

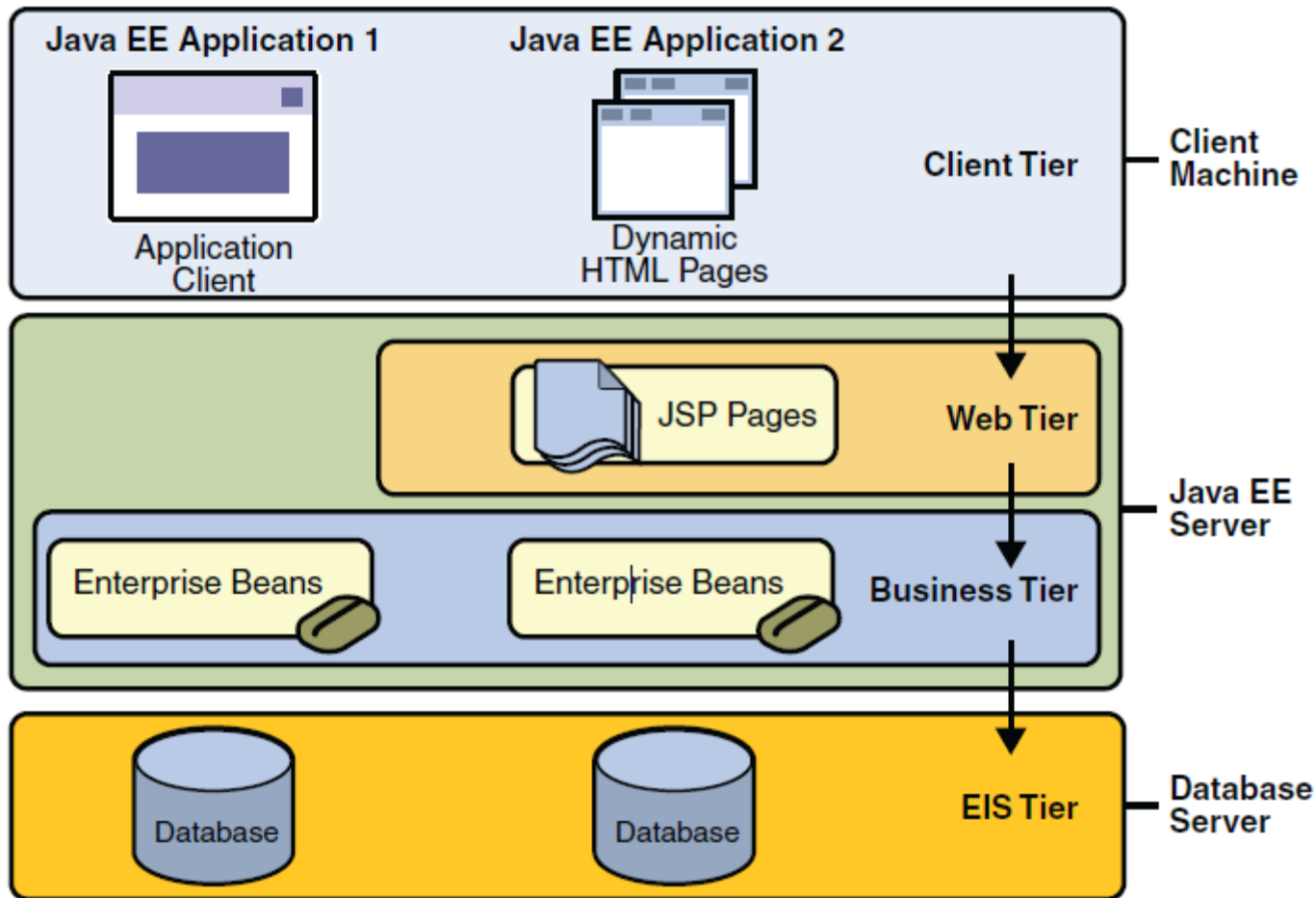
DEZVOLTAREA APLICATIILOR WEB

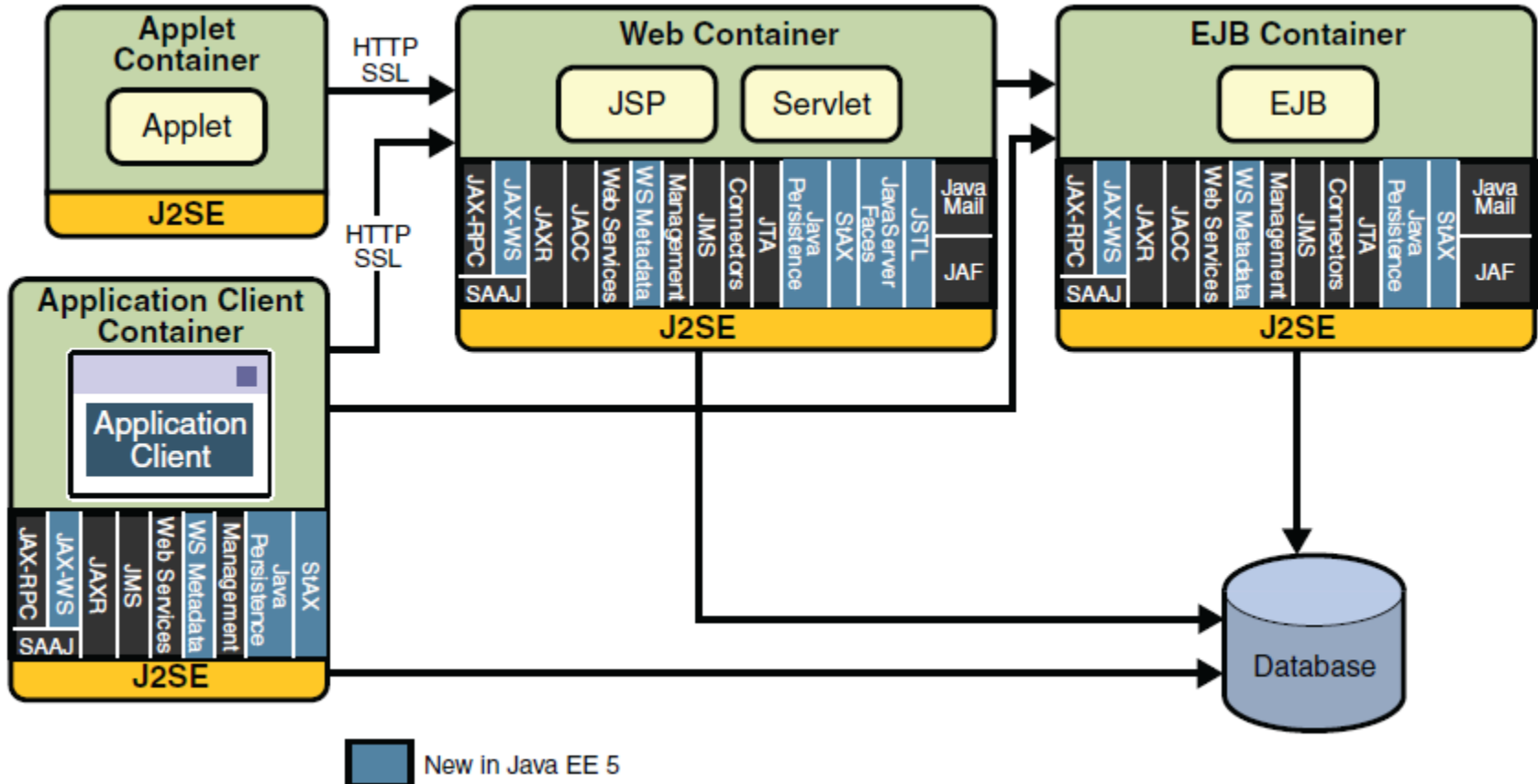
CURS 2

Lect. Univ. Dr. Mihai Stancu

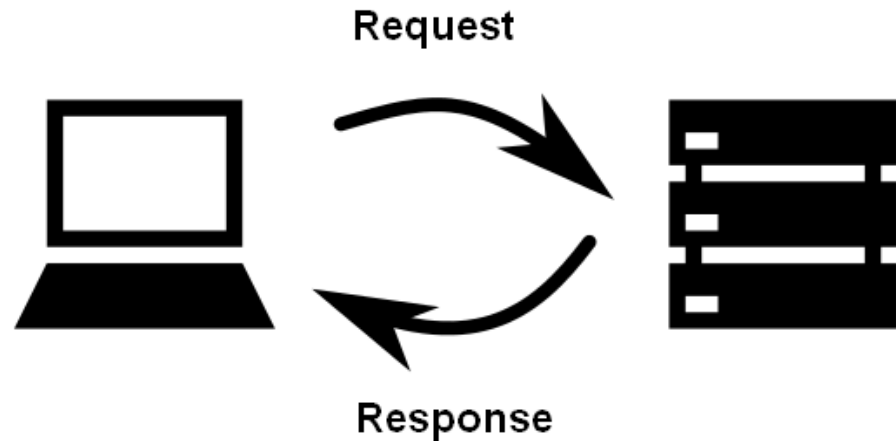
- suport (The Java EE 5Tutorial)
 - Capitolul 4 – Java Servlet Technology

➤ Distributed Multitiered Applications





- Ciclul de viață al unui Servlet
