RIA with ZK

Boost your productivity

Invited Talk

Java User Group Switzerland

Daniel Seiler
AdNovum Informatik AG
Submission Id: 7960

AGENDA

> Introduction
> ZK basics
> Lets build an application
> Advanced topics
> Custom components
> Some more examples
> ZK and the others
> What's coming next?
> Summary
> Q & A

Goals

Infect you with the ZK virus

You are able to explain the position of ZK in the current RIA Landscape

You know the main features, concepts and principles of ZK

The problem to solve

To build rich, interactive, fast and scalable, distributed business applications ...

... we need a framework and technology that ...

... maximizes our productivity by abstracting and hiding much of the complexity

... provides a rich set of prebuilt components and features

... is easy to extend

The big picture

- **Local offline tools** (Standalone, not distributed, ...)
  - Office tools
- **"Fat Client" Distributed applications** (Corba, RMI, WS , ...)
  - Eclipse RCP
- **Rich** (Async. update, sorting, drag & drop)
  I nternet (Communication with Webserver)
  Application (User interactions, data storage)
  - Runs in an external runtime environment (plugin or standalone)
    - Java FX
    - Flex (flash)
    - Laszlo (flash)
    - Curl
    - Captain Casa (Swing, JSF)
  - Runs directly in a browser (No plugin, Ajax)
    - Javascript libraries
      - jQuery
      - Prototype
      - Script.aculo.us
      - DWR
    - Frameworks
      - Echo2
      - GWT
      - ICE Faces
      - IT Mill → vaadin
      - ZK ←
- **Trad. web applications** (Page reloading, simple controls)

What is ZK?

From Wikipedia: http://en.wikipedia.org/wiki/ZK_Framework

ZK is an **open-source Ajax web application framework**, written in Java, that enables creation of rich graphical user interfaces for web applications with no JavaScript and little programming knowledge.

The core of ZK consists of an **Ajax-based event-driven** mechanism, over **123 XUL** and **83 XHTML-based components**, and a markup language for designing user interfaces. Programmers design their application pages in feature-rich XUL/XHTML components, and manipulate them upon events triggered by end user's activity. It is similar to the programming model found in desktop GUI-based applications.

ZK takes the so called **server-centric** approach where the content synchronization of components and the event pipelining between clients and servers are automatically done by the engine and Ajax plumbing codes are completely transparent to web application developers.

Selling points (see http://www.zkoss.org/product/zk.dsp)

Open Source: ZK is the leading open source Ajax + Mobile framework. ZK developer community is extremely active with 20+ translations, 100+ articles/blogs, and 100,000+ lines of codes, 1,000,000+ downloads, from 190+ countries.

Rich User Experience: 200+ off-the-shelf state-of-art XUL/HTML-compliant Ajax components. Numerous third party widgets: JFreeChart, JasperReports, Google Maps, FCKeditor, Timeline, Timeplot, ExtJS, Dojo and so on

Server push support: Comet and Client polling

Standards-based: ZK is a standard-compliant solution.

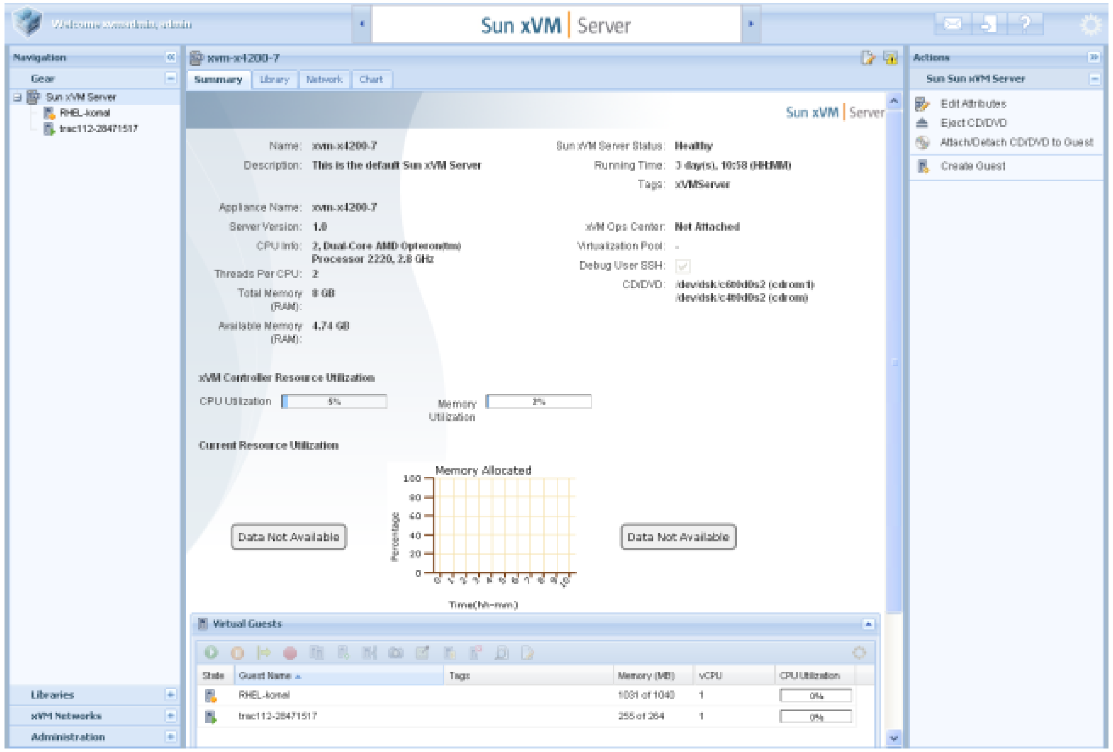
Extensibility and Customizability: ZK is fully customizable and extensible with modular and plug-and-play architecture.

Mobile Access: ZK extends the reach of enterprise Internet applications to 1+ billion mobile devices at minimal cost. ZK supports Java Mobile, Android, and various mobile browsers.

Security: ZK is designed from the ground up to be secure.

Selling points 2/2 (see http://www.openxvm.org/xvmsui.html)

ZK is used in production by many companies including some Fortune 500 corporations like Barclays, Swiss Re, Alcatel-Lucent, Toyate, etc.
Even Sun is using ZK (not JSF) for their management app of the xVM Server:



Name: xvm-x4200-7
Description: This is the default Sun xVM Server
Sun xVM Server Status: Healthy
Running Time: 3 day(s), 10:58 (HH:MM)
Tags: xVMServer
Appliance Name: xvm-x4200-7
Server Version: 1.0
xVM Ops Center: Not Attached
CPU Info: 2, Dual-Core AMD Opteron(tm) Processor 2220, 2.8 GHz
Virtualization Pool: -
Threads Per CPU: 2
Debug User SSH: ✓
Total Memory (RAM): 8 GB
CD/DVD: /dev/dsk/c6t0d0s2 (cdrom1) /dev/dsk/c4t0d0s2 (cdrom)
Available Memory (RAM): 4.74 GB

xVM Controller Resource Utilization

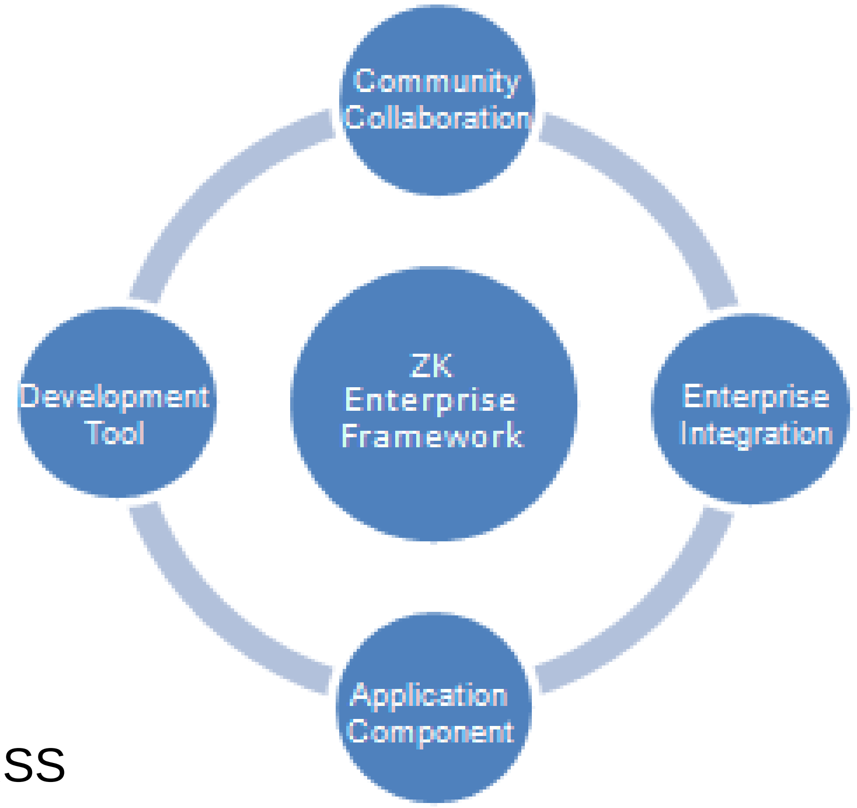
CPU Utilization 5%
Memory Utilization 2%

Current Resource Utilization

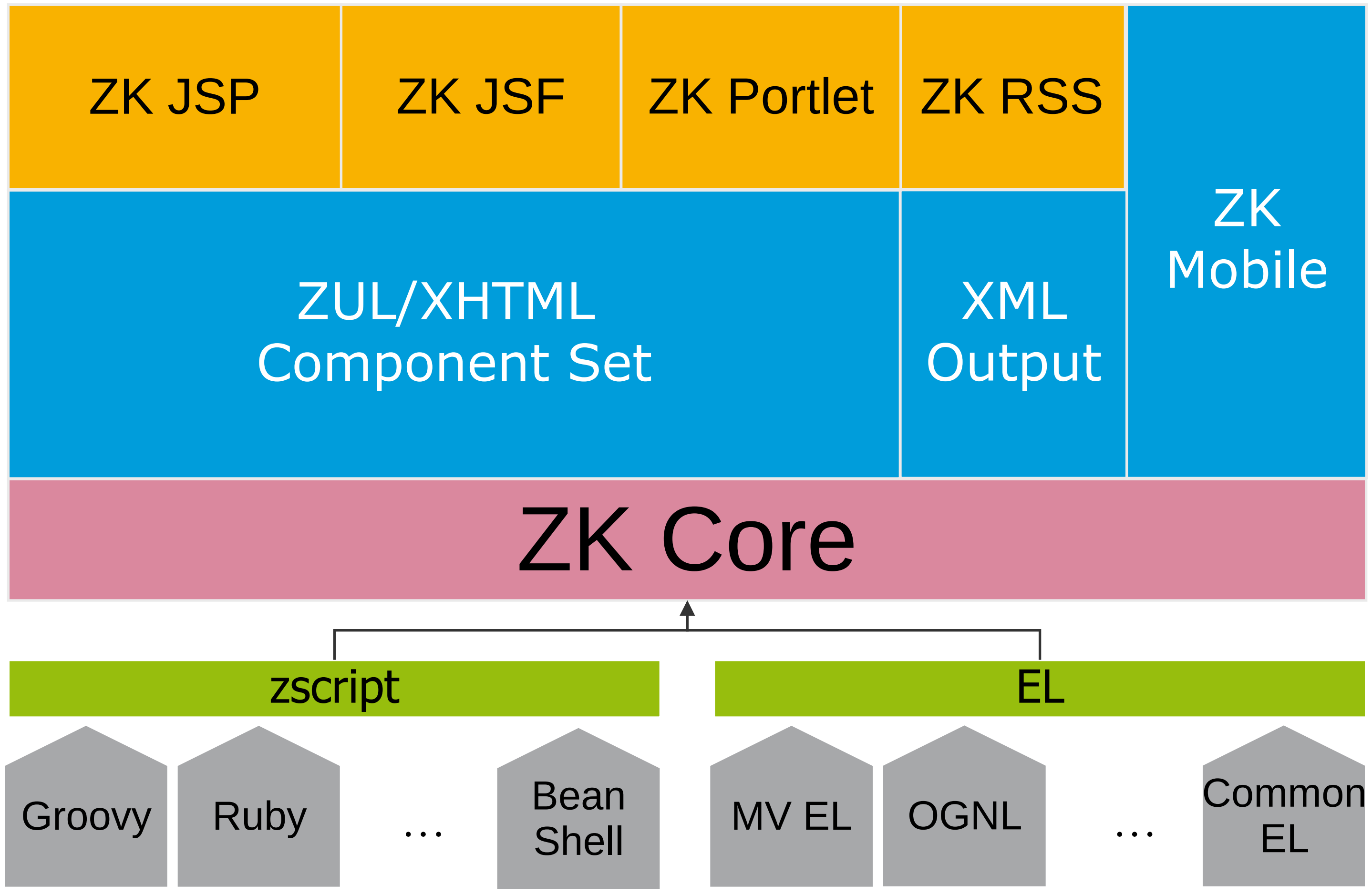
State	Guest Name	Tags	Memory (MB)	vCPU	CPU Utilization
	RHEL-konal		1031 of 1040	1	0%
	trac112-28471517		255 of 264	1	0%

