

15.–18.09.2008
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Mal ganz anders

Von Java nach JavaScript mit dem Google Web Toolkit

Adrian Bürki

Centris AG, Schweiz

Agenda

- Übersicht (10')
- Architektur (10')
- Konzepte (15')
- Bibliotheken (10')
- Demo (15')
- Fragen & Antworten (10')



Agenda

- **Übersicht (10')**
 - Was GWT verspricht
 - Was GWT ist
 - Wozu GWT gut ist
- Architektur (10')
- Konzepte (15')
- Bibliotheken (10')
- Demo (15')
- Fragen & Antworten (10')



Was GWT verspricht

- You write your front end in the Java programming language and GWT compiles your source into highly optimized JavaScript.

Was GWT ist

- Ein Toolkit für die Entwicklung von Web-Anwendungen
 - Java API
 - Tools
- Java2JavaScript Compiler
 - Erzeugt hoch effizienten JavaScript Code
 - Löst Browser Inkompatibilitäten
- Community
 - GWT Diskussionsgruppe (> 15 000 Mitglieder)

Wozu GWT gut ist

- Die Sicht des Entwicklers
 - Eine Programmiersprache (Java)
 - Einfach zu Lernen
 - Ajax wird kinderleicht
 - Browser unabhängig
- Die Management Sicht
 - Google steht hinter dem Toolkit
 - Vollständig freie Lizenz (Apache License v. 2.0)
 - Einsatz der vorhandenen Ressourcen (Menschen und Werkzeuge)

Agenda

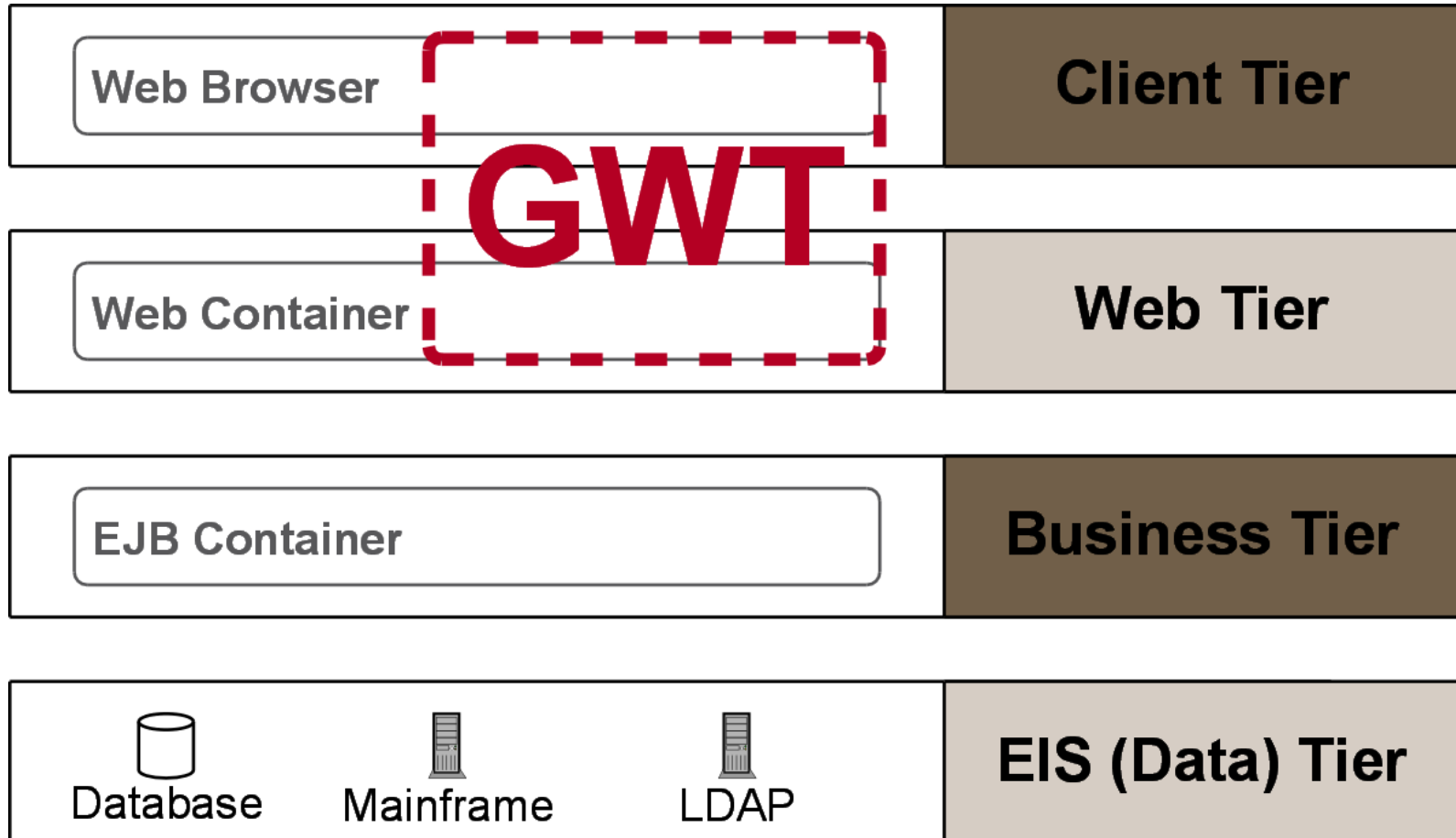
- Übersicht (10')
- **Architektur (10')**
 - **Wo passt GWT rein**
 - **GWT im Java EE Modell**
 - **GWT's Komponenten**
- Konzepte (15')
- Bibliotheken (10')
- Demo (15')
- Fragen & Antworten (10')



Wo passt GWT rein

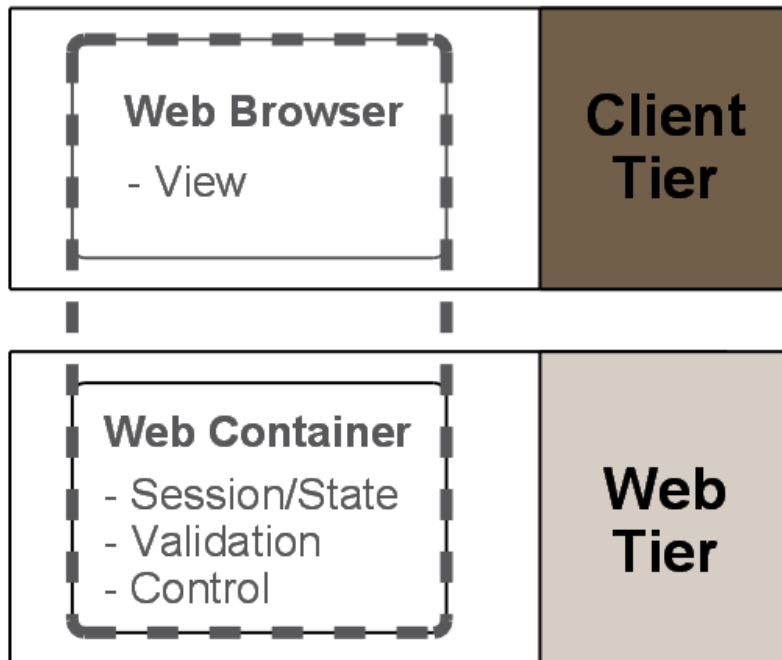
- Java EE Stack
 - Web-Anwendung
 - Web 2.0/Ajax
- Minimale Anforderungen
 - Servlet Engine
 - Gängige Browser mit JavaScript Unterstützung
 - IE 6 und 7
 - Firefox 1, 1.5, 2 und 3
 - Safari 2 und 3
 - Opera 9
 - (IE8 und Google Chrome)

GWT im Java EE Modell

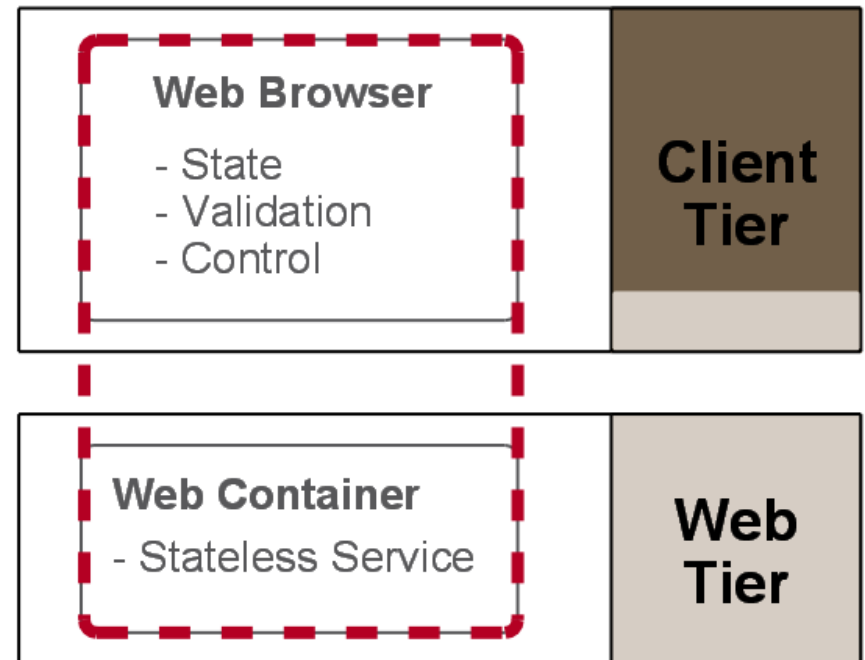


Verschmelzung der Client und Web-Tier

Classical



GWT



GWT's Komponenten

Third party libraries

- > GWT Widget Library
- > GWT Window Manager
- > Hibernate-4gwt
- > gcharts
- > ...

