

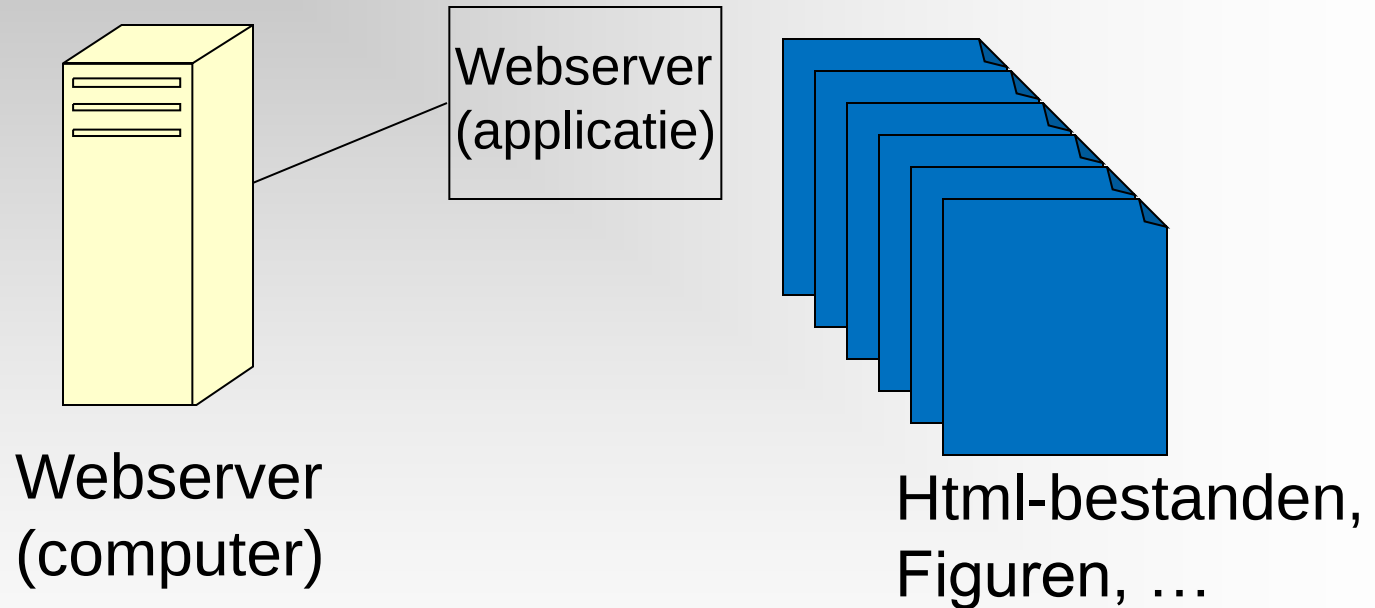
Servlets

Veerle Ongenae

Servlets

- Basisbegrippen
- Sessies
- ServletContext
- Levensloop van een servlet

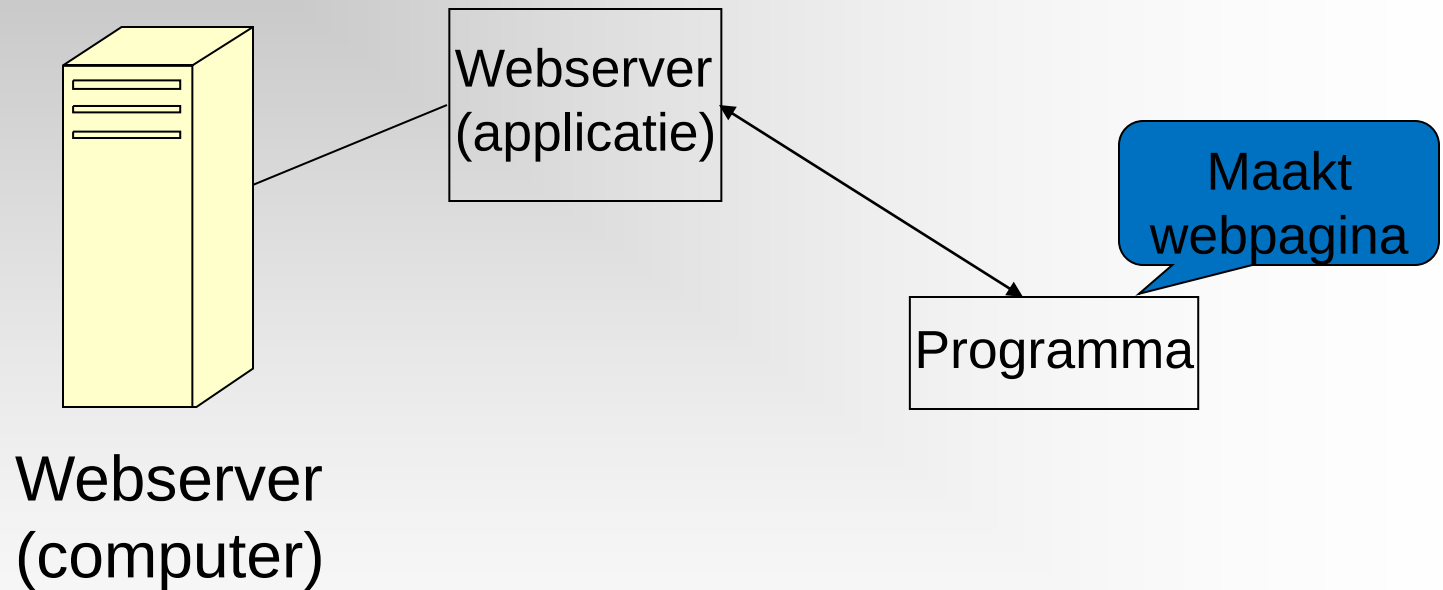
Webservers: statische pagina's



Dynamische webpagina's

- Webpagina's worden gegenereerd
 - Op server
 - Door “programma's”
 - Geïnterpreteerd (ASP, PHP, ...)
 - Gecompileerd (CGI, Servlets, ASP.NET, ...)