ZK Products

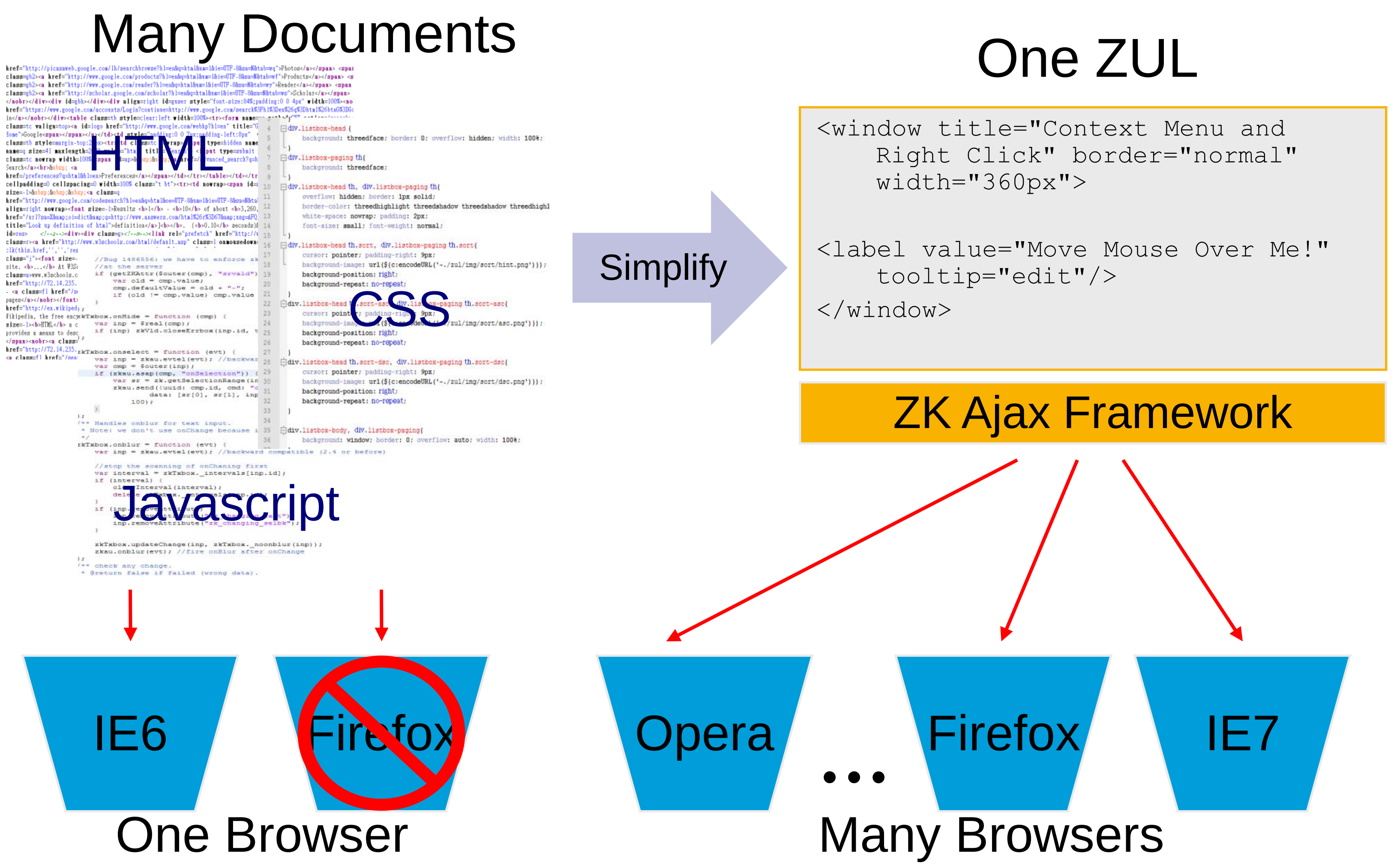
- Framework
  - ZK Framework
  - ZK Mobile
- Application component
  - ZK Spreadsheet, ZK Calendar
- Development
  - ZK Studio, ZK Jet (Firefox plugin)
- Enterprise Integration
  - Spring, EJB, ZK JSP, JSF, Portlet, RSS
- ZK Forge
  - Integrating third party widgets, such as ZK Gmaps, ZK FCKeditor, ZK Timeline, ZK Ext JS, ZK Dojo and so on



