Web-Programmierung (WPR)

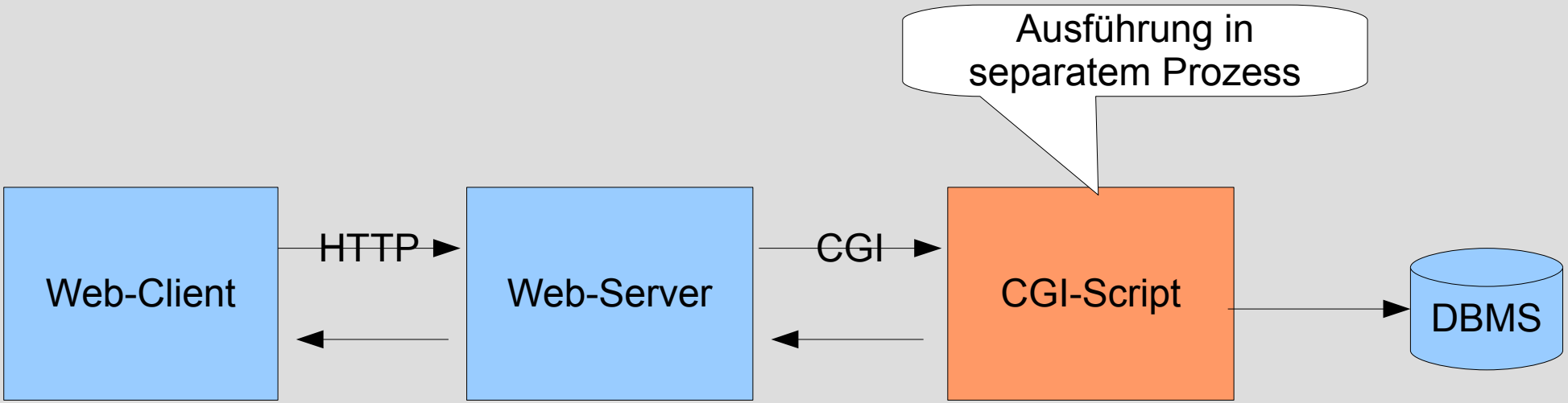
Vorlesung VIII.