- Partajarea informațiilor
- Inițializarea unui Servlet
- Scrierea metodelor de Service
- Filtrarea cererilor și răspunsurilor (request / response)
- Apelul către alte resurse Web
- Accesarea contextului Web
- Menținerea stării clientului (sessions)
- Finalizarea servlet-ului



Exemplu: `javaeetutorial5\examples\web\bookstore1\`

SERVLETS – CICLUL DE VIATA AL UNUI SERVLET

Când un request este mapat la un servlet, următorii pași:

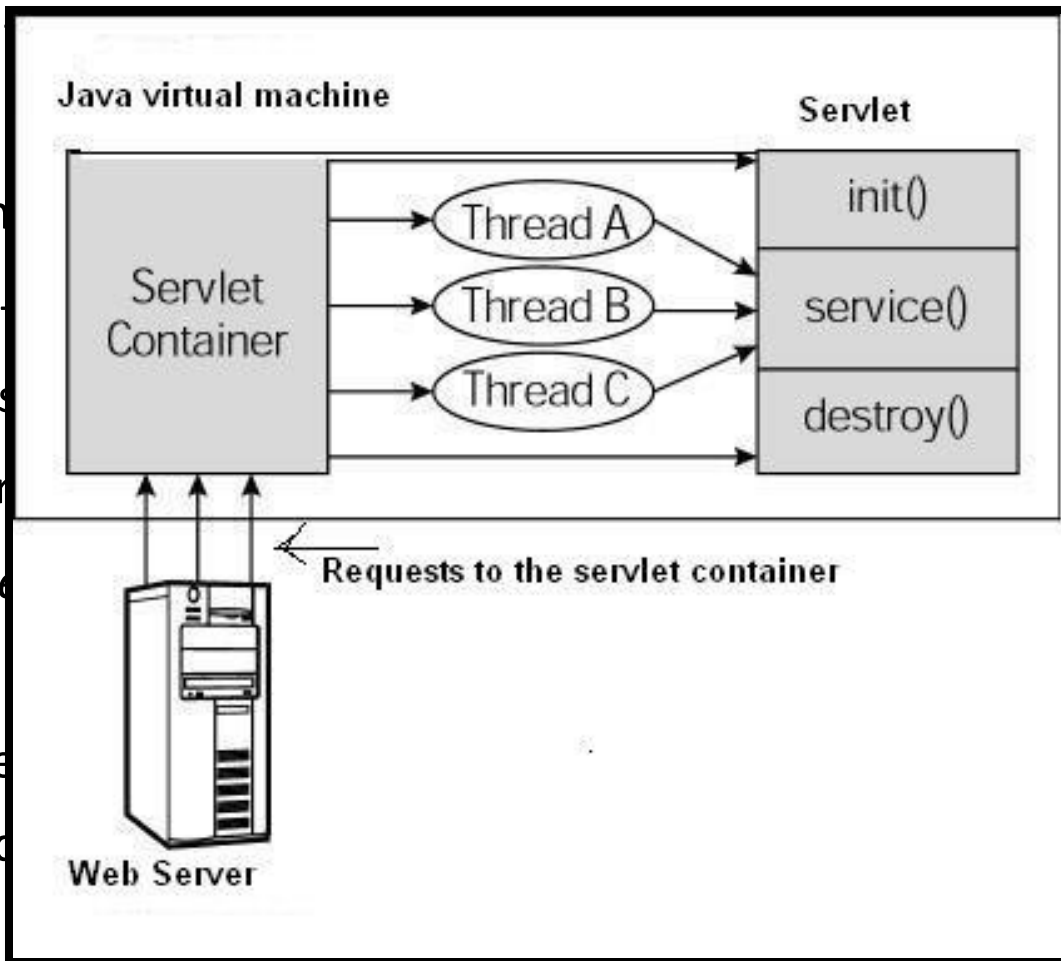
1. Dacă nu există încă o instanță a servletului:
 - a. încărcă clasa servlet
 - b. crea o instanță a clasei
 - c. inițializa instanța servletului
2. Invocă metoda de tip service() și returnează un response.

