
7. De applicatielaag

Applicatielaag introdactie

- standaards t.b.v. ondersteuning van applicaties, b.v.
 - Name en Directory servers (b.v DNS en X.500)
 - Netwerkmanagement (SNMP)
- Standaard toepassingen, b.v.
 - Email (SMTP en X.400)
 - WWW
 - multimedia
 - File Transfer (b.v. FTP)

Name services

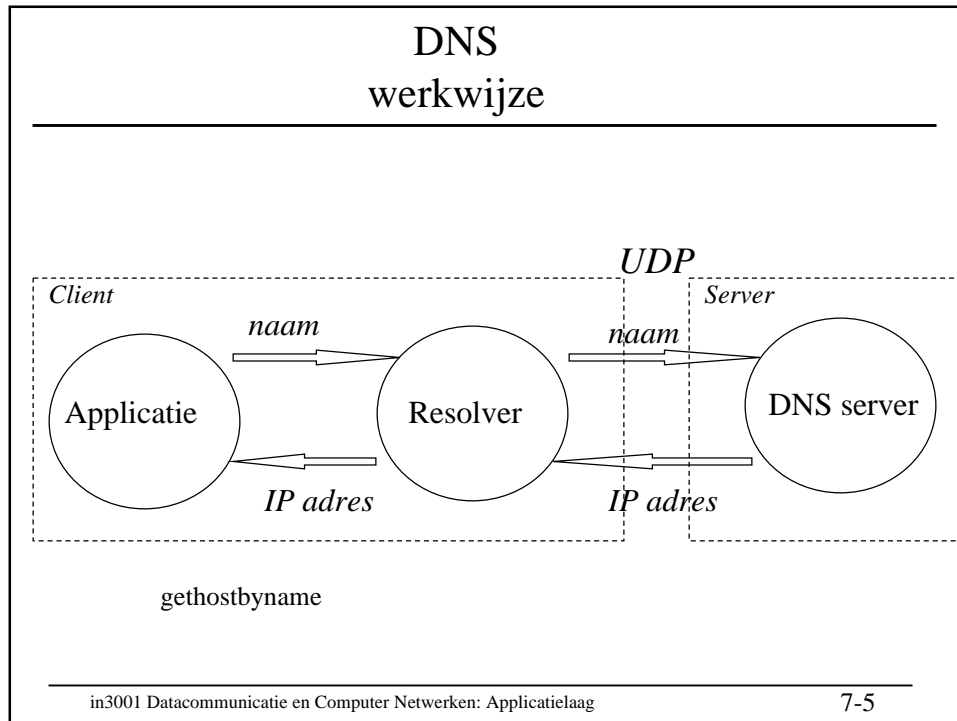
- Name service = Directory service
- Basis functie:
Het leveren van een adres bij een gegeven naam

Standaards:

- CCITT: X.500 Directory Services
- Internet: DNS (Domain name System)
- Internet: LDAP (Lightweight Directory Access Protocol)

DNS (Domain Naming System)

- DNS (vanaf 1987) is name service van Internet
- beschreven in RFC1034 en RFC 1035
- Voor 1987 één centrale masterfile met host-namen en adressen, te downloaden door iedere host.
problemen:
 - niet scalable
 - te centraal
- DNS bevat voornamelijk host-namen en IP-adressen
- in principe ieder object mogelijk



- ### DNS domeinen
-
- DNS hiërarchisch gepartioneerd
 - domeinnaam: *labels* gescheiden door '.'
 - hoogste level rechts, voorbeeld: **dutikos.twi.tudelft.nl**
 - domeinnamen, toplevel in twee soorten
 - generiek (voornamelijk in US), b.v. com, edu, gov, ... etc.
nu ook: biz, info, pro, aero, coop museum
 - land (buiten de US), (volgens ISO3166)
 - domeinnamen zijn: absoluut (eindigt met .) of relatief
 - DNS servers kennen geen relatieve namen. Client-software vult eventueel namen aan
 - binnen een domein mag '*domein autoriteit*' zelf de subdomein-indeling en namen kiezen
 - max 63 tekens per component, 255 per domein
-
- in3001 Datacommunicatie en Computer Netwerken: Applicatielaag 7-6

DNS database

- Per domain een set Resource Records (RR's)
- format: domain-name, time-to-live, type, class, value
- Resource Record (RR) kent verschillende recordtypen voor verschillende object typen, b.v.

type	betekenis
A	computeradres
NS	gezaghebbende name server
CNAME	alias naam
MX	mail host

etc...

