

WWW - Grundlagen und Technologie

Server-Side Technologien



Erik Wilde
TIK - ETH Zürich
Sommersemester 2001

Übersicht

- Anbindung Server/Applikation
 - HTTP Server ist spezialisiert auf HTTP
 - Konstruktion von *Web-based Applications*
- *Common Gateway Interface (CGI)*
 - ältester und einfachster Weg der Anbindung
 - sehr allgemeingültig, aber oft nicht optimal
- Weiterentwicklungen der Anbindung
 - Mechanismen zur Ausführung von Programmen
 - Mechanismen zum Einbetten von Code in HTML
- Kontrolle eines laufenden HTTP Servers
 - Erstellung und Interpretation von Statistiken

Aufbau einer Web-Applikation

- das Interface ist HTTP und HTML
 - Kommunikation mit dem Benutzer über HTTP
 - UI Design in HTML, u.U. DHTML oder Applets
- klassisches *3-tier Design* trennt
 - eine Komponente zur Darstellung
 - eine Komponente für die *Business Logic*
 - eine Komponente für die Datenhaltung (DBMS)
- nur die Darstellungssicht ist Web-basiert
 - Umbau auf anderes UI sollte einfach möglich sein
 - Business Logic und DB unabhängig vom Web

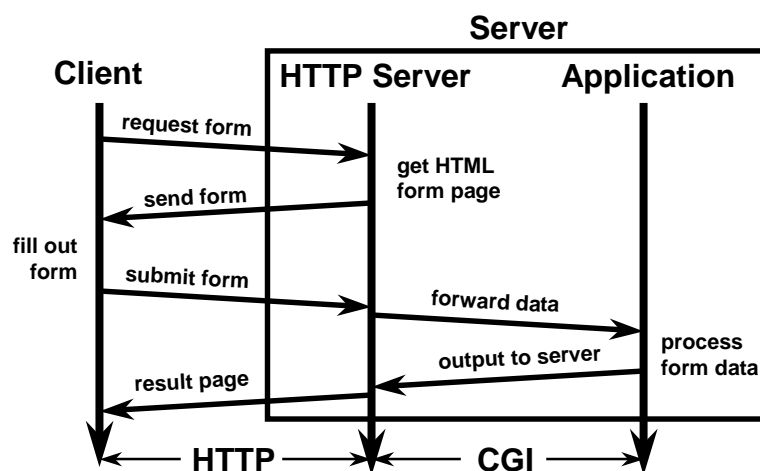
Erster Ansatz für eine Trennung

- das generelle Modell auf dem Web ist einfach
 - ein Benutzer schickt einen Request...
 - ...der Server verarbeitet des Request...
 - ...und schickt einen Response
- einfache Abbildung auf IPC möglich
 - der Request wird weitergeleitet zur Applikation...
 - ...die Applikation bearbeitet den Request...
 - ...und erzeugt als Resultat einen Response

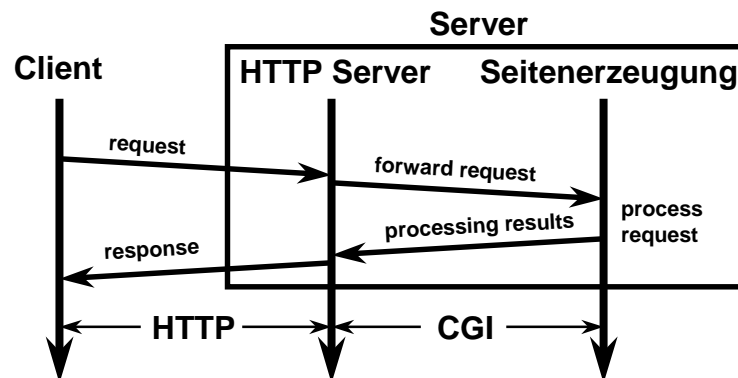
Common Gateway Interface (CGI)

- Verbindung von Programmen und Servern
 - HTML Seiten sind statische Dokumente
 - von Scripten generierte Seiten sind dynamisch
- die Verwendung von CGI ist vielfältig
 - Shell-Scripts, Perl-Scripts, C-Programme, ...
 - die Bearbeitung sollte nicht zu lange dauern
- Server und CGI-Skript kommunizieren durch
 - die Kommandozeile (beim Aufruf übergeben)
 - Environment-Variablen (vor dem Aufruf gesetzt)
 - Standard-Eingabe
 - Standard-Ausgabe

