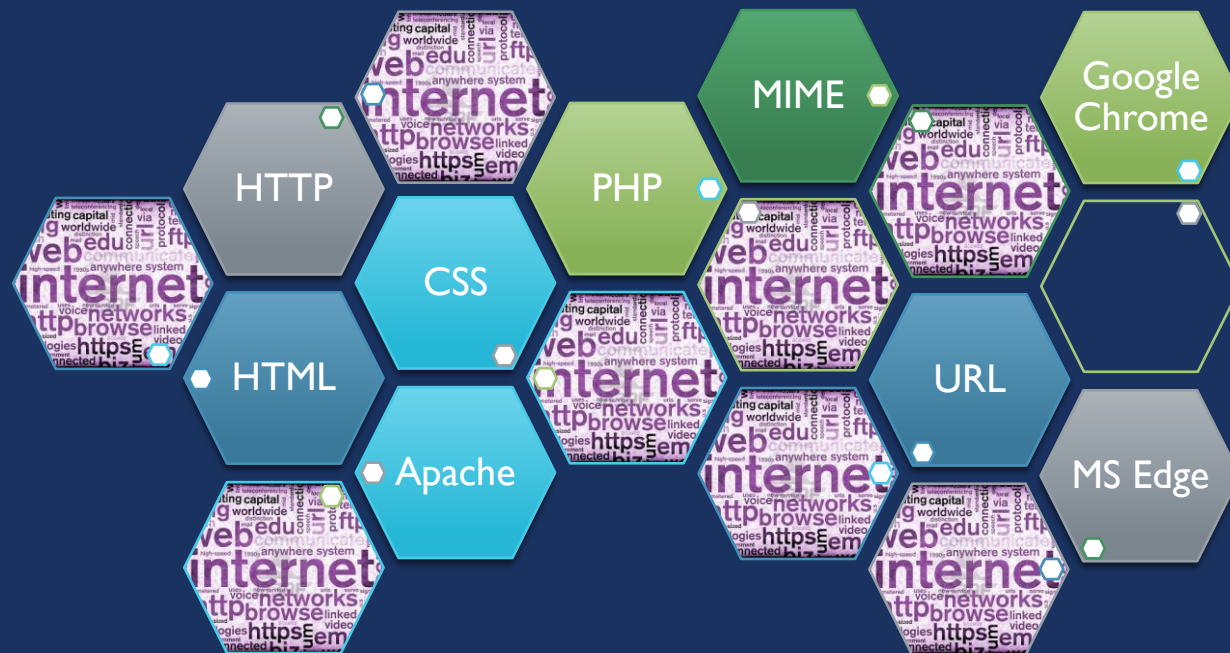


WEBARCHITEKTUR UND GESCHICHTE

PROF. ANDREAS HARTMANN –

WEBTECHNOLOGIEN



INHALT

- Historie
- Web-Browser
- Markup language (HTML), Cascading style sheets (CSS)
- Scripting
- Uniform Resource Identifier (URI)
- Protokolle (HTTP)
- Internet Architektur
- MIME-Typen
- Architektur im World Wide Web

Python Remains Most Popular Programming Language

Popularity of each programming language based on share of tutorial searches in Google



Yearly trend compares percent change from Feb 2019 to Feb 2020
Sources: GitHub, Google Trends

QUELLEN, LITERATUR

- Quellenangaben

Der Inhalt richtet sich – soweit nicht anders gekennzeichnet – aus an:

- Webarchitektur: Erik Wilde, UC Berkeley, (<http://dret.net/lectures/web-fall10/>)
- Es handelt sich um eigene Abbildungen, wenn keine Quelle angegeben ist.
- Einige Inhalte sind an Wikipedia angelehnt. Als Quellangaben wurden die dort angegebenen Referenzen geprüft.

