

Eintopf

zvaný

Java Enterprise Edition

Okem pozorovatele

Přemek Brada

Katedra informatiky a výpočetní techniky

Západočeská univerzita v Plzni



Obsah

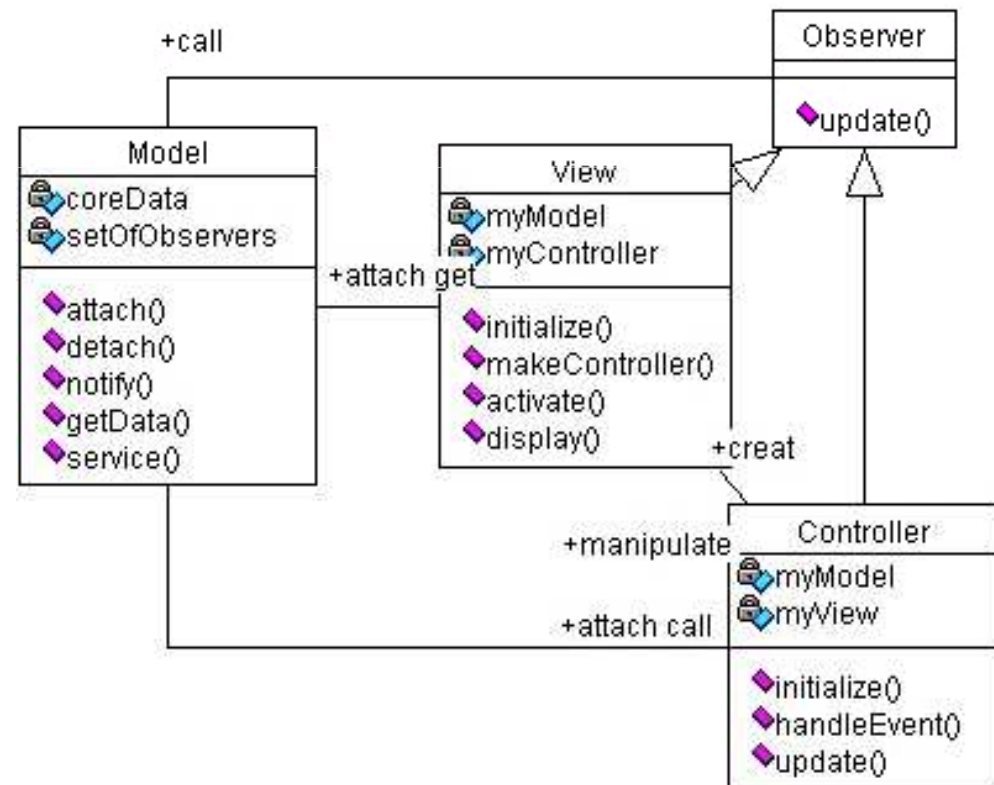
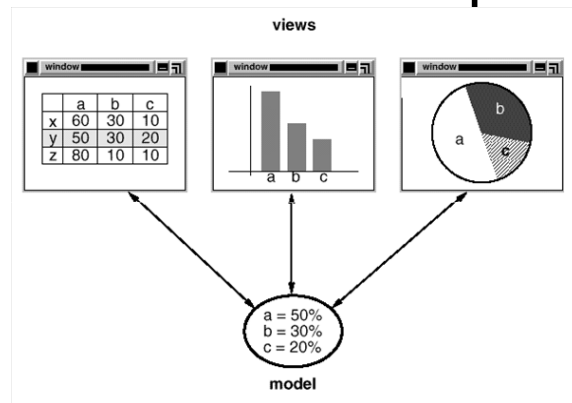
- Co to je J2EE a co vše tento eintopf obsahuje
- Základní principy (MVC)
- Java platformy
- Struktura a technologie pro JEE aplikace
- Další frameworky
- Vývoj platformy Java Enterprise Edition
- Shrnutí a poučení

