

---

# Web-Engineering

## 1 / Einführung

# World Wide Web

---

- Ressourcen, z.B. Dokumente, identifizierbar bereitstellen
- Hypertexte : enthalten Verweise auf Ressourcen
- Multimediale Komponenten (Ton/Bild/Video) einbeziehen
- Ergänzung / Integration anderer Internet-Dienste
  - Wie telnet, ftp, ssh, Email-Dienste (pop,imap,smtp) etc.
- Verwendung vorhandener Standardprotokolle
  - TCP, IP

# Basiskonzepte (1)

---

- Bereitstellung Ressourcen
  - Client- / Server-Architektur
    - (viele) Webclients zur Anforderung und Anzeige von Ressourcen
    - (einzelne) Webserver zur zentralen Bereitstellung und Auslieferung der Ressourcen
    - Zustandsloses Protokoll zur Client-Server-Kommunikation :
      - Anforderung Ressource
      - Auslieferung Ressource

# Basiskonzepte (2)

---

- Ressourcen / Dokumente
  - Hypertext
    - Verweise auf Ressourcen oder Marken in Dokumenten
    - Semantische Netze
  - Strukturierte Dokumente
    - Gliederung
    - Auszeichnung von Texten zur Kennzeichnung der Bedeutung
  - Präsentation
    - Standardisiert
    - benutzerdefiniert

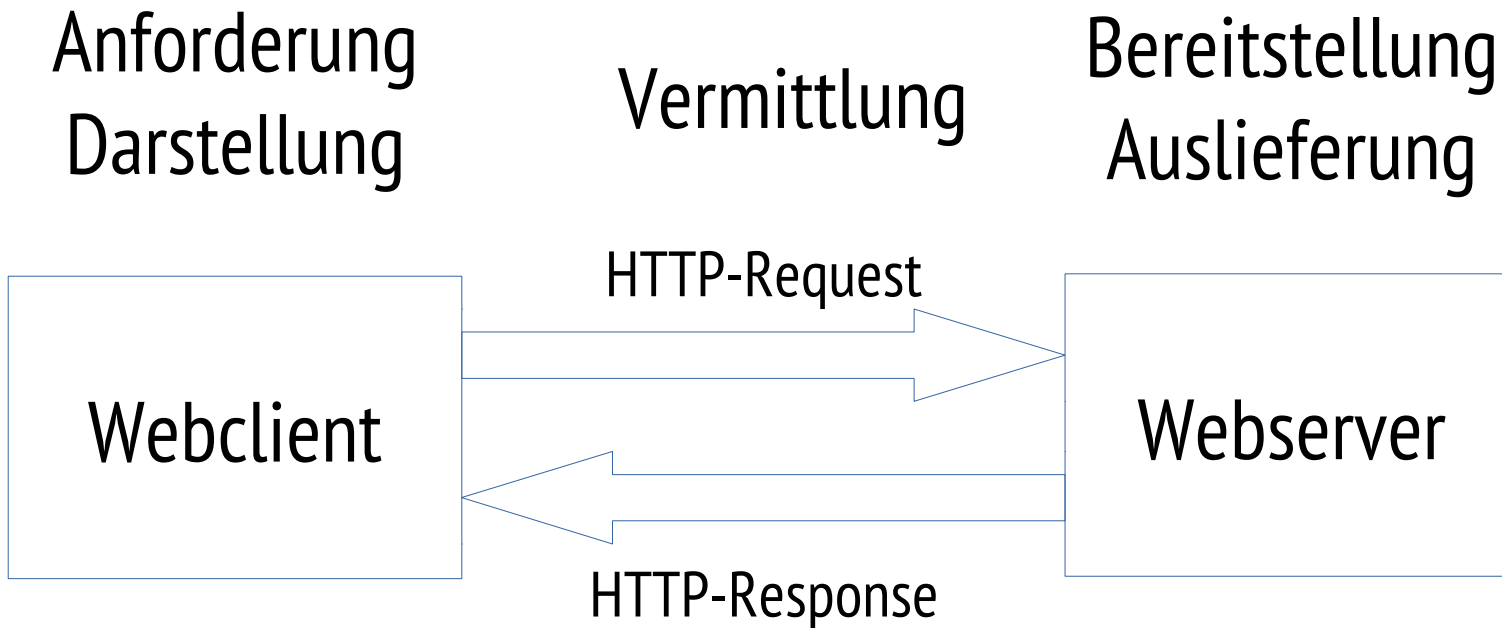
# Basiskonzepte (3): Erweiterungen

---

- Ursprünglich weitgehend statische Sicht auf Dokumente
- Benutzerinteraktionen vorgesehen für
  - Die Verwendung von Verweisen (Hyperlinks)
  - Einfache Formulare
- Weiterentwicklung:
  - „Dokument“ verallgemeinert als Anforderung einer Ressource, die auch dynamisch erstellt, bearbeitet oder ausgeführt werden kann ⇒ Web-Applikationen, Web-Services
  - Größere Dynamik Benutzerschnittstelle
    - Veränderung der clientseitigen Datenstrukturen
    - Erweiterte Funktionalität Interaktionselemente
    - Integration von Medien

