Levensloop servlet

Veerle Ongenae

Servlets

- Basisbegrippen
- Sessies
- ServletContext
- Doorsturen HTTP-aanvragen
- Levensloop

Webcontainer

- Java Runtime
- Implementatie Java Servlet API
- Laden, oproepen, initialiseren, … van servlets

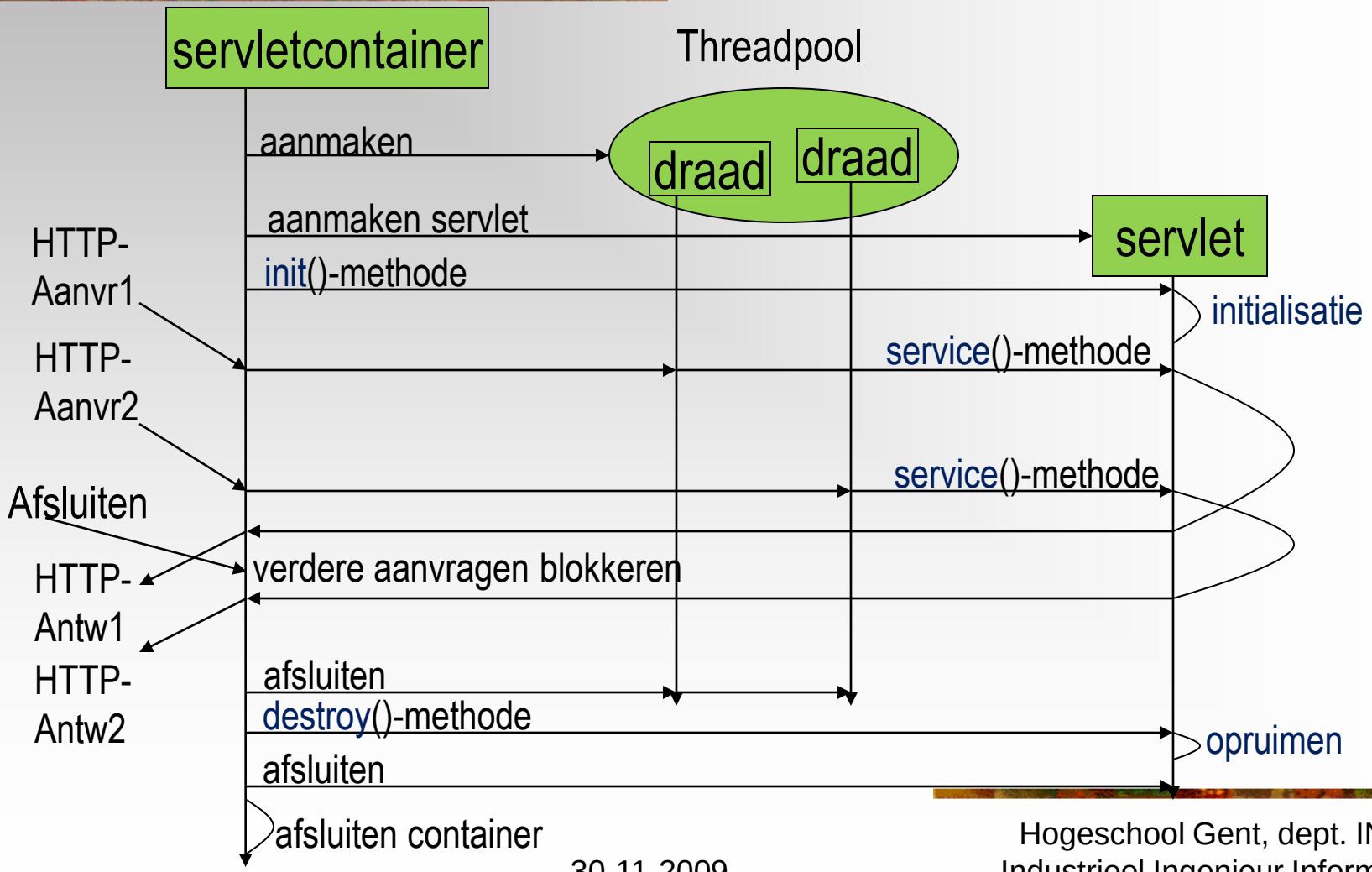
Verwerking HTTP-aanvraag

- Webserver: aanvraag correspondeert met applicatie in webcontainer
- Webcontainer bepaalt wie de aanvraag moet afhandelen (JSP, servlet, bestand)
- Webcontainer bepaalt servlet-object of maakt één aan
- Webcontainer delegeert aanvraag naar servlet en geeft `HttpServletRequest`- en `HttpServletResponse`-objecten mee