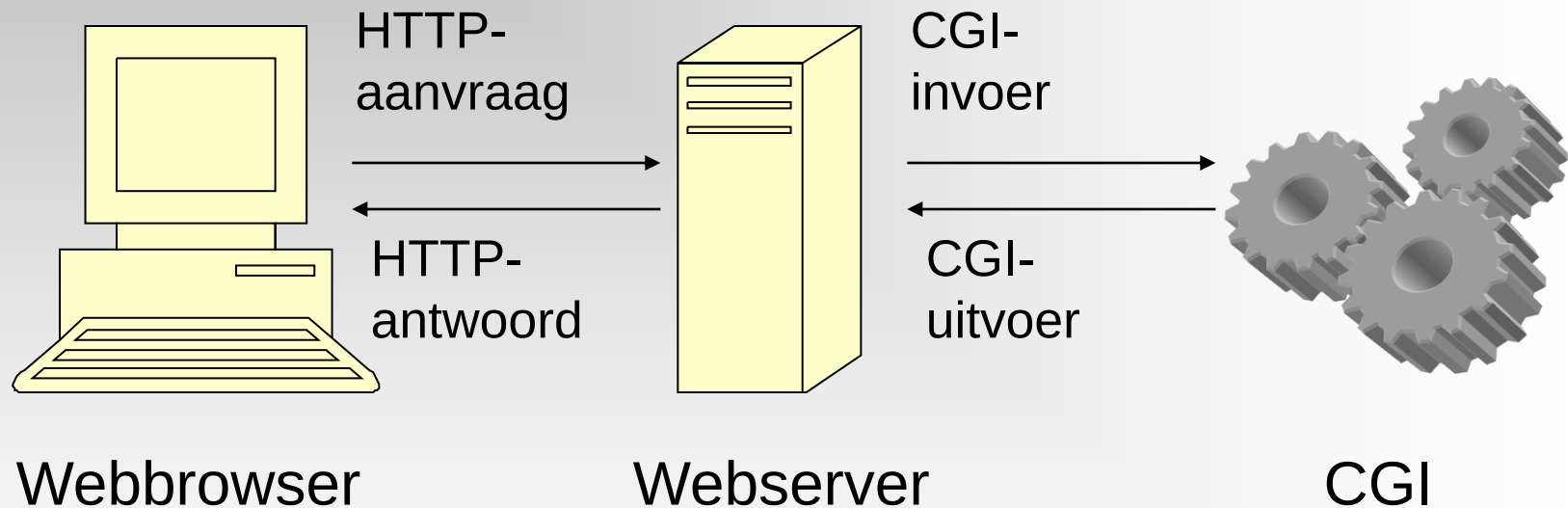
Webserver: dynamische pagina's



CGI

- Common Gateway Interface
- Uitbreiden functionaliteit webserver
- Conventies, standaard: CGI
- Programma's die deze conventie volgen:
CGI-scripts
- Verschillende programmeertalen mogelijk

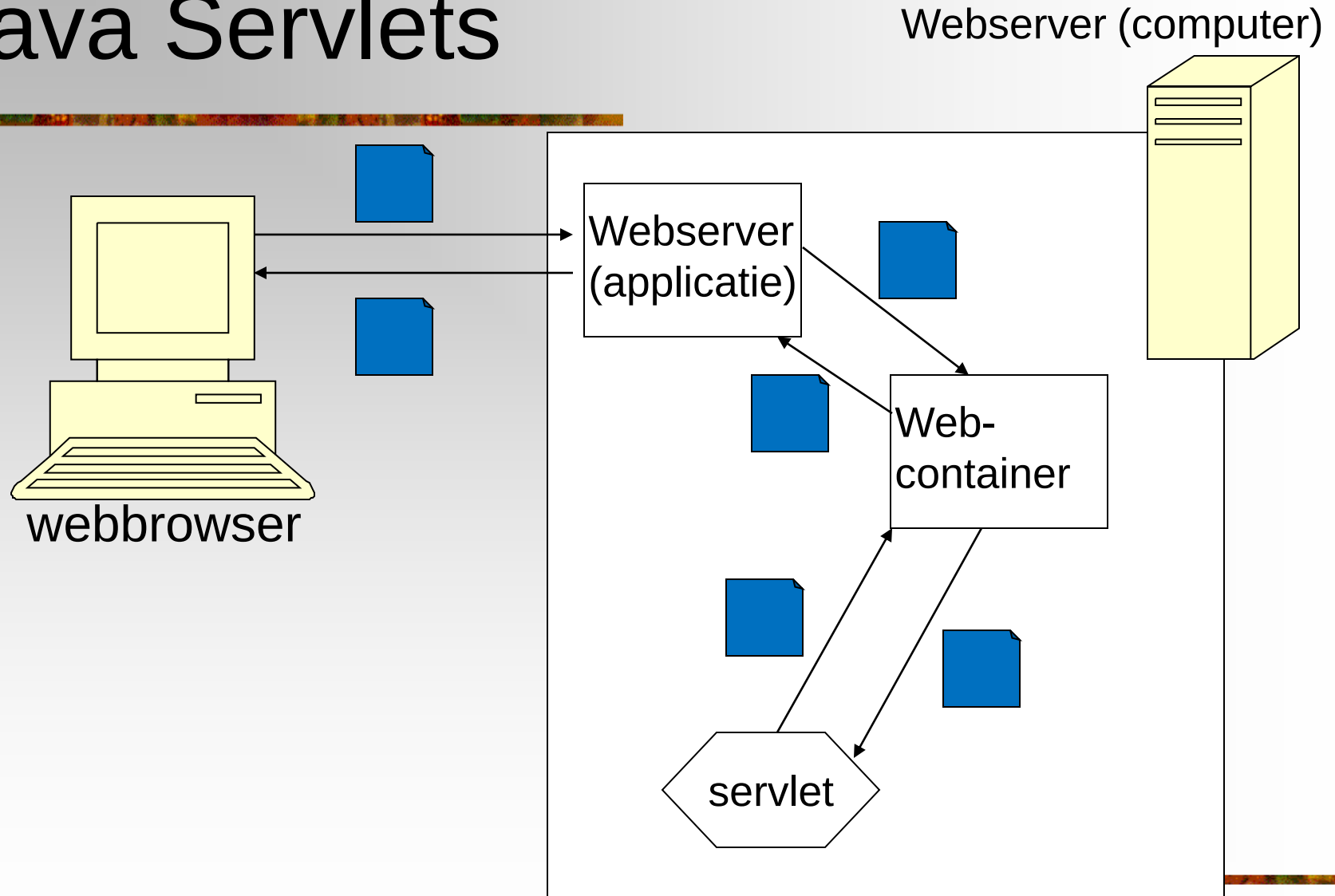
Werkwijze CGI



Java-Servlets

- Java-servlets zijn
 - kleine, platform onafhankelijke
 - server-side klassen in java
 - die de functionaliteit van de webserver uitbreiden
- Opmerking: eigenlijk ook voor andere vraag/antwoord-georiënteerde servers

