

# Enterprise Java (BI-EJA)

## Technologie programování v jazyku Java (X36TJV)

Ing. Zdeněk Troníček, Ph.D.

Katedra softwarového inženýrství

Fakulta informačních technologií ČVUT v Praze



Letní semestr 2010/2011, přednáška č. 3

<https://edux.fit.cvut.cz/courses/BI-EJA>

<https://edux.feld.cvut.cz/courses/X36TJV>

© Zdeněk Troníček, 2011

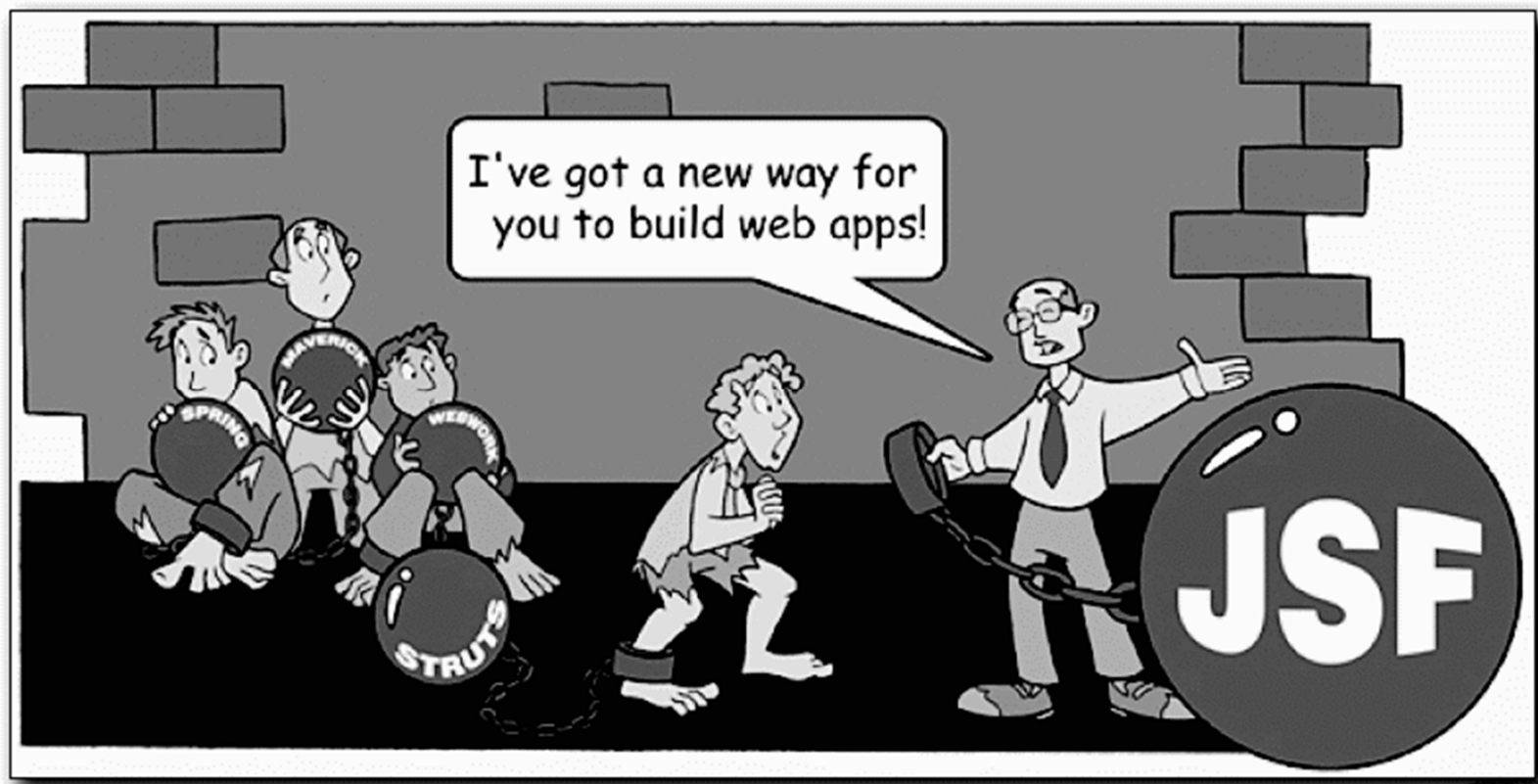
# Agenda

## Java Server Faces (JSF)

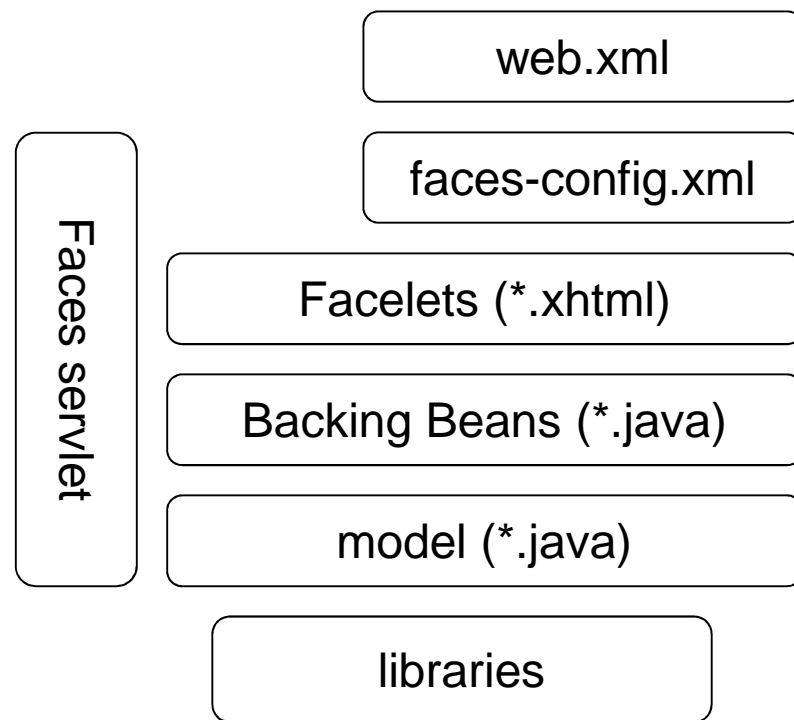
- Aplikace v JSF
- Zpracování žádosti, backing beans
- Navigace
- Šablony
- Konverze a validace
- Lokalizace
- Životní cyklus žádosti
- JSF komponenty



