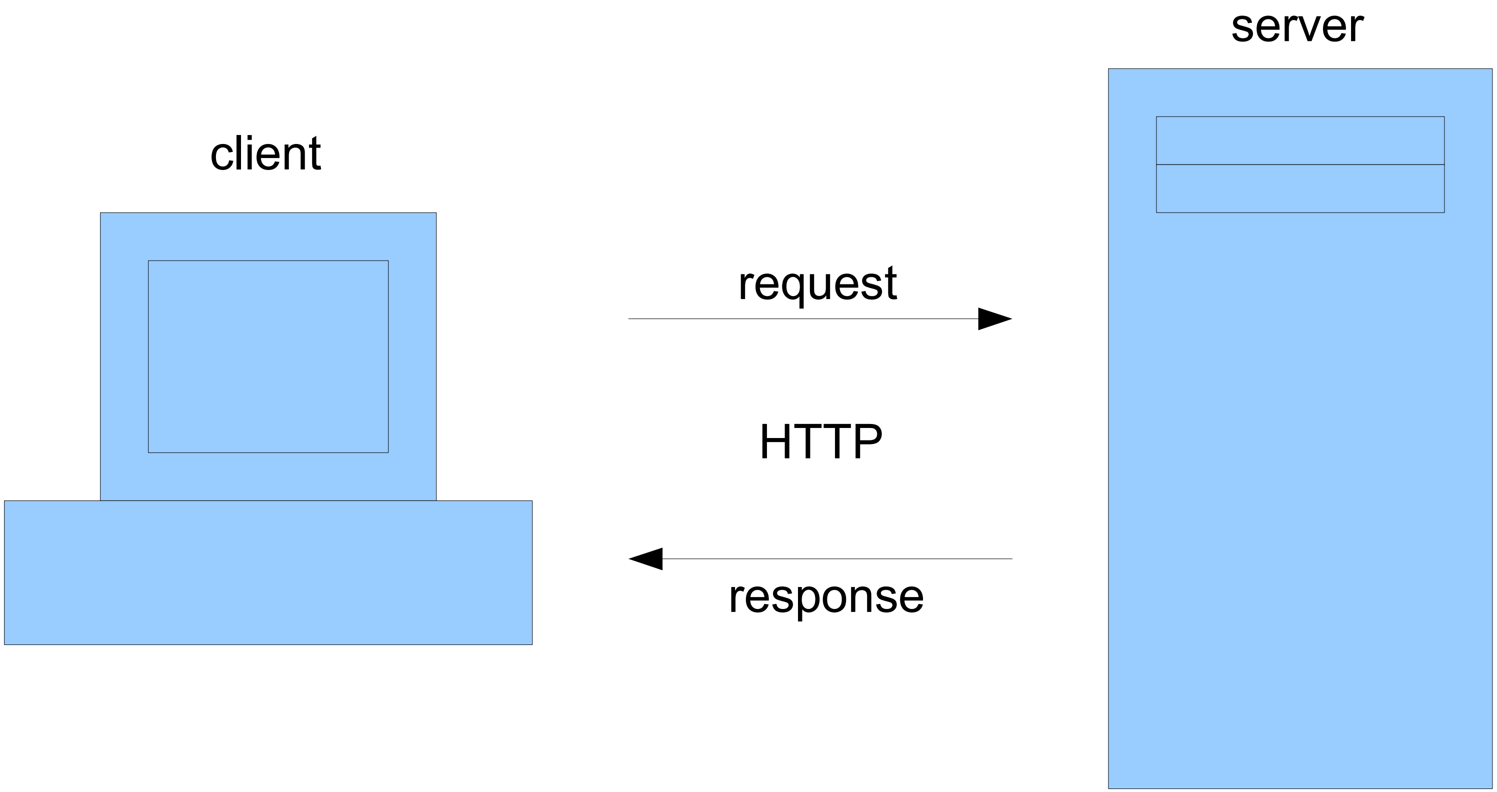
