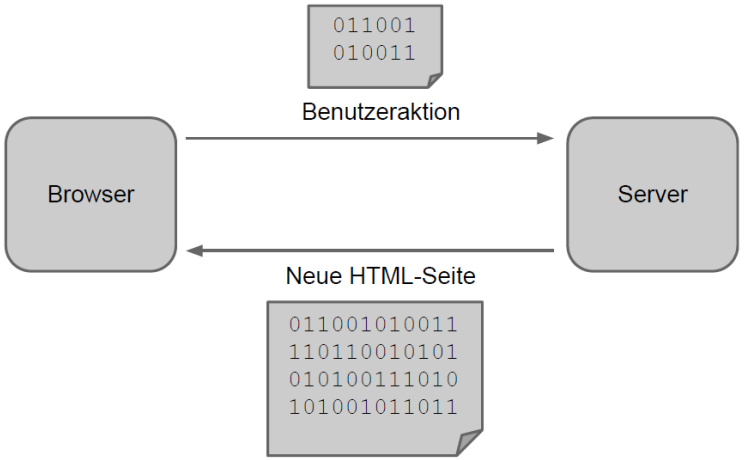
ISW: Software Engineering WS 2015/16

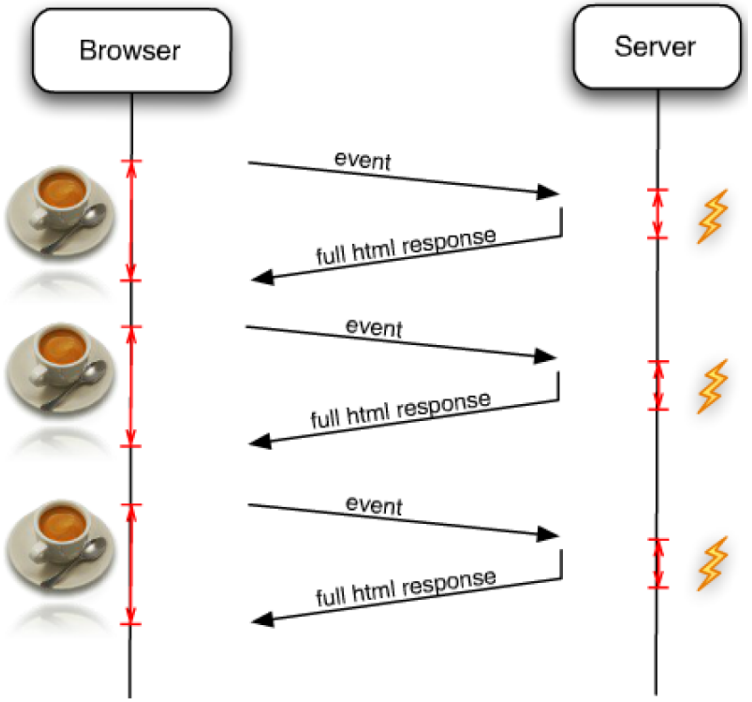
Einführung Google Web Toolkit

Marcus Seiler
Institute of Computer Science
Chair of Software Engineering
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
marcus.seiler@informatik.uni-heidelberg.de

Motivation

- Klassische HTML-Applikationen



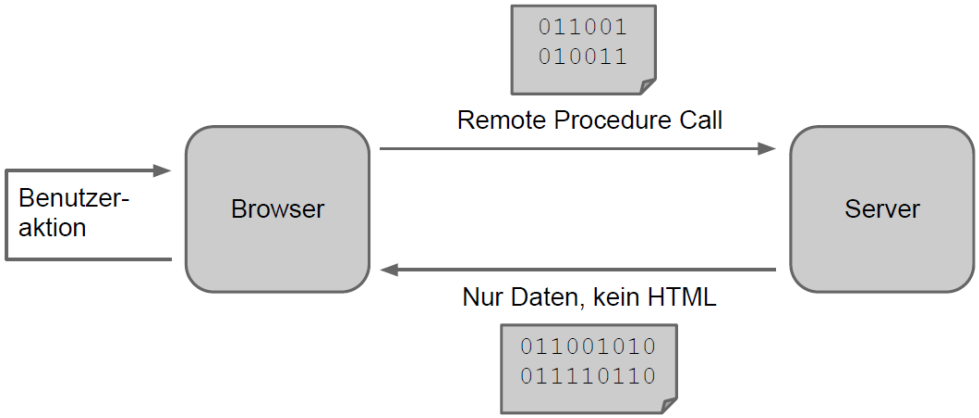


- Nachteile
  - Lange Wartezeiten für NutzerInnen
  - Jeweils komplettes Laden der ganzen Internet-Seite