2020 *This Is What Happens In An Internet Minute*



WEBARCHITEKTUR

HISTORIE UND ÜBERBLICK

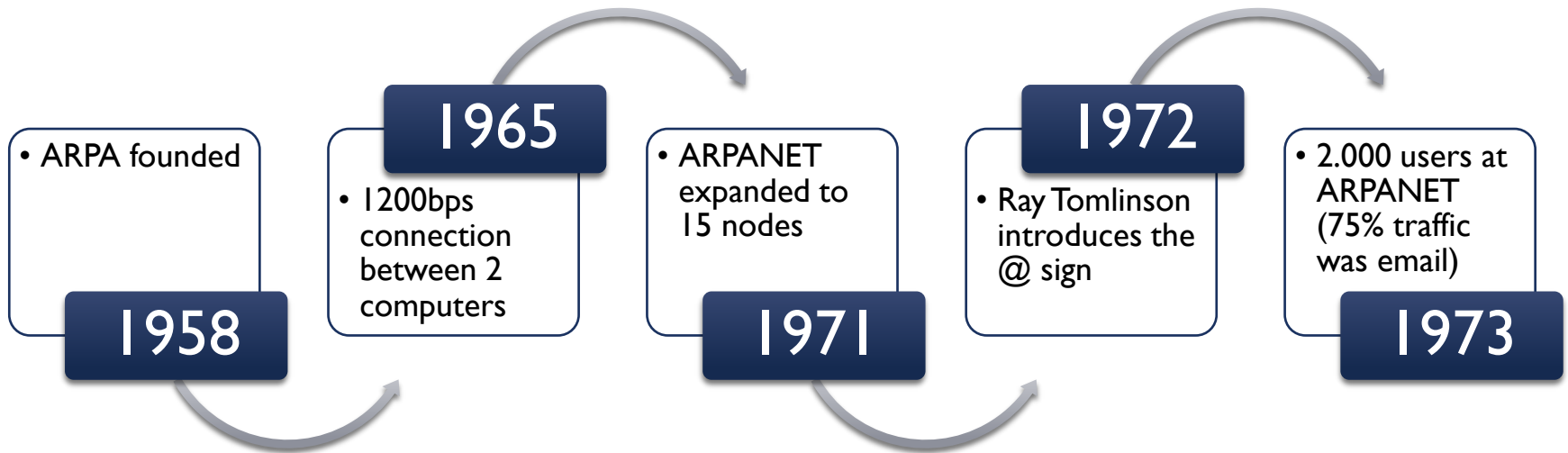
BEGRIFFE

Webarchitektur wird durch eine Reihe historisch gewachsener Standards beschrieben. Sie müssen die zugehörigen Begriffe kennen und erläutern können.

HISTORIE

Wir beginnen mit einem geschichtlichen Rückblick. So wird der Zusammenhang der Begriffe deutlich und es wird klar, weshalb bestimmte Standards das Web definieren.

GESCHICHTSSTUNDE TEIL I

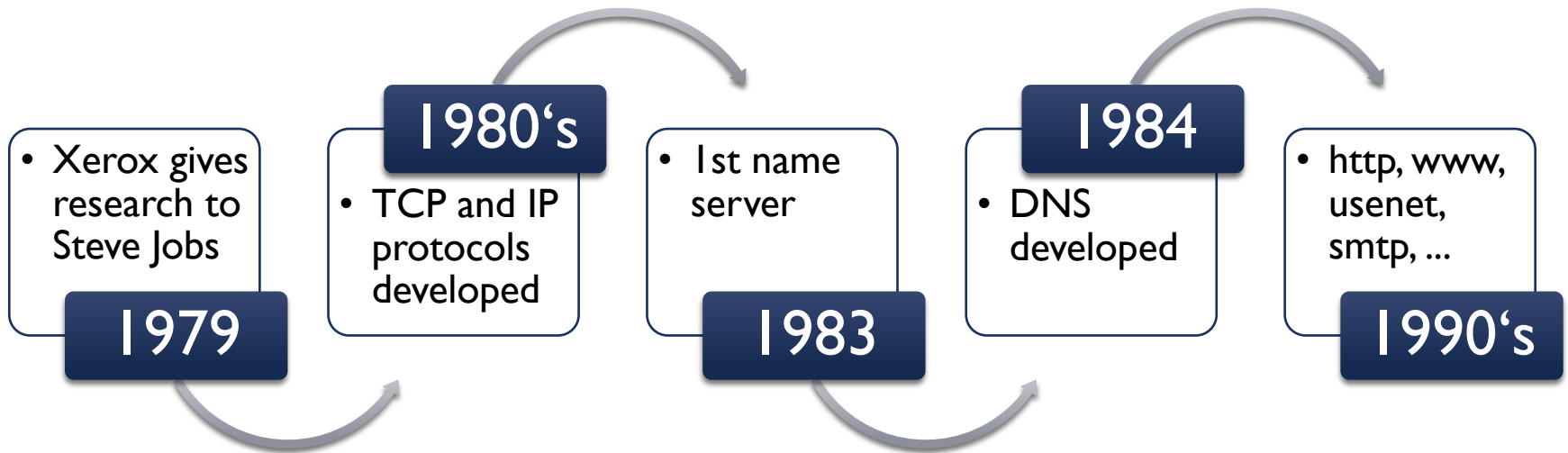


Quelle: <https://www.linkedin.com/pulse/world-wide-web-internet-same-thing-right-rob-burgner>

MERKEN

Zu Beginn ging es darum, Computer über lokale Netzwerke hinweg zu verbinden und eine Kommunikation zu ermöglichen. Textnachrichten standen dabei primär im Vordergrund. Als Standardisierung wurde hierbei u.a. der Emailstandard und die Emailadresse definiert.

GESCHICHTSSTUNDE TEIL2



Quelle: <https://www.linkedin.com/pulse/world-wide-web-internet-same-thing-right-rob-burgner>

MERKEN

Weiterhin wurden die Kommunikationsprotokolle standardisiert: TCP/IP. Ein anderes Ergebnis war die eindeutige Internetadresse von Servern. Anfang der 90er Jahre ging es dann rasant voran. Beachten Sie, dass Microsoft und Apple die Verbreitung von Home-PCs massiv vorangebracht hatten. PCs waren nicht mehr ausschließlich Firmen und Universitäten vorbehalten. Mit einem Schlag waren Millionen von Menschen am „Internet“ interessiert.

Dabei ging es von Anfang an um die Information.