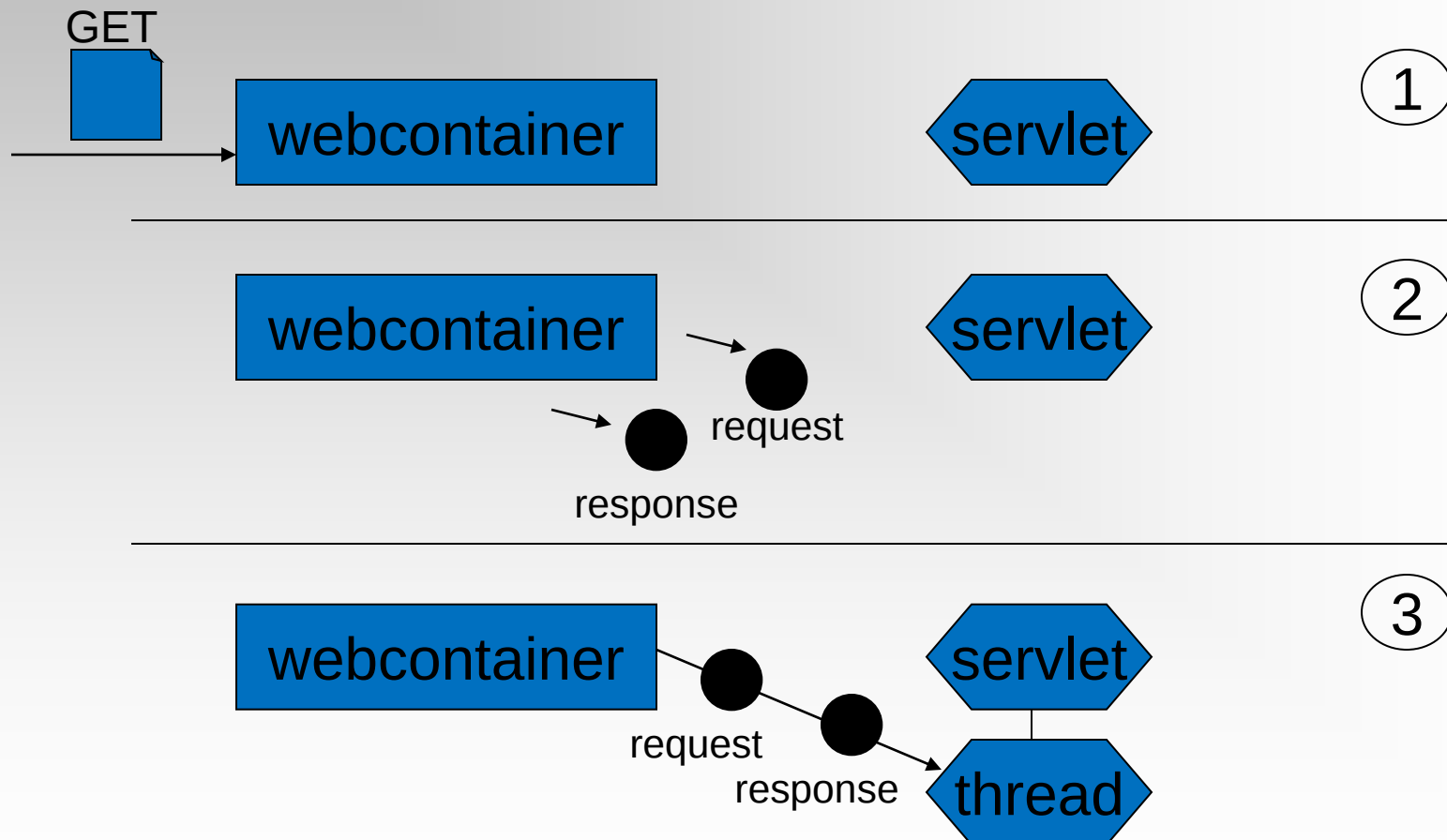
Java Servlets



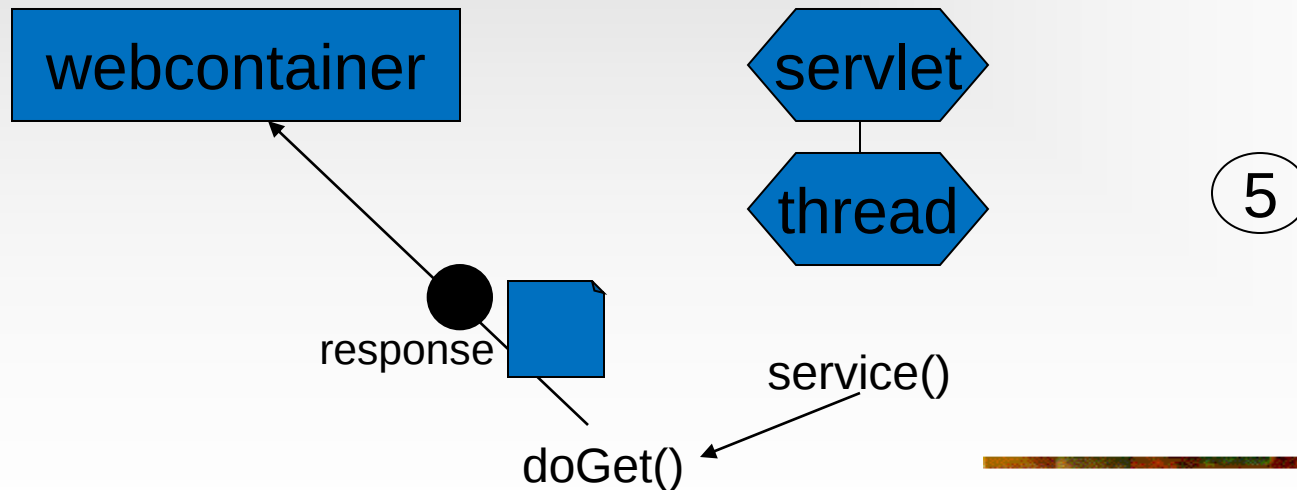
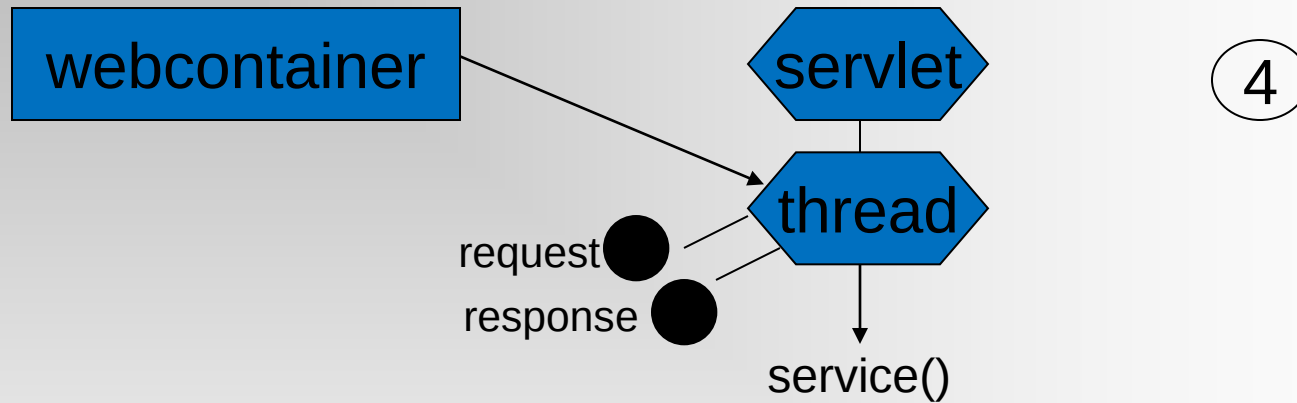
Webcontainer

- Beheert het servlet
 - Java Runtime
 - Communicatie met de webserver
 - Beheer levensloop servlet
 - Klassen laden, instanties aanmaken, initialisatie, oproepen methodes servlet, ..
 - Ondersteuning “multithreading”
 - Verschillende gelijktijdige aanvragen
 - Beveiliging
 - Via configuratie
 - ServletContainer

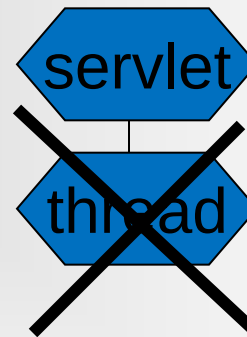
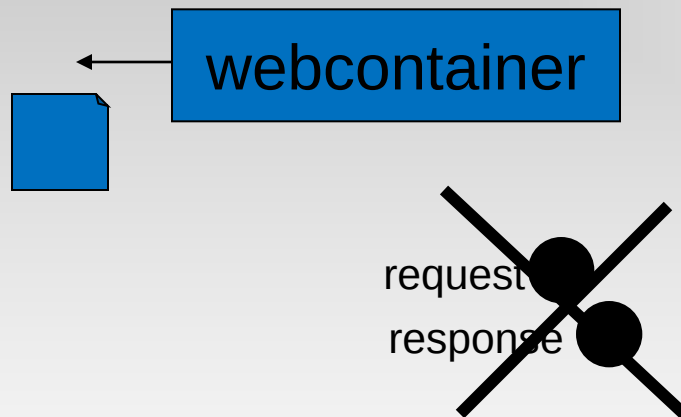
Afhandelen aanvragen