# Basiskonzepte (4)

---



# Basiskonzepte (5)

---

- Übertragung textorientiert
  - „Klartext“, keine Verschlüsselung
  - Verschlüsselung nur in den untergeordneten Protokollen
- Übertragung binärer Daten möglich
- Verschiedene Varianten im Einsatz:
  - HTTP 1.0 (seltener)
  - HTTP 1.1 (gängig, effizientere TCP/IP-Verbindungen)
  - HTTP 2.0 (aktuell, effizienter, z.T. bidirektional)

# Webclient (1)

---

- Auch als „Webbrowser“ bezeichnet
  - to browse : blättern, durchsuchen
  - Bezeichnung bezieht sich auf die ursprüngliche Hauptaufgabe (Dokumente anfordern, darstellen)
- Aufgaben
  - Anforderungen (Requests) per HTTP erzeugen
  - Rückmeldungen (Responses) verarbeiten
    - Dokumentbeschreibung interpretieren
    - Dokument als Datenstruktur intern aufbauen
    - Dokument präsentieren („render“)
    - Benutzerinteraktionen verwalten
  - Programme ausführen (z.B. javascript mit Hilfe eines Interpreters)

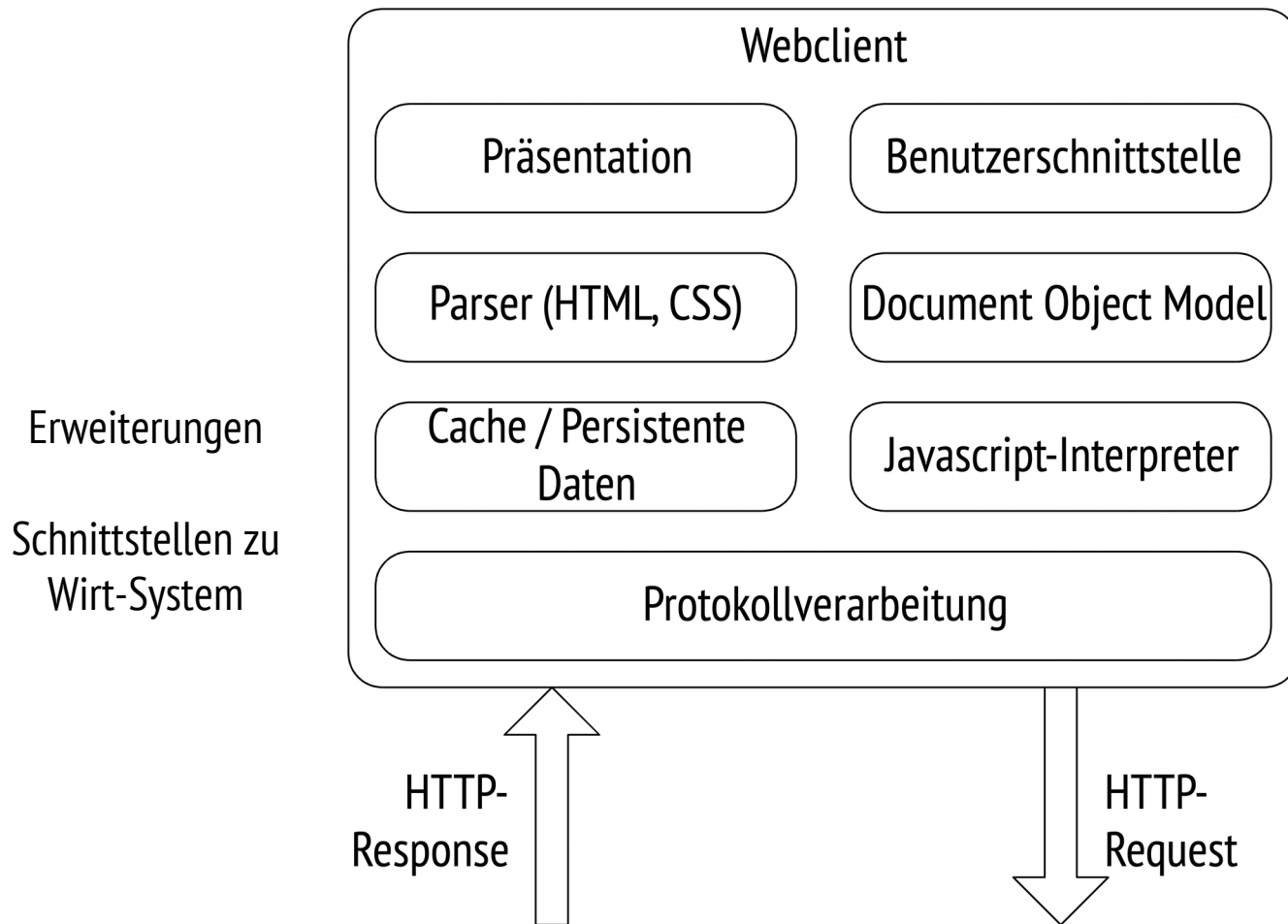


# Webclient (2)

---

- Weitere Eigenschaften
  - Zwischenspeicherung (Cache) von Inhalten und Ressourcen (Medien etc.)
  - Speicherung persistenter Daten auf dem Client-System
    - „Cookies“ / Weitere Mechanismen
    - Einstellungen etc.
  - Kein transparenter Zugriff auf die Fähigkeiten des Client-System !
    - Insbesondere keine transparente Nutzung der Ressourcen des Client (Dateisystem, Prozesse, Hauptspeicher, Dienste)

# Webclient (3)



# Webserver

---

- Aufgaben
  - Protokollbearbeitung
    - Anforderungen (Requests) der Webclients bearbeiten
    - Gelieferte „Namen“ = Adresse des Dokuments auf die physikalische Ablage abbilden
    - Rückmeldungen (Responses) erzeugen
