

basieren auf Client-Server-Infrastrukturen. Bei jeder Aktion, bei der über das Internet zwischen zwei Rechnern Daten übermittelt werden, gibt es einen Host, der Dienste bereitstellt und auf Anfragen reagiert (Server), und einen Host, der Anfragen stellt und die Antworten verarbeitet (Client). Dienste, die von einem Server bereitgestellt werden, sind z. B. die Übermittlung von Webseiten, die Bereitstellung von Datenbankzugriffen oder die Vermittlung von E-Mails.

Die Bezeichnungen "Client" und "Server" verdeutlichen die Beziehung zwischen den beiden grundsätzlich involvierten Rechnerarten: Der Client stellt eine Anfrage zur Benutzung eines bestimmten Dienstes an den Server, der wiederum die Anfrage entgegennimmt, auswertet und den Client durch das Bereitstellen des gewünschten Dienstes bedient. Diese Kommunikation erfolgt mittels so genannter Transaktionen. Eine Transaktion muss bestimmte Kriterien erfüllen, so darf sie z. B. keine anderen Transaktionen beeinflussen und muss nach ihrer Ausführung einen konsistenten Datenbestand hinterlassen.

Protokolle

Für die Vernetzung von Rechnern ist ein einheitliches Protokoll der Datenübertragung notwendig, um allen unterschiedlichen Rechnerarten und Betriebssystemen den Datenaustausch untereinander zu ermöglichen. Dies übernimmt das Protokoll TCP/IP ("TCP over IP" ausgesprochen). Es stellt den kleinsten gemeinsamen Nenner des Datenverkehrs im Internet dar. Alle Daten, die im Internet transportiert werden, sei es in Form von Webseiten, E-Mails oder Downloads, werden auf dieselbe Art und Weise übermittelt. TCP/IP setzt sich aus folgenden Protokollen zusammen, die bei der Datenübertragung zusammenarbeiten:

TCP = Transmission Control Protocol (Protokoll für Übertragungskontrolle)

IP = Internet Protocol

Alle Daten, die über das Internet übertragen werden, z. B. eine E-Mail, sind in kleine Pakete unterteilt. Jedes Paket enthält Angaben, an welche Adresse es geschickt werden soll und eine Nummer, die angibt, das wievielte Paket innerhalb der Sendung es ist. Diese Adressierung übernimmt das IP. TCP sorgt dafür, dass die Pakete auch wirklich beim Empfänger ankommen und in der richtigen Reihenfolge wieder zusammengesetzt werden.

Damit das Internet Protocol (IP) seine Arbeit auch ordnungsgemäß durchführen kann, wird jedem Rechner, der an das Internet angeschlossen ist, eine eindeutige IP-Adresse zugewiesen. Rechnern, die ständig Verbindung mit dem Internet haben, wird eine dauerhafte IP-Adresse zugewiesen (statische IP-Adresse). Hierzu gehören in der Regel auch alle Rechner innerhalb eines Universitätsnetzes, es sei denn, sie sind Teil eines WLAN. Für private Nutzer, die sich nur gelegentlich über einen Internet-Anbieter per Modem ins Internet einwählen, stellt dieser Anbieter für jede Einwahl automatisch eine wechselnde IP-Adresse für den Rechner zur Verfügung (dynamische IP-Adresse).

