

**Veranstaltung
10033013**

Systementwicklung

HTTP, Apache und PHP

**Uwe H. Suhl und Chris Bizer
SS 2008**

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Gliederung

- 1. Architektur Web-basierter Anwendungssysteme**
 - Logisches 3-Schichten-Modell
- 2. HTTP**
 - Requests und Responses
- 3. Apache**
 - Installation und Konfiguration
 - Logging und Zugriffsschutz
- 4. PHP 1**
 - Grundsätzliche Sprachkonstrukte
 - Übergabe von Parametern zwischen Skripten
 - Serverseitige Auswertung von Formulardaten
 - Seitengenerierung mittels Require-Befehlen

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

1. Architektur Web-basierter Anwendungssysteme

- Anwendungssysteme lassen sich in drei Schichten unterteilen:



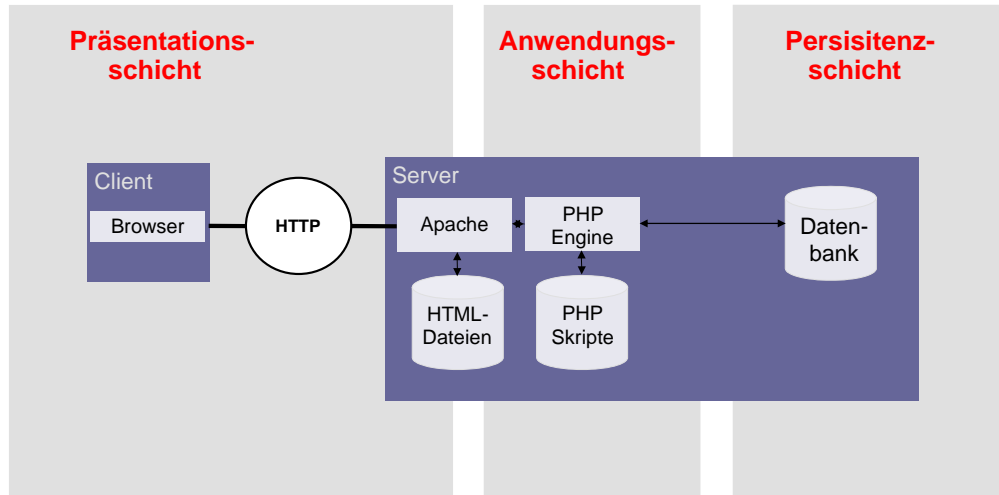
Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Das logische 3-Schichten-Modell

- **Aufgaben der Präsentationsschicht**
 - Darstellung und Erfassung von Daten
 - Interaktion mit dem Nutzer
 - Derzeit setzen sich Webbrowser als universelle Präsentationsumgebung durch
- **Aufgaben der Anwendungsschicht**
 - Geschäfts- und Applikationslogik
 - Transformation und Weiterverarbeitung von Daten
 - Anstoßen von Prozessen in anderen Anwendungen
 - Sicherheit und Zugriffskontrolle
- **Aufgabe der Persistenzschicht**
 - Dauerhafte Speicherung von Anwendungsdaten in Datenbanken oder Dateien

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Architektur unseres Sonnenschein-Systems



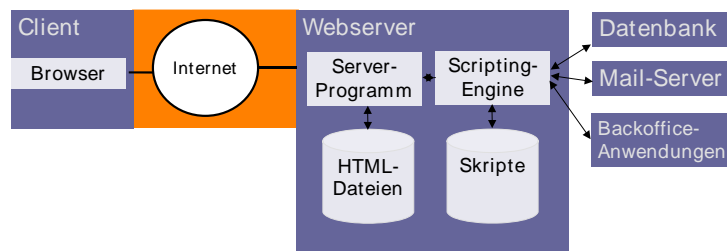
Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

2. Das HTTP-Protokoll

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Was ist das HTTP-Protokoll?

- Was ist ein Protokoll?
 - Gemeinsame Sprache zwischen Client und Server
- HTTP ist das Protokoll zur Kommunikation zwischen Browser und Webserver.
- HTTP ist somit das Protokoll, auf dem das WWW beruht.



- Warum interessant?
 - Hintergrundinformation zum besseren Verständnis einer Web-Applikation
 - Hintergrundinformation zum Thema Privacy

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Aufruf einer HTML-Seite (1)

1. Eingabe einer URL durch den Benutzer in den Browser

`http://www.wiwiss.fu-berlin.de/suhl/index.htm`

Protokoll Domain-Name Dokumentenpfad

2. Browser erkennt das HTTP-Protokoll

- Aufgabe: Anfragen an den Webserver stellen/Übertragung der Antwort

3. Browser stellt Anfrage (Request) an einen Domainname-Server

- Aufgabe: Übersetzung von Domain-Namen in IP-Adressen

4. Domain Name Server übersetzt den Domainnamen

- `www.wiwiss.fu-berlin.de` ⇒ IP 160.45.136.52
- Vorgehensweise: Tabelle mit Domainnamen und zugehörigen IP-Adressen

5. Aufbau einer TCP/IP-Verbindung mit dem Server

- Notwendige Informationen: IP-Adresse, Port-Nummer
- Portnummer dient der Diensterkennung
- HTTP: Port-Nr. 80; FTP: Port-Nr. 20,21; DNS: Port-Nr. 53; SMTP: Port-Nr. 25

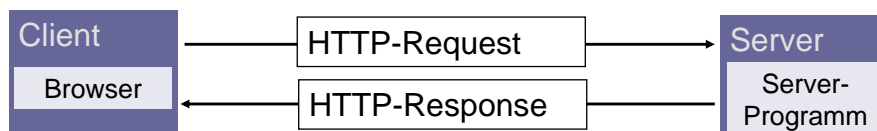
Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Aufruf einer HTML-Seite (2)

