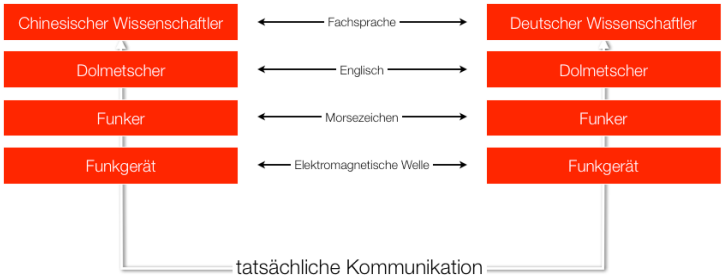
Modul: Grundlagen digitaler Medientechnik

Die Dienste des Internet

Paul F. Siegert

1. Layering

Mit dem Aufbau des ARPAET waren eine Reihe technischer Probleme zu lösen die unmöglich von einer Forschergruppe allein bewerkstelligt werden konnten. Zwischen diesen Gruppen gab es Auseinandersetzungen. Um die Konflikte nicht eskalieren zu lassen, ging man den Weg einer strikten Arbeitsteilung, wodurch gleichzeitig die Komplexität der Softwareentwicklung erheblich reduziert werden konnte. Man beschloss, die Software in eigenständige, hierarchisch gestaffelte Schichten (Layer) zu unterteilen, die jeweils durch klar definierte Schnittstellen aufeinander aufbauten. Somit ist das Layering sowohl ein technische, wie soziale Erfindung.



Die Methode des Layering wurde schnell zu einem allgemeinen Denkmodell in der Netzwerktechnik. Die hierarchisch angeordneten Layer kapselten klar definierte Problemkreise genau ein und beschrieben deren Funktionalität. Schnittstellen zwischen diesen Ebenen regelten deren Zusammenwirken. Jede Ebene benutzte die Ressourcen der unter ihr liegenden und generierte Daten für die über ihr liegende Schicht. Daraus ergaben sich zwei große Vorteile:

Durch das Layering gliederte sich das komplexe technische Problem in einzelne Teilaufgaben. Sobald die Entwickler sich über die Schnittstelle geeinigt hatten, konnten die Komponenten getrennt und dezentral entwickelt werden. Gleichzeitig wurden die Schichten jederzeit austauschbar, was bei einer Weiterentwicklung der Software große Vorteile brachte.

Protokolle

Bei einem Kommunikationsaufbau zwischen zwei Rechnern kommuniziert jede Layerschicht mit ihrem Pendant auf der anderen Seite über ein so genanntes Protokoll. Ein Protokoll ist also eine technische Spezifikation, wie innerhalb eines bestimmten Layers kommuniziert wird. Der Protokollstapel des ARPANET sieht folgendermassen aus:

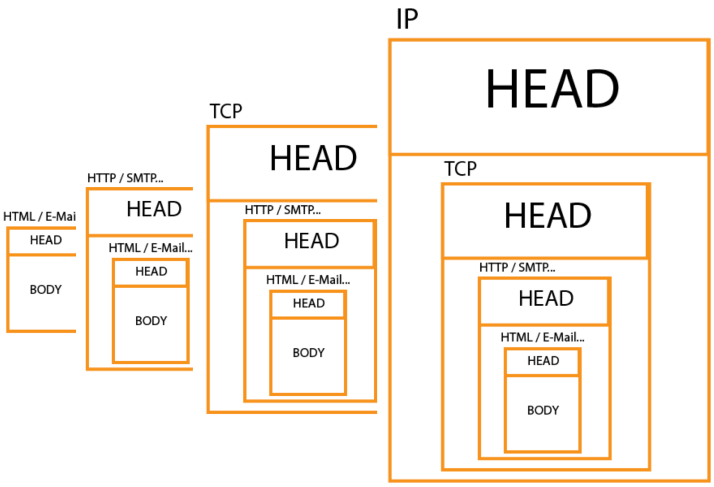
Anwendungsschicht	Anwendungsschicht	HTTP / FTP / SMTP / POP / ...
Darstellungsschicht		
Sitzungsschicht		
Transportschicht	Transportschicht	TCP / UDP
Vermittlungsschicht	Internetschicht	IP
Sicherungsschicht	Netzzugangsschicht	Ethernet / FDDI / ...
Bitübertragungsschicht		

Die ersten Protokolle auf der Anwendungsschicht, die für das ARPANET entwickelt wurden waren Telnet (Fernbedienung von Rechnern) und FTP (Übertragung von Dateien).