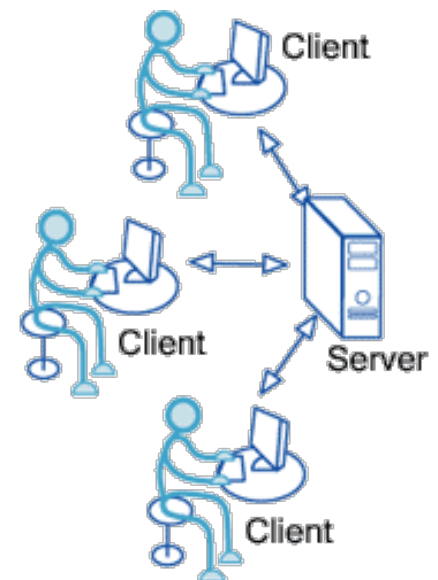
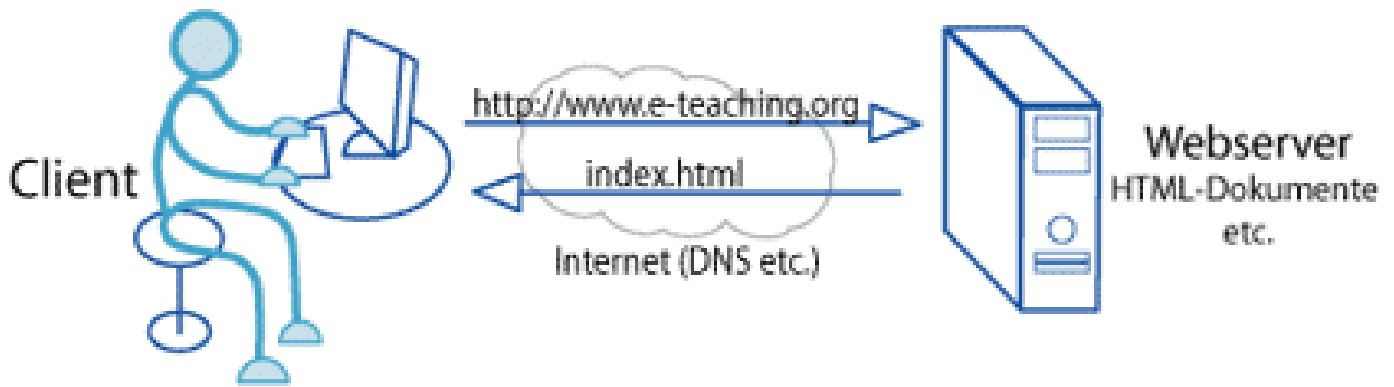


Abb. 1: Client-Server-Struktur



Kommunikation zwischen Client(-Browser) und Webserver

Anwendungen

- **Webserver:** Einen Großteil aller im Internet befindlichen Server bilden die Webserver. Diese Server sind darauf spezialisiert, Client-Anfragen, die über das Hypertext-Transfer-Protokoll (HTTP) erfolgen, zu beantworten. Ein Autor einer Website legt die einzelnen Dateien (zumeist in HTML verfasste Dokumente) auf einem Webserver ab, indem er sie z. B. per FTP dorthin überträgt (sofern der Webserver auch dieses Protokoll unterstützt). Die Website ist fortan über die IP-Adresse des Webserver (z. B. 85.56.13.2), bzw. über die durch das Domain Name System darauf abgebildete menschenlesbarere Textform (z. B. www.e-teaching.org) erreichbar. Ein Interessent kann diese Website mit Hilfe eines Browsers, der auf seinem Client-Rechner installiert ist, erreichen. Der Browser stellt dabei die HTTP-Anfrage an den Server. Dieser interpretiert die Anfrage und sendet die Dokumente wiederum per HTTP an den Browser. Der Browser stellt die Dokumente schließlich auf dem Bildschirm des Interessenten dar.
- **Mailserver:** Auch der E-Mail-Verkehr findet auf Grundlage einer Client-Server-Infrastruktur statt. Mailserver sind dafür zuständig, E-Mails zu empfangen, zu speichern, zu versenden oder weiterzuleiten. Auch hierfür muss der Server spezielle Protokolle wie SMTP, POP3 oder IMAP implementieren. Wenn ein Anwender seine E-Mails zu Hause mit einem E-Mail-Client wie z. B. Thunderbird (siehe Steckbrief) verwaltet, so kommuniziert dieser Client im Hintergrund bei jeder Aktion mit dem zuständigen Mailserver.
- **Print-Server:** In vielen Institutionen fungieren Print-Server als Mittler zwischen den Rechnern der Angestellten, den Clients, und den verschiedenen bereitgestellten Druckern. Die Print-Server nehmen die Druckaufträge der Clients entgegen und senden sie der Reihe nach an die entsprechenden Drucker. Manche, vor allem teurere Drucker haben die Funktionalitäten eines Print-Servers bereits integriert und können direkt an das Netz angeschlossen werden und somit als Netzwerkdrucker dienen.
- **Anwendungsserver:** Ein weiteres vielfältiges Anwendungsgebiet des Client-Server-Paradigmas bilden die Anwendungsserver. Das sind Server, die bestimmte Anwendungen bereitstellen. Dies können beispielsweise die Interpreter von (serverseitigen) Skriptsprachen wie PHP, Perl oder Python oder Schnittstellen für Datenbank Anwendungen sein. Anwendungsserver bieten somit eine Laufzeitumgebung für eine vom Client aufgerufene Anwendung. Sie werden üblicherweise als Ergänzung eines bereits installierten Webserver wie Apache implementiert. Ein Beispiel eines Anwendungsservers ist der quelloffene Webanwendungsserver ZOPE, der auch als Grundlage des Content Management Systems (CMS) Plone (siehe Steckbrief) verwendet wird, durch welches das Portal e-teaching.org verwaltet wird. ZOPE bietet u. a. eine eigene Objektdatenbank, einen Interpreter der Programmiersprache Python und einen eigenen HTTP-Server. Weitere Beispiele sind Anwendungsserver wie z. B. Apache Tomcat, die die Ausführung von in Java geschriebenen Programmen ermöglichen.

