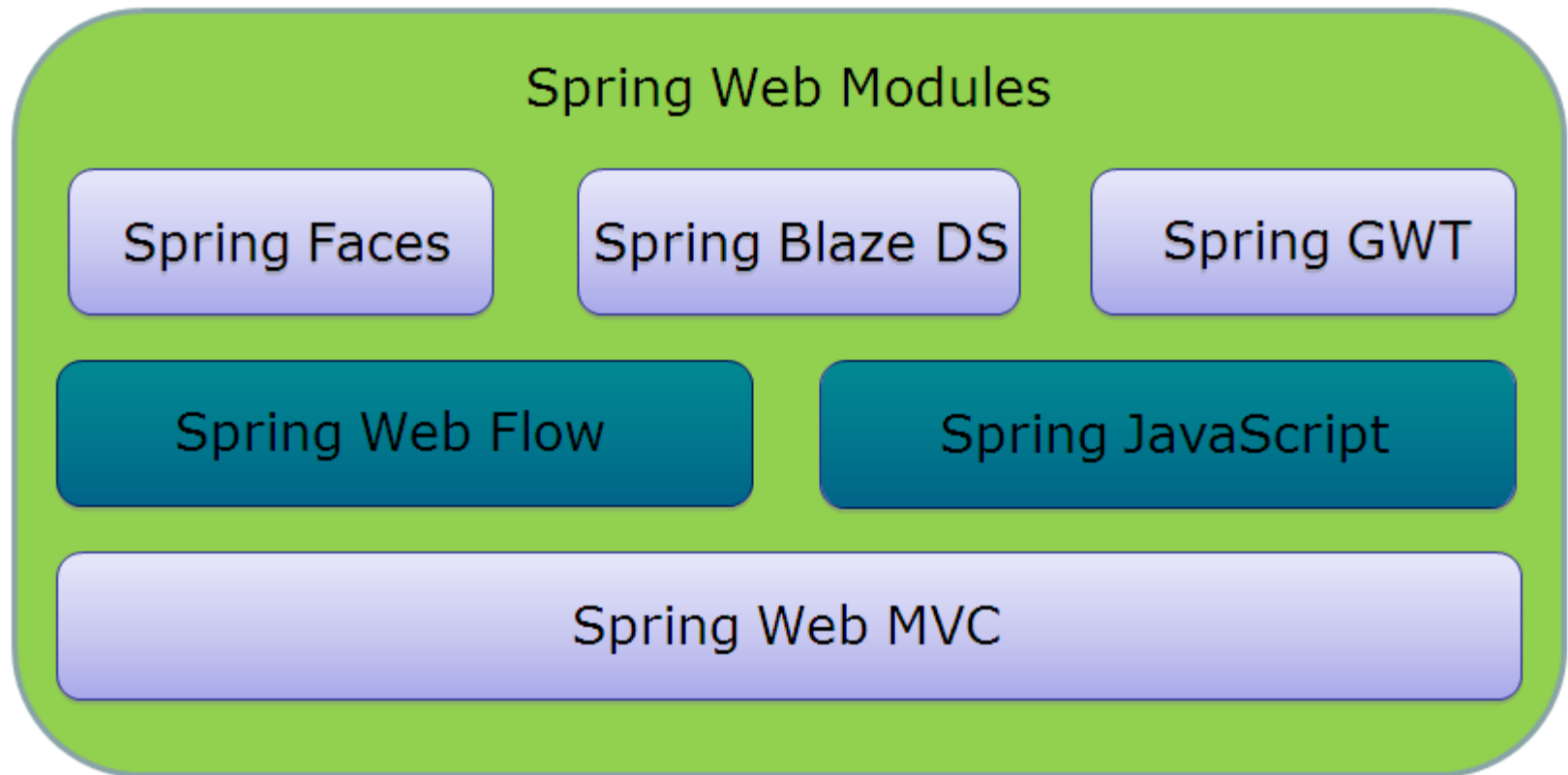


SPRING FRAMEWORK

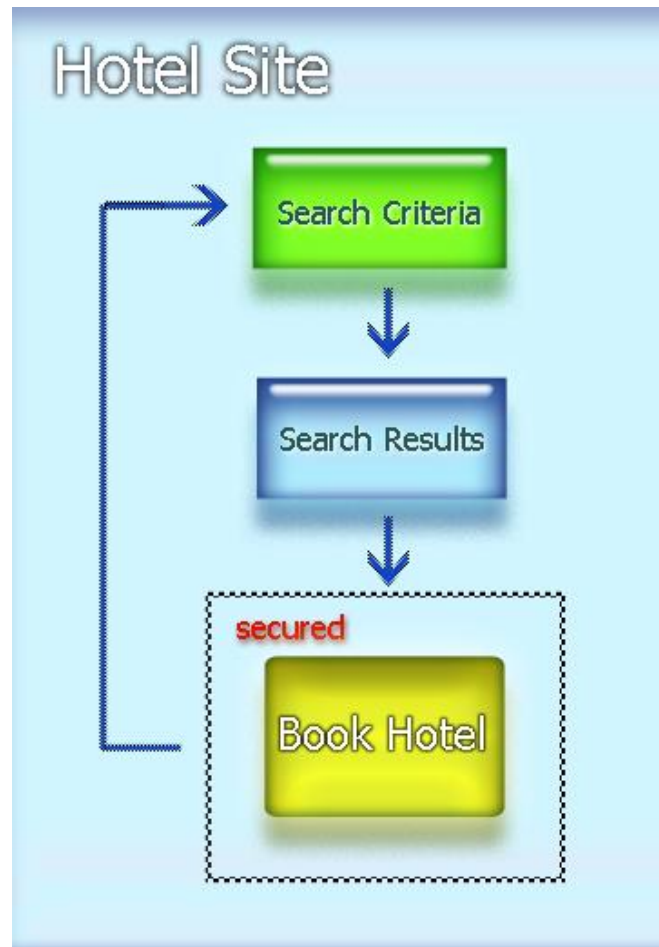
The Spring WEB stack