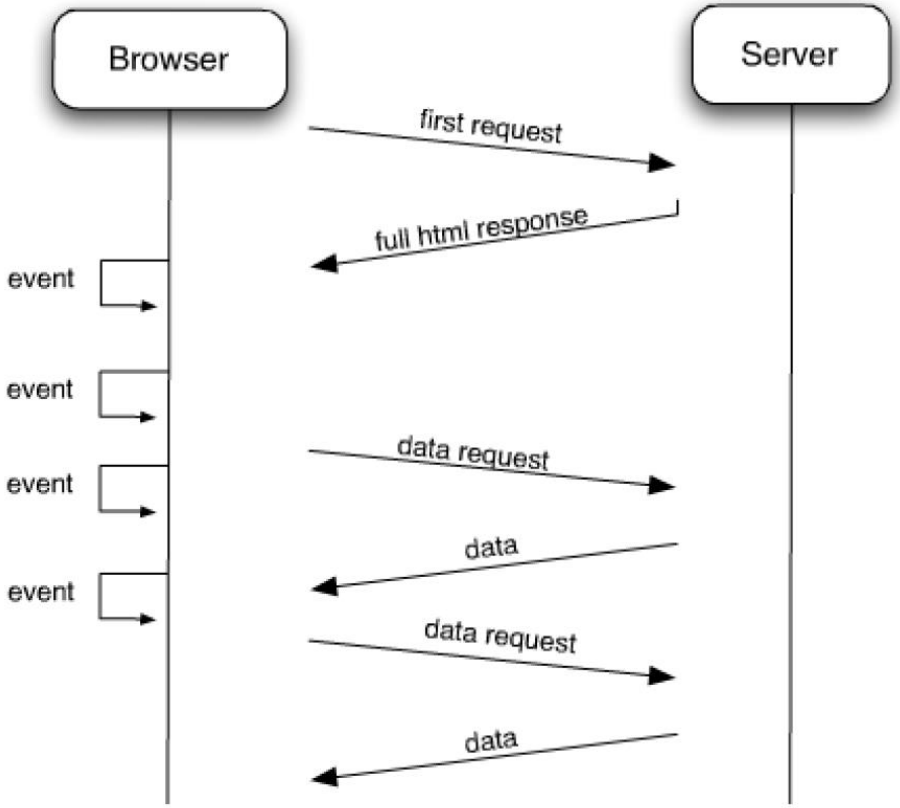
Motivation

- Moderne HTML-Applikationen
  - Statische Webseiten werden durch „User-Generated-Content“ ersetzt
    - NutzerInnen erstellen, bearbeiten und verteilen Inhalte selbst
    - NutzerInnen werden unterstützt von interaktiven Anwendungen