ZK overall picture



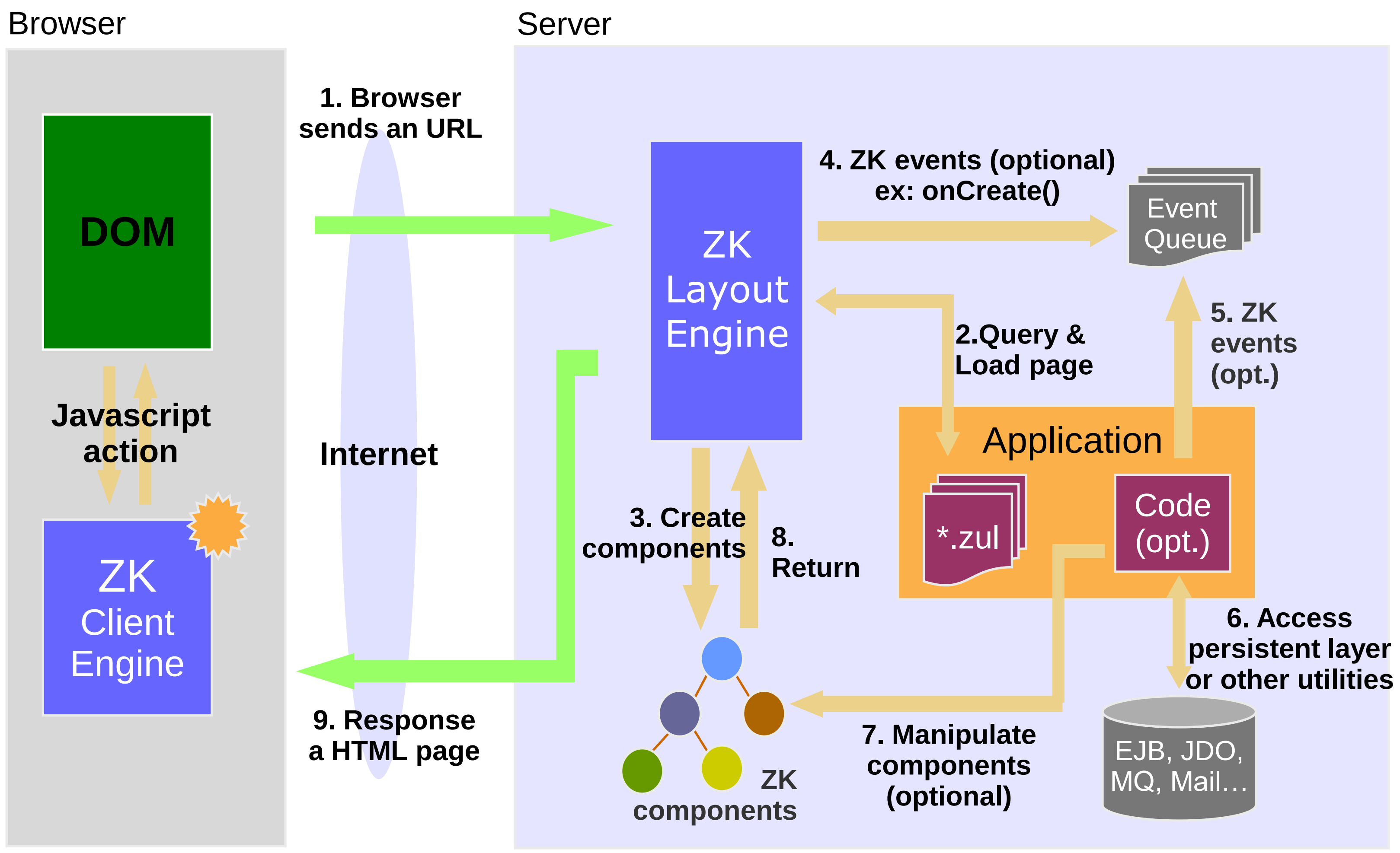
ZK's Ajax solution



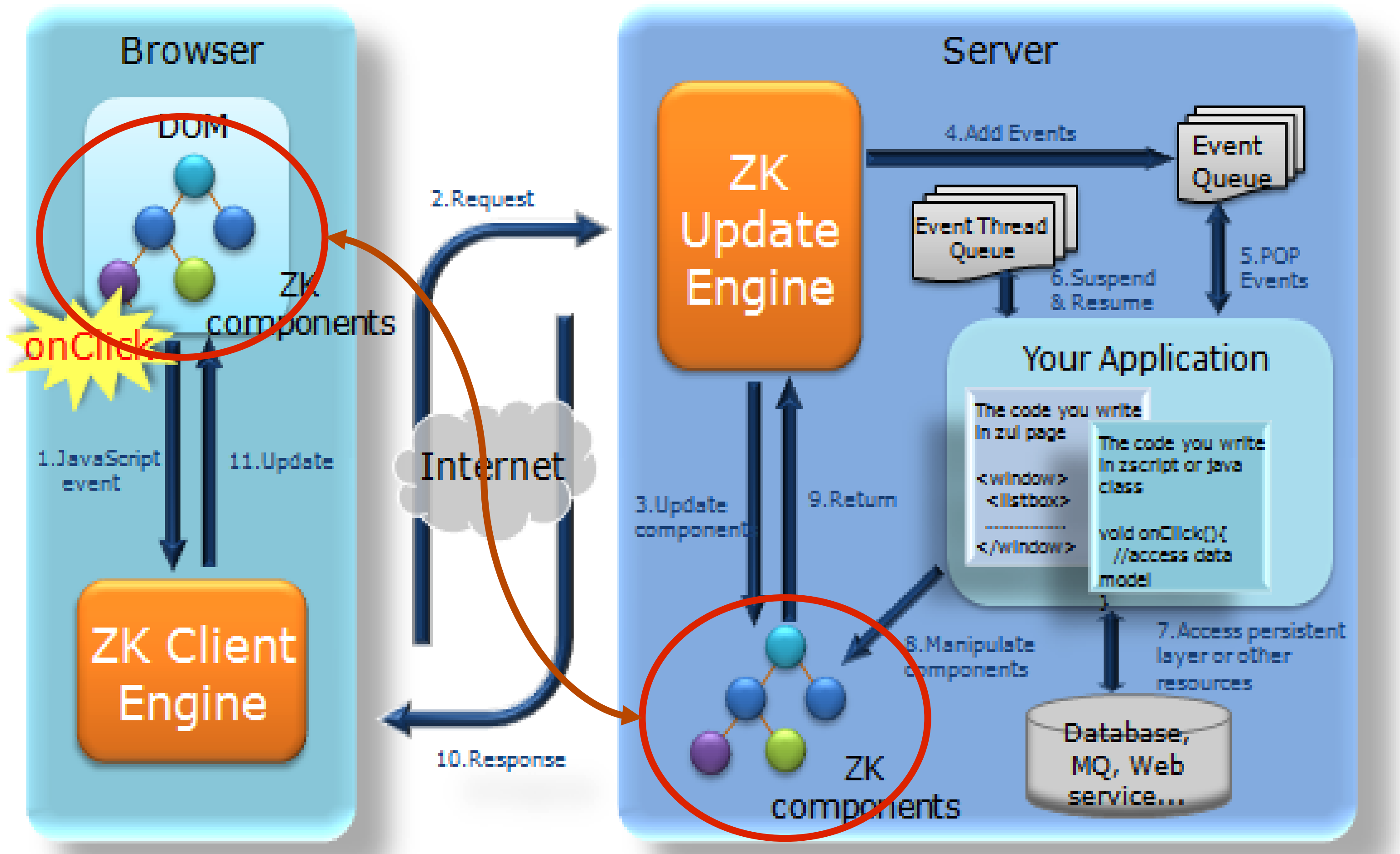
```
<window title="Context Menu and
    Right Click" border="normal"
    width="360px">

<label value="Move Mouse Over Me!"
    tooltip="edit"/>
</window>
```

ZK Generating an HTML page



ZK doing Ajax

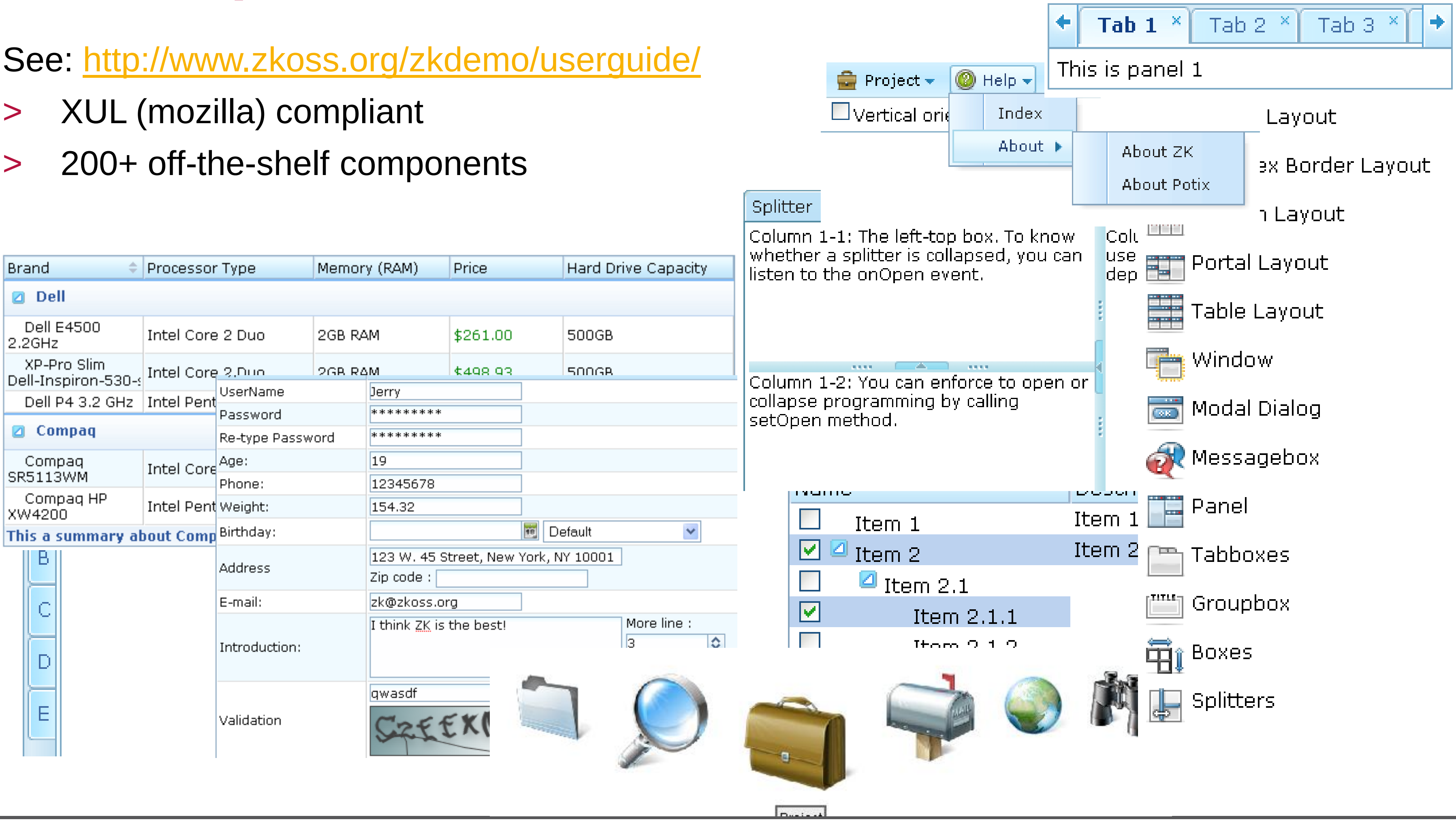


ZK Components

See: http://www.zkoss.org/zkdemo/userguide/

> XUL (mozilla) compliant

> 200+ off-the-shelf components



Tab 1 × Tab 2 × Tab 3 ×

This is panel 1

Project ▾ Help ▾

Vertical ori

Index

About ▸

About ZK

About Potix

Layout

ex Border Layout

n Layout

Portal Layout

Table Layout

Window

Modal Dialog

Messagebox

Panel

Tabboxes

Groupbox

Boxes

Splitters

Splitter

Column 1-1: The left-top box. To know whether a splitter is collapsed, you can listen to the onOpen event.

Colu use dep

Column 1-2: You can enforce to open or collapse programming by calling setOpen method.

Brand	Processor Type	Memory (RAM)	Price	Hard Drive Capacity
Dell				
Dell E4500 2.2GHz	Intel Core 2 Duo	2GB RAM	$261.00	500GB
XP-Pro Slim Dell-Inspiron-530-s	Intel Core 2 Duo	2GB RAM	$498.93	500GB
Dell P4 3.2 GHz	Intel Pent			
Compaq				
Compaq SR5113WM	Intel Core			
Compaq HP XW4200	Intel Pent			
This a summary about Comp				

B C D E

UserName	Jerry
Password	*********
Re-type Password	*********
Age:	19
Phone:	12345678
Weight:	154.32
Birthday:	Default
Address	123 W. 45 Street, New York, NY 10001 Zip code :
E-mail:	zk@zkoss.org
Introduction:	I think ZK is the best! More line : 3
Validation	qwasdf

	Name	Descr
☐	Item 1	Item 1
✓	Item 2	Item 2
☐	Item 2.1	
✓	Item 2.1.1	
☐	Item 2.1.2	

JAZOON09 THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY JUNE 22–25, 2009 ZURICH