Afhandelen aanvragen



Afhandelen aanvragen



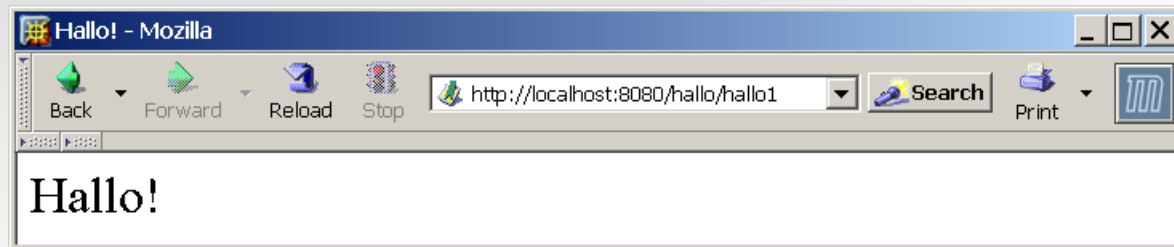
6

Servlets versus CGI's

- Sneller (start niet telkens een nieuw proces voor elke vraag)
 - Schaalbaar (meer gelijktijdige gebruikers)
- Platform onafhankelijk
- Voorzien API's voor programmatie aan de server-kant
 - Eenvoudiger
- Oplossing CGI-probleem met verschillende clients

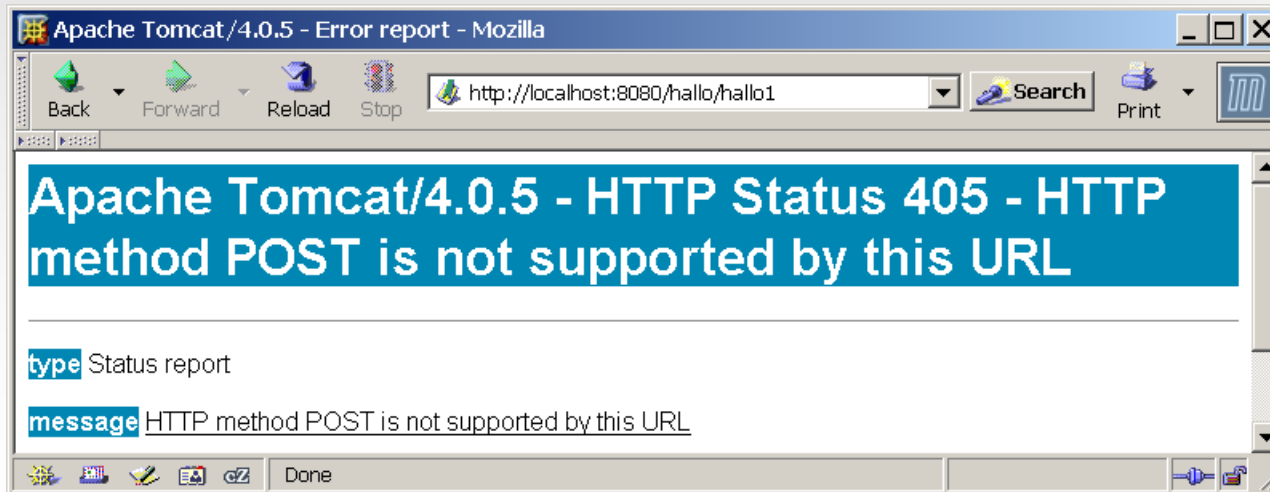
Voorbeeld

- <http://localhost:8080/hallo/hallo1>
 - Intikken URL
 - Pagina met het woordje “Hallo”



Voorbeeld

- <http://localhost:8080/hallo/geefNaam1.html>
 - Formulier
 - Actie /hallo/hallo1
 - Aanvraagmethode: POST
 - NIET ondersteund



Voorbeeld: servletcode

- `be.hogent.iii.servlets.HelloServlet1`
 - Ondersteunt enkel GET-aanvraagmethode
 - Genereert pagina met het woordje “hallo”

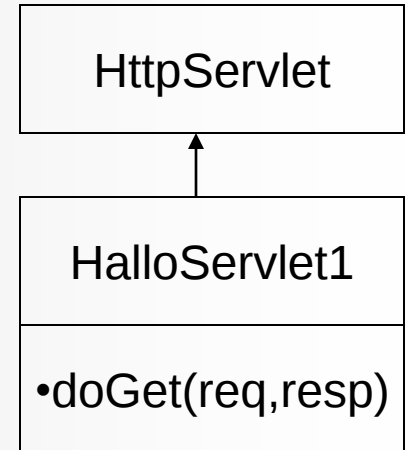
Werkwijze servlet

- Afgeleide klasse van HttpServlet
- Methode implementeren

public void **doGet**(HttpServletRequest req,
HttpServletResponse resp) throws
ServletException, IOException

- Werkwijze

- Headers instellen
- Stream/Writer ophalen
- Schrijven naar stream of writer
- Afsluiten stream of writer