      - Angeforderte Dokumente und Medien übertragen
      - Oder Fehlermeldungen liefern
  - Anforderungen und Fehler protokollieren
  - Erweiterte Request-Bearbeitung:
    - Erzeugung Response durch Programmausführung

# HTTP (1)

---

- HTTP = Hypertext Transfer Protocol
- Protokoll : Vereinbarung zwischen zwei Partnern über den Austausch von Leistungen und Daten
- Zustandslos :
  - Request / Response sind in sich abgeschlossene Transaktion
  - Unabhängig von vorangegangenen Transaktionen
  - Ggf. Zeitüberwachung (Timeout) des Response

# HTTP (2)

---

- Klartext, d.h. keinerlei Binärform oder Verschlüsselung
  - Variante HTTPS : ‚sichere‘ Übertragung durch Verschlüsselung des gesamten Datenaustauschs
- Varianten
  - Neu: Version 2.0
    - Komprimierungen, Zusammenfassung von Anforderungen
  - Aktuell : Version 1.1
    - Dauerhafte Verbindungen möglich (→ Netzwerkmanagement)
    - Eindeutige Kennungen für Dokumente (→ Caching)
  - Version 1.0
    - Einführung des Befehls POST
    - Medientypen verwenden
  - Version 0.9
    - Nicht mehr aktuell, nur noch historisch interessant

# HTTP (3)

---

HTTP2: (siehe RFC 7540)

- abwärtskompatibel
- Performance-Probleme bei HTTP 1.0 / 1.1:
  - Bei vielen Requests sind viele TCP/IP-Verbindungen erforderlich
  - Umfangreiche HTTP-Header verschlechtern die Performance der TCP/IP-Übertragung
- Neue Eigenschaften
  - Maßnahmen zur besseren Nutzung der TCP/IP-Verbindungen
    - Frames: einzelne Requests / Responses aufteilen
    - Flow Control / Prioritäten
    - Request Multiplexing / Streams
  - Server Push
    - Server an Client (ohne vorherige Anfrage durch Client!)