Atunci când container-ul trebuie să proceseze un request, se va face prin apelul metodei service().

Trateaza:

- Evenimente ale ciclului de viață a Servlet-ului
- Erori de servlet

See: bookstore1/src/java/com/sun/bookstore1/listeners/



SERVLETS – EVENIMENTE ALE CICLULUI DE VIATA

Object	Event	Listener Interface and Event Class (cu <i>javax.servlet.</i> ca prefix de pachet)
Web context	<ul style="list-style-type: none">• Inițializare și distrugere• Atribut adăugat, șters sau înlocuit	<ul style="list-style-type: none">• ServletContextListener and ServletContextEvent• ServletContextAttributeListener and ServletContextAttributeEvent
Session	<ul style="list-style-type: none">• invalidare, activare, inactivare (passivation) și expirare (timeout)• Atribut adăugat, șters sau înlocuit	<ul style="list-style-type: none">• http.HttpSessionListener, http.HttpSessionActivationListener, and http.HttpSessionEvent• http.HttpSessionAttributeListener and http.HttpSessionBindingEvent
Request	<ul style="list-style-type: none">• O cerere de servlet (ServletRequest) a început să fie procesată de către componentele web• Atribut adăugat, șters sau înlocuit	<ul style="list-style-type: none">• ServletRequestListener and ServletRequestEvent• ServletRequestAttributeListener and ServletRequestAttributeEvent