ingredience

přílohy

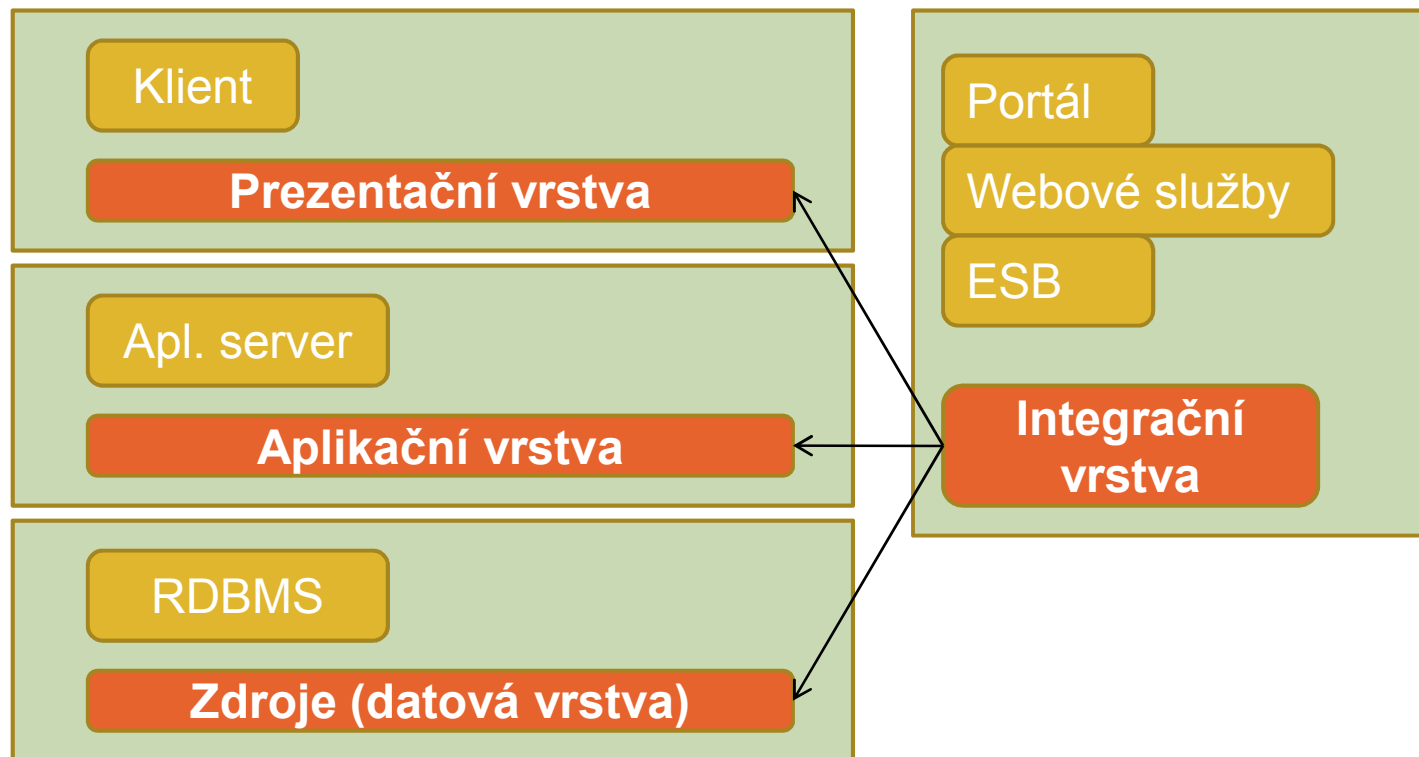
Model-View-Controller

- Jeden ze základních návrhových vzorů
 - více vizuálních podob stejného obsahu

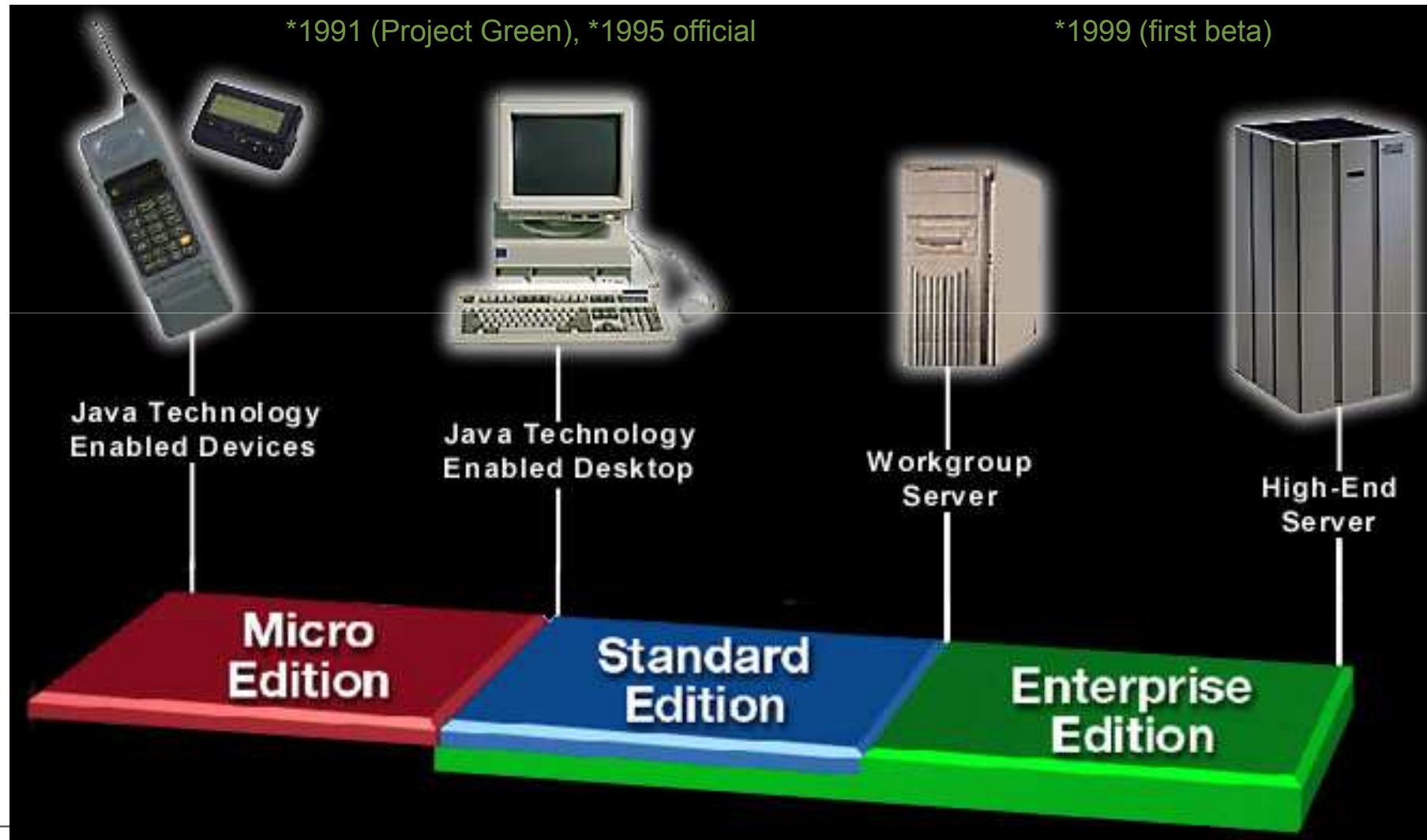


Vícevrstvé aplikace

"MVC ve velkém" – lokalizace změn v aplikaci



Platforma Java



Platforma Java EE

- Cíl = server-side, velké aplikace

- Infrastruktura

- klient (tenký, tlustý)
- aplikační server – kontejner
- aplikace, komponenty
- databázový server
- konfigurace, deskriptory



web (HTML et al), desktop (Swing)