Web UI class library

JRE emulation library

Tools

- > ApplicationCreator
- > ProjectCreator
- > I18NCreator
- > JUnitCreator
- > BenchmarkViewer

Java-to-JavaScript Compiler

Hosted mode browser

Agenda

- Übersicht (10')
- Architektur (10')
- **Konzepte (15')**
 - API's
 - **Remote Procedure Calls**
 - **Deferred Binding**
- Bibliotheken (10')
- Demo (15')
- Fragen & Antworten (10')

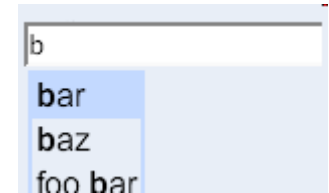


JRE Emulation Library

- Ermöglicht das Übersetzen von Java nach JavaScript
- Nur in dieser Bibliothek enthaltene Klassen können übersetzt werden
- Die Emulation Library enthält :
 - Teile des *java.lang* Packets
 - Teile des *java.util* Packets
 - Das *java.io.Serializable* Interface
- Vorsicht bei der Verwendung von Regular Expressions
- Vorsicht bei der Verwendung von Fließkommazahlen
- Im Zweifelsfall; Blick in die Dokumentation:
<http://code.google.com/webtoolkit/documentation/jre.html>

Web UI class library

- API zum Erstellen von komplexen User Interfaces
- Handhabung ähnlich wie Swing/AWT oder SWT

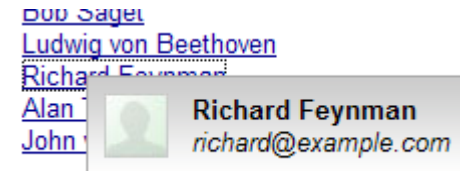


- Die Web UI Library enthält:
 - Vorgefertigte Komponenten (so genannte Widgets)
 - Buttons, Text Boxen, Menus, Dialoge usw...



- Panels um die Komponenten anzuordnen
 - Flow, Horizontal, Vertical und Dock Layout
 - Split Panels, Tabs und Tabellen

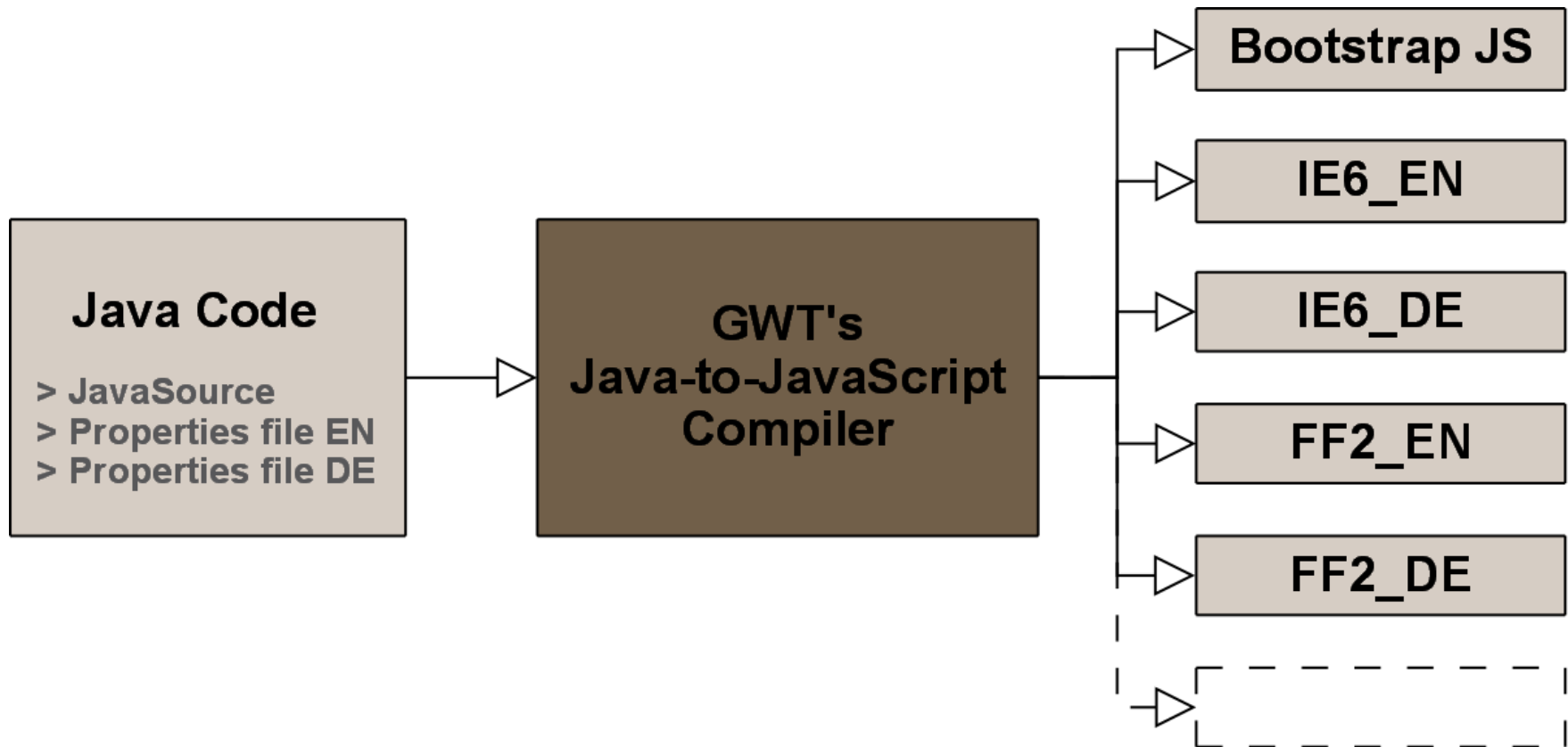
- Zusätzliche Hilfsmittel
 - Event Handling
 - Image Bundles
 - Internationalisierung (I18N)
 - Cookies und History Management



Deferred Binding (1)

- Das Problem => Die Applikation läuft in JavaScript
 - Keine Java Reflection und kein dynamisches Laden von Klassen
- Die Lösung => **Deferred binding**
 - Binding passiert schon während dem Kompilieren
- Der Vorteil => Es wird nur heruntergeladen, was wirklich benötigt wird
 - Download Grösse und Startup Time werden verbessert
- In der Praxis => Keine dynamischen Klassenaufrufe
 - *GWT.create(MyClass.class)* anstelle *Class.forName("MyClass")*

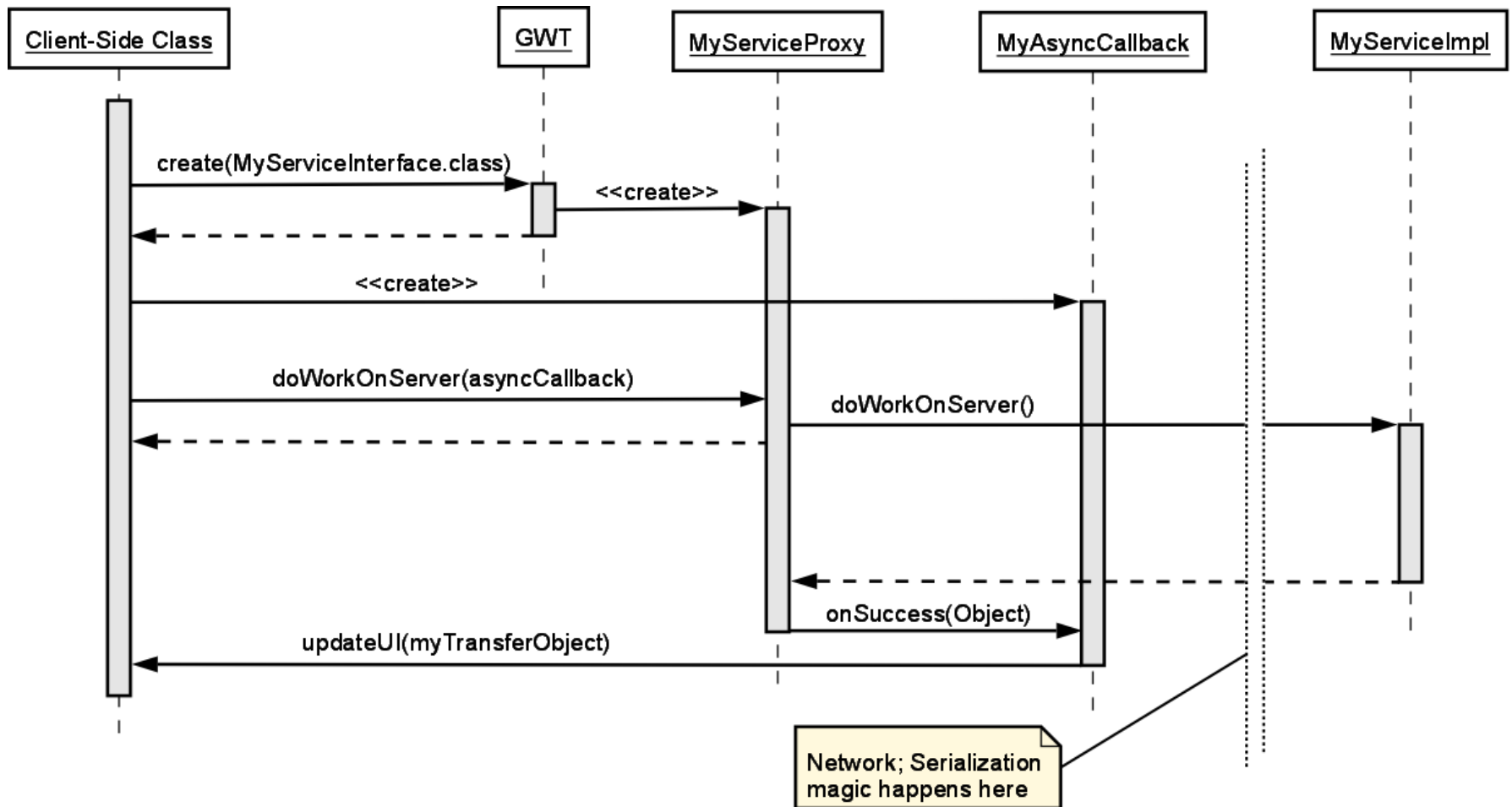
Deferred Binding (2)