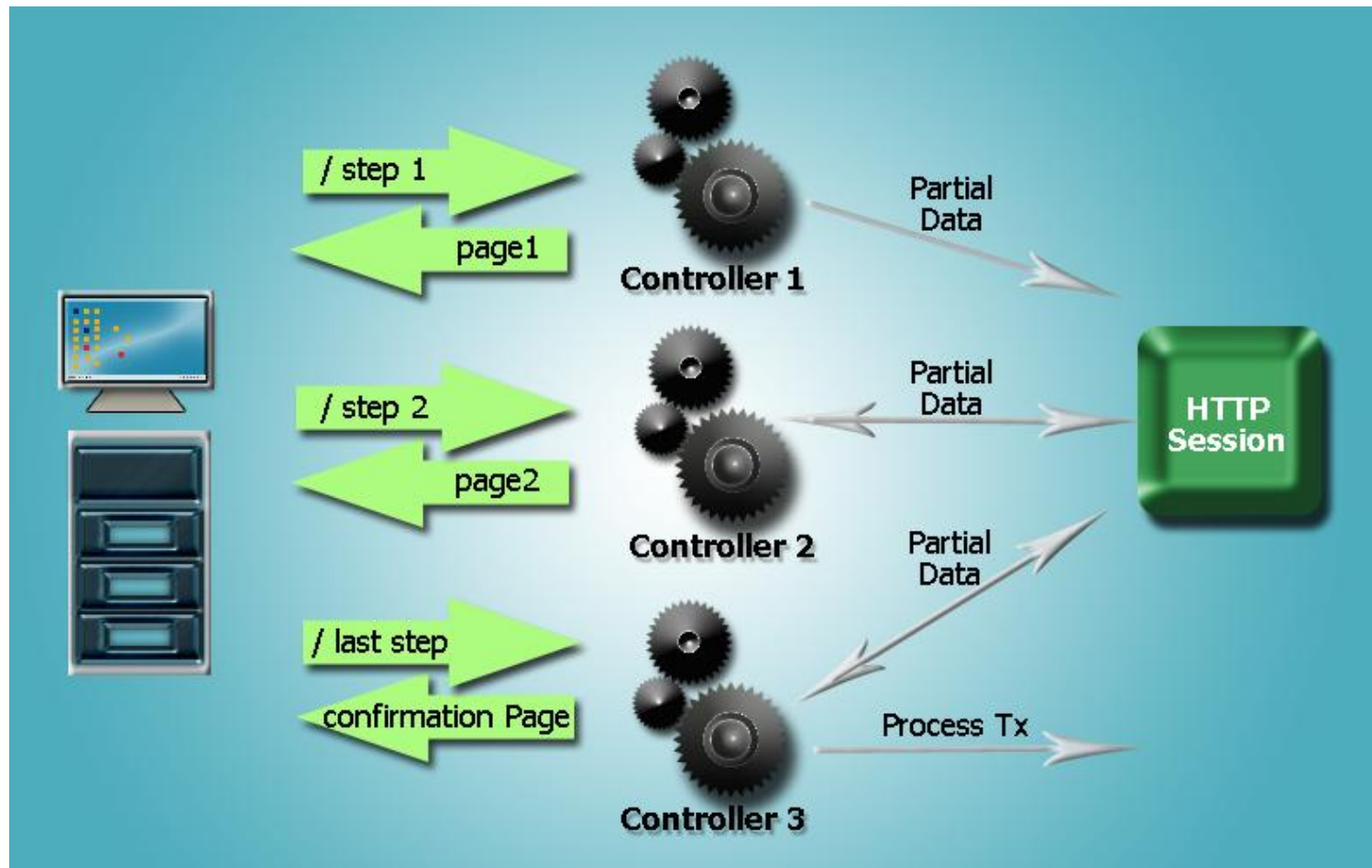
Introduction

- MVC extension
- focused on the definition and execution of page flow