AdNovum

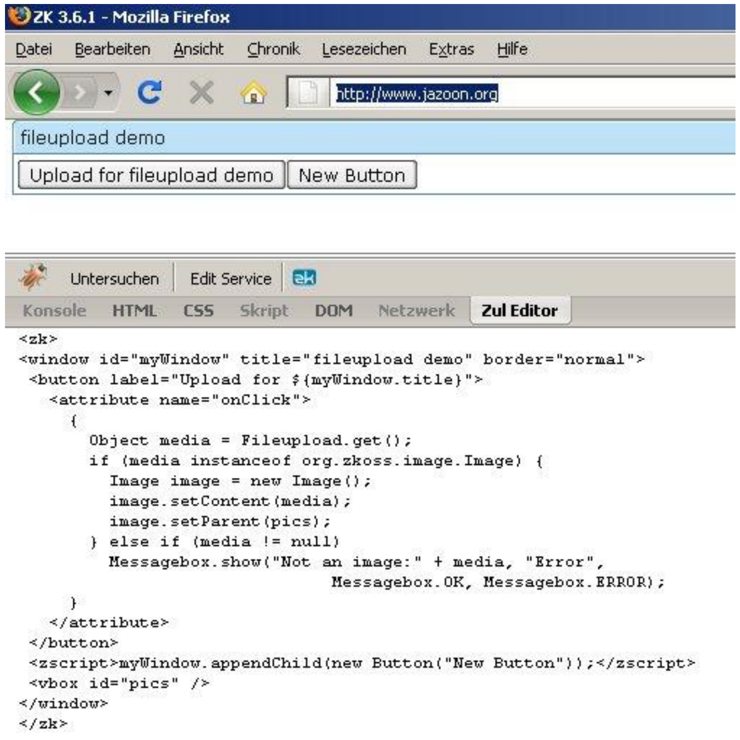
netcetera

Sun microsystems

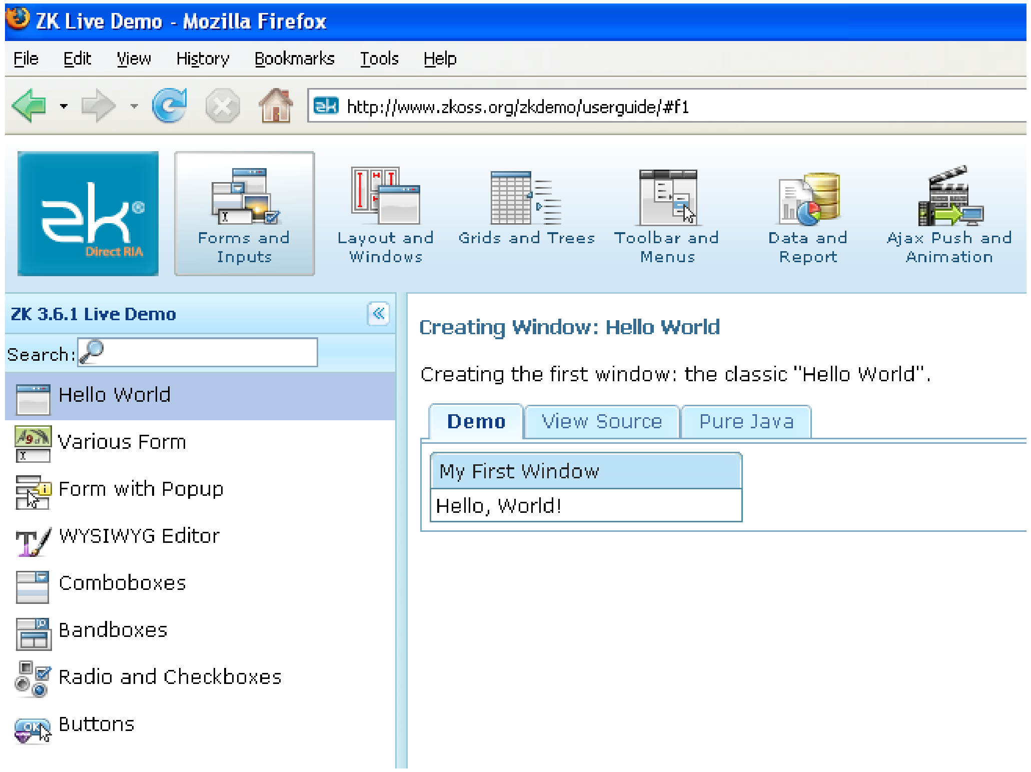
Fast prototyping

ZK Explorer: http://www.zkoss.org/zkdemo/userguide/

ZK Jet, Firefox plugin: https://addons.mozilla.org/en-US/firefox/addon/9626


ZK 3.6.1 - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://www.jazoon.org
fileupload demo
Upload for fileupload demo
New Button
Untersuchen Edit Service
Konsole HTML CSS Skript DOM Netzwerk Zul Editor

```
<zk>
<window id="myWindow" title="fileupload demo" border="normal">
 <button label="Upload for ${myWindow.title}">
   <attribute name="onClick">
     {
       Object media = Fileupload.get();
       if (media instanceof org.zkoss.image.Image) {
         Image image = new Image();
         image.setContent(media);
         image.setParent(pics);
       } else if (media != null)
         Messagebox.show("Not an image:" + media, "Error",
                                   Messagebox.OK, Messagebox.ERROR);
     }
   </attribute>
 </button>
 <zscript>myWindow.appendChild(new Button("New Button"));</zscript>
 <vbox id="pics" />
</window>
</zk>
```



Creating Window: Hello World

Creating the first window: the classic "Hello World".

ZUL, EL and zscript

```
<window id="myWindow" title="fileupload demo" border="normal">
  <button label="Upload for ${myWindow.title}">
    <attribute name="onClick">
      {
        Object media = Fileupload.get();
        if (media instanceof org.zkoss.image.Image) {
          Image image = new Image();
          image.setContent(media);
          image.setParent(pics);
        } else if (media != null)
          Messagebox.show("Not an image:" + media, "Error",
                              Messagebox.OK, Messagebox.ERROR);
      }
    </attribute>
  </button>
  <zscript>myWindow.appendChild(new Button("New Button"));</zscript>
  <vbox id="pics" />
</window>
```

if, unless and forEach

```
<window>
  <zscript>
    newBtn = true;
    contacts = new String[] {"Monday", "Tuesday","Wednesday"};
  </zscript>
  <button label="New" if="${newBtn}">
    <attribute name="onClick">
      alert("I am a new Button!");
    </attribute>
  </button>
  <button label="Old" unless="${newBtn}" />
  <separator />
  <listbox width="100px">
    <listitem label="${each}" forEach="${contacts}" />
  </listbox>
</window>
```

Try it out on the ZK components Demo page ...

The life cycle of loading pages

1) The page initial phase

2) The component creation phase

3) The event processing phase

4) The rendering phase

The page initial phase

1. The page initial phase
2. The component creation phase
3. The event processing phase
4. The rendering phase

init processing instruction gets processes (if defined) by calling the doInit method of a class or by calling a zscript.
In this phase the page is not yet attached to the desktop!

```
<?xml version="1.0" encoding="UTF-8"?>
<?page id="userGuide" title="ZK Live Demo"?>
<?init class="MyInit"?> or <?init zscript=“myinit.zs“?>
<zk>
  <window id="win" border="normal" width="200px"
   sizable="true">
    <caption image="/img/inet.png" label="Hi there!"/>
    <checkbox label="Hello, Wolrd!"/>
    <button label="center"
          onCreate="win.setBtn(self)"/>
  </window>
</zk>
```

The component creation phase

1. The page initial phase
2. The component creation phase
3. The event processing phase
4. The rendering phase

In this phase, ZK loader interprets a ZUML page. It creates and initializes components accordingly. It takes several steps as described on the next slide...

```
<?xml version="1.0" encoding="UTF-8"?>
<?page id="userGuide" title="ZK Live Demo"?>
<?init class="MyInit"?> or <?init zscript="myinit.zs"?>
<zk>
  <window id="win" border="normal" width="200px"
   sizable="true">
    <caption image="/img/inet.png" label="Hi there!"/>
    <checkbox label="Hello, Wolrd!"/>
    <button label="center"
            onCreate="win.setBtn(self)"/>
  </window>
</zk>
```

The component creation phase

1) For each element, it examines the **if and unless attribute** to decide whether it is effective. If not, the element and all of its child elements are ignored.

2) If the **forEach attribute** is specified with a collection of items, ZK repeats the following steps for each item in the collection.

3) **Creates the component** based on the element name, or by use of the class specified in the use attribute, if any.

4) **Initializes the members** one-by-one based on the order in which attributes are specified on the ZUML page.

5) **Interprets the nested elements** and repeats the whole procedure.

6) **Invokes the afterCompose method** if the component implements the org.zkoss.zk.ui.ext.AfterCompose interface.

7) After all children are created, the **onCreate event is sent** to this component. Notice that the onCreate events are posted for child components first.

Note: a developer can perform some application-specific initialization by listening to the onCreate event or implementing AfterCompose. AfterCompose is called in the Component Creation Phase, while the onCreate event is handled by an event listener.

The event processing phase

1. The page initial phase
2. The component creation phase
3. The event processing phase
4. The rendering phase

In this phase, ZK invokes each listener for each event queued for this desktop one-by-one.
An independent thread is started to invoke each listener, so it could be suspended without affecting the processing of other events.
During the processing, an event listener might fire other events.

```
<?xml version="1.0" encoding="UTF-8"?>
<?page id="userGuide" title="ZK Live Demo"?>
<?init class="MyInit"?> or <?init zscript="myinit.zs"?>
<zk>
  <window id="win" border="normal" width="200px"
   sizable="true">
    <caption image="/img/inet.png" label="Hi there!"/>
    <checkbox label="Hello, Wolrd!"/>
    <button label="center"
          onCreate="win.setBtn(self)"/>
  </window>
</zk>
```

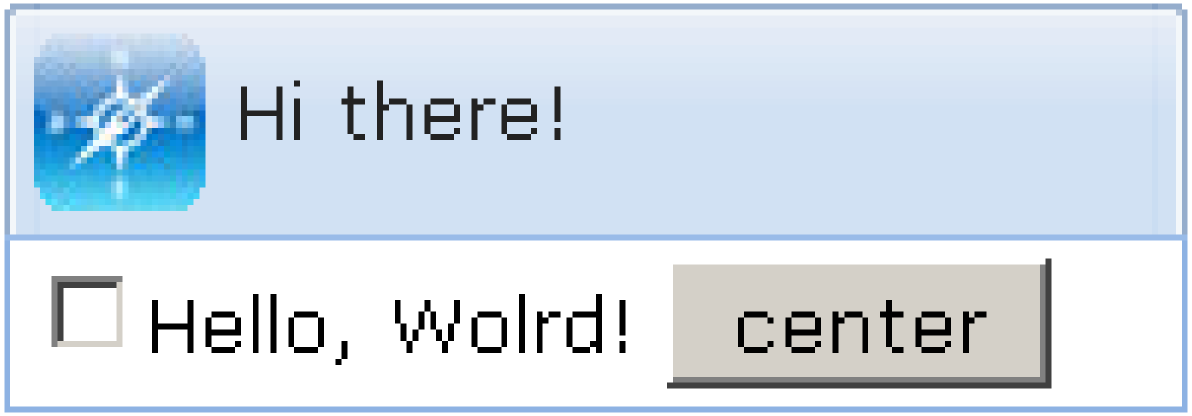
The rendering phase

1. The page initial phase
2. The component creation phase
3. The event processing phase
4. The rendering phase

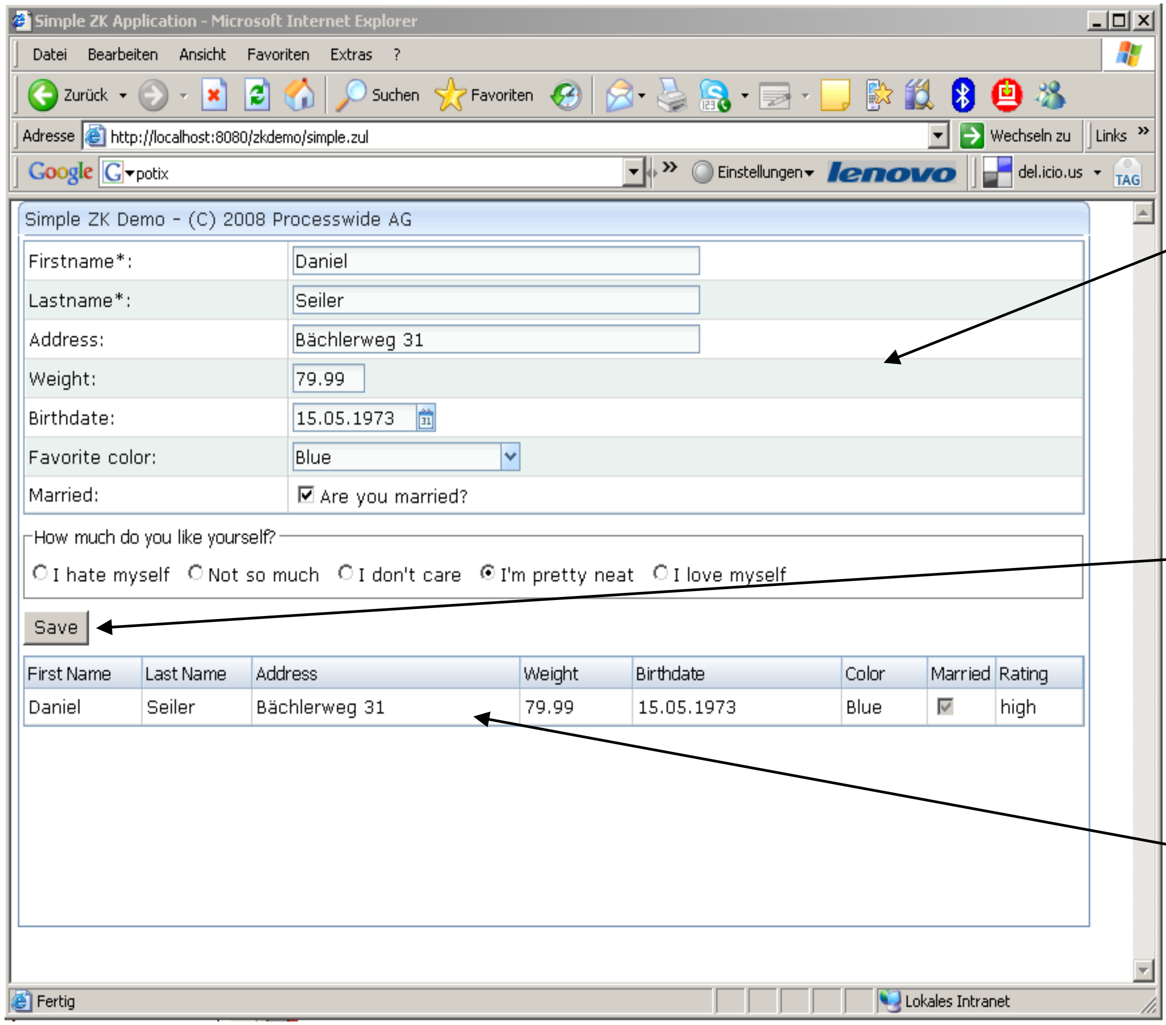
After all events are processed, ZK renders these components into a regular HTML page and sends this page to the browser.
To render a component, the redraw method is called. The implementation of a component shall not alter any content of the component in this method.

```
<?xml version="1.0" encoding="UTF-8"?>
<?page id="userGuide" title="ZK Live Demo"?>
<?init clas
<zk>
   <wind
    sizab
      <ca
      <ch
      <bu

   </window>
</zk>
```



First application - HelloWorldX


Simple ZK Application - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Adresse
http://localhost:8080/zkdemo/simple.zul
Wechseln zu
Links
Google
potix
Einstellungen
lenovo
del.icio.us
TAG
Fertig
Lokales Intranet

Simple ZK Demo - (C) 2008 Processwide AG

Firstname*:	Daniel
Lastname*:	Seiler
Address:	Bächlerweg 31
Weight:	79.99
Birthdate:	15.05.1973
Favorite color:	Blue
Married:	☑ Are you married?

