

Fortgeschrittene GWT Best Practices

Agenda

Kurzer Einstieg in GWT

Methoden zur Client-Server-Kommunikation
in GWT

Continuous Integration und Continuous Delivery
im GWT-Umfeld

Kurzer Einstieg in GWT

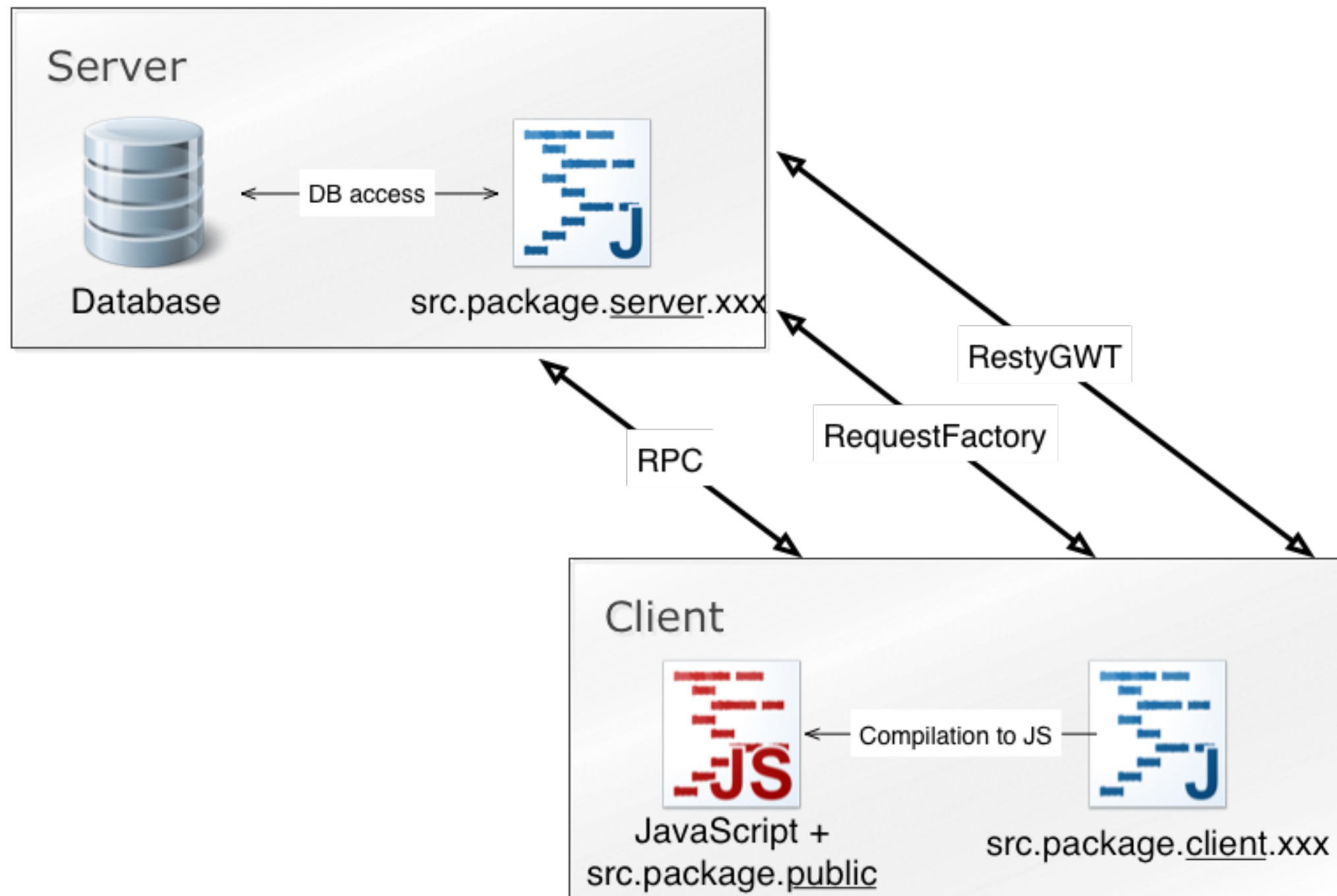
- Google Web Toolkit
- Entwicklung von Rich Internet Applications (Web Anwendungen)
- Aktuelle Version 2.6.1 (2.7. in den Startlöchern)
- Doku unter www.gwtproject.org

Kurzer Einstieg in GWT

Das besondere an GWT

- Entwicklung mit reinem Java
- Cross Compiling (Java -> JavaScript)
- Open Source, unabhängig von Google