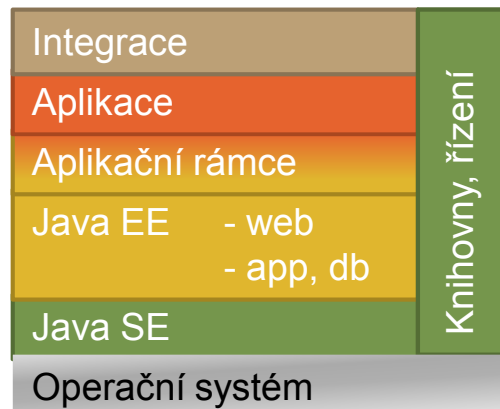


životní cyklus
správa zdrojů
distribuovanost
clustering
komunikace

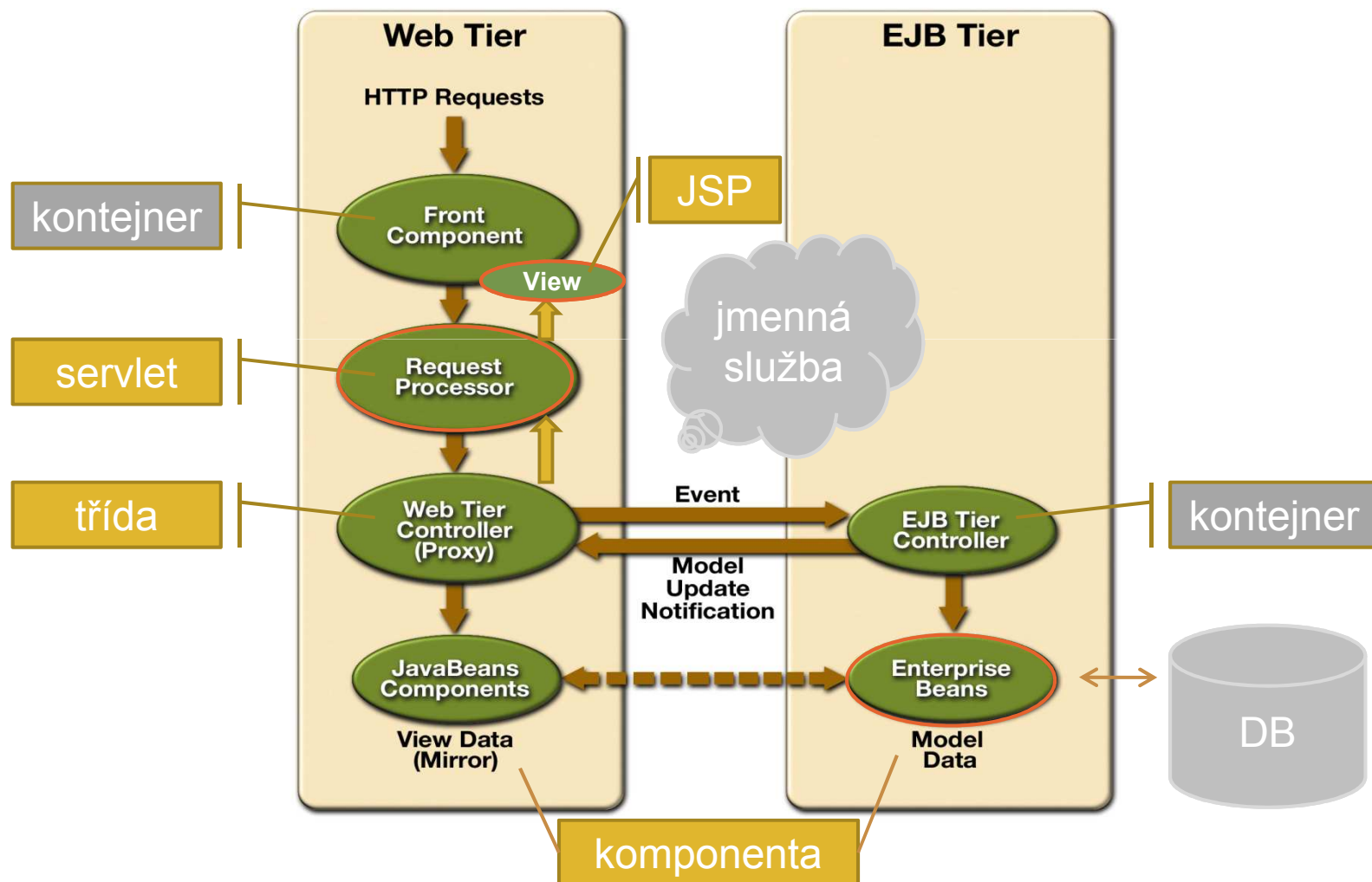


jednoduchost
výkon
škálovatelnost
bezpečnost

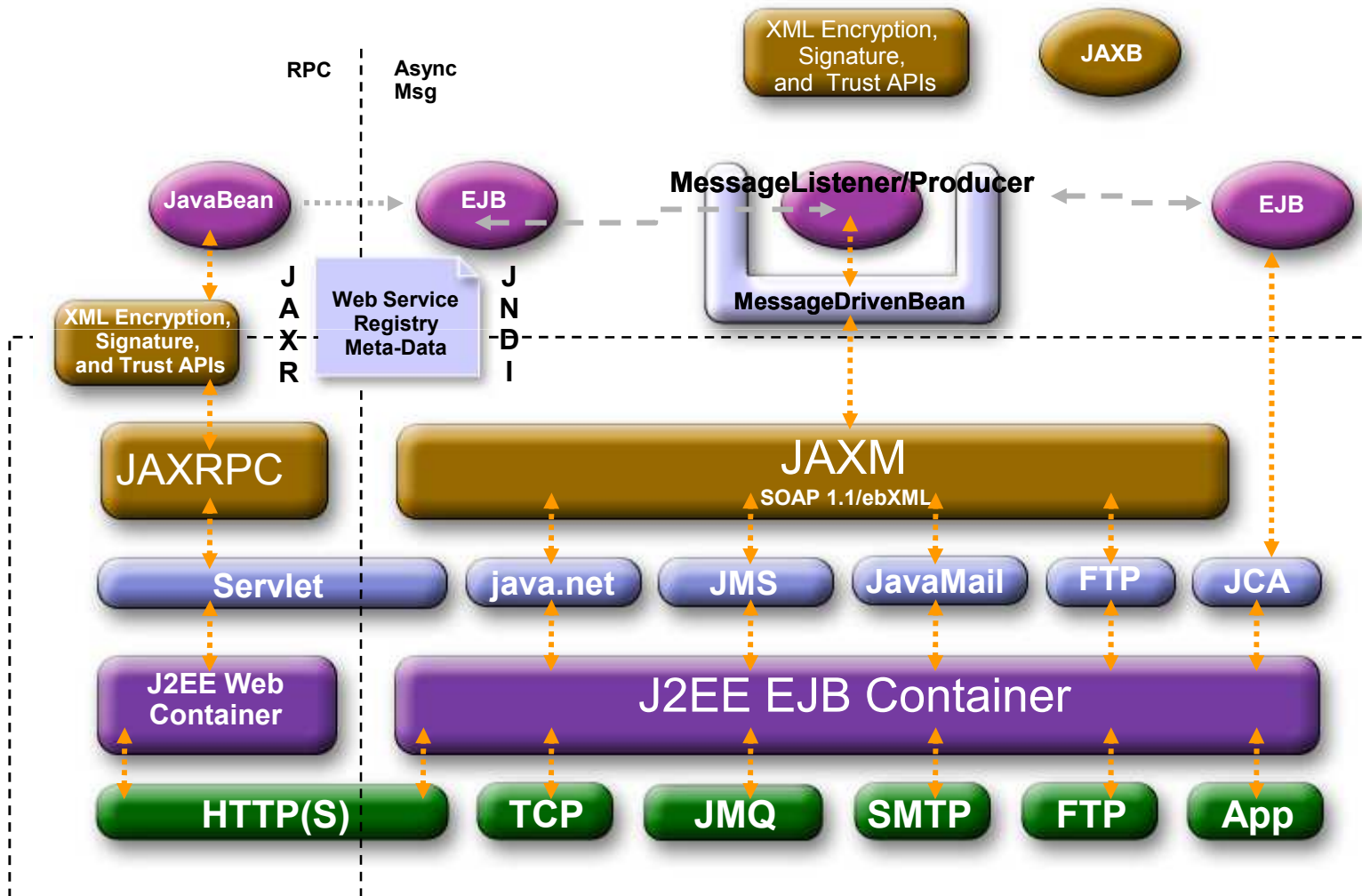
Geronimo, Orion
JBoss, Tomcat,
Jetty, ...
BEA WebLogic
IBM WebSphere
SunOne, ...