2. Ports

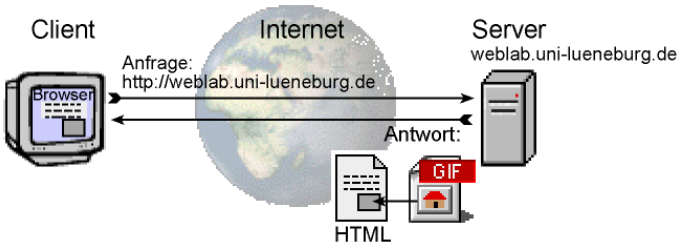
Wenn nun das IP-Protokoll dafür sorgt, dass die Datenpakete auf dem richtigen Rechner ankommen, kann es sein, das Pakete zum Beispiel einer Internetradio-Übertragung und eines WWW-Seitenaufrufs gleichzeitig eintreffen. Daher ist jedem Dienst ein so genannter Port zugeordnet, der im Header der Transportschicht (TCP) steht und anhand dessen die Pakete ihrer weiteren Verwendung zugeführt werden können. Gängige Ports sind z.B.:

- 20 und 21 für FTP
- 80 für HTTP
- 25 für SMTP



Die TCP-Pakete transportieren in ihrem Body Daten, deren Form durch die Protokolle für die einzelnen Dienste geregelt ist.

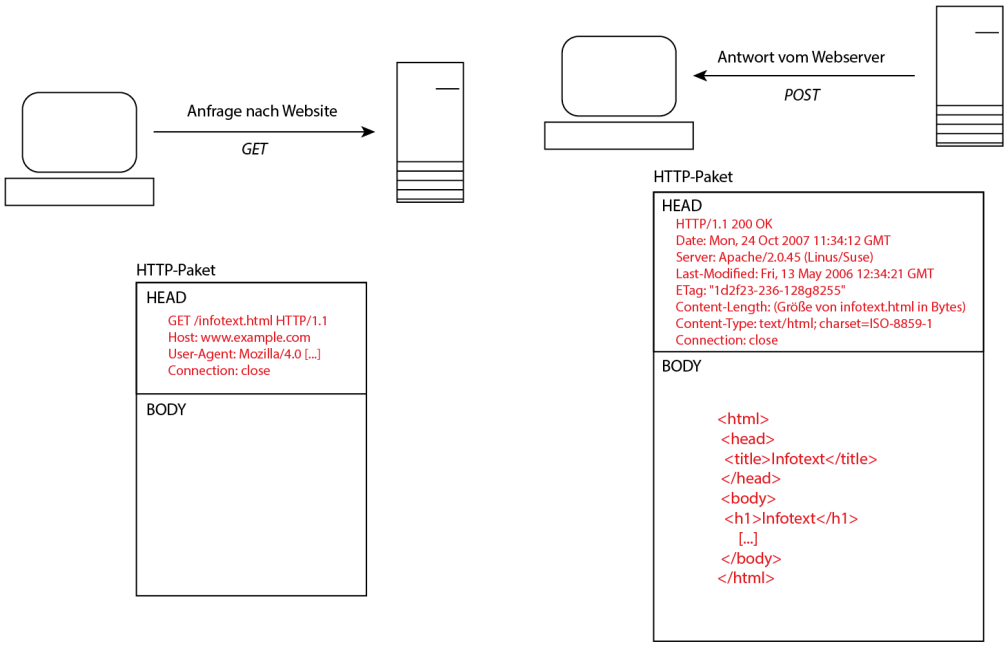
3. WWW



HTTP

Gehen wir einmal davon aus, das ein Nutzer eine WWW-Seite aufruft. Auf der Anwendungsschicht ist dafür das HTTP (Hypertext Transfer Protocol) zuständig, welches ja auch immer zu Beginn einer URL im Browser genannt wird.

