

12.–15.09.2010  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## Sternenkollision

JSF im Portal

Werner Eberling / Sascha Groß

MATHEMA Software GmbH

# Agenda

---

- Was ist ein Portal?
- Warum Portletbridges?
- Portlet Events & Resource Requests
- Parallelbetrieb

# Die Sache mit den Webanwendungen

The image displays three overlapping screenshots of popular German web applications:

- Amazon.de:** The top-left screenshot shows the Amazon Germany homepage. It features a navigation menu on the left with categories like 'Bücher', 'Musik, DVD & Games', and 'Computer & Büro'. The main banner advertises 'Der neue Kindle' (The new Kindle) with a price of \$139 Wi-Fi or \$189 Free 3G-Wi-Fi. There are also sections for 'Amazon Prime' and 'Weitere Artikel für Sie' (Further articles for you).
- Google.de:** The bottom-center screenshot shows the Google Germany homepage. It features the classic Google logo, a search bar, and navigation links for 'Web', 'Bilder', 'Videos', 'Maps', 'News', 'Shopping', 'E-Mail', and 'Mehr'. The search bar includes a 'Google-Suche' button and a 'Auf gut Glück!' (Just in case!) button.
- wetter.de:** The top-right screenshot shows the weather website 'wetter.de'. It features a search bar for location, a navigation menu, and a main section for 'Wettervorhersage: Wie ist das Wetter in' (Weather forecast: How is the weather in). It includes a map of Germany with weather icons for various cities and a temperature scale on the right.

# Die Sache mit den Webanwendungen

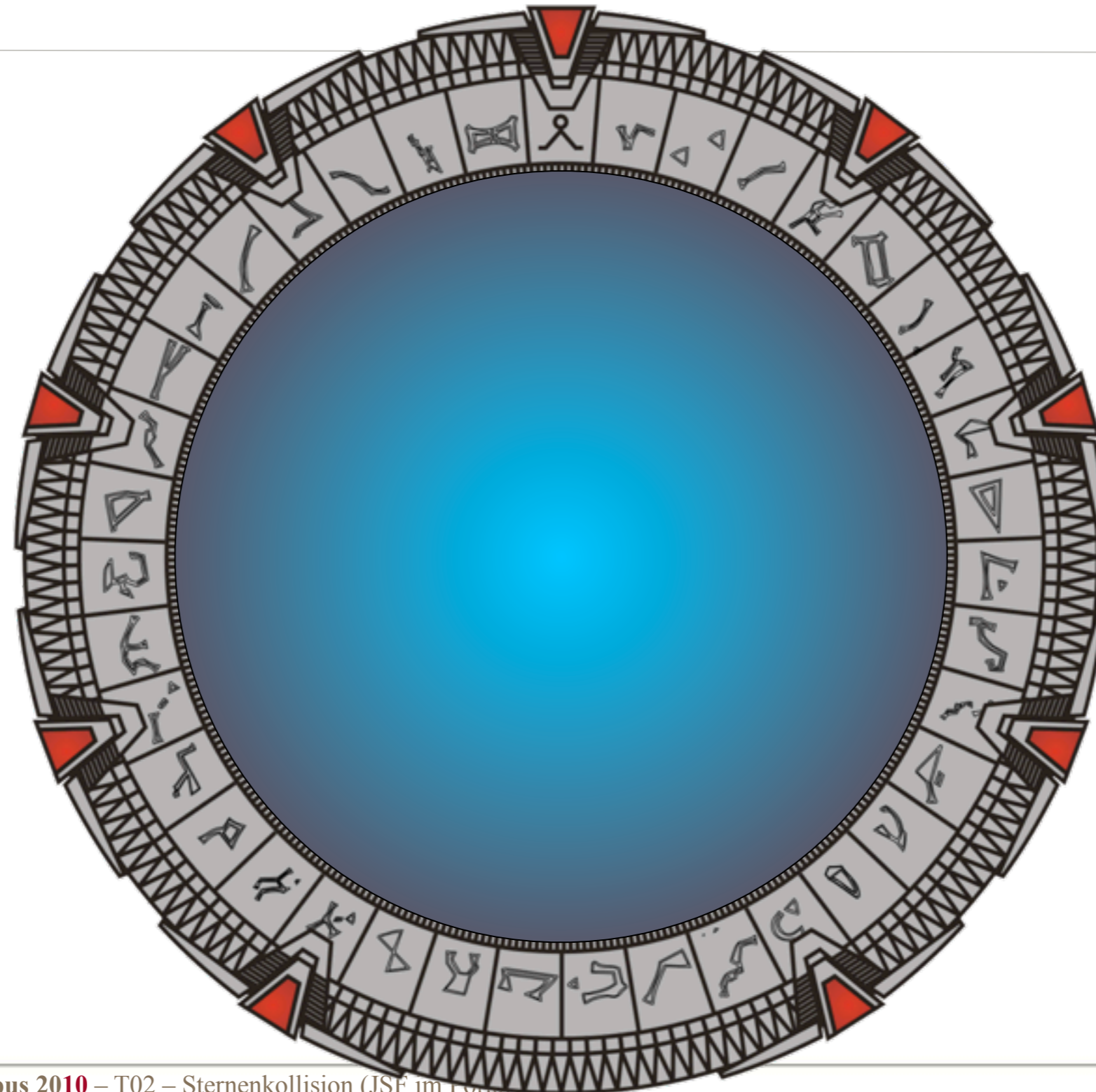
---

- Sind state-of-the-art
- Kommen selten alleine
- Sollen heutzutage möglichst personalisiert sein

Wohin bringt uns das ?



# Wohin bringt uns das ?



# iGoogle™

[Erweiterte Suche](#)  
[Sprachoptionen](#)

Google-Suche

Auf gut Glück!

Diese Seite immer als meine Google-Startseite anzeigen. [Google als Startseite festlegen](#)

[Design Klassisch ändern](#) | [Gadgets hinzufügen](#) »

## Startseite

- YouTube
- Google News
- Wetter
- Datum & Uhrzeit
- SPIEGEL ONLINE - ...
- Wikipedia, die freie ...
- ToDo
- BILD.de alle Artikel
- Amazon für iGoogle

Aktualisierungen  
Freunde

## Chat

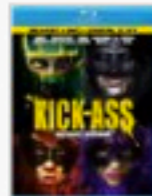

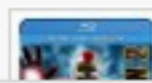
[Sign-in](#) to chat  
on iGoogle

### Amazon für iGoogle

« [Bestseller](#) [Suche](#) [Preisvergleich](#) [Golg](#) »

DVD  amazon.com.

Action & Adventure

-  **Kick-Ass**  
~ Nicolas Cage, ~ Mark Strong  
\$24.99 [Preisvergleich](#)  
[Größer](#)
-  **Lost: The Complete Sixth and Final Season**  
~ Matthew Fox, ~ Evangeline Lilly  
\$37.99 [Preisvergleich](#)  
[Größer](#)
-  **Iron Man 2**  
~ Robert Downey Jr., ~

### Datum & Uhrzeit



Di  
**17**  
AUG

### Wetter

[Nürnberg, Bayern](#)



**15°C** Aktuell: Leichter Regen  
Wind: W mit 21 km/h  
Feuchtigkeit: 85 %

Di.	Mi.	Do.	Fr.
			