Common Gateway Interface(CGI)
&
PHP

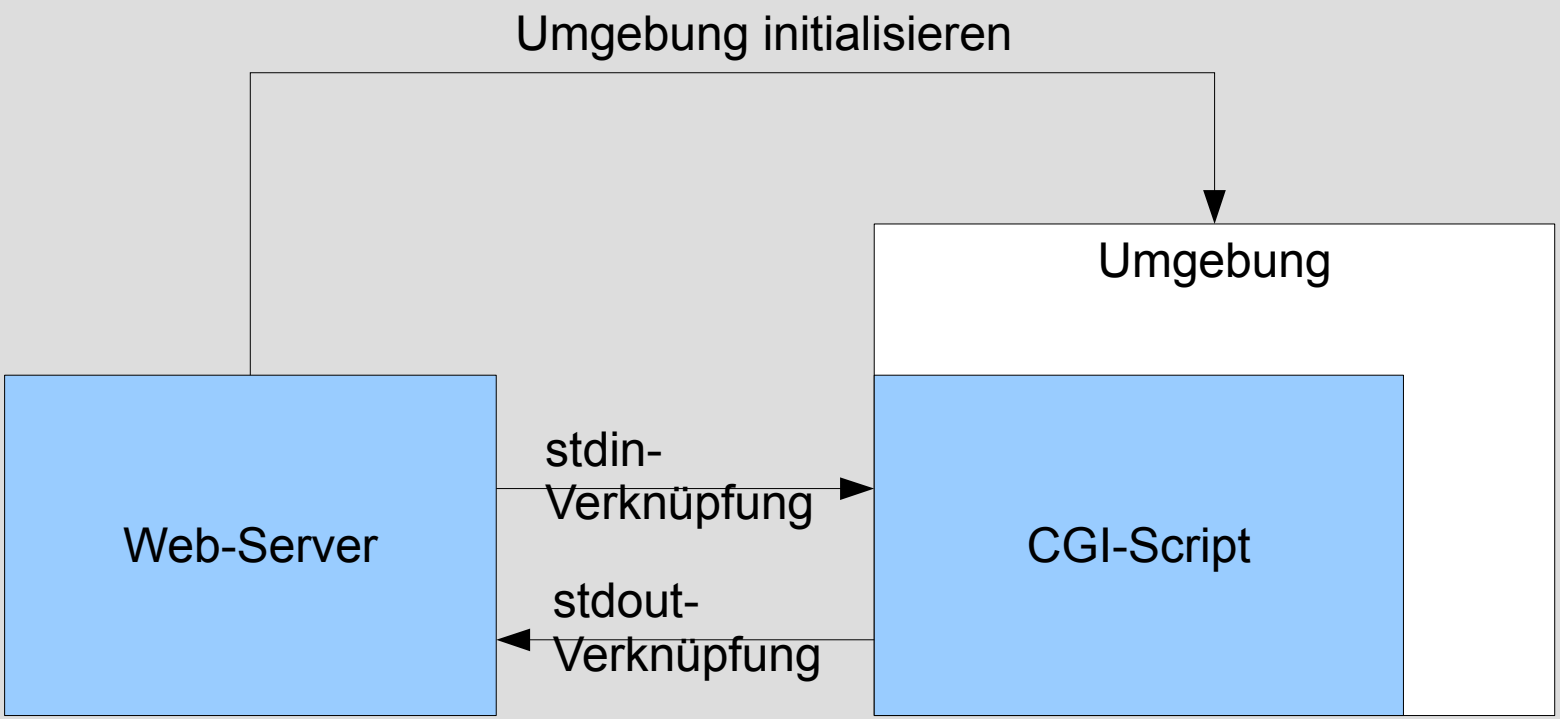
mailto:wpr@gruner.org

12 Common Gateway Interface

- Von allen Webservern unterstützt
- Anzubindende Programme
  => Gateway zu bestehenden Programmen



12 Common Gateway Interface



12 Common Gateway Interface

Sprache	Beschreibung
C/C++	Compiler, plattformabhängig
sh,ksh,csh	Interpreter, UNIX-Umgebung
tcl	Interpreter, UNIX-Umgebung
PHP	Interpreter, plattformabhängig
Visual Basic	Windows-Umgebung

- Bereitstellung (typischerweise)
  cgi-bin

  http://hostname/cgi-bin/*programmname*

12.1 Informationsbeschaffung

- Übertragung von Formular-Feldern

Umgebungsvariable	**Beschreibung**
HTTP_USER_AGENT	Browser des Client
...	
REQUEST_METHOD	HTTP-Methode (GET oder POST)
QUERY_STRING	URL codierte Benutzereingaben

Methode	**Beschreibung**
GET	QUERY_STRING enthält Formularinformationen
POST	Daten können von STDIN gelesen werden. Die Länge wird aus der Umgebungsvariablen CONTENT_LENGHT gelesen.

12.4 Programmiermodell

- CG'I ist unabhängig von der Programmiersprache
- Es gibt kein CGI API
- Programmiermodell
  =Umgebung durch Web-Server
  - Standardeingabe stdin
  - Standardausgabe stdout
  - Umgebungsvariablen
- CGI-Programmierung ist HTTP-Programmierung im Raw-Mode

12.6 Fazit

- Anzahl gleichzeitiger Request ist eingeschränkt (Perl/CGI)
- Für größere Web-Lösungen
  - Nicht genügend performant
  - Nicht genügend skalierbar