6. Browser fordert Dokument an (Request)
 - **HTTP-Request:** „Bitte schicke mir das Dokument /suhl/index.htm“
7. Server prüft HTTP-Request
 - Gibt es die angeforderte Datei?
 - Darf der Benutzer auf die Datei zugreifen?
8. Server sendet **HTTP-Response** (Antwort)
 - Möglichkeiten:
 - „OK. Schicke Datei“
 - „Habe Datei nicht gefunden!“
 - „Zugriff auf Datei nicht erlaubt“
 - „Server Fehler“
9. Browser stellt die empfangene Datei dar
 - Anzeigen der HTML-Seite
 - ggf. weitere Browser-Requests für Seitenelemente wie Grafiken
 - ggf. Ausführen von Programmcode in der Datei
 - ggf. Zugriff auf externe Programme zur Darstellung (Flash, PDF)

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Ablauf einer HTTP-Transaktion



- **Inhalt eines Requests**
 - Dokumentenanforderung : „Bitte schicke mir das Dokument /suhl/index.htm“
 - Informationen über den Client: „Ich bin ein Internet Explorer 5 und spreche am liebsten deutsch.“
- **Inhalt einer Response**
 - Angefordertes Dokument: „Alles hat geklappt, anbei schicke ich das angeforderte Dokument. Bei dem Dokument handelt es sich um eine HTML-Seite“
 - Informationen über den Server: „Ich bin ein Apache Server, verstehe PHP und laufe unter Windows NT.“

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

HTTP-Beispiel-Request

```
GET /books/index.htm HTTP/1.1
Accept: image/png, image/gif, *.*
Accept-Language: de, en
Accept-Encoding: gzip
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
Host: www.amazon.com
Referer: http://www.amazon.com/index.htm
```

■ Dokumentenanforderung

```
GET /books/index.htm HTTP/1.1
└─┬──────────┬──────────┬──────────┘
  Methode   Dokument   Protokoll-Version
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

HTTP-Methoden

Methode	Erklärung
GET	Dokument vom Server abrufen
HEAD	Nur Informationen über das Dokument gewünscht, nicht das Dokument selber.
POST	Client sendet Informationen an den Server (Wird zur Übertragung von Formular-Daten verwendet)
PUT	Dokument auf dem Server speichern
DELETE	Dokument auf dem Server löschen
OPTIONS	Anfrage an den Server, welche HTTP-Methoden er unterstützt

- **PUT und DELETE sind bei Produktiv-Servern meistens deaktiviert bzw. auf bestimmte Nutzer beschränkt.**

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Aufbau einer HTTP-Response

```
HTTP/1.1 200 OK
Date: Mon, 18 Jan 2005 20:54:26 GMT
Server: Apache/1.3.6 (UNIX)
Last-Modified: Mon, 06 Dec 2003 14:06:11 GMT
Content-length: 6345
Content-Type: text/html
```