Client/Server Kommunikation (1)

- Das Problem => Daten zwischen Client und Server austauschen
 - Transfer von JavaScript zu einem beliebigen Server
- Die Lösung => **Remote Procedure Call**
 - Serverseitig läuft ein Java Servlet
- Der Vorteil => GWT verbirgt die Komplexität und übernimmt die Serialisierung
 - Gleiches Handling für serverseitige und lokale Methoden
 - Aufrufe sind asynchron (nicht blockierend)
 - Alles was nach Java Regeln serialisierbar ist kann übertragen werden
- In der Praxis => Einhalten des RPC „Vertrags“
 - Zwei Interfaces und eine Klasse (Servlet) müssen erstellt werden

Client/Server Kommunikation (2)



Agenda

- Übersicht (10')
- Architektur (10')
- Konzepte (15')
- **Bibliotheken (10')**
 - Was bieten Bibliotheken von Drittanbietern
 - Verfügbare Bibliotheken
 - Nützliche Bibliotheken
- Demo (15')
- Fragen & Antworten (10')



Was bieten Bibliotheken von Drittanbietern

- Fehlende UI Komponenten
 - Erweiterte Tabellen
 - Diagramm Komponenten
 - Google Maps Widget
- Fehlende Funktionalität
 - Spring Integration
 - Hibernate Integration
 - Data Binding
 - Google Gears Integration

Verfügbare Bibliotheken

- <http://code.google.com/p/google-web-toolkit-incubator/>
- <http://code.google.com/p/gwt-maven/>
- <http://code.google.com/p/gwittir/>
- <http://code.google.com/p/gwtx/>
- <http://www.asquare.net/gwttk/>
- <http://www.gwtwindowmanager.org/>
- <http://code.google.com/p/xmlrpc-gwt/>
- <http://code.google.com/p/rocket-gwt/>
- <http://gwt.bouwkamp.com/>
- <http://jwc.sourceforge.net/cgi-bin/bin/view>
- <http://code.google.com/p/gwt-voices/>
- <http://extjs.com/products/gxt/>
- <http://gwt-widget.sourceforge.net/>
- <http://code.google.com/p/emite/>
- <http://code.google.com/p/mogowtools/>
- <http://code.google.com/p/gwt-simple-persistence/>
- <http://code.google.com/p/htmltemplatewidget/>
- <http://code.google.com/p/gwt-rest/>
- <http://code.google.com/p/gwt-rolodex/>
- <http://code.google.com/p/gdf/>
- <http://hibernate4gwt.sourceforge.net/>
- <http://code.google.com/p/gwtgmaps/>
- <http://code.google.com/p/gwt-jsviz/>
- <http://code.google.com/p/gwt-dnd/>
- <http://code.google.com/p/gwt-advanced-table/>
- <http://code.google.com/p/gwt-dom/>
- <http://code.google.com/p/gchart/>
- <http://code.google.com/p/gwtai/>
- <http://code.google.com/p/ofcgwt/>
- <http://code.google.com/p/gwt-spring-hibernate-reference/>
- <http://code.google.com/p/dwr4gwt/>
- <http://code.google.com/p/gwtx/>
- <http://code.google.com/p/gwt-chronoscope/>
- <http://code.google.com/p/struts2gwtplugin/>
- <http://code.google.com/p/gwt-scheduler/>
- <http://code.google.com/p/gwt-pf/>
- <http://code.google.com/p/gwtsimiletimeline/>
- <http://code.google.com/p/gwt-xtender/>
- <http://code.google.com/p/gwt-object-exporter/>
- <http://code.google.com/p/gwt-stuff/>
- <http://code.google.com/p/geez/>
- <http://code.google.com/p/webirc/>
- <http://code.google.com/p/gwtlib/>
- <http://code.google.com/p/gwtphp/>
- <http://code.google.com/p/gwt-model/>
- <http://code.google.com/p/gwtjdbc/>
- <http://code.google.com/p/gwt-fx/>
- <http://code.google.com/p/gwt-commons-logging/>
- <http://code.google.com/p/xmlrpc-gwt/>
- <http://code.google.com/p/lotura/>
- <http://gwtiger.org/>
- <http://code.google.com/p/gwt-google-apis/>
- <http://code.google.com/p/gwtexpress/>
- <http://code.google.com/p/gwt-jsonizer/>
- <http://code.google.com/p/gwt-s3/>
- <http://www.codelathe.com/gwtssh/>
- To be continued...

GWT-WL/SL

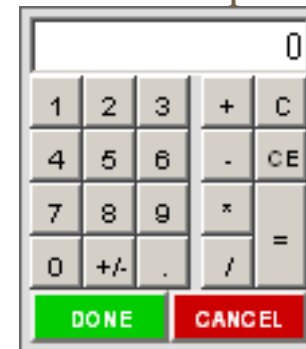
- Eine der ersten Bibliotheken
- Aufgeteilt in zwei Teile:
 - GWT Server Library:
 - Serverseitige Komponenten
 - Fokus auf Spring Integration
 - GWT Widget Library:
 - UI Komponenten
 - Clientseitige Helferklassen
- Apache License v2.0

<http://gwt-widget.sourceforge.net/>

Effects integration



Calculator panel



hibernate4gwt

- Serverseitige Hibernate (Java ORM) Integration
- Persistente Hibernate Objekte (und Lazy Collections) mit GWT nutzen
- Apache License v2.0



<http://hibernate4gwt.sourceforge.net/>

GWT Window Manager

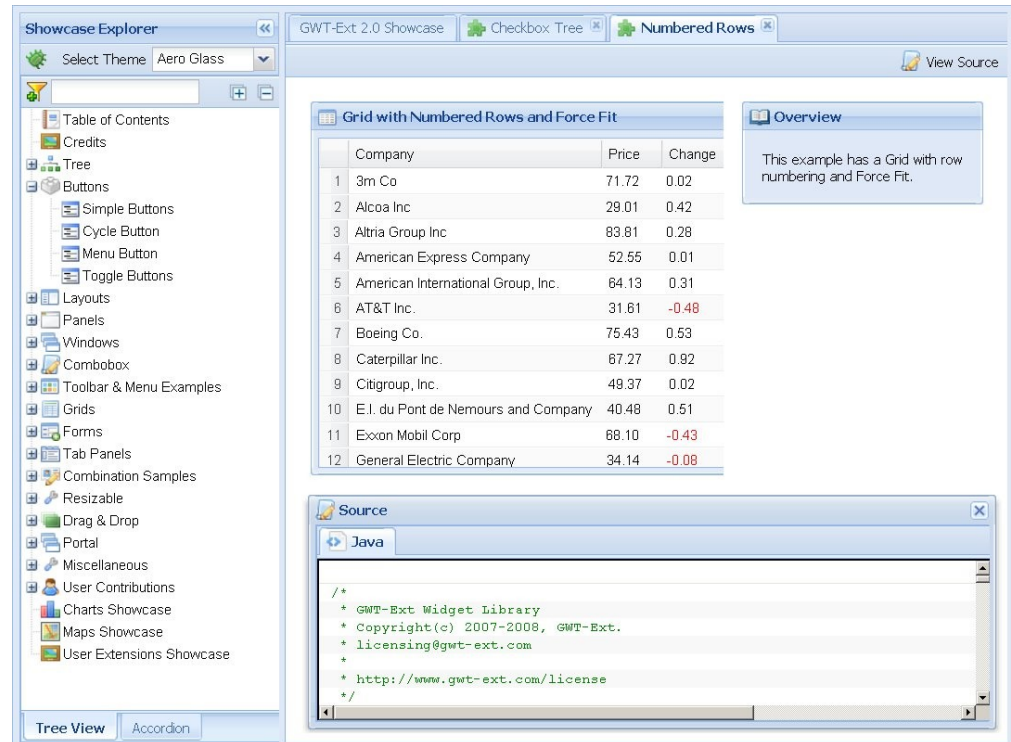
- MDI (Multiple Document Interface) Framework
 - Desktopähnliche Umgebung
 - Dialoge, Fenster und Taskbar
- Mehrere Designs
- Apache License v2.0



<http://www.gwtwindowmanager.org/>

GXT (Ext GWT)

- Wrapper für ExtJS
- Anpassbare UI Komponenten
- Mehrere Motive (CSS)
- Zwei Lizenzen Modell:
 - Kommerzielle Lizenz
 - GNU GPL License v3
<http://extjs.com/products/gxt/>



The screenshot shows the GWT-Ext 2.0 Showcase application. The main window displays a grid titled "Grid with Numbered Rows and Force Fit" with the following data:

	Company	Price	Change
1	3m Co	71.72	0.02
2	Alcoa Inc	29.01	0.42
3	Altria Group Inc	63.81	0.28
4	American Express Company	52.55	0.01
5	American International Group, Inc.	84.13	0.31
6	AT&T Inc.	31.61	-0.48
7	Boeing Co.	75.43	0.53
8	Caterpillar Inc.	67.27	0.92
9	Citigroup, Inc.	49.37	0.02
10	E.I. du Pont de Nemours and Company	40.48	0.51
11	Exxon Mobil Corp	68.10	-0.43
12	General Electric Company	34.14	-0.08