  → Rich-Internet-Application
  - Realisiert durch HTML, CSS, JavaScript und AJAX

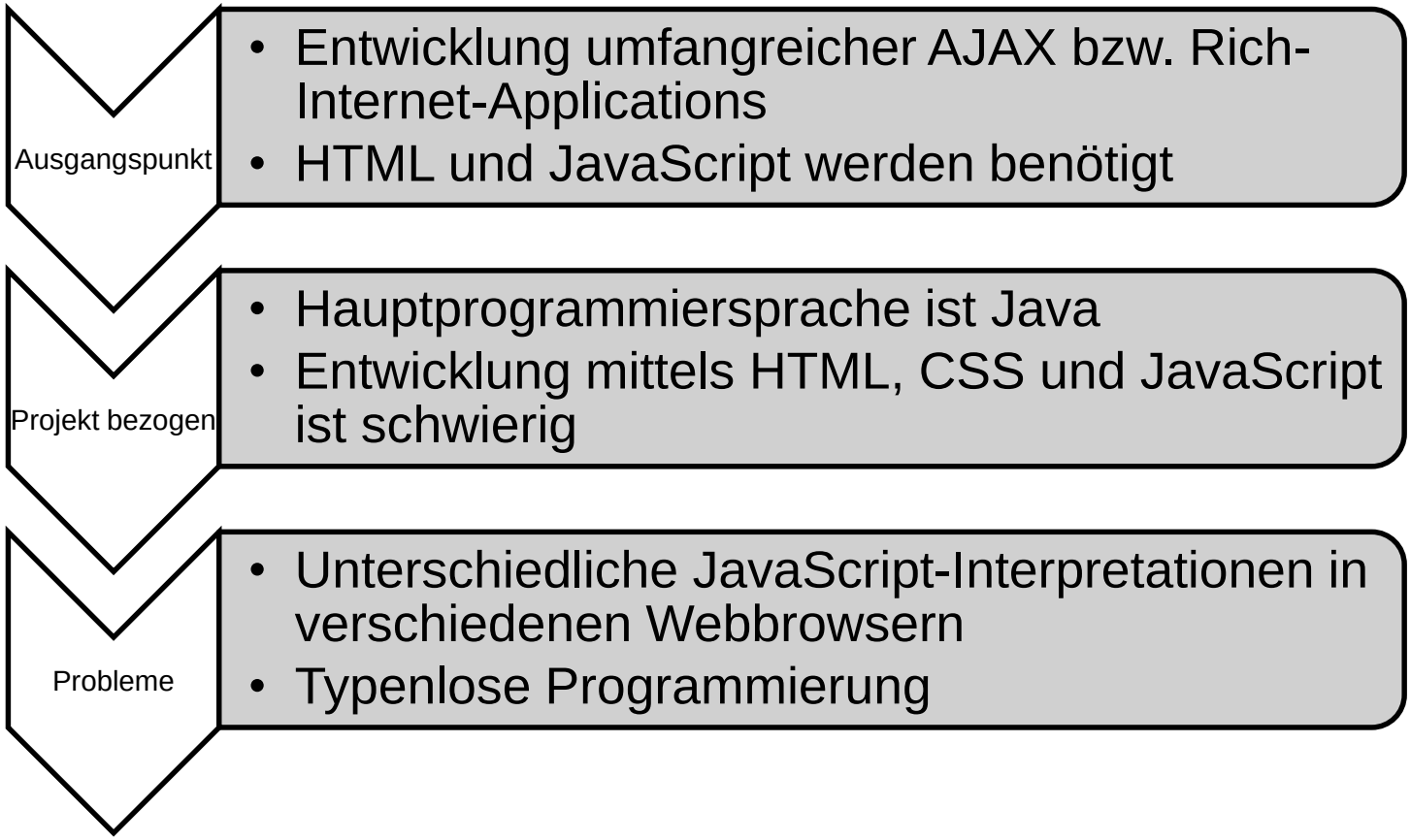


Motivation - AJAX



- AJAX = Asynchronous JavaScript and XML
- Asynchrone Datenübertragung durch XML oder JSON
- Benutzeraktionen, wie Validieren von Daten, Navigieren zwischen Elementen auf der Webseite können im Client übernommen werden

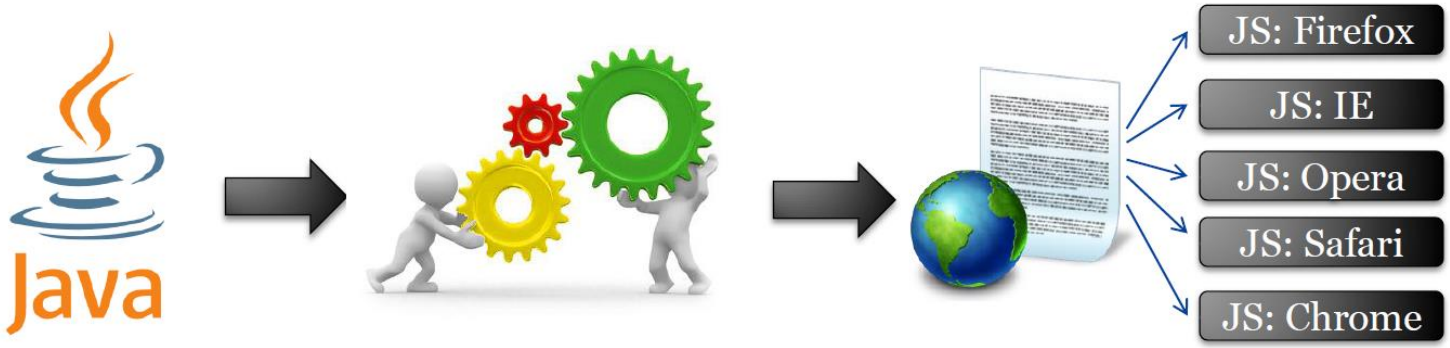
Motivation – Warum GWT?