```
<html>
  <head><title>CB CD-Shop</title></head>
  <body><h1>Willkommen beim CB CD-Shop</h1> ....
```

■ Erste Zeile der Response

HTTP/1.1 200 OK



Protokoll-Version Statuscode Erläuterung

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Server-Response-Codes

Code	Erklärung
200 OK	OK, die Response enthält das angeforderte Dokument
301 Moved Permanently	Angeforderte Ressource ist unter neuer URL zu erreichen. Die URL wird mit dem Location-Header übergeben.
304 Not Modified	Das Dokument hat sich seit dem letztem Aufruf nicht verändert.
400 Bad Request	Syntaxfehler im Request
401 Unauthorized	Die Ressource ist passwortgeschützt. Der Client wird aufgefordert, sich mit Benutzername und Kennwort zu autorisieren.
403 Forbidden	Zugriff verweigert, ohne Möglichkeit zur Autorisation.
404 Not Found	Das angeforderte Dokument existiert nicht.
500 Internal Server Error	Ein Programm- oder Konfigurations-Fehler des Servers macht die Bearbeitung der Anfrage unmöglich.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

3. Der Apache-Webserver

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Was macht ein Webserver?

Grundfunktionen eines Webservers:

- **Warten auf Anfragen**
 - Der Server horcht an Port 80 nach HTTP-Requests.
- **Anfragen beantworten, indem er:**
 - entweder ein Dokument aus dem Dateisystem zurückliefert oder
 - ein Programm auf dem Server startet, das ein Dokument erzeugt.

Zusatzfunktionen:

- **Berechtigung der Anfrage prüfen**
- **Anfragen protokollieren (Logging)**

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

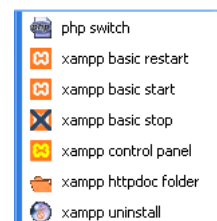
Der Apache Webserver

- **Apache ist der weltweit verbreitetste Webserver**
 - **50.42% Marktanteil** (April 2008)
 - **Apache wird von 83 Millionen Websites eingesetzt** (April 2008)
 - **Vergleich:**
 - **Microsoft Internet Information Server: 35.33 % Marktanteil (58 Mio. Sites)**
- **Apache ist für eine breite Palette von Plattformen verfügbar**
 - **Unix, Linux, Window 2000, XP, Vista**
 - **Netware 5.x, OS/2, BS 2000, AS 400**
- **Apache ist OpenSource**
 - **Apache ist kostenlos als kompiliertes Programm und als Quellcode verfügbar (Bezug über apachefriends.org oder httpd.apache.org).**
 - **Apache wird von einer Gruppe freiwillig und unbezahlt arbeitender Programmierer ständig weiterentwickelt (ASF: <http://www.apache.org>).**
- **Aktuelle Version: Apache 2.2.9**
- **Dokumentation zu Apache**
 - <http://httpd.apache.org/docs-2.0/>

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Apache installieren und ausführen

- **Installation auf Windows-Plattformen**
 - **Aktuelle Version XAMPP von apachefriends.org herunterladen.**
 - **Installationsroutine ausführen.**
 - **Server über das Startmenü starten.**
 - **Server testen durch Eingabe von <http://localhost/>**
- **XAMPP enthält:**
 - **Webserver: Apache 2.2.8**
 - **Skriptsprache: PHP 5.2.5**
 - **Datenbank: MySQL 5.0.51, PHPMyAdmin 2.11.4**
 - **FTP-Server: FileZilla FTP Server**
 - **Verschlüsselungssoftware: mod_ssl, Openssl**



Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Verzeichnisstruktur von Apache im Pool

apache2/	_____	Apache Root-Verzeichnis
conf/	_____	Verzeichnis für Konfigurationsdateien
httpd.conf		Apache-Konfigurationsdatei
mime.types		Dateiendungen => MINE-Types
cgi-bin/	_____	Verzeichnis für CGI-Programme
cgi.exe		
htdocs/	_____	Verzeichnis für HTML-Dokumente
index.htm		
kategorien.htm		
logs/	_____	Verzeichnis für Protokoll-Dateien
access.log		Zugriffs-Protokoll
error.log		Fehler-Protokoll
modules/	_____	Verzeichnis für Erweiterungsmodule
mod_ssl.so		
php4/	_____	PHP Verzeichnis
php.ini		
bin/		
apache.exe	_____	Apache-Programmdatei
ApacheMonitor.exe		

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Konfiguration von Apache

- Apache wird über die Konfigurationsdatei „httpd.conf“ administriert

Direktive	Bedeutung
Servername www.wiwiss.fu-berlin.de	Domainname des Servers festlegen
Listen 80	Port, an dem der Server auf Anfragen wartet
TimeOut 30	Maximale Bearbeitungszeit festlegen
ServerRoot "C:/apache"	Hauptverzeichnis des Servers festlegen
DocumentRoot "C:/apache/htdocs"	Verzeichnis für HTML-Dokumente festlegen
ServerAdmin chirs@bizer.de	Email-Adresse des Administrators
DirectoryIndex index.htm	Standarddatei, wenn Anfragen keinen Ressourcennamen enthalten

- Neustart nach Änderungen in der Konfiguration erforderlich!

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Logging

- Apache protokolliert alle Zugriffe und Fehler

- Zugriffe normalerweise in „logs/access.log“
- Fehler normalerweise in „logs/errors.log“

- Logging-Direktiven für die Konfigurationsdatei „httpd.conf“:

Direktive	Bedeutung
LogFormat "%h %i" kurz	Definition eines angepassten Log-Formats (Formatstring, Name des Formats) (Erklärung Formatstring: nächste Folie)
CustomLog logs/access2.log kurz	Festlegung der Protokoll-Datei im benutzerspezifischen Format (Dateiname, Name des Formats)
ErrorLog logs/error.log	Festlegung der Fehler-Protokoll-Datei
HostnameLookups Off	IP-Adresse oder Hostname aufzeichnen (On Off)

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Definition eigener Log-Formate

- LogFormat "Formatstring" Name

Direktive	Bedeutung
%h	IP-Adresse oder Hostname des Clients
%t	Uhrzeit des Zugriffs
%r	Erste Zeile des HTTP-Requests (GET ...)
%U	Angeforderte URL
%s	Server-Response-Code (z.B. 200 oder 404)
%b	Größe des übertragenen Dokuments in Bytes
%u	Benutzername eines authentifizierten Nutzers
%{header}i	Header-Zeile des HTTP-Requests z.B. %{User-agent}i : Browser des Benutzers %{Referer}i : Zuletzt aufgerufene Seite
%{header}o	Header-Zeile der HTTP-Response z.B. %{Content-Type}o

- Der Formatstring kann neben den Direktiven auch beliebigen Text enthalten
- Beispiel: LogFormat "Host %h Request %r Browser %{user-agent}i" Kurzlog

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Auswertung der Log-Files

- Bei stark frequentierten Websites entstehen täglich Giga-Bytes an Log-Informationen.
- Zur Auswertung werden sogenannte Log-Analyzer eingesetzt.
- Funktionsumfang der Analyzer
 - Breites Spektrum von einfachen statistischen Auswertungen bis zu
 - fortgeschrittenen Data-Mining-Techniken.
- Ein einfacher Log-Analyzer ist im XAMPP Paket enthalten.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Zugriffsschutz

- Der Zugriff auf einzelne Server-Bereiche bzw. Dokumente lässt sich beschränken
- Berechtigungstypen
 - Berechtigungen nach IP-Adresse bzw. Domainnamen
 - Berechtigungen mittels Authentifizierung (Name/Passwort)
- Zur Definition von Berechtigungen muss festgelegt werden:
 1. wer berechtigt ist
 2. für welchen Bereich auf dem Server die Berechtigung gelten soll
- Bereiche werden über sogenannte Block-Direktiven definiert

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Block-Direktiven

- Über Block-Direktiven wird definiert, auf welchen Bereich sich die eingeschlossenen Direktiven beziehen.

- Beschränkung auf ein Verzeichnis