HTML Form Handling



Dynamische Seitenerzeugung



WWW (SS2001) - Server-Side Technologien

7

CGI Environment Variablen

- vom Request unabhängige Variablen
 - **SERVER_SOFTWARE**: Name und Version des WWW-Servers
 - **GATEWAY_INTERFACE**: CGI-Version
- je nach Request u.U. unterschiedlich gesetzte Variablen
 - **SERVER_NAME**: der Server-Name (DNS-Name)
 - **SERVER_PROTOCOL**: für die Anfrage verwendetes Protokoll
 - **SERVER_PORT**: für die Anfrage verwendeter Port
 - **REMOTE_HOST** oder **REMOTE_ADDR**: Ursprung der Anfrage

WWW (SS2001) - Server-Side Technologien

8

Resultate von CGI-Scripts

- Kontakt zum Server über Standard-Ausgabe
 - das Resultat der Anfrage (HTML Seite)
 - Verweis auf das Ergebnis (URI oder File-Pfad)
- meist erzeugt der Server den HTTP-Header
 - Erzeugung eines korrekten Headers mit *cgiutils*
 - ansonsten parsed headers mit Server-Direktiven
- Server-Direktiven steuern den Server
 - **CONTENT** gibt den MIME-Typ der Antwort an
 - **LOCATION** gibt an, wo das Ergebnis ist
 - **STATUS** gibt an, eine Status-Meldung zu senden

Vorteile von CGI

- Trennung in verschiedene Prozesse
 - HTTP-Server enthält allgemeine Web-Funktionen
 - Anwendungsspezifisches in separater Applikation
- Unabhängigkeit vom HTTP-Server
 - Applikation hängt nicht vom Server ab
 - von nahezu allen Servern unterstützt
- einfache Verwendung
 - sehr einfache Schnittstellendefinition
 - einfach erlernbar, einfach zu verwenden
- Unterstützung durch Software
 - Programmierbibliotheken in vielen Sprachen
 - bestehende Anwendungslösungen (HTML Forms)

Nachteile von CGI

- langsame Lösung
 - Implementierung über Interprozesskommunikation
 - Performance hängt sehr stark von der Plattform ab
 - Neustart der Programme bei jedem Aufruf
- auf einen Rechner beschränkt
 - verwendete Mechanismen sind immer lokal
 - häufige Variante: CGI Stubs
 - FastCGI als neue Variante (Apache Modul)
- Ausgabe sind immer komplette HTML Seiten
 - entspricht oft nicht der Anwendung
 - oft HTML als fixes Gerüst mit dynamischen Teilen

Alternativen zu CGI

- CGI Probleme sind Performance und Design
- Lösung des Performance-Problems
 - Überwindung der Schwachstellen von CGI
 - IPC und Effizienz müssen besser werden
- Lösung des Design-Problems
 - Entwicklung spezifischer Sprachen
 - Entwicklung spezifischer Softwareprodukte

APIs für Web-Server

- *Server Application Programming Interface*
- engere Bindung an den Web-Server
 - effizienter (gleicher Prozess wie Web-Server)
 - riskanter (Fehler beendet auch den Web-Server)
 - weniger flexibel (Bindung an ein bestimmtes SAPI)
- Internet SAPI (ISAPI) von Microsoft
 - Applikationen als Windows DLLs
- Netscape SAPI (NSAPI) von Netscape
 - verfügbar auf Unix-Varianten und Windows NT
- Modul-SAPI des Apache-Server
- weitere Variante: Server-Side Java (Servlets)

PHP3/4

- Script-Sprache zur Einbettung in HTML
 - syntaktisch ähnlich zu Perl und C
- Server verarbeitet Seite vor dem Senden
 - einfache Variante: CGI-Einbindung von PHP
 - effiziente Variante: PHP-Integration in den Server
- Verwendung weitergehender Mechanismen
 - Kombination von PHP und MySQL
 - Abfragen an eine Datenbank aus dem Script
- Unterstützung verschiedener Datenbanken
 - einfache Integration in bestehende Umgebung
 - einfache Erzeugung dynamischer Web-Seiten

Einbettung von PHP in HTML (I)

```
<?
/* includes */
require(".././././lib/common.php3");
require(".././././lib/lib_people.php3");
require(".././././lib/database.php3");
require(".././././lib/lib_access.php3");

/* Define variables */
$type = "Employee";
$editmode = 0; /* List of employees should not be editable */
$header = 1;

/* List System Engineering Group */
echo("<h3>System Engineering</h3>");
$group = "System Engineering";
$name = "";
$function = "";
view_list_people($group,$name,$function,$type,"",
                $header,$editmode,$database,$server);
...