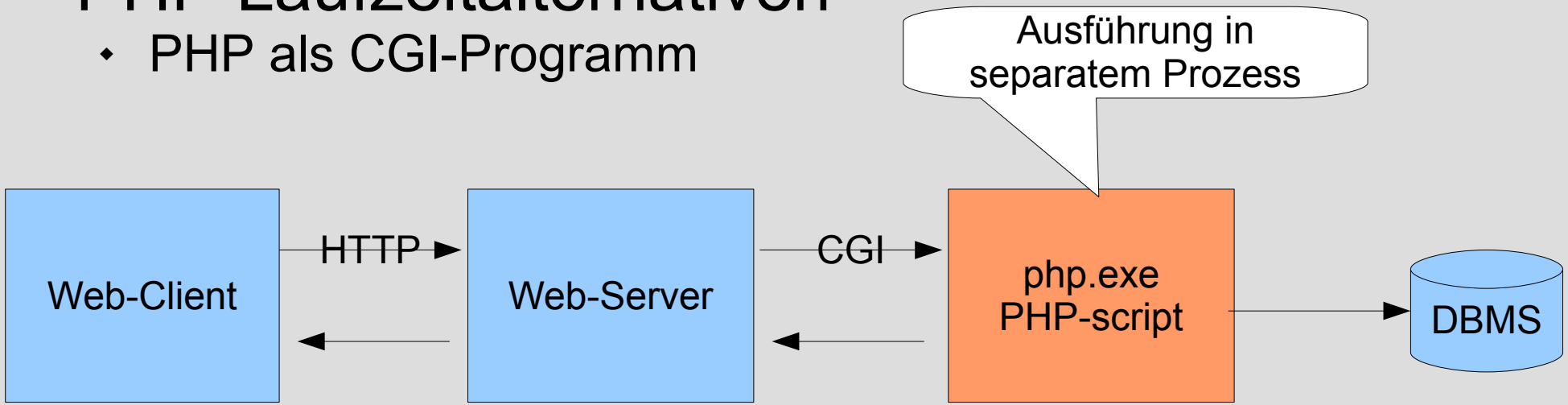
- Größter Vorteil Perl/CGI
  - CGI wird von jedem Web-Server angeboten
  - Interpreter-Impl. existiert für relevante Betriebssysteme
  - Provider Angebot (Perl/CGI) ist groß
  - Lizenzkostenfreie Web-Applikationen realisierbar

13 PHP

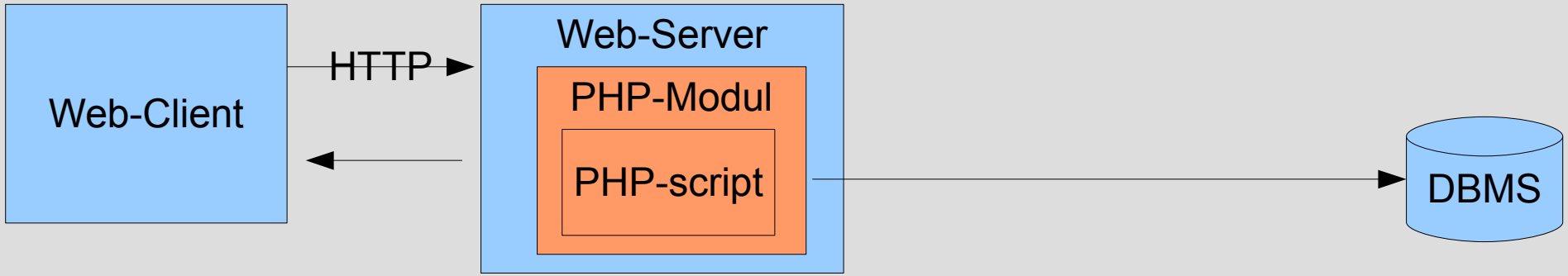
- Serverseitige Web-Plattform
- Skriptsprache mit Perl-ähnlicher Syntax
- Prozedurale Sprache + objektor. Extensions
- Umfangreiche Bibliothek vordef. Funktionen
- Umfassende Datenbankunterstützung
- Open Source

13 PHP

- PHP Laufzeitalternativen
  - PHP als CGI-Programm



  - PHP als Server-Modul



13.1 Grundlagen

- **PHP-Merkmale**
  - Einfachheit
  - Vertrautheit
  - Effizienz
  - Portablilität
  - Beschränkung auf das Notwendige

- **Implementierung**
  - Keine separaten Programmdateiein in PHP
  - Nur dynamische Teile in PHP implementiert

```
<?php ..... ?>
```

