1 Das Web, das unbekannte Wesen

typischer Bereich	Begriff	
100m	Local Area Network: **LAN**	Gebäude, Campus
10 km	Metropolitan Area Network: **MAN**	Stadt
1.000 km	Wide Area Network: **WAN**	Land
10.000 km	Intenret (Global Area Network: GAN)	die Erde – und mehr?

Internet-Protokolle

- ip: Internet Protocol (IPv4 und IPv6)
- tcp: Transmission Control Protocol
- DNS: Domain Name Service
- FTP: File Transfer Protocol (Port 21)
- SMTP: Simple Mail Transfer Protocol
- POP: Post Office Protocol
- IMAP: Internet Message Access Protocol

Was ist ein Web-Server?

- Software, die permanent läuft
- wartet auf tcp-Netzwerkport 80 auf Anfragen und beantwortet diese
- schreibt Protokolldateien
- ist konfigurierbar und sicher

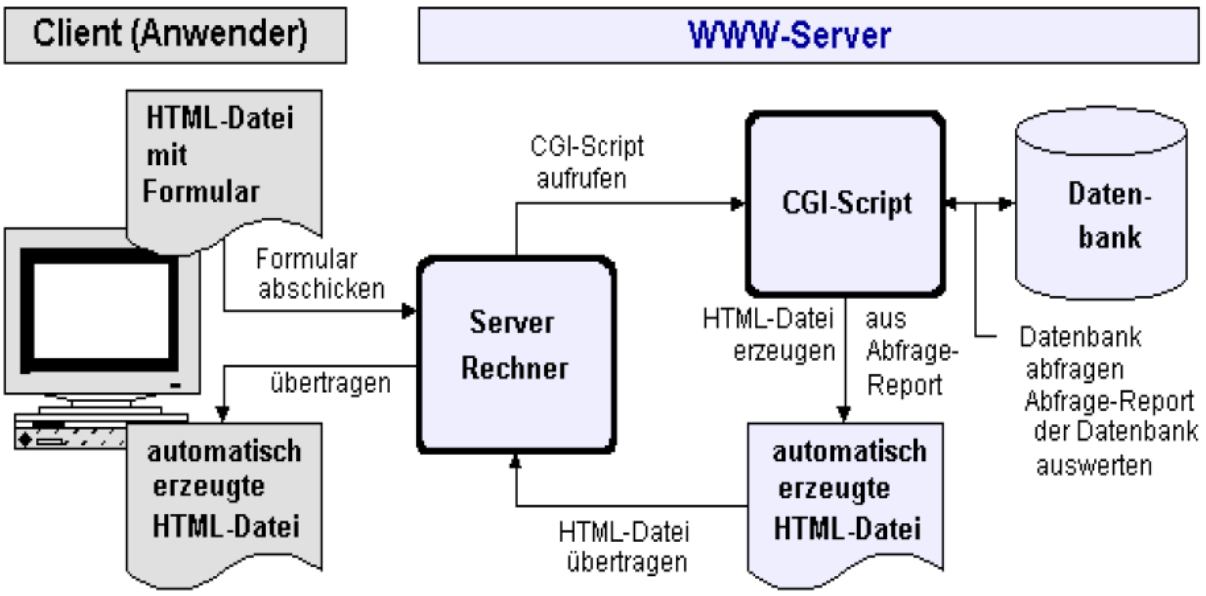
Kommunikation zwischen Client und Server

- GET (Anfordern), POST (wie GET, aber seperates IO), HEAD (Header-Informationen), PUT (Upload), TRACE (Proxys Ausweisen), DELETE (Entfernet auf dem Server), OPTIONS (mögliche http-Anweisungen), CONNECT (Proxy)
- Simulation über telnet <server> 80
- Antwort besteht aus Antwort-Code, Header-Infos, Dokument in HTML-Formatierung
- Server-Antwortcodes →

200	ok	400	bad request
201	created	402	unauthorized
202	accept	403	forbidden
204	no content	404	not found
300	multiple choices	500	internal server error
301	moved permanently	501	not implemented
302	moved temporairly	502	bad gateway
304	not modified	503	service unavailable

Dynamik im Web

- Dynamik beim Client (JavaScript, Flash, …)
- Dynamik beim Server (CGI (mit Perl), PHP)