Kurzer Einstieg in GWT



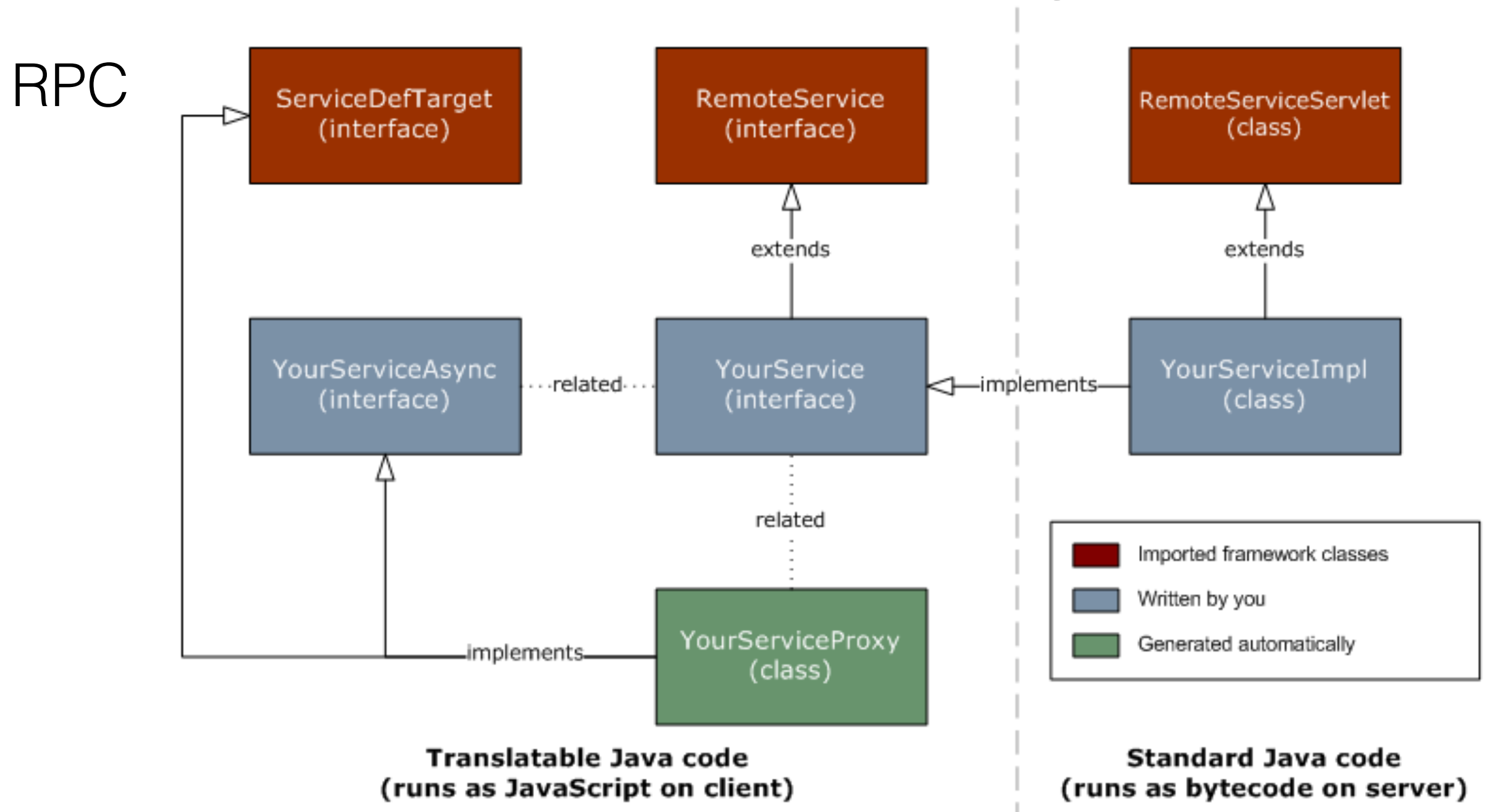
Agenda

Kurzer Einstieg in GWT

Methoden zur Client-Server-Kommunikation in GWT

Continuous Integration und Continuous Delivery
im GWT-Umfeld

Methoden zur Client-Server-Kommunikation in GWT



Quelle: <https://code.google.com/p/google-web-toolkit-doc-1-5/wiki/DevGuidePlumbingDiagram>

Methoden zur Client-Server-Kommunikation in GWT

RequestFactory

- Entity
- Deklarriere EntityProxy<interface> oder ValueProxy<interface>, das zur serverseitigen Entity passt
- Implementiere RequestFactory<interface> um Server Requests absenden zu können
- ohne CRUD-Code in Entity: Implementiere & deklariere Locator, ServiceLocator <interface>
- erstelle Service Methoden für die Entity
- Benutze EventBus von GWT
- Registriere RequestFactoryServlet in web.xml
- Schreibe Code um Requests zu instanziiieren und für die Behandlung der Response

Enabling RequestFactory the simple way

The title of this section might imply that RequestFactory is simple, but that probably isn't the case.