11°   17°	13°   19°	12°   19°	14°   21°

### YouTube

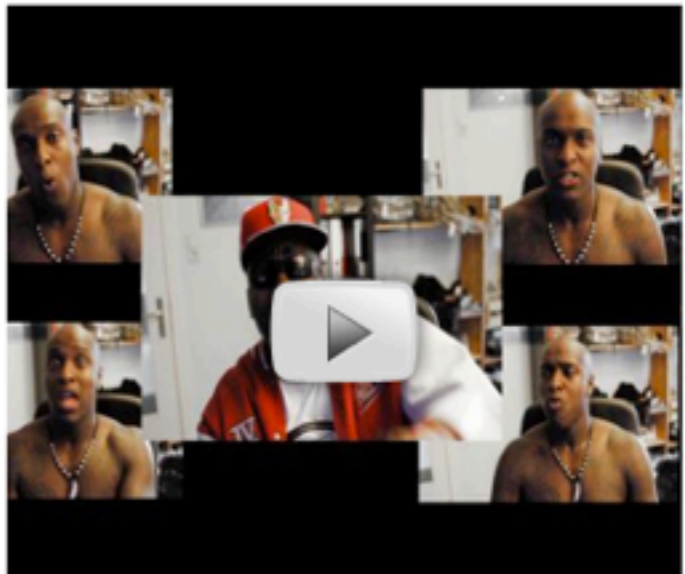
Für dieses Gadget Folgendes zulassen:

- Mich und meine [Freunde](#) kennenlernen
- Meine Aktivitäten in Aktualisierungen posten

[Speichern](#) [Weitere Informationen](#)

[Suchen](#)

Top-Favoriten von heute



# Was ist ein Portal?

---

- Webanwendung bestehend aus vielen kleinen Webanwendungen
- Ermöglicht Personalisierung
- Bietet Mehrwertdienste
  - SingleSignOn
  - Content-Management
  - ...



## Was ist ein Portlet?

---

- Eine Webanwendung die als Teil eines Portals läuft
- Erzeugt HTML
- Interagiert mit dem Benutzer
- Vergleichbar mit einem JavaServlet
- Benötigt einen Portlet-Container bzw. Portal-Server

# Eine Auswahl von Portal-Servern

---

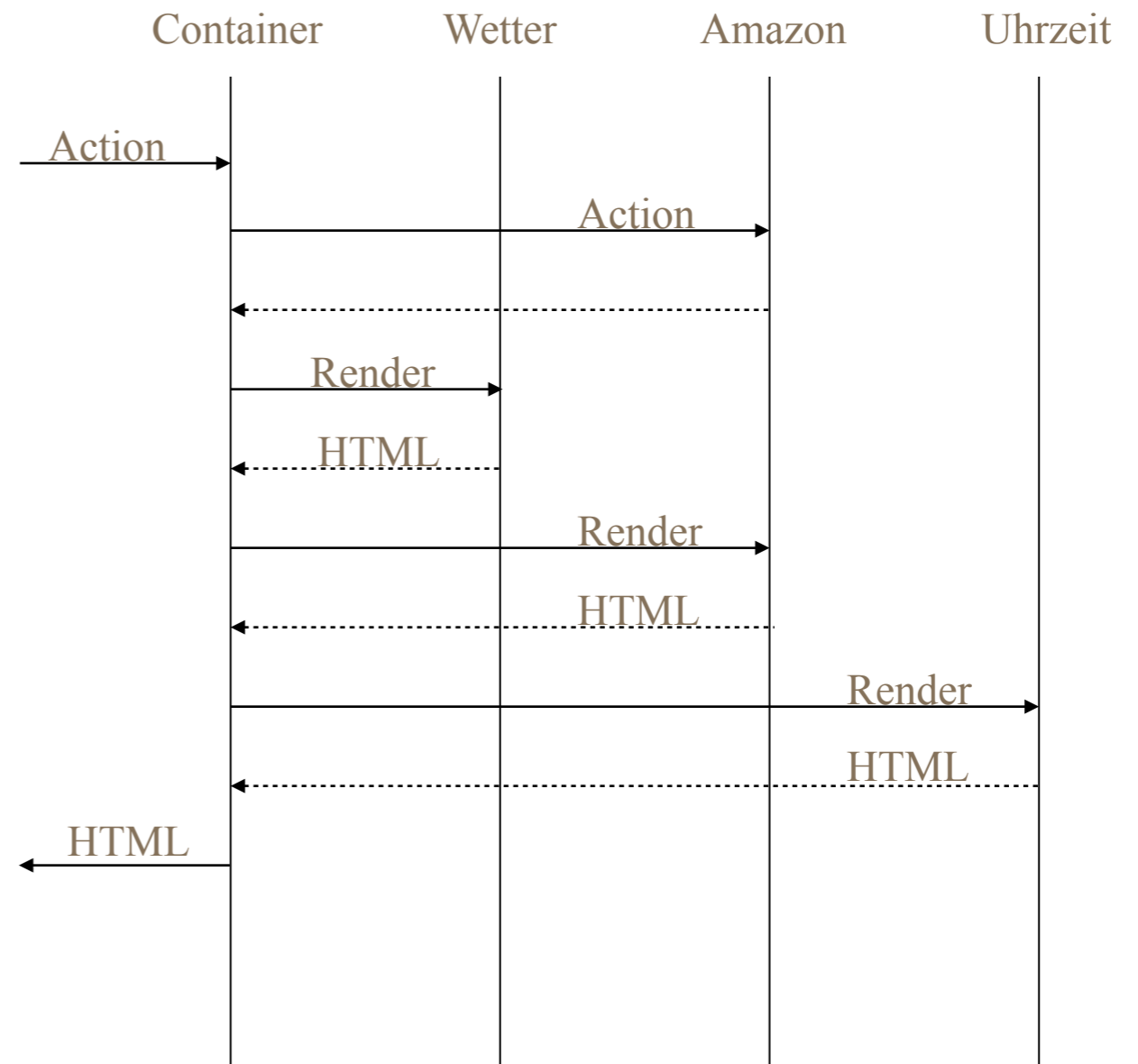


# Portal-Server und Portlet

---

- Portal-Server
  - Eigenständige Webanwendung
  - Definiert u.a. Layout und Abfolge der einzelnen Webseiten
  - Realisiert Sicherheitskonzepte
  - Ermöglicht Content-Verwaltung
- Portlet
  - Spezifische Anwendung innerhalb des Portals
  - Realisiert ein klar definierte Fachlichkeit
  - Teil einer (oder mehrerer) Webseite(n) eines Portal-Servers