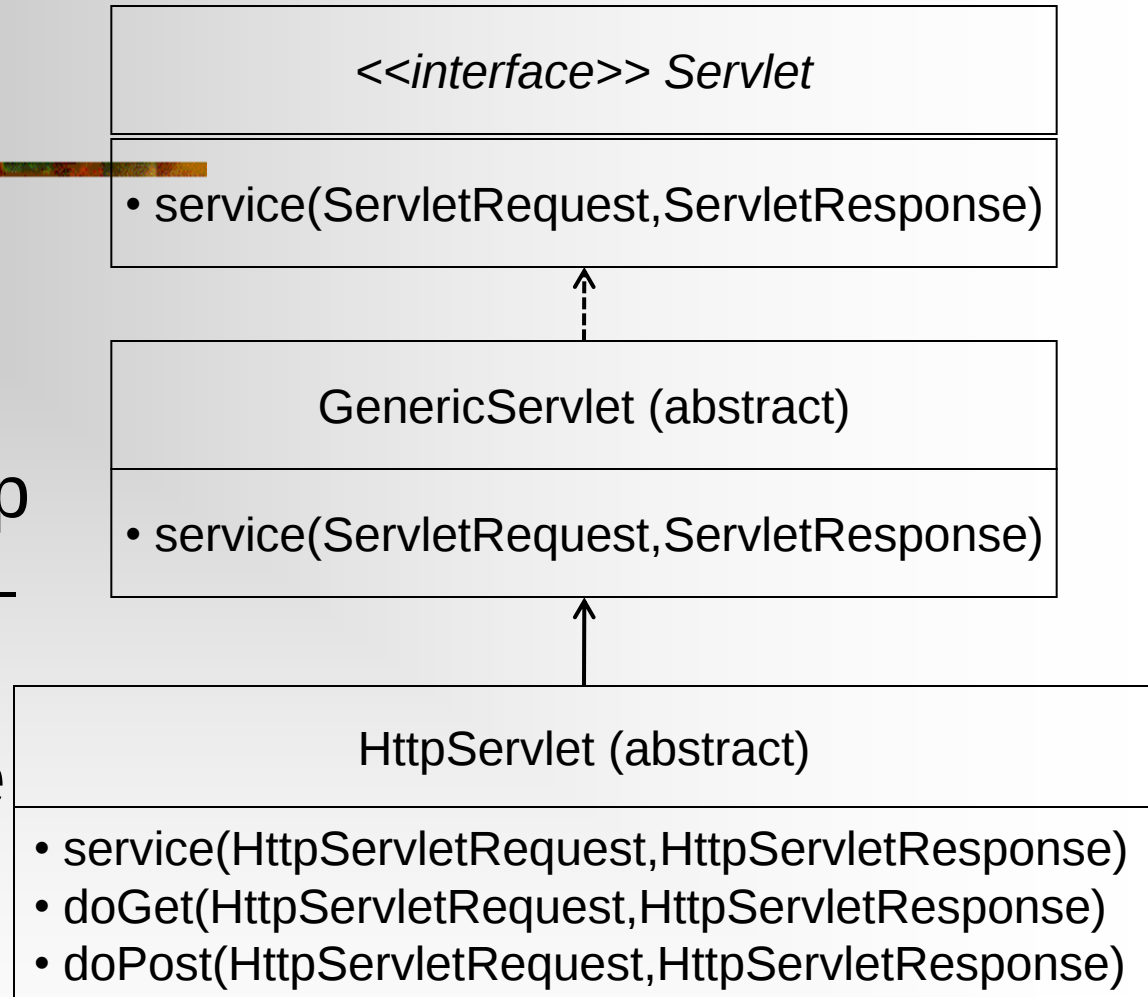
Servlet API

■ Pakketten

- javax.servlet
- javax.servlet.http
 - Specifiek HTTP-protocol

■ Servlet-interface

- Contract
 - Servlet
 - Webcontainer



Servlet API

- Interface

`javax.servlet.http.HttpServletRequest`

- Communicatie van de client naar de server

- Interface

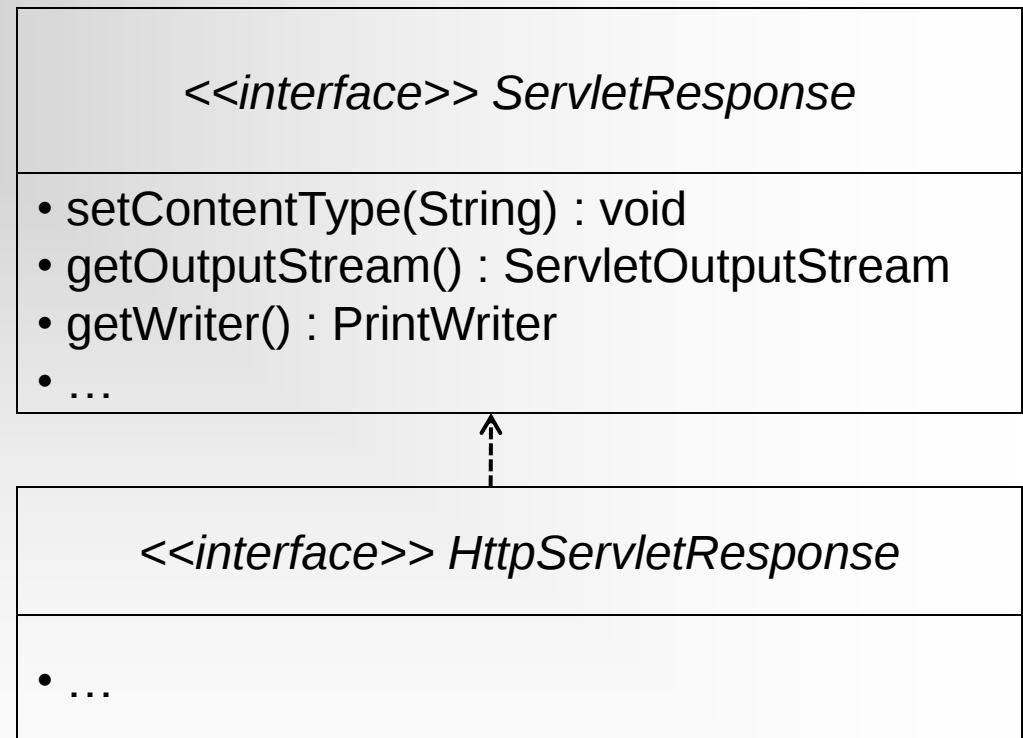
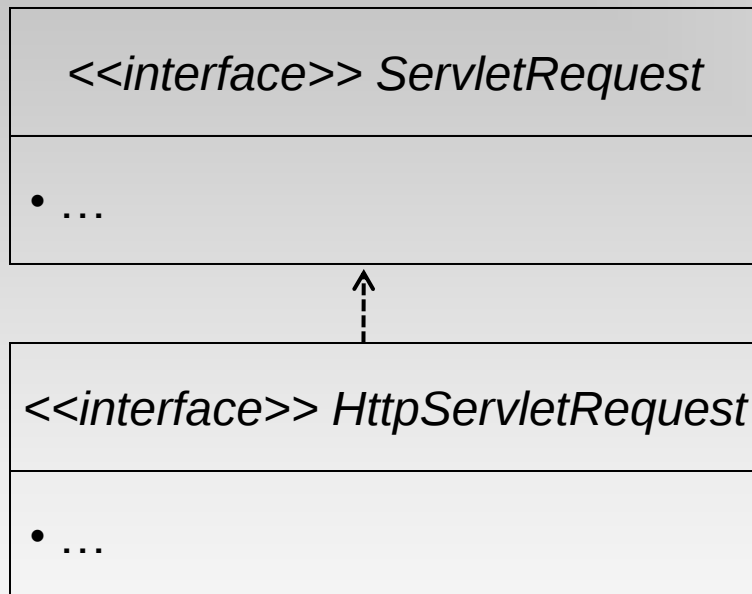
`javax.servlet.http.HttpServletResponse`

- Communicatie van de server naar de client

Servlet-interface

- Contract tussen servlet en webcontainer
 - Methodes ivm levenscyclus servlet
 - Methodes ivm configuratie
- Webcontainer “huisvest” servlet
 - Speelt aanvragen door naar servlet-objecten
 - Beheert servletobjecten

Servlet API



Overzicht API

- Interface ServletResponse

```
public void setContentType(String mimeType);  
public PrintWriter getWriter() throws IOException;  
public ServletOutputStream getOutputStream() throws  
    IOException;
```

- De interface HttpServletResponse is afgeleid van ServletResponse

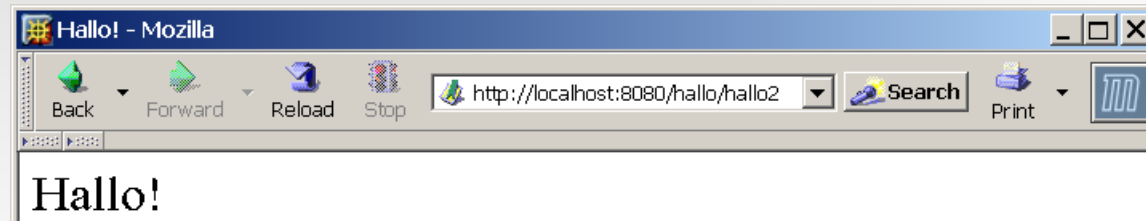
- Manipuleren headers HTTP-antwoord
- ...

- De klasse ServletException

- Aanvraag kan niet verwerkt worden

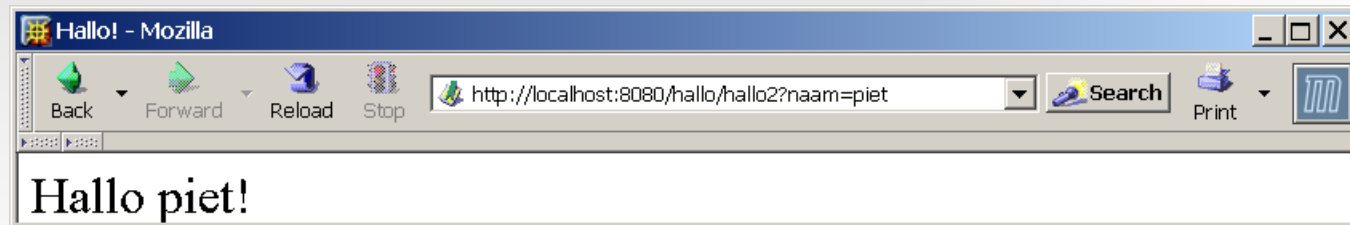
Voorbeelden

- <http://localhost:8080/hallo/hallo2>
 - URL intikken
 - Pagina met het woordje “Hallo”