NEWS FROM 1981

https://www.youtube.com/watch?v=X84muuaySVQ&ab_channel=maugantiasx





DER WEB-BROWSER

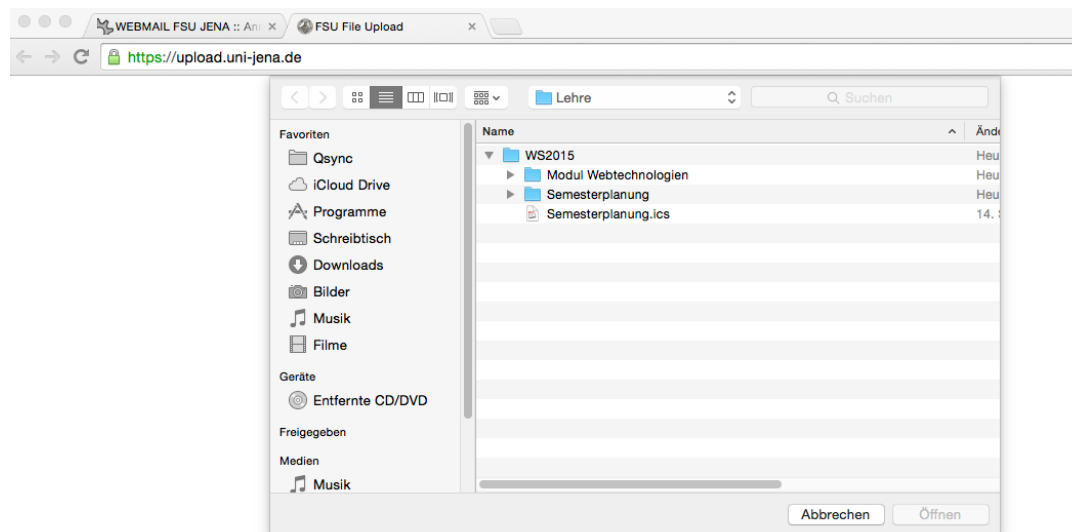
WEBARCHITEKTUR UND STANDARDS (TEIL I)

WEB-BROWSER

- Kommunikation, z.B. zwischen Browser und (Web-)Server oder zwischen Browser und Betriebssystem
- Anzeige eines *korrekt* dargestellten HTML-Dokuments auf der Anzeige des Anwenders (Rendering)
- Ausführung von Scripten (JIT oder Interpreter)
- Zusatzfunktionen

Der Browser ist ebenfalls eine Schnittstelle zum lokalen Filesystem!

BROWSER - DATEISYSTEM



können. Diese Kennung hat eine Gültigkeitsdauer von 48 Stunden und muss zur Verwendung durch den Kommunikationspartner von Ihnen an diesen übermittelt werden. Die Kennung kann innerhalb der Geltungsdauer auch mehrfach verwendet werden.

Diese temporären Zugangsdaten werden Ihnen auch per E-Mail zugestellt.

<https://upload.uni-jena.de>.

Bei Problemen wenden Sie sich an postmaster@uni-jena.de oder verwenden die [alte Methode](#)

AUFGABEN DES BROWSERS

- URI korrekt analysieren und Daten vom Server holen
- HTML, CSS, Script analysieren und ein DOM erstellen (Parsing)
- Layout-Vorgaben anwenden und Webseite darstellen (Rendering)
- Script analysieren und bei Bedarf ausführen
- auf Nutzereingaben reagieren

DAS BROWSERPROBLEM

- Browser wurden als native Programme für Betriebssysteme entwickelt und am Standard des Herstellers ausgerichtet.
 - MS Internet Explorer / Edge
 - Mozilla/Firefox
 - Apple Safari
 - Google Chrome
- Die Entwickler stehen zu einander in Konkurrenz und so ist ein funktionierender Standard problematisch.
- Google geht noch einen Schritt weiter – ChromeOS.

DIE KOMMUNIKATION

- Die Kommunikation erfolgt über Standardprotokolle.
- Das Protokoll wird anhand der URI spezifiziert:
 - http/https: Web Protokolle (hypertext transfer protocol)
 - file: Zugriff auf lokale Daten
 - ftp: Zugriff auf entfernte Daten (file transfer protokol)
 - mailto: Browser startet das E-Mailprogramm
 - tel: z.B. auf dem Smartphone die App zum Telefonieren starten
 - sms: z.B. auf dem Smartphone die App für SMS starten
- Mehr erfahren: JavaME – generic connection framework

CLIENT - SICHERHEIT

- Sandbox-Prinzip
- Diese Idee wurde jedoch von der Realität überholt.
- Obwohl Browser lokale Informationen zurückhalten sollen, gibt es zahlreiche Angriffsstellen:
 - Cookies
 - Browser speichert Daten aus Formularen (z.B. Adressvervollständigung)
 - Browser speichert Passwörter
 - Webseiten verwenden Nutzerverhalten (Facebook's like)

HTML & WEBSEITE

Tatsächlich verarbeiten Browser weit mehr als nur HTML.