# JSF 1.0



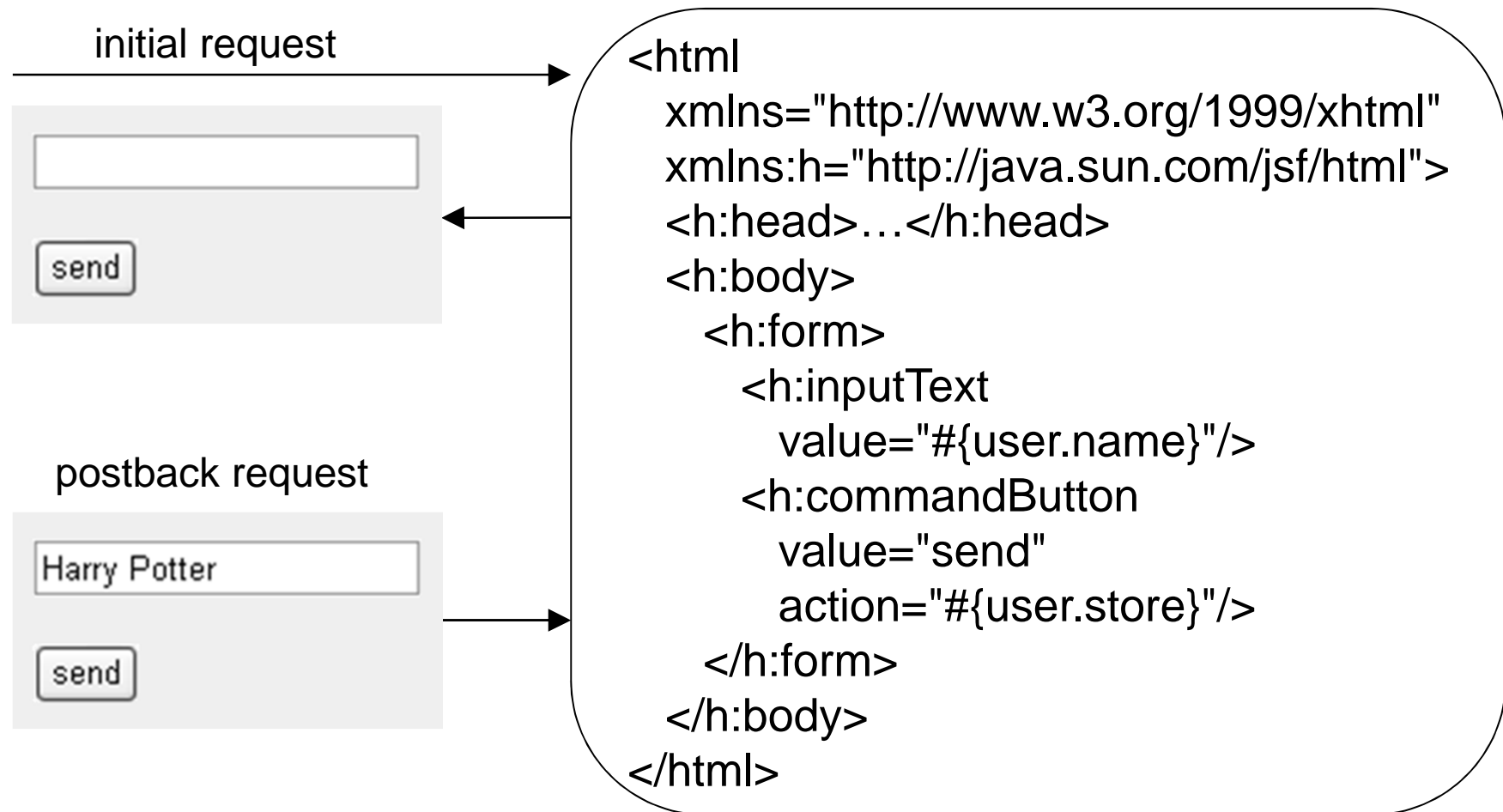
# JSF aplikace



```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>First Application</title>
  </h:head>
  <h:body>
    Hello, #{user.name}
  </h:body>
</html>
```

expression language: #{...}

# JSF Requests



# Backing bean

```
@ManagedBean(name = "user")
@RequestScoped
public class UserBean {
    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String store() {
        ...
        return "display";
    }
}
```

`#{user.name}`

initial response: getter  
postback response: getter

`<h:inputText  
value="#{user.name}"/>`

initial response: getter  
postback request: setter  
postback response: getter

# Request processing

```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html">
  <h:head>...</h:head>
  <h:body>
    <h:form>
      <h:inputText
        value="#{user.name}"/>
      <h:commandButton
        value="send"
        action="#{user.store}"/>
    </h:form>
  </h:body>
</html>
```

Initial:  
UserBean.getName()

Postback:  
UserBean.setName()  
UserBean.store()  
UserBean.getName()

# Scope (doba života)

Jaká je doba života backing beany?

- request – po dobu zpracování požadavku
- view – dokud jsme na dané stránce
- session – po dobu sezení
- application – po dobu běhu aplikace
- custom – dobu života řídíme programově



# Nastavení scope

- @RequestScoped
- @ViewScoped
- @SessionScoped
- @ApplicationScoped
- @NoneScoped
- @CustomScoped
- Flash scope

```
@ManagedBean(name = "user")
@SessionScoped
public class UserManagedBean {

    public UserManagedBean() { }

    ...
}
```

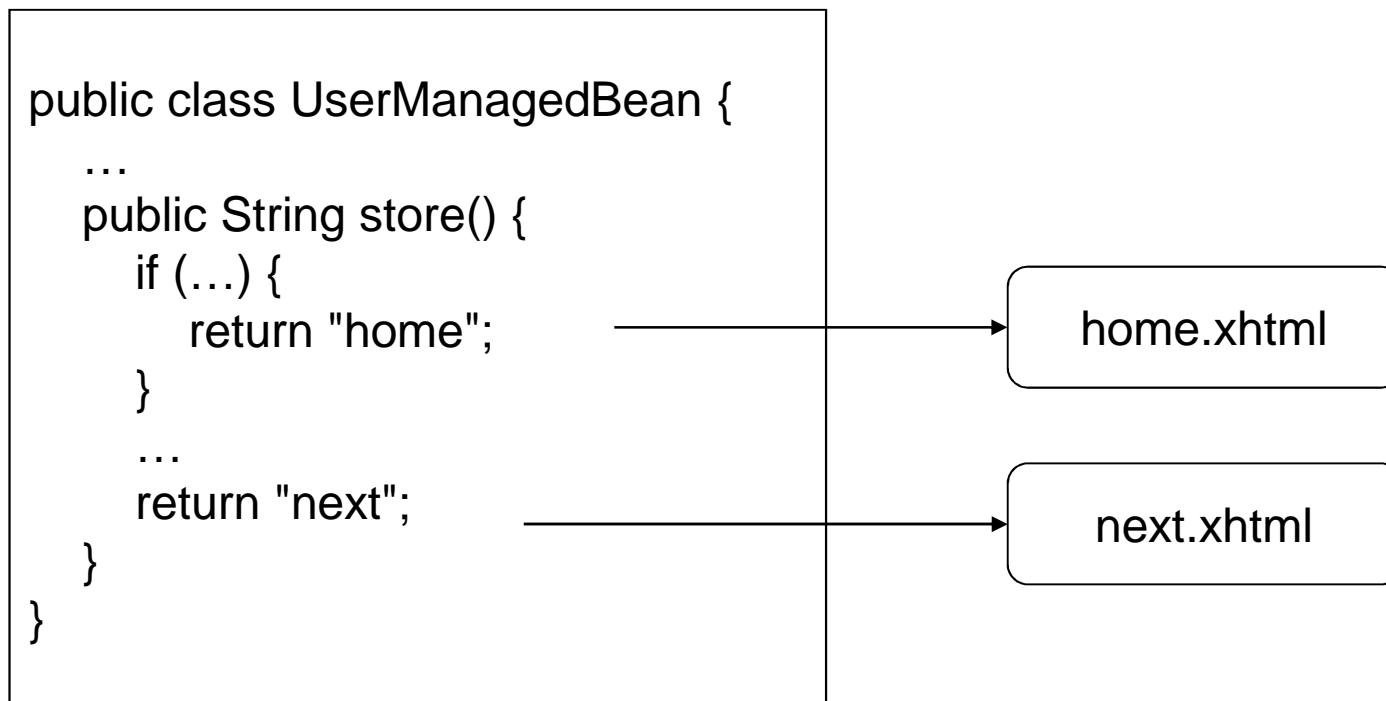
# Navigace

Kam se přejde po stisku tlačítka?