- Integration
 - Spring MVC
 - JSF
 - Portlet

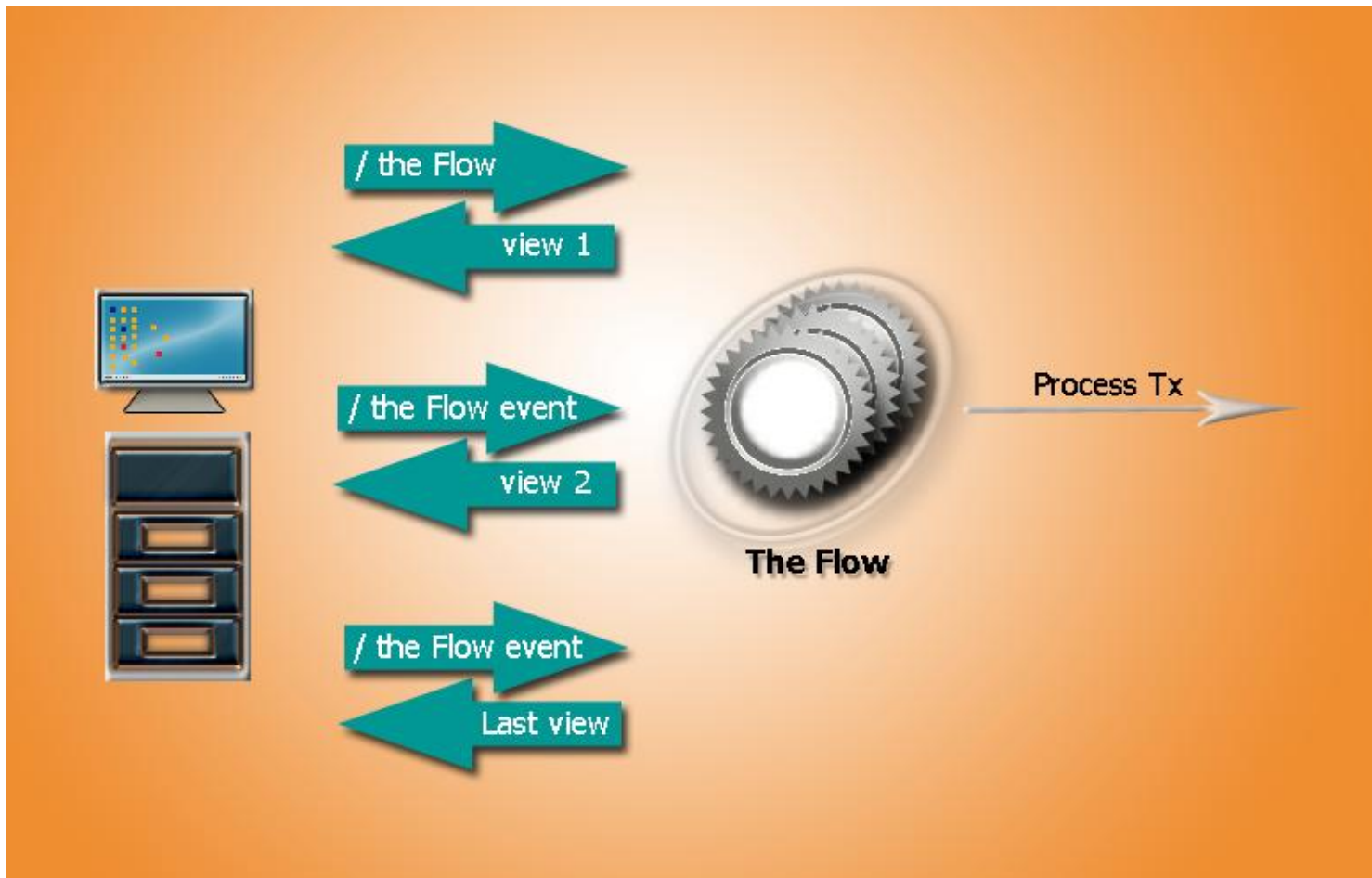
What is flow?