How much do you like yourself?

○ I hate myself ○ Not so much ○ I don't care ● I'm pretty neat ○ I love myself

Save

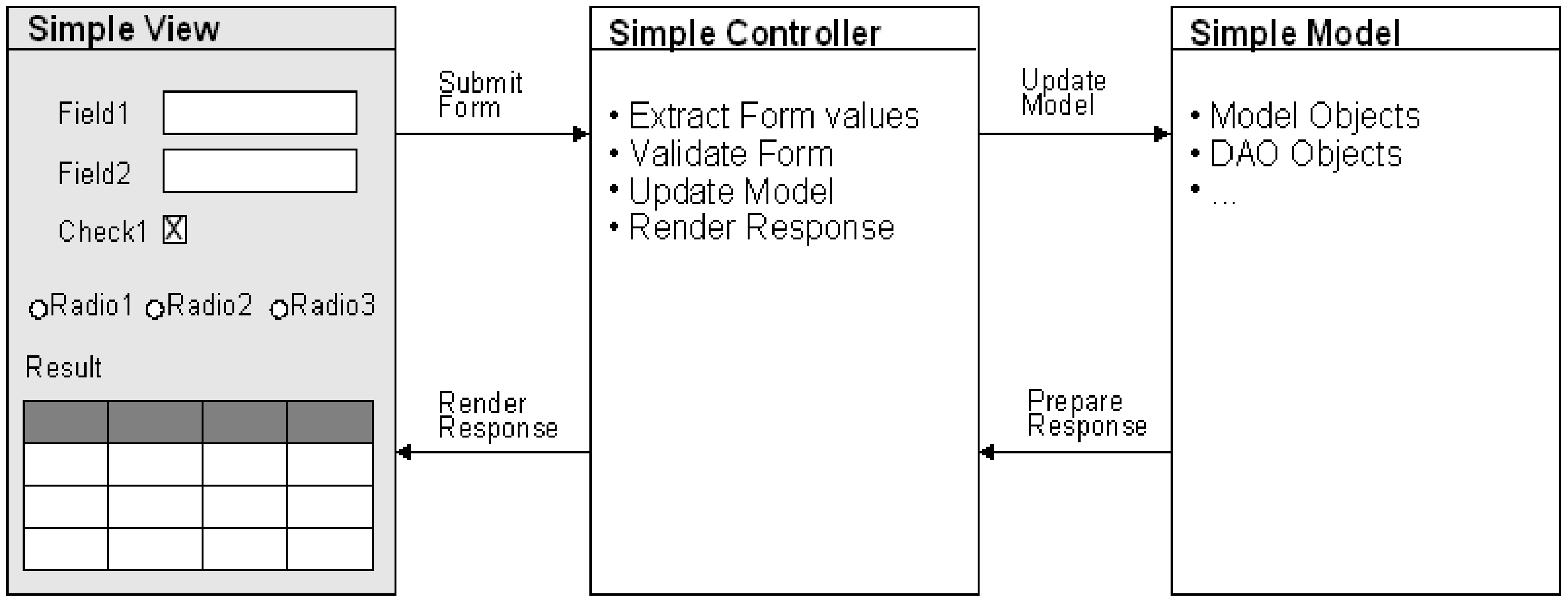
First Name	Last Name	Address	Weight	Birthdate	Color	Married	Rating
Daniel	Seiler	Bächlerweg 31	79.99	15.05.1973	Blue	☑	high

1) User enter their personal data

2) Data gets sent to the server and stored in a DB

3) Table gets updated with the new user

Architecture



simple.zul

```
<?page id="simplePage" title="Simple ZK Application"?>
<zk>
<window id="simpleWindow"
  use="com.processwide.demo.zk.window.SimpleWindow"
  title="Simple ZK Demo" border="normal" width="800px" height="500px">
  <caption label="${simpleWindow.myCaption}" />
  <grid>
    <rows>
      <row>Firstname*:<textbox id="firstname" width="300px" /></row>
      ...
    </rows>
  </grid>
  ...
  <button label="Save" onClick="simpleWindow.addPerson()" />
  <grid id="personsGrid"/>
</window>
</zk>
```

Declaration of external Java class as view handler

Expression language, default: common-el

Scripting code, default: BeanShell

Placeholder for dynamically updated result table

SimpleWindow.java 1/2

```
public class SimpleWindow extends Window {
  private Grid personsGrid;
  public void onCreate() {
    createPersonsGrid();
  }
  private void createPersonsGrid() {
    personsGrid =
      (Grid)Path.getComponent("/simpleWindow/personsGrid");
    // add all the persons
    List<Person> persons =
      PersonDAO.getInstance().getAllPersons();
    for (Iterator<Person> it = persons.iterator();
      it.hasNext();) {
      Person person = it.next();
      addPersonRecord(person);
    }
  } ...
```

SimpleWindow.java 2/2

```
public void addPerson() {
  // extracting the person data
  Textbox firstName = (Textbox)
    Path.getComponent("/simpleWindow/firstname");
  …
  Person personBean = new Person();
  personBean.setFirstname(firstName.getValue());
  …
  PersonDAO.getInstance().addPerson(personBean);
  addPersonRecord(personBean);
}
public void addPersonRecord(Person person) {
  Row row = new Row();
  Rows rows = personsGrid.getRows();
  rows.appendChild(row);
  row.appendChild(new Label(person.getFirstname()));
  …
}
```

Add a delete button

```
public void addPersonRecord(final Person person) {
  final Row row = new Row();
  final Rows rows = personsGrid.getRows();
  rows.appendChild(row);
  // create the delete button
  Button deleteBtn = new Button("delete");
  deleteBtn.addEventListener("onClick", new EventListener() {
    public void onEvent(final Event arg0) throws Exception {
      // delete the person from the database
      PersonDAO.getInstance().deletePerson(person.getId());
      // update the table
      rows.removeChild(row);
      }
    });
  row.appendChild(deleteBtn);
  ...
```

Sorting 1/2

See: http://docs.zkoss.org/wiki/Multiple_Field_Sorting_on_Listbox

1) Manual (programatic) sorting

For each column or listheader define an ascending and a descending comparator

```
<column label="Last Name"
   sortAscending="${ascLastnameComparator}"
   sortDescending="${descLastnameComparator}" />
```

2) Automatic sorting (since ZK Verison 3.5.3)

Fixed table data (without a listmodel):

```
<column label="First Name" sort="auto"/>
```

Dynamic table data (with a listmodel):

```
<column label="First Name" sort="auto(firstname, lastname)" />
```

Sorting 2/2

See: http://docs.zkoss.org/wiki/Grids,_Trees_and_Listbox#The_onSort_Event

To build application that scale and can handle big data sets, the sorting should happen on the database (ORDER BY) and be combined with a proper paging (a db query for each page):

Add a dummy comparator or an auto sort attribute to the column in order to have the column send out onSort events. Also add an id to identify this column in the composer:

```
<column id="addressCol" label="Address" sort="auto(address)" />
// add the event listener to the column to handle sorting
addressCol.addEventListener("onSort", new EventListener(){
  public void onEvent(Event event) throws Exception {
     // TODO do the sorting on the db (ORDER BY)
    // prevent calling of built in sort
     event.stopPropagation();
```

Paging that scales

```
<vbox>
 <listbox id="box" />
 <paging id="paging" />
</vbox>
```

Set the number of actual items as the "totalSize" of the paging controller:

```
paging.setTotalSize(1000000);
```

Now, you have to intercept the "onPaging" event:

```
paging.addEventListener("onPaging", new EventListener() {
   public void onEvent(Event e) {
      PagingEvent pe = (PagingEvent)e;
      int pageNo = pe.getActivePage();
      // retrieve items pageNo*pagesize ..(pageNo+1)*pagesize
      // and assign their content to listitem 0 .. pagesize
   }
});
```

Advanced concepts

- Data binding

```
<textbox id="firstname" value="@{person.firstname}"
width="300px"/>
<grid id="myGrid" model="@{myListmodel}" />
```

- Enhanced MVC

```
<window id="simpleWindow2" ...
  apply="com.processwide.demo.zk.controller.SimpleComposer">
public class SimpleComposer extends GenericForwardComposer {
  public void onClick$saveBtn(Event evt) {
    Person person = (Person)page.getVariable("person");
    ...
```

- Custom components

Data binding

Controlling the data binding manager:

save-when: Tag expression to tell Data Binding Manager when to save data from the component's attribute into the data source

<comp attr=”@{bean.beanAttr, **save when='compId.evtName'**}“/>

load-when: Tag expression to tell Data Binding Manager when to load data from the data source into the component's attribute

<comp attr=”@{bean.beanAttr, **load-when='compId.evtName'**}“/>

access: Tag to control access policy of bean attributes

<comp attr=”@{bean.beanAttr, **access='both|load|save|none'}**“/>

converter: Tag expression to tell Data Binding Manager how to convert data from the data source into the component's attribute

<comp attr=”@{bean.beanAttr, **converter='class-name'}**“/>

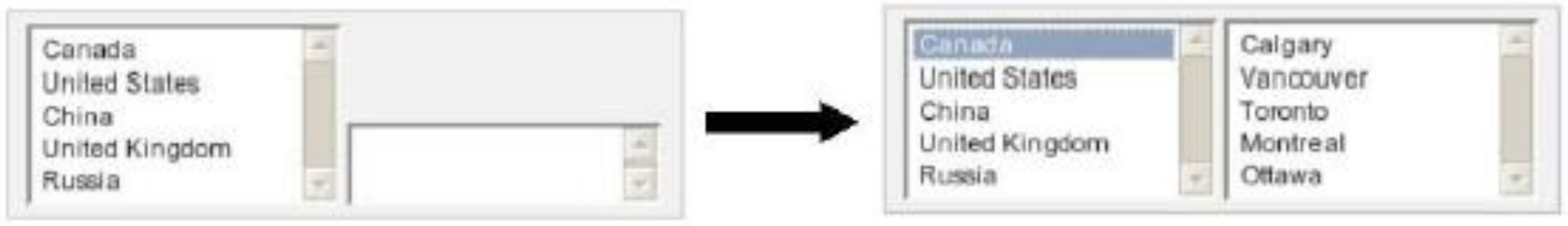
ZK Custom components

The ability to implement custom components easily and intuitively is one of the reasons for the productivity gain ZK brings compared to other frameworks

Create a custom component by ...