# HTTP (4)

---

- Requests:
  - Message-Header
    - Request-Line
      - Methode, z.B. GET oder POST
      - Angefordertes Dokument / angeforderte Ressource
      - HTTP-Version
    - Verschiedene Informationen / Einstellungen
  - Message-Body
    - Z.B. Daten, die zum Webserver übertragen werden sollen
- Wichtigste Methoden:
  - GET zur Anforderung von Dokumenten
  - POST zur Anforderung von Leistungen mit Übertragung von (umfangreicheren) Daten vom Webclient zum Webserver

# HTTP (5)

---

- Responses:
  - Message-Header
    - Status-Line
      - HTTP-Version
      - Statuscode
      - Erläuterung
  - Message-Body
    - Der eigentliche Inhalt der Antwort



# HTTP (6)

---

- Einfache Datenübertragung mit GET und POST
  - Daten (z.B. aus Formularen) des Webclient werden stets als Key/Value-Paare übertragen
  - Bei GET
    - Daten werden an die Dokument-Adresse angehängen
    - Sind daher bei Webbrowsern i.d.R. in der Adresseingabe sichtbar
    - Sind auch in der Liste der besuchten Dokumente sichtbar
    - Datenmenge ist begrenzt; Sonderzeichen kodieren !
  - Bei POST
    - Getrennte Übertragung der Daten
    - Umfangreicher möglich

# HTTP (7)

---

- Weitere Methoden
  - PUT : Dateien hochladen / Daten ändern
  - DELETE : Ressource auf dem Server löschen
  - HEAD : nur Header senden
  - TRACE : Anfrage spiegeln
  - OPTIONS : Fähigkeiten des Server mitteilen
  - CONNECT : spezielle Verbindungen herstellen
- Festlegungen zur Semantik: RFC 7231 (→ Präsentation)

# Identifikation von Ressourcen (1)

---

- Uniform Resource Identifier
  - Ziel : nach einheitlicher Syntax benannte Adressen, die im Anwendungskontext eindeutig sind
  - Anwendungskontext Web : URI muss „weltweit“ eindeutig sein
    - D.h. aber auch : ein im Anwendungskontext Web gültigen URI kann man als eindeutigen Bezeichner nutzen !
  - i.d.R. logische Adresse, d.h. Umsetzung in eine physikalische Adresse (z.B. welcher Webserver, welche Datei auf dem Webserver) durch verschiedene Mechanismen
- Unterbegriff URL (uniform resource locator) : Auffinden von Ressourcen beschreiben
- URN (uniform resource name) : inzwischen veralteter Begriff

# Identifikation von Ressourcen (2)

---

- URI-Syntax allgemein :
  - <Schema>:<Schema-spezifischer Teil>
  - Typische Schemata : http ftp mailto file
- Schema http :  
http://[<Benutzer>[:<Passwort>]@]  
Server[:<Port>]/  
[<Pfad>] [<?<Anfrage>] [<#<Fragment>]
- Beispiele :  
http://www.hsnr.de/  
http://lionel.kr.hs-niederrhein.de/~beims/testseite.html#web

# Identifikation von Ressourcen (3)

---

- Schema http:
  - Authority : Benutzer ... Port
  - Server : eindeutige Identifikation einer Domain oder IP-Adresse
  - Port : Kennzeichnung eines Dienstes, der auf dem Server ausgeführt wird
  - Pfad : der logische Zugriffspfad zur Ressource, muss vom Webserver interpretiert und umgesetzt werden
  - Anfrage : bei GET übertragene Werte
  - Fragment : Verweis auf einen *Anker* im Dokument (nur sinnvoll, wenn die Ressource ein HTML- oder XHTML-Dokument ist)

# SGML, HTML, XML, XHTML (1)

---

- SGML : Standard Generalized Markup Language
  - Texte mit Auszeichnungen versehen, die Hinweise auf die logische Struktur und Bedeutung geben
  - i.d.R. ergeben sich Baumstrukturen
  - Strikte Trennung von Struktur (SGML-Dokument) und Präsentation
    - Bei einheitlicher Struktur kann Präsentation auf unterschiedlichen Medien in verschiedener Weise erfolgen
  - Definition der Dokument-Struktur
    - Sog. DTD = Document Type Definition
    - Damit Überprüfung der Gültigkeit eines Dokuments möglich (Validierung)

# SGML, HTML, XML, XHTML (2)

---

- Markup : Elemente, i.d.R. mit öffnender und schließender Marke (*Tag*)  
`<title>Beispiel eines Titel</title>`
- Typische Anwendungen von SGML:
  - HTML
  - Docbook
- Nachteil von SGML:
  - Kompliziert in der Anwendung