JxEE aplikace: základní kostra



JxEE aplikace: technologie



Standardní přílohy

- JCR
 - Content Repository API (JSR 170)
 - hierarchický obsah → CMS, DMS
 - JMX
 - Management Extensions
 - Portlety
 - JSR 168, (286)
 - integrace na prezentační vrstvě
-

Ingredience a přílohy cizokrajné

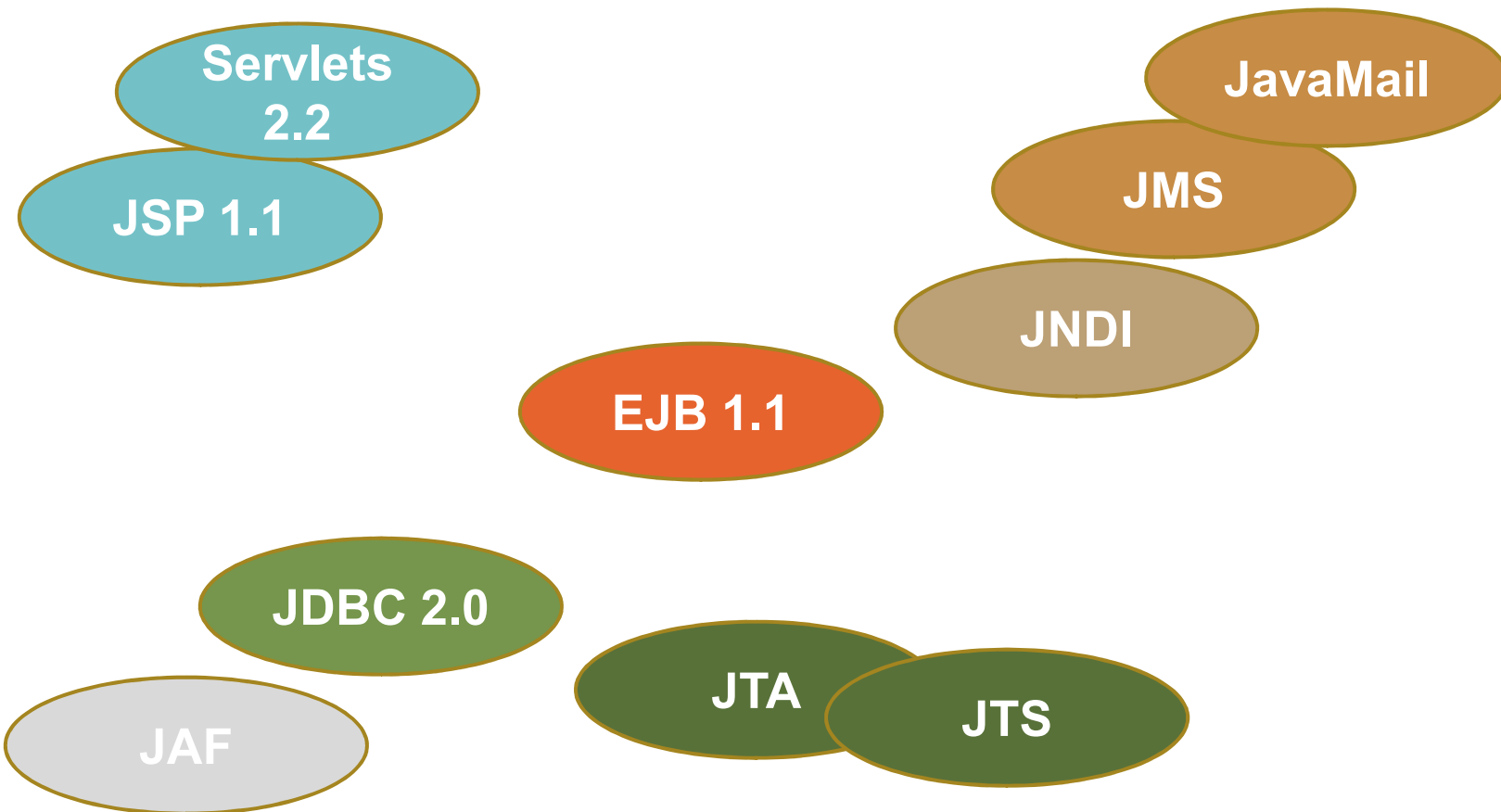
Framework is the word

- Prezentační
 - Struts, Tapestry, Wicket, ...
 - Aplikační
 - Spring, Picocontainer
 - Datová
 - Hibernate
 - Spring
 - Seam
-