Možnosti:

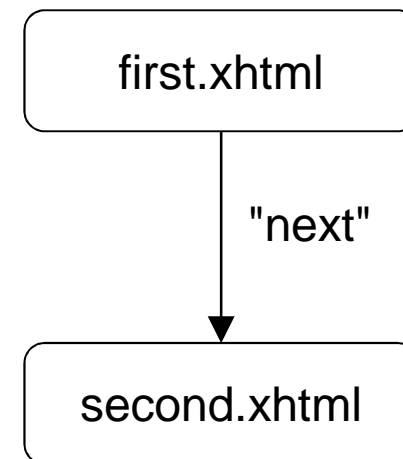
1. implicitní navigace
2. statická navigace
3. dynamická navigace

# Implicitní navigace



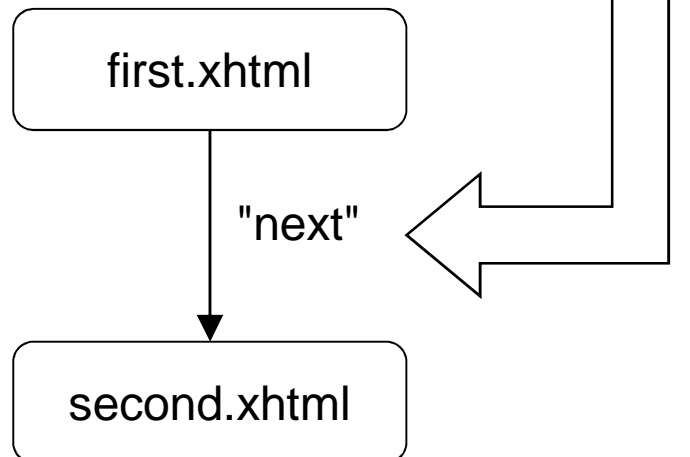
# Navigační pravidla

```
<faces-config>
  <navigation-rule>
    <from-view-id>/first.xhtml</from-view-id>
    <navigation-case>
      <from-outcome>next</from-outcome>
      <to-view-id>/second.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
  ...
</faces-config>
```



# Statická navigace

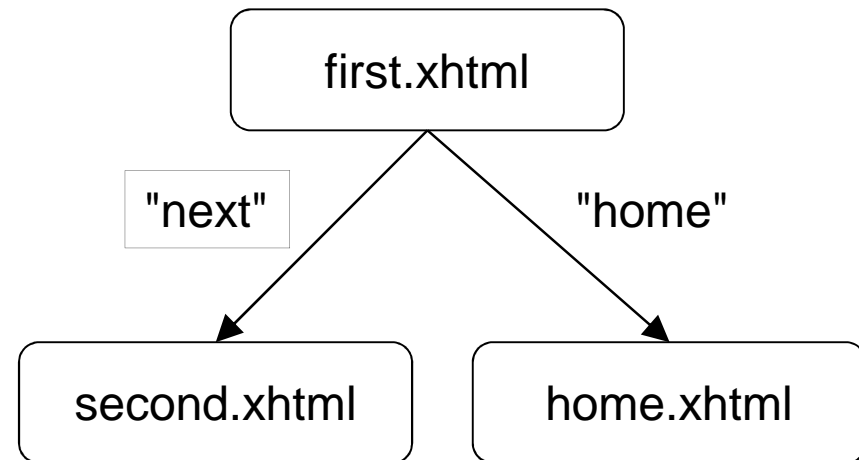
```
<h:commandButton value="send" action="next"/>
```



# Dynamická navigace

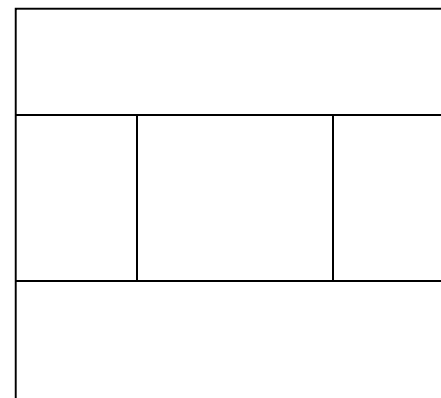
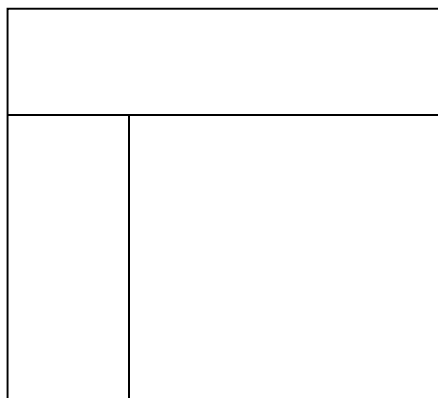
```
<h:commandButton value="send" action="#{user.store}"/>
```

```
@ManagedBean(name = "user")
public class UserManagedBean {
    ...
    public String store() {
        if (...) {
            return "home";
        }
        return "next";
    }
}
```