- Root server adressen in systeem configuratie file. Deze worden in cache geplaatst tijdens het starten van DNS

DNS queries

queries ten behoeve van

- eenvoudige host-naam resolutie
b.v. ftp geeft host-naam, wenst IP-adres
- vinden van een E-mail server
(Er is één E-mail server per domein)
- omgekeerde resolutie
(zoekt host-naam bij IP-adres)
- informatie over de host
(gevaarlijk ?)
- vinden van well-known services, b.v. telnet, ftp

DNS name servers (1)

- scaling probleem opgelost door
 - partitionering
 - replication en caching in de buurt waar gegevens nodig zijn
- DNS name space opgedeeld in zones, per zone te registreren:
 - gegevens voor namen in een domein, uitgezonderd de subdomeinen met eigen beheer
 - namen en adressen van gezaghebbende (authoritative) services
 - namen en adressen van servers in subzones
 - zone management parameters

DNS name servers (2)

- Iedere zone bevat:
 - Eén primaire server heeft (authoritative) informatie in database)
 - Eén of meer secundaire servers, (krijgen informatie van primaire server)
 - (secundaire servers kunnen zich ook buiten de zone bevinden)
- caching mag door iedere server gegevens krijgen een *time-to-live* waarde mee

DNS

navigatie en query processing

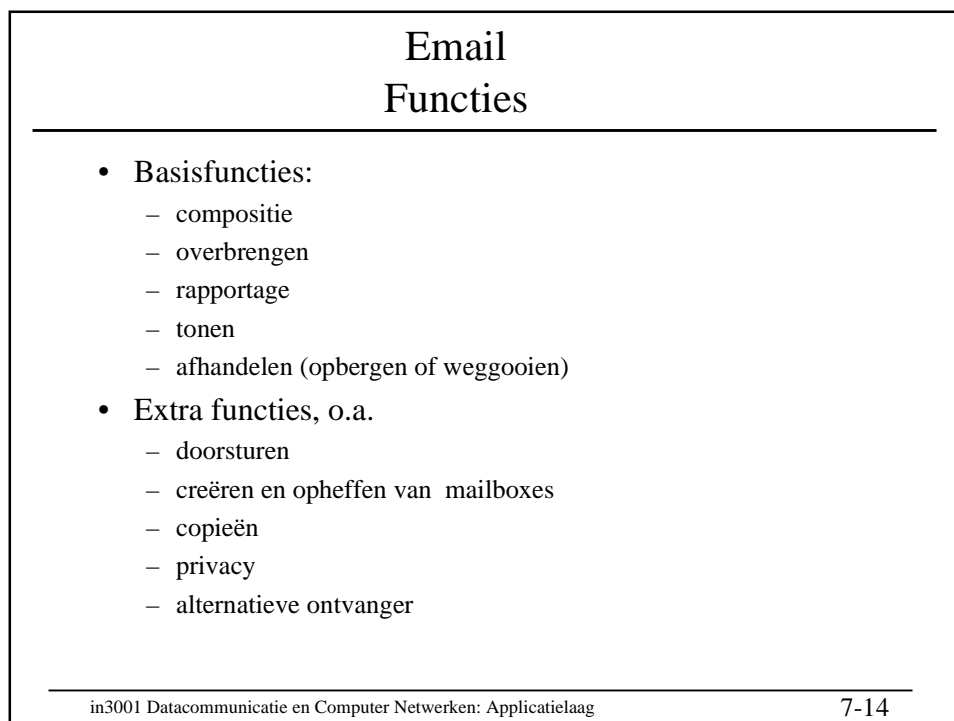
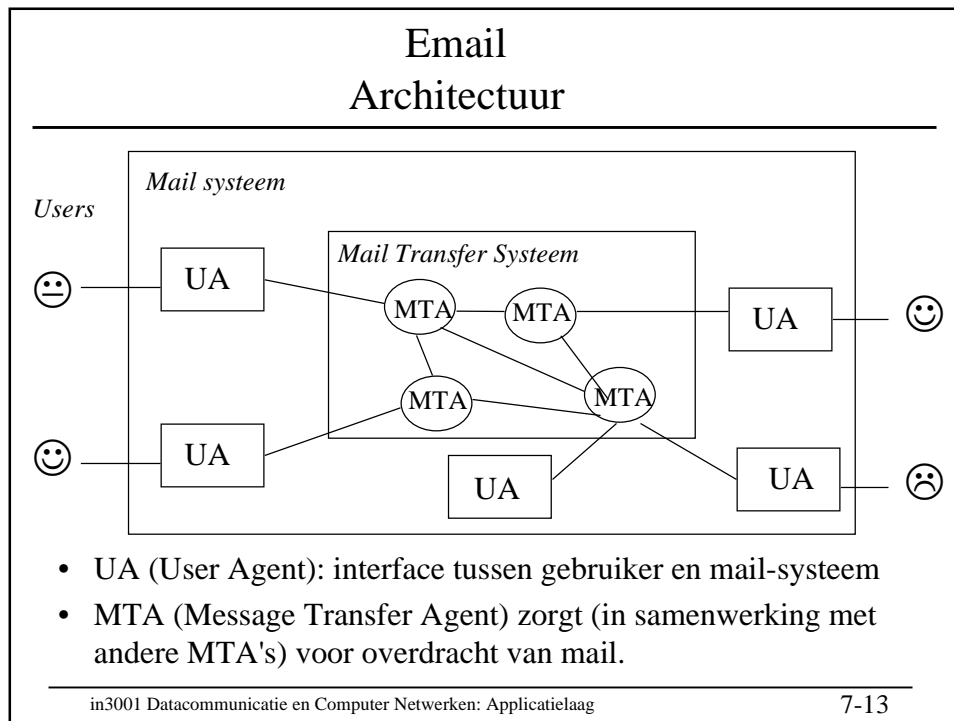
- navigatie en query processing d.m.v resolver en server
- DNS-architectuur voorziet in
 - recursieve navigatie
 - iteratieve navigatie
- resolver kiest methode (maar: niet alle servers kennen de recursieve techniek)
- meer queries in 1 query-bericht toegestaan