Quelle: GWT in Action 2nd Edition, Seite 240

Methoden zur Client-Server-Kommunikation in GWT

RestyGWT

```
@Path("path/to/your/rest/resource")
public interface MyService extends RestService
{
    @GET
    @Path("{id}")
    void doXYZ(@PathParam("id") String id, MethodCallback<List<XYZ> callback);
}
```

```
MyService restService = GWT.create(MyService.class);
restService.doXYZ("4711", new MethodCallback<List<XYZ>>()
{
    public void onSuccess(Method method, List<XYZ> response)
    {
        ...
    }

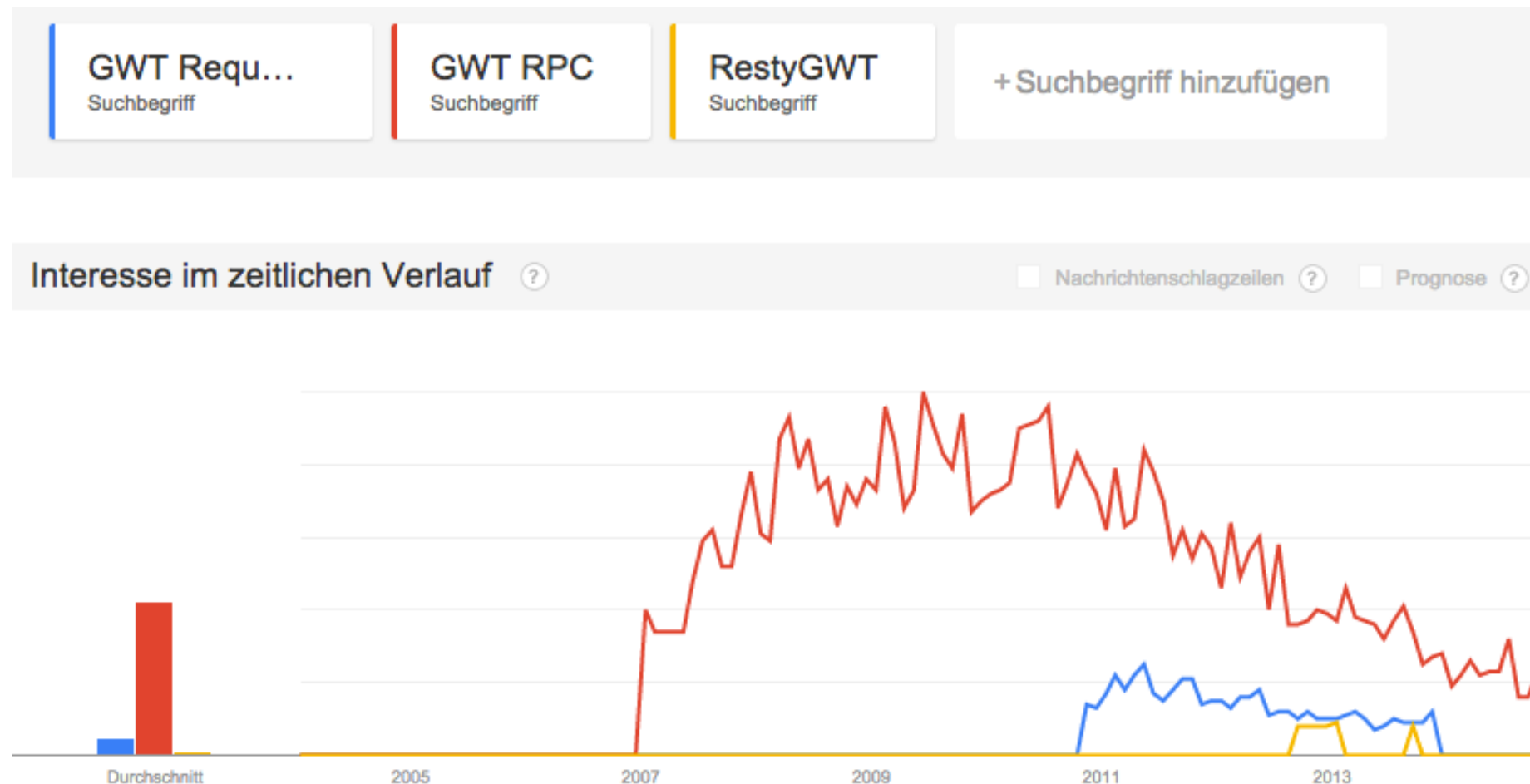
    public void onFailure(Method method, Throwable exception)
    {
        ...
    }
});
```