The 'Source' window shows the following Java code:

```

/*
 * GWT-Ext Widget Library
 * Copyright (c) 2007-2008, GWT-Ext.
 * licensing@gwt-ext.com
 *
 * http://www.gwt-ext.com/license
 */
    
```

gwt-chronoscope

- Time series dataset

Visualisierung

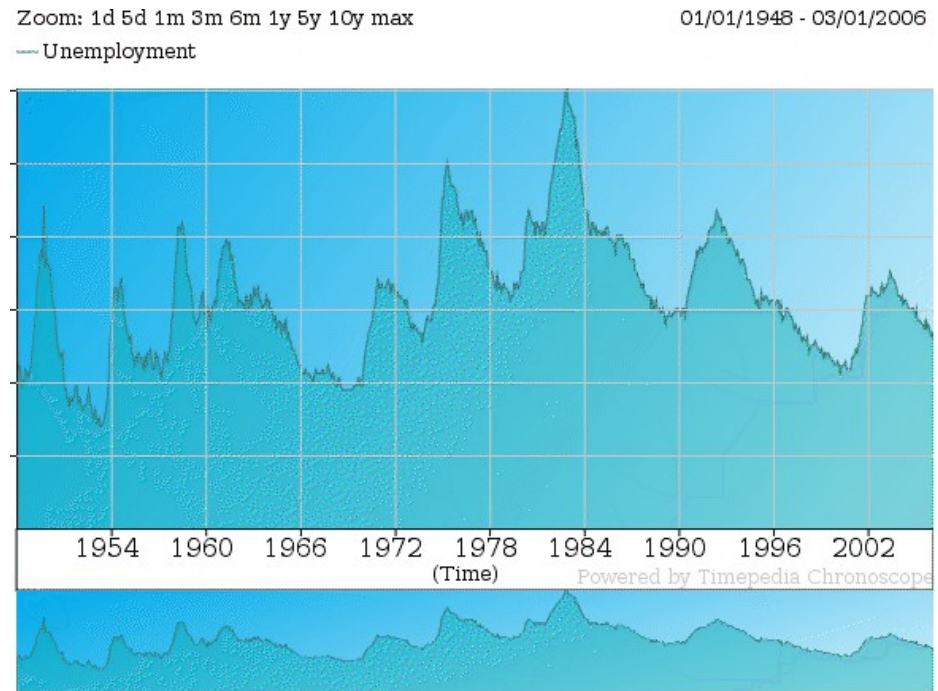
- Skalierbar

- Keyboard und Maus Unterstützung

- LGPL 2.1

- Firefox, Safari und Opera werden unterstützt

<http://code.google.com/p/gwt-chronoscope/>



- IE Support in Arbeit

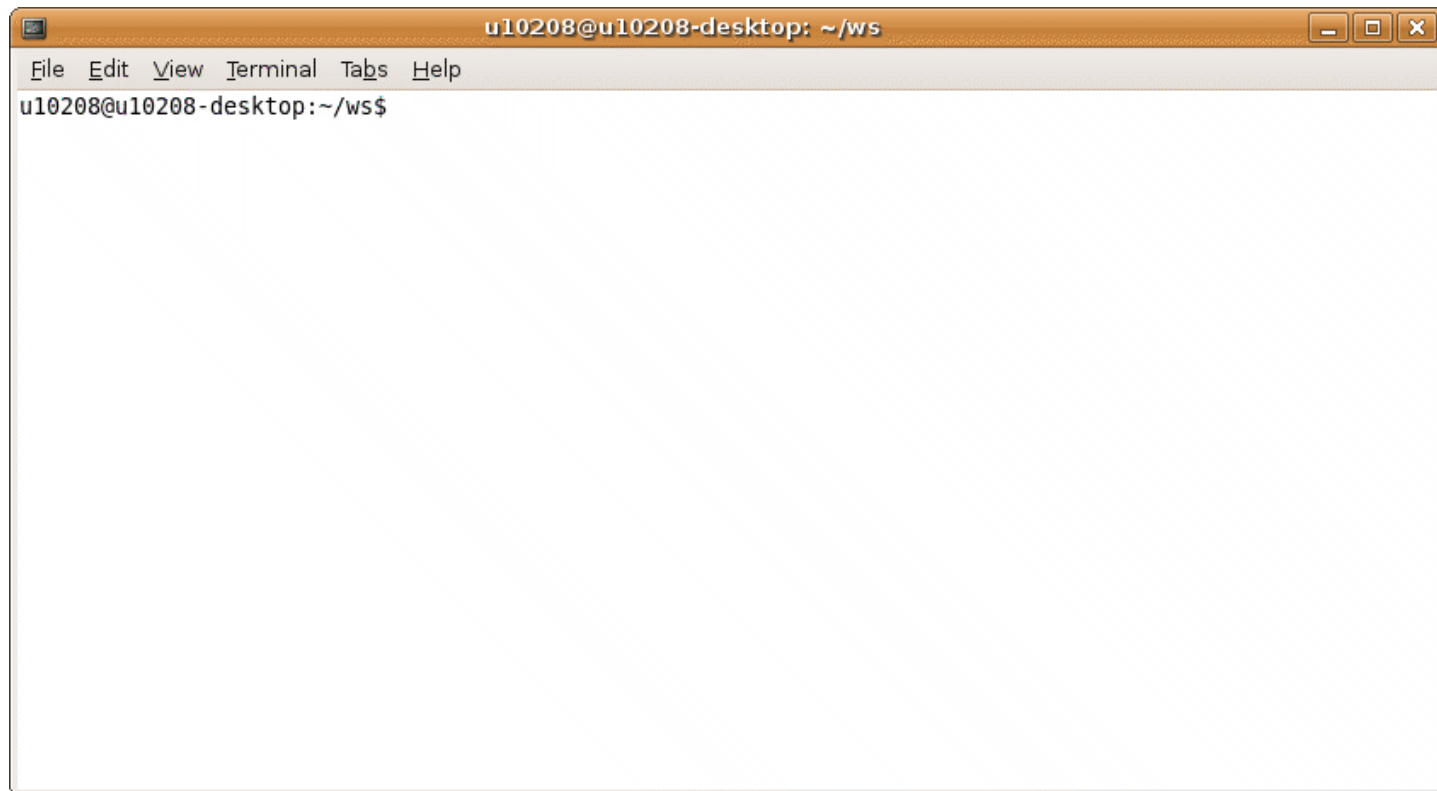
Agenda

- Übersicht (10')
- Architektur (10')
- Konzepte (15')
- Bibliotheken (10')
- **Demo (15')**
 - **Projekt kreieren**
 - **Bibliothek einbinden**
 - **Ausführen und Fehler suchen**
- Fragen & Antworten (10')

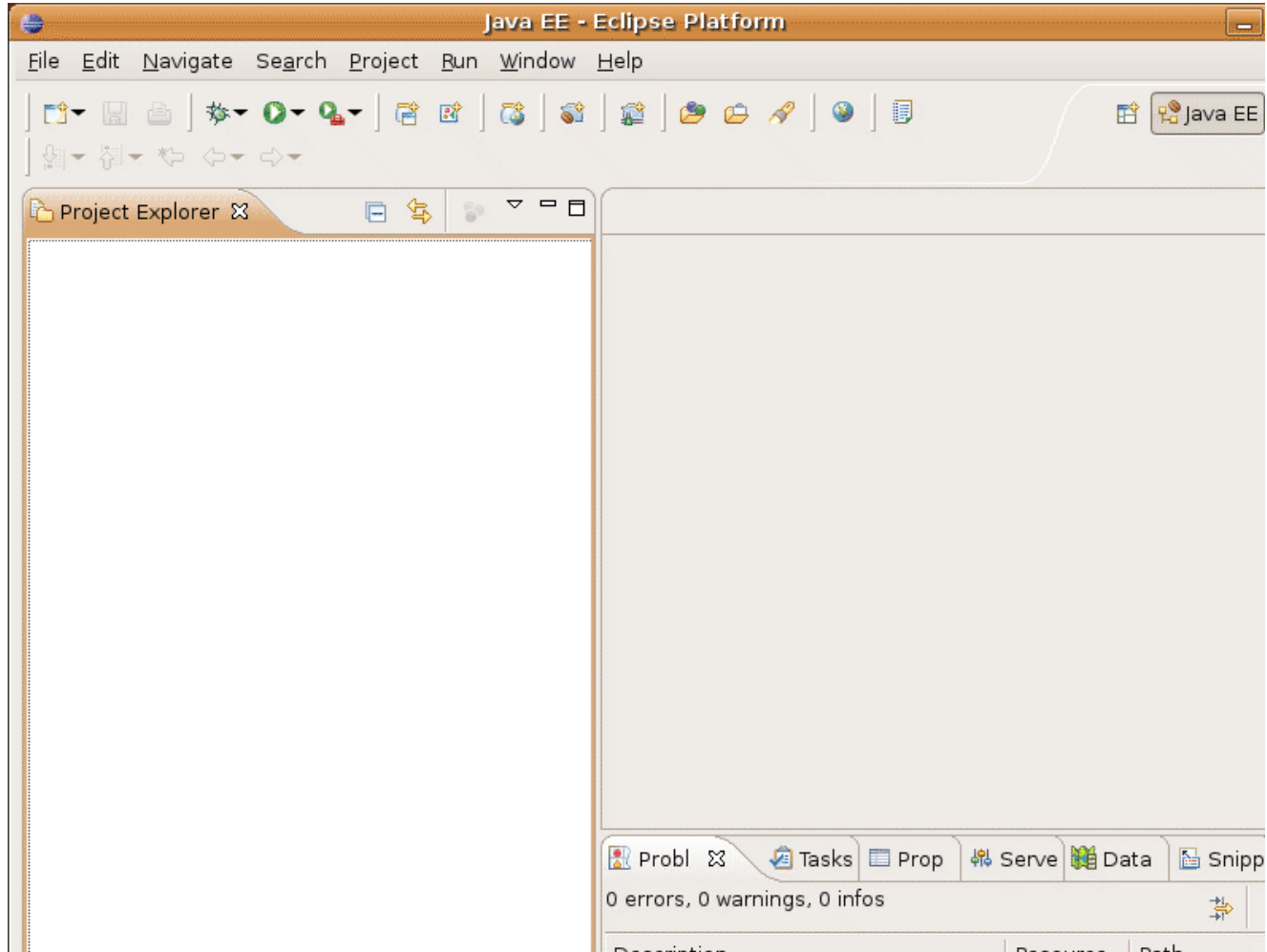


Eclipse Projekt kreieren

projectCreator [-ant projectName] [-eclipse projectName] [-out dir] [-overwrite] [-ignore]
applicationCreator [-eclipse projectName] [-out dir] [-overwrite] [-ignore] className