# Aktion oder Darstellung ?

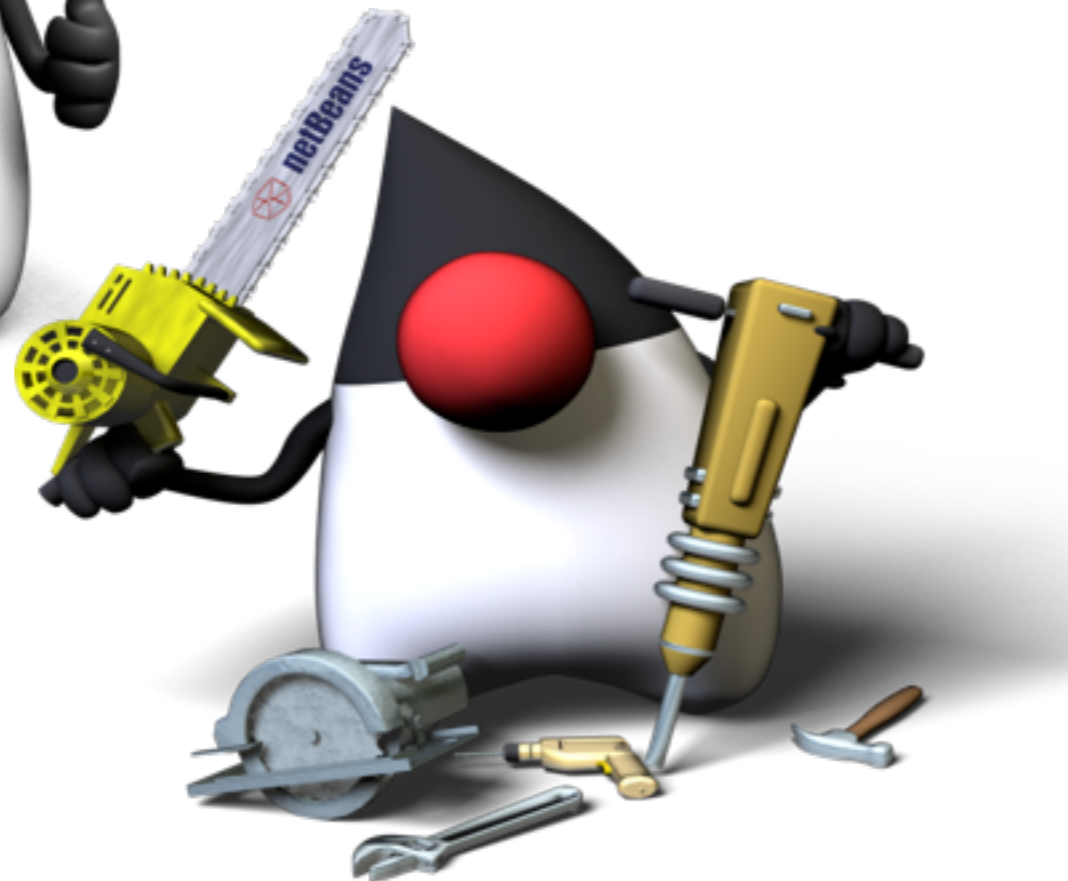


# Ansichtssache - Portletmodes

VIEW



EDIT



HELP



<https://duke.dev.java.net>

# Wie sieht's technisch aus (1/3)?

- Implementierung eines HelloWorld-Portlets

```
import javax.portlet.*;
import java.io.IOException;

public class GreetingPortlet extends GenericPortlet {

    @Override
    protected void doView(RenderRequest request, RenderResponse response)
        throws PortletException, IOException {
        request.setAttribute("name", request.getPortletSession().getAttribute("name"));
        getPortletContext().getRequestDispatcher("/hello.jsp").forward(request, response);
    }

    @Override
    public void processAction(ActionRequest request, ActionResponse response)
        throws PortletException, IOException {
        request.getPortletSession().setAttribute("name", request.getParameter("name"));
    }
}
```



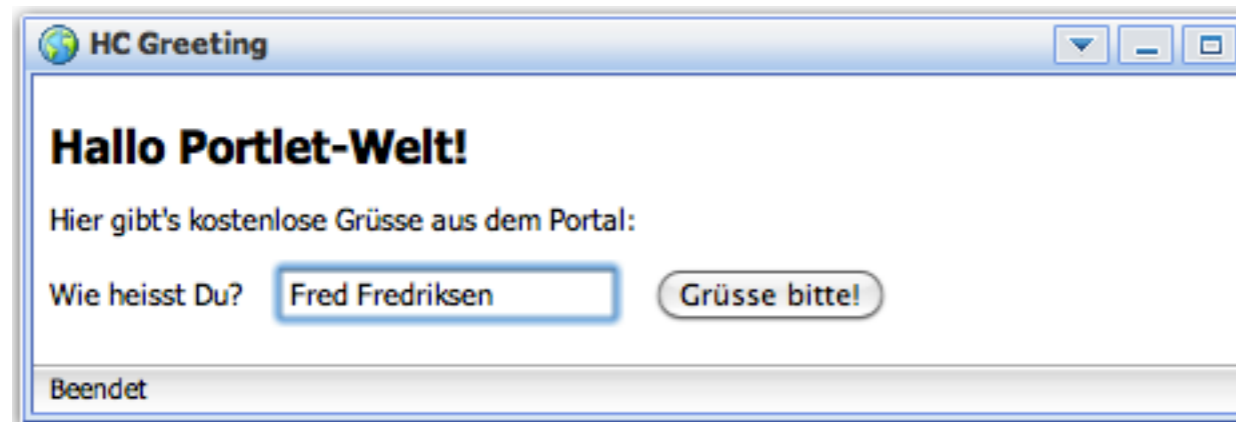
# Wie sieht's technisch aus (3/3)?