2 Auszeichnungssprachen für das Web

HTML – Hyper Text Markup Language

- Erste Zeile: Dokumententyp (strict, transitional, frameset)
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
- Header-Tags
  - <link type=”text/css” rel=“stylesheet“ href=“style.css“ media=”screen”>
  - <meta name=”author” content=”Max Mustermann”>
- HTML Validierung mit validator.w3.org
- CSS (Cascading Stylesheets) zur Trennung von Inhalt und Layout
- CSS-Platzierungsvarianten: Im Tag, Im Header, Im ausgelagerten Stylesheet

XML – Extensibale Markup Language

- Metasprache zur Datenstrukturierung → Baumstruktur
- jedes Tag wird geschlossen, case sensitiv, beliebig erweiterbar
- Grundsyntax: <?xml version=“1.0“ encoding=“UTF-8“>
- XML-Dokumente sind
  - wohlgeformt (weel-formed), wenn formal korrekt
  - gültig (valid), wenn sie einer vorgegebenen Struktur genügen (Doctype Definition)

3 Clientseitige Web-Programmierung mit JavaScript

Diskussion clientseitige Web-Programmierung

- **Vorteile:**
  - kein Netzwerk → schnelle Response
  - aktive Interaktion mit dem User („Maus-Effekte“)
  - Performance durch Nutzung der Client-CPU
- **Nachteile:**
  - Security auf dem Client
  - keine Kontrolle der Software auf dem Server
  - de facto keine Datenbank-Anbindung möglich

JavaScript

- nicht typisiert (dynamisch typisiert), wird interpretiert, objektorientiert ohne Klassen
- JavaScript kann vom Benutzer verboten werden
- <script type=“text/javascript“> </script>
- Gegenstück: <noscript>Warnung kein Javascript aktiviert</noscript>
- Typabfrage über typeof „Variablenbezeichner“
- Arrays sind Listen, numerischer Index genauso wie assoziatives Array
  - var liste = new array(); liste[0] = 42; liste[‘tu’] = “WSI”;
- besondere Methoden: alert(arg), parseInt(arg), parseFloat(arg), isNaN(arg) (not a Number)
- <SCRIPT language="JavaScript" src="file.js">

4 JavaScript mit Document Object Model

Document Object Model

- Objekte für die Interaktion JavaScript <-> HTML
- Drei Wurzeln: window (aktuelles Browserfenster), screen (Bildschirm), navigator (Browser)

Das navigator-Objekt

- zur Verwaltung des Browser
- Attribute/Methoden →

product: Gecko
vendor: Google Inc.
plugins: [object PluginArray]
vendorSub:
language: de
userAgent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/532.5 (KHTML, like Gecko) Chrome/4.1.249.1064 Safari/532.5
mimeTypes: [object MimeTypeArray]
productSub: 20030107
appVersion: 5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/532.5 (KHTML, like Gecko) Chrome/4.1.249.1064 Safari/532.5
appCodeName: Mozilla
cookieEnabled: true
platform: Win32
appName: Netscape
onLine: true
javaEnabled: function javaEnabled() { [native code] }
getStorageUpdates: function getStorageUpdates() { [native code] }
registerProtocolHandler: function registerProtocolHandler() { [native code] }
registerContentHandler: function registerContentHandler() { [native code] }
javaEnabled(): true

Das screen-Objekt

- Informationen zum Bildschirm
- Attribute:
  - availabeHeight
  - availabeWidth
  - colorDepth

Das window-Objekt

- wichtige Untergliederungen: document, frames, event, history, location
- window ist Standard, kann somit meistens weggelassen werden
- **location**: protocol, hostname, port, pathname, href, host, reload()
- **history**: back(), forward(), go(n)
- document: write(…), title, lastModified, bgColor und fgColor

Event-Behandlung mit JavaScript

- typische Events: Mausklick, “Mouseover”, Seite neu laden, Fenster schließen
- Möglichkeiten: direkter JavaScript-Link oder Event-Handle (z.B. mit onFocus)

Weitere Client-Techniken

- Java Applet (echtes Java auf dem Client)
- Adobe (Macromedia) Flash
- MS Silverlight (beschränkt auf Windows)

5 Serverseitige Web-Programmierung mit CGI – Teil 1

Common Gateway Interface (CGI)

