

Programming Advanced java

Webcomponenten



Webcomponenten

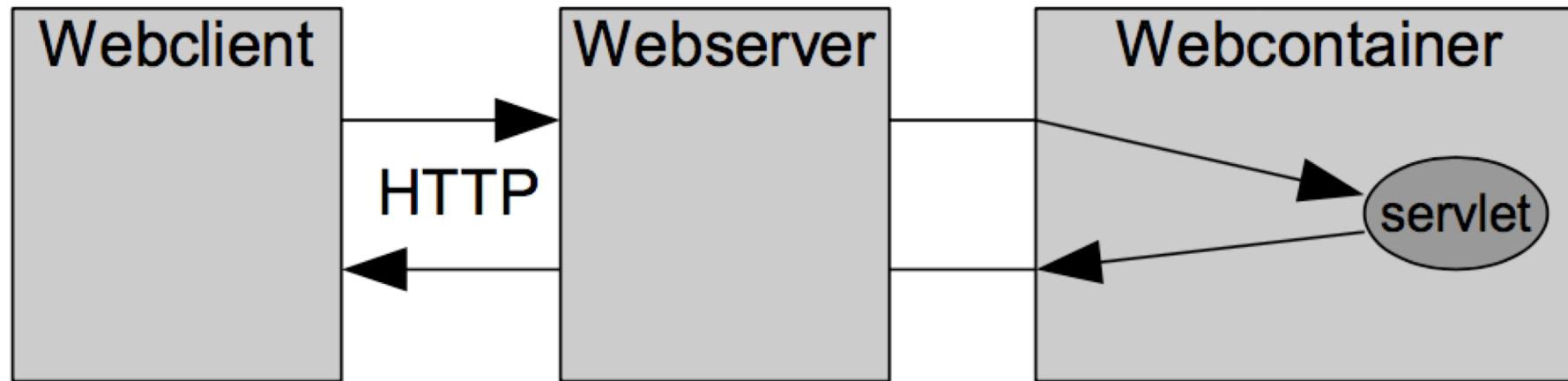
1. Webcontainer
2. Java webapplicatie
- 3. Servlets**
4. RESTful Web Services

Webcomponenten – servlets - inleiding

- Inleiding
- Servlets configureren / implementeren
- Lifecycle
- Service methoden (GET, POST, PUT, DELETE)
- Scope objecten
- Servlet interactie
- Overige

Webcomponenten – servlets - inleiding

- java klasse die een service biedt
- web applicatie kan bestaan uit één of meerdere servlets