13.1 Grundlagen

- Dateiname
  - Endung: **`.php`**
  - Definition erfolgt beim WebServer
    z.B. Apache http-Server (httpd.conf)
    `AddType application/x-httpd-php .php`

- Kommentare
  - Einzeilig: `//` oder #
  - Mehrzeilig: `/* ..... */`

13.1 Grundlagen

- Tagging
  - short tagging

```
<% echo „Brave new PHP World; %/>
```

  - script tagging

```
<script language="php">
  echo „Brave new PHP World;
</script>
```

  - Ausgabeanweisung (siehe JSP)

```
<%= „Brave new PHP World“ %/>
```

- Seitenabruf
  - PHP-Dateien werden interpretiert
  - Ablage wie HTML-Seiten

13.1 Grundlagen

- Konfiguration
  - Zentrale Konfigurationsdatei: php.ini

13.2 Sprachüberblick

- Datentypen
  - integer, double, string, boolean
  - array, object
- Typzuweisung
  - Typzuweisung bei Wertzuweisung
  - Expliziter Typecast ist möglich (Cast Operator)

13.2 Sprachüberblick

- Operatoren
  - Arithmetische Operatoren (+,-,*,/,%)
  - Zuweisungsoperatoren (=,+=,-=,..)
  - Vergleichsoperatoren (>,>=,<=,==,!=)
  - Stringkonkatentation (.)
  - Logische Operatoren (&&, ||, xor, ...)
  - UnäreOperatoren (+,-,++,--,!,(typ)
  - Ternäre Bedingsoperator (?:)

13.2 Sprachüberblick

- Variablen
  - Bezeichner beginnen mit $
  - Stringliterale mit „“,
    => Variablensubstitution wird durchgeführt
  - Stringliterale mit ' (Hochkomma)
    => keine Substitution

13.2 Sprachüberblick

- Arrays
  - Gewöhnliche Arrays

```
$listCommonArray=array(13,4,5);
```

// Anhängen eines Elements

```
$listCommonArray[]=18;
echo „drittes Element: $listCommonArray[2]\n“;
```

- Assoziativ Arrays

```
$listAssociativeArray=(
    13=>“eins“,
    19=>“58“);
echo „zweites Element: $listAssociate[19] der
  Liste“;
```

13.2 Sprachüberblick

- Kontrollstrukturen
  - ```
    if(boolscher Ausdruck){
       ...
    }elseif (boolscher Ausdruck){
       ...
    }else{
       ...
    }
    ```
  - ```
    switch(Skalaranweisung){
    case literal: ... ;
    default: ... ;
    }
    ```

13.2 Sprachüberblick

- Kontrollstrukturen (Schleifen)
  - `while(`***`boolscher Ausdruck`***`){`
    `Anweisungen..`
    `}`
  - `do{`
    `Anweisungen...`
    `}while(`***`boolscher Ausdruck`***`);`
  - `for(Anweisung1;`***`boolscher Ausdruck;`***`Anweisung3)`
    `{`
    `Anweisungen...`
    `}`
  - `foreach(array as arrayelement)`
    `{... }`

13.2 Sprachüberblick

- Funktionen
  Schlüsselwort **`function`**

```
function quadrat($parameter)
{
$result=$parameter * $ parameter;
return $result;
}
```

- **Globale Variablen Deklaration**

```
global $g_my_reference;
```

13.2 Sprachüberblick

- Klassen (Funktionen und Attribute)
-
```php
<?php
class Cart {
    var $items; // Items in our shopping cart
    // Add $num articles of $artnr to the cart
    function add_item($artnr, $num) {
        $this->items[$artnr] += $num;
    }
    // Take $num articles of $artnr out of the cart
    function remove_item($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } else {
            return false;
        }
    }
}
?>
```

13.2 Sprachüberblick

- Reguläre Ausdrücke
- PHP stellt dafür Funktionen zur Verfügung z.B.:

Funktion	Beschreibung
preg_match()	Test einer Zeichenfolge
preg_match_all()	Test einer Zeichenfolge
preg_replace()	Modifikation einer Zeichenfolge
preg_split()	Aufteilung einer Zeichenfolge in Token