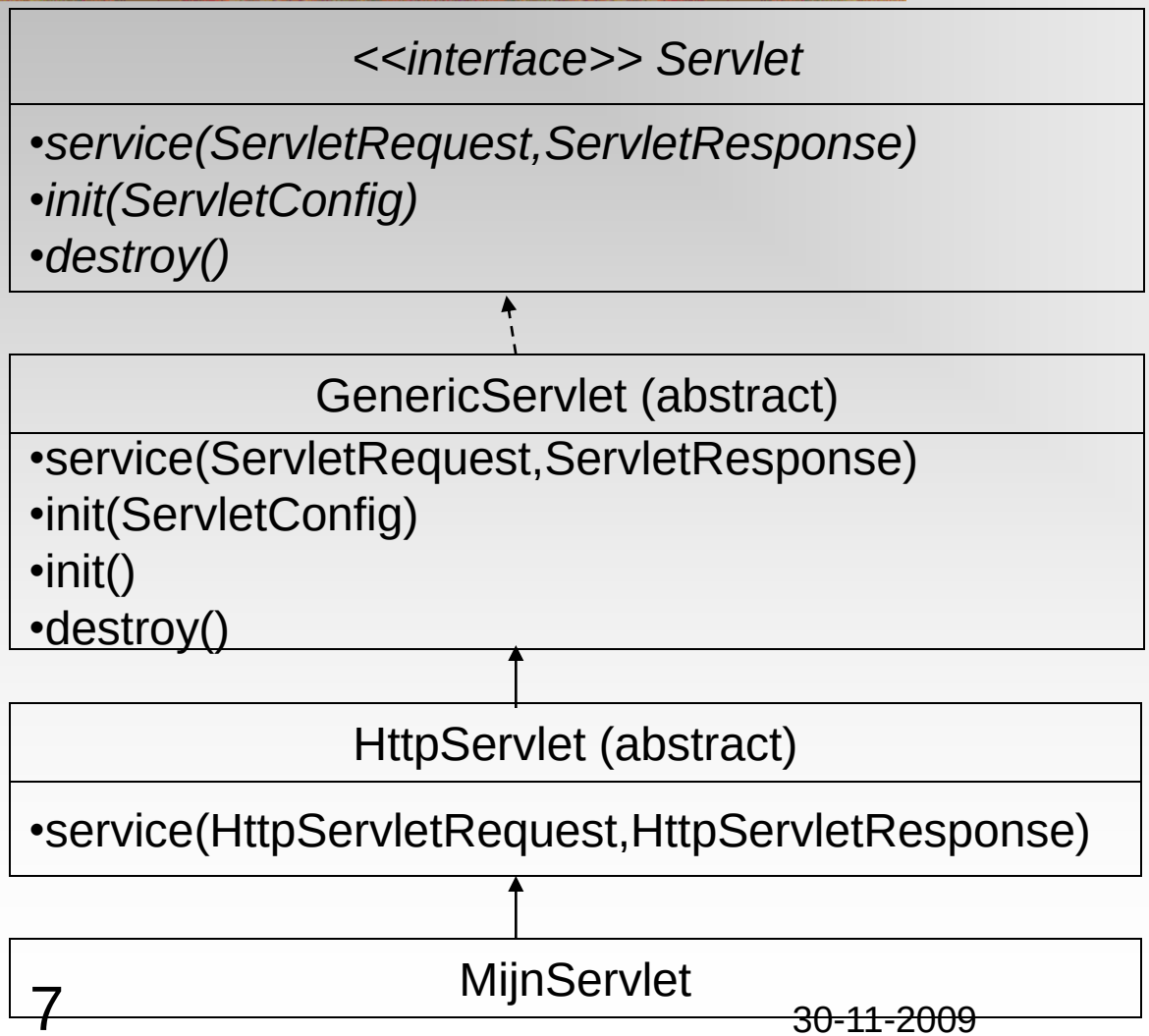
Levensloop van een servlet

- Laden, servlet-object aanmaken en initialiseren servlet
- Nul of meerdere aanvragen afhandelen
- Opkuisen en vernietigen servlet-object

Figuur


servletcontainer
Threadpool
draad
draad
servlet
aanmaken
aanmaken servlet
init()-methode
initialisatie
HTTP-
Aanvr1
HTTP-
Aanvr2
service()-methode
service()-methode
Afsluiten
verdere aanvragen blokkeren
HTTP-
Antw1
HTTP-
Antw2
afsluiten
destroy()-methode
opruimen
afsluiten
afsluiten container