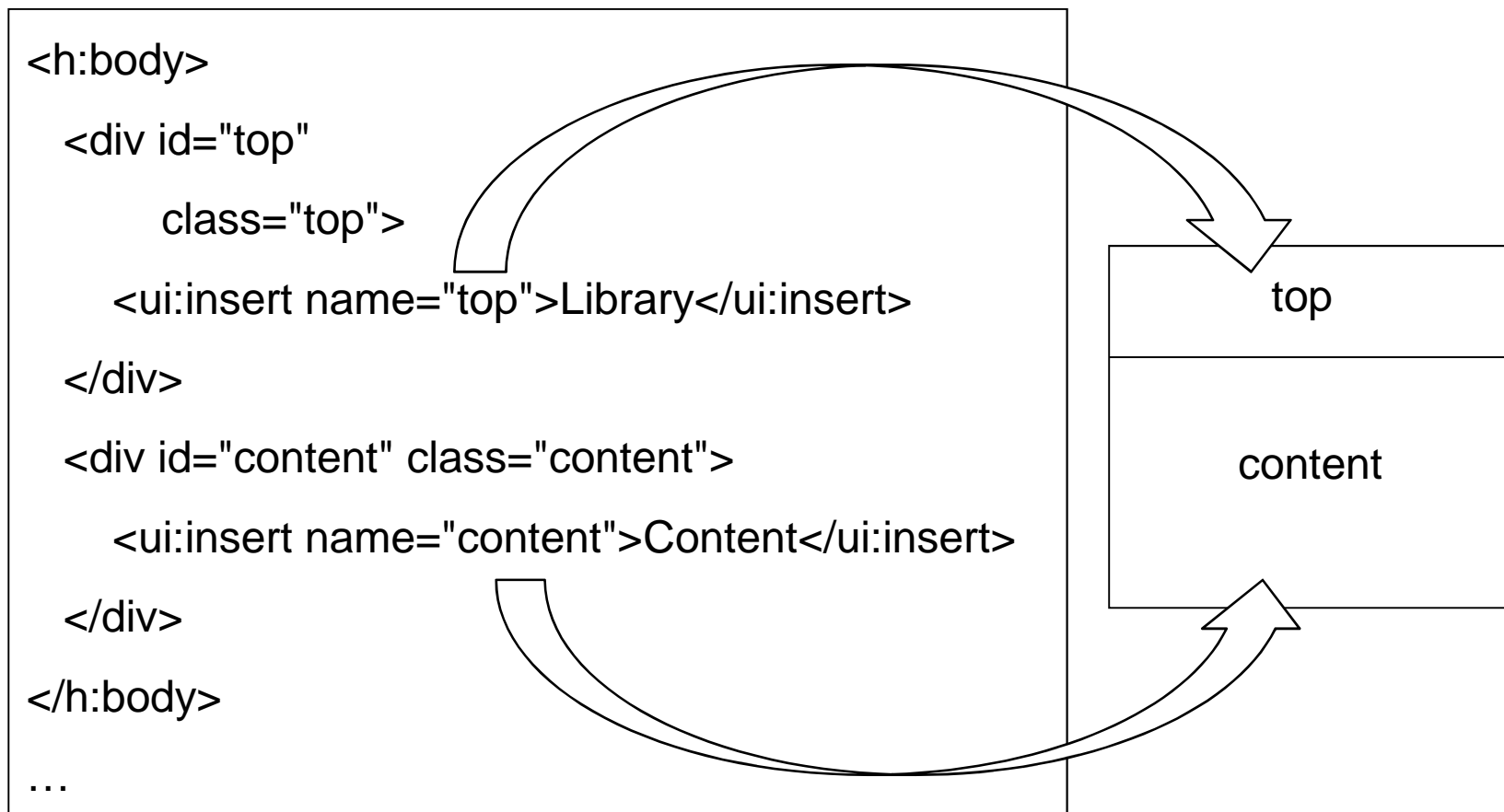


# Šablony

pro jednotný vzhled stránek



# Template

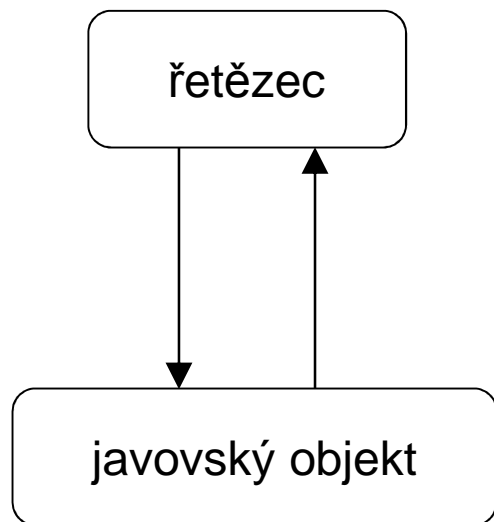




# Template client

```
<ui:composition xmlns:ui="http://java.sun.com/jsf/facelets"
  template="BasicTemplate.xhtml"
  xmlns:h="http://java.sun.com/jsf/html">
  <ui:define name="content">
    <h:form>
      Author: <h:inputText value="#{searchBean.author}"/>
      Title: <h:inputText value="#{searchBean.title}"/>
      <h:commandButton value="search" action="#{searchBean.search}"/>
    </h:form>
  </ui:define>
</ui:composition>
```

# Konverze



```
<h:inputText value="#{dvd.year}"
  required="true">
  <f:convertDateTime pattern="yyyy"/>
</h:inputText>
```

BigDecimalConverter  
BigIntegerConverter  
BooleanConverter  
ByteConverter  
CharacterConverter

DateTimeConverter  
DoubleConverter  
EnumConverter  
FloatConverter  
IntegerConverter

LongConverter  
NumberConverter  
ShortConverter

# Vlastní konvertor

```
@FacesConverter( forClass = Phone.class )
public class PhoneConverter implements Converter {

    public Object getAsObject( FacesContext fc, UIComponent comp,
        String value ) {
        // String → Object (ConverterException)
    }

    public String getAsString( FacesContext fc, UIComponent comp,
        Object value ) {
        // Object → String (ConverterException)
    }
}
```

# Validate

form data validation  
business-logic validation

```
<h:inputText value="#{user.weight}">  
  <f:validateLongRange minimum="1" maximum="200"/>  
</h:inputText>
```

```
<f:validateDoubleRange .../>  
<f:validateLongRange .../>  
<f:validateLength .../>  
<f:validateRegex pattern="\d\d\d\s+\d\d"/>  
<f:validateRequired .../>  
<f:validateBean .../>
```

# Vlastní validátor

```
@FacesValidator( "phoneValidator" )
public class PhoneValidator implements Validator {

    public void validate( FacesContext fc, UIComponent comp,
        Object value ) throws ValidatorException {
        ...
    }
}
```

```
<h:inputText value="#{user.phone}">
    <f:validator validatorId="phoneValidator"/>
</h:inputText>
```

# Bean validation

@AssertFalse  
@AssertTrue  
@DecimalMax  
@DecimalMin  
@Digits  
@Future  
@Max

@Min  
@NotNull  
@Null  
@Past  
@Pattern  
@Size

```
@ManagedBean( name = "user" )  
@RequestScoped  
public class UserManagedBean {  
  
    @Pattern( regexp = "[a-z]+@(fit|fel).cvut.cz" )  
    private String email;  
  
    ...  
}
```

# Lokalizace (soubor .properties)

Jméno souboru:

`<name>{-|_}language{-|_}country{-|_}variant.properties`

Obsah souboru:

`klíč = hodnota`

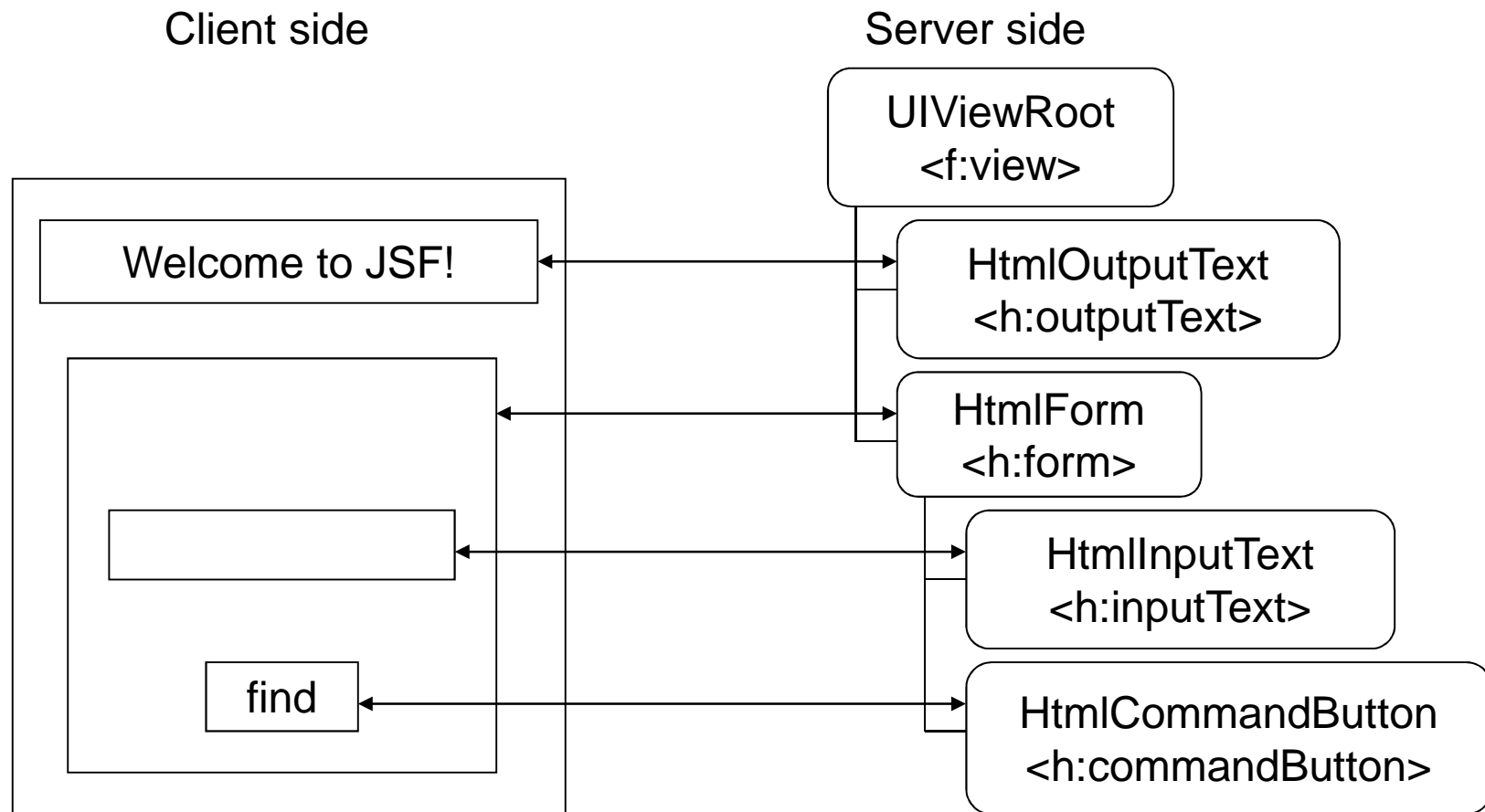
messages\_en.properties

```
username=Username  
birth=Birth  
phone=Phone  
weight=Weight
```