Voorbeelden

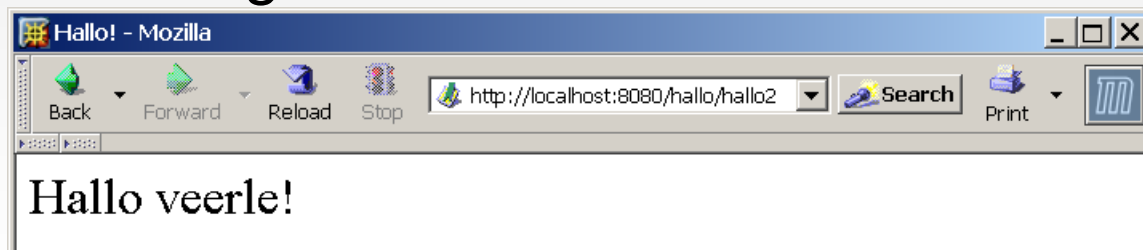
- URL met querystring
 - <http://localhost:8080/hallo/hallo2?naam=piet>
 - Pagina met “Hallo piet”



Voorbeelden



- Formulier (geefNaam2.html) met actie hallo2 (aanvraagmethode: POST)
 - Pagina met “Hallo NAAM”



Voorbeelden

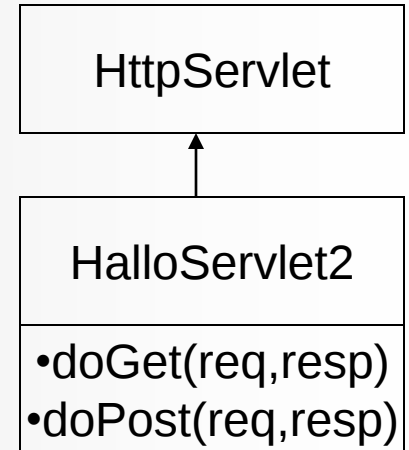
- `be.hogent.iii.servlets.HelloServlet2`
 - Ondersteunt GET- en POST-actie
 - Parameter naam
 - Genereert webpagina met het woordje “hallo”
+ waarde van de parameter naam

Werkwijze servlet

- Afgeleide klasse van HttpServlet
- Methode implementeren

```
public void doGet(HttpServletRequest req,  
    HttpServletResponse resp) throws  
    ServletException, IOException;
```

```
public void doPost(HttpServletRequest req,  
    HttpServletResponse resp) throws  
    ServletException, IOException
```



Werkwijze doGet of doPost

■ Werkwijze

■ Aanvraag

- Parameter “naam” ophalen

■ Antwoord

- Headers instellen
- Stream/Writer ophalen
- Schrijven naar stream of writer
- Afsluiten stream of writer

Overzicht API (deels)

- De interface ServletRequest

 - public String **getParameter**(String nm);

 - public String[] **getParameterValues**(String param);

 - public Enumeration **getParameterNames**();

- De interface HttpServletRequest is afgeleid van ServletRequest

 - Ophalen informatie headers

 - ...

Hoe vindt webcontainer servlet?

- Afbeelding tussen URL en servlet
- Configuratie
 - web.xml (Deployment Descriptor)
- Servlet heeft drie namen
 - Padnaam class-bestand
 - Geheime interne naam
 - Publieke URL-naam

URL - management

- Eenvoudige URL's
- Persistente URL's
 - URL's mogen niet veranderen
- Weerspiegelen NIET de interne (bestands) structuur
 - Bestandsstructuur kan wijzigen
 - Relatie leggen tussen URL's en interne structuur
 - Verbergen interne structuur
- UI voor de gebruiker

Het bestand web.xml

- Configuratie van een webapplicatie
 - Relatie tussen URL's en servlets
 - ...
- Webapplicatie
 - Eén of meerdere servlets
 - Statische pagina's
 - ...

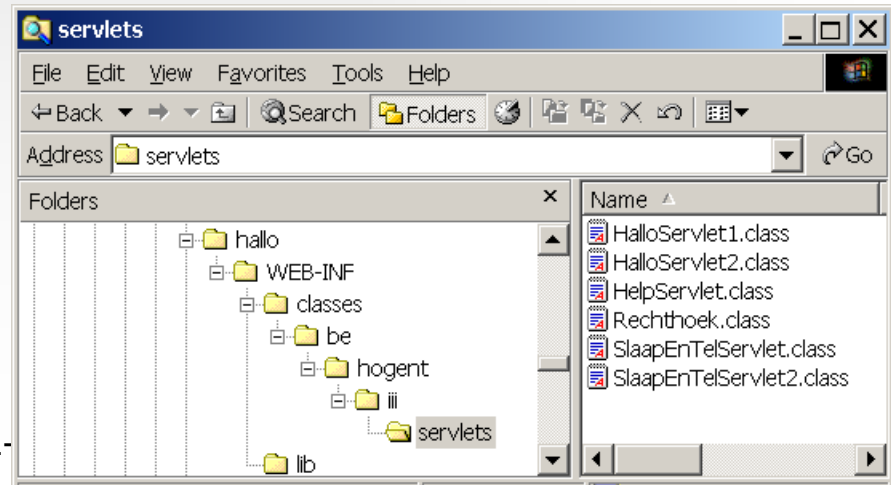
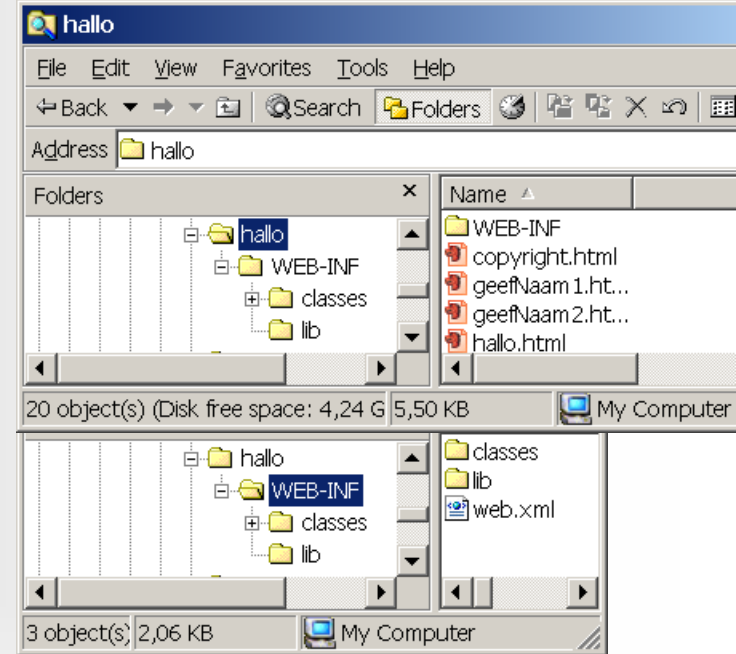
Configuratie

■ web.xml

```
<web-app>
  <servlet>
    <servlet-name>HalloServlet1</servlet-name>
    <servlet-class>
      be.hogent.iii.servlets.HalloServlet1
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HalloServlet1</servlet-name>
    <url-pattern>/hallo1</url-pattern>
  </servlet-mapping>
</web-app>
```

Structuur webapplicatie

- Basismap (naamWebapp)
 - HTML-bestanden, JSP's, figuren, ... (eventueel in submappen)
 - Map WEB-INF
 - web.xml: configuratie webapplicatie
 - Map classes: .class-bestanden (bv. `be/hogent/iii/servlets/HelloServlet.class`)
 - Map lib: .jar-bestanden (bv. JDBC-driver)