Email

Standaards

- CCITT: X.400 (versie 1984)
- OSI : MOTIS
- CCITT: X.400 (versie 1988)
MOTIS en X.400 zijn nagenoeg hetzelfde
- Internet: RFC822 (+ rfc821) = defacto standaard

Internet en X.400 gebruiken verschillende soorten adressen.



Email

Componenten van een brief

- **Envelop**
 - bevat informatie ten behoeve van MTA's: i.h.a. informatie nodig voor het op de juiste wijze overbrengen van de brief
- **Inhoud**, bestaat uit:
 - briefhoofd (vergelijk normale brief)
 - het eigenlijke bericht (de 'message body')

Email

User Agent

- User Agent (mail reader) is interface tussen gebruiker en mail-systeem
- Voor verzending nodig:
 - het bericht (b.v. aanmaken met editor)
 - E-mail adress (Bij Internet in DNS-formaat, bij X.400 anders)
 - parameters, b.v. prioriteit, security, ...
- Lezen en verwerken van mail:
 - overzicht nodig
 - per brief te beslissen over acties, b.v. forward, reply, delete, undelete, save, print

Email RFC 822

Enige details:

- header bevat zowel envelop als briefhoofd
 - Ieder headerveld bestaat uit één regel, met:
<veldnaam> : <waarde>
 - velden t.b.v. MTA, b.v.
to : <bestemming>
from: <creator>
sender: <afzender>
 - velden t.b.v UA, b.v.
In-Reply-To: <antwoord op brief ...>, **Subject**: <het onderwerp>
- body
 - iedere inhoud mag (dus niet alleen printbare tekst)

Email MIME (1)

- MIME (Multipurpose Internet Mail Extensions), structureert de message body
- maakt het mogelijk verschillende delen van de body verschillend te behandelen. b.v.
 - ander alfabet gebruiken
 - beeld verwerken
 - geluid verwerken
- zendende en ontvangende programma's moeten hiervoor worden aangepast (uitgebreid).

Email MIME (2)

Nieuwe header velden t.b.v. MIME, o.a.