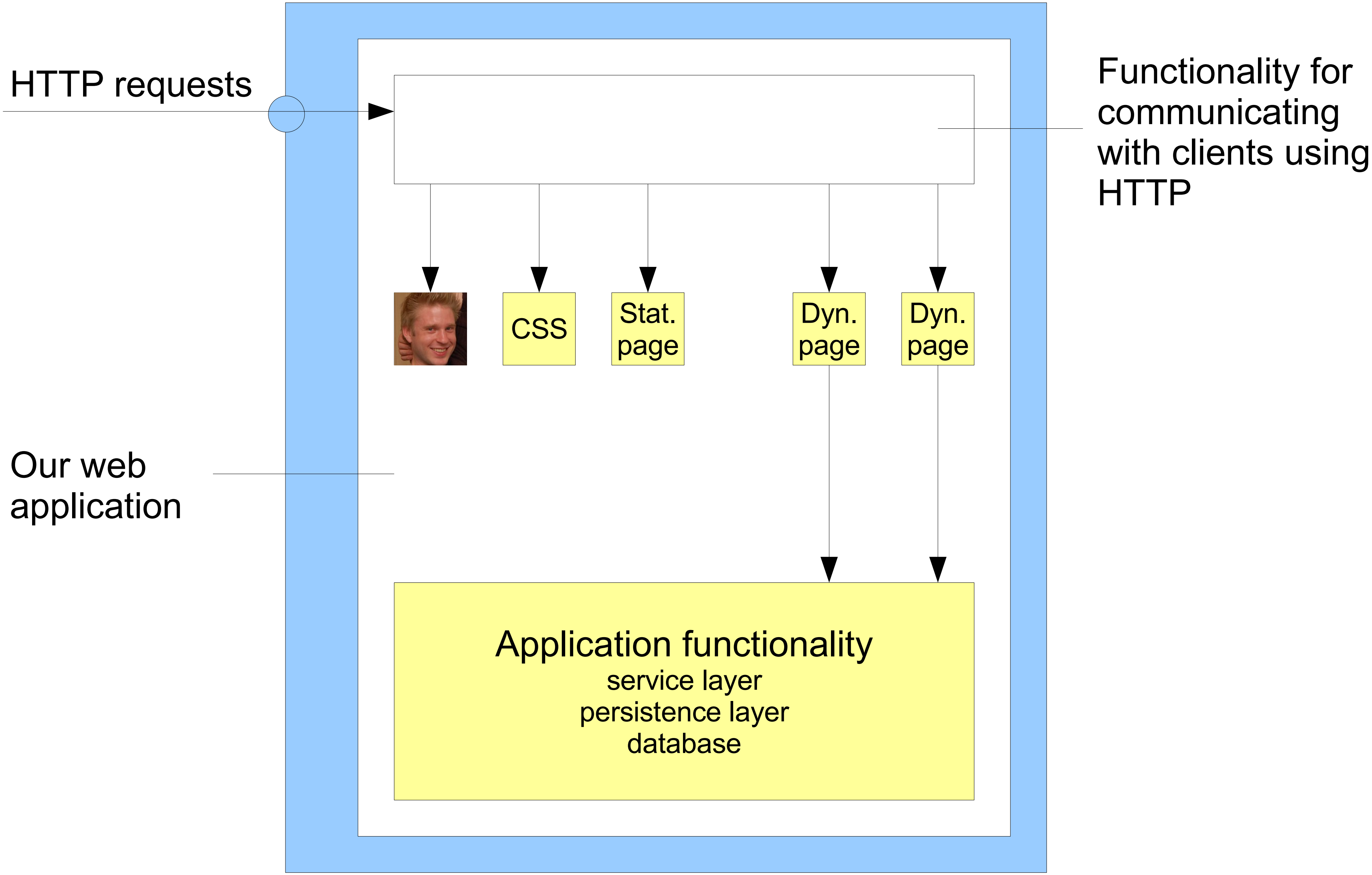
Java and Web

