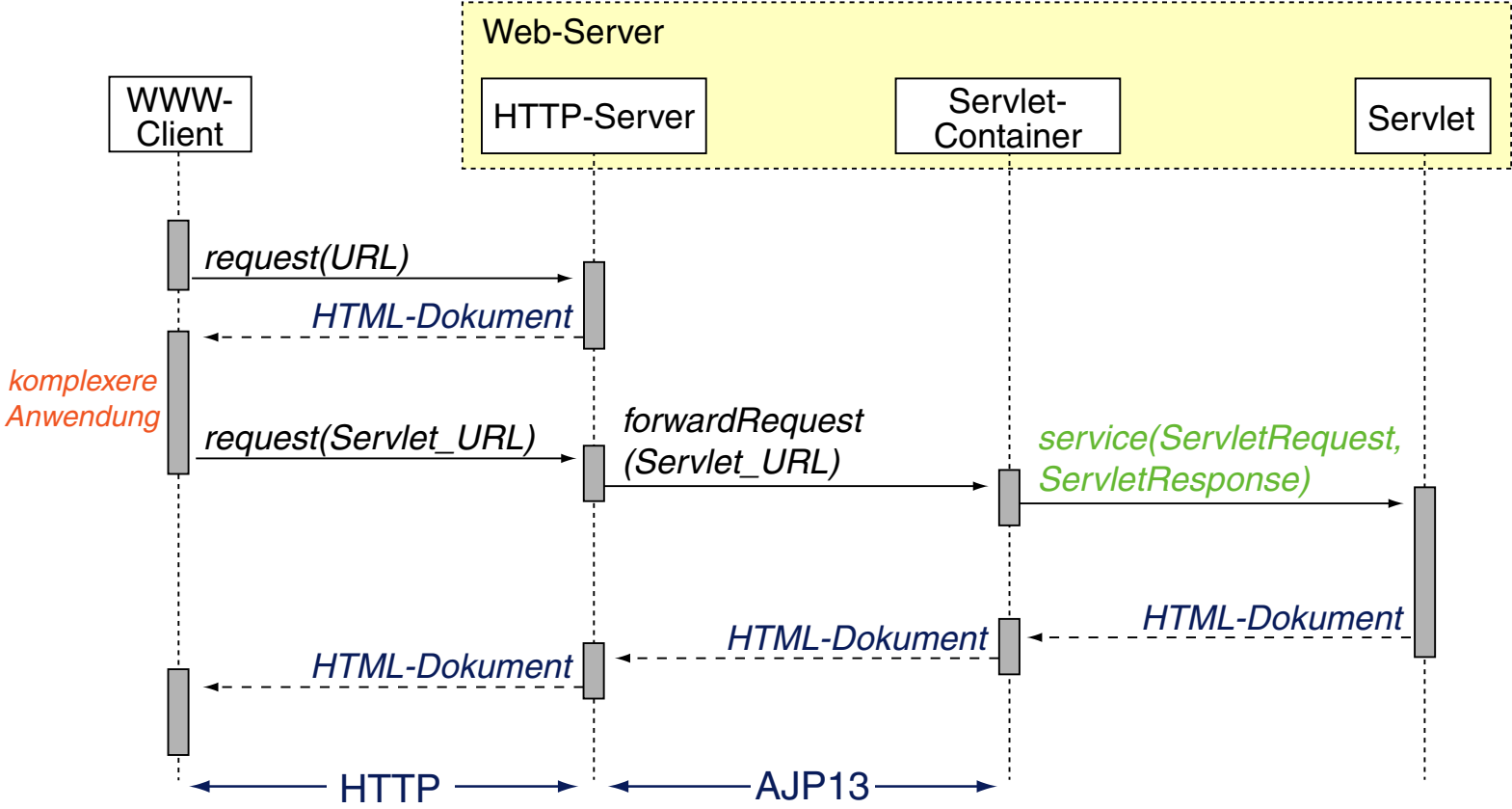


# Apache Tomcat

## Servlet-Interaktion



Für diese Übung nutzen wir Tomcat zugleich als HTTP-Server und Servlet-Container.

# Apache Tomcat

## Tomcat starten

Laden Sie ein Tomcat Core Archiv von [der Tomcat-Webseite](#) herunter und entpacken Sie es

### ❑ Unix /MacOS

- Eventuell fehlende Ausführungsrechte setzen:

```
cd apache-tomcat-*/bin
```

```
chmod +x startup.sh shutdown.sh catalina.sh
```

- Starten:

```
./startup.sh oder bash ./startup.sh
```

### ❑ Windows

- Starten:

```
cd apache-tomcat-*\bin
```

```
startup.bat
```

Tomcat bedient standardmäßig Anfragen auf Port 8080.

# Apache Tomcat

## Deployment Beispiel: ServletHelloWorld

Laden Sie die `helloworld.war` von der Kursseite herunter und legen Sie dieses in den `webapps`-Ordner des entpackten Tomcat Archivs.

Tomcat entpackt das Archiv automatisch:

```
webapps/helloworld/WEB-INF/web.xml
```

```
webapps/helloworld/WEB-INF/classes/servlet/ServletHelloWorld.class
```

Öffnen Sie <http://localhost:8080/helloworld/HelloWorld> im Browser:

# Hello World!

# Apache Tomcat

## Deployment Beispiel: ServletHelloWorld

```
package servlet;

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

public class ServletHelloWorld extends HttpServlet {

    public void init() throws ServletException {
        // Nothing to do here.
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Use "response" to specify the HTTP response line and headers
        // (e.g. specifying the content type, setting cookies).
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        // Use "out" to send content to browser.
        out.print("<!DOCTYPE html>"
            + "<html><head><title>Hello World</title></head>"
            + "<body><h1>Hello World!</h1></body></html>");
    }
}
```

# Apache Tomcat

## Deployment Beispiel: ServletHelloWorld

http	Anfrage per HTTP ...
localhost	... an diesen Rechner ...
:8080	... an Prozess unter Port 8080 (Tomcat).
helloworld	<b>Context-Path</b> der anzufragenden Web-Anwendung
/ServletHelloWorld	<b>Anfrage</b> an die Web-Anwendung

→ Konfiguration der Web-Anwendung in `webapps/helloworld/WEB-INF/web.xml` :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app ...>
  <servlet>
    <servlet-name>ServletHelloWorld</servlet-name>
    <servlet-class>servlet.ServletHelloWorld</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ServletHelloWorld</servlet-name>
    <url-pattern>/HelloWorld</url-pattern>
  </servlet-mapping>
</web-app>
```

→ Nutzung des Servlets

`webapps/helloworld/WEB-INF/classes/servlet/ServletHelloWorld.class`

# Apache Tomcat

## Extra-Übung

Optional: Testen Sie ihr Verständnis des Deployment-Prozesses

- ❑ Laden Sie die `ServletHelloWorld.java` von der Kursseite herunter
- ❑ Verändern Sie das Servlet, so dass es den Text "Goodbye World!" ausgibt
- ❑ Deployen und testen Sie das veränderte Servlet

Zum kompilieren von Servlets benötigen Sie die Bibliothek `javax.servlet-3.0.jar`. Diese finden Sie zum Beispiel im entpackten Tomcat-Archiv im Ordner `lib`. Geben Sie diese beim kompilieren im Classpath an:

```
javac -classpath ../javax.servlet-3.0.jar servlet/ServletHelloWorld.java
```