- Content-Type: b.v.
 - text/plain
 - image/gif
 - video/mpeg
- Content-Transfer-Encoding, b.v.
 - ASCII (max 1000 tekens per regel)
 - base64 encoding : 24 bits in 4 groepen van 6, een groep van 6 bits wordt voorgesteld door een alfabetisch teken, een cijfer of + en /
 - quoted-printable encoding
(‘unprintable’ wordt =XX ; ‘=’ wordt =3D)

Email MIME (3)

Zeven typen: b.v.

- | | |
|----------------------------|----------------------------------|
| – text/html | Web page |
| – image/gif | plaatje |
| – video/mpeg | filmpje |
| – application/octet-stream | willekeurige reeks bytes |
| – application/postscript | postscript 'programma' |
| – message external body | body elders te halen |
| – multipart/mixed | verschillende onderdelen |
| – multipart/alternative | b.v. verschillende talen |
| – multipart/parallel | b.v. video clip |
| – multipart/digest | reeks berichten verzameld in één |

Email transfer

- Brieven worden doorgegeven van MTA naar MTA (store and forward)
- Steeds transportverbinding van MTA naar MTA nodig
- In eenvoudigste geval (Internet): transportverbinding van bron-MTA naar doel-MTA
- Internet gebruikt SMTP-protocol (boven TCP, server luistert op poort 25)
- Bij verschillende mailsystemen b.v. Internet en X.400 gateways nodig (op applicatie niveau)

Email Van server naar gebruiker

- Als gebruikersmachine geen mailserver bevat is er een protocol nodig voor communicatie tussen gebruiker en mailbox in mailserver.
- Verschillende standaards:
 - POP3 : Haalt mail uit mailbox, plaatst deze in machine van gebruiker.
 - IMAP : mail blijft in mailbox: gebruiker kan deze vanaf verschillende machines raadplegen en manipuleren
- andere faciliteiten:
 - filters
 - vacation daemon

Belangrijke verschillen POP3 en IMAP

	POP3	IMAP
Waar Email bewaren?	op gebruikers PC	op Server
Hoe Email lezen?	offline	online
verbindingstijd	kort	lang
gebruik server resources	weinig	veel
meerdere mailboxes?	nee	ja
voor mobiele gebruikers?	nee	ja
kiezen wat te downloaden?	nee	ja
eenvoudige imlementatie?	ja	nee
veel gebruikt?	ja	groeiend

WWW World Wide Web

- Web bestaat uit door links met elkaar verbonden documenten, verspreid over Internetmachines (servers)
- documenten te bekijken d.m.v. browsers(clients):
 - grafisch b.v. Mosaic, Netscape, Internet Explorer
 - text-only: b.v. Lynx
- documenten worden meestal *pages* genoemd
- client/server relatie tussen browser en server
- oorspronkelijk gebaseerd op *hypertext* nu ook *hypermedia*
- soms *external viewers* (*helper applications*) nodig

