

Web-Programmierung (WPR)

Vorlesung V.

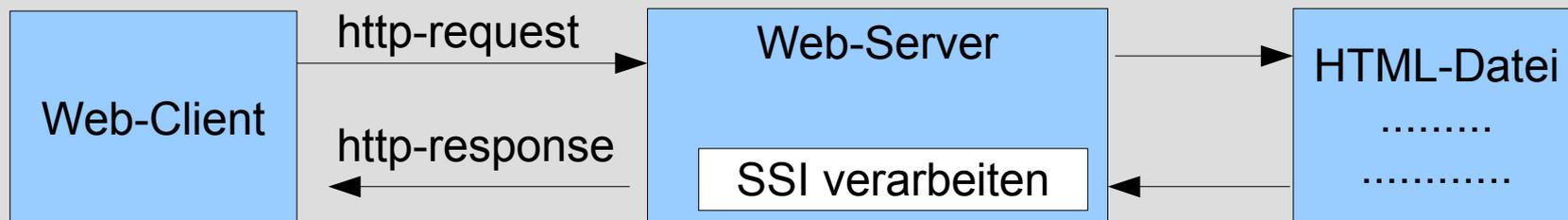
Grundlagen Server

Manfred Gruner

<mailto:wpr@gruner.org>

10.1 Server Side Includes

- Server Side Includes
 - Einfügen von dynamischen Fragmenten in statische HTML-Seiten
 - Alternative u. Ergänzung zur serverseitigen Programmen,



10.1 Server Side Includes

- Einbinden von Informationen
 - Umgebungsvariablen
+ spezielle SSI-Variablen
 - Dateiinhalte
 - Dateigröße und Modifikationsdatum
 - Ausgaben externer Programme

10.1 Server Side Includes

- Nachteile
 - Parsen beeinträchtigt Server-Performance
 - Potenzielles Sicherheitsrisiko
 - Nicht alle http-Server unterstützen SSI
- Hinweis
 - Namensgebung:
Dateien mit SSI ==> *.shtml

10.1 Server Side Includes

SSI-Anweisung

`<!--#SSI-Kommando`

`paramtername="parameterwert" -->`

Umgebungsvariablen

SSI-Variable

DOCUMENT_NAME

DOCUMENT_URI

QUERY_STRING_UNESCAPED

DATE_LOCAL

DATE_GMT

LAST_MODIFIED

Beschreibung

Dateiname

URL des Dokumentes

Query-String

Date/Time der lokalen Zeitzone

Date/Time GMT(Greenwich Mean Time)

Datum der letzten Modifikation

10.1 Server Side Includes

- Beispiel Umgebungsvariablen

```
<!--#echo var="REMOTE_HOST" -->
```

- Beispiel Einbinden von Externen Dateien

```
<!--#include file="power.html" -->
```