MEHR ALS HTML

Standard
Medientypen
(Built-In):

Textformat/XML HTML, CSS

Bildformate wie GIF, JPG, PNG, ICO

Weitere
Medientypen und
Programm-
ausführung über
Addons und
Plugins:

Open Document, Office Open XML

PDF

Java Programme (Applets)

Adobe Flash



HTML, CSS, SCRIPT

WEBARCHITEKTUR UND STANDARDS (TEIL 2)

HTML, CSS, SCRIPT

- Das ist Gegenstand der Präsenzveranstaltungen zu:
 - HTML5 und CSS3
 - Javascript
 - PHP und MySQL
 - Responsive Design
 - Apps entwickeln mit HTML5 und CSS3



URI – UNIFORM RESOURCE IDENTIFIER

WEBARCHITEKTUR UND STANDARDS (TEIL 3)

KONZEPT

Uniform-Resource-Identification ist ein grundlegendes Konzept des Internet.

Die logische Architektur des Internet ist um Ressourcen organisiert.

Ziel ist es, über ein einfaches und transparentes Schema Ressourcen zu identifizieren.

Es wird primär zur Kommunikation zwischen Server (Angebot) und Clients (Nachfrage) eingesetzt.

SERVERSEITE

- Wir betrachten Webserver mit Diensten, die über das Internet (einschließlich Intranet) angeboten werden.
- Server erhalten vom Client eine Anfrage unter Angabe der URI.
- Die Anfrage kann zusätzliche Informationen enthalten, z.B. *query information*.
- Der Server verarbeitet die Anfrage (oder reicht sie weiter) und sendet dem Client das Ergebnis.

SCHEMA UND API

- Prominentester Vertreter ist HTTP.
- Was passiert auf ProgrammierEbene?
 - HTTP arbeitet wie eine Schnittstelle.
 - APIs des Clients und des Servers implementieren das Schema.
 - Beispiel ist das Generic Connection Framework von JavaME.
 - Die APIs ist für den Kommunikationspartner transparent und nicht interessant.

AUFBAU

URI = scheme ":" hier-part ["?" query] ["#" fragment]

- scheme: Schema oder Protokoll
 - http
- hier-part: hierarchischer Teil oder Authority + Path
 - //de.wikipedia.org/wiki/Uniform_Resource_Identifier
- query: Abfrageteil oder Query String
 - ?title=Uniform_Resource_Identifier&action=submit
- fragment: ergänzende Angaben zum Query String
 - #something



HTTP

WEBARCHITEKTUR UND STANDARDS (TEIL 4)

HTTP://

„Das Hypertext Transfer Protocol (HTTP, englisch für Hypertext-Übertragungsprotokoll) ist ein zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz...“

Quelle: https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol

OSI-MODELL

- Open Systems Interconnection Model (OSI)
- International Organization for Standardization (ISO)
- Entwicklung begann in den 70/80ern
- Schichtenmodell mit 7 Schichten
- vergleiche auch: Referenzmodell DoD (Department of Defense) für Internetarchitektur

Siehe auch: <https://de.wikipedia.org/wiki/OSI-Modell>

HTTP IM OSI-MODELL

Schicht im OSI-Modell		DoD-Schicht	Protokolle (Bsp.)
7	Anwendung	Anwendung	http , ftp, https, smtp, ldap
6	Darstellung		
5	Sitzung		
4	Transport	Transport	TCP, UDP
3	Vermittlung	Internet	ICMP, IP
2	Sicherung	Netzzugriff	Ethernet
1	physikal. Datenübertragung		

Siehe auch: <https://de.wikipedia.org/wiki/OSI-Modell>

VERBINDUNG(-SAUFBAU)

http benötigt eine Transportschicht (OSI-Modell)

- das ist in den meisten Fällen: TCP

DNS löst den Namen in eine IP-Adresse auf

die Verbindung wird auf einem Port geöffnet (Standard 80)