Eclipse Projekt importieren



HTML Host Page

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4     <title>Demo</title>
5     <script type="text/javascript" language="javascript"
6     src="de.herbstcampus.Demo.nocache.js"></script>
7   </head>
8   <body>
9     <iframe src="javascript:\"" id="__gwt_historyFrame" tabIndex='-1'
10    style="position:absolute;width:0;height:0;border:0"></iframe>
11  </body>
12 </html>
```

EntryPoint Klasse (1)

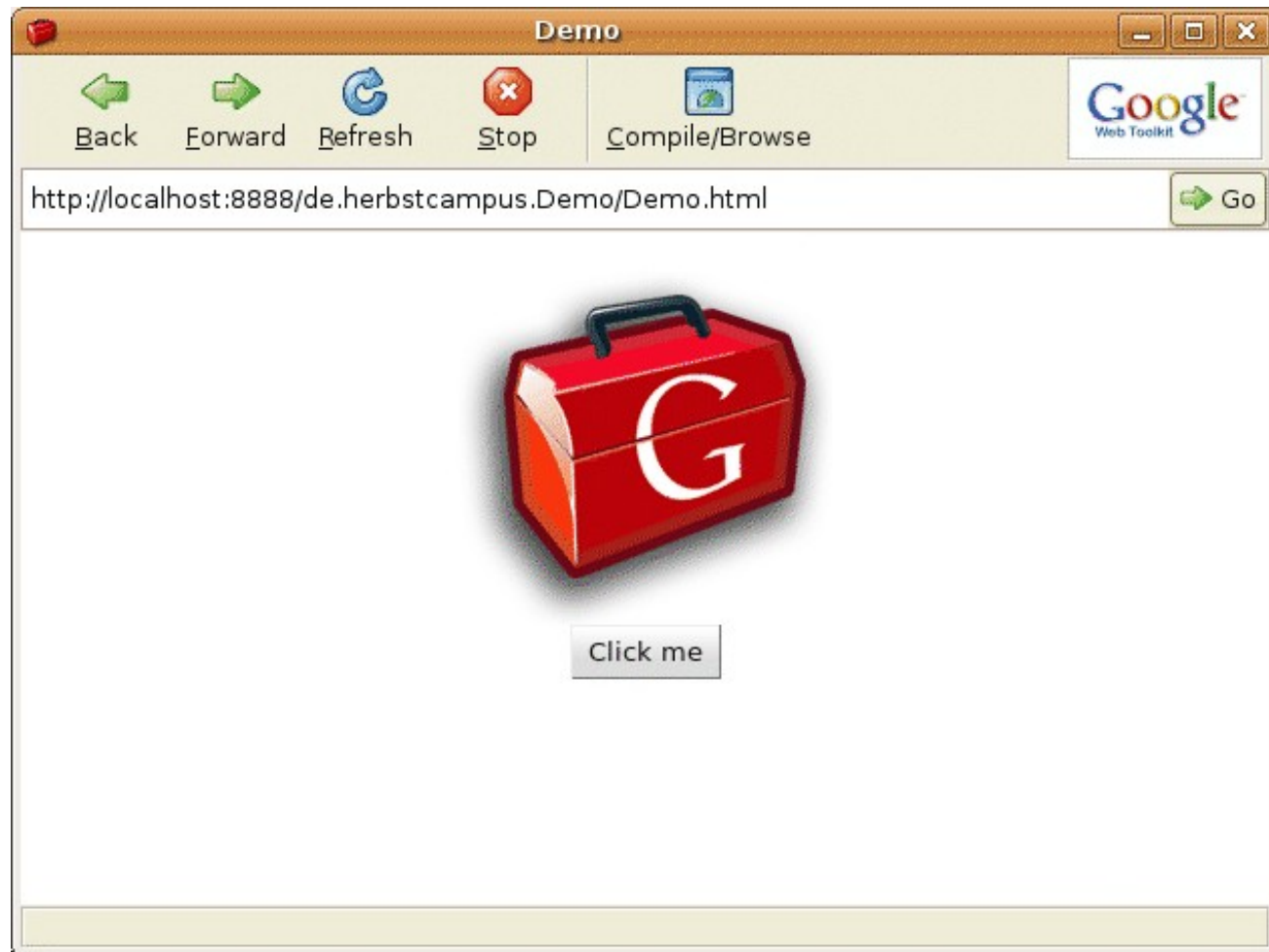
```
10 import com.google.gwt.user.client.ui.Widget;
11
12 public class Demo implements EntryPoint {
13
14     public void onModuleLoad() {
15         Image img = new Image("http://code.google.com/webtoolkit/logo-185x175.png");
16         Button button = new Button("Click me");
17
18         VerticalPanel vPanel = new VerticalPanel();
19         vPanel.addStyleName("widePanel");
20         vPanel.setHorizontalAlignment(VerticalPanel.ALIGN_CENTER);
21         vPanel.add(img);
22         vPanel.add(button);
23
24         RootPanel.get().add(vPanel);
25         ...

```

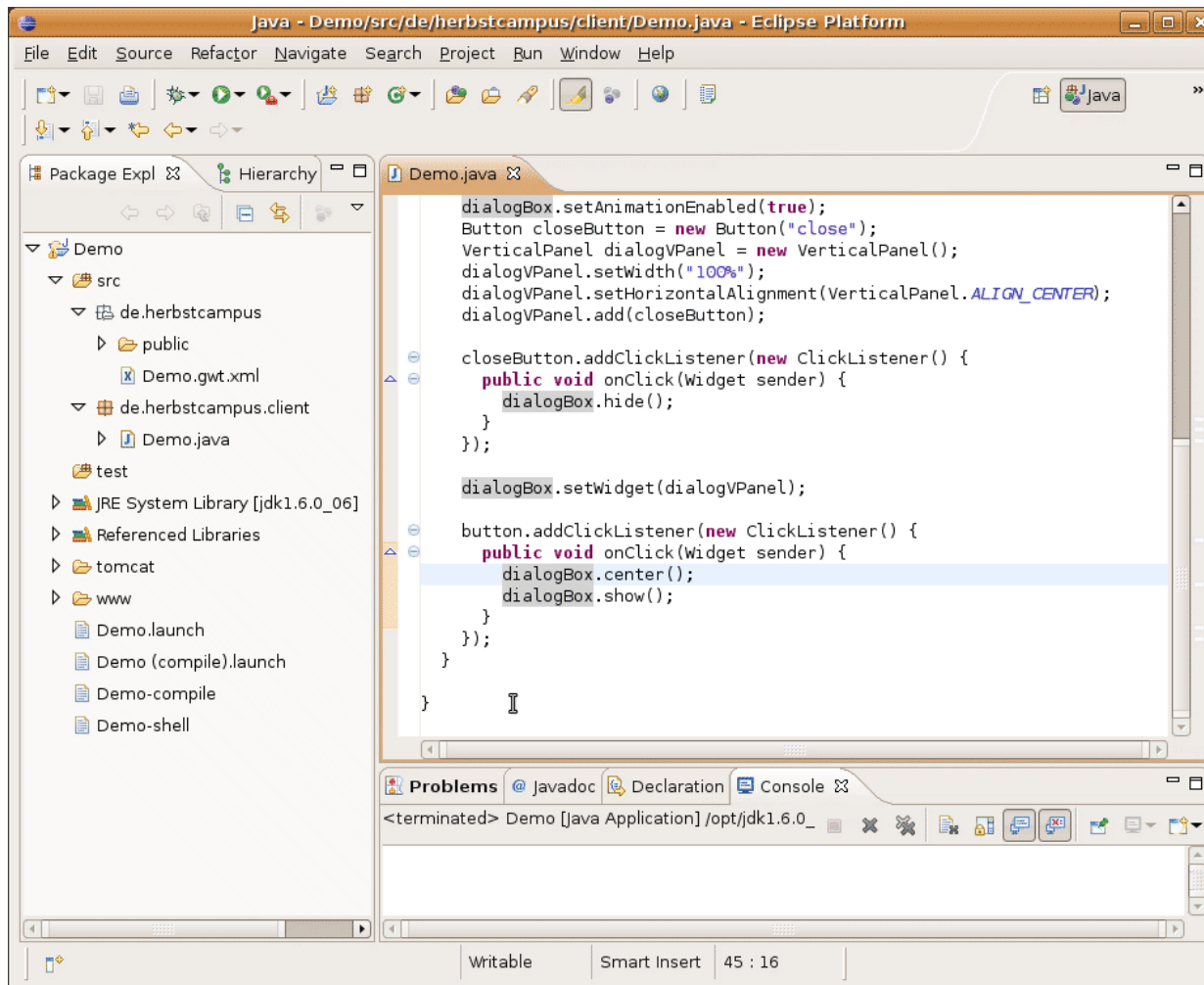
EntryPoint Klasse (2)