```

Einbettung von PHP in HTML (II)

```
function query_get_primary_group($username,$database,$server)
{
    $acl = "employee"; // a comma separated list of users or groups
    that have the right to
    // perform this query
    $user = getenv("REMOTE_USER");

    if ( isReadAllowed($acl,$user,$server,$database) ) {
        connectToDatabase($server,$database);
        $result = mysql_query("SELECT grp FROM people WHERE
                               username='".$username."'");
        $row = mysql_fetch_array($result);
        switch ($row["grp"]) {
            case "Computer Engineering":
                $group = "tecempl";
                break;
            case "Communication Systems":
                $group = "komempl";
                break;
            case "System Engineering":
                $group = "sytempl";
                break;
        }
    }
    ...
}

```

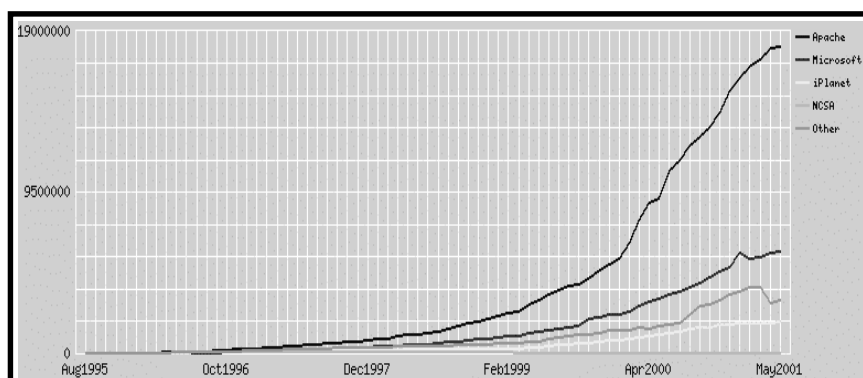
Active Server Pages (ASP)

- proprietäre Entwicklung von Microsoft
- umfassendere Funktionalität als CGI oder PHP
 - Grund jedoch COM und nicht ASP selber
- sehr enge Bindung an Microsoft-Produkte
 - Produkte für ASP auf Netscape und IBM Server
 - kein Zugriff auf Daten in nicht-Microsoft-Applikationen
- Scripting auf HTML-Seiten (Server-Side Scripts)
 - VBScript als *Default Scripting Language*
 - JScript oder *Plug-Ins* für andere Sprachen
- Zugriff auf COM (*ActiveX*) Komponenten
 - damit auch einfacher Zugriff auf ODBC-Datenbanken

WWW (SS2001) - Server-Side Technologien

17

Anzahl verwendeter Server (05/01)

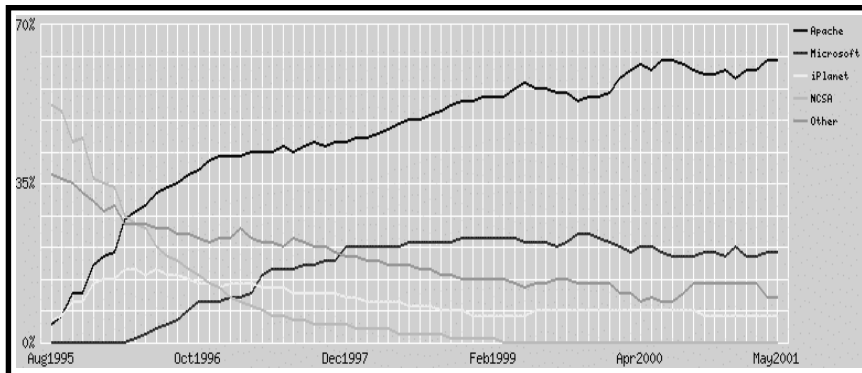


<http://www.netcraft.co.uk/Survey/>

WWW (SS2001) - Server-Side Technologien

18

Anteile verwendeter Server (05/01)



<http://www.netcraft.co.uk/Survey/>

WWW (SS2001) - Server-Side Technologien

19

Prozentuale Verteilung der Server

- Anzahl der laufenden Server
- keine Aussage über die Grösse der Server
 - Verwaltung grosser Web-Sites
 - Performance-Aspekte bei hoher Belastung
- keine zuverlässigen Statistiken über Anzahl der Responses (Anteil am Web-Traffic)
- Microsoft bietet bestmögliche Integration
- IBM liefert Apache als Web-Server aus
- Netscape setzt andere Schwerpunkte
 - auch klar bei der Browser-Entwicklung ersichtlich

WWW (SS2001) - Server-Side Technologien

20

Apache

- weitverbreitetster HTTP-Server (61%)
 - Microsoft Produkte auf 21% aller Server
- basiert auf dem NCSA httpd 1.3 (1995)
- public domain Projekt für viele verschiedene Unix-Plattformen (ebenso Windows)
- Design-Ziele: sicher, effizient, erweiterbar, aktuelle HTTP Entwicklungen
- <http://www.apache.org/> enthält komplette online-Dokumentation
- <http://www.apacheweek.com/> gibt viele nützliche Hinweise zur Konfiguration

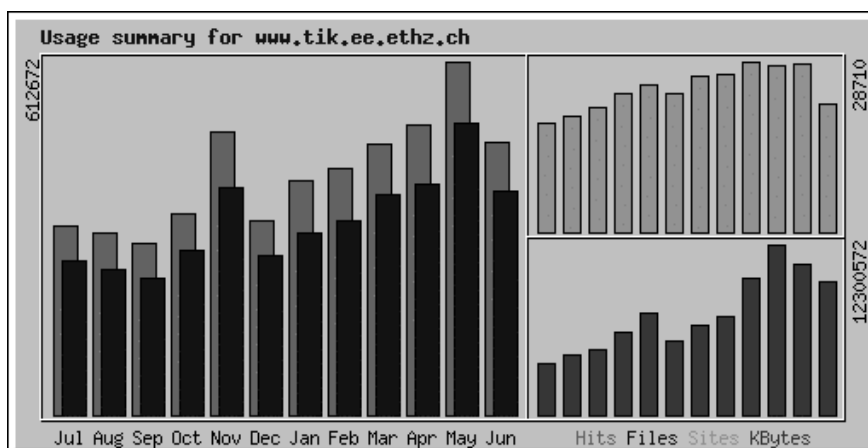
Apache Server-Side Includes (SSI)

- nützlich für mehrfache Verwendung von Files
- Server interpretiert das File vor dem Senden
- zwei Methoden für SSI Files:
 - Files haben die Endung `.shtml` anstatt `.html`
 - Files haben das *Execute-Bit* gesetzt (Unix-Varianten)
- verschiedene Direktiven
 - *include*: für die Einbindung anderer Files
 - *lastmod*: Datum der letzten Änderung
 - *flow control*: Seiten mit konditionalem HTML
 - *exec*: führt CGI-Scripts aus und fügt das Ergebnis in die Seite ein

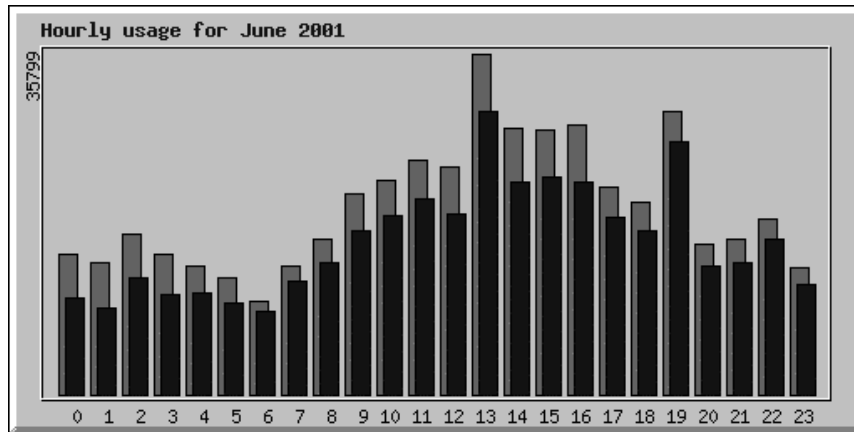
Statistiken TIK WWW Server

- Apache Server Version 1.3.6
- Information für interne und externe Zugriffe
- monatliche Hits (empfangene Requests)
 - 400'000, 10 min⁻¹ (Maximum 5 s⁻¹)
- monatlich übertragenes Datenvolumen
 - 8 GByte, 260 MByte/d, 3 kbit/s
- Zugriff nach User Agent
 - 47.2% Mozilla (Navigator + MSIE 5.x)
 - 34.5% MSIE (MSIE < 5.x)
- Statistik generiert mit *Webalizer*
 - <http://www.webalizer.com/>