- `http://www.ix.de:80`

http übermittelt Textnachrichten

NACHRICHTENAUFBAU

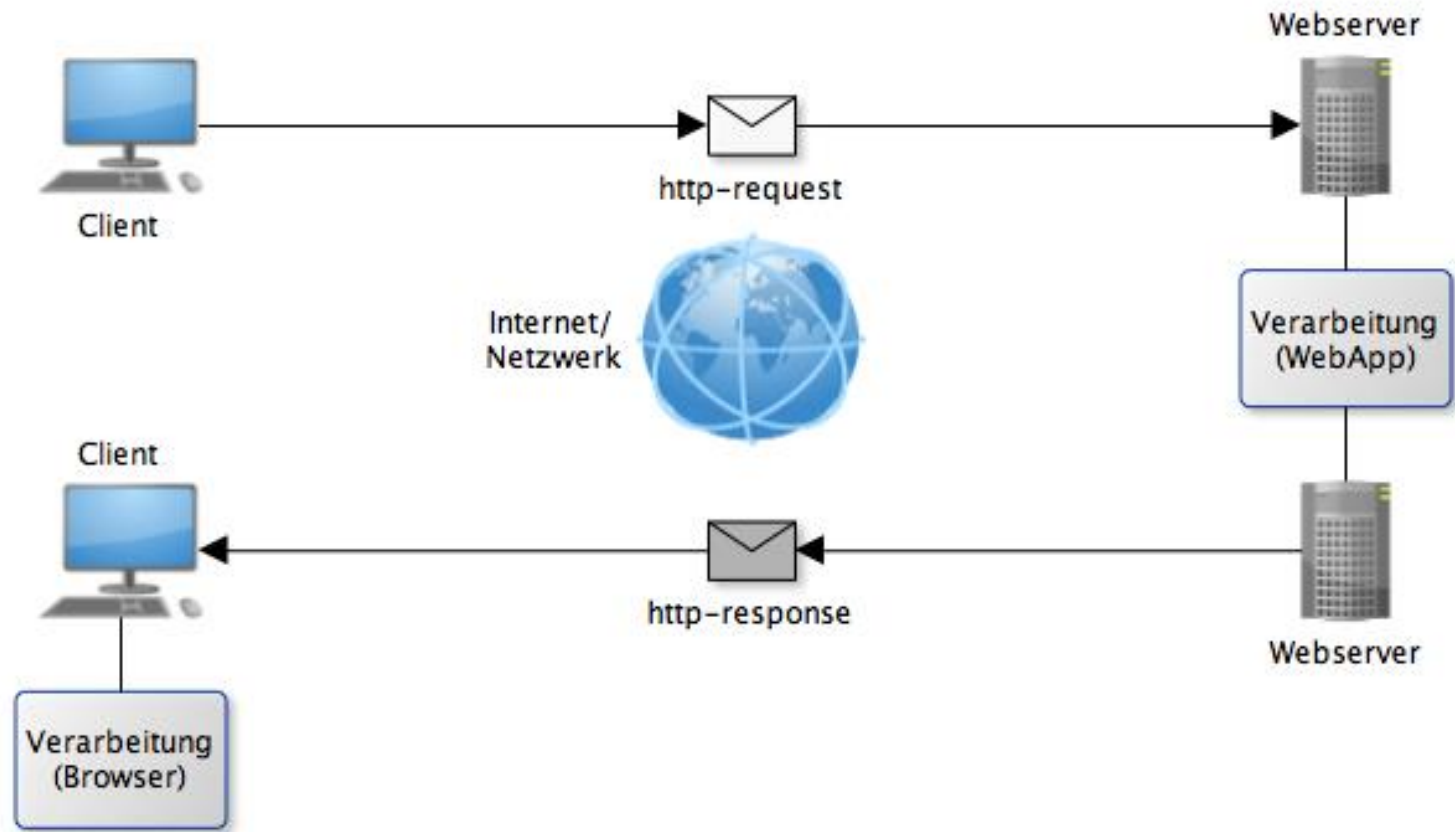
```
Startaufruf (Methode, Version)  
Header *  
Body ?
```

```
GET / HTTP/1.1  
Host: www.ix.de
```

ZUSÄTZLICHE ANGABEN

- können im Header definiert werden, z.B.:
 - Zeichenkodierung
 - unterstützte Komprimierungsmethoden
 - unterstützte Sprachen
- auf diese Art können Client und Server eine bestmögliche Antwort (Response) „verhandeln“

HTTP-CYCLE



REQUEST

Methoden sind GET, POST, HEAD und weitere

es wird immer eine URI benötigt

sie wird im Header angegeben mit HOST

die URI kann zusätzliche Informationen an den Server geben (siehe URI, query information)

der Textkörper ist optional (vergleiche POST)

Der Server...

- ...antwortet immer mit einem Status
- ...kann zusätzliche Informationen senden – im Header
- ...sendet im Body die Antwort an den Client (i.d.R. HTML, CSS, Script, Ressourcen)

STATUSMELDUNGEN

Code	Bedeutung	Beispiel
1xx	Informationen	102 – Processing
2xx	Erfolgreiche Operation	200 – OK
3xx	Umleitung	301 – Moved Permanently (Redirect)
4xx	Client-Fehler	404 – Not Found
5xx	Server-Fehler	503 – Service Unavailable

Spezifikation in RFC 7231 (<https://tools.ietf.org/html/rfc7231>)

HTTP - SICHERHEIT

- HTTP unterstützt Authentifizierung
- Client muss sich bei Server authentifizieren
- Typischer Ablauf:
 - Server meldet: 401 Unauthorized (d.h. Ressource gibt es nur gegen Auth.)
 - Umleitung auf eine Anmeldeseite (z.B. mit Login-Formular)
- Problem: http ist zustandslos
 - Authentifizierung wird nicht „gespeichert“
 - andere Lösungswege notwendig, wie z.B. Sessions/Cookies
 - jeder Browser verhält sich dazu anders → wenig Kontrolle!