Die per IP-Protokoll angekommenen Datenpakete werden durch TCP zu einer Nachricht zusammengesetzt und über den Port 80 an den HTTP-Server – also den Webserver – weitergeleitet. Dieser nutzt nun HTTP um sich mit dem anfragenden Browser zu verständigen. HTTP regelt welche Webseite aufgerufen werden soll bzw. gerade gesendet wird. Eine HTTP-Kommunikation könnte wie folgt aussehen (siehe Bild).



HTML

HTTP regelt also welches Dokument angefragt bzw. gesendet wird. Im Beispiel oben wurde die Seite „infotext.html“ vom Server „www.example.com“ angefragt und erfolgreich („200 ok“) zurückgegeben. Das was das HTTP-Paket im Body an den Browser übermittelt ist ein HTML-Dokument. HTML ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks, die vom Browser interpretiert werden muss. Auch das HTML-Dokument besteht wieder aus Head und Body. Im Header befinden sich Metainformationen für den Browser wie z.B. der verwendete Zeichensatz. Im Body befindet sich der Text, der auf der Webseite erscheinen soll.

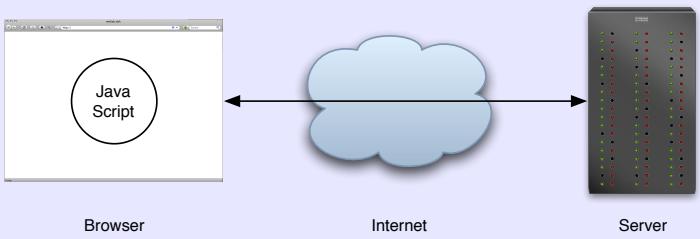
Damit ergibt sich eine Box-in der-Box Architektur, bei der sich im Body der einen Layerschicht immer das ganze Paket für die nächste befindet.

Dynamische Webseiten

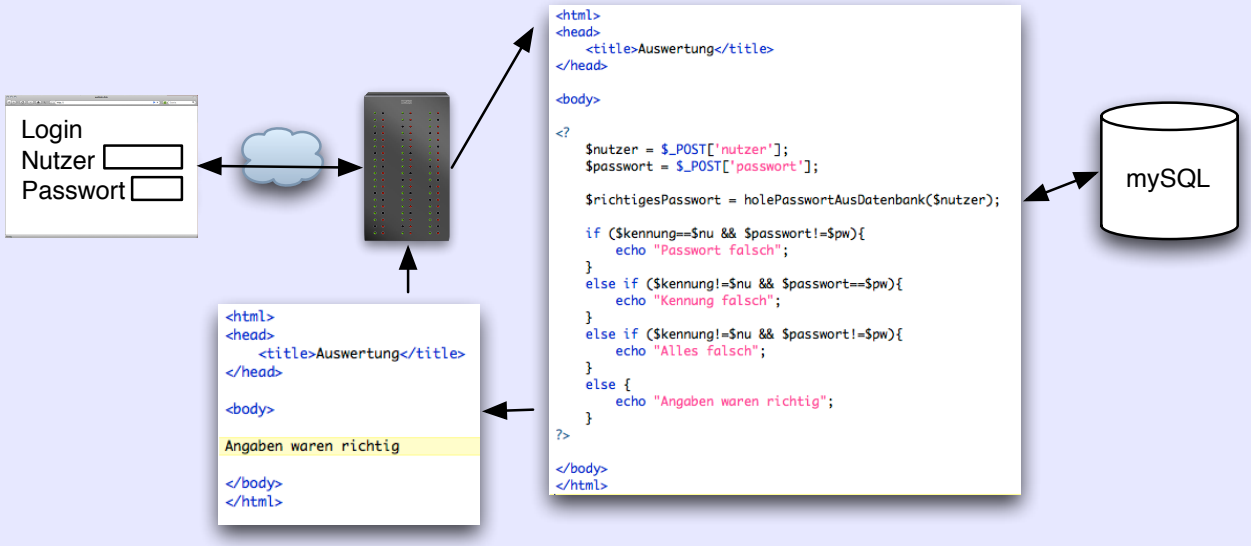
HTML ist eine statische Beschreibung der Struktur einer Webseite. Die Seite wird so angezeigt, wie sie einmal entworfen und auf dem Server abgelegt worden ist. Reine HTML-Seiten sind nicht in der Lage auf bestimmte Situationen zu reagieren bzw. ihr Aussehen dynamisch zu verändern. Um Commuities, Shopping- oder Informations-Systeme zu bauen, wird Programmfunktionalität und oft auch die Möglichkeit zur Speicherung von Daten benötigt. Diese Programmfunktionalität lässt sich an zwei Stellen des Prozesses einbinden: Auf der Seite des Clients oder auf der Seite des Servers.