Typical web application



Process driven application



Configuration

Web Flow schema

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:webflow="http://www.springframework.org/schema/webflow-config"
       xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/webflow-config
http://www.springframework.org/schema/webflow-config/spring-webflow-config-2.3.xsd">
```


Flow builder

□ basic

```
<webflow:flow-builder-services
    id="flowBuilderServices"
    view-factory-creator="mvcViewFactoryCreator"
    development="true"
    validator="validator"
    conversion-service="customConversionService"
    expression-parser="ognl"/>
```

Flow registry

□ basic

```
<webflow:flow-registry id="flowRegistry"
    flow-builder-services="builderServices"
    base-path="/WEB-INF"
    parent="parent">

    <!--flow locations-->
</webflow:flow-registry>
```

□ specify path

```
<webflow:flow-location id=""
    path="/hotels/booking/booking-flow.xml"/>
```

□ location pattern

```
<webflow:flow-location-pattern value="/**/*-flow.xml" />
```

Flow executor

□ basic

```
<webflow:flow-executor id="flowExecutor" flow-registry="ref">
    <!--listeners, repositories-->
</webflow:flow-executor>
```

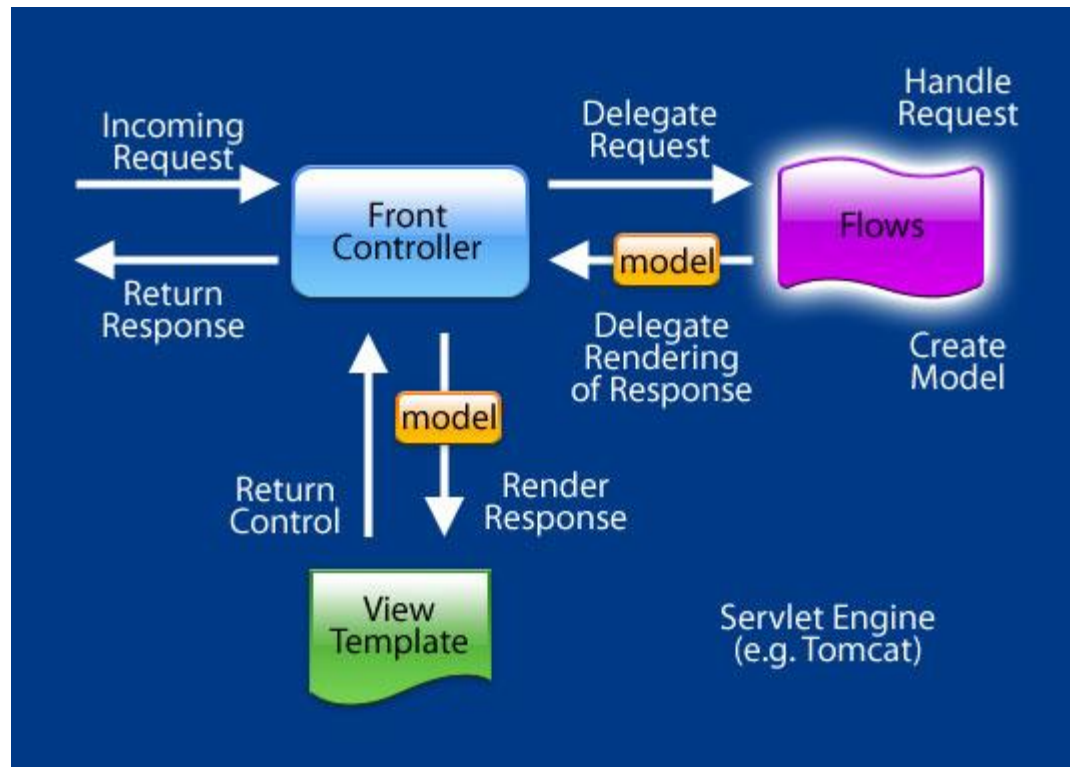
□ listener

```
<webflow:flow-execution-listeners>
    <webflow:listener ref="listener" criteria="*" />
</webflow:flow-execution-listeners>
```

□ repository

```
<webflow:flow-execution-repository max-executions="5"
                                     max-execution-snapshots="10" />
```

Workflow image



Dispatcher servlet