```
26     final DialogBox dialogBox = new DialogBox();
27     dialogBox.setText("Welcome to GWT!");
...     ...
41     button.addClickListener(new ClickListener() {
42
43         public void onClick(Widget sender) {
44             dialogBox.center();
45             dialogBox.show();
46         }
47
48     });
49 }
50
51 }
```


Demo ausführen



Fehlersuche



Google AJAX Search API Demo

- Anforderung
 - Eine Sammlung von ‚Such-Lesezeichen‘
 - Einbindung der Suchergebnisse in die Anwendung
 - Webanwendung mit klassischem Layout; Links Navigation und Rechts Inhalt
- Umsetzung
 - GWT Projekt ohne Server Teil
 - Integration einer Bibliothek aus den GWT Google APIs (Google AJAX Search 1.0)
 - <http://code.google.com/p/gwt-google-apis/>

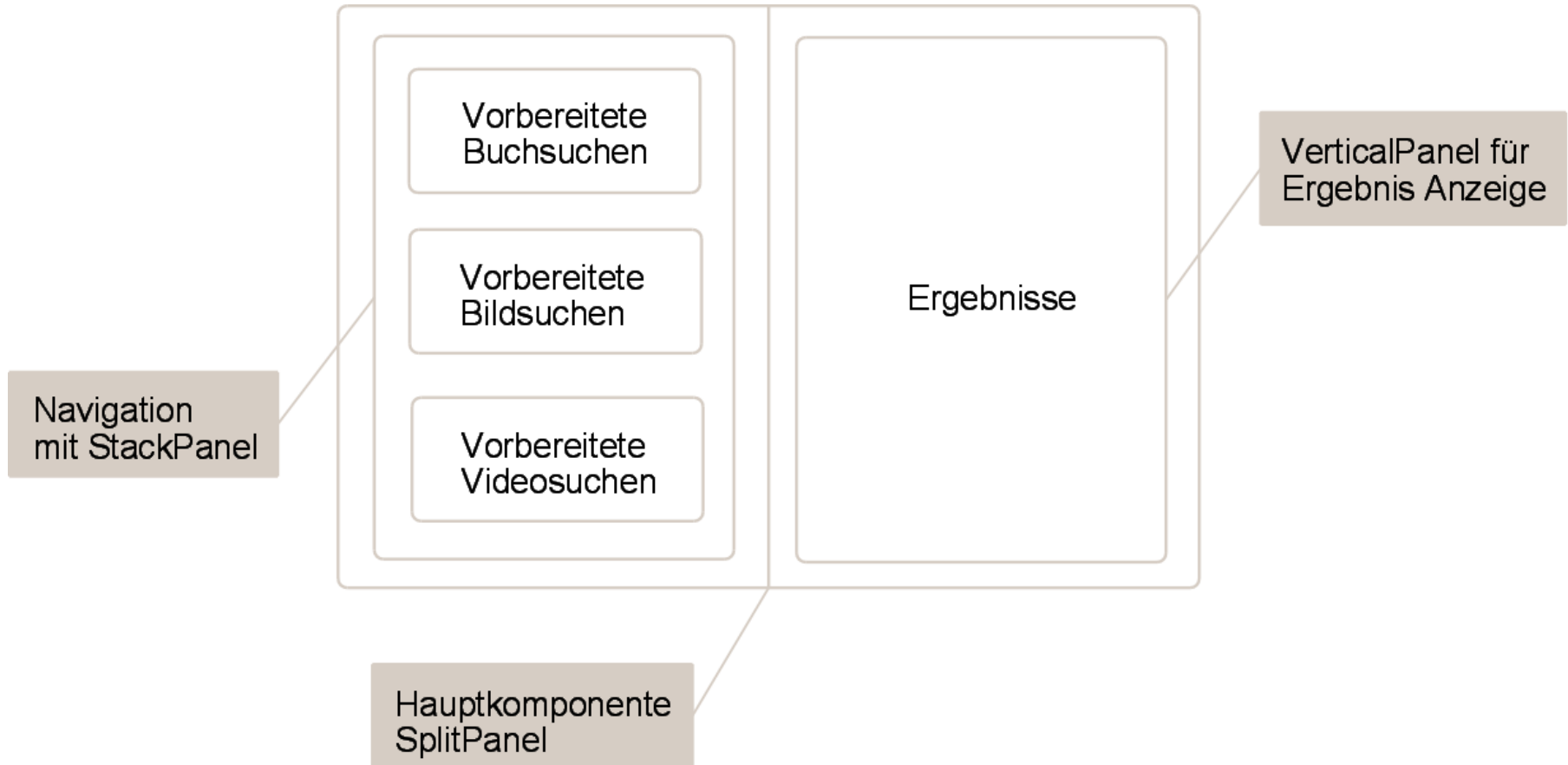
Google AJAX Search API Integrieren

- Download JAR Datei (gwt-search.jar)
- Einbinden der JAR Datei in den Klassenpfad
- Dem GWT Compiler mitteilen, dass wir von dieser Bibliothek abhängig sind (via **Demo.gwt.xml**)
- Der Google Search Dienst erfordert einen API Key und ein entsprechendes **<script>** Tag in der Seite um sich Anzumelden

Demo.gwt.xml

```
1 <module>
2   <inherits name='com.google.gwt.user.User' />
3   <inherits name='com.google.gwt.user.theme.standard.Standard'
4   />
5   <inherits name='com.google.gwt.search.Search' />
6
7   <entry-point class='de.herbstcampus.client.Demo' />
8
9   <stylesheet src='Demo.css' />
</module>
```

Anordnung Bedienoberfläche



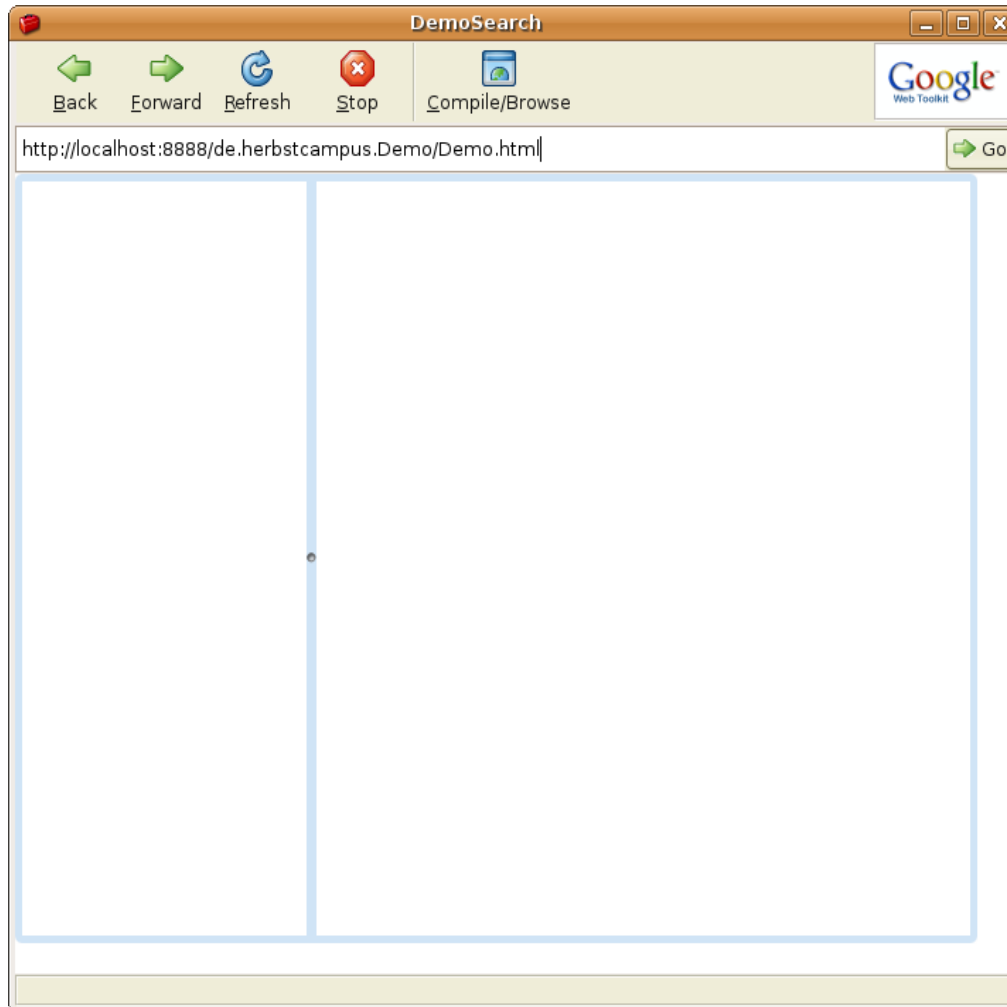
Demo Klasse (1)