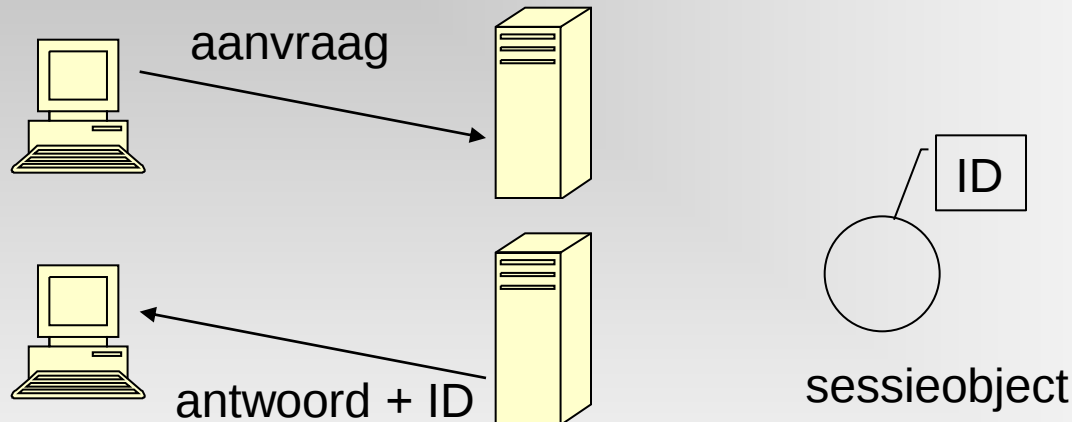
Servlets

- Basisbegrippen
- Sessies
- ServletContext
- Levensloop van een servlet

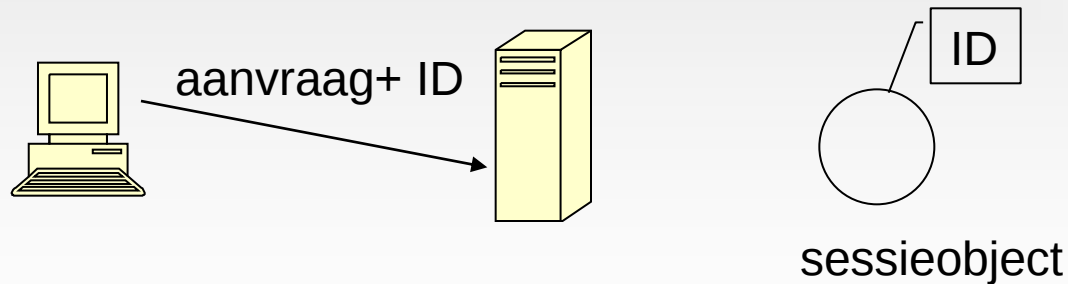
Opvolging van sessies (Session Tracking)

- HTTP: geen statusinformatie
- Informatie over een aantal requests van één gebruiker (=browser) bewaren gedurende een bepaalde tijd
- Webapplicatie met verschillende servlets: gebruikersinformatie delen tussen die servlets