- Möglichkeit, um im WWW serverseitig Programme bereitzustellen, die von „HTML-Seiten gestartet werden“ und HTML-Code produzieren
- **Vorraussetzungen:**
  - Webserver der CGI unterstützt
  - das Modul mod_cgi für den Apache-Webserver (default)
  - entsprechende Konfiguration in httpd.conf
- Datenübergabe mittels GET und POST
- **Umgebungsvariablen**: SERVER_NAME, SERVER_PROTOCOL, SERVER_PORT, PATH_INFO, SCRIPT_NAME, QUERY_NAME, REMOTE_HOST, REMOTE_ADDR, REMOTE_USER
- in printenc.pl stehen alle Umgebungsvariablen
- cgi-Programm muss ausführbar sein und im Unterordner cgi-bin liegen
  Ausgabe muss mit: Content-type: text/html\n\n beginnen
- Alle Ausführbaren/compilierten Sprachen möglich (Java nicht, weil es nur auf VM läuft)

Practical Extraction and Report Language (PERL)

- **Syntax**: #Kommentar, print(“blub”), $variable, Strings mit . verknüpfen
- **Typen**: integer, float, string
- Eingabe-Abfragen: $a = <STDIN>; chomp($a)
- Existenz einer Variable prüfen: defined($b)
- **Liste**: @beispiel = („eins“, „zwei“, „drei“);
  push(@lis, $d); $d = pop(@lis);
  unshift(@lis, $d); $d = shift(@lis);
- **Hash**: %ha = („ich“, „Koni“, „du“, „werAnders“)
  reverse %gruppe: vertauscht key-value Paare
  each(%hash) lieft Liste mit key-value-Paaren
  keys(%hash): Liste mit den n key’s eines Hash; values(%hash): Liste mit den n value’s

Typ	Zeichen	Beispiel	Beschreibng
Skalar	`$`	`$pi`	skalare Variable
Array/Liste	`@`	`@feld`	indizierte Variable
Hash	`%`	`%keyvalue`	assoziatives Array
Unterpro-gramm	`&`	`&sub`	Perlcode

- **Vergleichsoperatoren**: == numerischer Vergleich, eq String-Vergleich, lt (alphabetisch <, le <=)
- **Schleifen**: mit last; vorzeitiger Abbruch
- **Dateizugriff**:
  open(H, „<$file“); erstellt Filehandle H für Datei (zum Lesen)
  $zeile = <H>; liest eine Zeile von H (wie Tastatureingabe)
  open(H, „>$file“); öffnet Datei zum Schreiben
  open(H, „>>$file“); öffnet Datei zum Anfügen
- **Methoden**: sub name { } Wichtig ohne Übergabelisten, erster Parameter ist $_[0] etc.
- **Perl-Code importieren**: do(<filename>); use (importiert Modul)
- **Erweitertes Print**: print <<ENDE\nContent-type: text/html\n\n “text“ ENDE \n;
- **Time**: $zeit = gmtime(time);
  $zeit[0] (sek), $zeit[1] (min), $zeit[2] (h), $zeit[3] (days), $zeit[4]+1 (mon), $zeit[5]+1900 (year)

6 Serverseitige Web-Programmierung mit CGI

Shebang-Zeile

- #!<path_to_perl/perl –w
- Sagt dem System wo der Interpreter für die Sprache zu finden ist um sie auszuführen

Modul: CGI

- Funktionalität: Formularverarbeitung, Generierung von XHTML-Code, Nützliches für CGI
- im Perl-Script:
  use CGI; # Import des CGI-Moduls
  $cgi = new CGI;
  $wert = $cgi->param(“feld”); bzw. $formular = $cgi->Vars(); $wert = $formular->{“feld”};
- Über CGI last sich auch einfach auf Umgebungsvariablen zugreifen ($cgi->server_name()

Andere Module

- Datenbankverwaltung mit Modul DBI
- HTTP-Clients zu programmieren LWP::Simple oder LWP::UserAgent
  use LWP::Simple; $doc = get(“http://google.de); (würde google.de abrufen) etc.
- strict erzwingt einen besseren Programmierstiel (z.B. Variablen müssen deklariert werden)

7 Einführung in PHP

PHP: Hypertext Proprocessor

- zentrale php.ini Konfigurationsdatei
  <?php phpinfo(); ?> gibt alles wichtige zur Konfiguration gut formatiert aus
- **Konstanten**: define(„NAME“, „blub“);
- **Umgebungsvariablen**: $_COOKIE, $_ENC, $_SESSION, $_SERVER, $_GET[], $_REQUEST[]
- **Array-Funktionen**: unset($array), count($a), assort($a) (aufsteigend), rsort($a) (absteigend), ksort($a) (nach key), each($array) (um ein Array durchzulaufen)
- **Dateizugriff**:
  $fh = fopen($datei, $mode); (mögliche Modes: r lesen; r+ lesen und schreiben; a anfügen)
  $string = fgets($fh, $length); (liest Zeile, bzw. maximal length Zeichen)
  $i = fputs($fh, $string); schreibt Zeile
  $i = fclose($fh); schließt Zeile
- **Funktionsdeklaration**: function who($wer, $job = „HARTZ4“) { } (default setzen)
- **Referenzen**: $a = &$b; Inhalt wird nicht kopiert sondern Zeigt auf das selbe