WWW

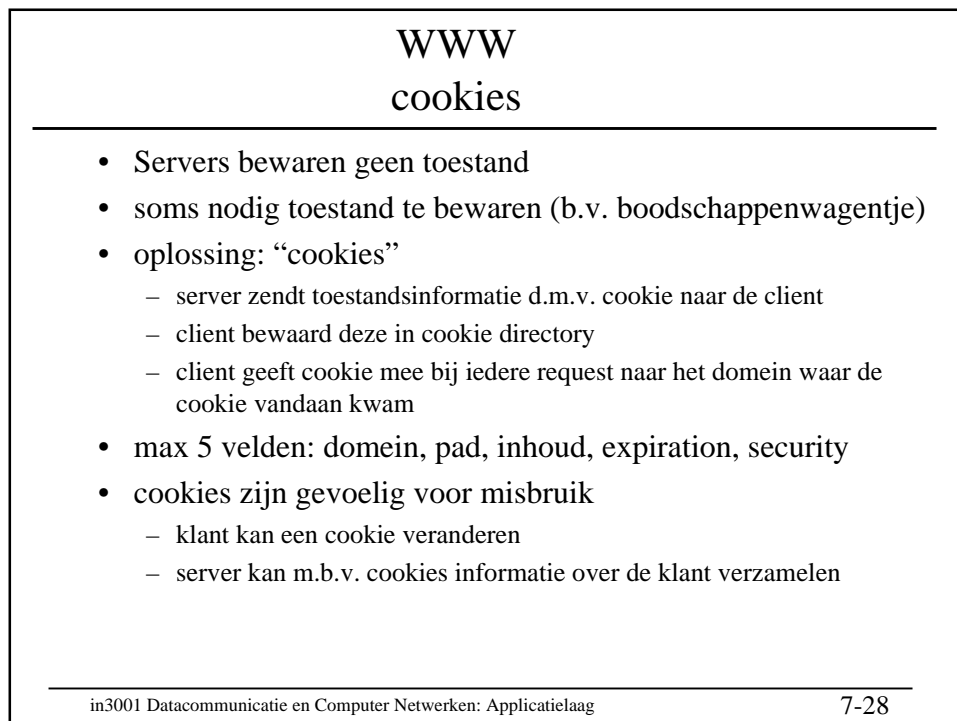
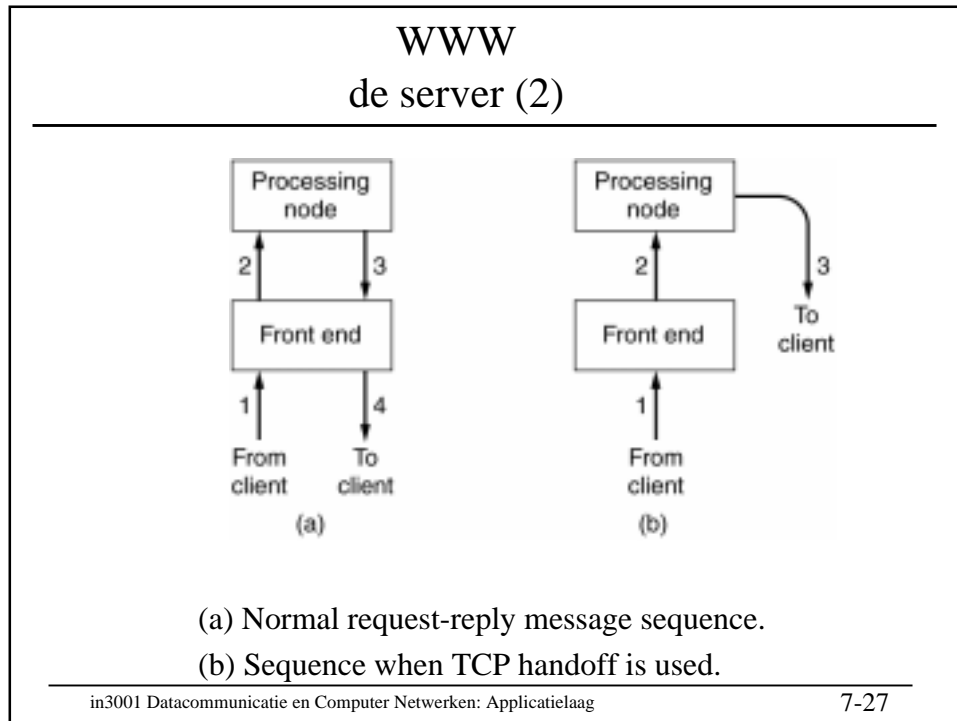
het opvragen van pages

- pages worden geïdentificeerd d.m.v. *URL (Universal resource Locator)*, URL bevat o.a. hostnaam en filenaam
- procedure voor opvragen van een page :
browser (de client)...
 - vraagt via DNS het IP-adres dat hoort bij de hostnaam
 - maakt TCP-verbinding met poort 80 op het gevonden IP-adres
 - geeft request, krijgt reply
 - heft TCP-verbinding op (*tenzij ...*)
 - toont tekst
 - haalt en toont bijbehorende files (b.v. plaatjes), zonodig via aparte TCP-verbindingen
 - maakt gebruik van
 - plug-in's worden geladen in browser
 - helper applicaties (aparte processen)

WWW

de server (1)

- Basisproces:
 - de server luistert naar (TCP) poort 80
 - ontvangt een request, met daarin de filenaam
 - zoekt file (de web page)
 - zendt de inhoud naar de client
- Bij grote aantallen requests: performance probleem oplossingen:
 - caching
 - multithreading (een thread per disk)
- Bij een nog grotere load: *server farm model*
 - groot aantal (gespecialiseerde) servers
 - eventueel een front-end voor het verdelen van de requests
 - eventueel antwoord van server door middel van TCP handoff