- clientseitig

Als clientseitige Programmiersprachen eignen sich JavaScript, Java Applets oder Flash. Der Programmcode wird dabei in die Webseite eingebettet und mit ihr übertragen. Entsprechend läuft das Programm dann auf dem Rechner desjenigen, der die Seite aufgerufen hat. Das entlastet zwar den Server von der Rechenlast und das Netz von häufigen Datenübertragungen, kann aber Sicherheitsprobleme geben, da – wie bei JavaScript – der Code im Klartext zu lesen ist.



- serverseitig



Liegt die Programmfunktionalität auf Seiten des Servers, so ruft der Nutzer zunächst eine Webseite über seinen Browser auf. Der Server, der diese Seite ausliefern soll erkennt anhand der Dateiendung (z.B.: .php oder .pl), dass es sich um eine Programmdatei handelt. Daher wird die Datei nicht direkt zurückgeschickt, sondern an eine Instanz (den Interpreter) weitergeleitet, die das Programm das in der Datei eingebettet ist abarbeitet. Beim Abar-

beiten des Programmcodes kann es sein das es Kontakt zu einer Datenbank aufnimmt, um Daten zu suchen, zu speichern o.ä. Hat z.B. der Nutzer ein Login-Formular ausgefüllt und an den Server geschickt, sucht das Programm in der Datenbank das Passwort für den angegebenen Nutzernamen und vergleicht es mit dem übermittelten Passwort. Stimmen beide überein, kann das Programm fortfahren. Andernfalls erstellt es eine HTML-Seite mit einer Fehlermeldung – z.B. „Die angegebenen Daten waren nicht korrekt. Versuchen Sie es erneut.“ – und schickt diese an den Client zurück.

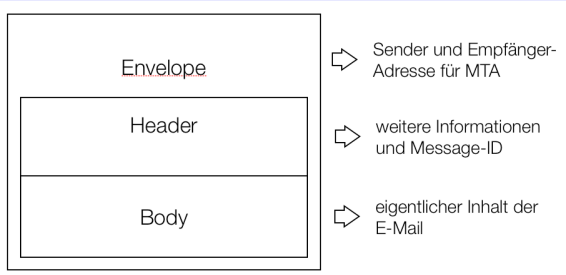
Die Programme auf der Serverseite können sehr komplex werden. Der Nutzer bekommt aber immer nur das Ergebnis, eine reine HTML-Seite, zurück.

4. E-Mail

SMTP (Simple Mail Transfer Protocol)

Eine E-Mail besteht aus einem lesbaren Teil, der sich wiederum in den Head (mit Feldern wie to, from, subject etc.) und den Body mit der eigentlichen Nachricht gliedert. Die Header-Informationen können genutzt werden, um die E-Mail-Programme sehr komfortabel zu gestalten.