Vývoj platformy

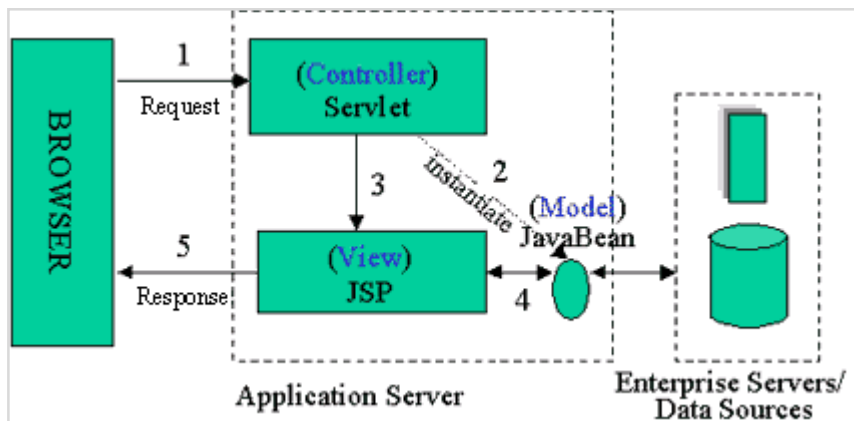
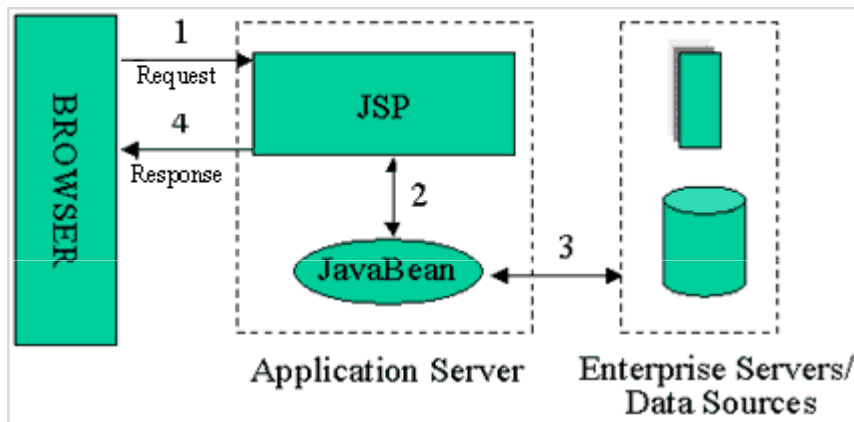


Verze 1.2 (1999)



Modely aplikace, JSP

- **Model 1, Model 2**



- **Kód stránky**

```
<%
MyBeanClass myBean = (MyBeanClass)
session.getAttribute("myBean") ;
String type = myBean.getType() ;
String imageUrl = myBean.getImageUrl() ;
%>
<B>The value of the type is <%= type %>.
</B>
<P><IMG SRC="<%= imageUrl %>">
```

```
<%@ taglib uri="/WEB-INF/c.tld" prefix="c" %>
<c:if test="${sessionScope.myBean.multiple}">
<c:forEach var="nb"
items="${sessionScope.myBean.collection}">
<B><c:out value="${nb.name}" /></B>
</c:forEach>
</c:if>
```

Servlet

```
public class DbExampleDisplay
extends HttpServlet
{
    private DbExampleUtil db = null;

    public void init()
    {
        this.db = new DbExampleUtil(this.getServletContext());
    }

    public void doGet(HttpServletRequest q, HttpServletResponse r) throws
    {
        boolean res = true;
        PrintWriter out = r.getWriter();
        ServletContext cx = this.getServletContext();

        String action = q.getParameter("action");
        if (action == null)
            action = "list";

        if (action.compareTo("add") == 0) {
            String nazev;
            float cena;
            nazev = q.getParameter("nazev");
            cena = Float.parseFloat(q.getParameter("cena"));
            cx.log("bad params for 'add': ",e);
            res = false;
        } else if (action.compareTo("del") == 0) {
            int id = Integer.parseInt(q.getParameter("id"));
            db.del(id);
        }
    }
}
```