messages\_cs.properties

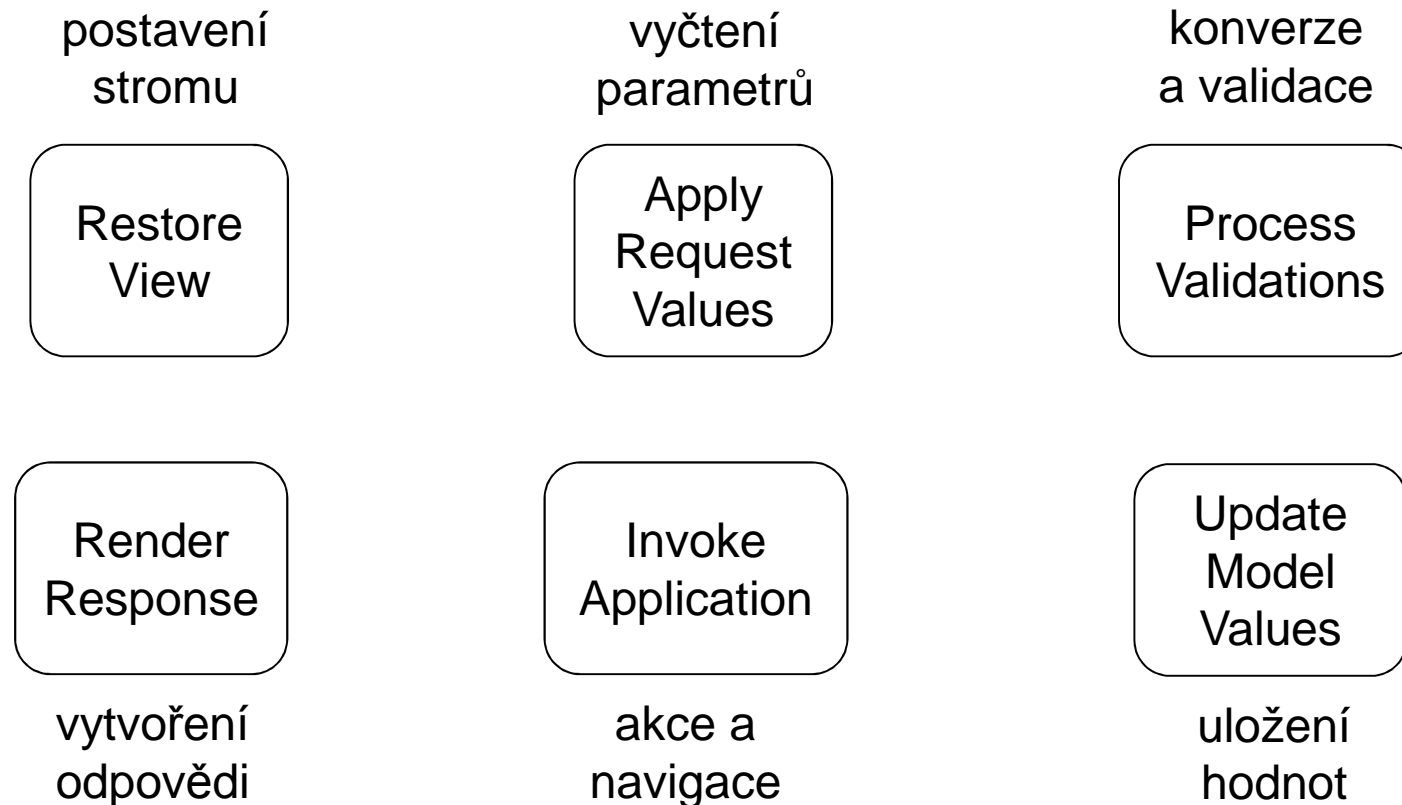
```
username=Uživatelské jméno  
birth=Datum narození  
phone=Telefon  
weight=Hmotnost
```