INTERNET ARCHITEKTUR

WEBARCHITEKTUR UND STANDARDS (TEIL 5)

BITTE DURCHARBEITEN (DNS, TCP, IP):

[HTTP://DRET.NET/LECTURES/WEB-FALL10/INTERNET](http://dret.net/lectures/web-fall10/internet)



MIME TYPEN

WEBARCHITEKTUR UND STANDARDS (TEIL 6)

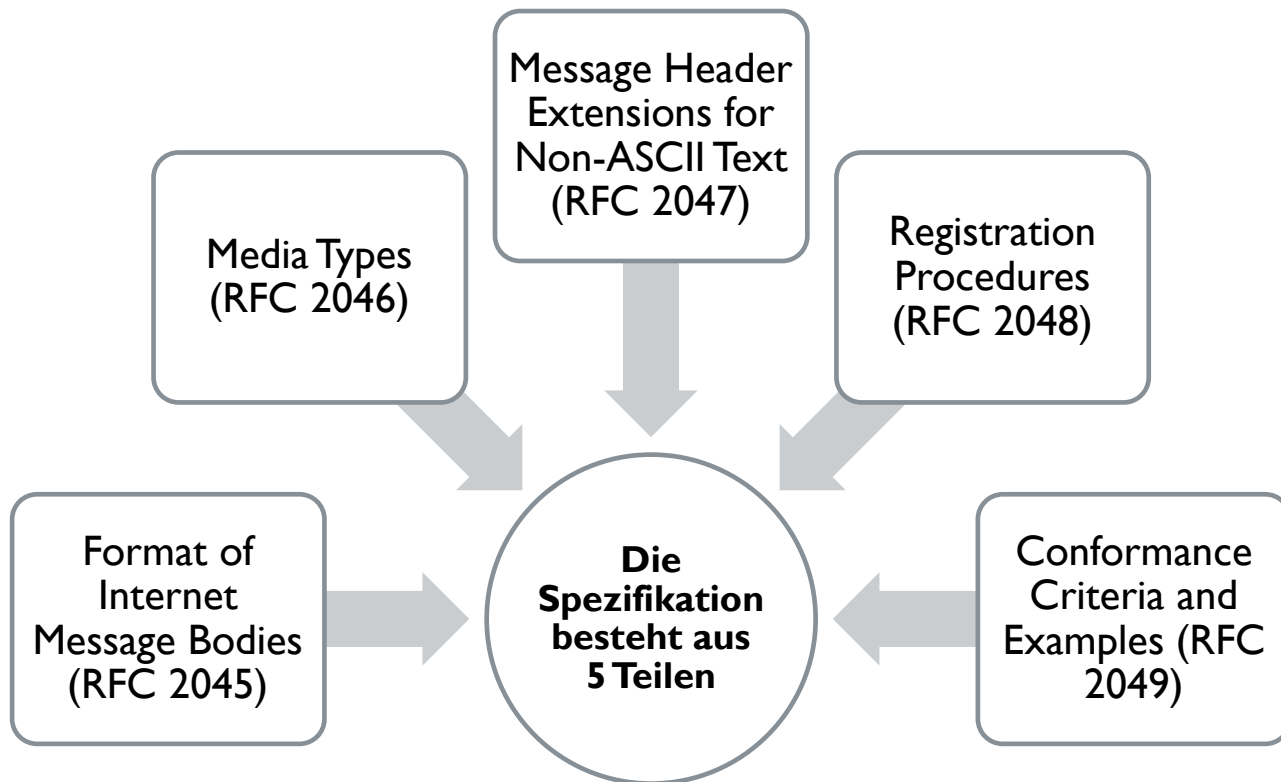
MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME)

„STD 11, RFC 822, defines a message representation protocol specifying considerable detail about US-ASCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

- (1) textual message bodies in character sets other than US-ASCII,
- (2) an extensible set of different formats for non-textual message bodies,
- (3) multi-part message bodies, and
- (4) textual header information in character sets other than US-ASCII.“

Quelle: Spezifikation RFC 2045 (<https://tools.ietf.org/html/rfc2045>)

MIME SPEZIFIKATION



HINTERGRUND



Erweiterungen des RFC 822 (E-Mail Standard)



Datenformat bis dahin nur ASCII (Text basiert)



Unterstützung für Sonderzeichen und Multimedia



Anwendung nicht nur bei E-Mail (z.B. HTTP)

SEMANTIK

Es geht erneut um die Bedeutung der übermittelten Daten (bzw. den Typ). Damit können Client und Server effizienter und mit weniger Fehlern Mediendateien austauschen.