```
10 public class Demo implements EntryPoint {
11     private VerticalPanel _resultsPanel;
12
13     public void onModuleLoad() {
14         HorizontalSplitPanel mainSplitPanel = new HorizontalSplitPanel();
15         mainSplitPanel.setSize("670px", "500px");
16         mainSplitPanel.setSplitPosition("30%");
17
18         _resultsPanel = new VerticalPanel();
19         _resultsPanel.setSpacing(5);
20
21         NavigationPanel navigationPanel = new NavigationPanel(this);
22
23         mainSplitPanel.setLeftWidget(navigationPanel);
24         mainSplitPanel.setRightWidget(_resultsPanel);
```

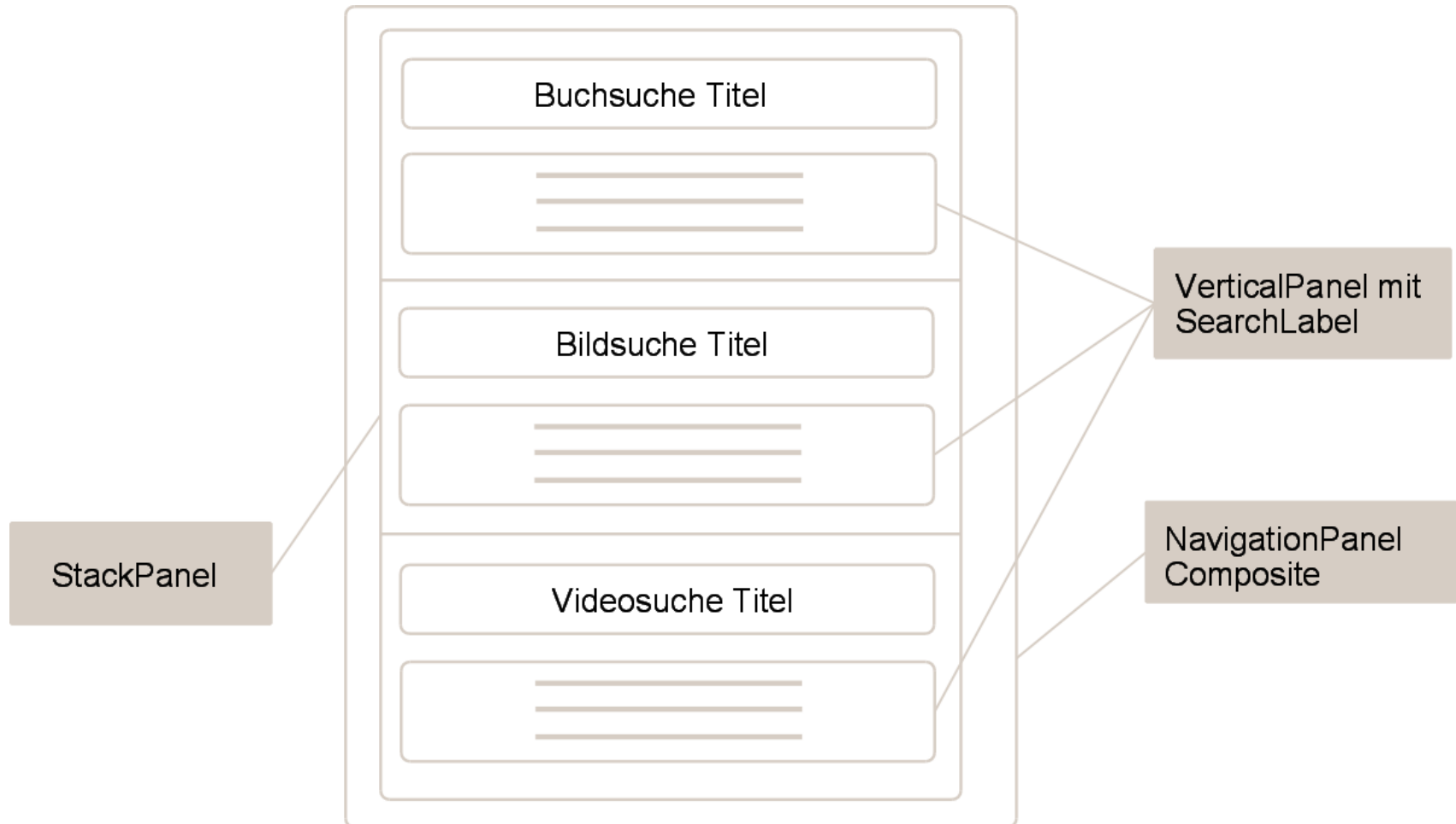
Demo Klasse (2)

```
26     DecoratorPanel decoratorPanel = new DecoratorPanel();
27     decoratorPanel.setWidget(mainSplitPanel);
28
29     RootPanel.get().add(decoratorPanel);
30 }
31
32 public void updateResults(Widget results) {
33     _resultsPanel.clear();
34     _resultsPanel.add(results);
35 }
36
37 }
```


Ausführen (zum Ersten)



Anordnung Navigation



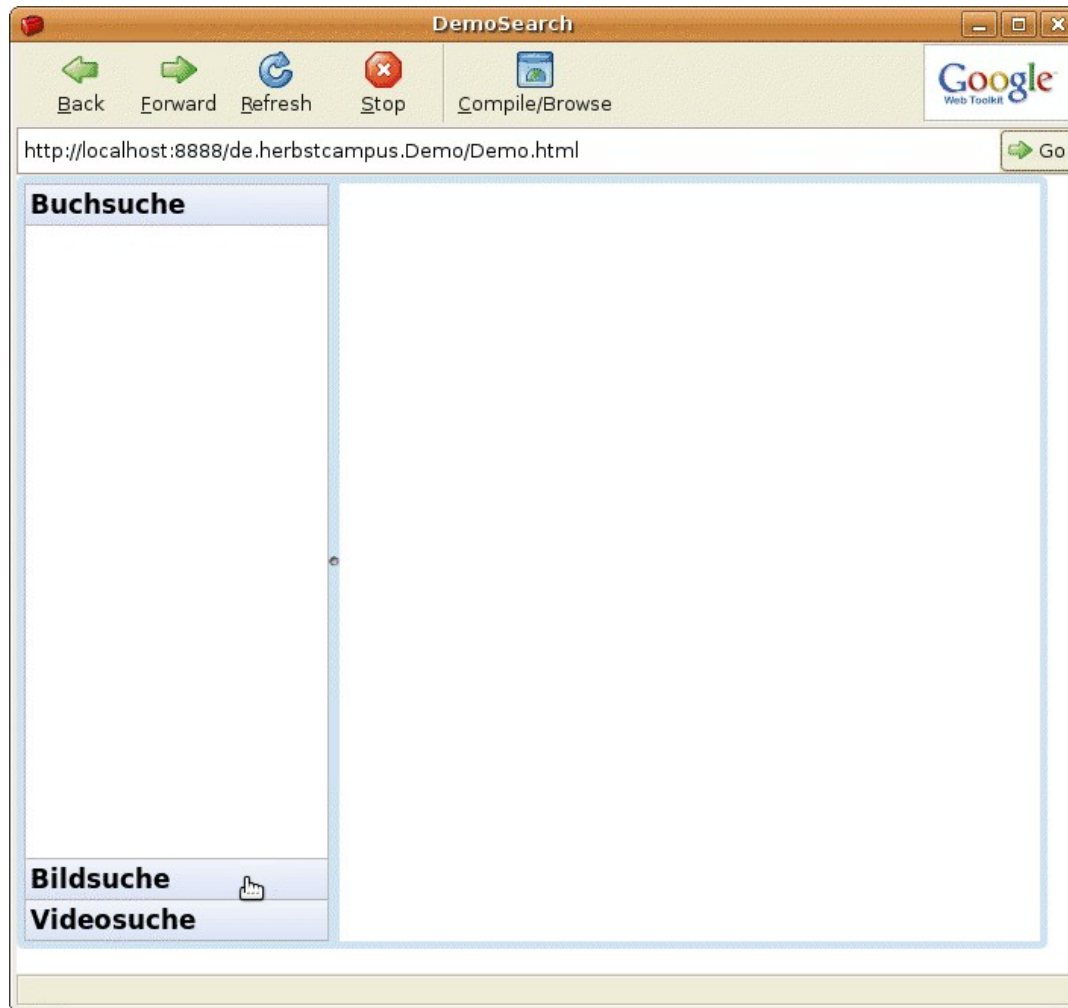
NavigationPanel (1)

```
10 public class NavigationPanel extends Composite {
11
12     public NavigationPanel(Demo parent) {
13         StackPanel mainStackPanel = new StackPanel();
14         mainStackPanel.setWidth("100%");
15
16         BookSearch bookSearch = new BookSearch();
17
18         VerticalPanel bookVPanel = new VerticalPanel();
19         bookVPanel.setSpacing(5);
20         bookVPanel.setWidth("100%");
21
22         bookVPanel.add(new SearchLabel("Java", bookSearch, parent));
23         bookVPanel.add(new SearchLabel("GWT", bookSearch, parent));
24         bookVPanel.add(new SearchLabel(".NET", bookSearch, parent));
25
```

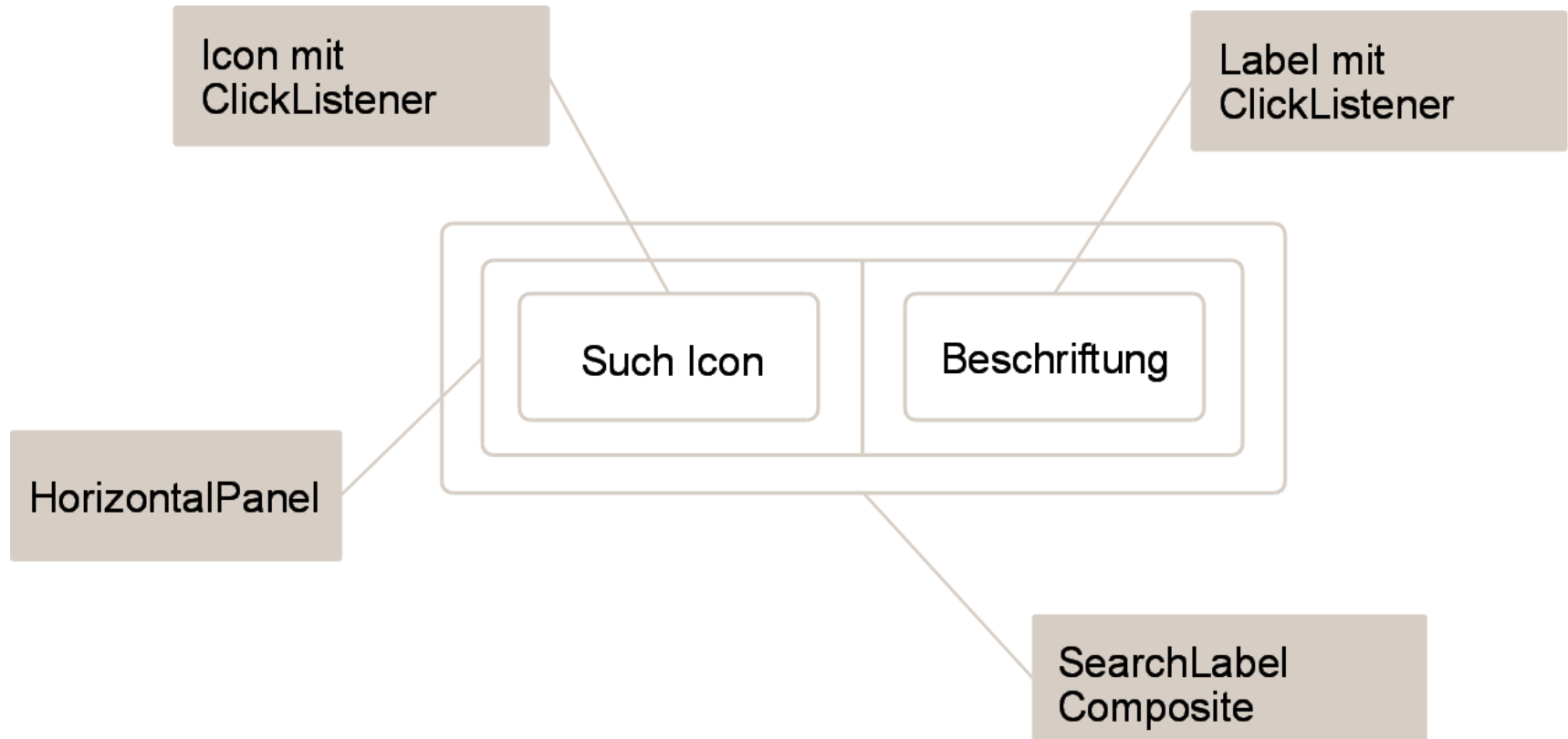
NavigationPanel (2)