Methoden zur Client-Server-Kommunikation in GWT

	Pro	Kontra
RPC	<ul style="list-style-type: none">- einfach- gut dokumentiert	<ul style="list-style-type: none">- Boilerplate Code- führt meist zu DTA-Pattern- Entities nicht im Client nutzbar- proprietär
RequestFactory	<ul style="list-style-type: none">- „Standard“ GWT Kommunikationstechnik- Integration mit Editor Framework- Delta-Daten Transfer	<ul style="list-style-type: none">- hohe Lernkurve- schlechte Dokumentation- Boilerplate Code- proprietär- viele zusätzliche Bibliotheken
RestyGWT	<ul style="list-style-type: none">- einfach- gut dokumentiert- Entities sind 1:1 im Client verwendbar- REST-Standard & JSON- andere Endgeräte/ Technologien leicht	<ul style="list-style-type: none">- (noch) kein fester Bestandteil von GWT

Methoden zur Client-Server-Kommunikation in GWT

- Diskussion „RPC vs. RequestFactory vs. RestyGWT“ unter <https://groups.google.com/forum/m/#!topic/google-web-toolkit/Dj7Z0GBIFuY>
- Entwicklung bei Google Trends (Stand 2014-10-16)



Methoden zur Client-Server-Kommunikation in GWT

Fazit

Methoden zur Client-Server-Kommunikation in GWT

Diskussionsbedarf?
Fragen?

Agenda

Kurzer Einstieg in GWT

Methoden zur Client-Server-Kommunikation
in GWT

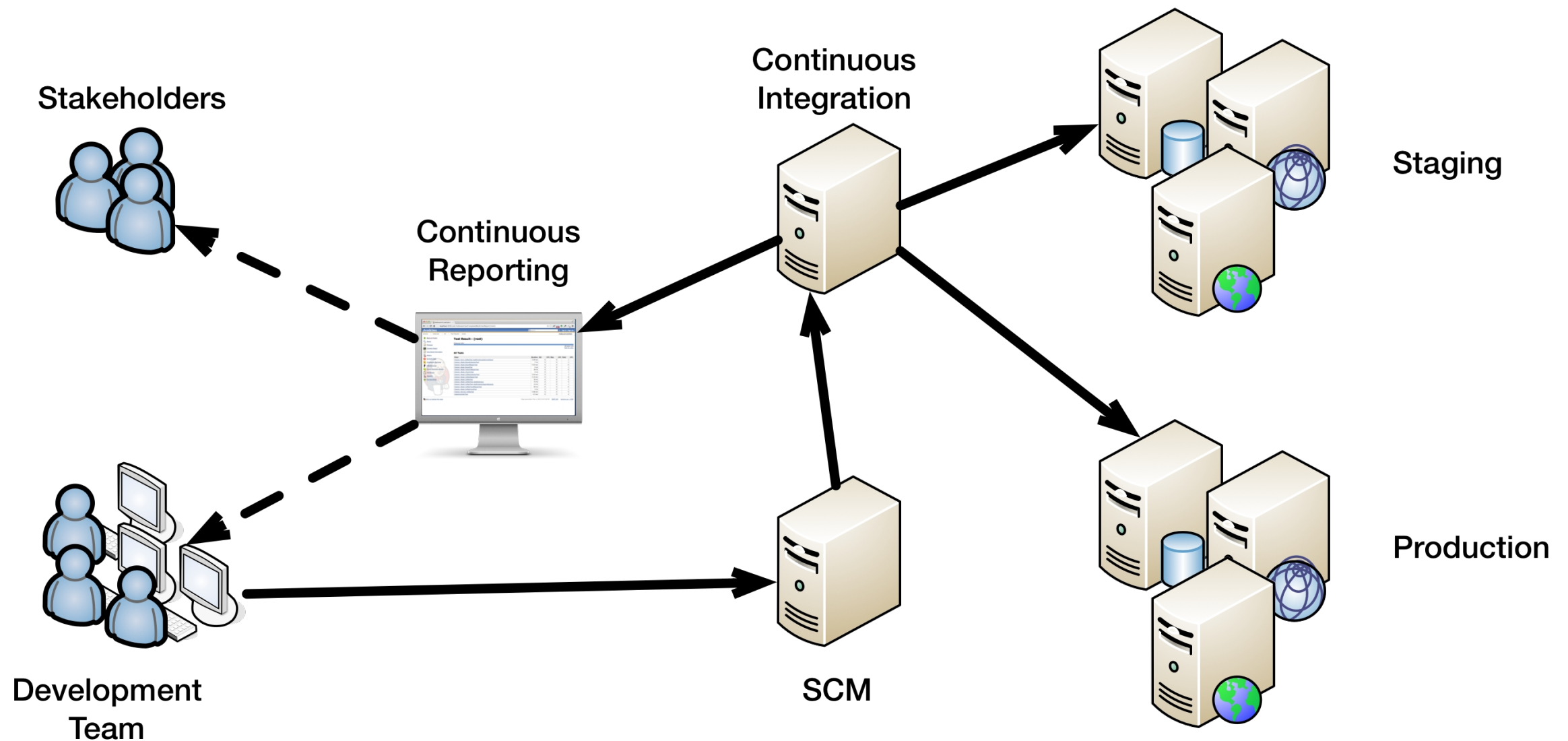
**Continuous Integration und Continuous
Delivery im GWT-Umfeld**