EJB

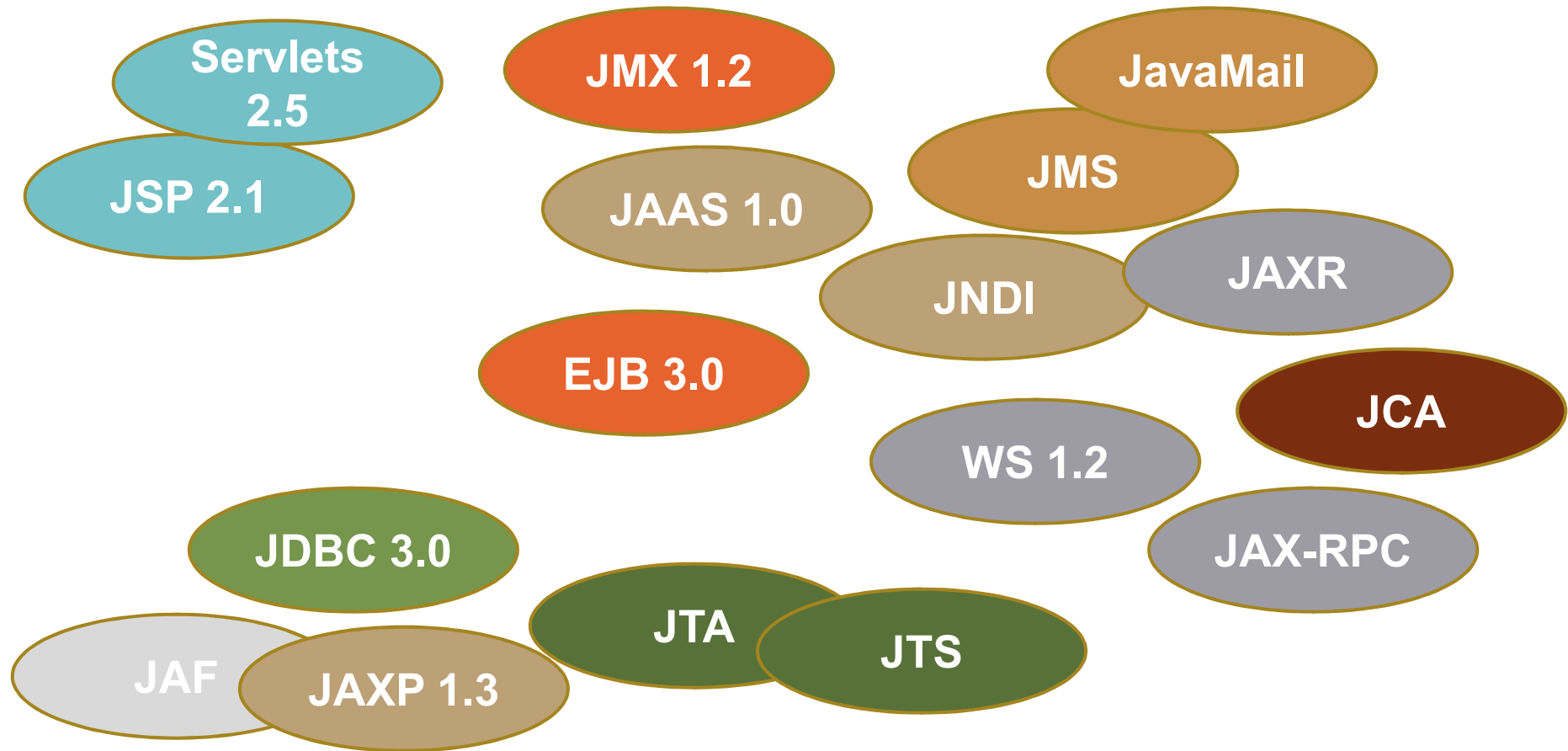
```
39 public class AccountEJB implements EntityBean {
40
41     private AccountModel accountDetails;
42     private EntityContext context;
43     private transient AccountDAO accountDao;
44
45     public AccountEJB() {}
46
47     public String ejbCreate (String userId, String password,
84     public void ejbRemove() throws RemoveException {}
95     public void setEntityContext(EntityContext ec) {}
98     public void ejbLoad() {
99         AccountDAO dao;
100         try{
101             dao = getDAO();
102             this.accountDetails = dao.load((String)context.g
103         } catch (AccountDAOFinderException se) {
104             throw new EJBException (se.getMessage());
105         } catch (AccountDAOSysException acs) {
106             throw new EJBException (acs.getMessage());
107         }
108     }
109
110     public void ejbStore() {}
120     public String ejbFindByPrimaryKey (String key) throws Fi
132     public void unsetEntityContext() {}
133     public void ejbActivate() {}
134     public void ejbPassivate() {}
137     public void ejbPostCreate(String userId, String password
141
142     // business methods
143     public void changeContactInformation(ContactInformation
144         String uid = accountDetails.getUserId();
145         String stat = accountDetails.getStatus();
146         this.accountDetails = new AccountModel(uid, stat, in
147     }
148
149     public AccountModel getDetails() {
150         return(this.accountDetails);
```

Entity bean

Část deskriptoru

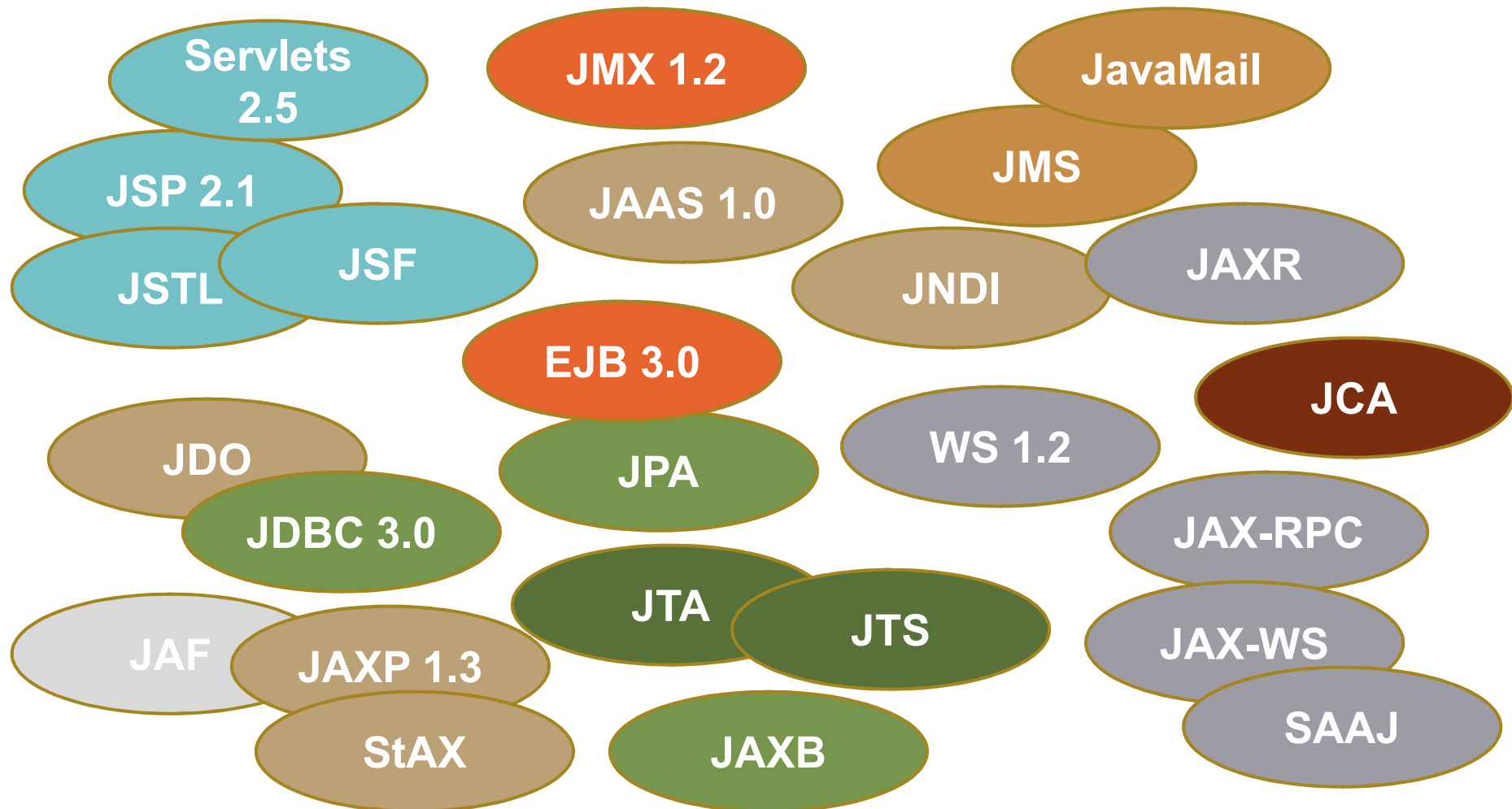
```
67
68 <entity>
69     <description>This is the Account EJB which represents
70         the information which is kept for each Customer</description>
71     <display-name>The Account</display-name>
72     <ejb-name>TheAccount</ejb-name>
73     <home>com.sun.j2ee.blueprints.customer.account.ejb.AccountHome</home>
74     <remote>com.sun.j2ee.blueprints.customer.account.ejb.Account</remote>
75     <ejb-class>com.sun.j2ee.blueprints.customer.account.ejb.AccountEJB</
76     <persistence-type>Bean</persistence-type>
77     <prim-key-class>java.lang.String</prim-key-class>
78     <reentrant>False</reentrant>
79     <env-entry>
80         <env-entry-name>ejb/account/AccountDAOClass</env-entry-name>
81         <env-entry-type>java.lang.String</env-entry-type>
82         <env-entry-value>com.sun.j2ee.blueprints.customer.account.dao.Acc
83     </env-entry>
84     <resource-ref>
85         <res-ref-name>jdbc/EstoreDataSource</res-ref-name>
86         <res-type>javax.sql.DataSource</res-type>
87         <res-auth>Container</res-auth>
88     </resource-ref>
89 </entity>
90 </enterprise-beans>
91
92 <assembly-descriptor>
93
94     <container-transaction>
95         <method>
96             <ejb-name>TheAccount</ejb-name>
97             <method-intf>Home</method-intf>
98             <method-name>remove</method-name>
99             <method-params>
100                 <method-param>java.lang.Object</method-param>
101             </method-params>
102         </method>
103         <trans-attribute>Required</trans-attribute>
```