ObjektOrientierte Programmierung in PHP

- Definition der Klasse über Schlüsselwort class
- Konstruktor: Methode __construct(…) { … }
- Destruktor: Methode __destruct(…) { … }
- Methode __toString() { … }
- Datenkapselung: public, protected, private
- Instanziierung: new
- klassische Vererbung (class B extends A), Zugriff auf Elternklasse über parent (vgl. super Java)
- weiteres: abstract, final, interface, try, catch, finally, static

8 Datenbankapplikationen mit PHP

DatenbankManagementsystem (DBMS)

- auf dem jeweiligen DBMS dann die Datenbank & regelt „konkurrente Zugriffe“ auf die DB
- bietet eine direkte Abfragesprache, um die DB anzusprechen
- ist in einer Dreischichtarchitektur realisiert
- Beispiele: Oracle, Informix, DB2, MySQL, …
- DBMS helfen bei Inkonsistenz, Redundanz, Mehrbenutzerzugriff, Regelung von Zugriffsrechten und Datensicherheit, Anlegen und Abfragen von DBs

Der mysql

- meistverbreitetes DBMS im Web
- **Syntax**: mysql –u user –p <datenbank>, show tables;
- **Structured Query Language** (SQL)
    - CREATE TABLE student (mtknr INT AUTO_INCREMENT PRIMARY KEY,
      name VARCHAR(64) NOT NULL, NOT NULL, fach VARCHAR(32) NOT NULL);
      INSERT INTO student (name, fach) VALUES (“Max”, “Informatik”);
    - SELECT COUNT(*) zählt die Anzahl der Treffereilen
      SELECT SUM (spalte) summiert Werte in der Spalte spalte
      SELECT AVG(spalte) arith. Mittel über Spalte
      SELECT MIN(spalte) Minimum der Spalte
      SELECT MAX(spalte) Maximum der Spalte
- **Verbindungsaufbau**:
  $channel = mysql_connect($host, $user, $passwd); (0 bei fehler)
  mysql_erno(); (gibt den letzten Fehlercode zurück)
  mysql_error(); (die zugehörige Beschreibung)
  mysql_select_db($dbname[,$connenctID]);
  mysql_query($query[,$id]); (entfällt als Argument die connect-ID $id, wird die zuletzt aktive Verbindung genommen (Rückgabewert: int > 0 bei Erfolg)
  mysql_affected_rows([$connectID]); (liefert die veränderten Datensätze zur Verb. $connectID
  $zeile = mysql_fetch_row($queryID); (Array einer Trefferzeile)
  mysql_fetch_array($queryID); (liefert auch assoziatives Array)
  mysql_result erlaubt direkten Zugriff auf eine Trefferzeile
  mysql_num_rows($queryID); (Anzahl der selektierten Datensätze)
  mysql_free_result($queryID); (setzt Speicher frei)
  mysql_close($connectID);
  $id = mysql_query($query, $connectID);

9 Cookies & Sessions & eKaay & AJAX

Individualisierung im Web

- wesentliche Eigenschaft, die Website kennt keine Vorgeschichte
- trivialer Ansatz: hidden fields
- Cookies und Sessions zur Personalisierung von Websites
- Cookie ist ein key-value-Paar und hat begrenzte Haltbarkeit
- PHP-Methode: setcookie(…);
- **Sessions**: session_start(), $_SESSION, session_destory()
- Sessions ohne Cookies: session->url() bzw. session->hidden()
- Session-Informationen mit Perl über CGI::Session::File
- **eKaay**: Scannen des Barcodes mit eKaay-App, übertragen von Benutzername, SessionID und berechnetem Kennwort an Account-Server, dieser öffnet den Account
- **AJAX**: Browser stellt JavaScript-gesteuert kontinuierlich weitere Anfragen an den Server
    - Drei Schritte:
      Erzeugung des XMLHttpRequest-Objekts: var http = new XMLHttpRequest();
      Methode open aus dieser Klasse (http-Methode (GET/POST))
      Anzeige des Ergebnisses über JavaScript-Eigenschaft onreadystatechange