



# HyperText Transfer Protocol HTTP



- OSI bedeutet Open System Interconnection (Offenes System für Kommunikationsverbindungen).
- Das Schichtenmodell beschreibt schematisch, wie die Kommunikation zwischen Computersystemen funktioniert.
- Jede Schicht definiert dabei bestimmte Protokolle bzw. Funktionen, die für eine bestimmte Aufgabe notwendig sind.
- Bei einer einfachen Kommunikation (zwischen exakt 2 Systemen) wird das Schichtenmodell zwei Mal durchlaufen:
  - Das erste Mal von oben nach unten beim Sender.
  - Das zweite Mal von unten nach oben beim Empfänger.
- Da es sich bei dem Schichtenmodell um ein Referenzmodell handelt, kann jede Schicht angepasst oder ausgetauscht werden.



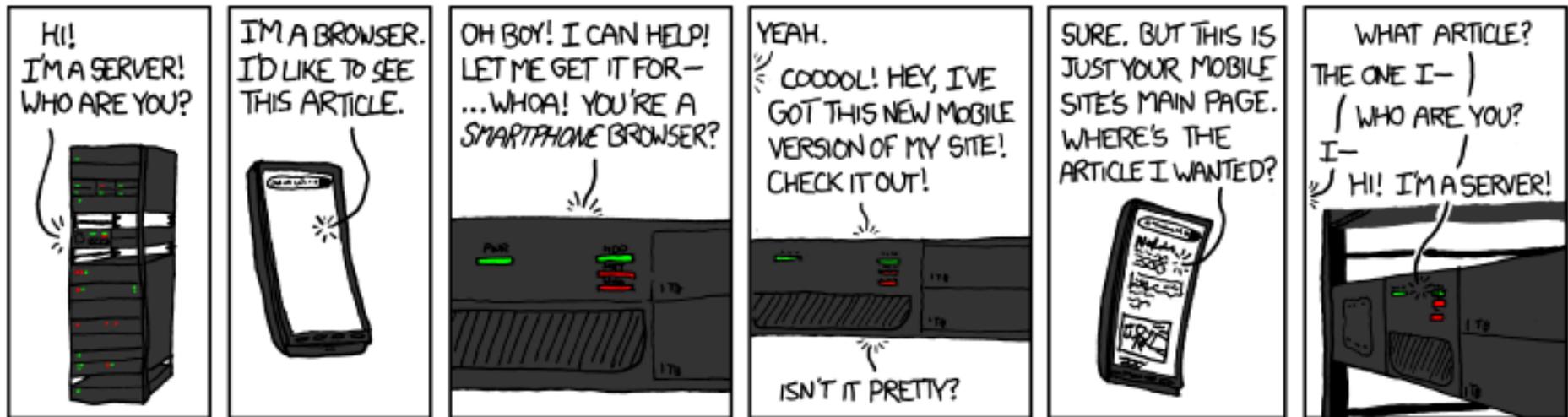
OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, OPC UA
Darstellung (6)		
Sitzung (5)		
		SOCKS
Transport (4)	Transport	TCP, UDP, SCTP
Vermittlung (3)	Internet	IP (IPv4, IPv6), ICMP (über IP)
Sicherung (2)	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI, IPoAC
Bitübertragung (1)		



- Das Hypertext Transfer Protocol ist ein Protokoll auf Anwendungsebene zur Übertragung von Daten über ein Netzwerk.
- Es wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web in einen Webbrowser zu laden.
- HTTP ist ein einfaches Klartext-Protokoll, bei dem standardmäßig keine Verschlüsselung des Übertragungsinhalts stattfindet.
- Es übermittelt textbasiert in ASCII-Zeichen und ist zeilenorientiert.
- Durch Erweiterung seiner Anfragemethoden, Header-Informationen und Statuscodes ist HTTP nicht auf Hypertext beschränkt, sondern wird zunehmend zum Austausch beliebiger Daten verwendet.



- HTTP ist ein zustandsloses Protokoll.
- Das bedeutet, dass jede Anfrage an den Server unabhängig von vorherigen Anfragen ist.
- Möchte man über mehrere Seiten hinweg einen angemeldeten Benutzer wiedererkennen, muss die Anwendung eine Sitzung implementieren.
- Dies geschieht über Session-ID's, die als Cookies gespeichert werden.





- Jede Nachricht besteht dabei aus zwei Teilen,
  - dem Nachrichtenkopf (Header) und
  - dem Nachrichtenkörper (Body).
- Der Nachrichtenkopf enthält Informationen über den Nachrichtenkörper wie verwendete Kodierungen oder den Inhaltstyp, damit dieser vom Empfänger korrekt interpretiert werden kann.
- Der Nachrichtenkörper enthält die Nutzdaten.
- Wenn auf einer Webseite der Link zur URL `http://www.example.net/infotext.html` aktiviert wird, so wird an den Computer mit dem Hostnamen `www.example.net` die Anfrage gerichtet, die Ressource `/infotext.html` zurückzusenden.



- Zur Kommunikation ist HTTP auf ein zuverlässiges Transportprotokoll angewiesen, wofür in nahezu allen Fällen TCP verwendet wird.
- Die Kommunikationseinheiten in HTTP zwischen Client und Server werden als Nachrichten bezeichnet, von denen es zwei unterschiedliche Arten gibt:
  - die Anfrage (engl. Request) vom Client an den Server und
  - die Antwort (engl. Response) als Reaktion darauf vom Server zum Client.
- Der offizielle TCP-Port für HTTP ist 80, weitere typische Ports für Test-Server sind 8080 und 8000.



- Ein Port ist der Teil einer Netzwerk-Adresse, der die Zuordnung von Verbindungen und Datenpaketen zu Server- und Client-Programmen durch Betriebssysteme bewirkt.
- Zu jeder Verbindung dieser beiden Protokolle gehören zwei Ports, je einer auf Seiten des Clients und des Servers.
- Gültige Portnummern sind 1-65535.
- Primär sind Ports ein Merkmal zur Unterscheidung mehrerer Verbindungen zwischen demselben Paar von Kommunikations-Endpunkten.
- Ein solcher Endpunkt besteht aus einer IP-Adresse und einem Port und wird als Socket bezeichnet.



1. Startende Server fordern beim Betriebssystem bestimmte Ports an, um Verbindungen annehmen zu können.
2. Dies sind in aller Regel well-known Ports, da hier die Portnummer nicht veröffentlicht werden bzw. bekannt sein muss.
3. Die Nummern sind vielmehr den betreffenden Programmen fest zugeordnet.
4. Clients fordern bei ihrem Betriebssystem i.d.R. einen zufälligen Port an (meist >30000), um Verbindungen aufbauen zu können.
5. Der Verbindungsaufbau erfolgt stets durch den Client, der dazu den Port des Servers kennen muss.
6. Durch den Verbindungsaufbau erfährt der Server den Port des Clients, sodass er diesem konkreten Anwendungsprozess des Clients direkt antworten kann.



# Beispiel für die Notwendigkeit von Ports bei HTTP



- Ein Webbrowser kann während eines Downloads einen weiteren Download vom selben Server starten.
- Das funktioniert, weil der Browser auf dem Client dann einen weiteren Port öffnet und eine zusätzliche Verbindung zum selben Port 80 des Servers aufbaut.
- Der Server antwortet den unterschiedlichen Ports des Browsers mit unterschiedlichen jeweils zusammengehörigen Inhalten.
- Für eine Unterscheidung der Verbindungen genügen also verschiedene Portnummern an nur einem der beiden Endpunkte.