Verze 1.4 (2003)



Verze 5 (dnes)

$\Sigma=29$



JSF

```
<h:panelGrid>
  <h:dataTable value="#{inventoryManagerBean.inventoryForCategory}" var="items"
    border="1" frame="none" style="text-align:center;">
    <h:column>
      <h:inputText value="#{items.item.name}"/>
      <f:facet name="header">
        <h:outputText value="Name"/>
      </f:facet>
    </h:column>
    ...
    <h:column>
      <f:facet name="header">
        <h:outputText value="Select"/>
      </f:facet>
      <h:commandButton value="orders"
        type="submit"
        actionListener="#{inventoryManagerBean.itemSelected}"
        action="orderItem">
        <f:facet name="extraParameter">
          <f:param name="commandButtonParam"
            value="#{items.item.SKU}"/>
        </f:facet>
      </h:commandButton>
    </h:column>
  </h:dataTable>
</h:panelGrid>
```

EJB

```
10 @Entity
11 public class Item {
12
13     protected long SKU;
14     protected String description;
15     protected String name;
16
17     @OneToOne
18     @JoinColumn(name="INVENTORY_SKU")
19     protected Inventory inventory;
20
21     @Id
22     @GeneratedValue
23     public long getSKU() {
24         return SKU;
25     }
26     public void setSKU(long sku) {
27         this.SKU = sku;
28     }
29
30     public String getDescription() {
31         return description;
32     }
33     public void setDescription(String description) {
34         this.description = description;
35     }
36
37     public String getName() {
38         return name;
39     }
40 }
```

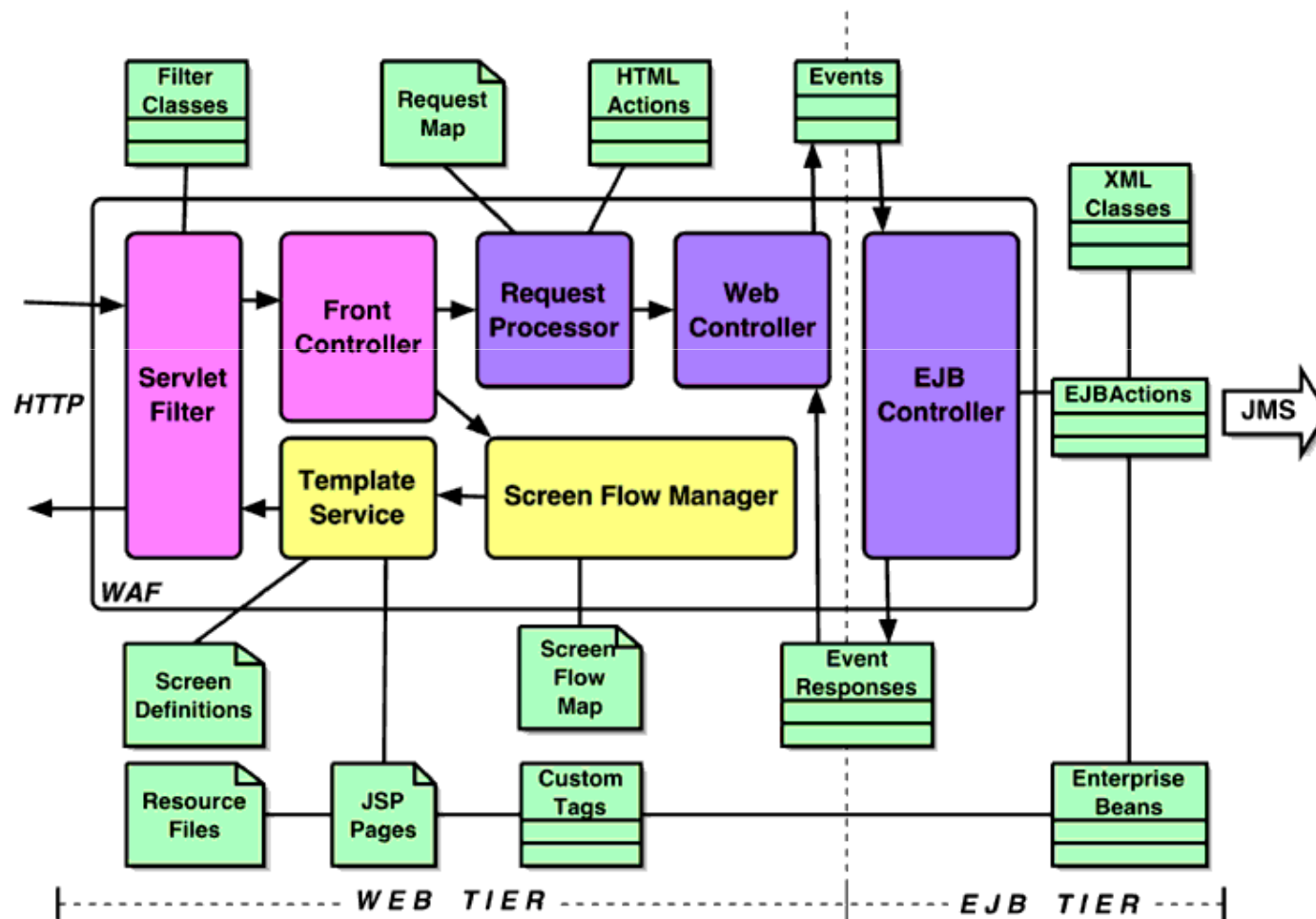
```
import declarations
└─ javax.persistence.Entity
└─ javax.persistence.GeneratedValue
└─ javax.persistence.Id
└─ javax.persistence.JoinColumn
└─ javax.persistence.OneToOne
└─ javax.persistence.Version

Item
├─ SKU : long
├─ description : String
├─ name : String
├─ inventory : Inventory
├─ getSKU()
├─ setSKU(long)
├─ getDescription()
├─ setDescription(String)
├─ getName()
├─ setName(String)
├─ getCategory()
├─ setCategory(String)
├─ setVersion(int)
├─ getVersion()
```