Lösung: Google Web Toolkit

Was ist GWT?

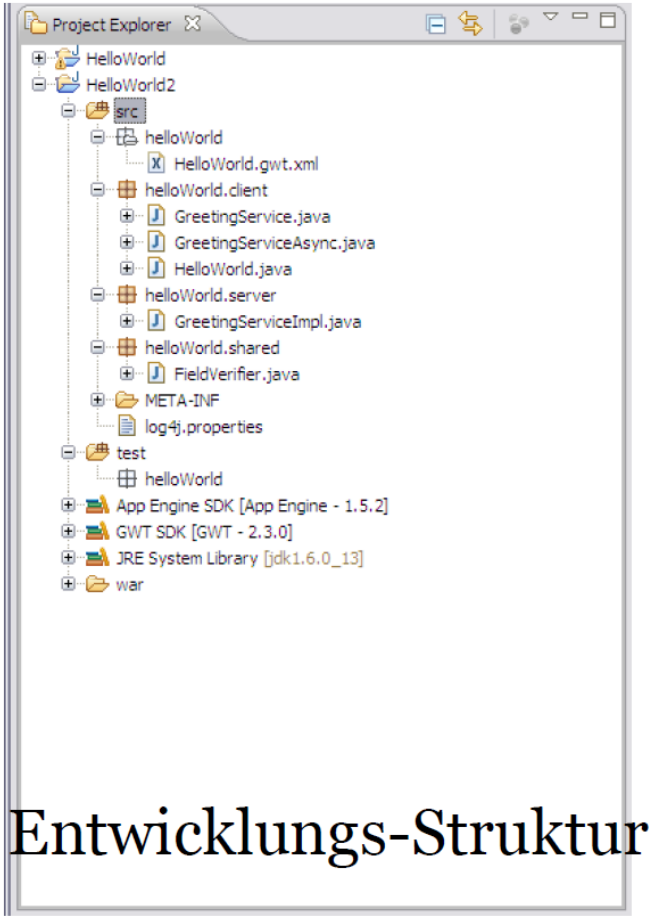
- Werkzeugkasten zur Erstellung von komplexen und interaktiven Desktop-ähnlichen AJAX-Applikationen
- Programmierung von Client und Server in Java
- GWT-Compiler übersetzt clientseitigen Java-Code zu HTML, CSS und JavaScript



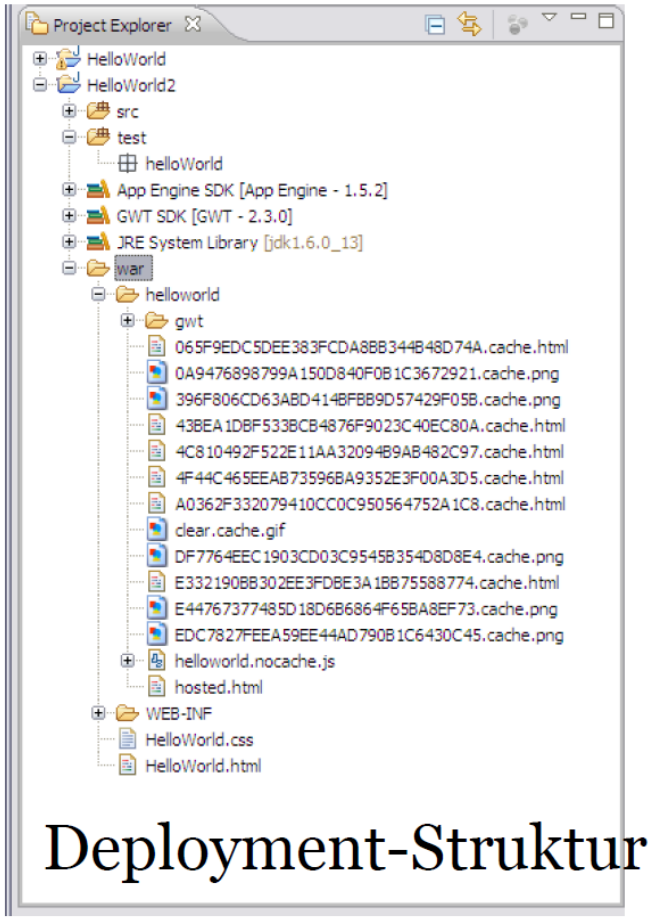
GWT Features

- Wird unterstützt von Entwicklungsumgebungen, z.B. Eclipse
- Umfangreiche Widgetbibliothek
- Unit-Testing mittels JUnit ist möglich
- Debuggen des clientseitigen Codes ist möglich
- JavaScript Native Interface (JSNI)
- Remote Procedure Calls (RPC)