- Der obligatorische Deskriptor (portlet.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd/etc/opt/
  SUNWportal/dtd/portlet.xsd" version="2.0">
  <portlet>
    <portlet-name>jsf.portlet-greetingPortlet</portlet-name>
    <portlet-class>de.mathema.jsf.portlet.example.portlet.GreetingPortlet
    </portlet-class>
    <supports>
      <mime-type>text/html</mime-type>
      <portlet-mode>VIEW</portlet-mode>
    </supports>
    <portlet-info>
      <title>herbstcampus.greeting.portlet</title>
    </portlet-info>
  </portlet>
</portlet-app>
```



# Und so sieht's im Portal aus



## So einfach geht's

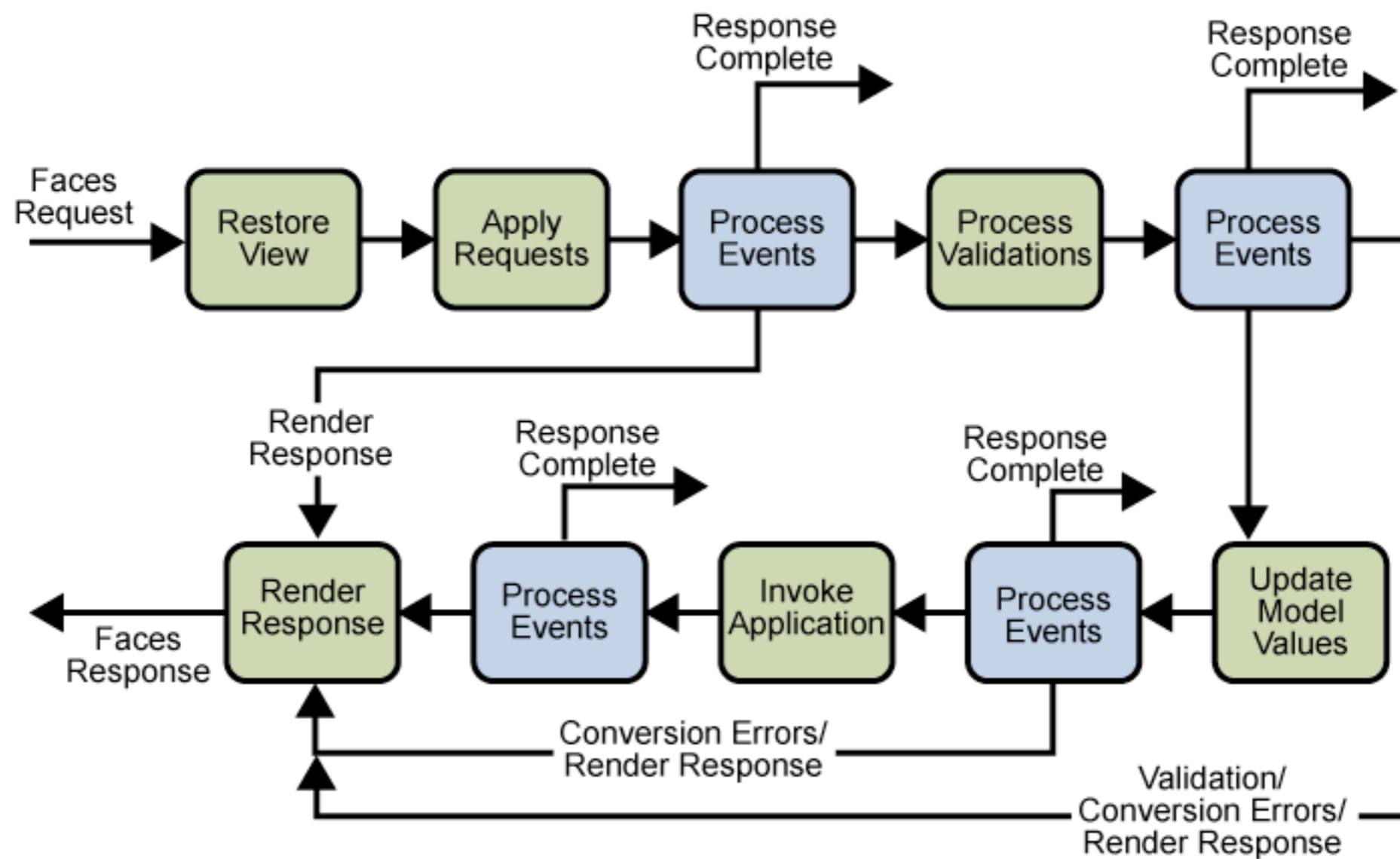
---

- Schneller Einstieg
- MVC-Pattern, wie schon bei JavaServlets möglich
- ABER: Ein komfortableres Framework wäre wünschenswert

# Warum Portletbridges?

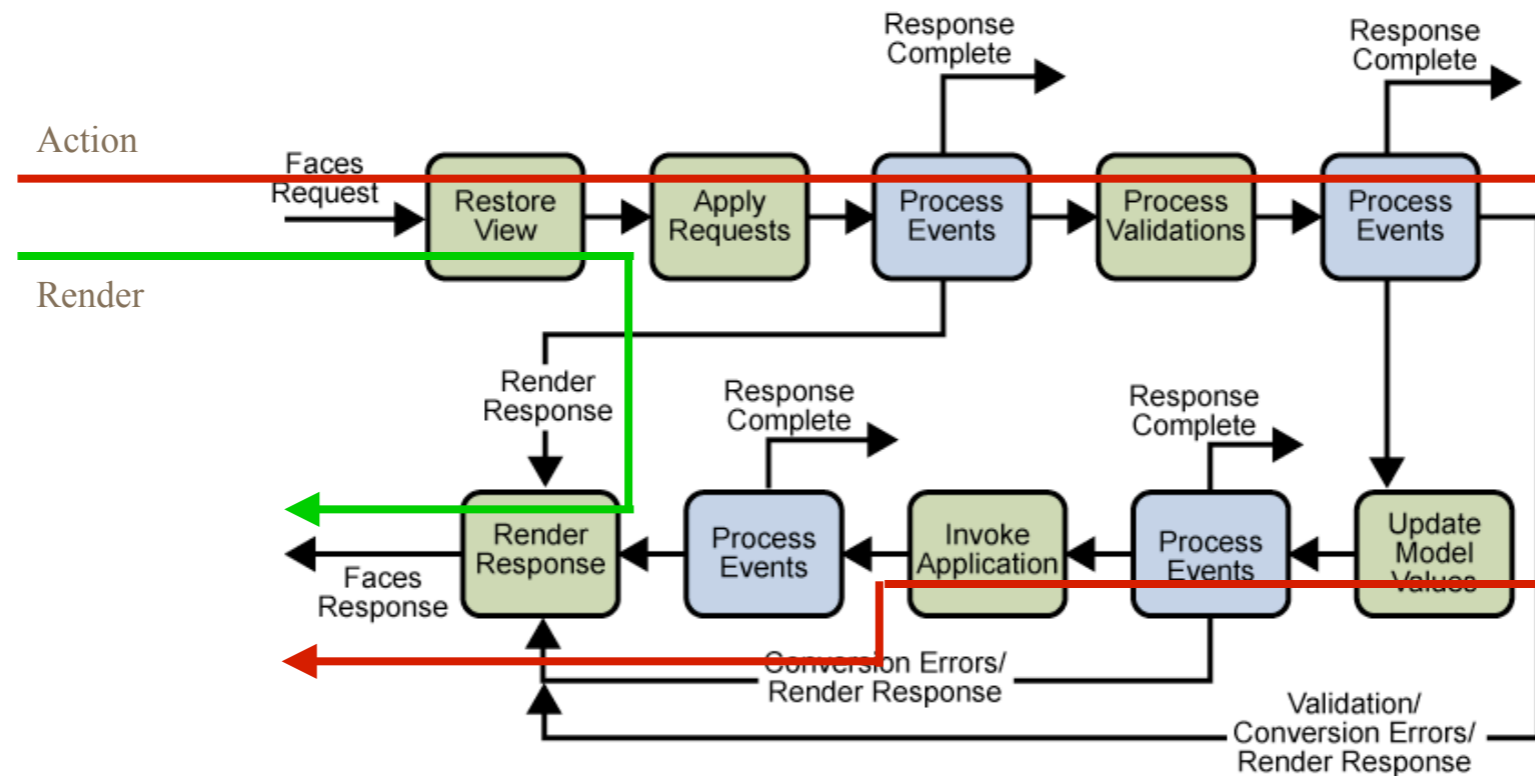


# Der JSF Lifecycle



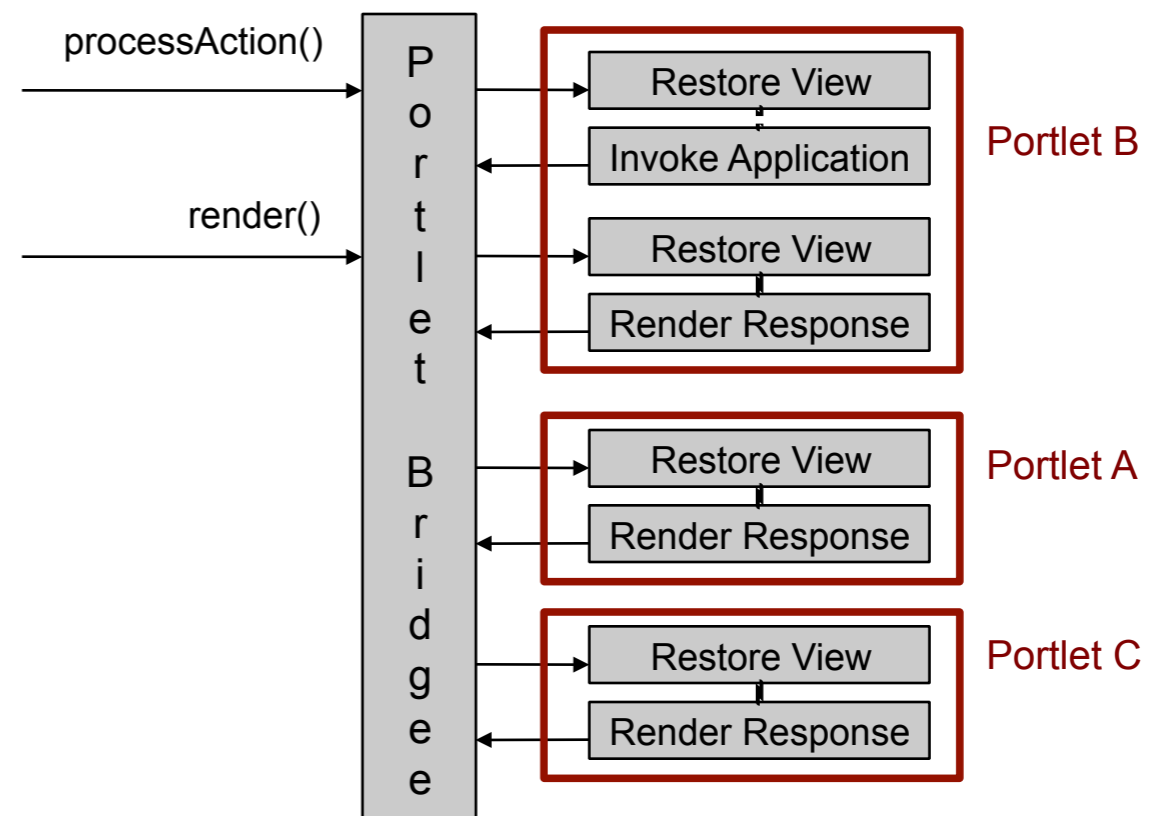
# JSF vs. Portlets

- Portlets unterscheiden zwischen
  - Abarbeitung einer Aktion
  - Darstellung
- JSF verbindet Aktionsverarbeitung und Darstellung



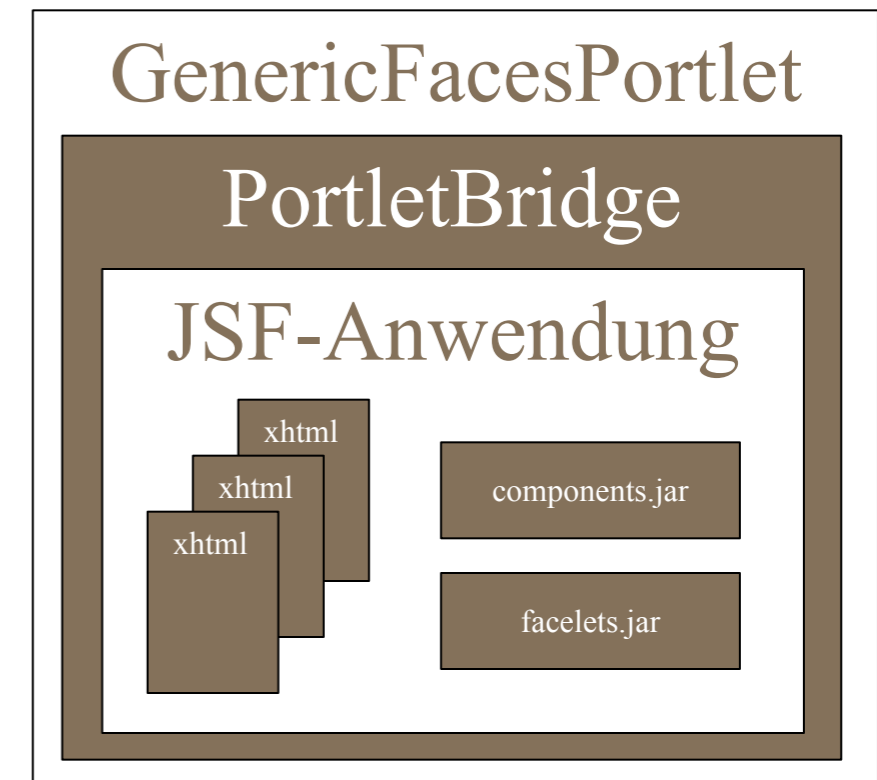
# Portletbridges

- Bilden JSF-Lifecycle auf Portlet-Phasen ab
- Sind inzwischen standardisiert
  - JSR-301 (JSF 1.2 und Portlet 1.0)
  - JSR-329 (JSF 1.2 und Portlet 2.0)
- Für JSF 2.0 z.Zt. noch keine Bridge verfügbar
  - Aber auf Basis von JSR-329 Bridges realisierbar



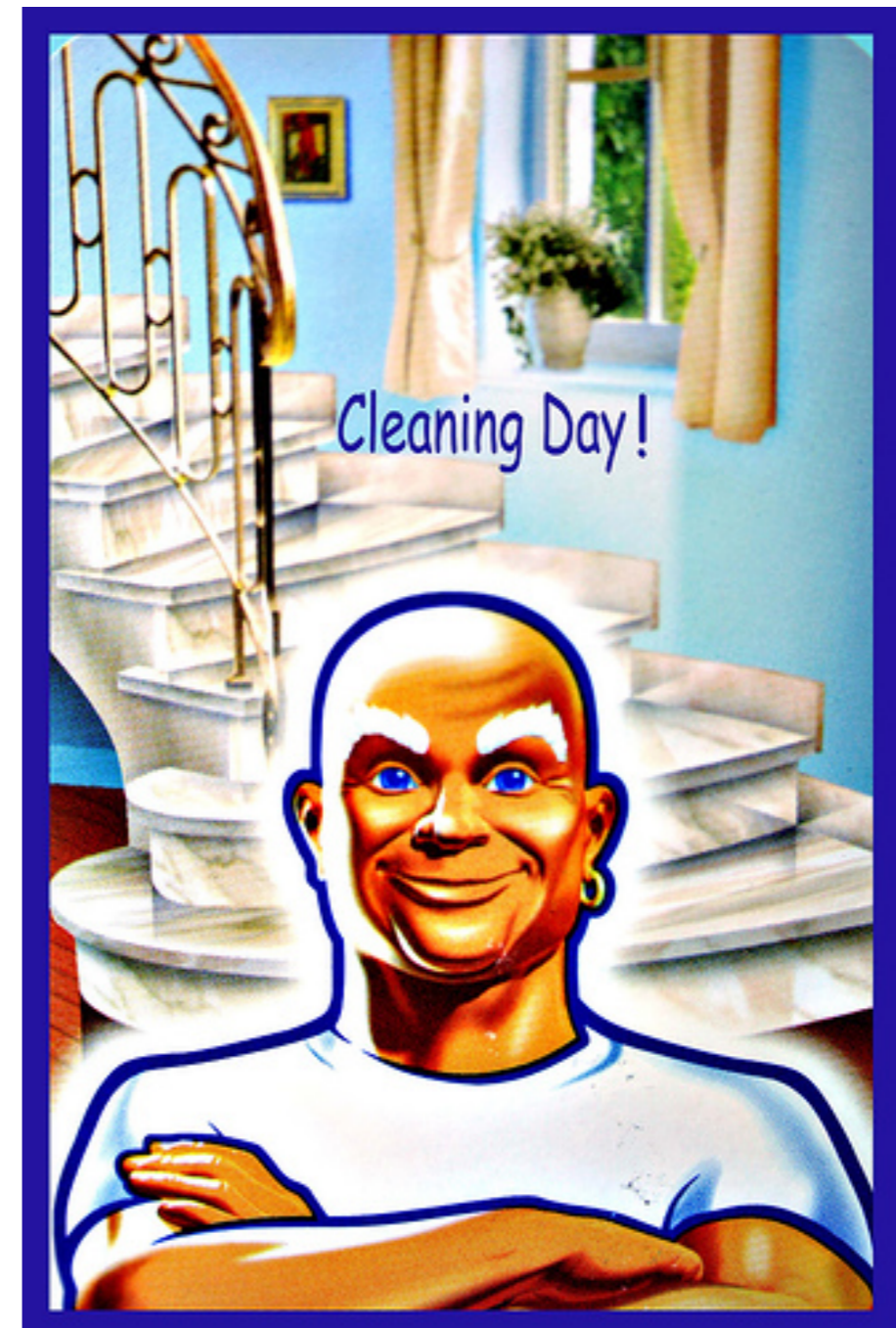
# JSF-Anwendungen im Portal

- Entwicklung einer „reinen“ JSF-Anwendung (KEIN Portlet!)
- Keine Portlet-Abhängigkeiten im Code
- Änderungen im Lifecycle sind transparent
- Spez. View-Handler für Portlet-Requests
- „Saubere“ JSF-Implementierung notwendig



# Sauber bleiben!

- Kein direkter Zugriff auf den Request
- Keine HTML, HEAD, BODY-Tags
- Vorsicht mit Redirects







## Wie sieht's technisch aus (2/4)?

- Controller und Modell in Form einer Managed Bean

```
@javax.faces.bean.SessionScoped
@javax.faces.bean.ManagedBean(name="hello")
public class GreetingBean {

    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String greet(){
        return "hello";
    }
}
```

# Wie sieht's technisch aus (3/4)?

- Die obligatorischen Deskriptoren

```
<faces-config version="2.0" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd">
  <application>
    <view-handler>org.jboss.portletbridge.application.PortletViewHandler</view-handler>
    <state-manager>org.jboss.portletbridge.application.PortletStateManager</state-manager>
  </application>
</faces-config>
```

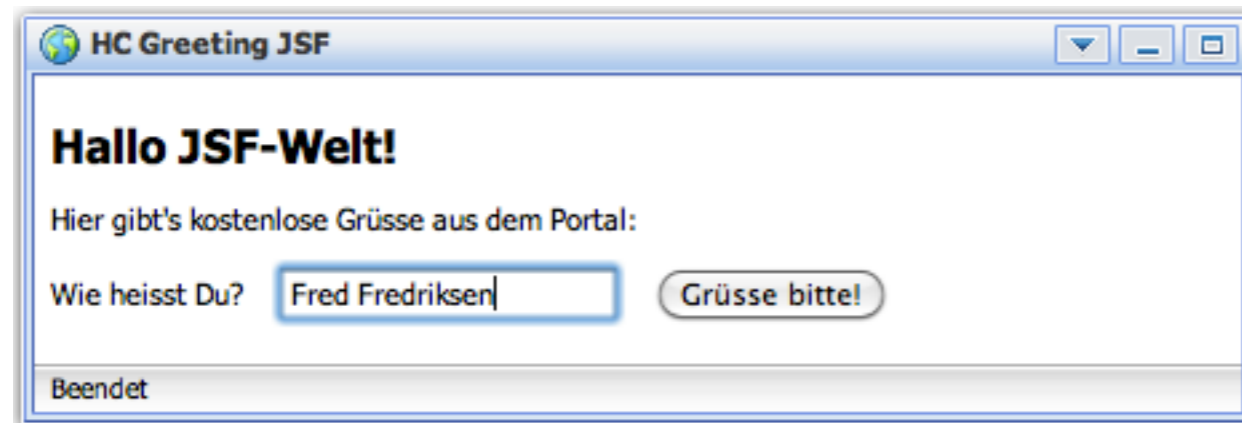
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">
  [...] <!-- klassische JSF-Konfiguration-->
</web-app>
```

# Wie sieht's technisch aus (4/4)?

- Noch ein Deskriptor (portlet.xml)

```
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd/etc/opt/
SUNWportal/dtd/portlet.xsd" version="2.0">
  <portlet>
    <portlet-name>jsf.portlet-greetingJSF</portlet-name>
    <portlet-class>javax.portlet.faces.GenericFacesPortlet</portlet-class>
    <supports>
      <mime-type>text/html</mime-type>
      <portlet-mode>VIEW</portlet-mode>
    </supports>
    <portlet-info>
      <title>herbstcampus.greeting.jsf</title>
    </portlet-info>
    <init-param>
      <name>javax.portlet.faces.defaultViewId.view</name>
      <value>/hello.xhtml</value>
    </init-param>
    <init-param>
      <name>javax.portlet.faces.preserveActionParams</name>
      <value>>true</value>
    </init-param>
  </portlet>
</portlet-app>
```

# Und so sieht's im Portal aus



# Kommunikation ist wichtig

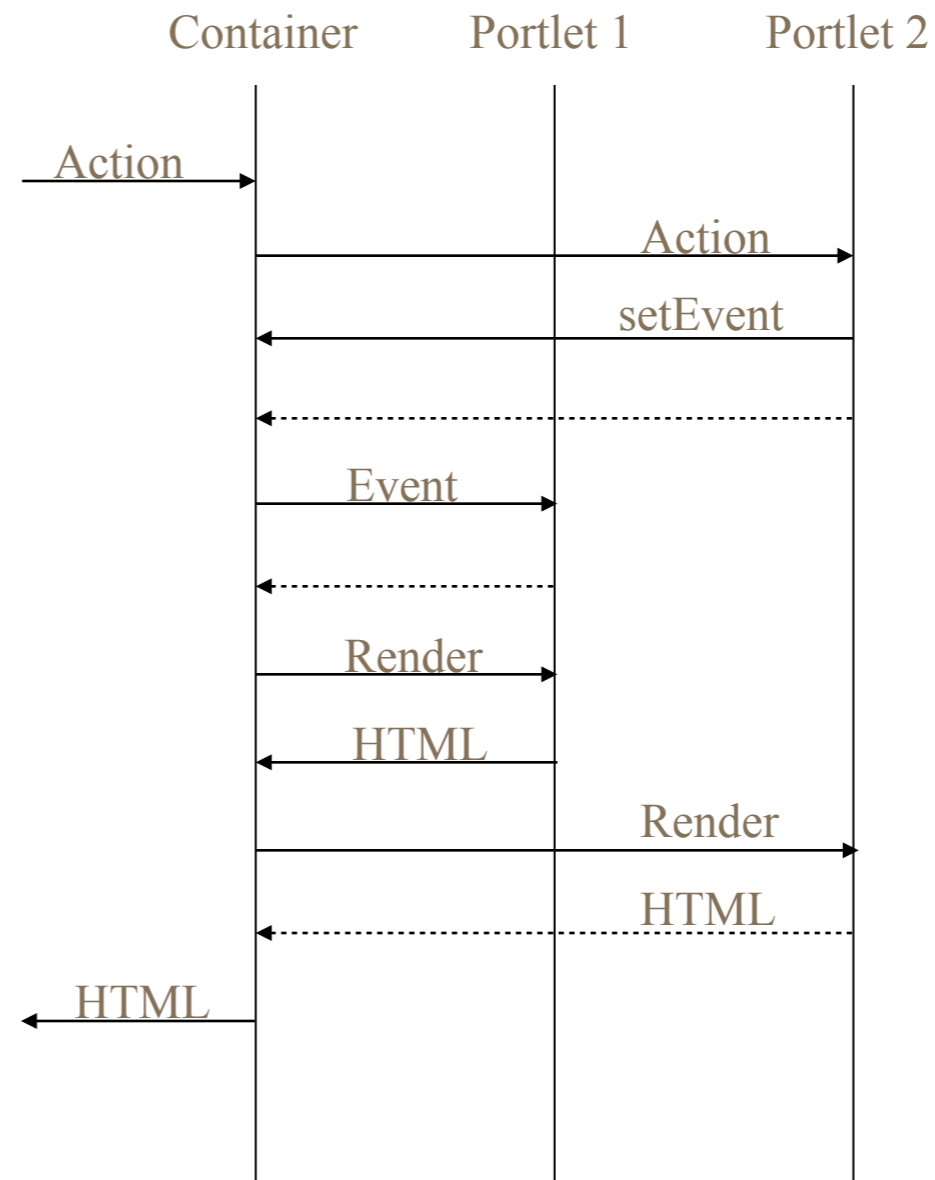
---

- Portlets sind Teile einer gemeinsamen Seite
- Kommunikation untereinander wünschenswert
- Portlet 2.0 Standard (JSR-286) bietet Abhilfe



# Portlet-Events (1/2)

- Observer-Pattern
- Portlets können Events empfangen
  - EventPhase
- Portlets können Events verschicken
  - Action-Phase
  - EventPhase



## Portlet-Events (2/2)

- Portlet-Events
  - besitzen einen Namen
  - können serialisierbare Objekte transportieren (JAX-B)

```
@javax.xml.bind.annotation.XmlRootElement
public class Registration implements java.io.Serializable {

    private String name;

    //...

    public Registration() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    //...
}
```



## Wie sieht's technisch aus (1/3)?

- Versenden eines Events (Portlet-Implementierung)

```
import javax.portlet.*;
import java.io.IOException;

public class RegistrationPortlet extends GenericPortlet {

    //...

    @Override
    public void processAction(ActionRequest request, ActionResponse response)
        throws PortletException, IOException {
        //...
        response.setEvent("registration", new Registration(name, firstname, email));
    }
}
```

## Wie sieht's technisch aus (2/3)?

- Empfangen eines Events (Portlet-Implementierung)

```
import javax.portlet.*;
import java.io.IOException;

public class RegistrationPortlet extends GenericPortlet {

    //...

    @Override
    public void processEvent(EventRequest request, EventResponse response)
        throws PortletException, IOException {
        if ("registration".equals(request.getEvent().getName())
            && request.getEvent().getValue() != null) {
            List<Registration> registrations = getRegistrations(request);
            registrations.add((Registration) request.getEvent().getValue());
            Collections.sort(registrations);
        }
    }
}
```

## Wie sieht's technisch aus (3/3)?

- Konfiguration von Sender, Empfänger und Event (portlet.xml)

```
<portlet-app>
  <portlet>
    <portlet-class>de.mathema.jsf.portlet.example.portlet.RegistrationPortlet</portlet-class>
    ...
    <supported-publishing-event>
      <name>registration</name>
    </supported-publishing-event>
  </portlet>

  <portlet>
    <portlet-class>de.mathema.jsf.portlet.example.portlet.ReportPortlet</portlet-class>
    ...
    <supported-processing-event>
      <name>registration</name>
    </supported-processing-event>
  </portlet>

  <event-definition>
    <name>registration</name>
    <value-type>de.mathema.jsf.portlet.example.portlet.Registration
    </value-type>
  </event-definition>
</portlet-app>
```

Ich hätte da eine Lieferung...



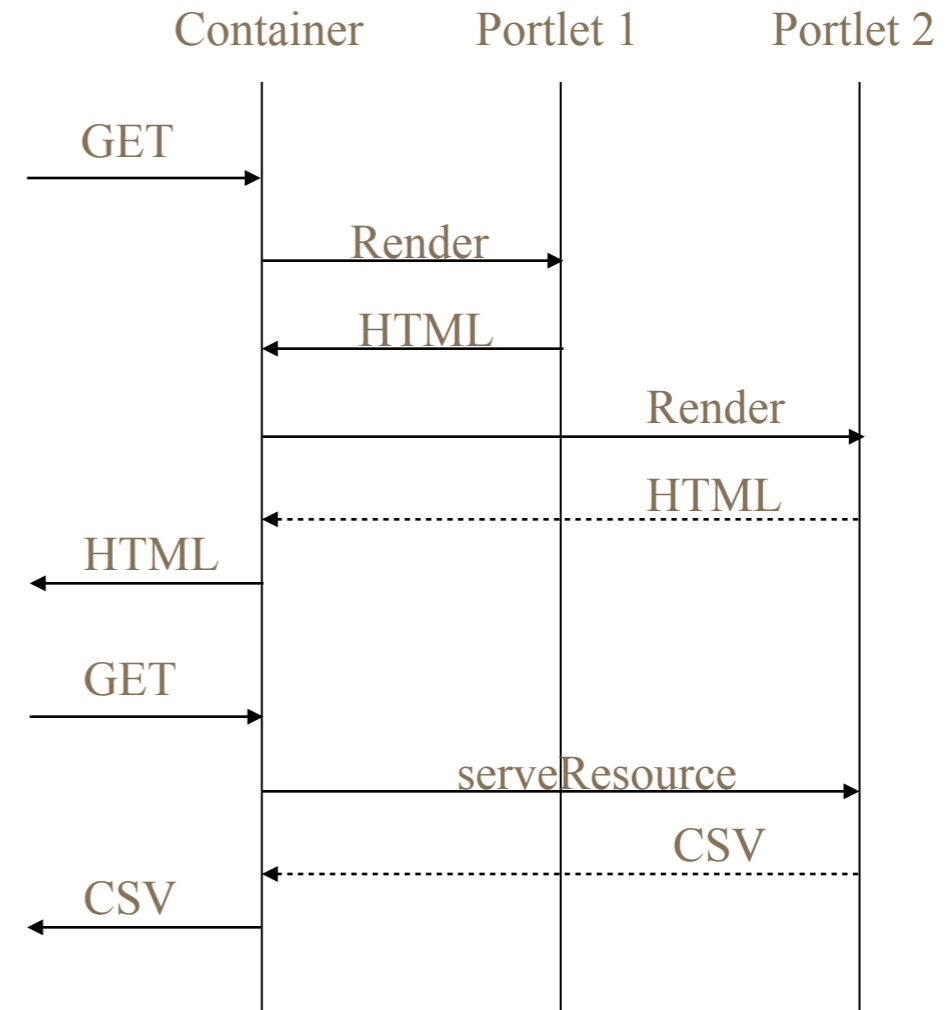
## Resource-Handling (1/2)

---

- Portalseiten bestehen nicht nur aus Text
- Bilder, Stylesheets, JavaScript müssen referenziert werden
- Evtl. werden dynamisch erzeugte Ressourcen ausgeliefert

## Resource-Handling (2/2)

- ResourceURL als abstrakte Referenz
  - Beinhaltet u.a. ResourceID
- Anfragen werden an das jeweilige Portlet weitergeleitet
- Portlet muss Ressourcen ausliefern



- Resource-Handling ist NICHT Teil des Action/Render Zyklus

# Wie sieht's technisch aus (1/2)?

- Erzeugung und Einbettung der ResourceURLs (JSP)

```
<%@ taglib prefix="p" uri="http://java.sun.com/portlet_2_0" %>
<div style='color: #444444; background: #E1DFD9;'>
  <p:actionURL var="toggleDetailsURL" name='viewAction' />
  <jsp:include page="/header.jsp" />
  <div style='padding:5px;font-size:${fontsize}px;'>
    <h2 style='color: #880000'>Registrierungsübersicht</h2>
    <p>Die folgenden Teilnehmer haben sich für den Herbstcampus 2010
registriert:</p>
    <table style='text-align:left;margin-left:20px;width:90%'>
      ...
    </table>
    <p style='margin-top:20px;'>
      <p:resourceURL var="downloadURL" id="ReportPortlet.Download" />
      <p:resourceURL var="iconURL" id="AbstractResourcePortlet.CsvIcon" />
      <a href='${downloadURL}' target='_BLANK'>
        <img src='${iconURL}' style='padding-right:5px;margin-top:-6px;'>
        Als Liste herunterladen
      </a>
    </p>
  </div>
```

## Wie sieht's technisch aus (2/2)?

- Ausliefern einer Ressource (Portlet-Implementierung)

```
import javax.portlet.*;
import java.io.IOException;

public class ReportPortlet extends GenericPortlet {

    //...

    @Override
    public void serveResource(ResourceRequest request, ResourceResponse response)
        throws PortletException, IOException {
        if (RESOURCE_DOWNLOAD.equals(request.getResourceID())) {
            response.setContentType("text/plain");
            PrintWriter writer = response.getWriter();
            writer.write(getRegistrationsAsCSVString(request));
            writer.flush();
            writer.close();
        }
    }
}
```



Und hinter der Brücke?



# Eventing mit JSF

---

- JSF kennt kein vergleichbares Event-Konzept
- Empfang von Events über Bridge möglich
  - JSR-286: `javax.portlet.faces.BridgeEventHandler`
- Versenden von Events nicht unterstützt
  - Eigenimplementierung notwendig
- **WICHTIG: Portal-Abhängigkeiten minimieren!**

# Empfangen von Events (1/2)

- Implementierung eines BridgeEventHandlers

```
public class EventHandler implements BridgeEventHandler {

    @Override
    public EventNavigationResult handleEvent(FacesContext context, Event event) {
        ELContext elContext = FacesContext.getCurrentInstance().getELContext();
        Object eventHandlingBean = elContext.getELResolver().getValue(
            elContext, null, EventHandlingBean.BEAN_NAME);
        if (eventHandlingBean instanceof EventHandlingBean) {
            ((EventHandlingBean) eventHandlingBean)
                .handle(new de.mathema.jsf.portlet.event.Event(
                    event));
        } else {
            System.out.println("No handler found!");
        }
        return null;
    }
}
```

## Empfangen von Events (2/2)

- Registrierung eines BridgeEventHandlers (portlet.xml)

```
<portlet>
  <portlet-name>jsf.portlet-reportJSF</portlet-name>
  <portlet-class>javax.portlet.faces.GenericFacesPortlet</portlet-class>
  ...
  <init-param>
    <name>javax.portlet.faces.bridgeEventHandler</name>
    <value>de.mathema.jsf.portlet.event.bridge.EventHandler</value>
  </init-param>
  ...
  <supported-processing-event>
    <name>registration</name>
  </supported-processing-event>
</portlet>
```

# Minimierung der Portal-Abhängigkeiten

---

- Ohne Portlet-spezifischen Code geht es nicht
- ABER: Kapselung in zentraler ManagedBean möglich
  - SessionScoped
  - Empfängt und versendet Events
  - Wandelt Events in portlet-unabhängige Typen um
  - Listener-Konzept zur Verteilung der Events

# EventBrokerBean (1/2)

- Empfang und Propagierung von Events

```
@SessionScoped @ManagedBean(name = EventHandlingBean.BEAN_NAME)
public class EventBrokerBean implements EventHandlingBean, EventSender {

    public static EventBrokerBean getCurrentInstance() {
        ELContext elContext = FacesContext.getCurrentInstance().getELContext();
        return (EventBrokerBean) elContext.getELResolver().getValue(elContext,
            null, EventHandlingBean.BEAN_NAME);
    }

    public void register(EventListener listener) {
        this.listener.add(listener);
    }

    public void unregister(EventListener listener) {
        this.listener.remove(listener);
    }

    public void handle(Event event) {
        for (EventListener aListener : this.listener) {
            aListener.processEvent(event);
        }
    }
    ...
}
```