Continuous Integration und Continuous Delivery im GWT-Umfeld

Unterschiede zu (Java-) Web Anwendungen

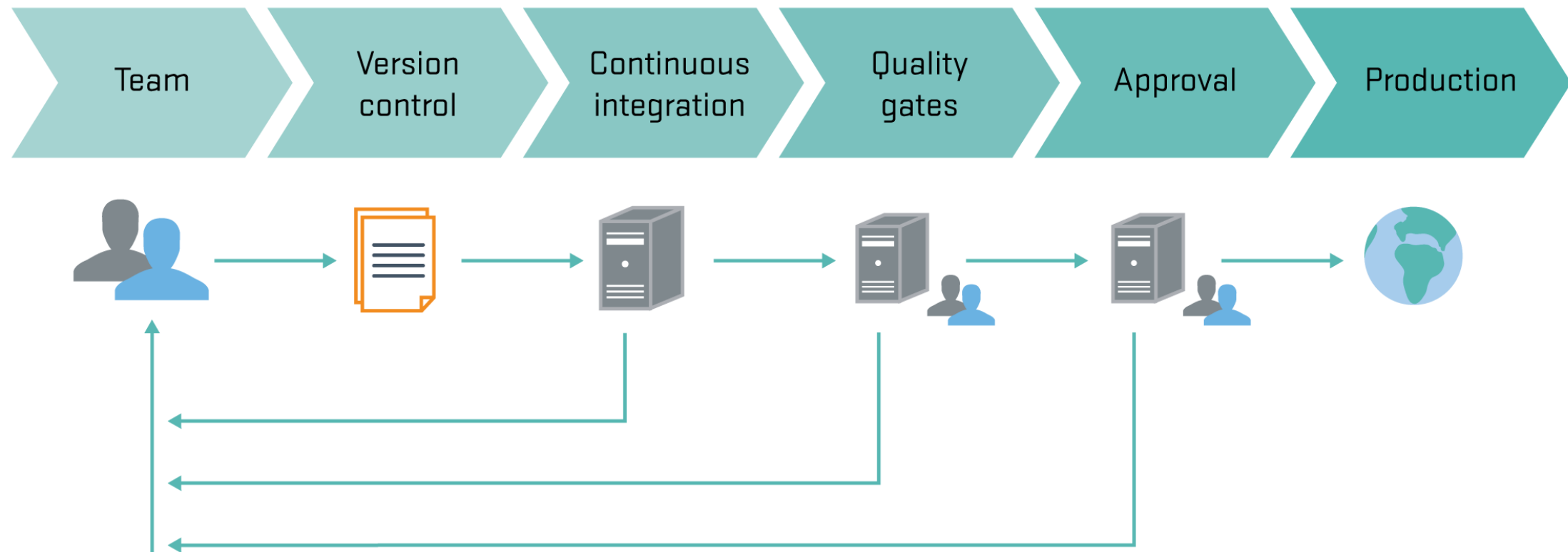
- Artefakte des GWT-Compiler:
 - Java → JavaScript
 - Grafiken, HTML-/CSS-Artefakte, ...
- Test: Testen GUI-Code (Client) → anderes Thema: GWTTestCase, GWTMockito, ...

Continuous Integration Workflow



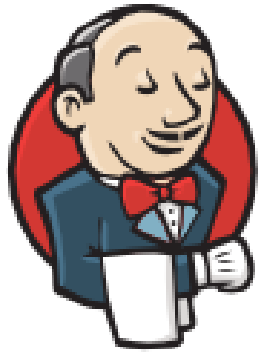
Quelle: <http://www.in2it.be/php-consultancy/continuous-deployment/>