```
<Directory "C:/APACHE/htdocs/intern">  
...  
</Directory>
```

- Beschränkung nach URL relativ zu DokumentRoot

```
<Location /aussendienst>  
...  
</Location>
```

- Beschränkung auf einzelne Dateien

```
<Files info.txt>  
...  
</Files>
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

IP- und Host-basierter Zugriffsschutz

- Nur die Rechner mit der IP Nummer 160.45.12.12 und 13 dürfen auf das Verzeichnis „intern“ zugreifen

```
<Directory "C:/APACHE/htdocs/intern">  
    Order deny,allow  
    Deny from all  
    Allow from 160.45.12.12 160.45.12.13  
</Directory>
```

- Kein Rechner, dessen Hostname auf .amazon.de endet, darf auf das Verzeichnis „intern2“ zugreifen

```
<Directory "C:/APACHE/htdocs/intern2">  
    Order allow,deny  
    Allow from all  
    Deny from .amazon.de  
</Directory>
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Authentifizierung über HTTP

- HTTP ermöglicht die Authentifizierung eines Nutzers mittels Benutzername und Passwort

- Ablauf der Authentifizierung

1. Client fordert eine Ressource an
2. Server erkennt, dass Ressource passwortgeschützt ist und fordert Client zur Authentifizierung auf

```
HTTP/1.1 401 Unauthorized
....
WWW-Authenticate: BASIC realm="Aussendienst"
```

3. Browser fragt Benutzername und Passwort ab und schickt die Eingaben an den Server (Benutzername:Passwort Base 64 kodiert)

```
GET /index.htm HTTP/1.1
....
Authorization: BASIC d2VibWFzZdGVyOnpvcWlhNHY=
```

4. Server liefert Dokument

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Passwort-basierter Zugriffsschutz

- Passwörter und Kennwörter werden in einer Textdatei gespeichert

mitarbeiter.txt

```
Mueller:yxz4
Meier:Badman
Schulze:435675
```

- Aktivierung des Zugriffsschutzes

```
<Directory "C:/APACHE/htdocs/intern">
  AuthType Basic
  AuthName Mitarbeiter
  AuthUserFile "C:/APACHE/mitarbeiter.txt"
  Require valid-user
</Directory>
```

- AuthType Basic : Art der Authentifizierung (Basic, Digest)
- AuthName Mitarbeiter : frei definierbarer Authentifizierungs-Bereich
- AuthFile ... : Pfad zur Datei mit Benutzernamen und Kennwörtern
- Require valid-user : Zugriffsbedingung festlegen (alle oder einzelne User)
Require user Mueller

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Sicherheitsmängel der HTTP-Authentifizierung

- **Benutzername und Passwort werden unverschlüsselt übertragen**
 - Der Base 64 Algorithmus ist kein Verschlüsselungsverfahren, sondern nur eine Transformation mittels eines allgemein bekannten Schemata
 - Ein Angreifer, der Datenpakete bei der Übertragung über das Internet abfängt, gelangt an das Passwort
- **Der Webserver erlaubt beliebig viele Authentifizierungs-Versuche**
 - Ein Angreifer kann das Passwort durch Ausprobieren erraten.
 - Es gibt entsprechende Tools, die Millionen gängiger Passwörter durchprobieren.
- **Fazit**
 - HTTP Authentifizierung bietet nur einen sehr eingeschränkten Schutz.
 - Bei echten Sicherheitsanforderungen sollte Secure Socket Layer (SSL) eingesetzt werden. Über Secure Socket Layer (SSL) können Server und Client mittels Zertifikate authentifiziert werden. Der gesamte HTTP Datenverkehr findet verschlüsselt statt.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

4. PHP

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Gliederung PHP

- Was ist PHP?
- Allgemeine Sprachkonstrukte
- Übergabe von Parametern zwischen Skripten
- Serverseitige Auswertung von Formulardaten
- Seitengenerierung mittels Require-Befehlen

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

PHP

- PHP ist eine serverseitige Skriptsprache zur dynamischen Erzeugung von Webseiten.
 - Den Grundstein zu PHP legte 1994 Rasmus Lerdorf der sich einige Tools für seine private Homepage programmierte.
 - Heute ist PHP das am meisten installierte Modul für Apache: 20,917,850 Domains (Netcraft, April 2008)
- PHP ist OpenSource
 - ca. 250 Entwickler beteiligen sich weltweit an der Entwicklung von PHP
 - der Quelltext und die Binaries von PHP sind frei verfügbar.
- PHP ist für Windows-, Unix-, Linux-Plattformen verfügbar.
- Die Syntax von PHP ist an C, JavaScript und Perl angelehnt.
- Aktuelle Version: PHP 5.2.6
- Erweiterbarkeit von PHP
 - Es gibt Funktionsbibliotheken zu fast jedem erdenklichen Thema (dynamische PDFs, XML, Java, COM ...).
 - Der Sprachumfang von PHP ist deshalb mittlerweile auf ca. 2000 Befehle und Funktionen angewachsen.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Beispielskript: Hello World 1000

test.php