GWT Projektstruktur



!=



Entwicklungs-Struktur vs. Deployment-Struktur

HTML-Grundgerüst

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">

    <link type="text/css" rel="stylesheet" href="BspProjekt.css">

    <title>Web Application Starter Project</title>

    <script type="text/javascript" language="javascript" src="bspprojekt/bspprojekt.nocache.js"></script>
  </head>

  <body>

      <!-- OPTIONAL: include this if you want history support -->
      <iframe src="javascript:''" id="__gwt_historyFrame" style="width:0;height:0;border:0"></iframe>

      <h1>ErstesBeispielEclipse</h1>

      <p>
          This is an example ...
      </p>
      <table align=center>
          <tr>
              <td id="slot1"></td><td id="slot2"></td>
          </tr>
      </table>
  </body>
</html>
```

Java-Grundgerüst

```
import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.event.dom.client.ClickEvent;
import com.google.gwt.event.dom.client.ClickHandler;
import com.google.gwt.user.client.ui.Button;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.RootPanel;

public class HelloWorld implements EntryPoint {

    public void onModuleLoad() {
        final Button button = new Button("Click me");
        final Label label = new Label();

        button.addClickHandler(new ClickHandler() {
        @Override
        public void onClick(ClickEvent event) {
            if (label.getText().equals(""))
                label.setText("Hello World!");
              else
                label.setText("");
            }
        });

        RootPanel.get("slot1").add(button);
        RootPanel.get("slot2").add(label);
    }
}
```

GWT-Konfiguration über Module

- Module bestehen aus
  - Client-Code, optional auch Server-Code
  - Mindestens einer HTML-Datei
  - Andere statische Ressourcen (Bilder, CSS-Dateien)

```
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to='helloworld'>
  <!-- Inherit the core Web Toolkit stuff.                  -->
  <inherits name='com.google.gwt.user.User'/>

  <!-- Inherit the default GWT style sheet.                 -->
  <inherits name='com.google.gwt.user.theme.clean.Clean'/>

  <!-- Other module inherits                                -->

  <!-- Specify the app entry point class.                   -->
  <entry-point class='helloWorld.client.HelloWorld'/>

  <!-- Specify the paths for translatable code              -->
  <source path='client'/>
  <source path='shared'/>

</module>
```

Entwicklung mit GWT

- Entwicklungsmodus
  - Applikation wird als Bytecode in der Java Virtual Machine (JVM) ausgeführt
  - Webbrowser ruft den Java-Code auf
  - Debugging kann genutzt werden
  - Kompiliert sehr schnell (Java → Java Bytecode)
  - Die meiste Zeit der Entwicklung findet in diesem Modus statt
- Webmodus
  - Clientseitiger Code wird von dem GWT-Compiler in JavaScript, CSS und HTML kompiliert
  - Optimiert für produktiven Einsatz
  - Kompilieren dauert länger (Java → JavaScript pro Konfiguration)
  - Keine JVM oder Browser-Plug-Ins notwendig

GWT Live-Demo

- Live-Demo zum Mitmachen
  - Installation GWT
  - Webanwendung für Bücher
    - 03-gwt-book-example.zip aus Moodle herunterladen

GWT Ressourcen

- Tutorials und Arbeitsblätter
  - Installation und Erste Schritte mit GWT
  - Arbeitsblatt GWT

- Offizielle GWT Dokumentation & Tutorials von Google
  - http://www.gwtproject.org/doc/latest/tutorial/
  - http://www.gwtproject.org/doc/latest/DevGuide.html
  - http://www.gwtproject.org/javadoc/latest/index.html

- GWT-Showcase & Widget Gallerie
  - Beispiele und Quelltext zu allen Widgets
  - http://www.gwtproject.org/doc/latest/RefWidgetGallery.html
  - http://samples.gwtproject.org/samples/Showcase/Showcase.html

Marcus Seiler
Institute of Computer Science
Chair of Software Engineering
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
marcus.seiler@informatik.uni-heidelberg.de

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG