

**PPS-Design einer eigenen WWW-Homepage
WS 2002/2003**

Praxis III - 06.12.2002:
CGI Scripts und Forms

David Hausheer (hausheer@tik.ee.ethz.ch)

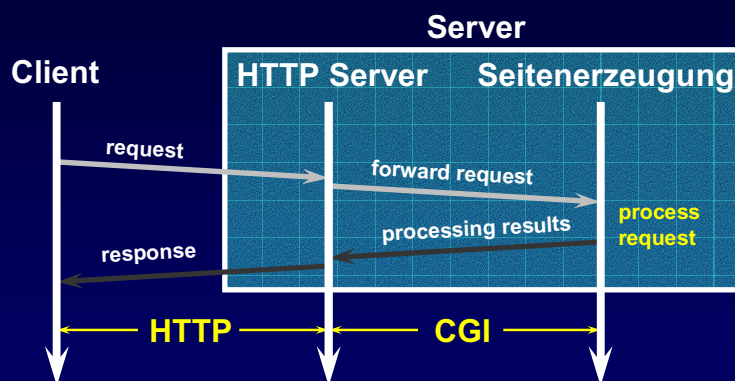
Anforderungen an eine Webseite

1. Navigationsmechanismen
 - Mehrere HTML Seiten sollen erstellt werden
 - Frames, Tabellen, Scripts
2. Einheitliche Gestaltung
 - Style-Sheets (CSS)
3. Dynamische Elemente
 - Seite oder Teile von HTML werden dynamisch erstellt (DHTML)
 - CGI-Scripts (Perl, PHP, shell, ...)
4. Animation
 - JavaScript, Java-Applet
 - evtl. Macromedia, Flash, ...

Common Gateway Interface (CGI) Scripts

- Ein **CGI Script** ist ein Program welches
 - auf dem *entfernten Webserver abgelegt* ist und
 - bei entsprechender *Benutzeranfrage ausgeführt* wird
- Ein CGI Script wird in einer Programmiersprache geschrieben welche entweder:
 - **Compiliert** auf dem Server liegt (C, C++, Ada)
 - **Interpretiert** wird durch einen Interpreter auf dem Server (perl, PHP, shell script)
- Webserver und CGI-Script kommunizieren durch
 - **Environment-Variablen** (vor dem Aufruf des Scripts gesetzt)
 - **Standard-Eingabe** (stdin)
 - **Standard-Ausgabe** (stdout)

Dynamische Seitengenerierung



Einfaches CGI Beispiel

Simple.cgi:

```
#!/bin/sh
echo "Content-Type: text/html"
echo
echo "<!DOCTYPE HTML PUBLIC "-//IETF">"
echo "This is a simple CGI script"
echo "<br>"
id
echo "<br>"
date
echo "<br>" $SERVER_SOFTWARE
echo "<br>" $REMOTE_HOST
```

Output:

http://www.tik.ee.ethz.ch/tik/education/lectures/PPS/web/WS02_03/cgi/Simple.cgi

```
This is a simple CGI script
uid=9905(httpd) gid=901(employee)
Thu Dec  4 18:46:36 MET 2002
Apache/1.3.26 Ben-SSL/1.23 (Unix)
vpn-global-012-dhcp.ethz.ch
```

CGI Environment Variablen für Script

- Vom Request **unabhängige** Variablen
 - SERVER_SOFTWARE: Name und Version des WWW-Servers
 - SERVER_NAME: der Server-Name (DNS-Name)
 - GATEWAY_INTERFACE: CGI-Version
- Je nach Request **unterschiedlich** gesetzte Variablen
 - SERVER_PROTOCOL: für die Anfrage verwendetes Protokoll
 - SERVER_PORT: für die Anfrage verwendeter Port
 - REMOTE_HOST oder REMOTE_ADDR: Ursprung der Anfrage
- http://www.tik.ee.ethz.ch/tik/education/lectures/PPS/web/WS02_03/cgi/Simple2.cgi