# Component tree

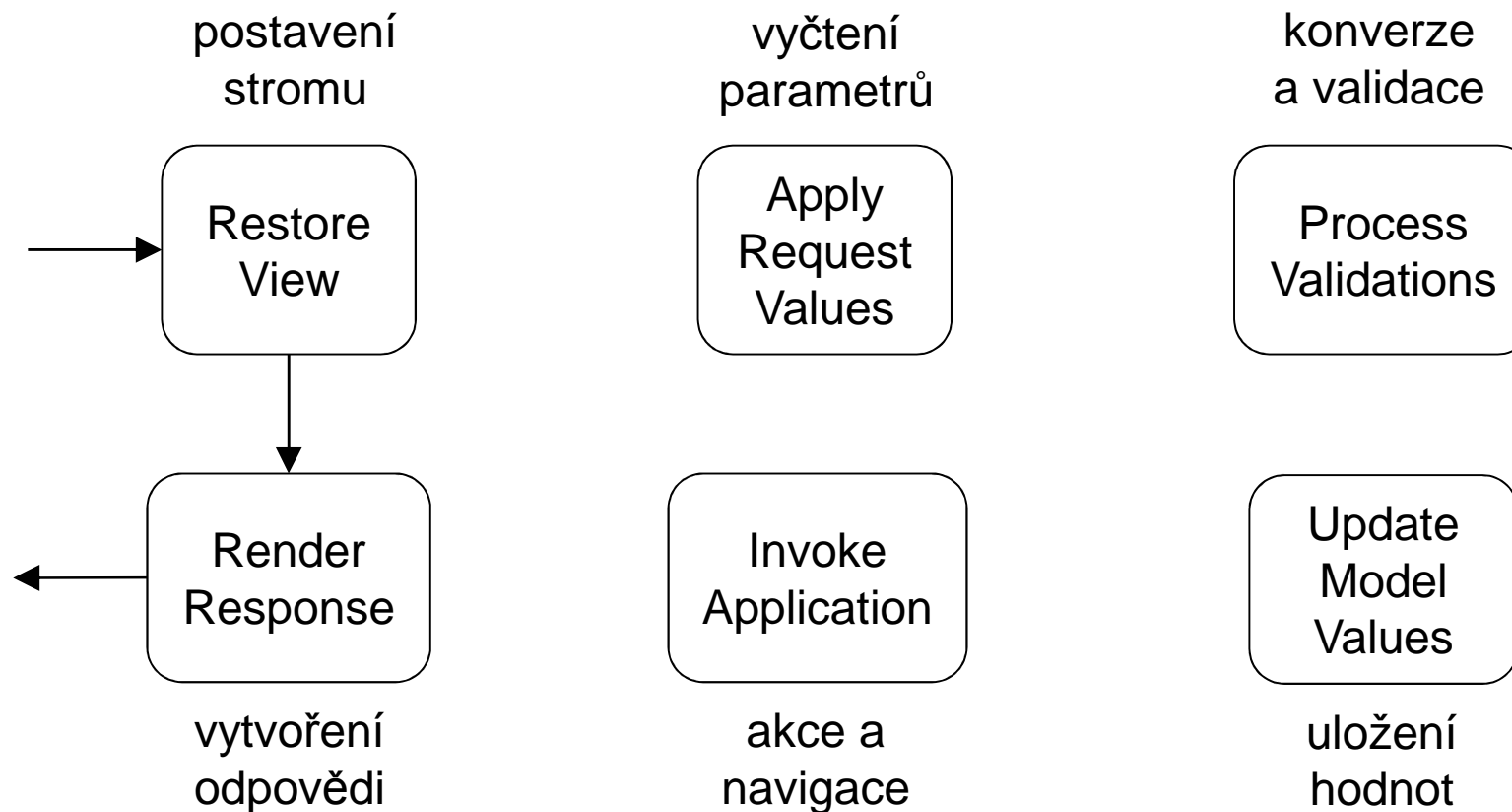




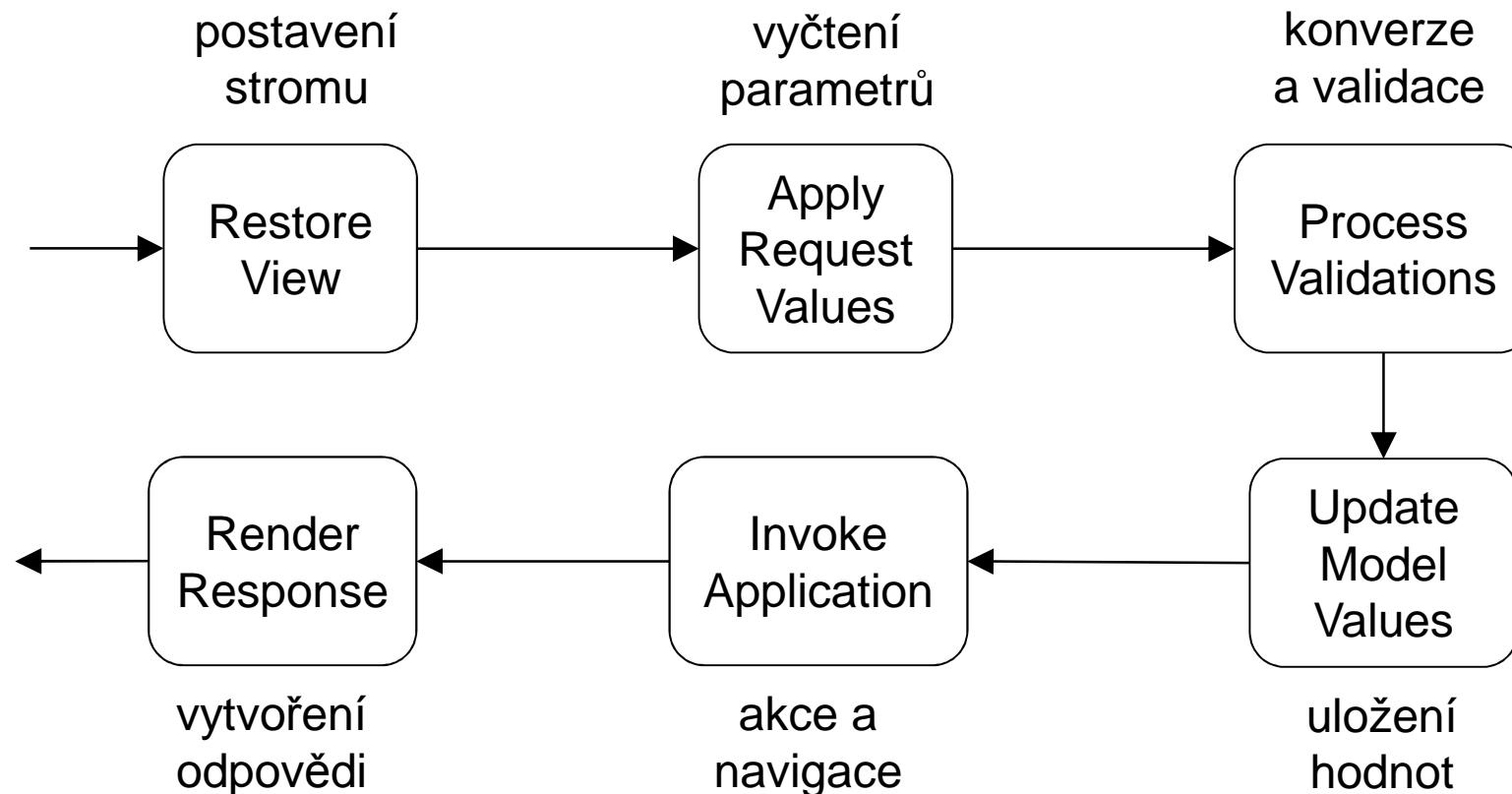
# Životní cyklus žádosti



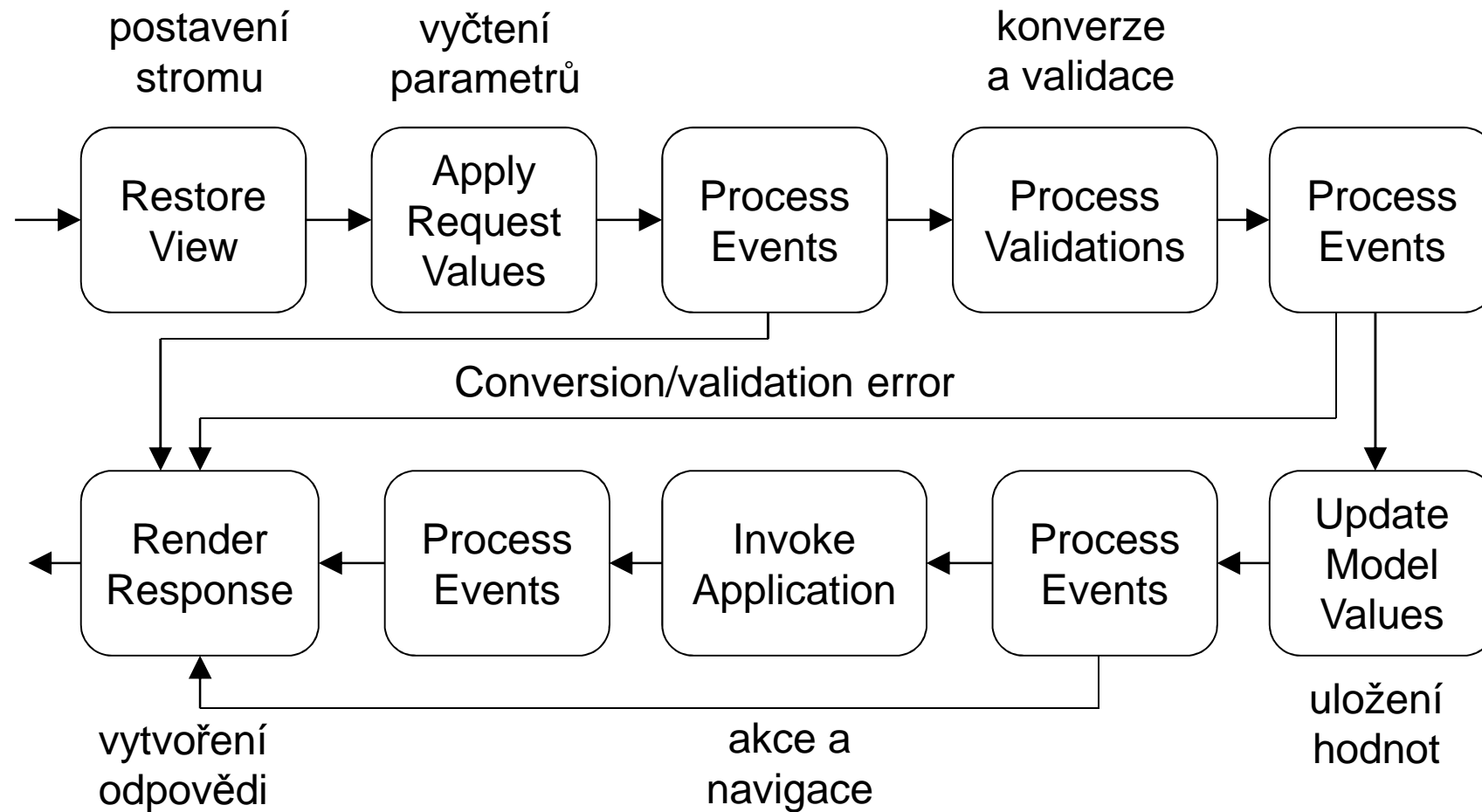
# Initial request



# Postback request

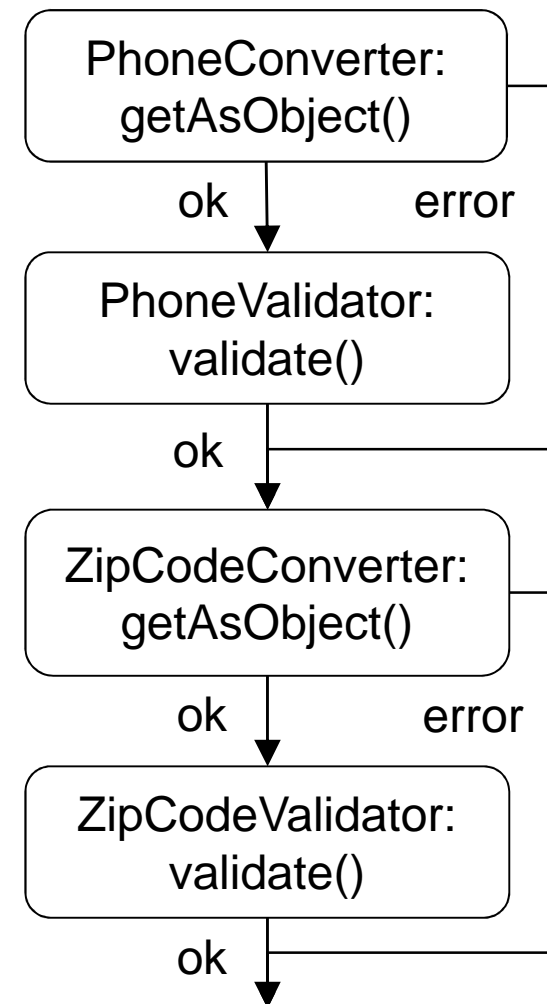


# Events

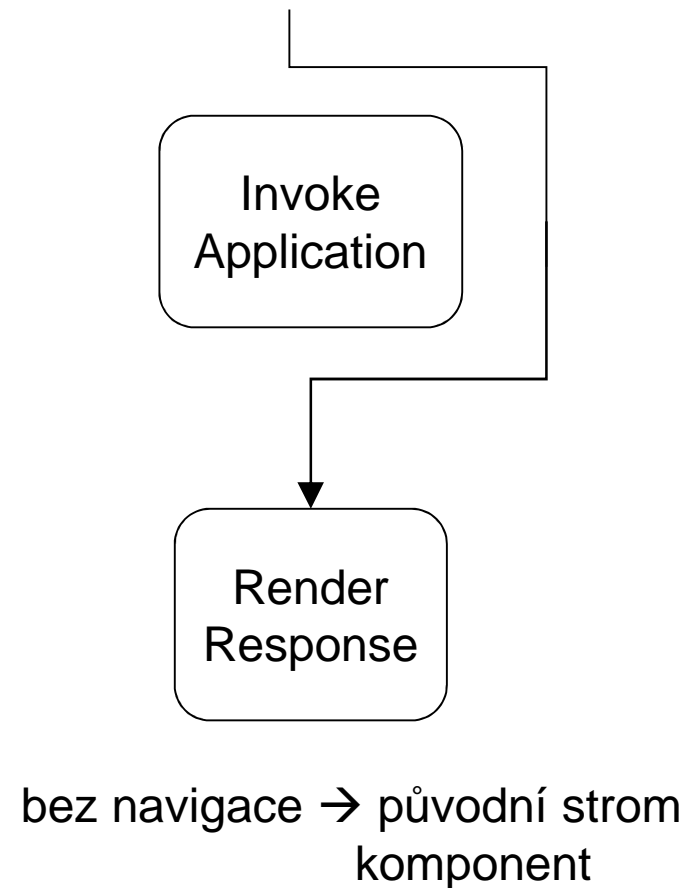
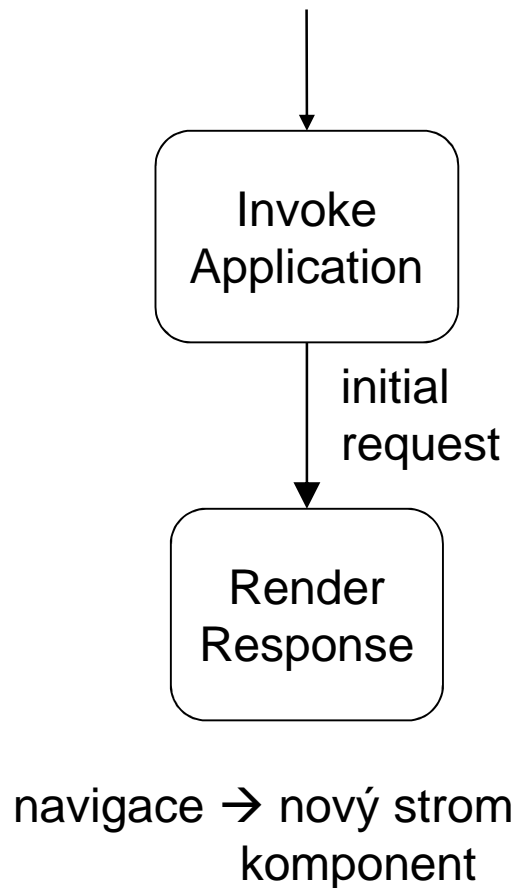


# Validation & conversion

```
<h:inputText value="#{user.phone}">
  <f:converter
    converterId="phoneConverter"/>
  <f:validator
    validatorId="phoneValidator"/>
</h:inputText>
<h:inputText value="#{addr.zipCode}">
  <f:converter
    converterId="zipCodeConverter"/>
  <f:validator
    validatorId="zipCodeValidator"/>
</h:inputText>
```