Continuous Deployment



Quelle: <https://www.getchef.com/solutions/continuous-delivery/>

CI-Engine & Build-Tool



Jenkins



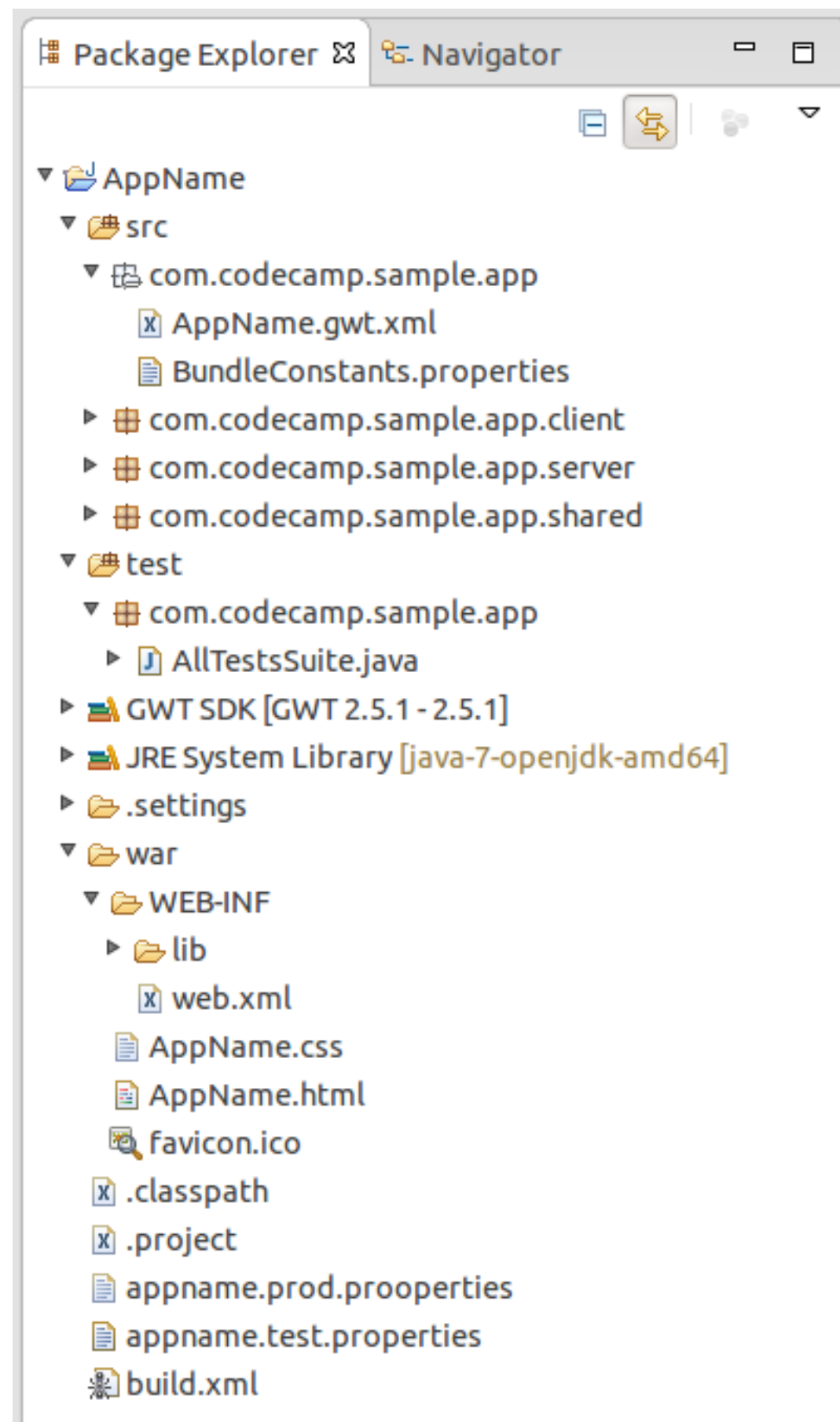
<https://blog.eveoh.nl/2012/01/using-google-web-toolkit-with-gradle/>