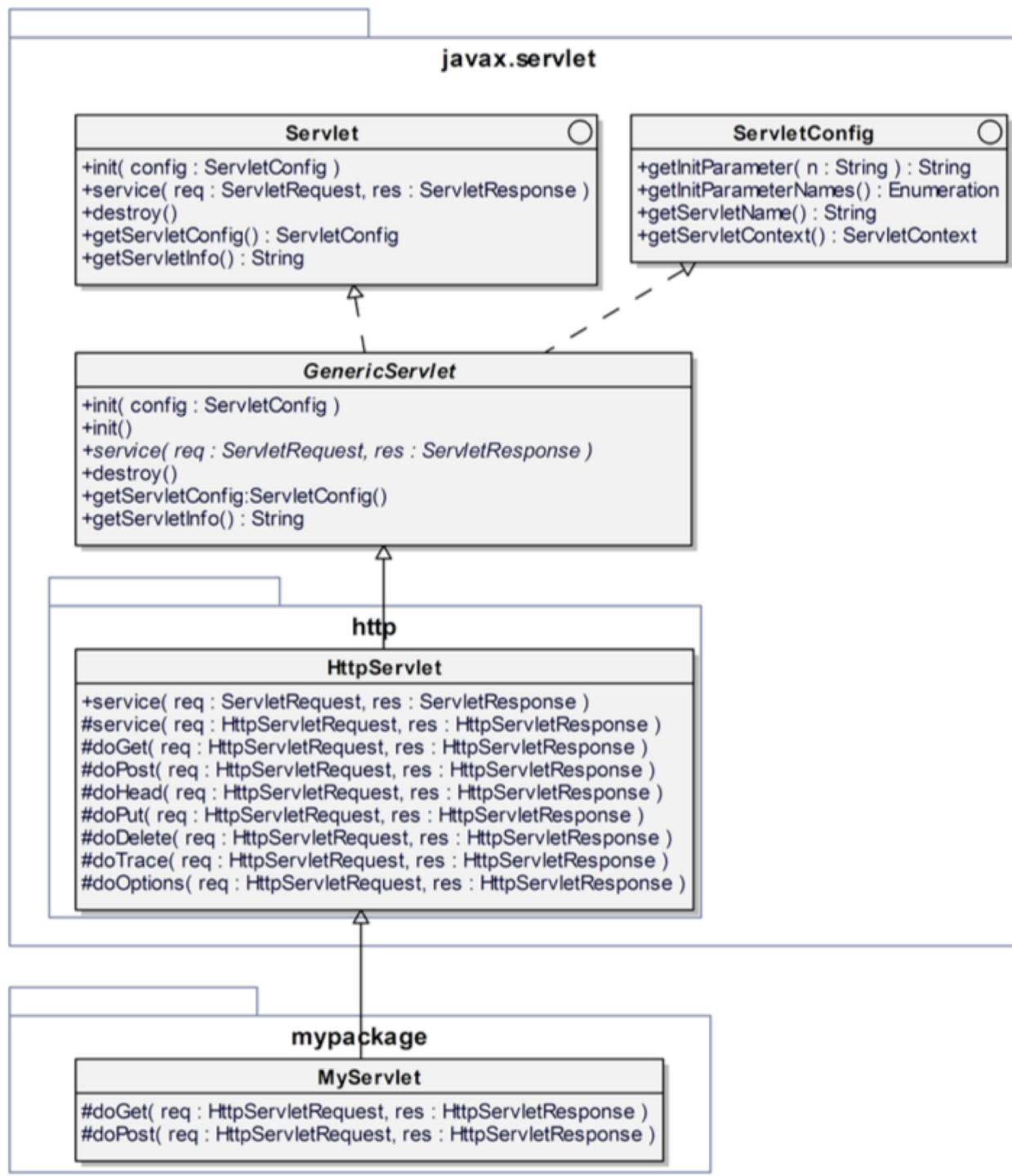


Webcomponenten – servlets - inleiding

Klassenhiërarchie

- javax.servlet.Servlet
 - javax.servlet.GenericServlet
 - javax.servlet.HttpServlet
 - doGet
 - doPost
 - doPut
 - doDelete
- JEE

```
<dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-web-api</artifactId>
    <version>7.0</version>
    <scope>provided</scope>
</dependency>
```



Webcomponenten – servlets – conf / impl

- Inleiding
- **Servlets configureren / implementeren**
- Lifecycle
- Service methoden (GET, POST, PUT, DELETE)
- Scope objecten
- Servlet interactie
- Overige

Webcomponenten – servlets – conf / impl

configuratie via

- deployment descriptor (web.xml)
- annotaties

Webcomponenten – servlets – conf / impl annotaties

@WebServlet

Element	Omschrijving
asyncSupported	Geeft aan of asynchrone afhandeling door deze <i>servlet</i> ondersteund wordt.
description	Een optionele beschrijving.
displayName	Een optionele naam die getoond kan worden door de webcontainer.
initParams	Een lijst van initialisatieparameters.
largeIcon	Grote afbeelding.
loadOnStartup	De opstartvolgorde.
name	Naam van de <i>servlet</i> .
smallIcon	Kleine afbeelding.
urlPatterns	Lijst van URL-patronen.
value	Lijst van URL-patronen.

Webcomponenten – servlets – conf / impl

annotaties

- Value als enige element

```
@WebServlet("/HelloWorld") == @WebServlet(value="/HelloWorld")
```

- Lijst van URL-patronen

```
@WebServlet(value={"/HelloWorld","/Hello"})
```

Webcomponenten – servlets – conf / impl

deployment descriptor

web.xml

```
@WebServlet("/HelloWorld")
public class MyServlet extends HttpServlet {
...
}
```