```
<servlet>
  <servlet-name>Spring MVC Dispatcher Servlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value></param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>Spring MVC Dispatcher Servlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

Dispatching

□ FlowHandlerMapping

```
<bean class="org.springframework.webflow.mvc.servlet.FlowHandlerMapping">  
    <property name="order" value="-1"/>  
    <property name="flowRegistry" ref="flowRegistry" />  
</bean>
```

□ FlowHandlerAdapter

```
<bean class="org.springframework.webflow.mvc.servlet.FlowHandlerAdapter">  
    <property name="flowExecutor" ref="flowExecutor"/>  
</bean>
```

Flow

Flow schema

```
<?xml version="1.0" encoding="UTF-8"?>
<flow xmlns="http://www.springframework.org/schema/webflow"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="
          http://www.springframework.org/schema/webflow
          http://www.springframework.org/schema/webflow/spring-webflow-2.0.xsd">

    ...states...

</flow>
```


Essential elements



- ☐ state
- ☐ transition
- ☐ flow data
- ☐ expression language

Input attribute

□ simple

```
<input name="hotelId" required="true" />
```

□ assign input value

```
<input name="hotelId" value="flowScope.bean.itemId" />
```

Output value

□ flow attribute

```
<end-state id="bookingConfirmed">  
    <output name="booking"/>  
</end-state>
```

□ specific

```
<end-state id="bookingConfirmed">  
    <output name="booking" value="booking.id"/>  
</end-state>
```