FILE EXTENSIONS

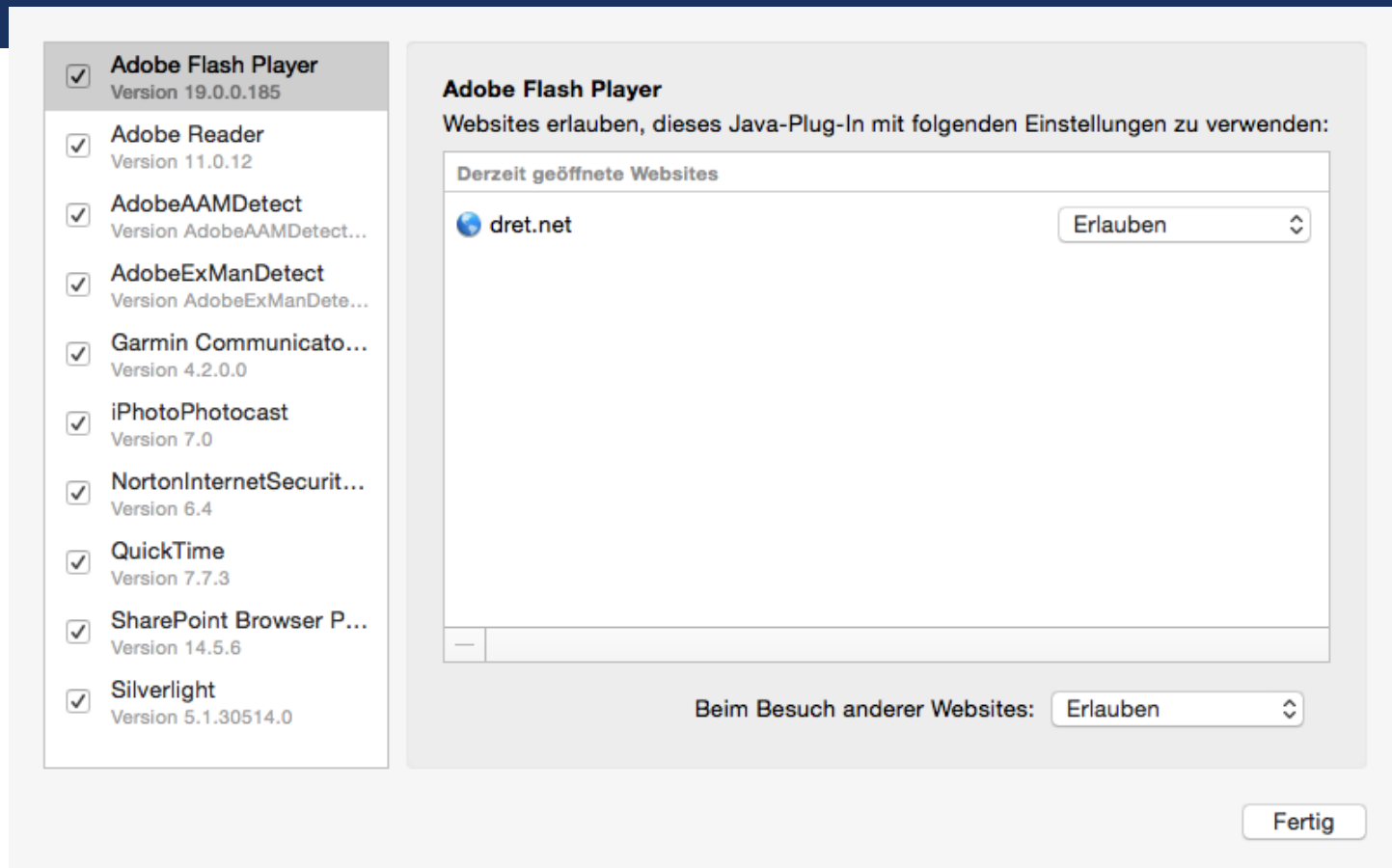
- Ursprung liegt bei der Behandlung von Dateien
- Mit welchem Programm soll das Betriebssystem eine Datei öffnen, damit diese korrekt angezeigt wird (z.B. TIFF)?
- Die Extension definiert den Typ.

- Frage: Was würde passieren, wenn der Browser keine eingebaute Unterstützung für die Anzeige von Bildern hätte?

VERARBEITUNG VON NON-ASCII

- Built-In: HTML, CSS, Script, GIF, JPG,...
- Plug-In: PDF, DOCX, Flash, Java-Bytecode,...
- Verknüpfungen zu externen Programmen: „...soll diese Datei geöffnet werden...“

EXTENSIONS IN SAFARI



ZURÜCK IN DIE REALITÄT



IT-Konzerne konkurrieren auf dem Markt



Behandlung der MIME-Typen kann teils völlig unterschiedlich sein



Implementierung keinesfalls identisch, siehe Apple vs. Microsoft

MIME CONTENT TYPES

Unterteilung in Types und Subtypes

type/subtype

Beispiel: text/html

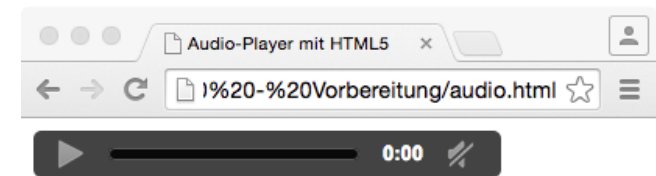
Registrierung bei <http://www.iana.org>

AUSZUG

MIME Type	Beispiel	Definition
text	text/html	Diskreter Typ
image	image/jpeg	Diskreter Typ
audio	audio/basic (8bit PCM)	Diskreter Typ
video	video/mpeg	Diskreter Typ
application	application/xml	Diskreter Typ
multipart	multipart/mixed	Kompositionstyp
message	message/rfc822	Nachrichtentyp