```
<html>
<head>
  <title>Testskript PHP</title>
</head>
<body>
<P>Der folgende Text wurden von PHP erzeugt:</P>
<?
for ($i=1; $i<1000; $i++) {
  echo "Zeile " . $i . ": Hello World! <BR>";
}
?>
</body>
</html>
```

HTML-Code

PHP-Code

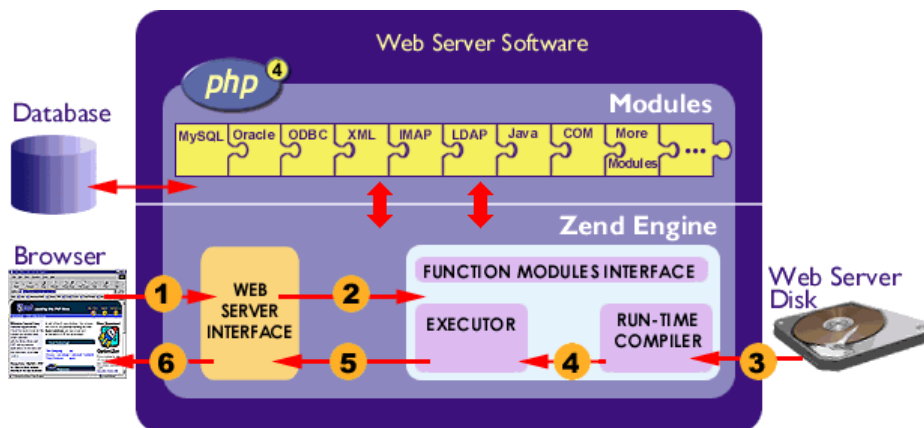
HTML-Code

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Die Software-Architektur von PHP

■ PHP besteht aus

- der Zend-Engine, die über einen Run-Time-Compiler (Interpiler) die Skripte ausführt und
- einer Vielzahl von Modulen zum Zugriff auf externe Ressourcen.



Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Installation von PHP

■ Bezug von PHP für Windows

- Über die PHP-Homepage: www.php.net/downloads.php
- XAMPP: Eine relativ aktuelle Version von PHP ist im XAMPP-Paket enthalten: www.apachefriends.org

■ Installation

- PHP wird in Apache oder Microsoft Internet Information Services als Modul integriert.
- Direktiven zur Integration als Apache Module in der Apache-Konfigurationsdatei „httpd.conf“:

```
LoadModule php5_module
" c:/apachefriends/xampp/apache/bin/php5apache2.dll "

AddType application/x-httpd-php .php .php5

(Wird von XAMPP automatisch erledigt)
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Konfiguration von PHP

- PHP wird über die Konfigurationsdatei „php.ini“, die sich im `C:\apachefriends\xampp\apache\bin\php.ini` Verzeichnis befindet, konfiguriert.
- Wichtige Konfigurations-Direktiven:

Direktive	Bedeutung
<code>memory_limit = 8M</code> <code>max_execution_time = 30</code>	Festlegung der maximalen Ausführungszeit je Skript und der Speichernutzung
<code>display_errors = On</code> <code>error_log = "C:\error.log"</code>	Festlegung wie PHP mit Fehlern umgehen soll.
<code>SMTP=smtp.server.de</code> <code>sendmail_from =</code> <code>php@server.de</code>	Festlegung der Konfiguration zum automatischen Verschicken von eMails.

- Wir benutzen in der Übung eine Standard-Konfiguration.
- Mittels der Funktion `phpinfo()` kann man sich aktuelle Konfiguration anzeigen lassen.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

PHP-Ressourcen in Netz

■ Internationale PHP-Sites

- www.php.net (Offizielle PHP-Homepage)
 - Download von PHP
 - Dokumentation online und zum Download (ca. 1000 Seiten)
 - Linksammlung zu vielen PHP-Ressourcen
- <http://pear.php.net/>
 - PEAR - PHP Extension and Application Repository, Code-Bibliothek
- www.zend.com
 - Code-Bibliothek, Diskussionsforen
 - Kommerzielle Angebote (Entwicklungsumgebung, Cache, Consulting)

■ Deutsche PHP-Sites

- www.php-center.de (Tutorials, Artikel, Tips & Tricks, Jobs)
- www.php-homepage.de (Downloads und Diskussionsforen)
- www.phpmag.de/ (Zeitschrift zu PHP)

■ Deutsche PHP-News-Group

- news://de.comp.lang.php
- FAQ zur Newsgroup von Kristian Koehntopp
<http://www.php-faq.de> (ca. 250 Seiten)

■ Empfohlenes PHP Tutorial: <http://tut.php-q.net/>

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Allgemeine Sprachkonstrukte von PHP

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Einbindung von PHP in den HTML-Code

■ Kurzschreibweise

```
<?
    echo „Hallo, ich bin die erste Anweisung<BR>“;
    echo „Und ich die zweite!“;
?>
```

■ alternative Schreibweise

```
<?php
    echo „Hallo, ich bin die erste Anweisung<BR>“;
    echo „Und ich die zweite!“;
?>
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Variablen

- Variablen werden durch ein \$-Zeichen gekennzeichnet
- Es wird zwischen Groß- und Kleinschreibung unterschieden
 - \$zaehler != \$Zaehler
- Datentypen und Typenumwandlung
 - Variablen müssen nicht deklariert werden.
 - Der Datentyp einer Variablen wird bei der ersten Zuweisung eines Wertes automatisch ermittelt.
 - Falls erforderlich findet eine automatische Typenumwandlung statt.
 - Intern gibt es folgende Datentypen: integer, double, string, array, object
- Zuweisung
 - \$a = 1; \$b = "Hallo"; \$c = "11";
- Variablen auf ihre Existenz testen bzw. löschen
 - Test mit isset: `if(isset($a)) { echo "Es gibt a." }`
 - Variablen löschen: `unset($a); echo "Es gibt a nicht mehr.";`

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Rechen- und String- und Ausgabeoperationen

■ Rechenoperationen