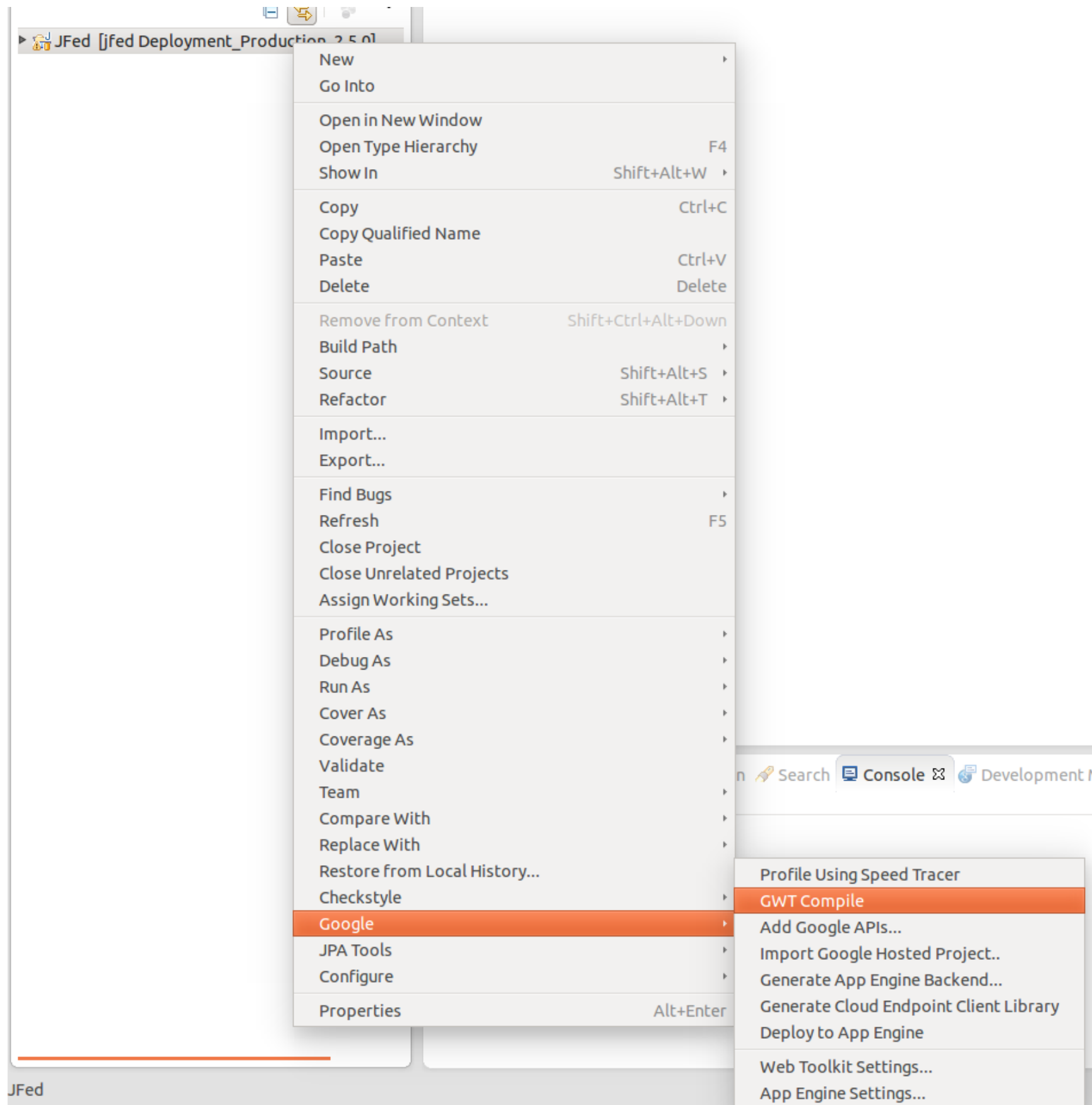


<http://www.communardo.de/home/techblog/2012/09/30/howto-gwt-eclipse-maven/>

<http://kent.spillner.org/blog/work/2009/11/14/java-build-tools.html>

Quellen: <http://jenkins-ci.org/>
<http://ant.apache.org/>





Ant: Compile (Server)

```
<property name="src.dir" location="src"/>
<property name="test.dir" location="test"/>
<property name="build.dir" location="war/WEB-INF/classes"/>
<property name="lib.dir" location="war/WEB-INF/lib"/>
<property name="gwtc.lib" location="/opt/gwt-2.5.1"/>

<path id="project.classpath">
  <fileset dir="${lib.dir}">
    <include name="*.jar"/>
  </fileset>
  <fileset dir="${gwtc.lib}">
    <include name="*.jar"/>
  </fileset>
  <pathelement location="${src.dir}"/>
  <pathelement location="${build.dir}"/>
</path>

<target name="compile">
  <echo message="Compiling Java Sources..." />
  <javac srcdir="${src.dir}" destdir="${build.dir}">
    <classpath refid="project.classpath"/>
  </javac>
</target>

<target name="compileTests" depends="compile">
  <echo message="Compiling Java Test Sources..." />
  <javac srcdir="${test.dir}" destdir="${build.dir}">
    <classpath refid="project.classpath"/>
  </javac>
</target>
```

Ant: Test (Server)

```
<property name="test.reports" value="reports" />

<!-- libs für Entwicklung (JUnit, ...) -->
<path id="java.class.path">
  <fileset dir="/usr/share/java" includes="*.jar" />
</path>

<target name="test" depends="compile, compileTest" description="Execute Junit-
Tests...">
  <junit printsummary="yes" fork="yes">
    <classpath refid="project.classpath"/>
    <classpath refid="java.class.path" />
    <formatter type="xml" />
    <batchtest todir="${test.reports}">
      <fileset dir="${build.dir}">
        <include name="**/AllTestsSuite.java" />
      </fileset>
    </batchtest>
  </junit>
  <echo message="Testergebnisse im Verzeichnis ${test.reports}" />
</target>
```