Uitwisselen kenteken



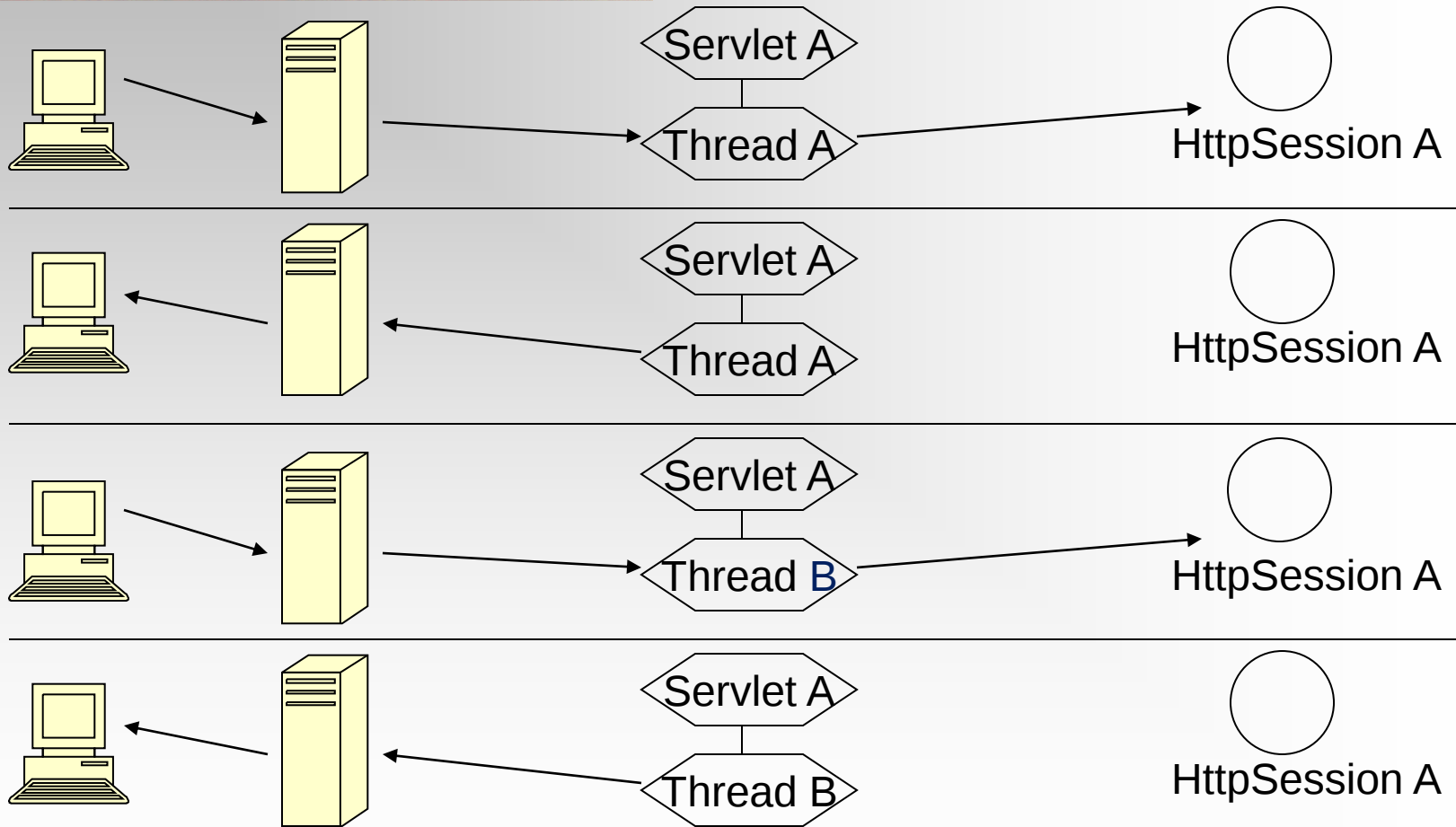
volgende
aanvragen



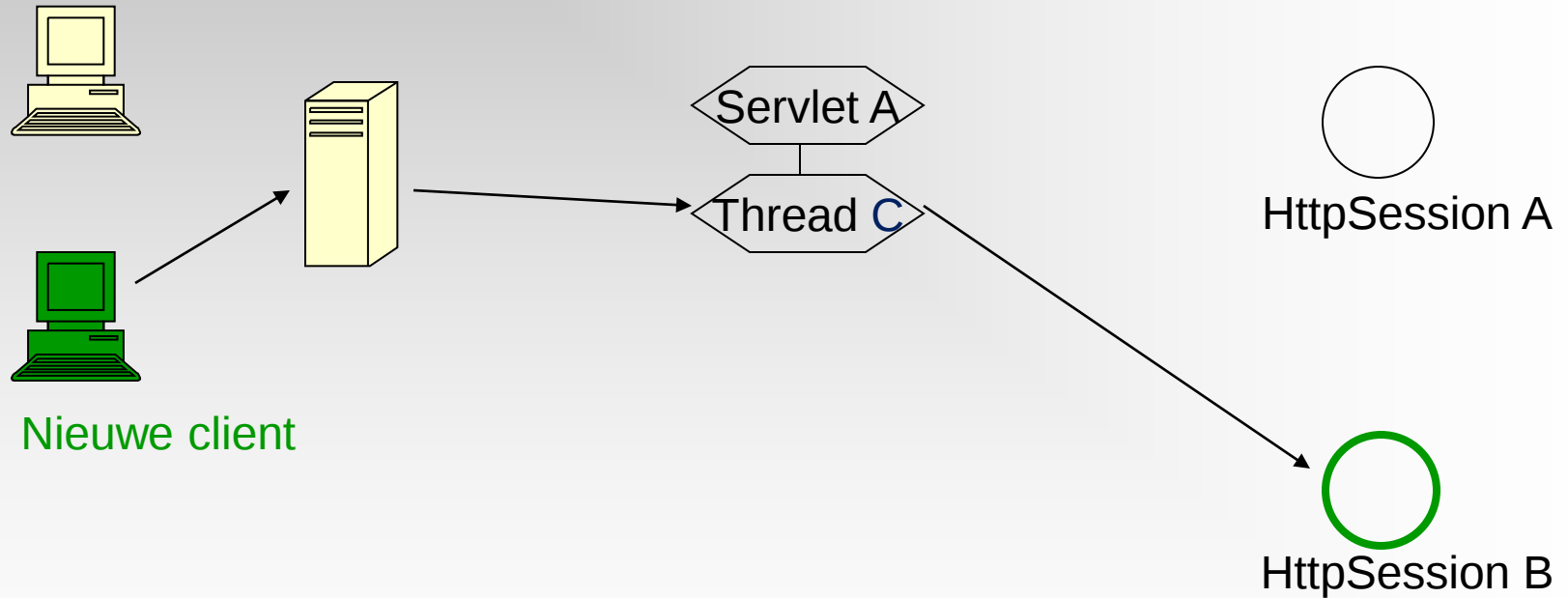
Principe sessies

- Uitwisselen kenteken tussen client en server
 - Het HTTP-antwoord van de eerste aanvraag krijgt een uniek kenteken mee
 - Bij elke volgende aanvraag wordt dit kenteken meegegeven
 - Kenteken is verbonden met een sessie-object op de server
- Mogelijke implementaties
 - Cookies
 - URL herschrijven
 - Verborgene formulierelementen
- Sessie-object
 - Daaraan kunnen attributen toegekend worden
 - Attributen: sleutel-objectparen

Sessies



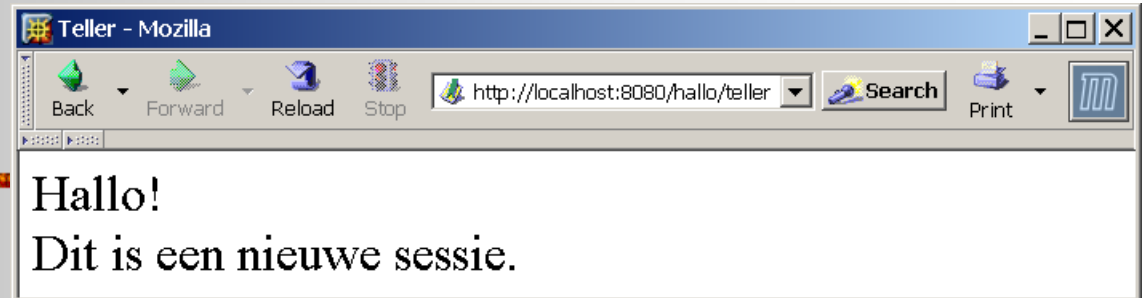
Sessies



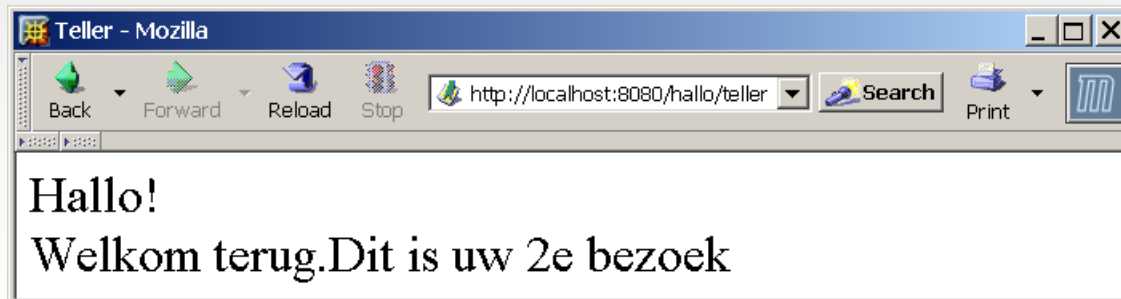
Werkwijze in servlet

- Ophalen sessie
 - Automatisch
 - Sessie-object maken en verbinden met een uniek id
 - Cookie aanmaken en associëren met id
 - Cookie toevoegen aan het antwoord-object
- Bewaren data in sessie of ophalen data van sessie
 - Sleutel-Object-combinaties (Sleutel = String en Object = object)
- Sessie afsluiten
 - Handmatig (methode invalidate())
 - Automatisch na bepaalde tijdsperiode

Voorbeeld



- Eerste maal
 - Mededeling + sessie aanmaken + teller op sessie
- Volgende keren
 - Mededeling + teller op sessie aanpassen
- `be.hogent.iii.servlets.TellerServlet`



Overzicht

■ HttpServletRequest

- `public HttpSession getSession()`
- `public HttpSession getSession(boolean create)`

■ HttpSession (interface)

- `public Object getAttribute(String naam)`
- `public setAttribute(String naam, Object waarde)`
- `public Enumeration getAttributeNames()`
- `public removeAttribute(String naam)`
- `public invalidate()`

Andere methodes HttpSession

- `public long getCreationTime()`
- `public String getId()`
- `public long getLastAccessedTime()`
- `public int getMaxInactiveInterval()`
- `public boolean isNew()`
- `public void setMaxInactiveInterval(int interval)`

Sessie-opvolging: Cookies

- Sessies opvolgen vereist gebruik van cookies
- HTTP-antwoord (na eerste aanvraag)

Set-Cookie:

JSESSIONID=215266E2484832741237D43
179502854;Path=/examples

- Bij volgende aanvragen:

Cookie:

JSESSIONID=B045A298B12327D793A131
7571C6F1E9

URL Herschrijven (URL Rewriting)

- Browsers die geen cookies (willen) ondersteunen: URL Herschrijven
- Alle links naar pagina's van de webapplicatie aanpassen
 - Unieke identificatie meegeven

URL Herschrijven (URL Rewriting)

- Interface HttpServletResponse
 - public String encodeURL(String url)
 - Enkel indien geen cookies

- De URL:

`http://localhost:8080/hallo/teller2`

- Vervangen door:

`http://localhost:8080/hallo/teller2;JSESSIONID=B045A298B12327D793A1317571C6F1E9`

Voorbeeld URL herschrijven



- Cookies weigeren
- URL herschrijven
- `be.hogent.iii.servlets.TellerServlet2`



Meer informatie

- Java Tutorial
(http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html)
- Professional Java Server Programming J2EE Edition (Wrox)
- Head First Servlets & JSP (O'Reilly)
- Servlet API
(http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/servlet/package-summary.html) en
(http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/servlet/http/package-summary.html)