```
$a = $b + $c;   $a = $b - $c;  
$a = $b * $c;   $a = $b / $c;
```

```
$a = $b % $c; // Modulus, Rest der ganzzahligen Division  
$Zahl++;      // inkrementieren  
$Zahl--;      // dekrementieren
```

■ Strings verbinden

(weitere Stringoperationen folgen in der übernächsten Übung)

```
$ergebnis = $name . " " . $nachname; // Ein . verbindet Strings
```

■ Zeichenketten in den HTML-Code ausgeben

```
echo $ergebnis; // Ausgabe einer Variablen in den HTML-Code  
print $ergebnis; // Ausgabe einer Variablen in den HTML-Code
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Kontrollstrukturen

■ Syntax

```
if($tag == "Montag"){  
    echo "Heute ist Montag!";  
} else {  
    echo "Heute ist nicht Montag!";  
}
```

■ Vergleichsoperatoren

- Gleich: `if($SinnDesLebens == 42) $ergebnis = 1;`
- Ungleich: `if($SinnDesLebens != 42) $ergebnis = 0;`
- Größer: `if($SinnDesLebens > 42) $ergebnis = 0;`
- Kleiner: `if($SinnDesLebens < 42) $ergebnis = 0;`

Logische Operatoren

- UND: `if(($Sinn == 42) and ($Wetter == "Schön")) $ergebnis = 1;`
- ODER: `if(($Sinn == 42) or ($Wetter == "Schön")) $ergebnis = 1;`
- NEGATION: `if !($Sinn == 42) $ergebnis = 0;`

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Schleifen

■ Schleifen mit „for“

```
for($i = 1; $i <= 100; $i++) {  
    $x = $i * $i;  
    echo "<br>Das Quadrat von " . $i . " ist " . $x;  
}
```

■ Schleifen mit „while“

```
$zaehler = 1;  
while($zaehler < 100) {  
    echo "zaehler ist jetzt:" . $zaehler . "<br>";  
    $zaehler++;  
}
```

■ Schleifen mit „do-while“

```
do {  
    $zaehler++;  
    echo "zaehler ist jetzt:" . $zaehler . "<br>";  
} while($zaehler < 100);
```

■ Schleifen abbrechen: break;

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Beispiel for() Schleife

- Schleife mit einer feststehenden Anzahl von Durchläufen.
- Bei jedem Durchlauf wird der Schleifenzähler um 1 erhöht.

■ Beispiel

```
$b = 0;  
for($a = 1; $a<=3; $a++) {  
    $b = $b + $a;  
}  
echo $b;
```

Tischtest zum
Beispiel

a	b
	0
1	
	1
2	
	3
3	
	6
4	

■ Vorgehen von PHP

1. \$b=0 setzen.
2. Zähler a auf den Startwert setzen.
3. Es wird geprüft, ob der Zähler \$a größer als der Endwert ist. In diesem Fall wird die Schleife beendet.
4. \$b = \$b + \$a wird berechnet.
5. Der Zähler \$a wird um 1 erhöht.
6. Sprung zurück zu Schritt 3

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Arrays

Ein Array ist eine **Datenstruktur**, die aus einer Menge von Variablen besteht.

- Anders gesagt: Ein Array ist eine Liste, deren Elemente sich über den Listennamen und die Nummer/Name des Elements ansprechen lassen.
- Beispiel

```
$messung[1] = 5;  
$messung[2] = 3;  
echo "Zweite Messung: " . $messung[2];
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Beispiel zu Arrays

```
$zahlen[1] = 5;  
$zahlen[2] = 7;  
$zahlen[3] = 1;  
$zahlen[4] = 3;  
while ($zahlen[2] > $zahlen[1]) {  
    $zahlen[2] = $zahlen[2] - 1;  
    $zahlen[3] = $zahlen[2];  
}  
$zahlen[4] = $zahlen[3];
```

Tischtest

zahlen[1]	zahlen[2]	zahlen[3]	zahlen[4]
0	0	0	0
5			
	7		
		1	
			3
	6		
		6	
	5		
		5	
			5

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Arrays

- In PHP können Arrays über Zahlen oder über Namen indiziert werden.
- Auch Arrays müssen nicht deklariert werden.
- Nummerierte Arrays

```
$mitarbeiter[1] = "Klaus";  
$mitarbeiter[2] = "Carolin";
```

- Über Namen indizierte Arrays

```
$gehalt["Fritz"] = 6000;  
$gehalt["Egon"] = 8000;
```

- Array-Funktionen

```
$anzahl = count($gehalt); // Elemente des Arrays zählen  
$gehalt = sort($gehalt); // Array sortieren  
$JN = in_array("Nadel", $heusack); // Prüft ob Wert in Array
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Übung zur Auffrischung

```
$a[1] = 4;  
$a[2] = 6;  
$a[3] = 8;  
do {  
    if ($a[3] > $a[2]) {  
        $a[3] = $a[3] - 1;  
    } else {  
        $a[3] = $a[3] + 1;  
    }  
    $a[1] = $a[1] + 1;  
} while (a[1] != a[2]);
```

- Welches Ergebnis liefert ein Tischtest des nebenstehenden Codes?