Diese Header-Angaben sind jedoch lediglich für die E-Mail-Programme gedacht. Technisch, zum Verschicken der E-Mail, sind sie irrelevant. Die Mail würde auch ankommen, wenn irgendetwas Beliebiges in den Header-Felden (incl. dem To:-Feld) steht.



Return-Path: < siegert@uni.leuphana.de>
Received: from [193.174.43.177] (account siegert HELO JURWS) by leuphana.de (CommuniGate Pro SMTP 5.1.16) with ESMTPA id 16223123; Mon, 06 Oct 2008 13:02:08 +0200
From: = "'Paul F. Siegert'" <siegert@uni.leuphana.de>
To: "'Uwe Mylatz'" < mylatz@uni.leuphana.de>
Cc: "'Max Mustermann'" <mustermann.max@uni.leuphana.de>
Subject: myStudy
Date: Mon, 6 Oct 2008 13:02:12 +0200
Message-ID: <003c01c927a2$fd4d5a20$f7e80e60$@leuphana.de>
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
X-Mailer: Microsoft Office Outlook 12.0
Content-Language: de

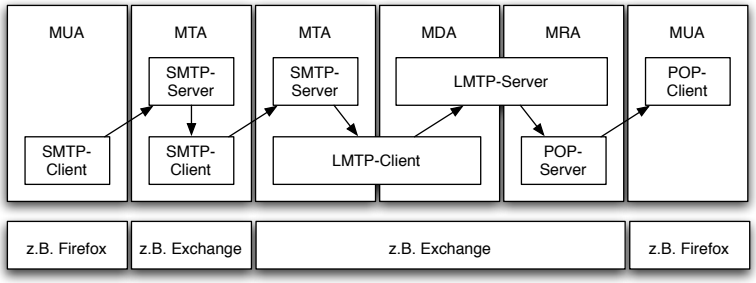
Lieber Uwe,
ich habe heute leider keine Zeit.
Gruß,
Paul.

Um die Mail zu versenden, wird sie erneut in einen sogenannten SMTP-Envelope eingetütet. Dieser Umschlag wird, wenn die Mail angekommen ist, gelöscht. Aber es sind nur diese Angaben, die für den Transport der Mail wichtig sind. Und das ist der Grund, weshalb die Herkunft von Spam so schwer nachzuvollziehen ist. Die sichtbaren Header-Felder lassen sich beliebig manipulieren, da sie für den Transport nicht verwendet werden.

Ein typischer Verbindungsaufbau mit dem Mailserver könnte folgendermassen aussehen:

Client	Server	Erklärung
HEAD (SMTP-Envelope)		
	220 mail.leuphana.de SMTP Leuphana Mailserver	Direkt nach dem Verbindungsaufbau über TCP meldet sich der SMTP-Server.
HELO kdga.leuphana.de		Der SMTP-Client meldet sich mit seinem Computernamen an. Diese Angabe wird nie auf ihre Richtigkeit geprüft!
	250 Ok	Der Server bestätigt den Erhalt und erwartet die Fortführung der Verbindung.
MAIL FROM: siegert@leuphana.de		Der Client meldet die Absender-Adresse für den MTA. Diese Angabe wird nie auf ihre Richtigkeit geprüft!
	250 Ok	Der Server bestätigt den Erhalt und erwartet die Fortführung der Verbindung.
RCPT TO: mylatz@leuphana.de		Der Client meldet die Empfänger-Adresse für den MTA.
	250 Ok	Der Server bestätigt den Erhalt und erwartet die Fortführung der Verbindung.
BODY		
DATA		Mit DATA leitet der Client das Senden der E-Mail ein.
	354 End data with .	Der Server teilt dem Client mit, dass die E-Mail mit einem Punkt (.) abgeschlossen werden soll.
From: siegert@leuphana.de		1. E-Mail-Zeile
To: muster@leuphana.de		2. E-Mail-Zeile
Subject: Testmail		3. E-Mail-Zeile
		Der Client signalisiert mit zweimal Zeilenumbruch den eigentlichen Nachrichtentext der E-Mail in der 4. E-Mail-Zeile.
Lieber Uwe,		5. E-Mail-Zeile
ich habe heute leider keine Zeit.		6. E-Mail-Zeile
Gruß, Paul		7. E-Mail-Zeile
.		Die E-Mail wird mit dem Punkt (.) abgeschlossen.
	250 Ok	Der Server quittiert den erfolgreichen Empfang der E-Mail.
QUIT		Der Client beendet die Verbindung zum Server.
	221 Byte	Der Server sagt "Auf Wiedersehen".