Data scope

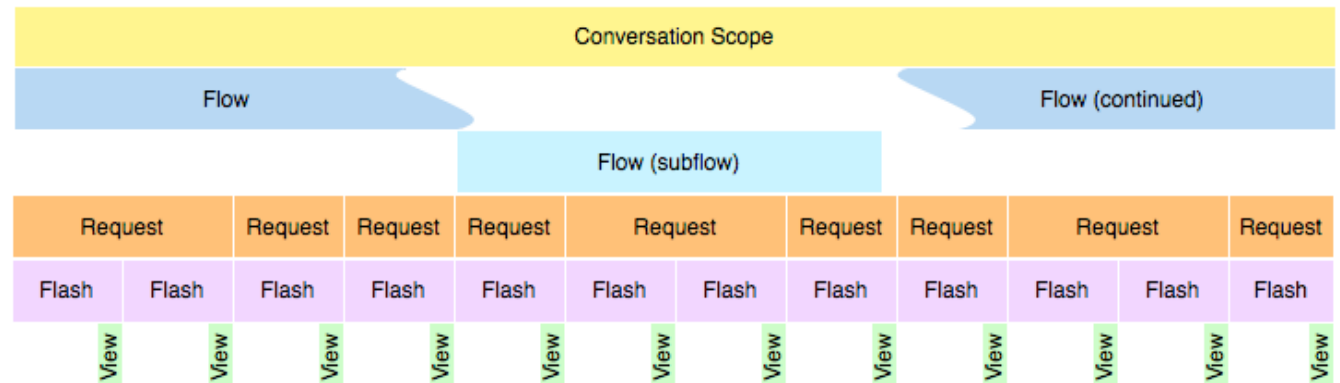
Conversation

Flow

Request

Flash

View



States



- ☐ view
- ☐ action
- ☐ decision
- ☐ subflow
- ☐ end

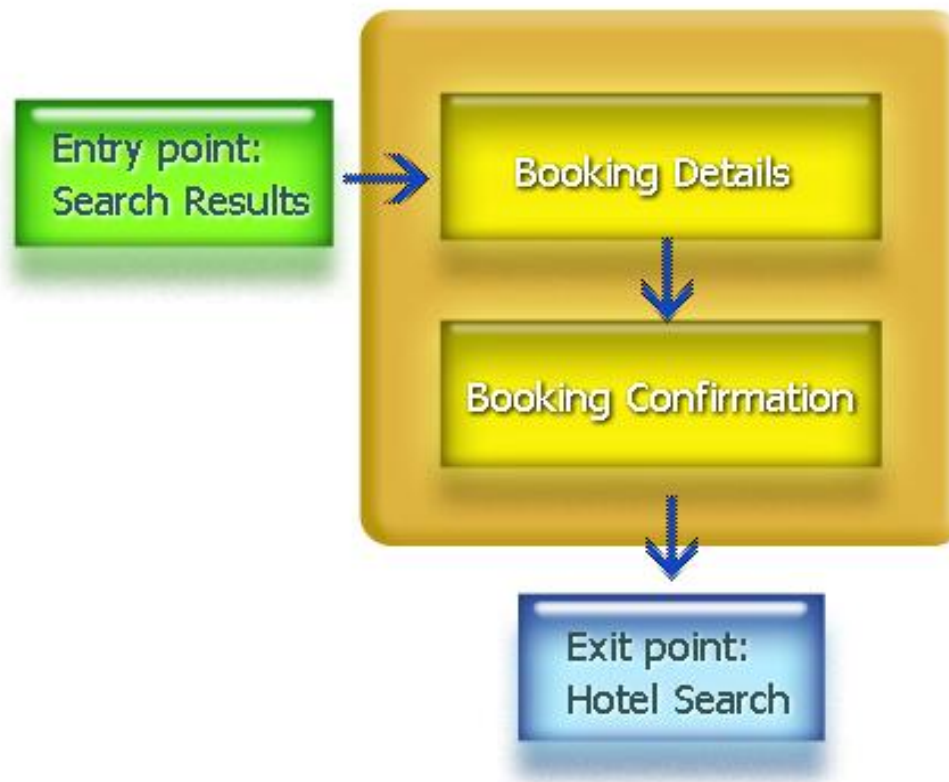
Expression language

- expression types
 - standard
 - template
- implementation
 - Spring EL
 - Unified EL
 - OGNL

Special variables

- ❑ `flowScope`, `viewScope`, etc.
- ❑ `requestParameters`
- ❑ `currentEvent`
- ❑ `currentUser`
- ❑ `messageContext`, `resourceBundle`
- ❑ `flowRequestContext`, `flowExecutionContext`
- ❑ `flowExecutionUrl`
- ❑ `externalContext`

Typical flow



Rendering views

Simple view

□ view state

```
<view-state id="enterBookingDetails" model="booking" view="path">  
  <transition on="proceed" to="reviewBooking" />  
  <transition on="cancel" to="cancel" />  
</view-state>
```

□ view detection

- by state id
- by relative view id
- by absolute view id

View resolution

- default
- mvc view resolver

```
<webflow:flow-builder-services view-factory-creator="mvc" />
```

```
<bean id="mvc"  
      class="org.springframework.webflow.mvc.builder.MvcViewFactoryCreator">  
  <property name="viewResolvers" ref="tilesViewResolver"/>  
  <property name="useSpringBeanBinding" value="true" />  
</bean>
```

Firing events

□ simple link

```
<a href="${flowExecutionUrl}&_eventId=cancel/">Cancel</a>
```

□ hidden field

```
<input type="submit" value="Cancel" />
```

```
<input type="hidden" name="_eventId" value="cancel" />
```

□ submit button

```
<button type="submit" name="_eventId_proceed">Proceed</button>
```

```
<button type="submit" name="_eventId_cancel">Cancel</button>
```

Advanced view

```
<view-state id="enterBookingDetails"
    view=""
    model="booking"
    parent=""
    popup="true/false"
    redirect="true/false" >

    <var name="" class="" />
    <binder>
        <binding property="checkinDate" />
    </binder>
</view-state>
```

Model binding

```
<on-start>
  <evaluate expression="bookingService.createBooking(hotelId) "
    result="flowScope.booking" />
</on-start>

<view-state id="enterBookingDetails" model="booking">
  <binder>
    <binding property="checkinDate" />
    <binding property="checkoutDate" />
    <binding property="creditCard" />
  </binder>
  <transition on="proceed" to="reviewBooking" />
  <transition on="cancel" to="cancel" bind="false" />
</view-state>
```

Validation

□ JSR-303

```
<webflow:flow-builder-services id="" validator="validator" />
<bean id="validator" class="...LocalValidatorFactoryBean"/>
```

□ model validate method

```
public class Booking implements Serializable {
    public void validateEnterBookingDetails(ValidationContext c) {}
}
```

□ separate validator

```
public class BookingValidator {
    public void validate(Booking b, ValidationContext c) {}
    public void validateEnterBookingDetails(...) {}
}
```

Execution actions

Actions

- **set variable**

```
<set name="flowScope.key" value="value"/>
```

- **evaluate business method**

```
<evaluate expression="service.find()" result="viewScope.key"/>
```

- **render view fragment**

```
<render fragments="body" />
```

Events

- flow
 - on-start
 - on-end
- state
 - on-entry
 - on-render (only for view)
 - on-exit
- transition

