

2 Das Client-Server-Modell

Es ist so alltäglich: Wir öffnen am PC oder Handy einen Browser, geben dort eine Adresse ein, und schon erhalten wir in kürzester Zeit die gewünschten Informationen. In diesem Fall stellen der PC bzw. das Handy mitsamt dem Browser den so genannten **Client** dar; dieser nimmt über das Internet (oder ein anderes Netzwerk) Kontakt mit einem anderen Rechner auf und fragt nach der gewünschten Webseiten-Datei. Diese Anfrage bezeichnet man auch als Request. Clients können aber nicht nur nach Webseiten, sondern auch nach anderen Dienstleistungen fragen (z. B. Emails). Die Rechner im Netzwerk, welche derartige Dienstleistungen zur Verfügung stellen, bezeichnet man als **Server**.

Je nach Situation bezeichnet man lediglich das Programm als Client, manchmal wiederum das Gerät, auf dem dieses Programm läuft. In vielen Fällen ist aber in dieser Hinsicht keine scharfe Unterscheidung nötig. Gleiches gilt für den Server.

Das Zusammenspiel von Client und Server wollen wir uns an einem einfachen Modell veranschaulichen: Die Firma *Elektronik GmbH* bietet verschiedene Dienste an. U. A. kann man bei ihr elektronische Geräte beziehen, aber auch Anleitungen zu diesem Thema. Außerdem vermittelt sie auch Handwerker aus verschiedenen Branchen, die Hilfe leisten können. Wenn nun Kunden (Klienten!) eine Dienstleistung wünschen, schreiben sie einen Brief an die Adresse der Firma. Zusätzlich geben sie aber auch die Nummer der Abteilung an, die für die gewünschte Dienstleistung zuständig ist.

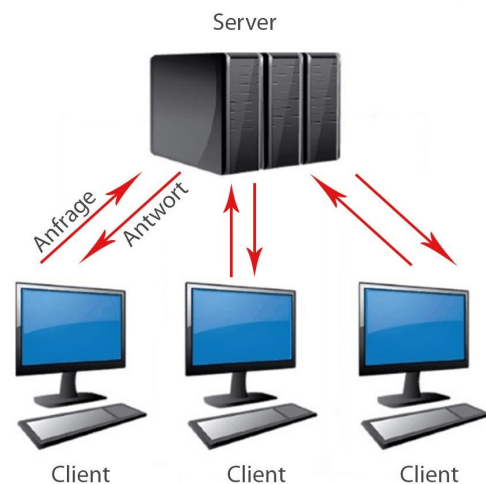


Abbildung 1

Dienstleistungsnummer	Dienstleistung
25	Artikel
120	Anleitungen
...	...
555	Geschäftsbedingungen

Diese Vorgehensweise bietet den Vorteil, dass die Anfrage des Kunden direkt an die richtige Abteilung gelangt, ohne dass es eines Verteilers bedarf, der das Anliegen des Kunden erkundet und den Brief dann entsprechend weiterleitet. Die Mitarbeiter der Abteilung 555 haben es besonders einfach: Sie brauchen sich nur die Adresse des Kunden (Absender) anzuschauen und die Geschäftsbedingungen an diese Adresse schicken.

Bei diesem Vorgang spielt der Transport der Briefe und Waren eine wichtige Rolle. Das Transportunternehmen muss die Briefe und Waren von einer Adresse zur anderen befördern. Dabei kann es verschiedene Transportmittel (LKW, Bahn, Fahrrad...) benutzen. Diese können die unterschiedlichsten Routen wählen, um ans Ziel zu gelangen. Weder unsere Firma noch die Kunden müssen hierüber Details kennen. Für sie reicht es aus, die Waren oder Briefe mit einer vollständigen Adresse zu versehen und dem Transportunternehmen zu übergeben.

Letztlich läuft die Interaktion zwischen dem Kunden und der *Elektronik GmbH* auf verschiedenen Ebenen ab:

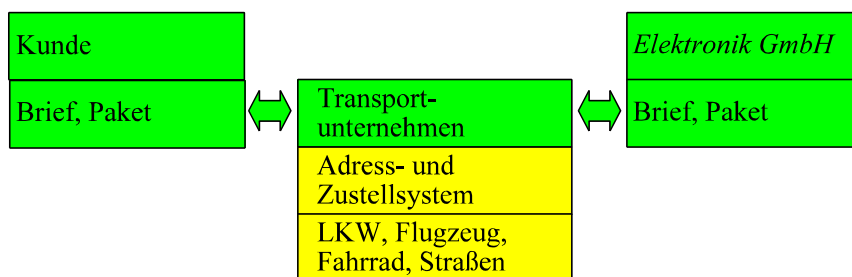


Abbildung 2

Auch die Kommunikation über das Internet kann man sich aus verschiedenen **Schichten** (engl. Layer) aufgebaut vorstellen (vereinfachtes **OSI-Schichten-Modell**):

Schicht	Bedeutung/Bezug	Protokolle	Kopplungsgeräte
5 bis 7	Anwendung	HTTP, FTP, SMTP	
4	Transport	TCP, UDP	Router
3	Netzwerk	IP	Router
2	Datenübertragung	Ethernet, Token Ring, ARCNET, Wlan	Bridge, Switch
1	Physikalische Übertragung		Hub, Repeater

Bei konkreten Realisierungen dieses Modells sind die Grenzen zwischen den einzelnen Schichten manchmal etwas unscharf; dies trifft aber ebenso auf unser Kunden-Firmen-Modell zu.