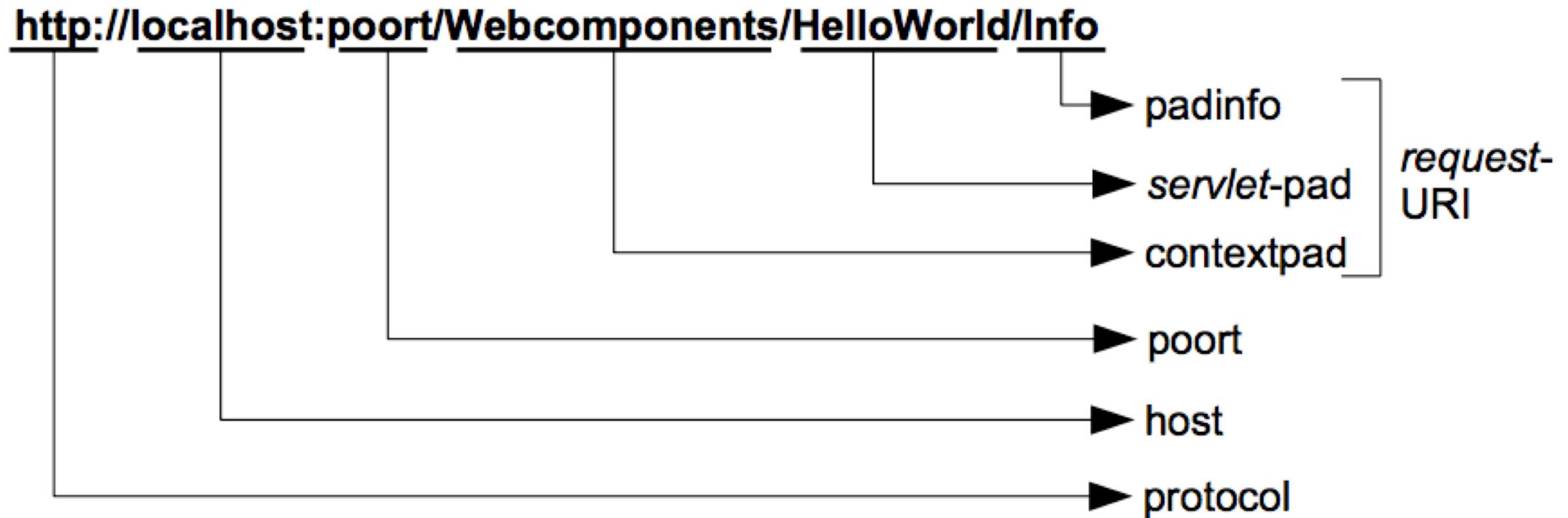


```
<servlet>
  <servlet-name>MyServlet</servlet-name>
  <servlet-class>be.pxl.student.web.MyServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>MyServlet</servlet-name>
  <url-pattern>/HelloWorld</url-pattern>
</servlet-mapping>
```