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Kommentare

■ Einzeilige Kommentare

```
// Das ist ein Kommentar  
print "Einzeilig, wie in C++."; // bis zum Zeilenende  
print "Einzeilig, wie in Unix Shell."; # bis zum Zeilenende
```

■ Mehrzeilige Kommentare

```
/*  
Alles Zeilen bis zum naechsten Auftreten  
der Zeichenkombination Sternchen und Slash  
sind Bestandteil des Kommentars.  
*/
```

■ Warum kommentieren? Quellcode wird lesbar und Fehlersuche einfacher

■ Was sollte kommentiert werden? (Bitte so bei unserem Katalog)

- Skriptdatei insgesamt (Autor, eMail, Datum letzte Änderung)
- Funktionen (Zweck der Funktion und Erklärung der Parameter)
- größere Anweisungsblöcke
- SQL-Kommandos

■ Kommentare werden nicht an den Browser weitergeleitet

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Übergabe von Parameter zwischen Skripten

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Parameterübergabe zwischen Skripten

- Für viele Anwendungen ist es nötig Skripten Parameter zu übergeben
 - z.B. Kategorienseite eines CD-Shops: Welche Kategorie?
 - z.B. CD-Anzeigeseite: Welche CD soll angezeigt werden?
- Parameter werden an URLs angehängt
`http://localhost/kategorie.php?name=Electronic`
- Wenn PHP einen Parameter empfängt wird automatisch eine Variable mit dem Namen und dem Wert des Parameters erzeugt und den Array `$_GET[]` zugeordnet.
- Zugriff auf Parameter: `$_GET["name"]`
- Syntax der angehängten Parameter
 - Parameter haben die Form: `Parametername=Wert`
 - `?` trennt URL und Parameter
 - `&` verbindet mehrere Parameter
- Sonder- und Leerzeichen in Parametern
 - Sonder- und Leerzeichen in URLs sind durch den Hexadezimalcode des entsprechenden ASCII-Zeichens zu ersetzen. Schreibweise: `%Hexcode`
 - PHP-Funktion `urlencode`: `$parameter=urlencode($parameter)`

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Codebeispiel: Parameterübergabe

- An die HTML-Links wird ein Parameter angehängt.

`index.htm`

```
...  
<a href="kategorie.php?kategorie=PopRock">PopRock</a><br>  
<a href="kategorie.php?kategorie=Alternativ">Alternativ</a><br>  
<a href="kategorie.php?kategorie=Electronic">Electronic</a><br>  
...
```

- In PHP wird automatisch eine entsprechende Variable erzeugt, die anschließend ausgegeben werden kann.

`kategorie.php`

```
...  
<H1>Kategorie: <? echo $_GET["kategorie"] ?></H1>  
...
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Auswertung von Formularen mit PHP

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Einfache Auswertung von Formularen

- Im HTML-Code des Formulars werden alle Felder eindeutig über name-Attribute bezeichnet.
- Das HTML-Formular ruft ein PHP-Skript auf.

Formular.htm

```
<form name="Test" method="post" action="Auswertung.php">
  <input type="text" name="vorname"><br>
  <input type="text" name="nachname"><br>
  <input type="submit" name="Abschicken" value="Abschicken">
</form>
```

- In PHP werden automatisch Variablen mit den Namen der Formularfelder erzeugt und je nach Übertragungsart method="post" oder "get" dem Array \$_GET oder \$_POST zugeordnet.

Auswertung.php

```
<P>Sie haben folgende Werte eingegeben:</P>
<?
  print "Vorname: " . $_POST["vorname"] . "<BR>";
  print "Nachname: " . $_POST["nachname"] . "<BR>";
?>
<P>Bis bald!</P>
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Anforderungen an die Formularauswertung

1. Vollständigkeit

- Alle notwendigen Formularfelder sollten ausgefüllt werden.
- Falls nicht
 - sollte der Benutzer einen Hinweis auf fehlende Felder erhalten und
 - es sollte keine Neueingabe bereits eingegebener Daten erforderlich sein.

2. Plausibilität

- Alle Daten sind auf Plausibilität zu prüfen (eMail-Adresse? Kreditkartennr.?).

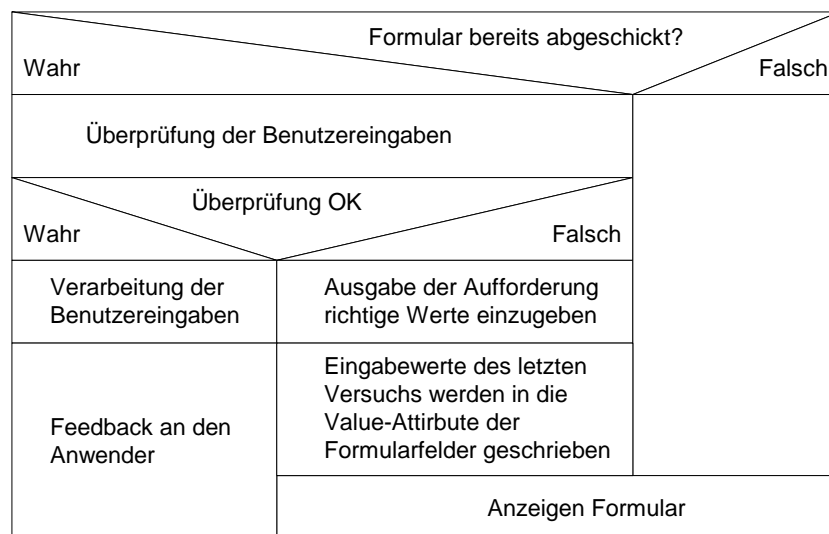
■ Exkurs: Sicherheit

- Eine client-seitige Prüfung reicht nicht aus, da JavaScript deaktiviert werden kann.
- Es sind Angriffe auf den Server mittels Formularinhalten möglich:
 - Über Zeichen wie " oder ; kann versucht werden Befehle beispielsweise in der Datenbank auszuführen.
 - Der Server bzw. die Datenbank kann mit Megabyte großen Formularinhalten überschwemmt werden. (Denial-of-Service-Angriff)
- Fazit: Bei professionellen Anwendungen sind Eingaben zusätzlich auf Größe und sicherheitsrelevante Zeichen zu prüfen.