Benutzung allgemein (00/01)



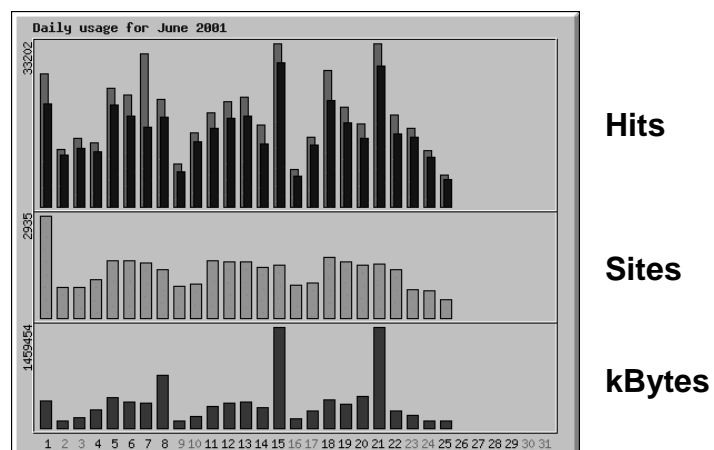
Benutzung pro Stunde (06/01)



WWW (SS2001) - Server-Side Technologien

25

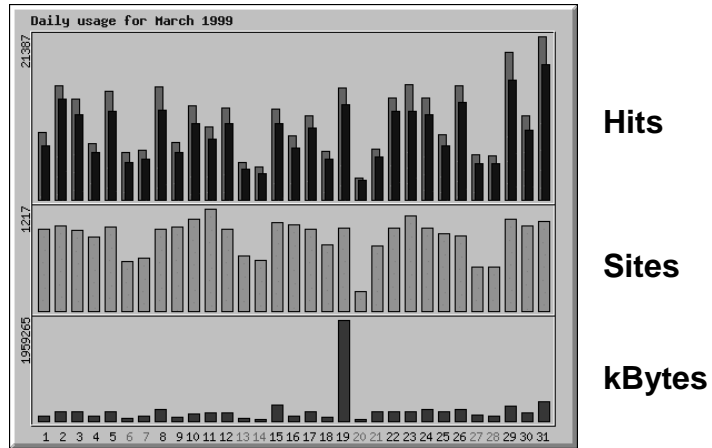
Benutzung pro Tag (06/01)



WWW (SS2001) - Server-Side Technologien

26

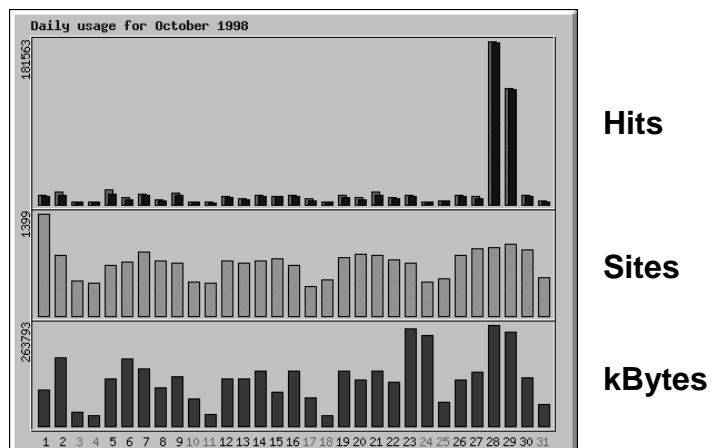
Benutzung pro Tag (03/99)



WWW (SS2001) - Server-Side Technologien

27

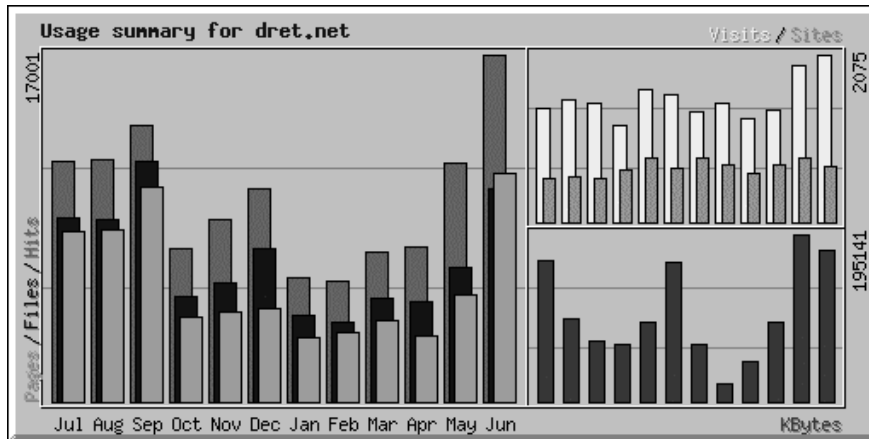
Benutzung pro Tag (10/98)