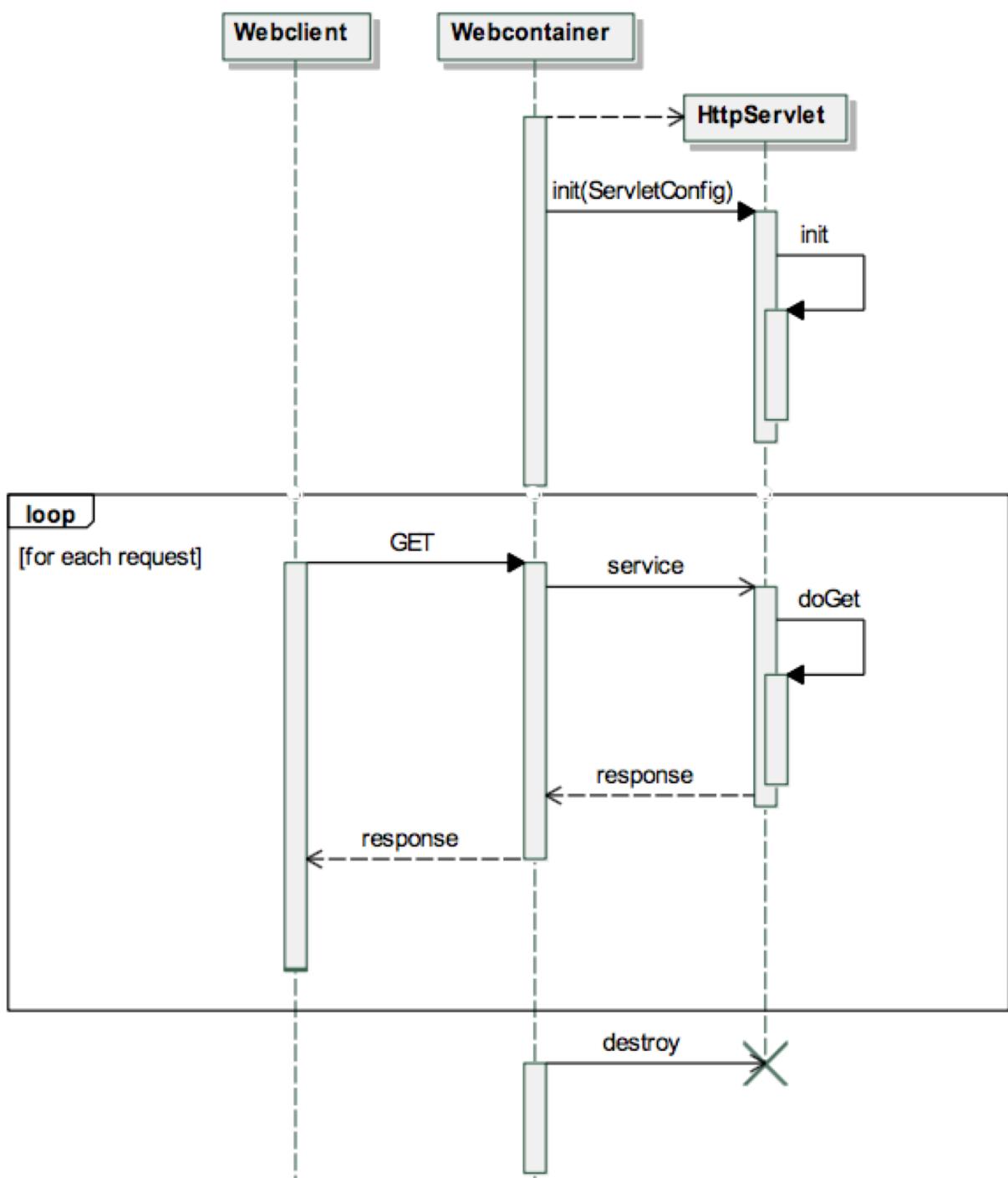
Webcomponenten – servlets – conf / impl

- URL patronen



Webcomponenten – servlets - Lifecycle

- Inleiding
- Servlets configureren / implementeren
- **Lifecycle**
- Service methoden (GET, POST, PUT, DELETE)
- Scope objecten
- Servlet interactie
- Overige



Webcomponenten – servlets - Lifecycle

- init ()
 - mogelijkheid tot init parameters
 - getInitParameter(...)
 - web.xml

```
<servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>be.pxl.student.web.MyServlet</servlet-class>
    <init-param>
        <param-name>someparam</param-name>
        <param-value>somevalue</param-value>
    </init-param>
</servlet>
```

- destroy ()

Webcomponenten – servlets - methoden

- Inleiding
- Servlets configureren / implementeren
- Lifecycle
- **Service methoden** (GET, POST, PUT, DELETE)
- Scope objecten
- Servlet interactie
- Overige

Webcomponenten – servlets - methoden

- service() methoden voor het HTTP protocol
 - doGet()
 - doPost()
 - doPut()
 - doDelete()

→ HttpServletRequest object
← HttpServletResponse object
- overige methoden
 - getServletConfig()
 - getServletInfo() : een beschrijving van de servlet
 - log()

Webcomponenten – servlets - methoden

- Het implementeren van een service method
 - 1. Opvragen van **parameters**
 - 2. **Request behandelen** en formuleren van een antwoord
 - 3. **Headers** instellen van het antwoord
 - Content-Type
 - CharacterEncoding
 - 4. Het schrijven van het **antwoord** naar een OutputStream of PrintWriter
- Bij een fout mag een ServletException of IOException gegenereert worden

Webcomponenten – servlets - methoden

<http://localhost:8080/MyWebApp/Echo?text=test>

```
import java.io.*;
import javax.servlet.http.*;

@WebServlet("/Echo")
public class EchoServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException {
        // Step 1 : get parameters
        String text = request.getParameter("text");
        // Step 2: generate content

        // Step 3: set headers
        response.setContentType("text/html");
        response.setCharacterEncoding("UTF-8");

        // Step 4: get PrintWriter
        try(PrintWriter out = response.getWriter()) {
            // Step 5: write content
            out.println("<!DOCTYPE html>");
            out.println("<html><head><title>Echo Servlet");
            out.println("</title></head><body>");
            out.println(text);
            out.println("</body></html>");
        }
    }
}
```