Example

```
<flow>
  <on-start>
    <evaluate
      expression="bookingService.createBooking(hotelId) "
      result="flowScope.booking" />
    </on-start>
  <view-state id="enterBookingDetails" model="booking">
    <on-render>
      <render fragments="body" />
    </on-render>
    <transition on="proceed" to="reviewBooking" />
    <transition on="cancel" to="cancel" bind="false" />
  </view-state>
</flow>
```

Business actions

□ POJO

```
<evaluate expression="service.businessMethod()" />
```

□ action

```
public class CustomAction implements Action {  
    public Event execute(RequestContext context) {}  
}
```

```
<evaluate expression="customAction" />
```

□ ~~multi-action~~

Action state

□ simple

```
<action-state id="addBooking">
  <evaluate expression="bookingService.addBooking()" />
  <transition to="cancel"/>
</action-state>
```

□ transitioning on evaluation

```
<action-state id="cancelBooking">
  <evaluate expression="bookingService.cancelBooking()" />
  <transition on="cancelled" to="endState"/>
  <transition on="bookingNotFound" to="booking"/>
  <transition on-exception="org.swf.AccessDenied" to="booking"/>
</action-state>
```

Event mapping

Return type	Event identifier
java.lang.String	the String value
java.lang.Boolean	yes / no
java.lang.Enum	the enum name
any other type	success

Named actions

```
<action-state id="callTwoMethods">
  <evaluate expression="service.methodOne()">
    <attribute name="name" value="methodOne" />
  </evaluate>
  <evaluate expression="service.methodTwo()">
    <attribute name="name" value="methodTwo" />
  </evaluate>

  <transition on="methodTwo.success" to="showResults" />
</action-state>
```

Decision

```
<decision-state id="hasItems">
  <if test="service.hasItems()"
    then="checkItemsState"
    else="end" />
</decision-state>
```


Subflow

```
<subflow-state id="addAccount" subflow="createAccount">
  <transition on="accountCreated" to="reviewBooking">
    <evaluate expression="bookingService.add(...)" />
  </transition>
  <transition on="creationCancelled" to="end" />
</subflow-state>
```

Transition

Advanced

```
<transition on="cancel"  
    on-exception=""  
    to="cancel"  
    bind="true"  
    validate="true"  
    history="preserve/discard/invalidate"/>
```

Global transition



```
<global-transitions>  
  <transition on="cancel" to="cancel" />  
</global-transitions>
```



Finish

End state

□ simple

```
<end-state id="cancel" />
```

□ custom view

```
<end-state id="success"  
    view="externalRedirect:controller?result=success"/>
```

□ output

```
<end-state id="bookingConfirmed">  
    <output name="bookingId" value="booking.id" />  
</end-state>
```

Redirects

- ❑ `servletRelative`
- ❑ `contextRelative`
- ❑ `serverRelative`
- ❑ `http://` or `https://`

FlowHandler

□ Interface

```
public interface FlowHandler {  
    public String getFlowId();  
    public MutableAttributeMap createExecutionInputMap (...);  
    public String handleExecutionOutcome (...);  
    public String handleException (...);  
}
```

□ Bean

```
<bean name="hotels/booking" class="test.BookingFlowHandler"/>
```


Spring JS

Configuration

□ server side

```
<mvc:resources mapping="/resources/**"
               location="/, classpath:/META-INF/web-resources/" />
<mvc:default-servlet-handler />
```

□ client side

```
<script type="text/javascript"
src="<c:url value="/resources/dojo/dojo.js" />"></script>
<script type="text/javascript"
src="<c:url value="/resources/spring/Spring.js" />"></script>
<script type="text/javascript"
src="<c:url value="/resources/spring/Spring-Dojo.js" />"></script>
```

Simple decoration

```
<form:input id="searchString" path="searchString"/>
<script type="text/javascript">
    Spring.addDecoration(new Spring.ElementDecoration({
        elementId : "searchString",
        widgetType : "dijit.form.ValidationTextBox",
        widgetAttrs : {promptMessage : "Search hotels by name."}
    }));
</script>
```

Regex

```
<form:input path="creditCard"/>
<script type="text/javascript">
    Spring.addDecoration(new Spring.ElementDecoration({
        elementId : "creditCard",
        widgetType : "dijit.form.ValidationTextBox",
        widgetAttrs : {
            required : true,
            invalidMessage : "A 16-digit card number is required.",
            regExp : "[0-9]{16}"
        }
    }));
</script>
```