- Extending existing Components
- Creating class deriving from org.zkoss.zk.ui. AbstractComponent

Custom component: Double Combo



Client Solution

All the possible values are loaded at page loading to the client and changes are handled on the client through Javascript

+ Once loaded it is fast
- For large data sets the initial loading becomes inpractical

Server Solution (see http://www.ch.ch/karte/index.html?lang=de)

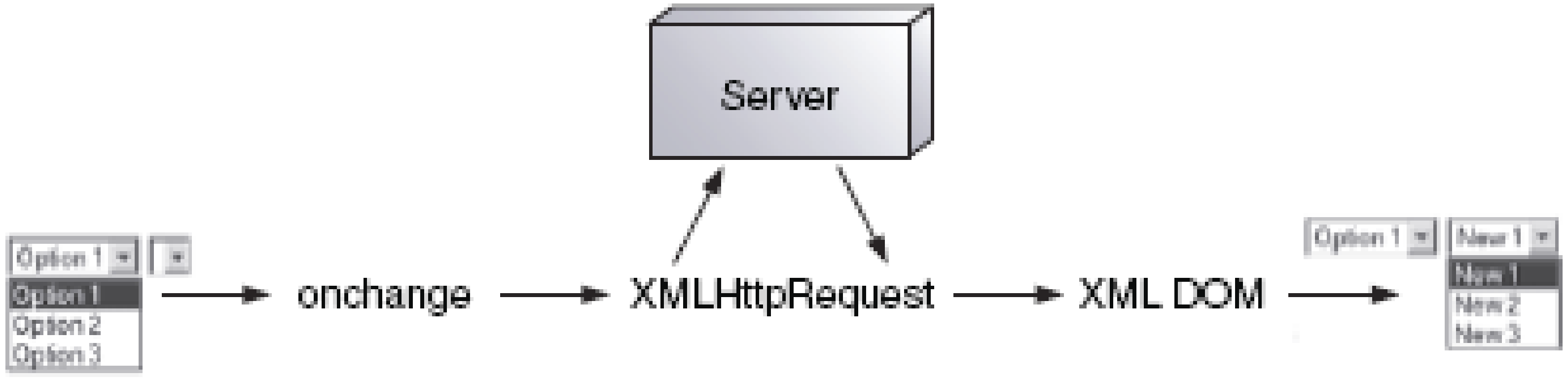
Each time the selection in the first combo changes, the whole page gets reloaded with the corresponding values filled into the second combo

+ Can handle large data sets
- The user experience, response time is bad due to reloading of the whole page

Ajax Solution

Each time the selection changes in combo1 an Ajax request is made and the combo2 gets updated without reloading the whole page

Double Combo: Ajax Solution



+ can handle large data sets
+ user experience and response time is good due to partial update of the page
- complex to implement without a framework like ZK!

Chapter 9 of the book 'Ajax in Action' (online available at: http://www.manning-source.com/books/crane/crane_ch09.pdf) takes around 16 pages to explain the implementation of this simple component!

Double Combo – ZK Implementation

Declaration:

```
<?component name="doubleCombo" extends="div" class="com.
   demo.zk.components.DoubleComboComponent"?>
```

Usage:

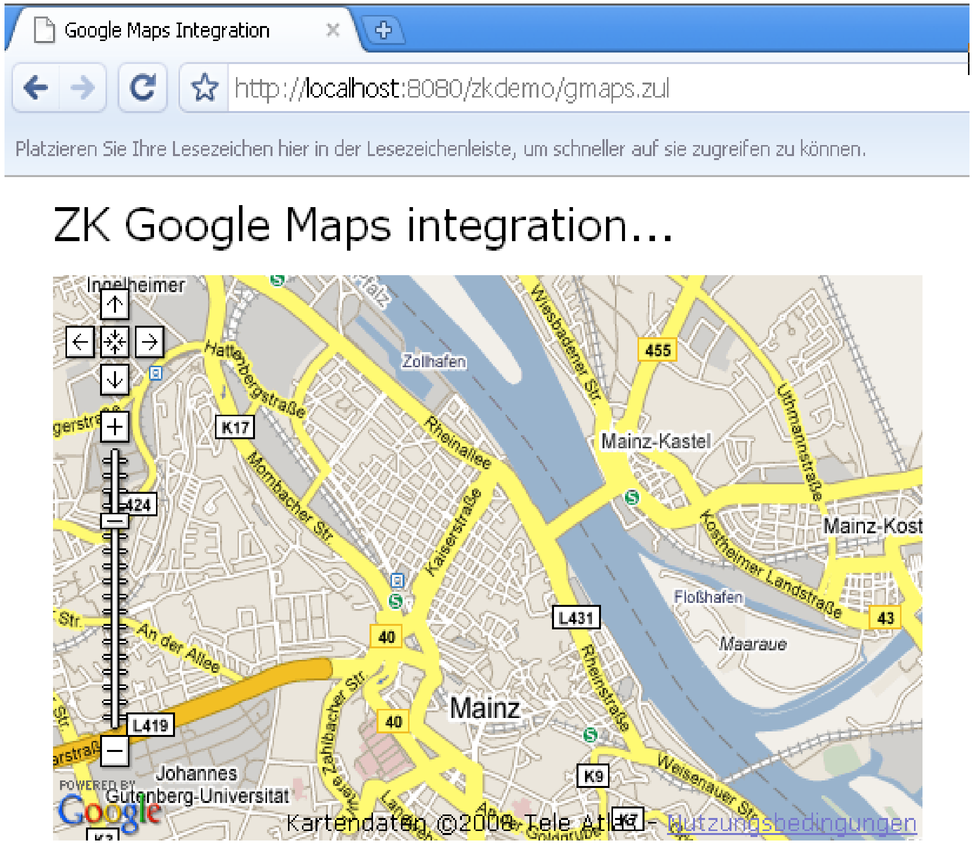
```
<doubleCombo
    orientation="horizontal"
    rows="5,5"
    dataSourceHandler="${componentsWindow.dataSourceHandler}"
    titleCombo1="Kantone in der Schweiz"
    titleCombo2="Gemeinden"
/>
```

Java:

```
package com.processwide.demo.zk.components;
public class DoubleComboComponent extends Div {
... ~40 lines of code
```

Integration with Google Maps

```
<gmaps width="500px" height="300px" lat="50.00385"
              lng="8.26778" showLargeCtrl="true"/>
```


Google Maps Integration
http://localhost:8080/zkdemo/gmaps.zul
Platzieren Sie Ihre Lesezeichen hier in der Lesezeichenleiste, um schneller auf sie zugreifen zu können.
Mainz
Mainz-Kastel
Zollhafen
Floßhafen
Maaraue
Johannes Gutenberg-Universität
POWERED BY Google
Kartendaten ©2008 Tele Atlas - Nutzungsbedingungen

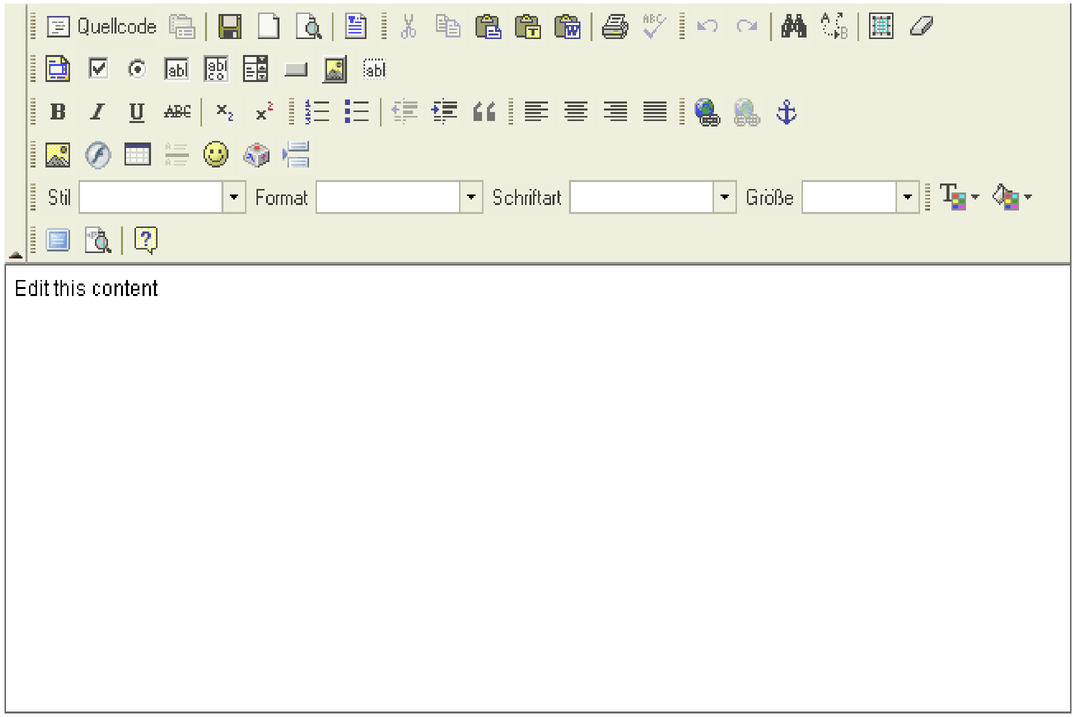
ZK Google Maps integration...

Integrating FCKEditor (http://www.fckeditor.net/)

```
<fckeditor value="Edit this content" width="650px"
  customConfigurationsPath="/js/fckconfig.js"
  toolbarSet="Simple" />
```