Webcomponenten – servlets - scope

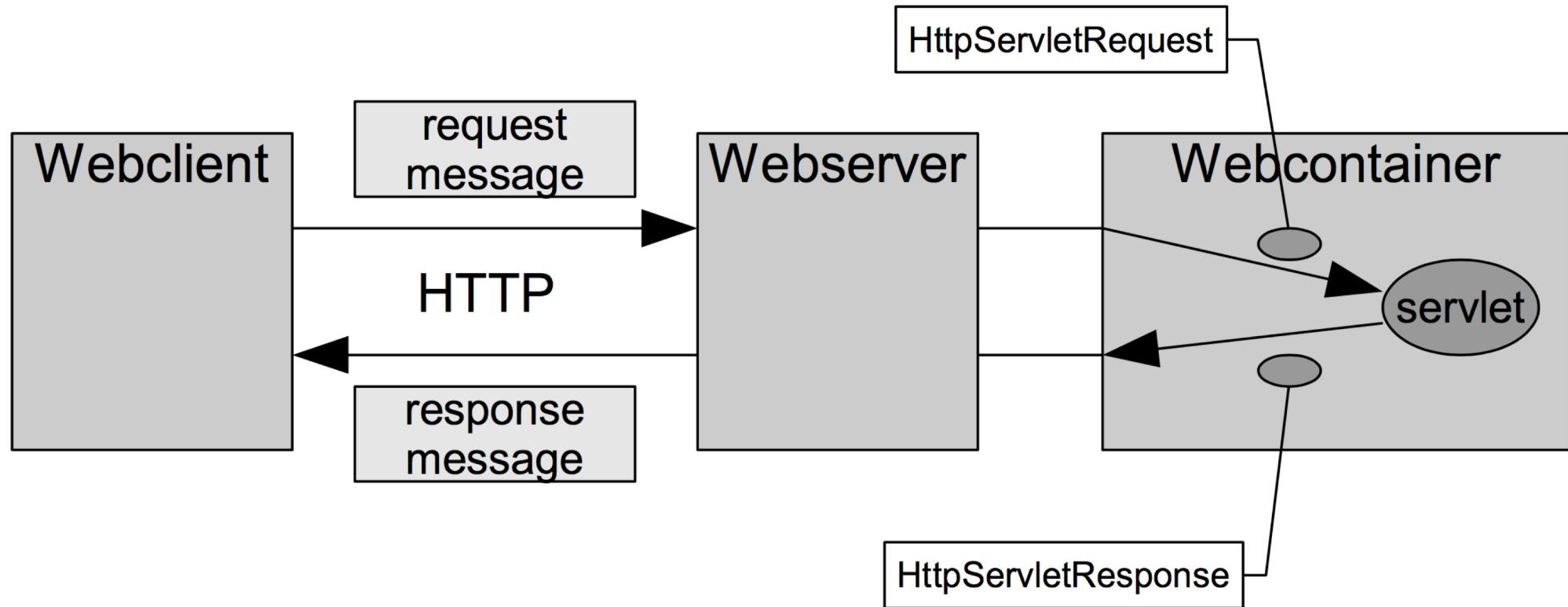
- Inleiding
- Servlets configureren / implementeren
- Lifecycle
- Service methoden (GET, POST, PUT, DELETE)
- **Scope objecten**
- Servlet interactie
- Overige

Webcomponenten – servlets - scope

- Allerhande objecten die gebruikt kunnen worden binnen een servlet
 - Request
 - HttpServletRequest
 - Response
 - HttpServletResponse
 - Session
 - HttpSession
 - Context
 - ServletContext

Webcomponenten – servlets - scope

Request / Response



Webcomponenten – servlets - scope

Request

Methode	Omschrijving
getInputStream() getReader()	Met deze methoden kan men een <i>inputstream/reader</i> bekomen waarmee men de <i>message body</i> kan lezen. Dit wordt enkel in speciale gevallen gebruikt waarbij men de inhoud zelf volledig wil verwerken.
getMethod()	Geeft de gebruikte HTTP-methode: GET,POST ...
getParameter() getParameterMap() getParameterNames() getParameterValues()	Met deze methoden kan men de meegegeven parameters opvragen. Dit geldt zowel voor de parameters die in de URL (GET) als in de <i>message body</i> worden meegegeven (POST).
getHeader() getHeaders() getHeaderNames() getIntHeader()	Met deze methoden kan men de <i>header</i> -gegevens uit de boodschap opvragen.
getRequestURI() getRequestURL() getServletPath() getPathInfo()	Geef de verschillende onderdelen van de opgegeven URL.
setAttribute() removeAttribute() getAttribute() getAttributeNames()	Met deze methoden kan men attributen toevoegen en opvragen. Attributen zijn objecten met een bepaalde naam.