WebWork

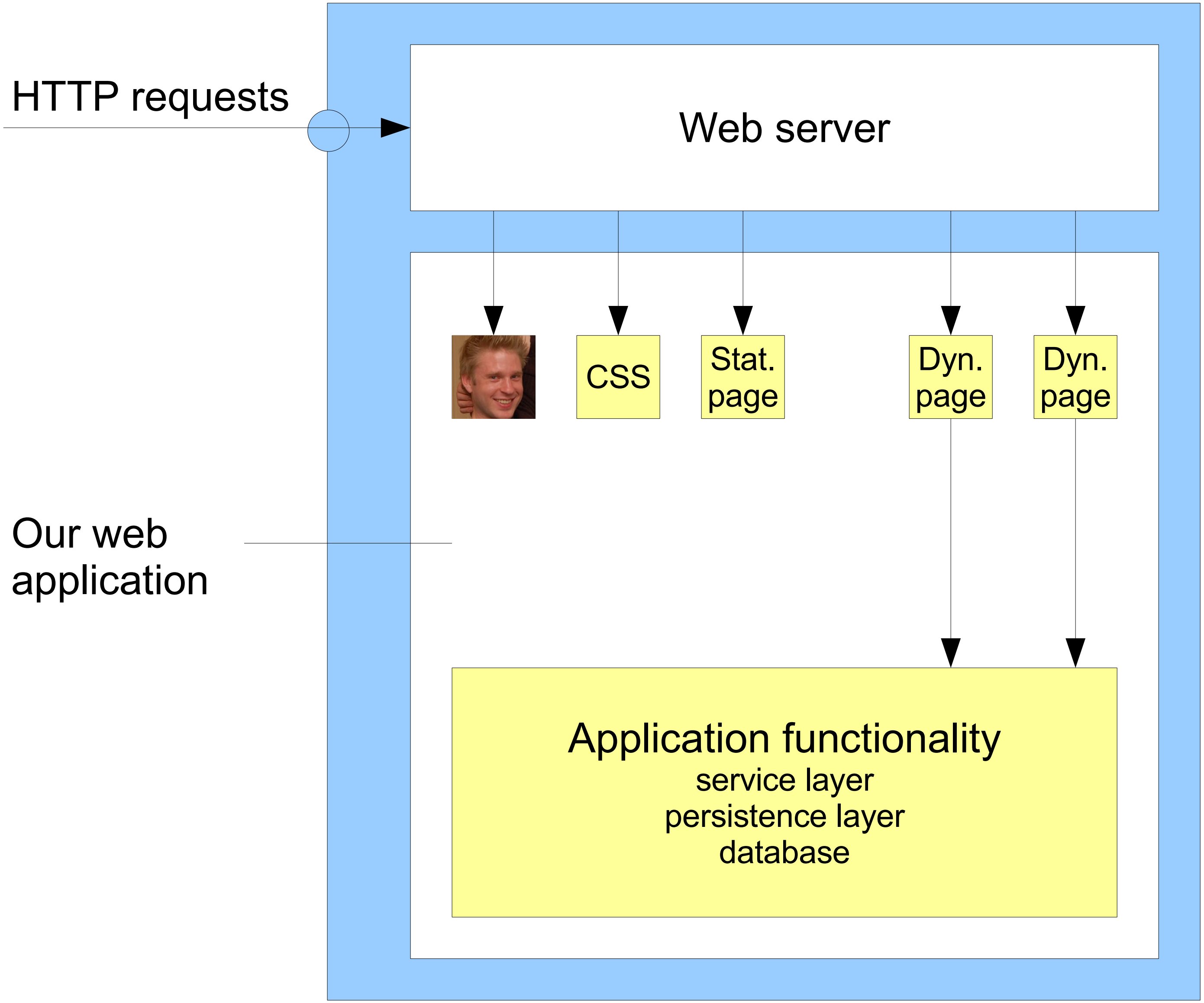
Client/Server



Inside the Server (1)



Inside the Server (2)



How does the web server know how to access resources in our application?