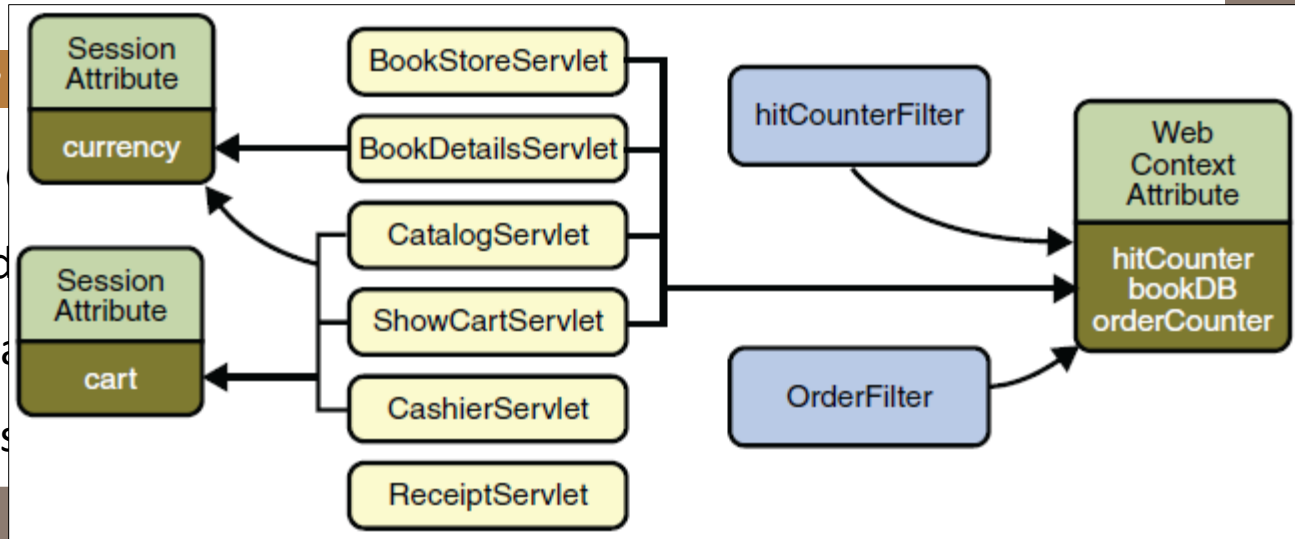
SERVLETS – PAR

Obiecte Scop (scope).

informațiilor sub formă de

scop predefinite. Acces

[get/set]Attribute ale clas



Scope Object	Class	
Web context	javax.servlet.ServletContext	Componente web într-un context web
Session	javax.servlet.http.HttpSession	Componente web ce trateaza o cerere ce aparține sesiunii
Request	Subtype of javax.servlet. ServletRequest	Componente web ce trateaza cererea.
Page	javax.servlet.jsp.JspContext	Pagina JSP care crează obiectul

Controlul accesului concurent la resurse partajate (fișiere, conexiuni la baze de date, conxiuni la rețea)

Accessarea bazelor de date (Java Persistence API)


```
public class CatalogServlet extends HttpServlet {
    private BookDBAO bookDB;
    public void init() throws ServletException {
        bookDB = (BookDBAO) getServletContext().
            getAttribute("bookDB");
        if (bookDB == null) {
            throw new UnavailableException(
                "Couldn't get database.");
        }
    }
}
```

Serviciul oferit de către un servlet este implementat într-o metodă de *service* a unui servlet generic (*GenericServlet*), într-o metodă *doMethod* (unde *Method* poate fi *Get*, *Delete*, *Options*, *Post*, *Put* sau *Trace*) a unui obiect *HttpServlet*, sau în orice altă metodă specifică protocolului definită într-o clasă ce implementează interfața *Servlet*.

- Preluarea informațiilor din cereri (Requests)
 - parameterii
 - obiecte – valori de atribute
 - informații despre request, client sau server specifice protocolului
- Construirea răspunsurilor (Responses)
 - output stream
 - content type
 - buffered output
 - i18n, character encoding

```
public class CatalogServlet extends HttpServlet {
    ...
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {
        // Set response content type
        response.setContentType("text/html");

        // Actual logic goes here.
        PrintWriter out = response.getWriter();
        out.println("<h1>" + message + "</h1>");
    }
    ...
}
```