13.3 Informationsbeschaffung

- PHP Laufzeitumgebung
  => globale Assoziative Arrays

Variablenname	Inhalt
$_GET	Formulardaten eines GET-Request
$_POST	Formulardaten eines POST-Request
$_SERVER	Server-Infos + selektive HTTP-Header-Infos
$_ENV	Zugriff auf Umgebungsvariablen
$_COOKIE	Mit Cookie übertragenes Array
$_SESSION	Sessionweit verfügbare Daten

13.4 Responsegenerierung

- Standardfunktion `header()`

```
<?
   if(!header_sent()){
      header(„Content-Type: text/plain“);
   }
?>
Dies ist ein geöhnlicher Text
```

13.4 Response-Generierung

- Server-Redirection

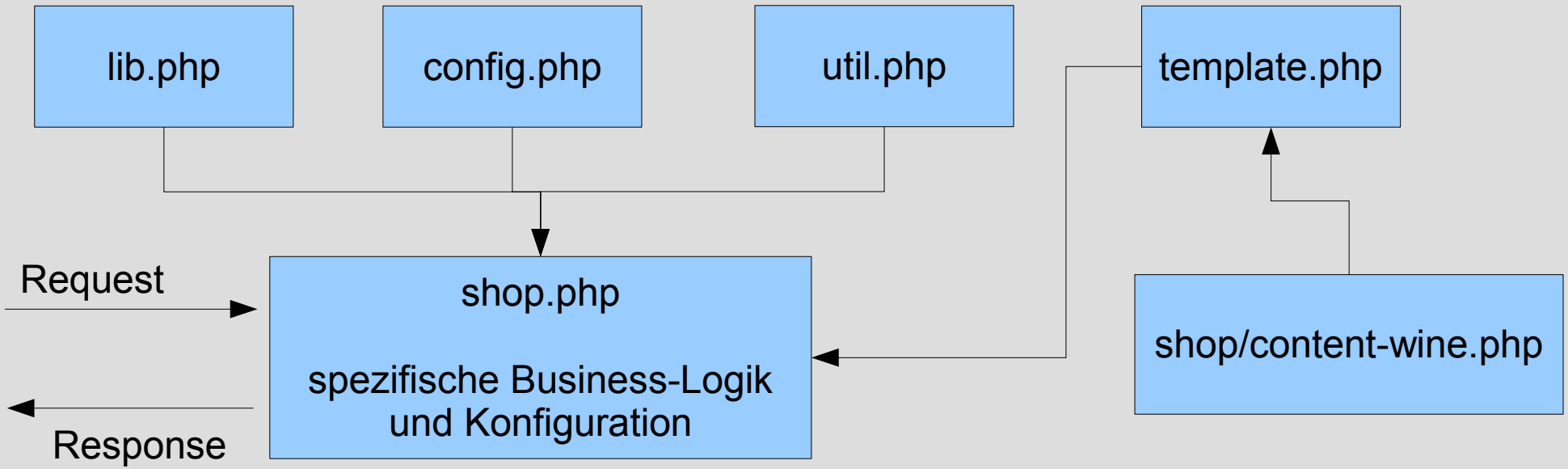
```
<?
   if($_GET['user']==“”||
      $_GET['password'==“”])
       $response=“error.html“;
    else
       $response=“confirmation.html“;
    if(!header_sent(){
      header(„Location: $response“);
      exit;
    }
?>
```

13.5 Programmiermodell

- Dokumentenzentrierte Web-Plattform
  s.a. CGI = Programmorientiert
- PHP-Dokumente
  - Primär HTML-Dokumente
  - Incl. Programmcode
- PHP-Code wird interpretiert
- Einheitliches API für einfache Anwendung

13.9 Modularisierung

- Modularisierung von Programmcode und HTML-Dokumenten
  - include(„header.html“);
  - include_once(„util.php“);



13.10 Fazit

- PHP ist einfach und leicht erlernbar
- Sprachumfang klein (da Web-Zentriert)
- Zuverlässige Integration
  in Apache http-Server
- Umfangreiche Bibliothek
  - Schnittstellen für XML, Datenbanken,...