BEISPIEL ANWENDUNG DES TYPUS AUDIO IN HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Audio-Player mit HTML5</title>
  </head>
  <body>
    <audio id="audio_with_controls" controls>
      <source src="track1.mp3" type="audio/mp3" />
      Ihr Browser kennt das HTML5-audio-Element
      noch nicht.
    </audio>
  </body>
</html>
```



ZEICHENKODIERUNG

- Neben dem MIME-Typ ist die Angabe des verwendeten Zeichensatzes von enormer Bedeutung. Nur so kann der Inhalt korrekt verarbeitet werden.
- `text/plain; charset=iso-8859-1`
- er wird in den Header Extensions definiert (MIME part 3)
- Beispiel: „Sch?ne Gr??e“ ohne korrekten Zeichensatz

ZUSAMMENFASSUNG

Die Spezifikation von MIME Typen bringt Ordnung und Semantik in die Übertragung von Inhalten.

Die Qualität ist stark von der Implementierung abhängig.

Gerade PlugIns und Zeichensätze bereiten auch heute noch massive Probleme.



WORLD WIDE WEB - ARCHITEKTUR

WEBARCHITEKTUR UND STANDARDS (TEIL 7)

SIEHE AUCH: [HTTP://WWW.W3.ORG/TR/WEBARCH/SUMMARY.HTML](http://www.w3.org/tr/webarch/summary.html)

TRENDS

Das WWW wurde gestaltet als ein offenes Netzwerk, in dem gegenseitig auf Ressourcen zugegriffen werden können.

Daher stand die URI und der Transport im Vordergrund.

Die Entwicklung richtet sich weiter auf Web-Anwendungen.

Der Trend geht weiter in Richtung von Services.



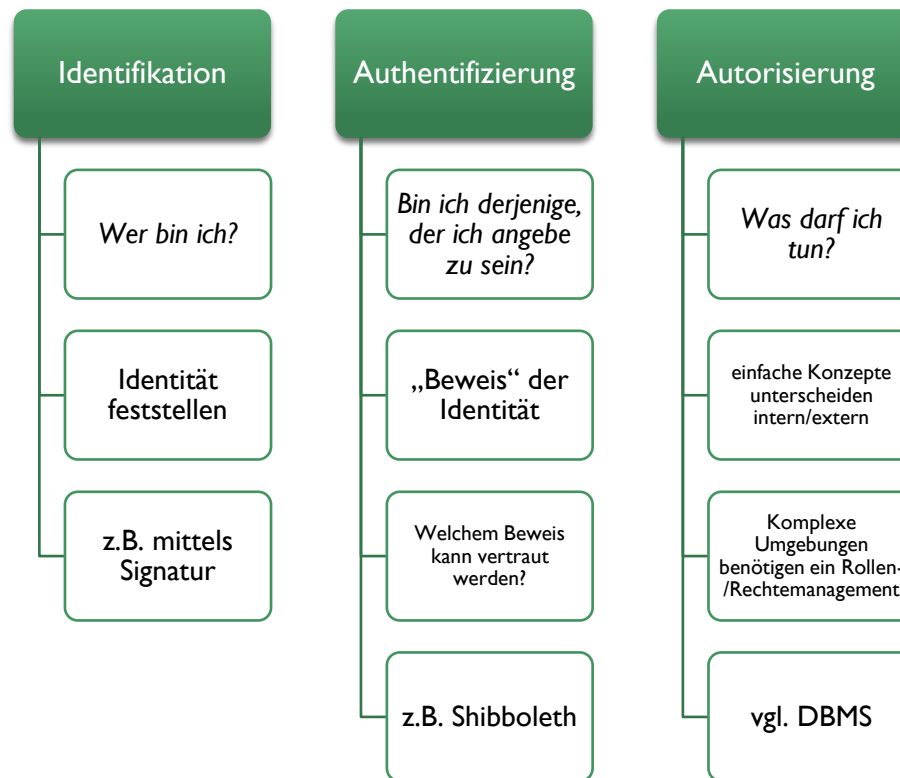
SICHERHEIT

WEBARCHITEKTUR UND STANDARDS (TEIL 8)

DAS PROBLEM

- Informationen werden als Text verschickt (Zeichenkodierung ist bekannt) und jeder kann „mitlesen“.
- Das ist perfekt für „man-in-the-middle“ Angriffe.

KONZEPTE



MEHR DAZU...

Schauen Sie sich doch einmal folgendes Konzept an:

<https://de.wikipedia.org/wiki/Shibboleth>

SICHERHEIT IM BROWSER

- Risiken betreffen:
 - Browser (Datendiebstahl durch Hersteller)
 - Computer (Virus)
 - Netzwerk (Sniffer)
 - Server (böartige Webseiten)
- Lösungen:
 - Browser-Sicherheitseinstellungen (Vertrauen ggü. Hersteller notwendig)
 - Sonderlösungen (z.B. Empfehlungen des BSI, Browser in VM)

SERVERZERTIFIKATE

- Zertifikate sollen „beweisen“, dass ein vom Server angebotener Dienst vertrauenswürdig ist.
- Beispiele: http, smtp, ftp
- Das geht jedoch nur dann, wenn:
 - das Zertifikat von einer Vertrauensstelle ausgestellt wird UND
 - der Client (Browser) die Vertrauensstelle kennt.
- ... *aber vertrauen wir der Vertrauensstelle?*

HTTPS

- Absicherung auf Protokollebene (http = Textnachrichten)
- Verschlüsselung erfolgt oberhalb des Transportlayers (tcp)
 - z.B. Secure Sockets Layer (ssl), transport layer security (tls)
- oder auf Netzwerkebene, z.B. VPN, WPA
- i.d.R. über moderne kryptografische Verfahren wie die asymmetrische Verschlüsselung (Public-Key)



ENDE