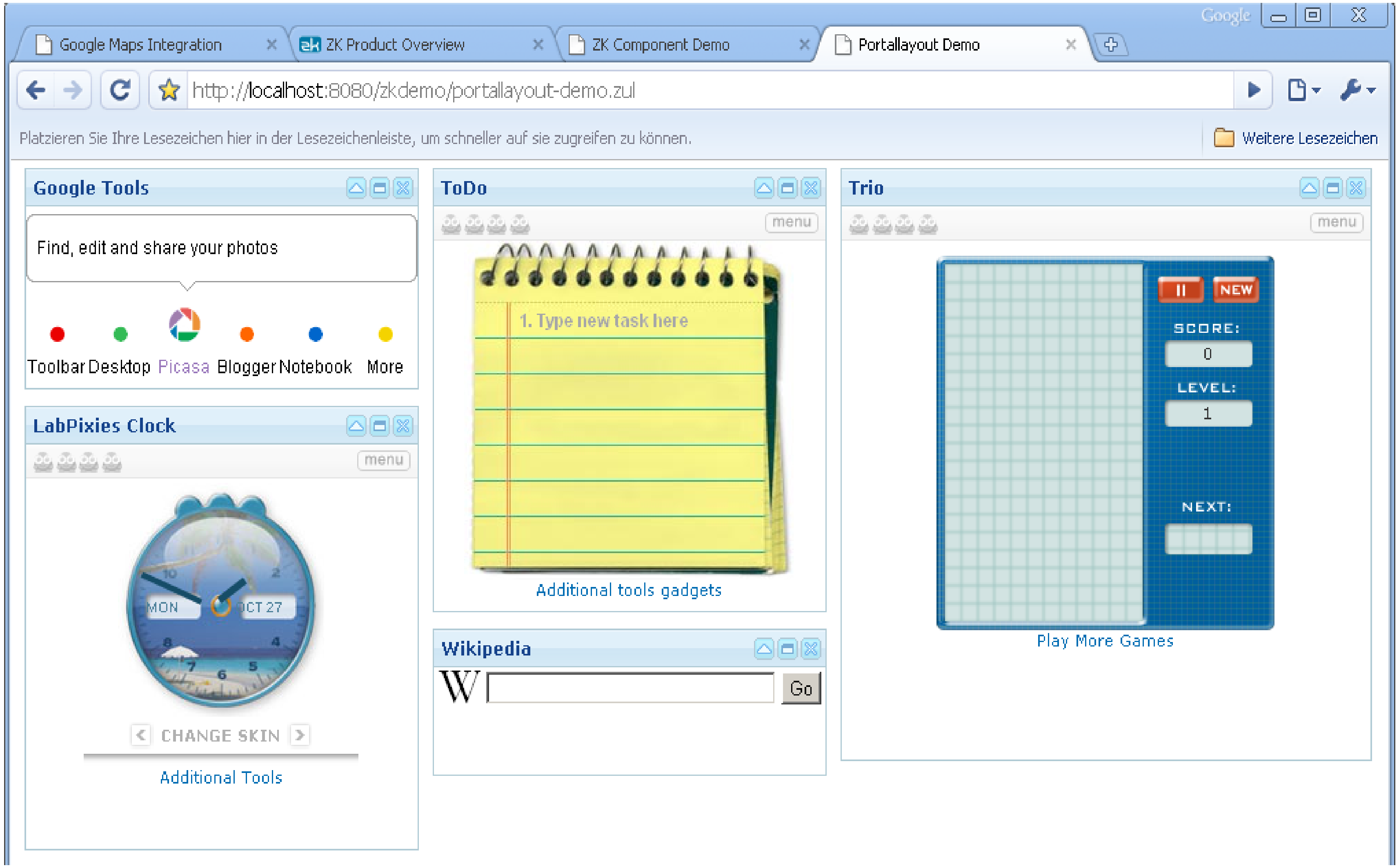
fckconfig.js :

```
FCKConfig.
  ToolbarSets["Simple"]=
[['Bold','Italic',
  'Underline','-',
  'FontFormat', 'Style',
  'FontSize','-',
  'TextColor','-',
  'OrderedList',
  'UnorderedList']
];
```



Edit this content

Portal layout (~ 40 Lines of ZUL)



Server push

// enable server push for this desktop
```
desktop.enableServerPush(true);

Executions.activate(desktop);
```

// do push

```
Executions.deactivate(desktop);
```

zk.xml
```
<device-config>
    <server-push-class>
        org.zkoss.zkmax.ui.comet.CometServerPush |
        org.zkoss.zkex.ui.impl.PollingServerPush
    </server-push-class>
</device-config>
```

ZK and the others – ICEfaces

Technology

ICEfaces is a JavaServer Faces implementation enriched by AJAX functionality: "The ICEfaces Component Suite provides a complete set of enhanced standard and custom JavaServer Faces (JSF) components".

Pros and Cons

(+) Good choice for adding Ajax to existing JSF applications

(+) Good choice for JSF experts that want to stick with JSF

(+) Rich and pretty set of UI components

(-) 3rd party components don't mix with ICEfaces components on the same page

(-) Being a JSF implementations, ICEfaces faces the inherent JSF limitations and complexity. (e.g. validation, complicated JSF lifecycle, configuration issues, JSP's or Facelets, ...)

Conclusion: Simplicity matters!

Compared to ZK, ICEfaces is more like a JSF components set than an Ajax framework.

ZK and the Google Web Toolkit (GWT)

Technology

GWT is a client side framework: develop in Java, compile to Javascript and run in the browser on the client. Each RPC call to the server needs to be explicitly programmed.

Pros and Cons

(+) Webapps can be developed like Swing applications without directly getting in contact with Javascript programming

(+) Open Source and developed by Google

(-) Limited set of Java classes available for the Java to Javascript compiler

(-) Server side part of the application needs to be integrated manually

(-) Limited set of widgets and difficult to extend

(-) Difficult to debug because of client centric nature

Conclusion

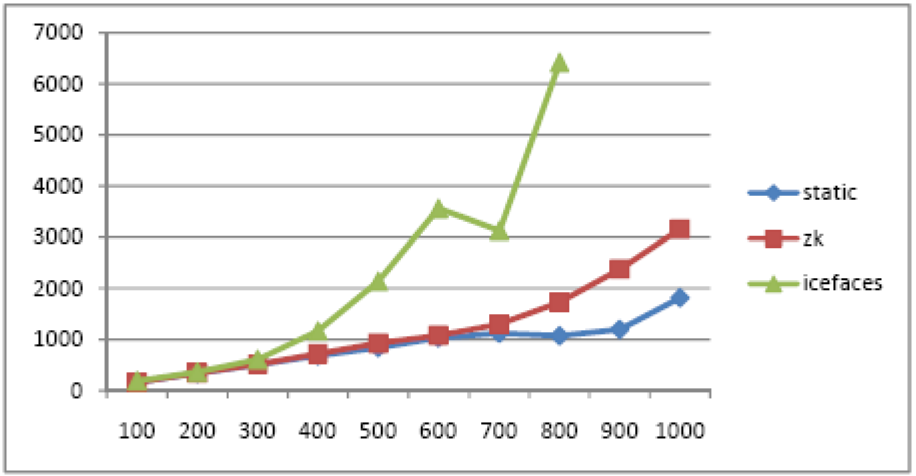
GWT follows an interesting path in creating client centric Ajax applications by compiling Java classes to Javascript. However, if the requirements in terms of richness and responsiveness demand a client centric solution, maybe other client centric technologies like Laszlo, JavaFX or Flex should be considered..

Performance

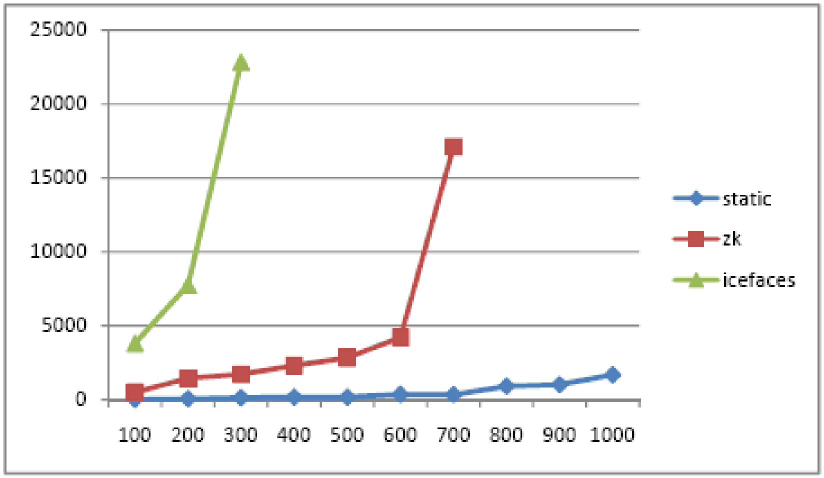
See: http://www.richability.com/resources/icefaces/performance.pdf

Comparison of JSF ICEfaces and ZK:

Grid with 15 records



Grid with 150 records



Simple form:
• The response time of ZK is as good as its static page, and its response is six times faster than ICEfaces with 900 concurrent users.
• In average, ICEfaces consumes 2 times the memory of ZK.

Grid with 15 records:
• Less than 3 seconds response time of ZK with 100 concurrent threads.
• The response time of ZK is 3 times faster than ICEfaces with 900 threads.
• In average, ICEfaces consumes 2 times the memory of ZK.

Grid with 150 records:
• The response time of ZK is less than 5 seconds with 600 concurrent threads.
• ZK is 10 times faster than ICEfaces with 300 concurrent threads.
• In average, ICEfaces consumes 2.5 times the memory of ZK.

Don't use large grids! Implement proper paging on the DB level

What's coming next? ZK 5 July 2009, http://docs.zkoss.org/wiki/Roadmap_2009

Integration with Google App Engine

(http://docs.zkoss.org/wiki/ZK/How-Tos/Installation/How_to_Integrate_ZK_with_Google_App_Engine)

Server Computing

> New generation of data binder
> Truly Ajax-based Web flow.

Client Computing - Full-fledged client-side UI objects (so called widgets):

> 100% server codes to maximize the productivity
> Mostly server codes and limited client codes to improve the response time for critical sections
> 100% client to have the offline capacity

GWT Integration

In addition to accessing ZK object-oriented widgets in JavaScript, developers will be allowed to program them in Java with GWT.

ZK Light

Stripped-down version of ZK 5 with the full-fledged widgets and the minimal server codes.

Summary

Compared to other AJAX Frameworks ZK is simply outstanding in ...

- Productivity
  - "Pay as you go" complexity
  - Fast prototyping
  - Clean separation between GUI design and backend coding
- Component library
- Building custom components
- Support and development activity of the provider
- Documentation
- Integration with other technologies (JSF, JSP, RSS, GMaps, Dojo, FCKEditor, Timeline, PayPal Service, Portlets, JFreeChart, JasperReports, ...

Questions

Daniel Seiler

AdNovum Informatik AG

http://www.adnovum.ch

daniel.seiler@adnovum.ch

ZUL and zscript

What is the result? Why?

```
<window border="normal">
  <label id="l" value="hi label"/>
  <zscript>
    l.value = "hi zscript";
  </zscript>
  ${l.value}
</window>
```