Methodes levensloop

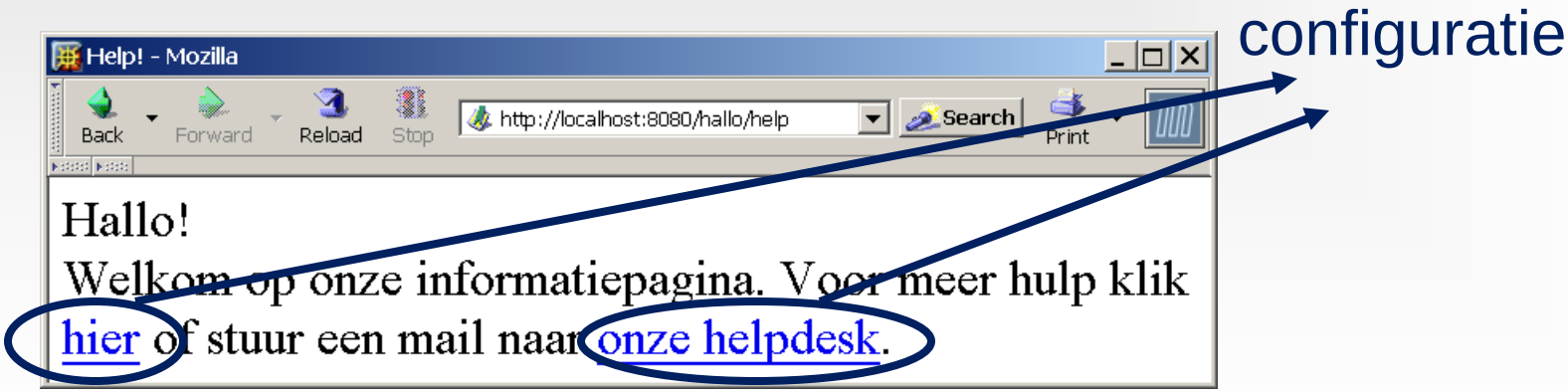


Initialisatie

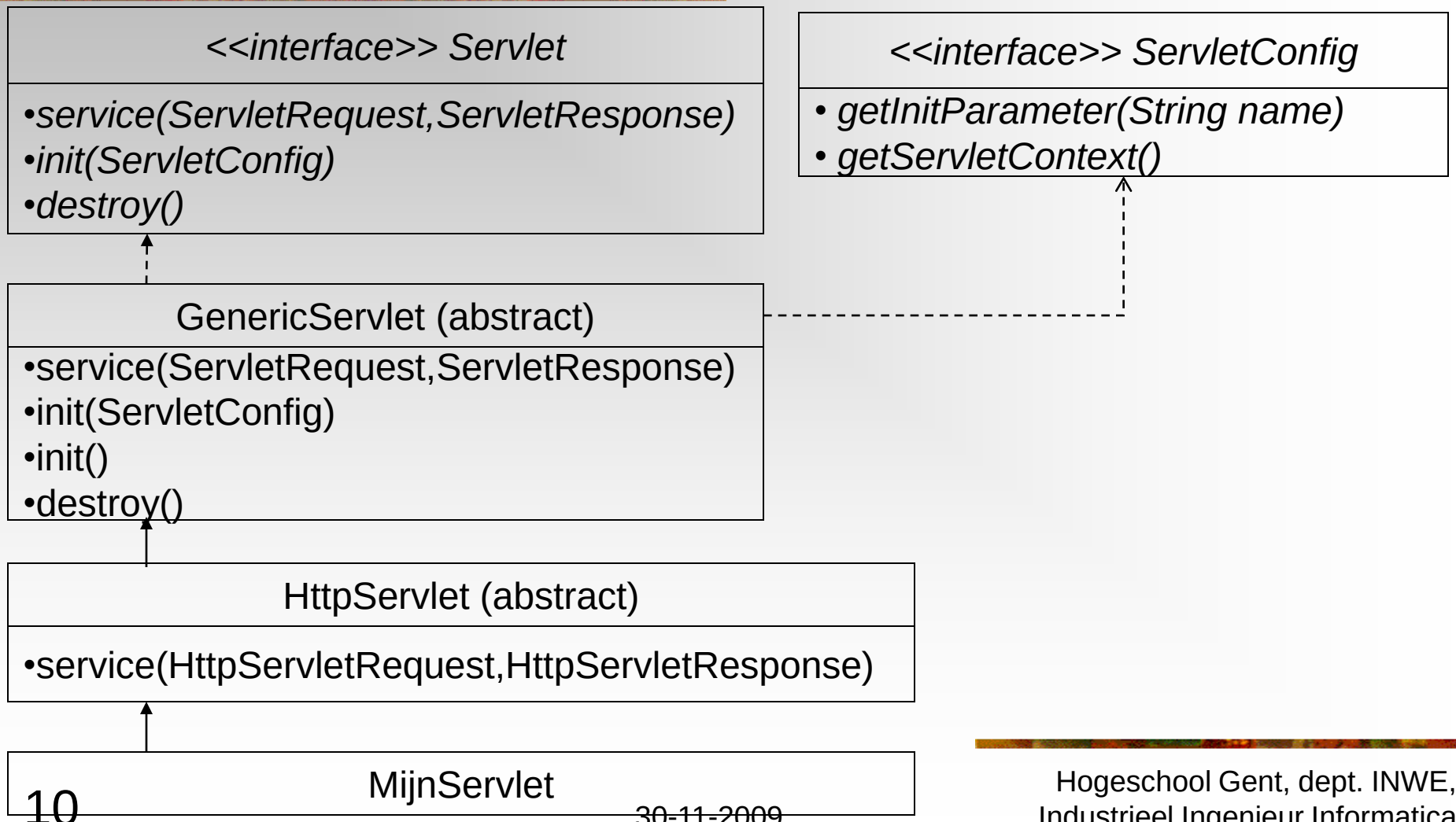
- init()–methode implementeren
- Wat?
  - Configuratie inlezen uit configuratiebestanden
  - Initialisatieparameters bepalen
  - Referentie naar objecten ophalen, …
  - Registreren gegevensbank, logbestand, connection pool, …
- Bij falen: UnavailableException, **NOOIT System.exit(…)**

Voorbeeld

- http://localhost:8080/hallo/help
  - Link: URL in configuratie (web.xml)
  - Email: Adres in configuratie (web.xml)
- be.hogent.iii.servlets.HelpServlet