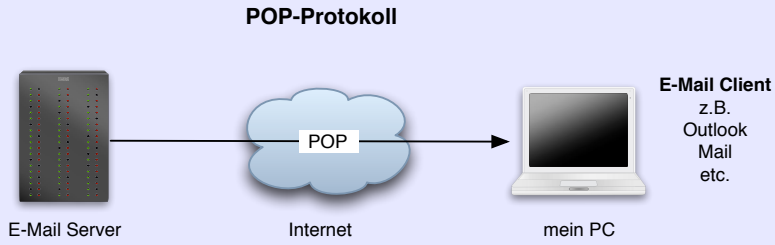
Der Mail User Agent (MUA; z.B. Outlook, Mail etc.) schickt die Mail an seinen Mail-Server, den Mail Transfer Agent (MTA). Dieser regelt nun über SMTP das Versenden der E-Mail an andere Mail-Server.



An der Zielinstitution angekommen, kann ein Mail Delivery Agent (MDA) die E-Mail lokal innerhalb des eigenen (Firmen-)Netzes über das Local Mail Transfer Protocol (LMTP) weiterleiten bis die Nachricht am Mail Retrieval Agent (MRA) angekommen ist. Von dort aus kann der Nutzer (am MUA) über die Protokolle POP oder IMAP seine Mail abrufen. Daher müssen in den E-Mail Clients für das Versenden (SMTP) und für das Abrufen (POP oder IMAP) unterschiedliche Angaben gemacht werden.

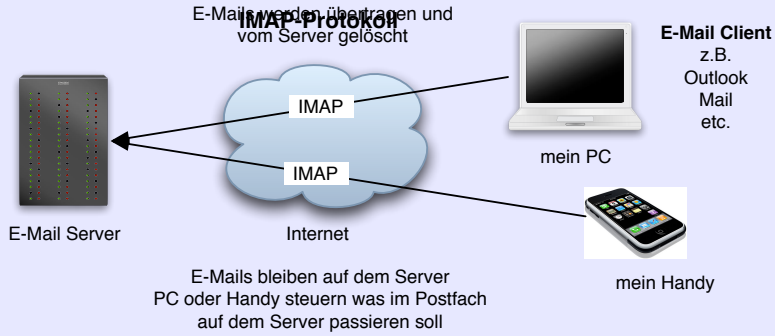
POP

Mit dem Post Office Protocol (POP) werden die E-Mails auf den Rechner des Nutzers geladen und dann auf dem Server gelöscht.



IMAP

Das Internet Message Access Protocol (IMAP) dagegen verwaltet die E-Mails auf dem Server. Die Geräte greifen quasi als Fernbedienung darauf zu. Der Vorteil ist dabei, das man so mit verschiedenen Clients (E-Mail Programm, Web-Anwendung, Smartphone etc.) auf sein E-Mail Postfach zugreifen kann, ohne auf den verschiedenen Geräten inkonsistente Bestände seine E-Mail-Kommunikation zu erzeugen.



Webmail

Webmail ist kein Protokoll für die E-Mail Kommunikation, sondern nur eine bestimmte Art des technischen Aufbaus. Zwischen User und MUA wird ein Webserver als Oberfläche dazwischengeschaltet. Dieser Webserver greift mittels IMAP auf den E-Mail-Server zu. Der User kommuniziert aber lediglich (über HTTP natürlich) mit einer dynamischen Website, die serverseitig das E-Mail-Konto als HTML-Seite darstellt.