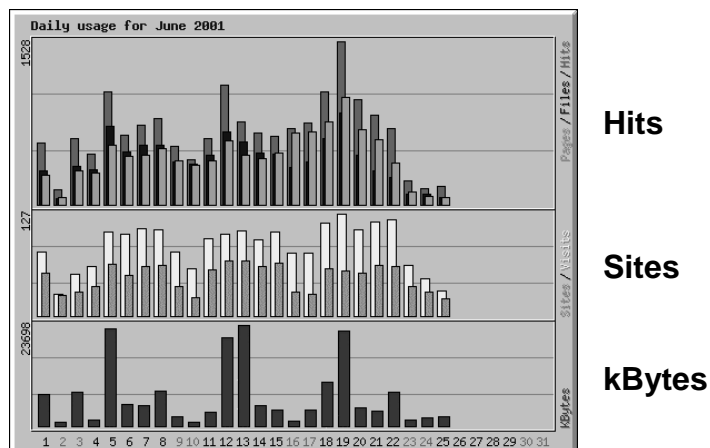
WWW (SS2001) - Server-Side Technologien

28

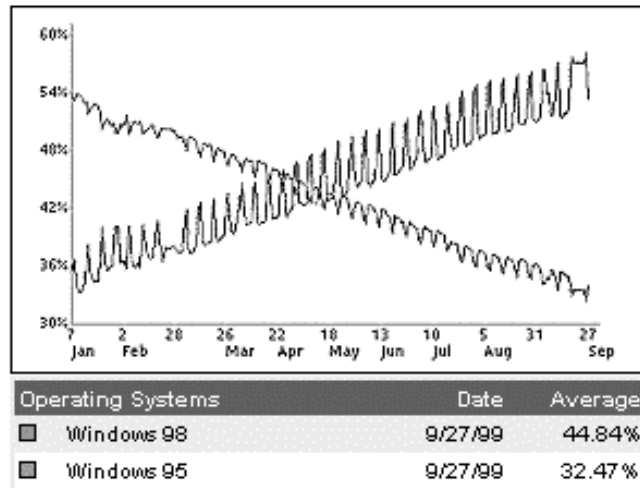
Neue Webalizer-Version



Benutzung pro Tag (06/01)



Windows 95 vs. Windows 98



WWW (SS2001) - Server-Side Technologien

31

Realisierungsmöglichkeiten

- viele Varianten für einzelne Probleme
- Abschätzung nach verschiedenen Kriterien
 - Realisierungsaufwand
 - Wartungsaufwand
 - Benutzbarkeit für den Provider
 - Benutzbarkeit für Anwender
 - Abhängigkeit von speziellen Technologien
 - Stabilität in einem schnell ändernden Umfeld
- fast alles wurde schon einmal gemacht
 - Evaluation bestehender Lösungen
 - Lernmöglichkeiten und Einsparungen

WWW (SS2001) - Server-Side Technologien

32

Einfügen des Änderungsdatums

- Frage des Designs der Web-Site
- 1. JavaScript
 - Verwendung eines <SCRIPT> auf der Seite
- 2. Server-Side Includes
 - Seite wird beim Server geparkt (Performance!)
- 3. Server-Side Scripting (ASP oder JSP)
 - vergleichbar der SSI-Lösung
- 4. Generierung über CGI
 - z.B. Datumsfeld eines Datenbankeintrages
- 5. Generieren der Web-Site
 - *Revision Control System (RCS)* oder ähnliches
- Design-Entscheidung je nach Anwendung

Zusammenfassung

- Web-Server als Komponente
 - Trend zu Web-based Applications
 - *Application Service Providers (ASP)*
 - Integration in immer mehr Softwareprodukte
- Anbindung über Interfaces
 - sehr locker und mit Einschränkungen (CGI)
 - besser und mit mehr Bindungen (SAPI)
 - Scripting-Sprachen als Alternativen (PHP, ASP, JSP)
- Betrieb eines HTTP Servers
 - offen für alle Zugriffe weltweit
 - achtsam Konfigurieren und Beobachten