- Beispiel Datei-Informationen

```
<!--#fsize file="/public/pr.tar.z" -->
```

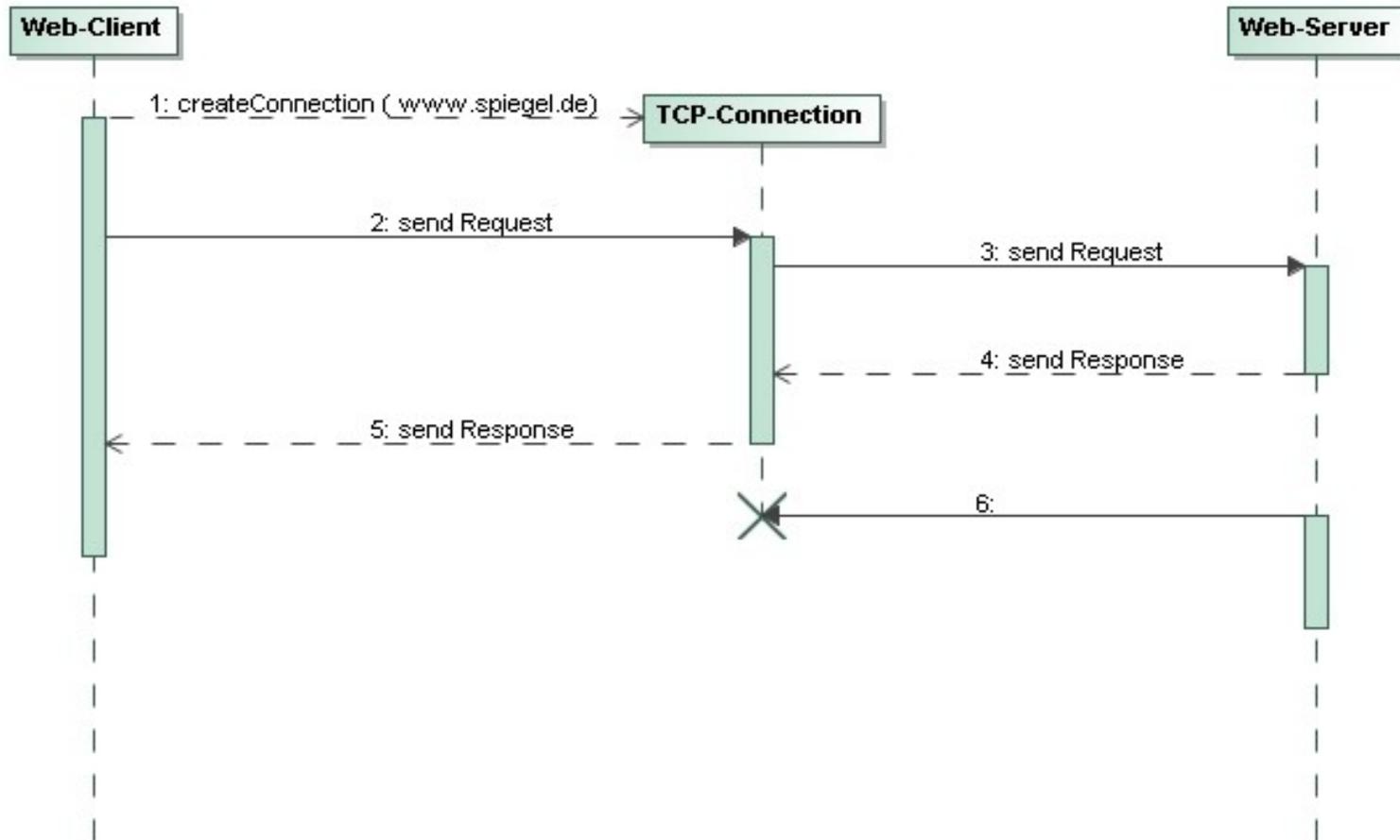
10.1 Server Side Includes

- Ausführen externer Programme
==> stdout wird in Datei-Umgelenkt

```
<!--#exec cmd="/usr/bin/ls  
-l $ftp_site/public/files" -->
```

10.2 HTTP-Kommunikation

interaction http request / response [http in action]



10.2 HTTP-Kommunikation

- Messages – HTTP-Request

```
Methode Request-URL HTTP-Version  
[Header]  
[Datenbereich]
```

```
GET /index.html HTTP/1.0  
Accept: image/gif, image/jpeg, */*  
Accept-charset: iso-8859-1, *, utf-8  
Accept-encoding: gzip  
Referer: http://www.google.de/index.html  
User-Agent: Mozilla/4.51 [en]
```

10.2 HTTP-Kommunikation

- Messages – HTTP-Response

```
HTTP-Version Statuscode Text  
[Header]  
[Datenbereich]
```

```
HTTP/1.1 200 OK  
Date: Sat, 11 Jan 2003 13:40:55 GMT  
Server: Apache/1.3.24 (Win32) PHP/4.3.0  
Content-Type: text/html  
<html>  
  ...  
</html>
```

10.2 HTTP-Kommunikation

HTTP-Methoden

Methode	Bedeutung
GET	Anforderung einer Resource
HEAD	Anforderung von Informationen über eine Resource
POST	Übertragung von Daten zum Server. Daten werden im Datenbereich übertragen