Ant: Compile (Client)

```
<property name="gwt.build.dir.1" location="war/jfed"/>
<property name="gwt.build.dir.2" location="war/WEB-INF/deploy"/>
<property name="gwt.module.name" value="com.codecamp.sample.app.AppName"/>

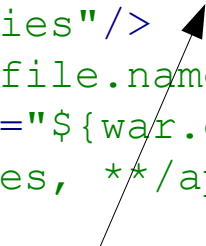
<target name="cleanGwtCompile">
  <echo message="Cleaning GWT classes..." />
  <delete includeemptydirs="true">
    <fileset dir="${gwt.build.dir.1}" includes="**/*"/>
  </delete>
  <delete includeemptydirs="true">
    <fileset dir="${gwt.build.dir.2}" includes="**/*"/>
  </delete>
</target>

<target name="gwtCompile" depends="compile, cleanGwtCompile">
  <echo message="Compiling GWT Sources..." />
  <java failonerror="true" fork="true" classname="com.google.gwt.dev.Compiler">
    <classpath>
      <path refid="project.classpath" />
    </classpath>
    <jvmarg line="-Xmx1024M"/>
    <arg value="${gwt.module.name}" />
  </java>
</target>
```

Ant: Build war

```
<property name="war.dir" location="war"/>
<property name="war.file.name" value="AppName.war"/>

<target name="buildWarTest" depends="gwtCompile">
  <echo message="Build war (test) for version ${buildVersion}..." />
  <echo message="Copying file appname.test.properties..." />
  <copy overwrite="true" file="appname.test.properties"
    tofile="${war.dir}/appname.properties"/>
  <echo message="Zipping war dir to ${war.file.name}..." />
  <zip destfile="${war.file.name}" basedir="${war.dir}"
    excludes="**/appname .test.properties, **/appname .prod.properties"/>
</target>
```



Umgebungs-spezifische Konfiguration (Staging)

Ant: Deploy war

```
<property name="archive.dir" location="/home/tomcat7/jfed"/>
<property name="webapps.dir" location="/var/lib/tomcat7/webapps"/>

<property file=
    "${src.dir}/com/codecamp/sample/appname/BundleConstants.properties" />

<target name="deployTest" depends="buildWarTest">
  <echo message="Deploying (test) version ${buildVersion}..." />
  <echo message="Archiving war file ${war.file.name} to
    ${archive.dir}/jfed_${buildVersion}..." />
  <mkdir dir="${archive.dir}/jfed_${buildVersion}"/>
  <copy overwrite="true" file="${war.file.name}"
    todir="${archive.dir}/jfed_${buildVersion}"/>
  <echo message="Delpoying war file ${war.file.name} to ${webapps.dir}..." />
  <copy overwrite="true" file="${war.file.name}" todir="${webapps.dir}"/>
</target>
```

aus Properties-Datei

Jenkins: Deploy Jobs

Übersicht [Jenkins] - Mozilla Firefox

Übersicht [Jenkins]

109.239.48.137:8080/jenkins/

Google

Webmail Google Kalender Google Kontakte tagesschau.de DIE WELT Wikipedia XING Facebook Google+ andrejanus NASHorn abschalten...

Jenkins

Suchen

André Janus | Abmelden

Jenkins

AUTO-AKTUALISIERUNG EINSCHALTEN

Beschreibung hinzufügen

Neuen Item anlegen

Benutzer

Build-Verlauf

Jenkins verwalten

Zugangsdaten

Meine Ansichten

Build Warteschlange

Keine Builds geplant

Build-Prozessor-Status

#	Status
1	Ruhend
2	Ruhend

S	W	Name ↓	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer
		JFed-deploy-test	2 Monate 25 Tage - #50	6 Monate 9 Tage - #36	6 Minuten 24 Sekunden
		JFed-preDeploy-prod	2 Monate 25 Tage - #10	6 Monate 19 Tage - #6	6 Minuten 42 Sekunden
		jqdv-test	1 Monat 18 Tage - #25	3 Monate 7 Tage - #17	3 Minuten 1 Sekunde
		sepa-test	20 Tage - #101	8 Monate 1 Tag - #30	20 Sekunden
		VMAp-deploy-test	Nicht anwendbar	Unbekannt	Nicht anwendbar
		VMAp-test	19 Tage - #156	19 Tage - #154	1 Minute 42 Sekunden

Symbol: [S](#) [M](#) [L](#)

Legende RSS Alle Builds RSS Nur Fehlschläge RSS Nur jeweils letzter Build

Hilf uns, diese Seite zu lokalisieren.

Erstelldatum dieser Seite: 21.10.2014 12:10:28 [REST API](#) [Jenkins ver. 1.549](#)

Continuous Integration und Continuous Delivery im GWT- Umfeld

Fazit

Continuous Integration und Continuous Delivery im GWT- Umfeld

Diskussionsbedarf?
Fragen?

Vielen Dank!



Dr. André Janus

www.xing.com/profile/Andre_Janus



Ralf Klemmer

www.xing.com/profile/Ralf_Klemmer2



www.codecamp-gbr.de