In der oberen Schicht befinden sich Anwendungsprogramme, mit denen z. B. Webseiten oder Emails von Clients und Servern über das Internet angefragt bzw. ausgetauscht werden können. Die Clients entsprechen den Kunden und ein Server entspricht einer Firma, z. B. unserer *Elektronik GmbH*. Bei der Übertragung im Internet greift man auf die so genannte **Transportschicht** zurück. Diese stellt als Zustellsystem Netzwerk-Protokolle zur Verfügung; von denen wird hier nur das so genannte **TCP** (Transmission Control Protocol) behandeln. Dieses Protokoll muss sich nicht im einzelnen darum kümmern, wie die Daten sich durch das Netzwerk bewegen oder welche Trans-

portmedien (Glasfaser, Kabel oder Funkverbindungen) dazu benutzt werden – genau so wenig, wie der Zusteller des Transportunternehmens sich nicht Gedanken darüber machen muss, wie das von ihm gesteuerte Fahrzeug funktioniert oder gebaut worden ist.

Wenn wir Programme mit Internet-Kommunikation für unseren ESP32 schreiben wollen, dann müssen wir lediglich wissen, wie unser Programm mit der Transportschicht arbeiten kann. Dazu stellt Micropython so genannte Sockets zur Verfügung, diese bieten entsprechende Werkzeuge an:

Ein Socket ist eine (Software-) Schnittstelle (Interface) zwischen einem Programm und den vom Betriebssystem (hier: Firmware des ESP32) angebotenen Kommunikationsprotokollen (bei uns TCP).

TCP benutzt beim Transport sogenannte IP-Adressen; eine solche **IP-Adresse** (IP = Internet Protocol) besteht aus vier Bytes (Streng genommen gibt es verschiedene IP-Adress-Arten; wir benutzen hier nur **IPv4**). Die IP-Adresse des Servers meiner Website ist z. Zt. 85.13.133.204. Da wir uns solche Nummernfolgen nur schwer merken können, benutzen wir statt dessen meist einen so genannten **Domain-Namen**. Bei meiner Website lautet er "www.g-heinrichs.de". Im Internet gibt es spezielle Server, die zu einem Domain-Namen die zugehörige IP-Adresse liefern können. Sie werden **DNS-Server** (Domain-Name-System-Server) genannt. Beim Transport der Daten arbeitet das Internet letztlich immer mit den IP-Adressen. Wenn ich also einen Domain-Namen bei meinem Browser eingebe, wird zuerst mit Hilfe eines solchen DNS-Servers die passende IP-Adresse ermittelt; nur mit dieser IP-Adresse können Informationen durch das Internet vom Client zum Server gelangen. Entsprechendes gilt für den umgekehrten Weg vom Server zum Client.

Ein Server besitzt nun i. A. verschiedene Ports. Diese entsprechen den Abteilungen bei unserer Firma **Elektronik GmbH**. Jeder **Port** ist – ähnlich wie den Abteilungen unserer Firma – für eine bestimmte Art von Dienstleistung zuständig, z. B. dem Versenden von Webseiten. Jeder Port wird durch eine Nummer gekennzeichnet, der so genannten **Portnummer**. Nach einer internationalen Vereinbarung erfolgt die Nachfrage nach Webseiten immer über den Port mit der Nummer 80. Allgemein können Portnummern Werte zwischen 0 und $2^{16}-1 = 65535$ besitzen. Die Kombination aus der IP-Adresse und der Portnummer beschreiben den Ort und die Art der Dienstleistung. Über ein- und dieselbe IP-Adresse können damit unterschiedliche Dienste zur Verfügung gestellt werden.

Damit zwei Rechner per TCP korrekt kommunizieren können, muss jeder der beiden Rechner (mindestens) eine IP-Adresse und (mindestens) eine Portnummer vorweisen. Für die Kombination aus IP-Adresse und Portnummer benutzt man häufig die abkürzende Schreibweise 85.13.133.204:80.

Die Ports von 0 bis 1023 sind für bestimmte Dienste fest vergeben und werden weltweit geregelt. Die Ports von 1024 bis 49152 werden **Registered Ports** genannt und sind für bestimmte Anwendungen reserviert. Die restlichen Ports stehen zur freien Verfügung.

Dienst	Portnummer
Daytime	13
FTP	20, 21
Telnet	23
SMTP	25
HTTP	80

Ein Blick hinter die Kulissen: Wer versucht, über die oben angegebene IP meine Website zu erreichen, wird merken, dass das nicht gelingt. Benutzt man hingegen den Domain-Namen, gibt es keine Probleme. Woran liegt das? Nun, da die IP-Nummern (zumindest bei IPv4) knapp sind, benutzen die Anbieter von Webspace einen Trick: Auf einem einzigen Server (mit einer festen IP-Adresse) befinden sich mehrere Websites, eine jede davon mit einem eigenen Domain-Namen. Eine Anfrage vom Browser wird bei all diesen Domain-Namen über die gemeinsame IP zu demselben (richtigen) Server gelangen. Wie wir in dem Kapitel über HTTP noch sehen werden, beinhaltet die Anfrage des Browsers nicht nur den Namen der gewünschten Datei, sondern immer auch den Domain-Namen; mit diesem wird nun dafür gesorgt, dass die Anfrage auf dem Server zu der gewünschten Website geleitet wird. Allein über die IP-Adresse, also ohne Domain-Name, ist eine korrekte Weiterleitung nicht möglich; man erhält in diesem Fall eine Fehlermeldung.

Wenn wir eine Anfrage an einen Server außerhalb unseres Heimnetzes richten, sollte man also lieber den Domain-Namen und nicht die IP-Adresse benutzen (wenn man sie denn kennt).