Meet:

Java Servlet Specification (JSR154)

Java Servlet API

A **servlet** is a Java(TM) technology-based Web component, managed by a container, that generates dynamic content. Like other Java technology-based components, servlets are platform-independent Java classes that are compiled to platform-neutral byte code that can be loaded dynamically into and run by a Java technology-enabled Web server. Containers, sometimes called servlet engines, are Web server extensions that provide servlet functionality. Servlets interact with Web clients via a request/response paradigm implemented by the servlet container.

The **servlet container** is a part of a Web server or application server that provides the network services over which requests and responses are sent, decodes MIME-based requests, and formats MIME-based responses. A servlet container also contains and manages servlets through their lifecycle.

Servlets (1)

```
package javax.servlet;

public interface Servlet
{
    public void service(ServletRequest req, ServletResponse res);

    // .. other methods
}
```

Servlets are generic, but are mostly used in conjunction with HTTP

The Servlet API has one generic package and one HTTP specific package:

```
javax.servlet;
javax.servlet.http;
```

Servlets (2)

```
package javax.servlet.http;

public abstract class HttpServlet extends GenericServlet
{
    protected void doGet(HttpServletRequest req, HttpServletResponse res);

    protected void doPost(HttpServletRequest req, HttpServletResponse res);

    // .. other methods
}
```

GenericServlet implements the servlet interface Servlet

The servlets are singletons and instantiated by the servlet container

Servlet Mapping

The web application is required to have a *web.xml* file in a specific location. The web.xml file can contain URL-to-servlet mappings

The servlet container reads the web.xml file and transforms and forwards requests according to the mappings

web.xml:

```
<web-app>
  ...

  <servlet>
    <servlet-name>feedback-servlet</servlet-name>
    <servlet-class>no.uio.inf5750.app.servlet.Feedback</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>feedback-servlet</servlet-name>
    <url-pattern>/feedback</url-pattern>
  </servlet-mapping>

  ...
</web-app>
```

Web Archive

WAR file: A zip file which can contain web resources and dependent libraries in addition to the contents of a regular JAR file

Structure of a WAR file:

/	Web resources (images, CSS files, etc)
/WEB-INF/	Private files and directories (not to be returned by the web server)
/WEB-INF/web.xml	web.xml
/WEB-INF/classes/	Classes and application resources (typical contents of a JAR file)
/WEB-INF/lib/*.jar	Dependent libraries

```
/index.html
/styles.css
/images/me.png
/WEB-INF/web.xml
/WEB-INF/classes/no/uio/inf5750/app/servlet/Feedback.class
/WEB-INF/lib/my-app-func.jar
/WEB-INF/lib/servlet-api.jar
/WEB-INF/lib/hibernate.jar
```