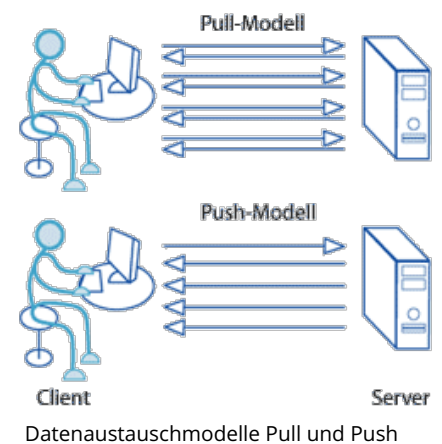
Klassifizierung

Neben der rein funktionalen Unterscheidung können Server auch bezüglich ihrer Betriebsweise differenziert werden. So gibt es einerseits zustandsabhängige Server, die während der Kommunikation mit einem Client bestimmte Informationen zwischenspeichern. Ein Beispiel hierfür ist ein FTP-Server, der vor der eigentlichen Datenübertragung Authentifizierungsinformationen mit dem Client austauscht und sich diese „merken“ muss. Andererseits gibt es

zustandsignorierende Server, die jede Anfrage eines Clients als unabhängige Transaktion behandeln. Dies verringert zwar die Serverkomplexität, da sich dieser nicht um die Zwischenspeicherung der Informationen zu kümmern braucht, vergrößert jedoch den Umfang einer jeden Anfrage, da bestimmte Informationen ggf. jedes Mal neu übertragen werden müssen. Stellvertretend hierfür sind einfache Webserver, die als Information lediglich die vom Client gewünschte URL benötigen.

Auch Clients lassen sich in verschiedene Kategorien einordnen. So wird beispielsweise zwischen der auf einem Client vorhandenen Hard- und Softwareausstattung und der damit verbundenen Rechenintelligenz des Clients unterschieden. Thin Clients sind in ihrer Ausstattung auf Funktionalitäten beschränkt, die zur Ein- und Ausgabe benötigt werden. Sie besitzen somit die geringste Rechenintelligenz und sind nur mit den nötigsten Ressourcen ausgestattet, beispielsweise werden keine Datenspeichermedien benötigt. Nicht nur gewöhnliche Anwendungen sondern auch das Betriebssystem wird meistens über das Netz vom Server oder über externe Datenträger gestartet. Im Gegensatz hierzu sind Fat Clients mit allen Hard- und Softwarekomponenten ausgestattet, die für den gesamten Datenverarbeitungsprozess benötigt werden. Dazu zählen nicht nur die dabei verwendeten Programme sondern beispielsweise auch die zur grafischen Ausgabe benötigte, meistens sehr leistungsfähige Grafikkarte. Lediglich die Datenverwaltung wird Servern überlassen, welche die Daten z. B. in Datenbanken ablegen. Die Rich Clients stellen eine flexiblere Form der Fat Clients dar. Sie können nicht nur eine sondern diverse Aufgaben bewältigen und sind somit wesentlich anpassungsfähiger.