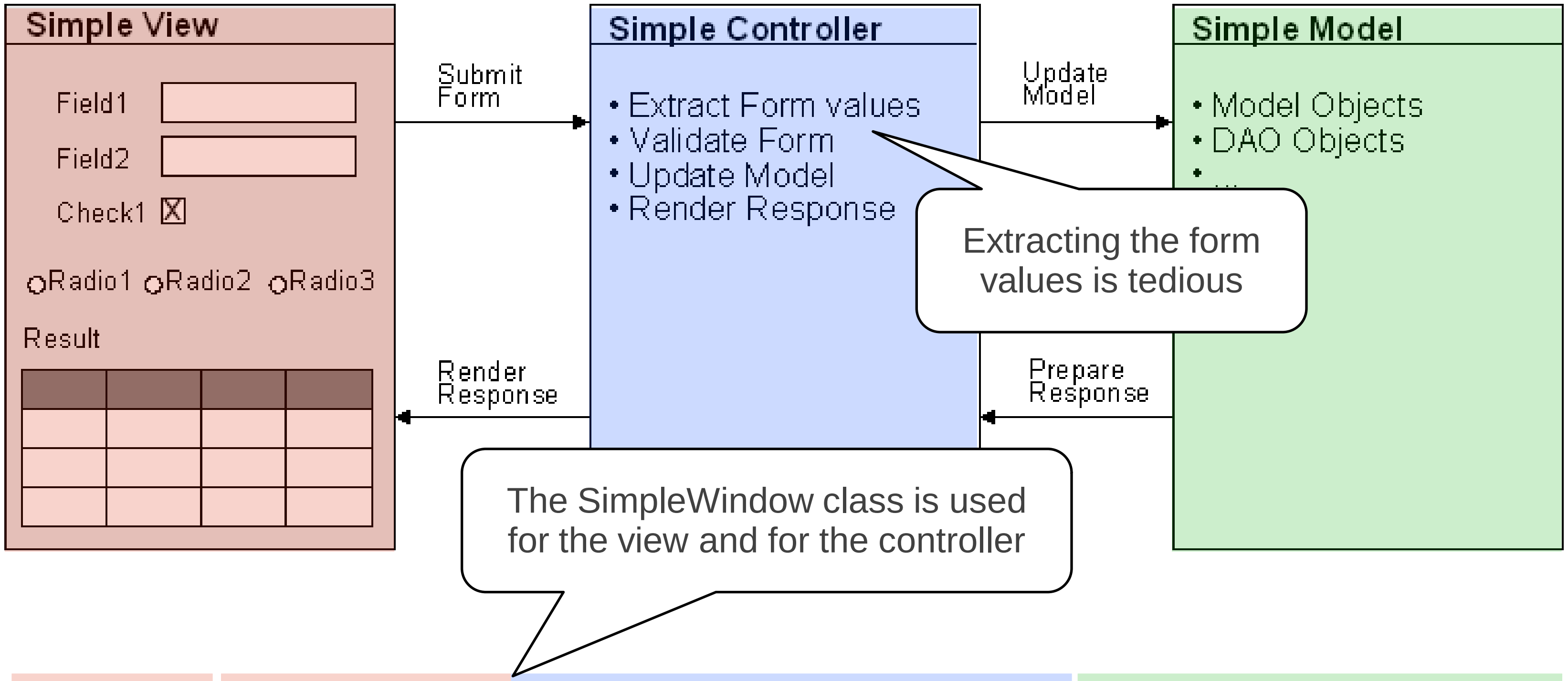
Result: hi zscript hi zscript

What is the result? Why?

```
<window border="normal">
  <label value="${l.value}"/>
  <label id="l" value="hi label"/>
</window>
```

Result: hi label

Improved application – what's wrong?



As is:	simple.zul	SimpleWindow.java	Person.java, PersonDAO.java
To be:	simple2.zul, DateConverter.java	SimpleComposer.java	Person.java, PersonDAO.java

Improved application – simple2.zul 1/2

```
<?init class="org.zkoss.zkplus.databind.AnnotateDataBinderInit" ?>
<?page id="simplePage" title="Simple ZK Application"?>
<zk>
<window id="simpleWindow2" use="com.processwide.demo.zk.window.SimpleWindow2"
      apply="com.processwide.demo.zk.controller.SimpleComposer">
      <grid>
            <rows>
                  <row>
                        Firstname*:
                        <textbox id="firstname"
                        value="@{person.firstname, save-when='lastname.onChange,saveBtn.onClick'}"
                        width="300px" />
                  </row>
                  <row>
                        Lastname*:
                        <textbox id="lastname" value="@{person.lastname}" width="300px" />
                  </row>
                  ...
                  <row>
                        Birthdate:
                        <datebox id="birthdate" value="@{person.birthdate}" format="dd.MM.yyyy" />
                  </row>
                  ..
                  <row>
                        Married:
                        <checkbox id="married" checked="@{person.married}" label="Are you married?" />
                  </row>
            </rows>
      </grid>
```

Initialize the data binding

Apply a proper controller class

Properties are bound to the 'person' variable and the moment of saving gets specified

Improved application – simple2.zul 2/2

The event handler is defined implicitly in the conroller. The button needs an 'id'

```
...
<button id="saveBtn" label="Save" ~~onClick="simpleWindow.addPerson()"~~ />
...
<grid id="personsGrid" model="@{simpleWindow2$SimpleComposer.listModel}">
  <columns sizable="true">
    <column label="First Name" />
    <column label="Last Name" />
    <column label="Address" />
    ...
  </columns>
  <rows>
    <row self="@{each='p'}">
      <label value="@{p.firstname}" />
      <label value="@{p.lastname}" />
      <label value="@{p.address}" />
      <label value="@{p.weight}" />
      <label value="@{p.birthdate, converter='com.processwide.demo.zk.window.DateConverter'}" />
      ...
    </row>
  </rows>
</grid>
</window>
</zk>
```

Assign a list model to the result grid

Assign each row entry a variable 'p'

Print out the person attributes. The birthdate gets converted first.

- no custom window class
- all view related functionality in ZUL
- proper MVC

Improved - SimpleComposer.java

```
public class SimpleComposer extends GenericForwardComposer {
        Radiogroup ratingRadiogroup;
        Component personsGrid;
        ListModelList listModel;

        @Override
        public void doAfterCompose(Component comp) throws Exception {
                super.doAfterCompose(comp);
                Person person = new Person();
                comp.setVariable("person", person, true); // for @{person.xxx}
                listModel = new ListModelList(PersonDAO.getInstance().getAllPersons());
                listModel.addListDataListener(new ListDataListener() {
                        public void onChange(ListDataEvent event) {
                                ((org.zkoss.zkplus.databind.AnnotateDataBinder)page.getVariable("binder")).
                                  loadComponent(personsGrid);
                        }
                });
        }

        public void onClick$saveBtn(Event evt) {
                Person person = (Person)self.getVariable("person",true);
                // for radiogroups the databinding is not supported
                person.setRating(ratingRadiogroup.getSelectedItem().getValue());
                // store the person in the db
                PersonDAO.getInstance().addPerson(person);
                // update the view
                listModel.add(person);
        }
}
```

Improved - DateConverter.java

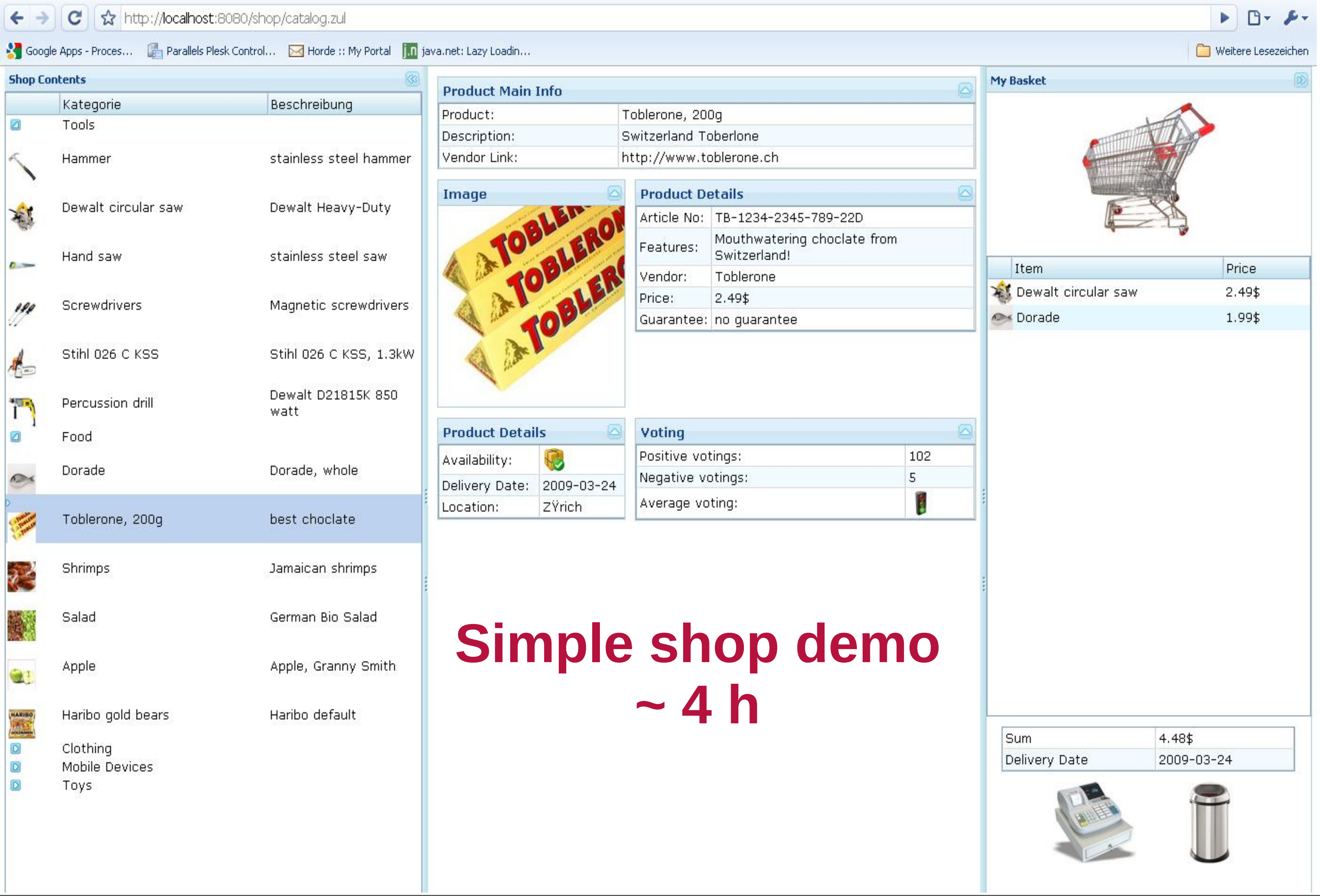
```
...
import org.zkoss.zkplus.databind.TypeConverter;

public class DateConverter implements TypeConverter {

      public static SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy");

      public Object coerceToBean(java.lang.Object val,org.zkoss.zk.ui.Component comp) {
            return null;
      }

      public Object coerceToUi(java.lang.Object val,org.zkoss.zk.ui.Component comp) {
            Date date = (Date)val;
            String dateString = null;
            if(date != null) {
                  dateString = sdf.format(date);
            }
            return dateString;
      }
}
```



http://localhost:8080/shop/catalog.zul

Shop Contents

Kategorie	Beschreibung
Tools	
Hammer	stainless steel hammer
Dewalt circular saw	Dewalt Heavy-Duty
Hand saw	stainless steel saw
Screwdrivers	Magnetic screwdrivers
Stihl 026 C KSS	Stihl 026 C KSS, 1.3kW
Percussion drill	Dewalt D21815K 850 watt
Food	
Dorade	Dorade, whole
Toblerone, 200g	best choclate
Shrimps	Jamaican shrimps
Salad	German Bio Salad
Apple	Apple, Granny Smith
Haribo gold bears	Haribo default
Clothing	
Mobile Devices	
Toys	

Product Main Info

Product:	Toblerone, 200g
Description:	Switzerland Toberlone
Vendor Link:	http://www.toblerone.ch

Image

Product Details

Article No:	TB-1234-2345-789-22D
Features:	Mouthwatering choclate from Switzerland!
Vendor:	Toblerone
Price:	2.49$
Guarantee:	no guarantee

Product Details

Availability:	✓
Delivery Date:	2009-03-24
Location:	ZŸrich

Voting

Positive votings:	102
Negative votings:	5
Average voting:	●

My Basket

Item	Price
Dewalt circular saw	2.49$
Dorade	1.99$

Sum	4.48$
Delivery Date	2009-03-24

Simple shop demo ~ 4 h

JAZOON09 THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY JUNE 22–25, 2009 ZURICH

ADNOVUM

netcetera

Sun microsystems

ZK and the others - Echo2

Technology

Echo2 is a server centric framework similar to ZK.

Pros and Cons

(+) Rich set of components

(-) Documentation

(-) Community support

(-) Integration of server side

Conclusion

From a technology point of view Echo2 and ZK are very similar. In terms of productivity, extensibility, community support and documentation ZK has big advantages.