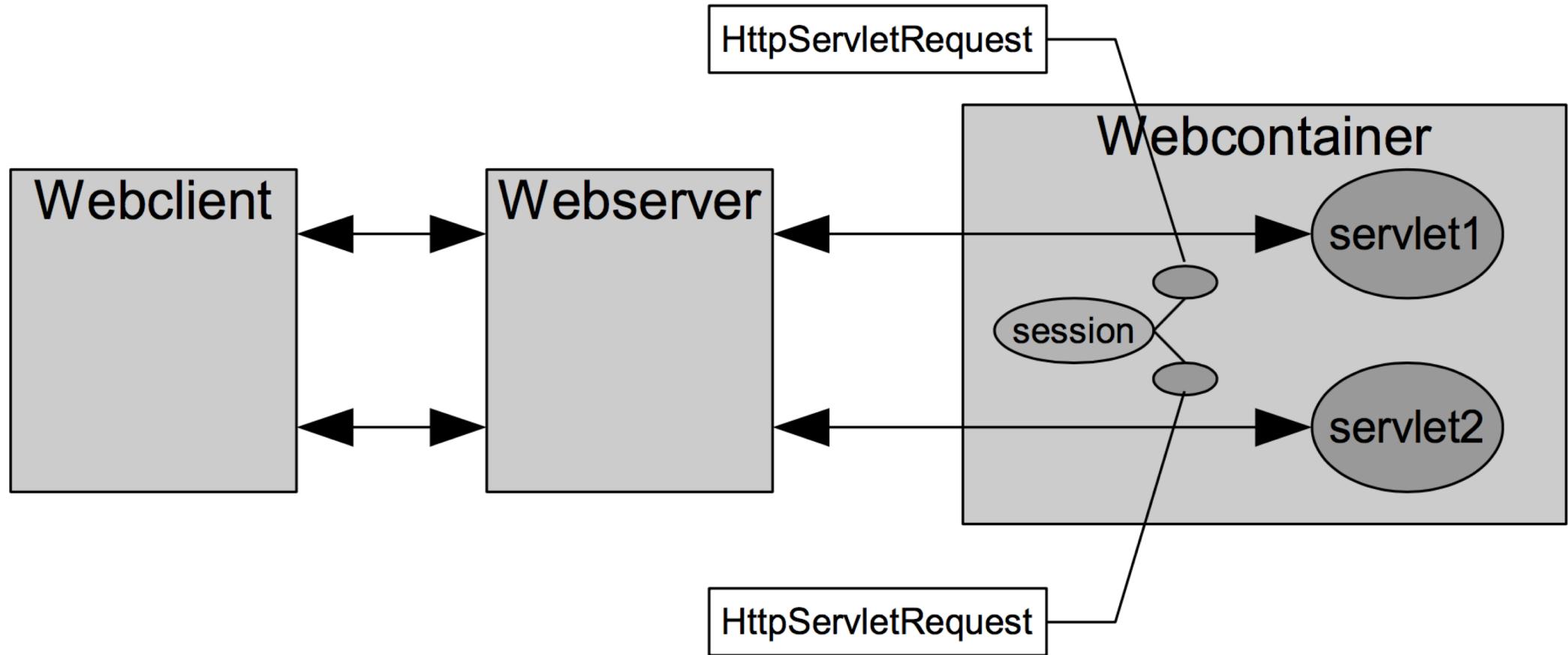
Webcomponenten – servlets - scope

Response

Methode	Omschrijving
getOutputStream() getWriter()	Met deze methoden kan men een <i>outputstream/writer</i> bekomen waarmee men de inhoud van de <i>message body</i> kan schrijven.
setContentType() setCharacterEncoding() setLocale()	Met deze methoden geeft men aan welke soort inhoud de <i>message body</i> heeft en welke codering gebruikt wordt.
setStatus() sendError()	Met deze methode wordt de statuscode ingesteld.
addHeader() addDateHeader() setHeader() setIntHeader()	Met deze methoden kan men de <i>header</i> -gegevens in de boodschap instellen.