# SGML, HTML, XML, XHTML (3)

---

- HTML
  - Ziel : Beschreibung von Hypertext-Dokumenten
  - Im Laufe der Entwicklung (Versionen 2, 3.2, 4, 5) erhebliche Erweiterungen und Änderungen
    - Dabei auch : Einführung von Elementen, die die Präsentation beeinflussen (sollen)
    - Beispiele :
      - Font-Element zur Spezifikation der Schriftart
      - Bold-Element zur Hervorhebung in einer bestimmten Weise
      - Italic-Element zur Hervorhebung in einer bestimmten Weise



# SGML, HTML, XML, XHTML (4)

---

- HTML
  - In neueren Versionen korrekt definiert, in früheren Versionen dagegen nicht eindeutig oder korrekt
  - Webbrowser lassen daher viele Beschreibungsfehler zu !
- Prinzipieller Aufbau eines HTML-Dokuments

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

# SGML, HTML, XML, XHTML (5)

---

- Head-Abschnitt
  - Nimmt Angaben zum Dokument auf
    - Z.B. Stichworte zur Charakterisierung des Inhalts, kann durch Suchmaschinen ausgewertet werden
  - Führt benötigte Ressourcen auf
- Body-Abschnitt
  - Enthält den eigentlichen Dokumentinhalt
  - Nur dieser Teil wird zur Präsentation ausgewertet

# SGML, HTML, XML, XHTML (6)

---

- XML
  - Vereinfachung von SGML
  - Ursprünglich ebenfalls nur für das Electronic Publishing gedacht
  - Inzwischen Beschreibungsstandard für viele Arten von Daten
  - Prüfmöglichkeiten :
    - „wohl geformt“ : wurde die XML-Syntax eingehalten ?
    - „valide“ : wurden die XML-Elemente richtig eingesetzt ?

# SGML, HTML, XML, XHTML (7)

---

- XML, Beispiel

```
<Vorlesung>  
  <Titel>Web-Engineering</Titel>  
  <Dozent>Beims</Dozent>  
</Vorlesung>
```

- „wohl geformt“ : prüfbar, weil nur die grundsätzliche XML-Syntax geprüft wird
- „valide“ : kann nicht geprüft werden, weil eine Beschreibung der zugelassenen Elemente und der zulässigen Baumstruktur fehlt
  - Solche Beschreibungen können als DTD (wie bei SGML) oder mit XML-Schema angegeben werden

# SGML, HTML, XML, XHTML (8)

---

- XHTML
  - Re-Definition von HTML mit den Mitteln von XML
  - Syntax eindeutiger und strenger
    - Schließende Marken zwingend erforderlich
    - Kleinschreibung
    - Attribute von Elementen in Anführungszeichen
  - Präsentationsspezifische Elemente entfernt
  - erweiterbar

# HTML5 (1)

---

- Weiterentwicklung von HTML4
- Getrieben durch Herstellerkonsortium, Übernahme auch durch das W3C
  - Konkurrierende Spezifikationen W3C – Hersteller
  - Konkurrierende Vorgehensweisen bei der Weiterentwicklung
- Löst HTML4 und XHTML ab
- Wichtige Erweiterungen wie *canvas*, neue Elemente zur Seitengestaltung, endgültiger Verzicht auf einige problematische Elemente
- Verschiedene Erweiterungen wie WebStorage, WebWorkers, WebSockets

# HTML5 (2)

---

## Wichtige neue Elemente:

- Zur Strukturierung der Inhalte: `article`, `footer`, `header`, `main`, `section`, `figure` / `figcaption`
- Zur Verbesserung der Benutzbarkeit: `aside`, `menu`, `nav`, weitere Sub-Typen bei `input`
- Zur Aufnahme spezieller Inhalte: `canvas`, `audio`, `video`, `svg`, `output`

# CSS

---

- Cascading Style Sheets

- Ziel : auf einfachem Weg angeben, wie die Elemente in einem (X)HTML-Dokument präsentiert werden
- Besteht aus Regeln
  - Die einen *Selektor* aufweisen, mit dem festgelegt wird, auf welche Elemente die Regel angewendet wird
  - Die keine, eine oder viele Key-Value-Paare enthalten, die die einzelnen Darstellungseigenschaften festlegen
- Beispiel : alle Absätze (Paragraphen : p) mit rotem Hintergrund versehen

```
p { background-color: red; }
```



# CSS3

---

- Weiterentwicklung des CSS-Standards
- In Zusammenhang mit HTML5
- Besteht aus vielen Einzelspezifikationen, mit unterschiedlichem Status (Gültigkeit / Verabschiedung als Standard)
- Relevant:
  - wesentliche Erweiterungen bei den Selektoren
  - Erweiterte Möglichkeiten bei Hintergründen
  - Web-Fonts