## EventBrokerBean (2/2)

- Versenden von Events

```
@SessionScoped
@ManagedBean(name = EventHandlingBean.BEAN_NAME)
public class EventBrokerBean implements EventHandlingBean, EventSender {

    ...

    @Override
    public void publishEvent(Event event) {
        Object response = FacesContext.getCurrentInstance()
            .getExternalContext().getResponse();
        if (response instanceof StateAwareResponse) {
            ((StateAwareResponse) response).setEvent(event.getName(), event
                .getPayload());
        } else {
            throw new IllegalStateException(
                "Unable to send event: Not in action or event phase");
        }
    }
}
```

# Wie sieht's in der Anwendung aus? (1/2)

- Empfangen von Events

```
@SessionScoped
@ManagedBean(name = "report")
public class ReportBean implements EventListener {

    public ReportBean() {
        EventBrokerBean.getCurrentInstance().register(this);
    }

    @Override
    public void processEvent(Event event) {
        if ("registration".equals(event.getName())) {
            registrations.add(new RegistrationWrapper((Registration) event.getPayload(),
                this));
        }
    }
    ...
}
```



# Wie sieht's in der Anwendung aus? (2/2)

- Versenden von Events

```
@RequestScoped
@ManagedBean(name = "registration")
public class RegistrationBean {

    ...
    public String register() {
        try {
            EventBrokerBean.getCurrentInstance().publishEvent(
                new Event("registration",
                    new Registration(name, firstName, email)));
        } catch (IllegalStateException e) {
            FacesContext.getCurrentInstance().addMessage(
                null,
                new FacesMessage(FacesMessage.SEVERITY_WARN, e.getClass()
                    .getName(), e.getMessage()));
        }
        return "confirmation";
    }
}
```

## Resource Handling mit JSF

---

- Resource Handling in JSF mit anderer Zielrichtung
- Eigenimplementierung ähnlich zu Eventing möglich
- ABER: Mit aktueller EL auch portal-unabhängige Lösung möglich

# Resource Handling mit JSF

- Einfache Weg für statische Ressourcen

```
@ManagedBean(name = "resources")
@ApplicationScoped
public class ResourceBean implements java.util.Map<String, String> {

    public String get(Object key) {
        FacesContext facesContext = FacesContext.getCurrentInstance();
        String url = null;
        if(null == key){
            url = null;
        } else if(null != facesContext){
            url = facesContext.getApplication().getViewHandler()
                .getResourceURL(facesContext, key.toString());
            url = facesContext.getExternalContext().encodeResourceURL(url);
        } else {
            url = key.toString();
        }
        return url;
    }
}
```

```

```

# Portables Resource Handling mit JSF

- Eigenimplementierung mit EL 2.0 und MethodExpression

```
<ui:composition template="/WEB-INF/template.xhtml"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:jp="http://www.mathema.de/components/jsf/portlet">

  Name <h:inputText value="#{hello.name}" />

  <h1>Resource</h1>
  <jp:resource resource="#{resourceBean.resource()}" value="resource" />
  <jp:resource resource="#{resourceBean.resource2(hello.name)}"
    value="resource2(hello.name)" />

  <jp:resource resource="#{report.downloadCSV()}" value="Als Liste herunterladen" />
```

# Portables Resource Handling mit JSF

- Ausliefern der Resource

```
public void downloadCSV() {
    StringBuilder csvString = new StringBuilder("name;vorname;email;\n");
    for (RegistrationWrapper registration : registrations) {
        csvString.append(registration.getRegistration().getName());
        csvString.append(";");
        csvString.append(registration.getRegistration().getFirstName());
        csvString.append(";");
        csvString.append(registration.getRegistration().getEmail());
        csvString.append(";\n");
    }
    ResourceDeliverUtil.deliver(FacesContext.getCurrentInstance(),
        new StringReader(csvString.toString()), "text/plain", "UTF-8",
        "registrations", true);
}
```

# Portables Resource Handling mit JSF

---

- Konfiguration in web.xml

```
<context-param>  
  <param-name>com.sun.faces.expressionFactory</param-name>  
  <param-value>de.odysseus.el.ExpressionFactoryImpl</param-value>  
</context-param>
```

- Verwendung von JUEL <http://juel.sourceforge.net/>

# Zweigleisig fahren



# Gründe für Parallelobetrieb

---

- Unterstützung verschiedener Webauftritte
- Erhöht Flexibilität
- Vereinfacht die Entwicklung



# Nötiges Handwerkszeug

- Grundsätzlich identische Vorgehensweise wie im Portal
  - Saubere JSF-Implementierung vorausgesetzt
- HTML, HEAD und BODY-Tags nur ausserhalb des Portals erlaubt
- Spezielle Taglib notwendig
- Wichtig:  
Keine Portal-Features nutzbar!



[http://commons.wikimedia.org/wiki/File:20060513\\_toolbox.jpg](http://commons.wikimedia.org/wiki/File:20060513_toolbox.jpg)

# Portal-aware Tags (1/4)

- Portlet oder Servlet?

```
public class PortletUtil {  
  
    public static boolean isPortletCall(javax.faces.FacesContext context) {  
        try {  
            return context.getExternalContext().getRequest() instanceof  
                javax.portlet.PortletRequest;  
        } catch (java.lang.NoClassDefFoundError e) {  
            // No PortletRequest.class found => no portlet call :)  
            return false;  
        }  
    }  
}
```

# Portal-aware Tags (2/4)

- HTML-Tag als JSF-Komponente

```
@FacesComponent("de.mathema.jsf.portlet.taglib.Html")
public class Html extends UIComponentBase {

    @Override
    public void encodeBegin(FacesContext context) throws IOException {
        if (!PortletUtil.isPortletCall(context)) {
            ResponseWriter writer = context.getResponseWriter();
            writer.startElement("html", null);
        }
    }

    @Override
    public void encodeEnd(FacesContext context) throws IOException {
        if (!PortletUtil.isPortletCall(context)) {
            ResponseWriter writer = context.getResponseWriter();
            writer.endElement("html");
        }
    }

    @Override
    public String getFamily() {
        return null;
    }
}
```

# Portal-aware Tags (3/4)

- Interessanter wird's beim BODY-Tag

```
@FacesComponent("de.mathema.jsf.portlet.taglib.Head")
public class Head extends HtmlHead {

    @Override
    public void encodeBegin(FacesContext context) throws IOException {
        if(!PortletUtil.isPortletCall(context)){
            super.encodeBegin(context);
        }else{
            ResponseWriter writer = context.getResponseWriter();
            writer.startElement("div", this);
            writer.writeAttribute("style", "display:none;visibility:hidden;", null);

            // now rendering resources registered for the head tag
            UIViewRoot viewRoot = context.getViewRoot();
            ListIterator<UIComponent> iter = viewRoot.getComponentResources(context,
                                                                              "head").listIterator();

            while (iter.hasNext()) {
                UIComponent resource = (UIComponent)iter.next();
                resource.encodeAll(context);
            }
        }
    }
    ...
}
```

# Portal-aware Tags (4/4)

- taglib.xml

```
<!DOCTYPE facelet-taglib PUBLIC "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
"http://java.sun.com/dtd/facelet-taglib_1_0.dtd">
<facelet-taglib>
  <namespace>http://www.mathema.de/components/jsf/portlet</namespace>
  <tag>
    <tag-name>html</tag-name>
    <component>
      <component-type>de.mathema.jsf.portlet.taglib.Html</component-type>
    </component>
  </tag>
  <tag>
    <tag-name>body</tag-name>
    <component>
      <component-type>de.mathema.jsf.portlet.taglib.Body</component-type>
    </component>
  </tag>
  ...
</facelet-taglib>
```

12.–15.09.2010  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

Vielen Dank!

Werner Eberling / Sascha Groß

MATHEMA Software GmbH