Other Mappable Classes

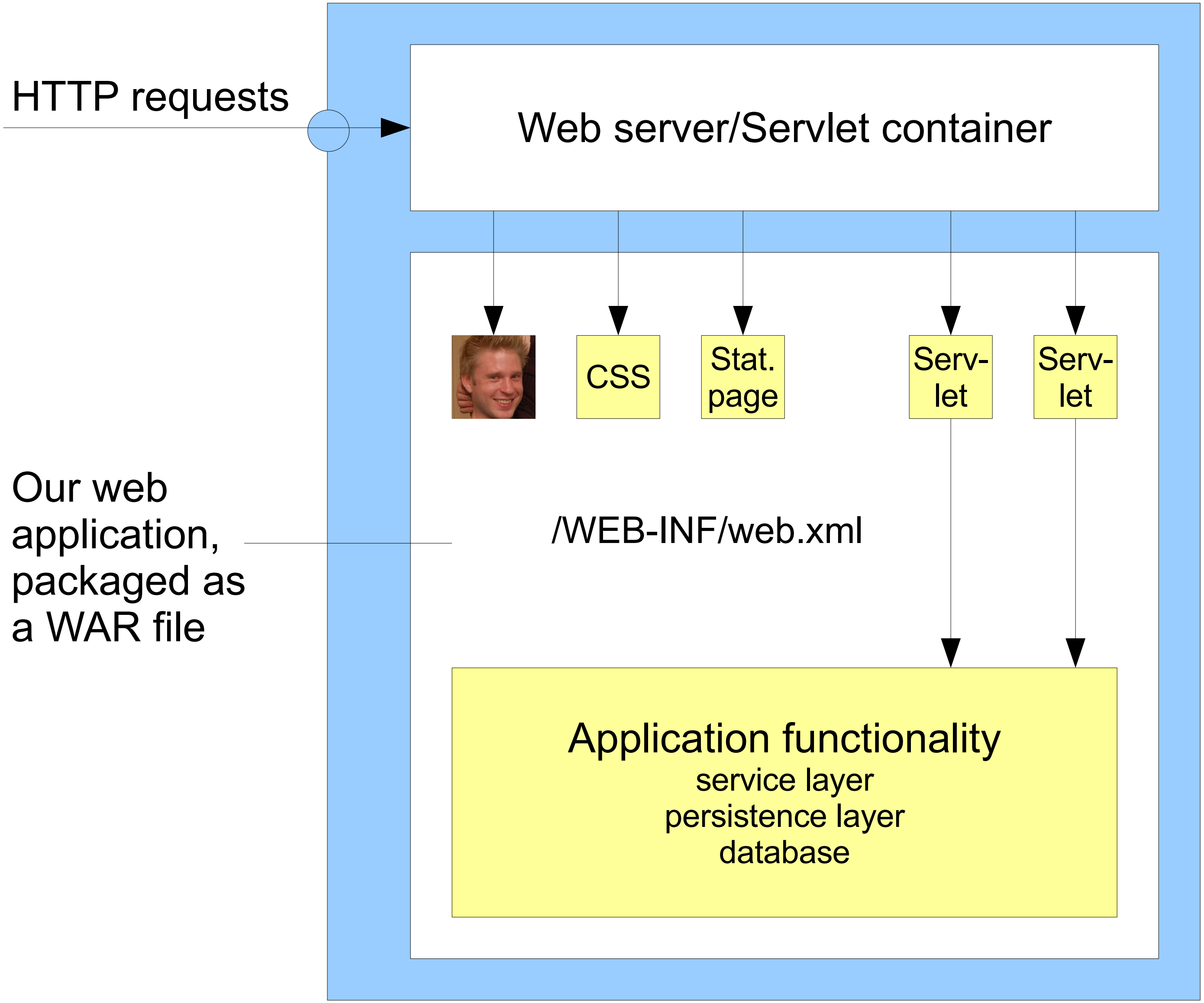
javax.servlet.Filter

“Around interceptors” for servlet requests