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Professionelle Auswertung von Formularen

- Das Formular-Skript ruft sich selber auf (Aff Formular-Technik).
- Darstellung der Anwendungslogik als Struktogramm:



Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Code: Affenformular (Anfang)

```
<html><head><title>Formulardemo</title></head><body>
<? // Pruefen ob Formular abgeschickt wurde oder nicht
if (isset($_POST["abgeschickt"])) {
    // Validierung: Pruefen ob das Feld ausgefuellt wurde
    $fehlermeldung = "";
    if (strlen($_POST["name"]) == 0) $fehlermeldung = $fehlermeldung . "Name ";
    // Pruefen ob ein Fehler gefunden wurde
    if ($fehlermeldung == "") {
        // Verarbeiten der Formulareingaben z.B. in Datenbank schreiben
        // Feedback an den Benutzer
        ?>
        <p>Vielen Dank!</p>
        <?
    } else {
        // Ausgabe der Fehlermeldung
        ?>
        <p>Bitte f&uuml;llen Sie folgende Felder aus: <? echo $fehlermeldung ?></p>
        <?
    } // Ende: Fehler oder nicht
} // Ende: Abgeschickt oder nicht
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Code: Affenformular (Fortsetzung)

```
// Formular anzeigen, falls :
// 1. Der Benutzer zum ersten mal auf die Seite kommt oder
// 2. das Formular schon abgeschickt wurde und die Pruefung einen Fehler ergab
if (!isset($_POST["abgeschickt"]) or ($fehlermeldung != "")) {
    ?>
    <h1>Bitte geben Sie ihren Namen ein:</h1>
    <form name="suche" method="post" action="rezensionschreiben_kurz.php">
    <input type="text" name="name" value='<? if (isset($_POST["name"]))
        echo $_POST["name"] ?>'>
    <input type="submit" name="abgeschickt" value="Submit">
    </form>
    <? } ?>
</body>
</html>
```

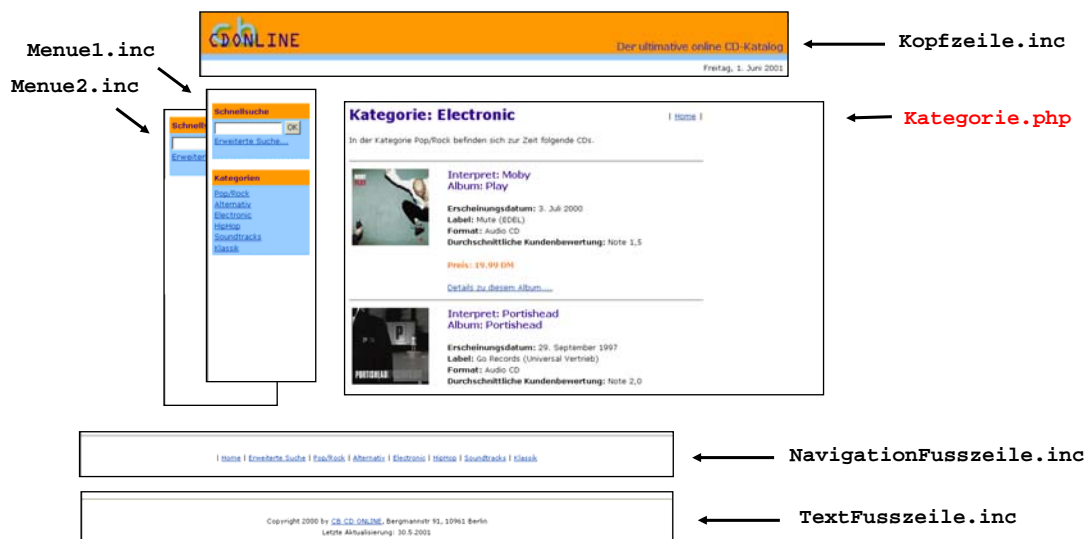
Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Seitengenerierung mittels Require-Befehlen

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Seitengenerierung mittels Require-Befehlen

Die Wartung von Webapplikationen lässt sich erleichtern, indem wiederkehrende Layout-Elemente zentral in eigenen Dateien gespeichert werden, die anschließend in eine Vielzahl von Seiten integriert werden.



Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)

Zusammenführung der Layoutelemente

Die Layout-Elemente werden anschließend mittels Require-Befehlen wieder zusammengesetzt.

```
index.php
<?
// Variablen die in den Includes verwendet werden festlegen
$Seitentitel = "CB CD-Shop - Kategorie: Electronic";
// Einbinden zentraler Funktionsbibliotheken (z.B. Formularprüffunktionen)
require("../includes/PhpScript/SeitenFunktionen.inc");
// Einbinden der Layoutbausteine Kopfzeile und Menue links
require("../includes/PageElements/Header.inc");
require("../includes/PageElements/KopfZeile.inc");
require("../includes/PageElements/MenuSucheKategorien.inc");
?>
    <h1>Kategorie: Electronic</h1>
    <p>Hier steht der eigentliche Inhalt der Seite.</p>
<?
// Einbinden der Fusszeilen
require("../includes/PageElements/NavigationFusszeile.inc");
require("../includes/PageElements/TextFusszeile.inc");
?>
```

Uwe H. Suhl, Chris Bizer: Systementwicklung – SS08 (Stand 19.6.08)