# Render response



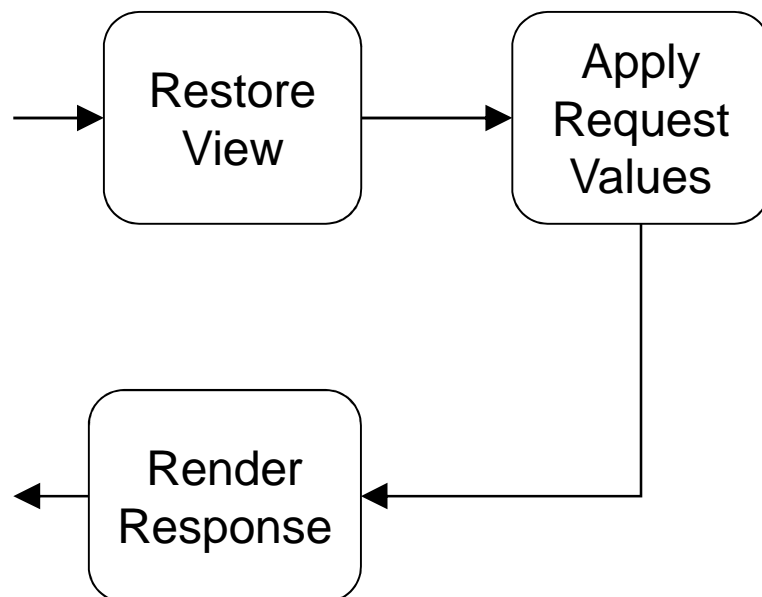
# Property “immediate” (1)

```
<h:commandLink immediate="true" ... />  
<h:commandButton immediate="true" ... />
```

ActionSource

ActionEvent

ActionListener



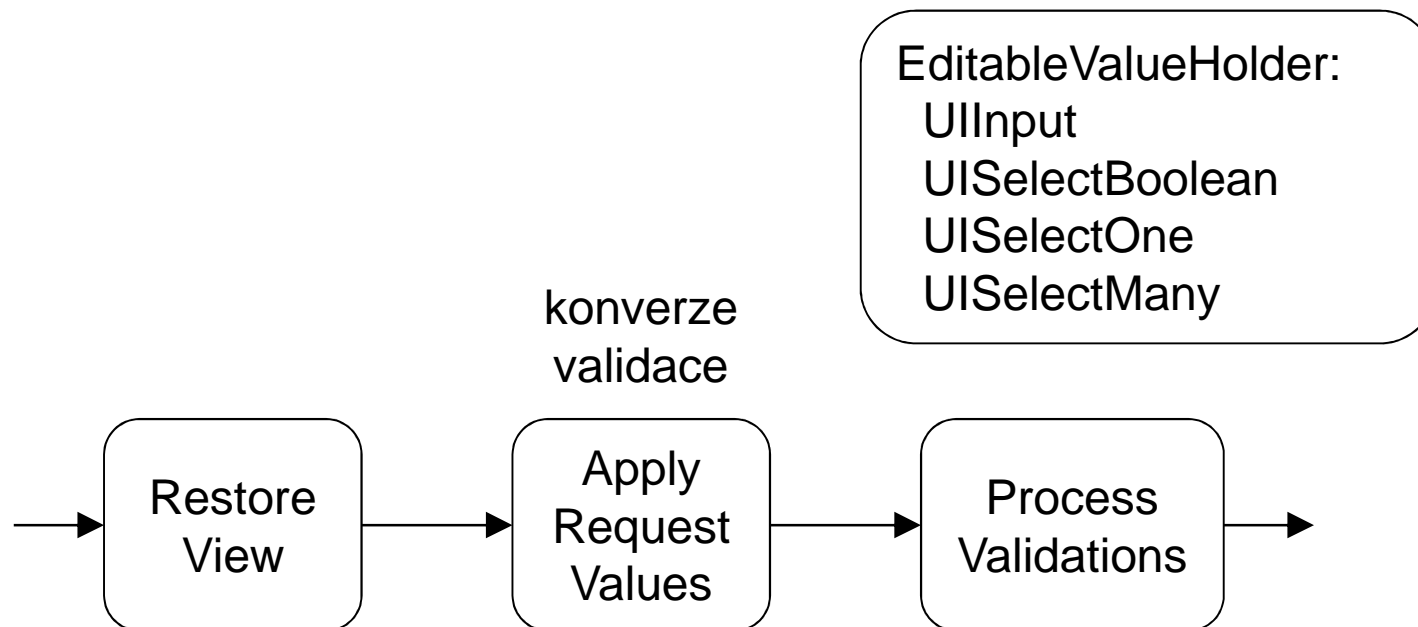
doručena  
ActionEvent

DefaultActionListener

1.akce  
2.navigace

# Property “immediate” (2)

```
<h:inputText immediate="true" ... />  
<h:selectBooleanCheckbox immediate="true" ... />
```





# JSF komponenty

Komponenta má tři části:

1. `UIComponent` – součást stromu komponent
2. `Tag` – použit ve stránce
3. `Renderer` – generuje HTML

## Standardní komponenty

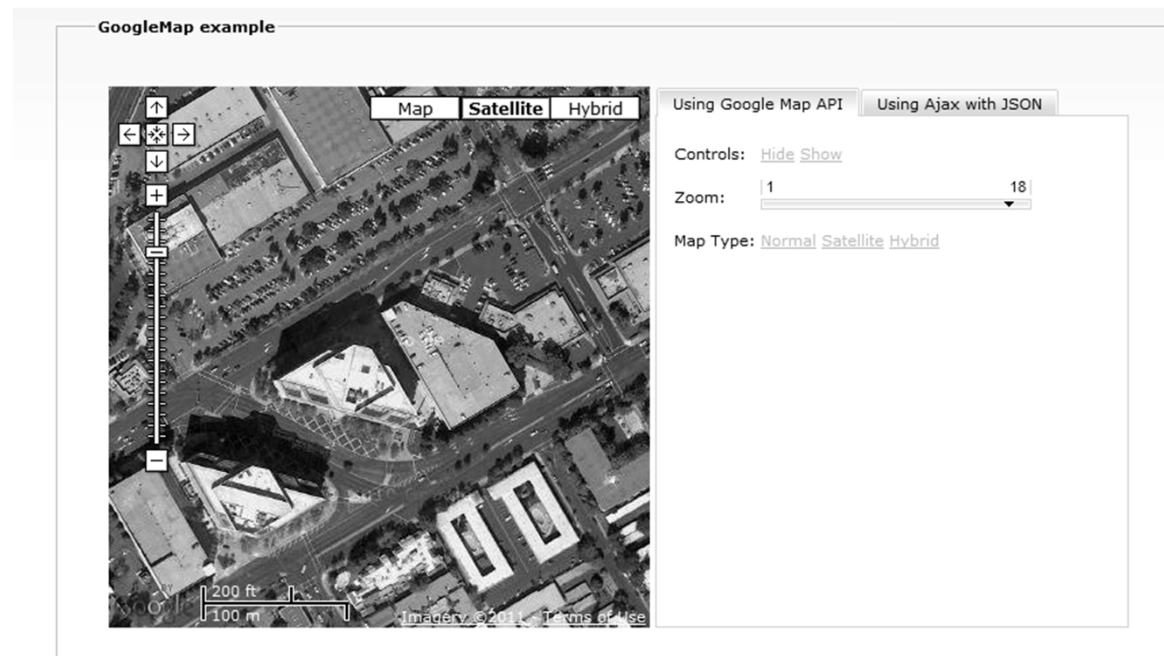
- `Tag`: `inputText`, `selectOneListbox`, `selectOneMenu`, `selectOneRadio`, `selectManyListbox`,...
- `UIComponent`: `UIInput`, `UISelectOne`, `UISelectMany`,...
- `Renderer`: `Text`, `Secret`, `Listbox`, `Menu`, `Radio`,...

Př.: `UIInput` + `Text` = `inputText`

`UISelectOne` + `Listbox` = `selectOneListbox`

# Knihovny komponent

- RichFaces, <http://www.jboss.org/richfaces>
- ICEFaces, <http://www.icefaces.org>
- ...



# Otázky & odpovědi

tronicek@fit.cvut.cz