10.2 HTTP-Kommunikation

HTTP Statuscodes

Code	Status	Bedeutung
200	Ok	Aktion erfolgreich durchgeführt
204	No Content	Aktion erfolgreich durchgeführt, aber keine Informationen
302	Found	URL der Resource wurde geändert, der aktuelle URL kann dem Header entnommen werden
304	Not Modified	Statuscode auf If- Modified-Since Request
401	Unauthorized	Resource benötigt zum Zugriff Authentifizierung
403	Forbidden	Resource ist vor Zugriff geschützt (z.B. Verzeichnisse)
404	Not Found	Resource ist nicht verfügbar (existiert nicht)
405	Method not allowed	Request-Methode ist für diese Resource nicht zulässig
500	Internal Server Error	Beim Ausführen eines serverseitigen Programms ist ein Fehler aufgetreten

10.2 HTTP-Kommunikation

Allgemeine Header

Connection	Keep-Alive Close
Date	Zeitpunkt der Messageerstellung

Request-Header

Authorization	Enthält Identität des Benutzers
Refererrerr	URL des Dokumentes aus dem Request erzeugt wurde
User-Agent	Browser-Kennung
If-Modified-Since	Response liefert entweder Resource oder Status „Not Modified“

10.2 HTTP-Kommunikation

Response-Header

Location	URL den der Browser anstelle des ursprünglichen anfordern soll
Server	Server-Kennung
Expires	Verfallsdatum der Resource
Cache-Control	Beeinflussung Cache Verhalten
WWW-Authenticate	Bei Status 401 => gewünschte Authentifizierungsverfahren

Objekt-Header

Content-Length	Länge des Objektes in Bytes
Content-Type	MIME-Type

10.2 HTTP-Kommunikation

- Authentifizierung

- 1.HTTP Basic

Username und Password ==> übermittelt

- 2.HTTP Digest

Username und digest ==> übermittelt

- 3.form based Authentication

Username und password werden über HTML-Formular übertragen (möglichst https)

- 4.HTTPS client authentication

10.2 HTTP-Kommunikation

- Beispiel HTTP-Basic Authentifizierung

Seite 225

10.2 HTTP-Kommunikation

- **Caching**

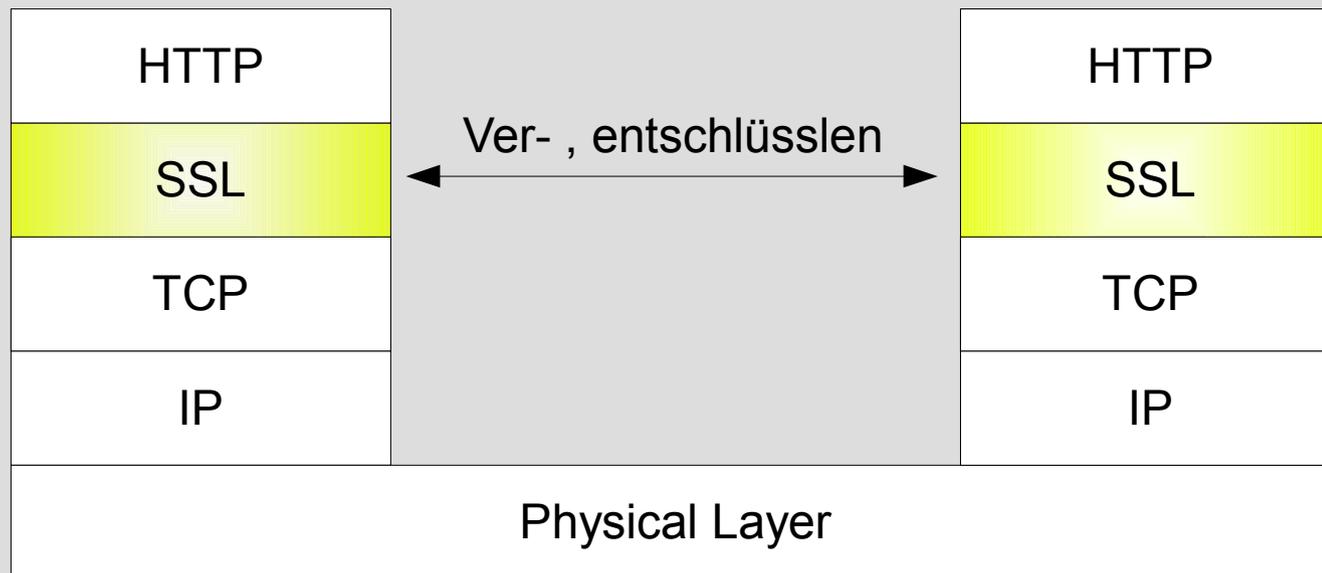
```
<meta http-equiv="Expires"  
content="Sat, 03.05.2008 17:35:00  
GMT">
```