WWW

web documents overzicht

- statische documenten
 - oorspronkelijk gebaseerd op HTML
 - XML en SML bieden betere mogelijkheden voor structurering
 - XHTML opvolger van HTML 4, is stricter dan HTML 4
- dynamische documenten
genereren dynamisch web pages (“on the fly”),
verschillende mogelijkheden:
 - Server-side web page generatie:
 - CGI Interface
 - PHP (of JSP of ASP)
 - client side web page generatie
 - Java script
 - applets (of ActiveX controls)

HTML (1)

- HTML (Hypertext Markup Language) wordt gebruikt om de inhoud van de pages te definiëren
- hyperlinks komen in de page voor in de vorm van *URL* 's :
format:
<protocol>://<machinenaam>/<filenaam>
 - <protocol> meestal **http**, maar ook **gopher**, **ftp**, ...
 - <machine> geeft naam zoals bekend in DNS
 - <filename> is volledige filename, of een afkorting,
b.v. **~userx/** betekent default web directory onder de home directory van userx, waarin de default filenaam **index.html** verwacht wordt.

HTML (2)

- HTML aanpassing van SGML (ISO)
- maakt gebruik van *Markup Commands (tags)*, b.v.
 - <HEAD> en </HEAD> (is begin en einde van een kop)
 - en (begin en einde van boldface)
- browser interpreteert tags bij het weergeven van de page, houdt hierbij rekening met de locale situatie. (kleur/geen kleur, groot/klein window etc.)
- HTML kent ook logische stijlen, b.v.
 - voor emphasis voor strong emphasis
- weergave van de logische stijlen wordt bepaald door *style sheet*.

HTML (3)

- HTML kent mogelijkheid voor het definiëren van invulformulieren: forms, hierin aangegeven:
 - De URL van het programma dat dit formulier moet verwerken
 - de namen en de grootte van de velden
 - de manier waarop de gegevens worden overgedragen

voorbeeld:

```
<form action="http://~usery/cgi-bin/order" method="GET" >
<p> Naam <input name="naam" size=20>
<p> Plaats <input name="woonplaats" size=16>
</form>
```

Javascript

- extra code in HTML file
- wordt uitgevoerd door de client
- Javascript code begrensd door tags:

voorbeeld:

```
<script language="javascript" type="text/javascript">  
document.writeln("Hello there");  
//en nog veel meer en nog slimmere code  
</script>
```

XML

- HTML bevat informatie over inhoud en formatting
- gewenst: scheiding tussen inhoud en formatting
b.v bij het doorzoeken van web pages
- XML (Extensible Markup Language) beschrijft de inhoud
- XSL (Extensible Style Language) beschrijft de formatting
- XML te tonen d.m.v. XSL

een XML file

```

<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="studenten.xsl" ?>
<students>
<student>
  <naam> S. Gonzales </naam>
  <nummer> 1234567 </nummer>
</student>

<student>
  <naam> N. Bonaparte </naam>
  <nummer> 1815 </nummer>
</student>

</students>