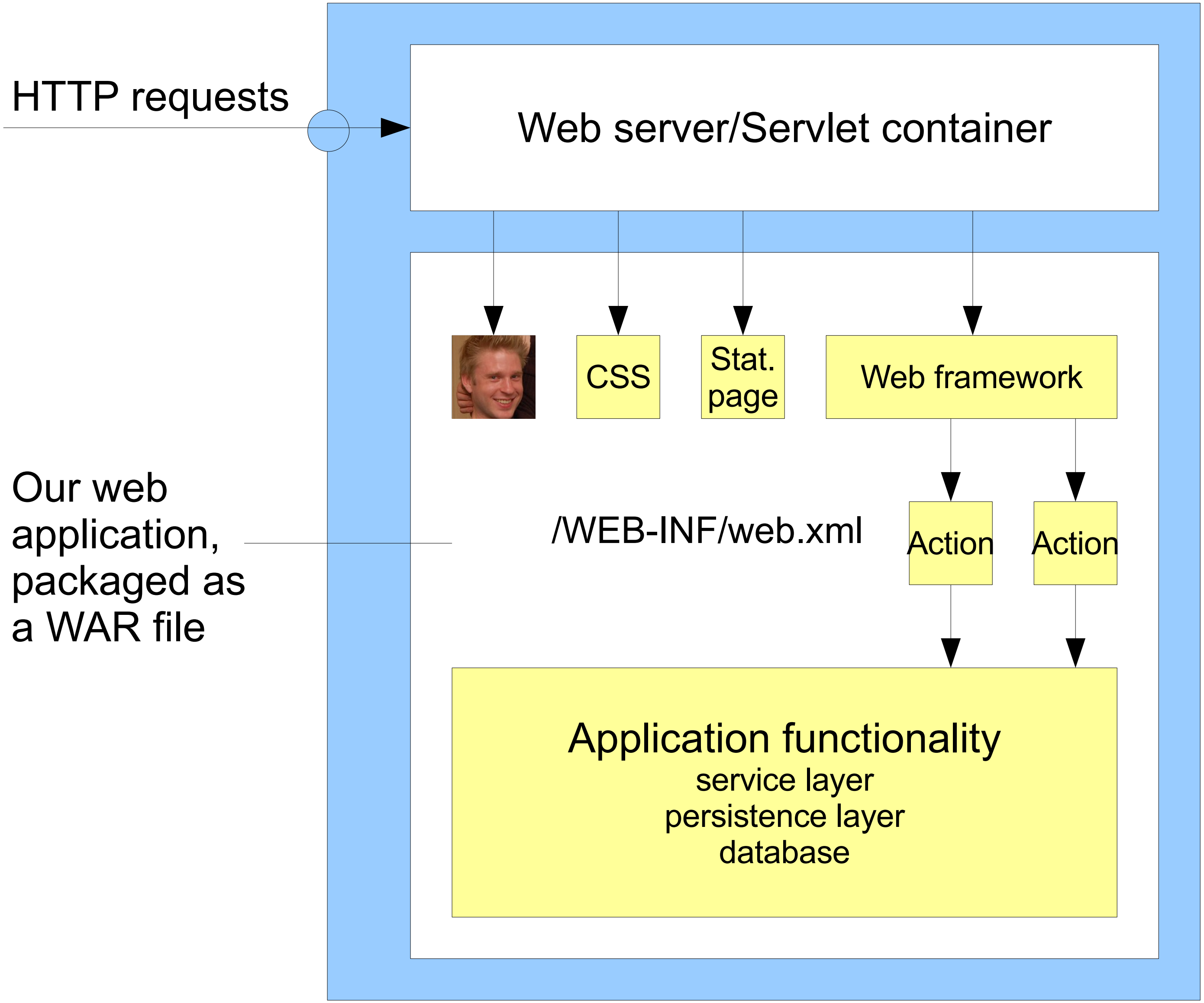
Various listeners

Context listeners
HTTP session listeners

Inside the Server (3)



Inside the Server (4)



WebWork (1)