Installieren von CGI Scripts auf Server

- Wie genau **Konfiguration** von CGI Scripts auszusehen hat, ist Webserver-spezifisch
- Web-Administrator kann bestimmen:
 - ob CGI Scripts überhaupt ausgeführt werden
 - wo diese Scripts liegen müssen
 - (z.B. /home/httpd/cgi-bin/)
 - Welche Endung Scripts haben müssen (.cgi)
- **Apache:**
 - Konfiguration in httpd.conf, srm.conf, access.conf
 - Definition von Scripthandler (in srm.conf), damit Webserver erkennt, was ausführbare Scripts sind:
 - AddHandler cgi-script .cgi
 - Script muss Execution-Bit gesetzt haben und Server muss Read-Permissions haben:
 - `chmod a+rx myscript.cgi`

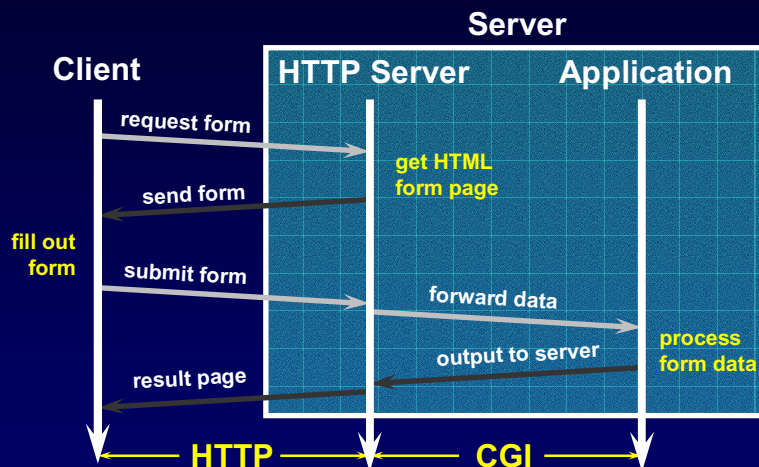
Methoden für Parameterübergabe

- **GET** Methode
 - Falls Form METHOD="GET" verwendet, bekommt CGI Programm den codierten String in der Environment-Variable QUERY_STRING.
 - http://www.tik.ee.ethz.ch/tik/education/lectures/PPS/web/WS02_03/cgi/Simple2.cgi?foo=bar&test=hallo
- **POST** Methode
 - Falls Form METHOD="POST" benutzt, erhält CGI Script die Formdaten über stdin
 - Der Webserver schickt dem CGI kein EOF beim Ende der Daten, CGI Script kann mittels CONTENT_LENGTH bestimmen wieviele Daten zu lesen sind.
 - flexiblere und sicherere Lösung als GET

Resultate von CGI Scripts

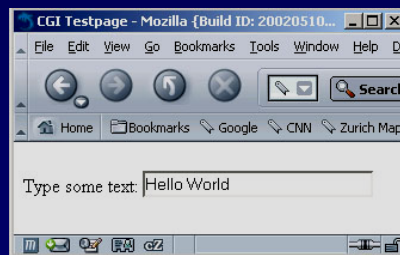
- Kontakt zum Web-Server über **Standard-Ausgabe**
- Alle CGI Resultate müssen mit einem **Header** beginnen, welcher Resultat beschreibt
- Mögliche **Resultate**:
 - HTML Seite
 - "Content-Type: text/html"
 - definiert MIME-Type der Antwort
 - Verweis auf das Ergebnis (URI oder File-Pfad)
 - "Location: http://redirectedsite.com"
 - Status-Meldung:
 - "Status 302"

HTML Form Handling



HTML Form

```
<html> <head> <title>CGI Testpage </title> </head>
<body><form action="http://myserver/test.cgi"
method="POST">
<b>Type some text:
<input type="text" name="textinput" size="30"
maxlength="50"> </p>
</form> </body> </html>
```



CGI Perl Script

```
#!/tiklopt1/bin/perl -w
use CGI qw(:standard);

my $value = param ('textinput');
print header(), start_html("Response"),
p("You typed: ", $value),
end_html();
```