Client-Server-Architekturen können auch bezüglich des Datenaustauschmodells unterschieden werden. Zum einen gibt es das Client-Pull-Modell, bei welchem der Client für die Steuerung der Kommunikation zuständig ist. Am einfachsten lässt sich dies erneut am Beispiel des WWW erklären: Der Browser (als Client) sendet eine Anfrage nach einer bestimmten Website an den Server. Klickt der Benutzer nun auf einen Navigationslink innerhalb dieser Website, so stellt der Browser damit die nächste Anfrage, die vom Server beantwortet werden soll. Der Client entscheidet also, wann welche Daten übertragen werden sollen. Als dazu gegensätzlicher Ansatz existiert das Server-Push-Modell. Hierbei obliegt die Kontrolle über den Datenaustausch dem Server. Ein Beispiel für den Einsatz dieses Modells ist das so genannte Streaming: Der Client sendet lediglich eine initiale Anfrage, welche Video- oder Audiodatei er betrachten bzw. anhören möchte. Die Auslieferung der einzelnen Teildaten wird fortan vom Server gesteuert. Je nach momentan herrschendem Andrang kann er auf diese Weise seine Ressourcen optimal einteilen.



Datenaustauschmodelle Pull und Push

Vorteile

Client-Server-Systeme bieten den Vorteil, dass z. B. Softwareaktualisierungen nur auf wenigen Systemen (den Servern) durchgeführt werden müssen. Die Clients bekommen davon meistens gar nichts mit. Zudem können auf den Client-Rechnern viele Ressourcen eingespart werden, da Daten zentral verfügbar sind und somit keine Mehrfachspeicherung der selben Daten auf den verschiedenen Client-Rechnern anfällt. Durch die zentrale Datenhaltung kann auch sehr leicht eine Zugriffskontrolle für die Daten verwirklicht werden. Clients müssen sich vor dem Zugriff auf bestimmte Daten beispielsweise bei dem Server authentifizieren.

Nachteile

Die Zentralisierung birgt aber auch ihre Gefahren. So droht beispielsweise bei einem gezielten Angriff auf den Server oder einem Hardware-Defekt der Ausfall des gesamten Netzes, da die Clients nicht mehr auf die Daten und Dienste des Servers zugreifen können. Die Server derartiger Infrastrukturen gelten daher als „single point of failure“. Eine redundante Ausstattung des Netzes mit mehreren Servern, die den gleichen Dienst erbringen und die gleichen Daten speichern, schafft in gewissem Maße Abhilfe und wird heutzutage in vielen Netzen, vor allem auch im Internet, eingesetzt. Zudem gilt bei Client-Server-Architekturen, dass die Bandbreite, die einzelnen Client-Server-Verbindungen zur Verfügung steht, mit steigender Anzahl solcher Verbindungen sinkt. Bei vielen gleichzeitigen Client-Anfragen kann auf Serverseite daher ein Stau entstehen, der längere Wartezeiten für die Clients verursacht. Die Skalierbarkeit, d. h.

die Anpassung an steigende Anforderungen, z. B. in Form einer zunehmenden Anzahl an Clients, stellt somit eines der größten Probleme dar, mit denen Client-Server-Infrastrukturen zu kämpfen haben.

Letzte Änderung: 21.07.2015

Zitation

e-teaching.org (2015). Client-Server. Zuletzt geändert am 21.07.2015. Leibniz-Institut für Wissensmedien:
https://www.e-teaching.org/technik/vernetzung/architektur/client-server/index_html. Zugriff am 27.10.2021

Barrierefreiheit [Direkt zum Inhalt](#) [Übersicht](#) [Erweiterte Suche](#) [Direkt zur Navigation](#) [Kontakt](#)