```
26     ImageSearch imageSearch = new ImageSearch();
27
28     VerticalPanel imageVPanel = new VerticalPanel();
29     imageVPanel.setSpacing(5);
30     imageVPanel.setWidth("100%");
31     imageVPanel.add(new SearchLabel("Athen", imageSearch, parent));
...     ...
43     VideoSearch videoSearch = new VideoSearch();
...     ...
53     mainStackPanel.add(bookVPanel, "Buchsuche");
54     mainStackPanel.add(imageVPanel, "Bildsuche");
55     mainStackPanel.add(videoVPanel, "Videosuche");
56
57     initWidget(mainStackPanel);
58 }
59 }
```

Ausführen (zum Zweiten)



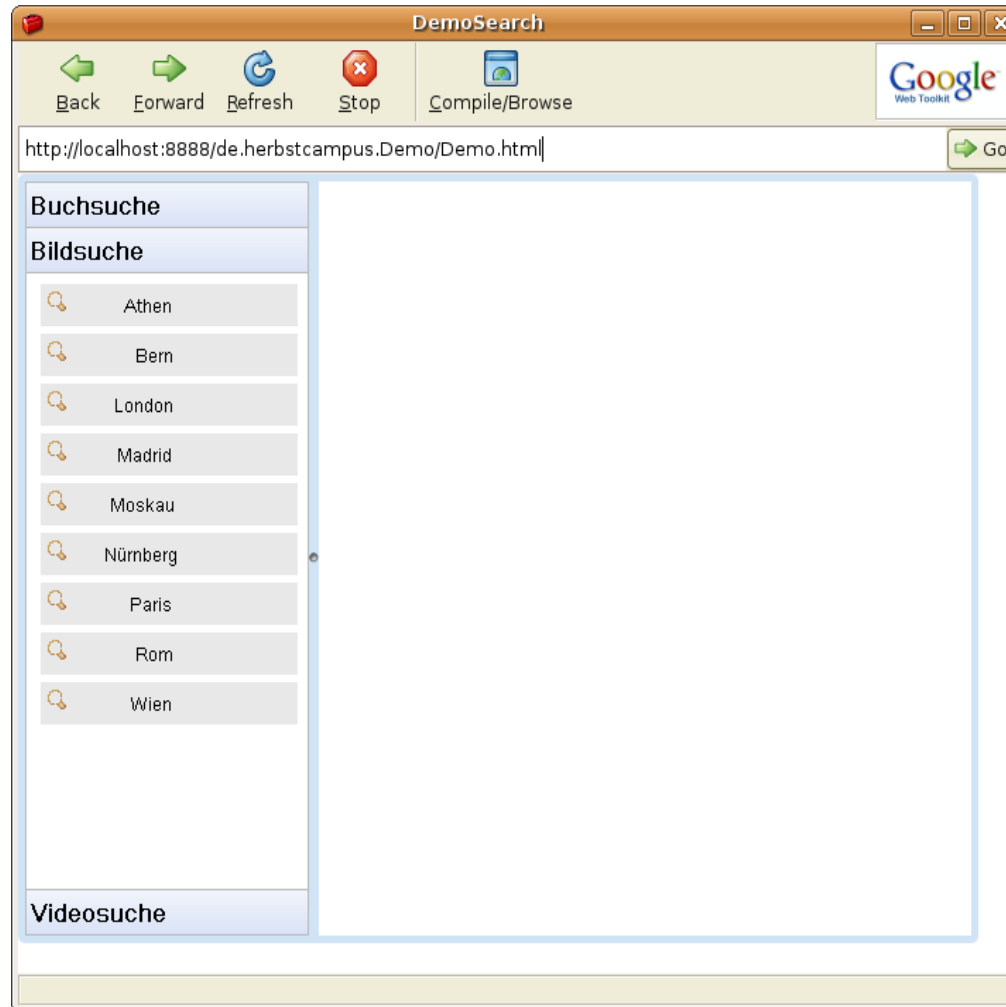
Anordnung SearchLabel



SearchLabel (1)

```
14 public class SearchLabel extends Composite {
15
16     public SearchLabel(final String query, final Search search, final Demo parent) {
17         HorizontalPanel mainPanel = new HorizontalPanel();
18         mainPanel.setStyleName("lblSearch");
19         mainPanel.setSpacing(5);
20
21         Label queryLabel = new Label(query);
22         Image searchImage = new Image("search.gif");
23
24         mainPanel.add(searchImage);
25         mainPanel.add(queryLabel);
26
27         initWidget(mainPanel);
28     }
29 }
```

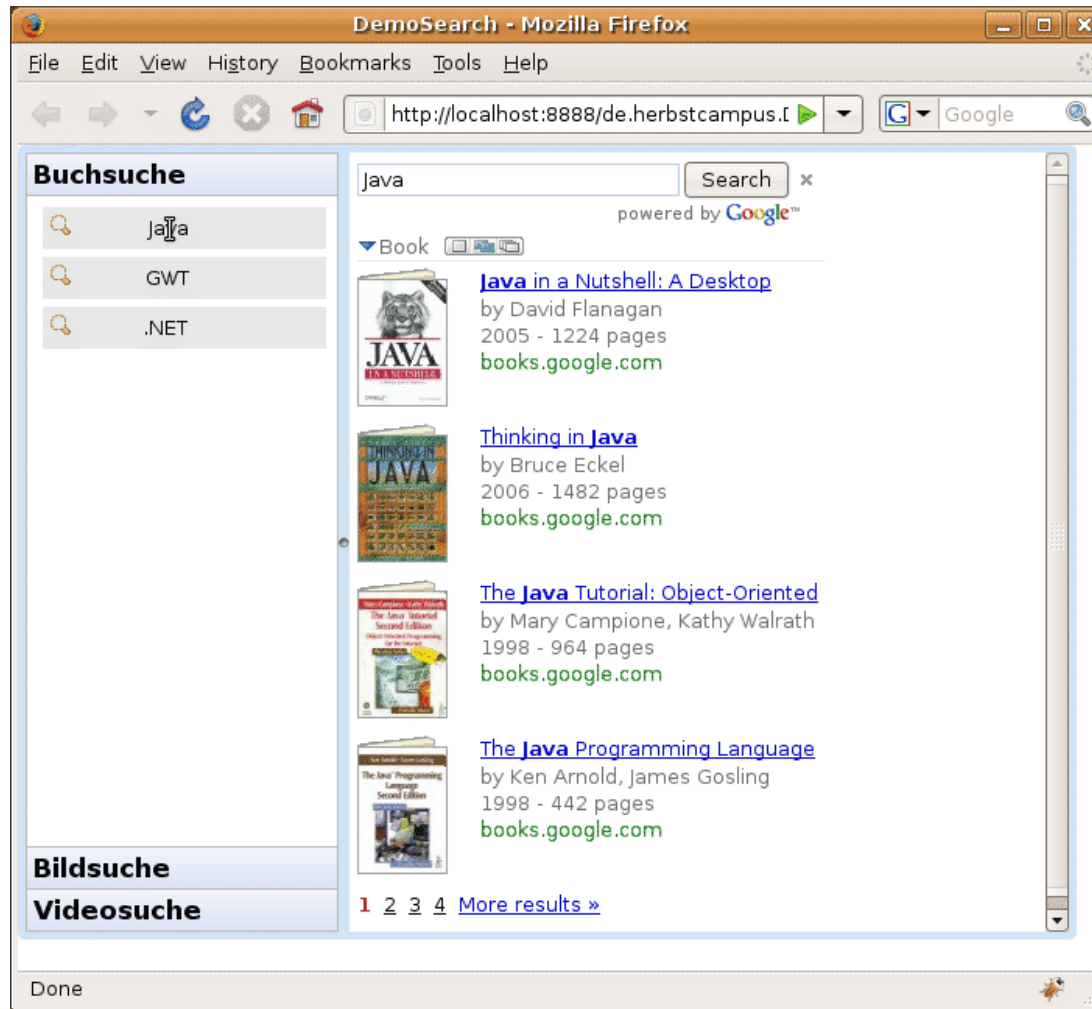
Ausführen (zum Dritten)



SearchLabel (2)

```
27      ClickListener searchClickListener = new ClickListener() {
28
29          public void onClick(Widget sender) {
30              SearchControlOptions options = new SearchControlOptions();
31              options.add(search, ExpandMode.OPEN);
32
33              SearchControl searchControl = new SearchControl(options);
34              searchControl.execute(query);
35
36              parent.updateResults(searchControl);
37          }
38
39      };
40
41      queryLabel.addClickListener(searchClickListener);
42      searchImage.addClickListener(searchClickListener);
```

Ausführen (zum Letzten)



15.–18.09.2008
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Adrian Bürki

Centris AG, Schweiz

Referent

Firma

Centris AG - more than IT

<http://www.centrisag.ch/>

Die Centris AG ist ein führender Anbieter von IT-Lösungen auf dem Schweizer Markt der Kranken- und Unfallversicherungen.

Privat

<http://illfounded.net>

GWT Projekte

<http://traceurl.com>

<http://code.google.com/p/gwtai/>