- **Cache-Control Header**

- no-cache (Überprüfung vor Auslieferung)
- no-store (Zwischenpufferung nicht erlaubt)
- max-age (Legt Verfallszeitpunkt fest- siehe auch „expires“; Angabe in sec.)

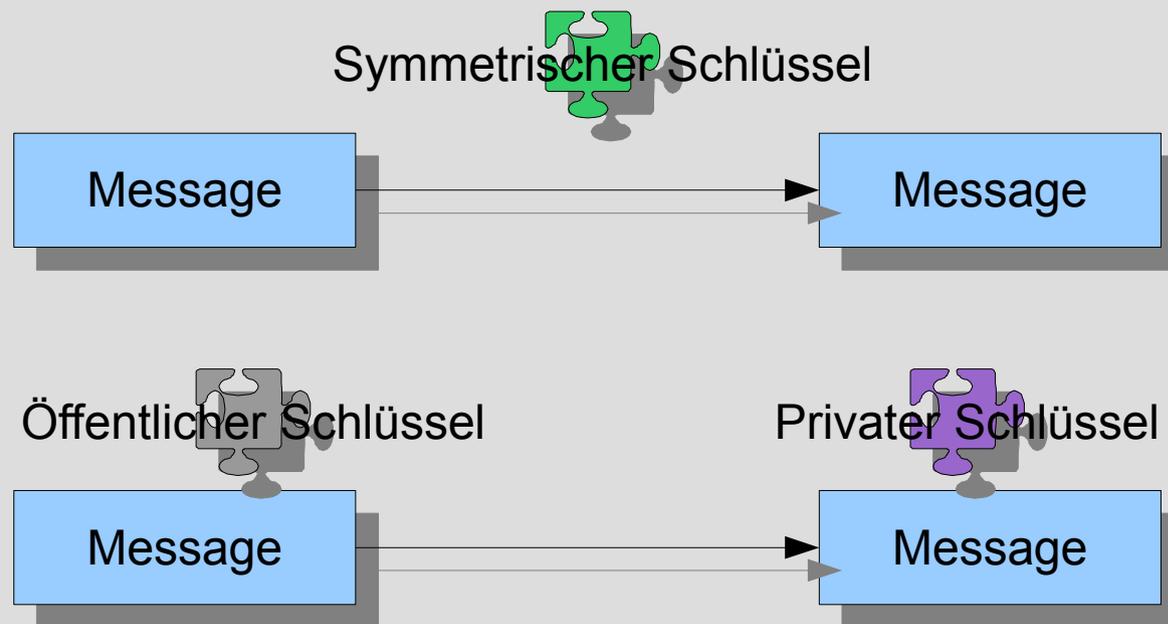
10.3 Sichere Datenübertragung

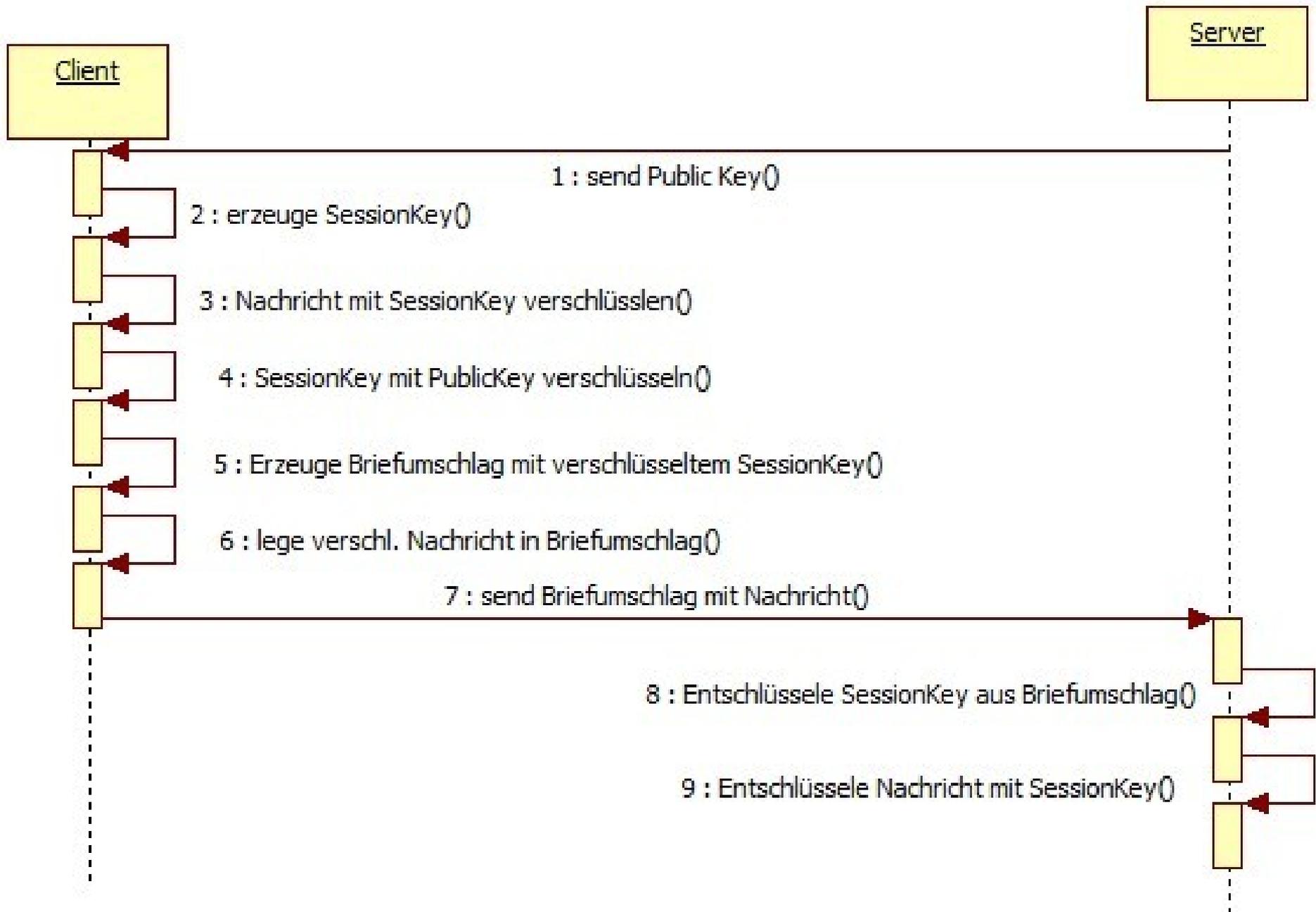
- HTTPS == HTTP over SSL
 - Secure Socket Layer (SSL)



10.3 Sichere Datenübertragung

- Verschlüsselungsverfahren
 - Symmetrische Verschlüsselung
 - Asymmetrische Verschlüsselung





10.4 Session Management

- Problematik:
Eine Aktion über mehrere HTML-Seiten
- Lösungsmöglichkeiten
 - Hidden Fields
 - URL Rewriting
 - Cookies