Date

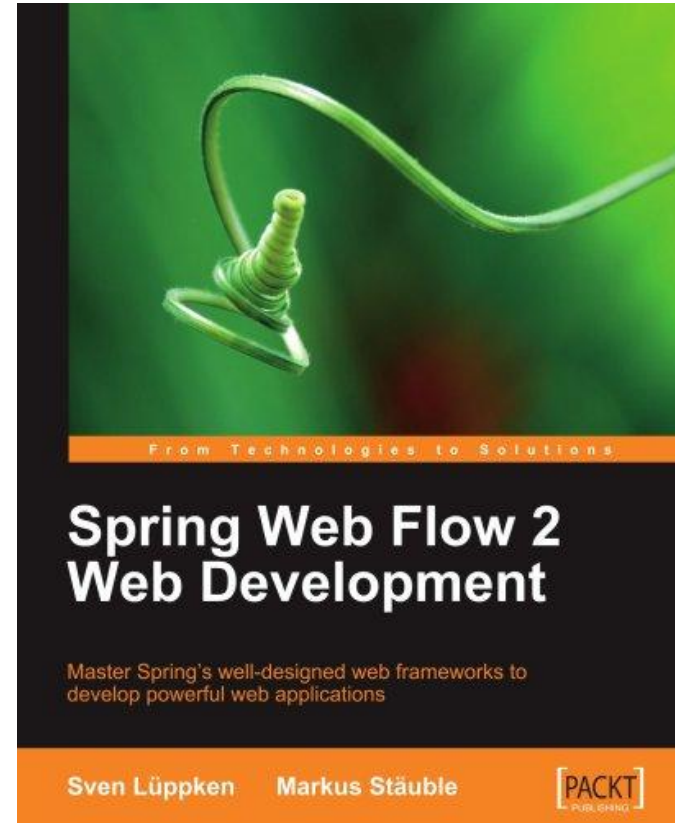
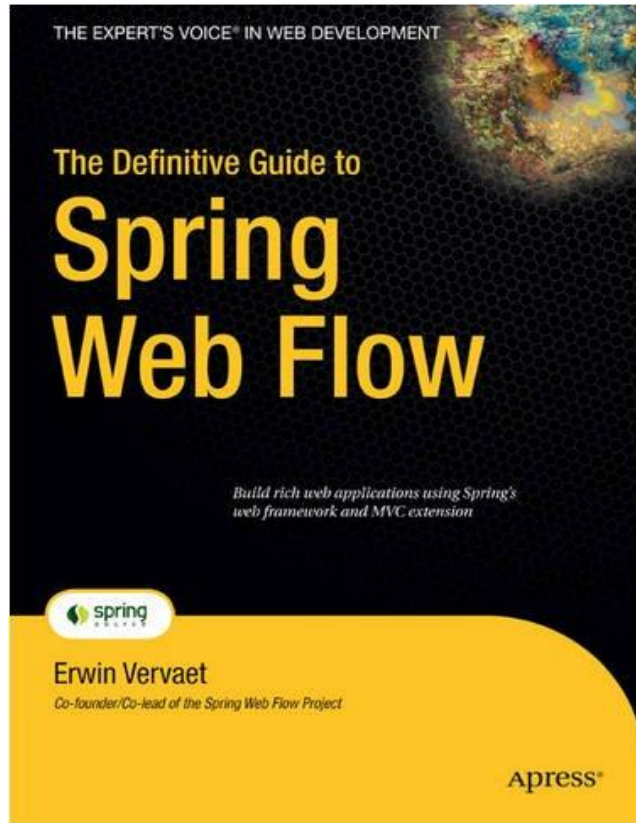
```
<form:input path="checkoutDate"/>
<script type="text/javascript">
    Spring.addDecoration(new Spring.ElementDecoration({
        elementId : "checkoutDate",
        widgetType : "dijit.form.DateTextBox",
        widgetAttrs : {
            datePattern : "MM-dd-yyyy",
            required : true
        }
    }));
</script>
```

Conclusion



- ❑ handles all navigation steps (including back, refresh)
- ❑ clear state management
- ❑ full integration with Spring MVC and other

Books



Links

- Web Flow reference

<http://static.springframework.org/spring-webflow/docs/2.3.x/reference/html/index.html>

- samples

<https://src.springframework.org/svn/spring-samples/>

- forum

<http://forum.springframework.org/forumdisplay.php?f=36>

- Spring Web Flow 2 Web Development

<http://www.springsource.org/node/1286>

- Refcardz

<http://refcardz.dzone.com/refcardz/spring-web-flow>

Questions



The end



noskov.d@gmail.com



<http://www.linkedin.com/in/noskovd>



<http://www.slideshare.net/analizator/presentations>