Latest version is 2.2.6 - www.opensymphony.com/webwork

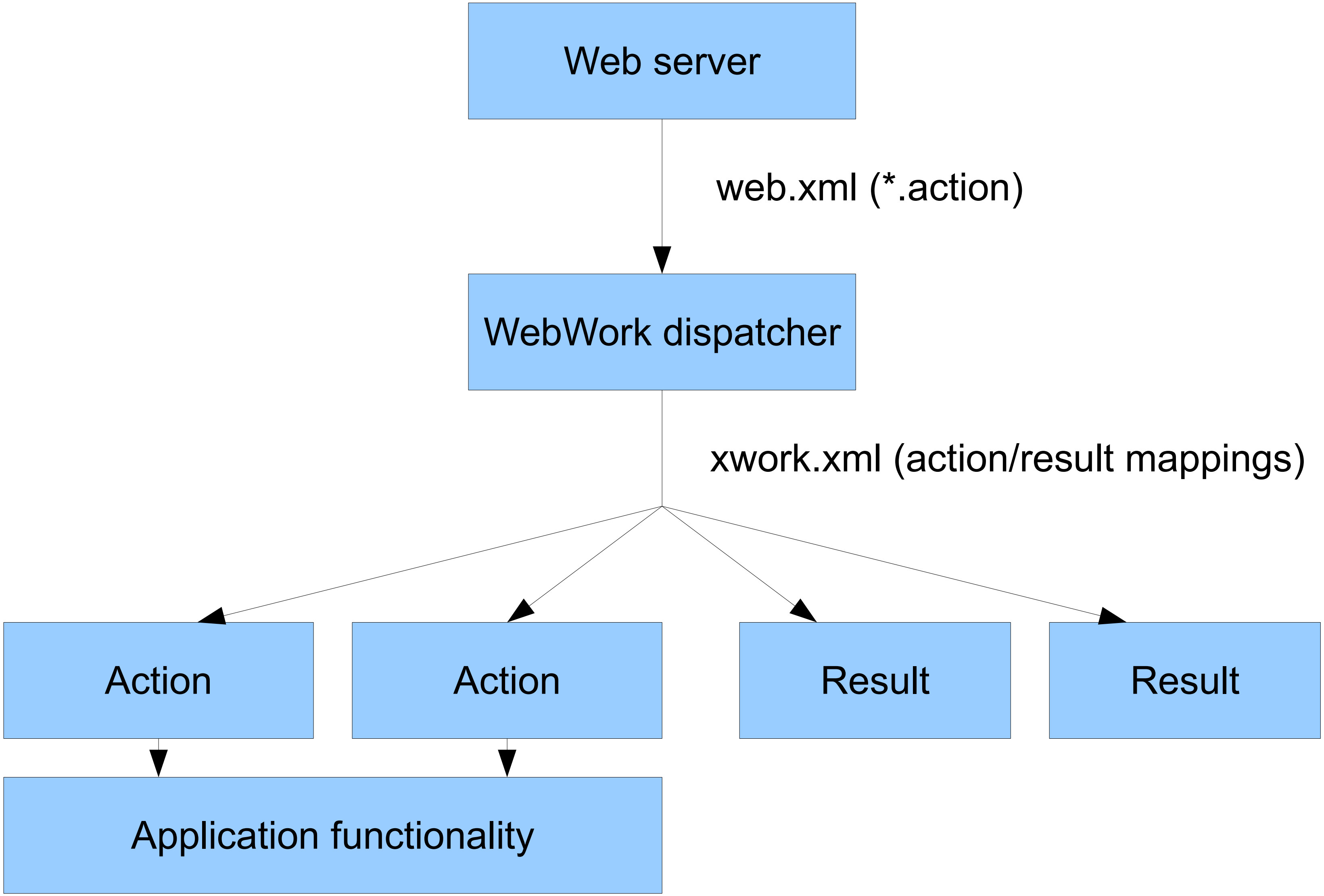
Based on XWork (command pattern framework)

Redirects requests to action classes, and returns a result based on the outcome of the action class execution

Supports:

validation
type conversion
IoC
interceptors
i18n
views/result types (Velocity, FreeMarker, JSP, ...)
modularization - namespaces

WebWork (2)



Web server

web.xml (*.action)

WebWork dispatcher

xwork.xml (action/result mappings)