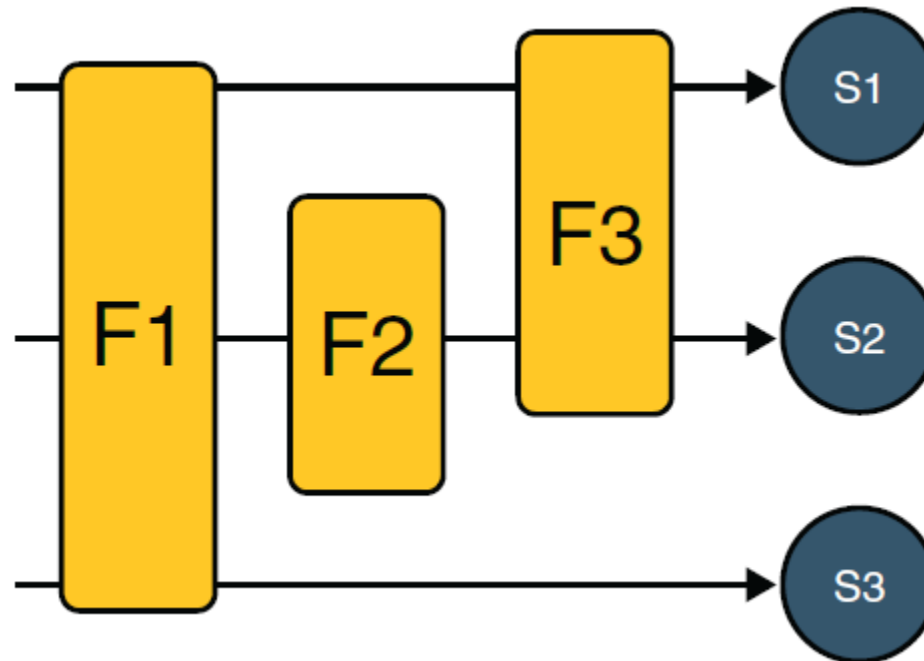
Un **filtru** este un obiect care poate transforma antetul (header) și/sau conținutul (content) unui *request* sau al unui *response*. De obicei, filtrele nu generează răspuns, ci mai degrabă se “atașează” unei resurse web de orice tip.

Principalele sarcini pe care le poate îndeplini un filtru sunt:

- Interogarea request-ului și acționarea în consecință.
- Blocarea perechii request-response pentru a nu trece mai departe.
- Modificarea datelor (header/content) din request. (Customizing the request)
- Modificarea datelor (header/content) din response. (Customizing the response)
- Interacțiunea cu resurse externe.

Aplicabilitatea filtrelor: autentificare, log-are, conversie în imagini, compresia datelor, encriptare, token-izarea stream-urilor, transformări XML etc.

- interfețe: Filter, FilterChain și FilterConfig
- metode: *doFilter*, *init* and *destroy* methods
- Filter mappings: de ex. */private/**



Vezi: `com.sun.bookstore1.filters.HitCounterFilter`

- Includerea altor resurse în *response*

```
RequestDispatcher dispatcher = getServletContext().  
    getRequestDispatcher("/banner");  
if (dispatcher != null) {  
    dispatcher.include(request, response);  
}
```

- Transferul controlului către altă componentă web

```
RequestDispatcher dispatcher = request.getRequestDispatcher(  
    "/template.jsp");  
if (dispatcher != null) {  
    dispatcher.forward(request, response);  
}
```

- **Accesarea contextului web**

```
ServletContext context = filterConfig.getServletContext();
```

- **Mentținerea stării clientului**

```
HttpSession session = request.getSession();  
ShoppingCart cart = (ShoppingCart) session.getAttribute("cart");
```

- **Finalizarea servlet-ului**

```
public void destroy() {  
    bookDB = null;  
}
```

- September 2010 (original publicată în 2007)
- Eric Jendrock, Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, Kim Haase
- ghid pentru dezvoltarea aplicațiilor “enterprise” folosind Java EE



CUVINTE CHEIE

- Servlet
- session
- request
- response
- filtre

- <http://docs.oracle.com/javaee/5/tutorial/doc/p1.html>
- <http://docs.oracle.com/javase/tutorial/>
- <http://www.oracle.com/technetwork/java/index-142838.html>