```

Een XSL file

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html> <body>
<table border="2">
  <tr>  <th> Student </th>  <th> Nummer </th>  </tr>
<xsl:for-each select="students/student">
<tr>
  <td> <xsl:value-of select="naam"/></td>
  <td> <xsl:value-of select="nummer"/></td>
</tr>
</xsl:for-each>
</table>
</body> </html> </xsl:template> </xsl:stylesheet>

```

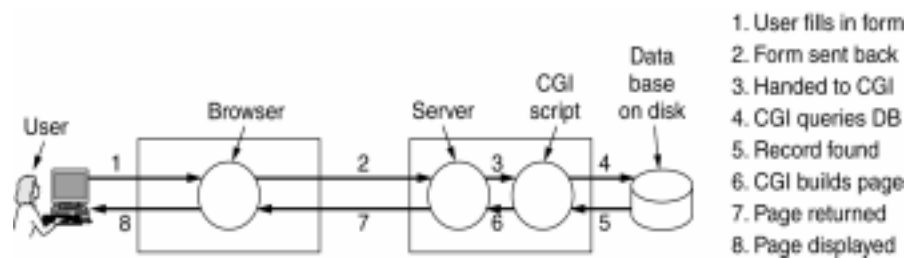
XML en SML resultaat

Student	Nummer
S. Gonzales	1234567
N. Bonaparte	1815

CGI (1)

- Common gateway Interface (CGI)
- Door client ingevulde velden van een form worden door het cgi interface doorgegeven aan een cgi-programma of cgi-script .
- Het programma of script wordt uitgevoerd in de server
- waarden worden doorgegeven in de vorm van een string, b.v. naam=Jansen&adres=brink+6+Ons+Dorp
- cgi-programma maakt een html page, met daarin de uitkomst (b.v. het resultaat van een database aanroep)
- de page wordt via het cgi-interfcae teruggestuurd naar de client.

CGI (2)



PHP (1)

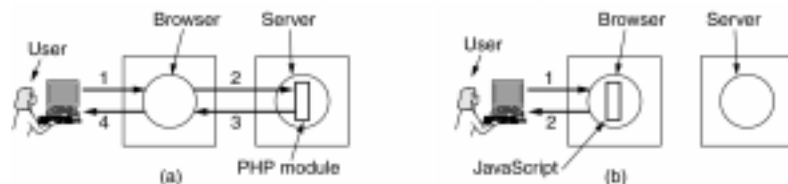
- PHP: Hypertext Preprocessor (PHP)
- page source bevat code van php script
- php script wordt uitgevoerd in de server
- filenaam moet eindigen met .php
- werkt alleen met php enabled server
- voordeel van php zeer eenvoudige parameter overdracht,

voorbeeld:

```
<html><body>  
<?php echo $naam ?>  
</body></html>
```

dit plaatst de waarde van het veld naam in de page

PHP (3)



a. situatie bij PHP

b. situatie bij JavaScript

Ook combinaties van PHP en JavaScript zijn mogelijk

HTTP protocol (1)

HTTP (HyperText Transfer Protocol)

- HTTP is een ASCII protocol (d.w.z. er worden printbare tekens uitgewisseld)
- HTTP kent verschillende *request methods*, b.v.
 - GET vraagt een page
 - HEAD vraagt een header
 - PUT geeft opdracht een page op te slaan
 - POST voegt iets toe aan, b.v. aan nieuwsgroep
 - OPTIONS vraagt eigenschappen van server of van een file

HTTP (2)

Message headers, zijn de parameters van de request en reply methods.

voorbeelden;

- request headers:

- User-Agent informatie over browser en platform
- Accept-Charset informatie over acceptabele charcacter sets
- Host de naam van de server
- Cookie geeft een cookie mee

- response headers

- Content-Type het MIME type van de page
- Location verwijst naar een andere bestemming
- Last-Modified Tijdstempel van de laatste wijziging
- Set-Cookie opdracht tot het bewaren van een cookie

HTTP protocol (3)

- Op iedere request volgt een *reply* , bestaande uit
 - een *status regel*
3 cijferige statuscode (5 groepen, 1xx t/m 5xx)
 - andere informatie (b.v. de gevraagde page)

Web Performance

Drie manieren om te performance te verhogen:

- caching
- server replication
- Content Delivery Networks

performance: caching

- caching door middel van een proxy
 - in gebruikers PC
 - in een locale server
 - in de ISP server
- hierarchisch cachen
- Vraag: hoe lang cachen?
 - methode 1: gebruik Last-Modified header: lang niet gewijzigd, betekent zal waarschijnlijk ook in de toekomst niet snel wijzigen.
 - methode 2: (RFC 2616) GET met If-Modified-Since request header
 - methode 3: een combinatie van 1 en 2: gebruik gedurende een periode ΔT methode 1, daarna methode 2.
 ΔT hangt af van hoe lang geleden de laatste wijziging van die page was

performance: Server replication

- Drukke servers kunnen *mirroring* gebruiken, d.w.z één of meer copieën elders op de wereld.
- probleem: *flash crowds* (plotseling grote belangstelling)
mogelijke oplossing:
 - automatisch klonen van server.
van tevoren overeenstemming nodig met nieuwe gastheren
 - dynamische replica's van veel gevraagde pages

performance: Content Delivery Networks (1)

- Content providers (b.v. kranten) maken gebruik van diensten van CDN
- CDN (Content Delivery Network) plaatst servers bij ISP's
- gebruiker komt via oorspronkelijk adres (URL) van Content Provider bij CDN, CDN kiest gunstigste server en stuurt request door.

Performance: Content Delivery Networks (2)