Methodes levensloop



Initialisatie servlet

- Methode implementeren

  public void **init**() throws ServletException

- Werkwijze
  - Ophalen waarde initialisatieparameters
  - Methode interface ServletConfig
    - Geïmplementeerd door GenericServlet

  public String **getInitParameter**(String name)

  - Initialisatieparameter in het bestand web.xml

Configuratie: web.xml

```
<servlet>
    <servlet-name>info</servlet-name>
    <servlet-class>
      be.hogent.iii.servlets.HelpServlet</servlet-class>
    <init-param>
        <param-name>helppagina</param-name>
        <param-value>index.html</param-value>
    </init-param>
    <init-param>
        <param-name>helpemail</param-name>
        <param-value>help@info.be</param-value>
    </init-param>
 </servlet>
```

Parameters en attributen

- Parameters
  - String/String-paren
  - Worden niet veranderd
  - HTTP-parameters
    - Webclient → webserver
    - HTML-formulier
  - Contextparameters
    - Configuratie webapplicatie (web.xml)
    - Op ServletContext
  - Initialisatieparameters
    - Configuratie één servlet (web.xml)
    - ServletConfig
- Attributen
  - String/Object-paren
  - Aanpasbaar
  - Scope
    - Request
      - Verschillend per aanvraag
      - Servlets die aanvraag afhandelen
    - Session
      - Op sessie-object
      - Verschillend per gebruiker
    - Application
      - Op ServletContext-object
      - Zelfde voor alle gebruikers

```
public class MyServlet ... {
    ...
    public void init() throws ServletException {
        // bv. verbinding maken met een gegevensbank
        try {
            ...
        } catch(Exception e) {
            throw new UnavailableException ("Kan
              geen verbinding maken met de
              gegevensbank");
        }
    }
    ...
}
```

Servlet niet beschikbaar

- Niet beschikbaar

public **UnavailableException**(String msg)

- Tijdelijk niet beschikbaar

public **UnavailableException**(String msg,
int seconds)

Opruimen servlet

- De container aanvaardt geen aanvragen meer voor de servlet
- De container wacht tot alle actieve draden die service-methode van de servlet afhandelen beëindigd zijn
- De container voert de destroy-methode van het servletobject uit
- De referentie naar het servletobject wordt verwijderd en het servletobject is klaar voor garbage collection

Opkuisen

- destroy()-methode
- Wat?
  - Afsluiten langdurige processen, …

Vervolg overzicht API

- Abstracte klasse GenericServlet

```
public String
   getInitParameter(String name);
public void init() throws ServletException;
public void destroy();
public abstract void
   service(ServletRequest req,
   ServletResponse res) throws
   ServletException, IOException;
```

Vervolg overzicht API

- Abstracte klasse HttpServlet

  protected void **service**(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException;

  - Doorsturen aanvraag naar de juiste doXXX-methode

  public void **service**(ServletRequest req, ServletResponse res) throws ServletException, IOException;

  - Doorsturen aanvraag naar “protected service-methode”
  - **NIET OVERSCHRIJVEN**

Voorbeeld parallel uitvoeren service methode

- http://localhost:8080/hallo/tel1
  - De waarde van de teller wijzigt tijdens het uitschrijven van de pagina
- http://localhost:8080/hallo/tel2
  - De waarde van de teller wijzigt niet tijdens het uitschrijven van de pagina
  - Er wordt slechts één pagina tegelijk aangemaakt, de andere moeten wachten

Voorbeeld

- be.hogent.iii.servlets.SlaapEnTelServlet
- be.hogent.iii.servlets.SlaapEnTelServlet2

Oplossing ?

- synchronised methode
  - Unieke lock met elk object
  - Oproepen methode -> blokkeert object
  - Einde methode -> lock verwijderd
- Let op
  - Verschillende servlet-objecten
- Externe objecten (bv. op ServletContext) met gesynchroniseerde methodes
- Synchroniseren op de ServletContext
- Gebruik geen globale variabelen