Action

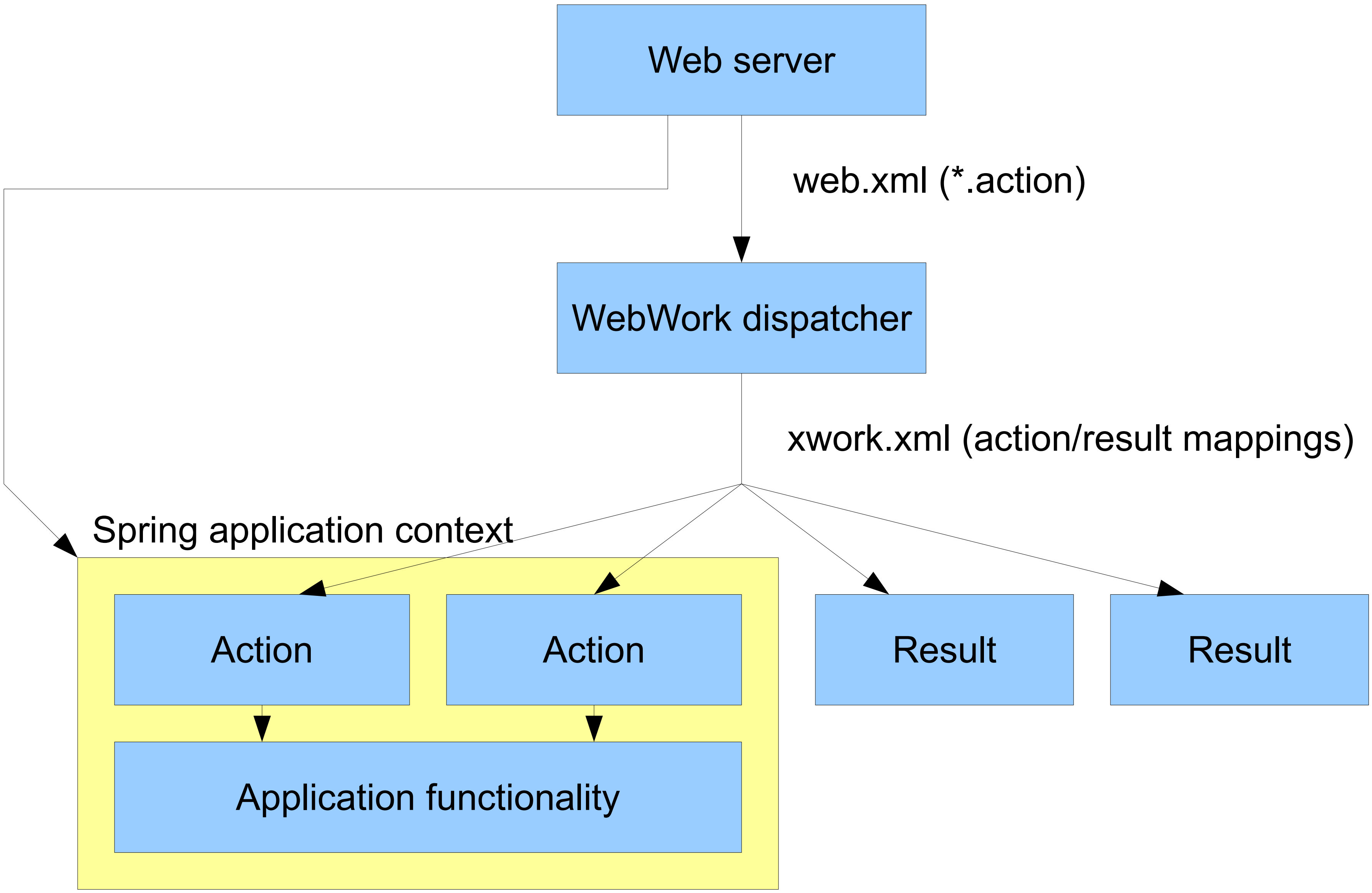
Action

Result

Result

Application functionality

WebWork and Spring


Web server
web.xml (*.action)
WebWork dispatcher
xwork.xml (action/result mappings)
Spring application context
Action
Action
Result
Result
Application functionality

Web Servers/Servlet Containers

Tomcat

Jetty

Wikipedia lists 23 servlet containers:
http://en.wikipedia.org/wiki/Java_Servlet#Servlet_containers

WebWork Example

Downloadable from the Teaching plan

http://www.uio.no/studier/emner/matnat/ifi/INF5750/h07/undervisningsplan.xml

Rules for Web Modules in DHIS 2

Must depend on

dhis-web-commons (jar)
dhis-web-commons-resources (war, <type>war</type>)

XWork package must

include and extend dhis-web-commons instead of webwork-default
have the same name as the artifactId
have the same namespace as the artifactId (with a leading /)

webwork.properties is defined centrally

Typical Action Mapping in DHIS 2

```
<action name="dataElement"
    class="org.hisp.dhis.dd.action.dataelement.GetDataElementListAction">

  <result name="success" type="velocity">/main.vm</result>

  <param name="page">
      /dhis-web-maintenance-datadictionary/dataElement.vm</param>

  <param name="menu">
      /dhis-web-maintenance-datadictionary/menu/MainMenu.vm</param>

  <param name="javascripts">
      javascript/dataElement.js,
      javascript/filterTable.js
  </param>

  <interceptor-ref name="transactionStack"/>

</action>
```

Questions?