Webcomponenten – servlets - scope

Sessie



Webcomponenten – servlets - scope

Sessie

Methode	Omschrijving
setAttribute ()	Hiermee kunnen we om het even welk object als attribuut aan het sessie-object toevoegen. Ieder attribuut heeft een naam die als parameter meegegeven wordt.
getAttribute ()	Hiermee kunnen we een attribuut opvragen.
getAttributeNames ()	Geeft een lijst van alle attributen die toegekend zijn aan het sessie-object.
isNew ()	Geeft aan of deze sessie nieuw is en dus nog niet bevestigd is door de gebruiker.
removeAttribute ()	Hiermee kan een attribuut verwijderd worden.

Webcomponenten – servlets - scope

Sessie

- Gebruik
 - `request.getSession()`
 - `session.setAttribute (...)`
 - `session.getAttribute (...)`
- Mogelijke implementaties
 - Cookies
 - URL-codering
 - SSL-sessies

Webcomponenten – servlets - scope

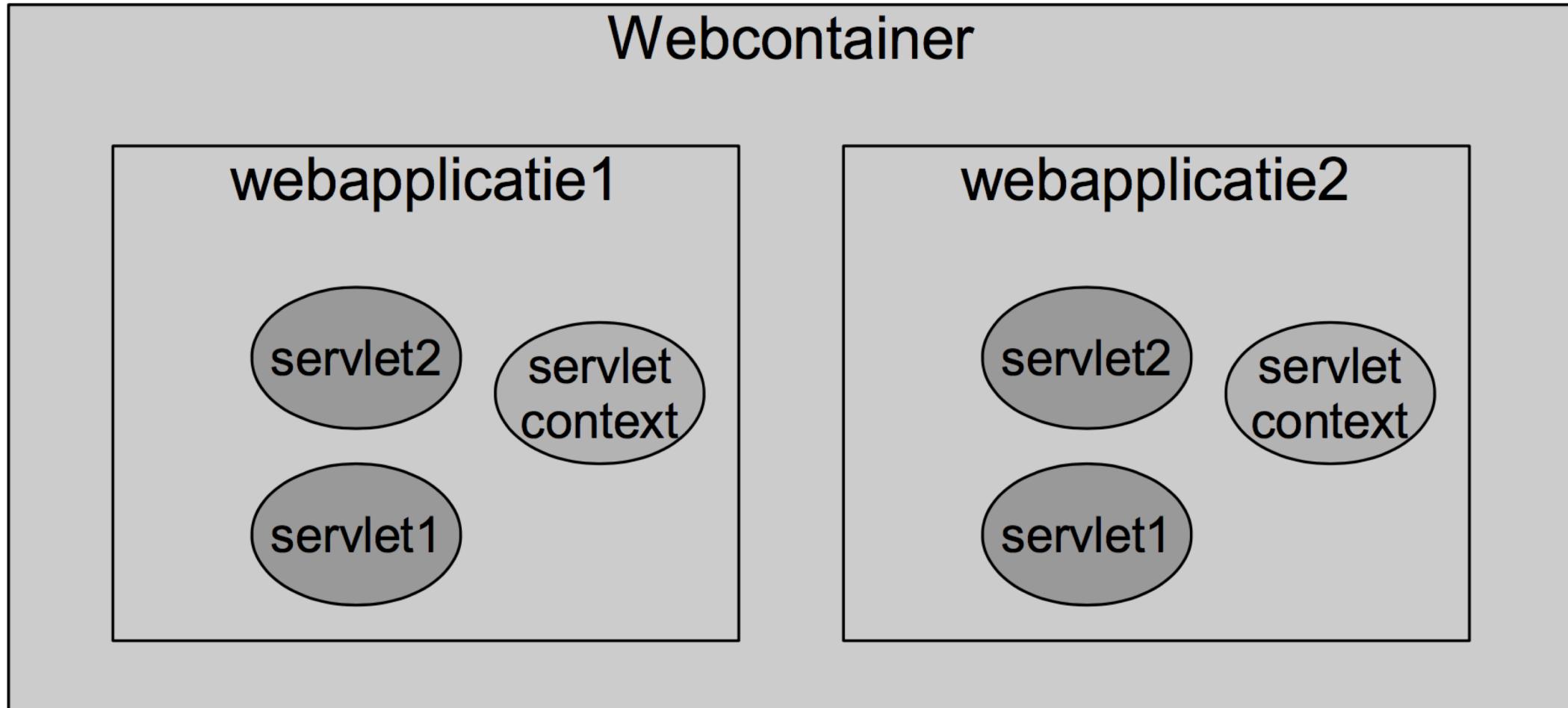
Sessie configuratie

- *setSessionTrackingModes* method op *servlet context*
- Via deployment descriptor in web.xml <session-config>
 - tracking-mode
 - session-timeout

```
...
<session-config>
    <tracking-mode>COOKIE</tracking-mode>
</session-config>
...
```

Webcomponenten – servlets - scope

Context (~ application scope)



Webcomponenten – servlets - scope

Context (~ application scope)

Methode	Omschrijving
<code>setAttribute()</code> <code>getAttribute()</code> <code>removeAttribute()</code> <code>getAttributeNames()</code>	Met deze methoden kunnen we attributen toevoegen, opvragen en verwijderen.
<code>getInitParameter()</code> <code>getInitParameterNames()</code>	Met deze methoden kunnen we de initialisatieparameters van de servlet context opvragen.
<code>getResource()</code> <code>getResourceAsStream()</code> <code>getResourcePaths()</code>	Met deze methoden kunnen we bestanden opvragen die zich binnen de webapplicatie (WAR) bevinden.
<code>getContext()</code>	Met deze methode kunnen we het <i>Servlet-context</i> -object van een andere webapplicatie opvragen.
<code>getRealPath()</code>	Met deze methode kunnen we het absolute pad opvragen van een bestand dat zich binnen de webapplicatie bevindt.
<code>getContextPath()</code>	Geeft het contextpad van de huidige webapplicatie.
<code>getSessionCookieConfig()</code>	Geeft de configuratie van de <i>session tracking cookie</i> .