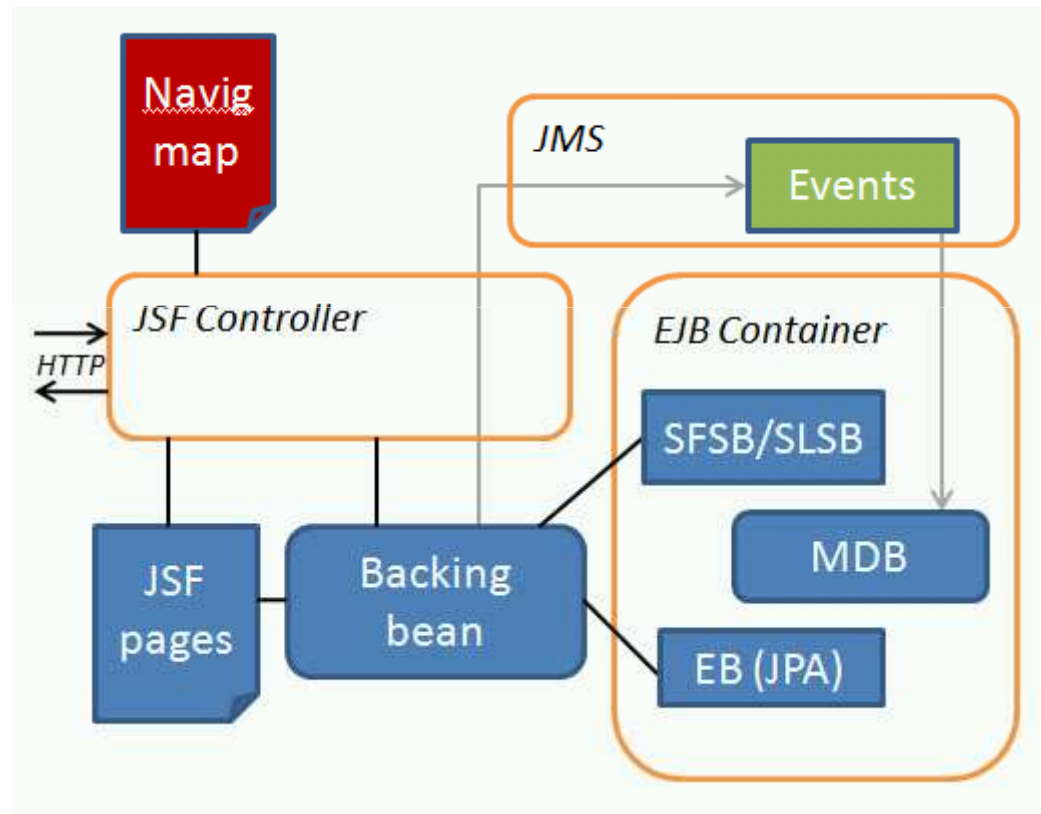
Shrnutí



Typická J2EE 1.x webová aplikace

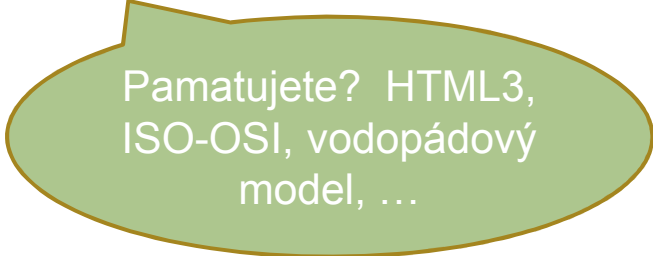


Typická JEE 5 webová aplikace



Poučení z vývoje

- Java Enterprise Edition: Eintopf (vše v jednom kotli)
 - web, aplikační logika, persistence, messaging, web services, jmenné služby, bezpečnost, vzdálená správa, ...
- Černí koně
 - OSGi = JSR 291
 - Seam, Terracota
- Historie o tom, jak si život razí cestu
 - heavy architecture, XML, tuny kódu
 - POJO, IoC, anotace
- JSR 316 alias Java EE verze 6



Pamatujete? HTML3,
ISO-OSI, vodopádový
model, ...

Děkuji za pozornost