<code>setSessionTrakingModes()</code>	Hiermee kan men de wijze van sessietracing instellen.
<code>addListener()</code>	Hiermee kan men programmatisch een <i>listener</i> instellen.
<code>addFilter()</code>	Hiermee kan men programmatisch een filter instellen.
<code>log()</code>	Hiermee kan men boodschappen naar het logbestand schrijven.

Webcomponenten – servlets - scope

- Session events
 - HttpSessionListener
 - sessionCreated
 - sessionDestroyed
- Context events
 - ServletContextListener
 - contextInitialized
 - contextDestroyed

Webcomponenten – servlets - scope

- Inleiding
- Servlets configureren / implementeren
- Lifecycle
- Service methoden (GET, POST, PUT, DELETE)
- Scope objecten
- **Servlet interactie**
- Overige

Webcomponenten – servlets - interactie

- **include**
 - insluiten van resultaat van een andere resource
→ `getServletContext().getNamedDispatcher(<servlet-name>).include()`
- **forward**
 - doorsturen request naar andere resource
 - wordt gebruikt bij MVC (voor views worden JSP's gebruikt)
 - `request.getRequestDispatcher(<page-name>).forward(request, response)`
- **redirect**
 - doorverwijzing via browser
 - doorverwijzing kan naar URL van andere webserver
 - `response.sendRedirect(<url>)`

Webcomponenten – servlets - scope

- Inleiding
- Servlets configureren / implementeren
- Lifecycle
- Service methoden (GET, POST, PUT, DELETE)
- Scope objecten
- Servlet interactie
- Overige

Webcomponenten – servlets - overige

- Multithreaded
 - meerdere verzoeken tegelijkertijd = meerdere threads
- Cookies
 - request.getCookies()
 - response.addCookie(...)
- Foutafhandeling
 - ServletException
 - IOException
 - foutpagina configureren in web.xml

```
<error-page>
    <exception-type>javax.servlet.annotation.WebServlet</exception-type>
    <location>/error-page.html</location>
</error-page>

<error-page>
    <error-code>404</error-code>
    <location>/error-page-404.html</location>
</error-page>
```

Beers - servlet

- Maak een BeerServlet met volgende functionaliteit

- /beers

- Geef alle bieren uit de table Beers terug in een lijst

- /beers/?id=...

- Geef van het bier met de meegegeven id alle gegevens terug

- Geef een error page terug als het bier niet gevonden wordt

Opmaak is hier van minder belang!