

CAS Software AG

Richtig informiert. Jederzeit und überall.



Erfahrungsbericht Anwendungsentwicklung mit RAP

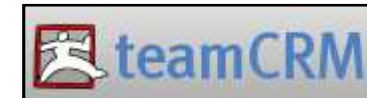
Frank Appel, Innoopract Informationssystem GmbH
Stefan Röck, CAS Software AG



Agenda

- Vorstellung teamCRM
- RAP-Framework Basics
- Custom Widgets
- Integration mit Browser-Addins
- Deployment

- teamCRM



- CRM-Anwendung für kleine und mittelständische Unternehmen
- Vertrieb: Software as a service (SaaS)
- Zentrales Hosting
- Mandantenfähig
- CRM-Funktionalitäten
 - Adressmanagement
 - Termine+Aufgaben
 - Kalender
 - E-Mail
 - Verkaufschancen
 - Reports, ...

Meilensteine

- Dez 2006: Projektbeginn
 - Technologieauswahl: GWT vs. RAP
 - PRO GWT
 - Relativ weit fortgeschritten
 - „Community“, Widgets, Libs
 - PRO RAP
 - Bewährte Konzepte (SWT, JFace) → Dokumentation
 - Ansprechendes Layout
 - Erweiterbarkeit durch komponentenorientierte Laufzeitumgebung
- Mär 2007: Start der Entwicklung
 - Agiles Vorgehen (SCRUM)
 - Regelmäßige Abstimmung mit RAP-Team von Innoopract
- Mär 2008: Erste öffentliche Demo auf der Cebit
- Aug 2008: Release v1

Demo



The screenshot displays the teamCRM web application interface. At the top, the user is logged in as Britta Glatt on 24.06.08 at 14:05. The main window shows the contact profile for Dr. Graziella Balducci, including her photo, name, company (Westautomatik GmbH), and address. The interface is divided into several sections: a left sidebar with navigation options like 'Kontakte', 'Kalender', and 'E-Mails'; a top toolbar with actions like 'Speichern', 'Drucken', and 'E-Mail senden'; and a main content area with tabs for 'Allgemein', 'Details', 'Lageplan', 'Kampagnen', 'Akte', and 'Journal'. The contact details are organized into sections: 'Person' (Name, Firma, Abteilung, Funktion, Kategorie, Klassifikation), 'Anschriften' (Geschäftlich, Besuch, Privat), 'Kontakt' (Telefon, Mobil, Sekretär(in), E-Mail, Homepage, Fax, Fax PC), and 'Notizen'. A search bar is visible at the top, and a list of recent searches is shown in the sidebar.

Was ist RAP?

Plug-ins, Plug-ins, Plug-ins...

OSGi spezifiziert ein dynamisches Komponentenmodell:

1. Module – Kapselung und Deklaration von Abhängigkeiten
2. Life Cycle – API für Lifecycle Management
3. Service Registry – Bereitstellung von Funktionalität
4. Security layer – Einschränken von Funktionalität

Eclipse stellt mit Equinox eine OSGi Implementierung zur Verfügung

Was ist RAP?

Bereitstellung eines leistungsstarken UI Konzepts

1. Standard Widget Toolkit (SWT)

- stellt der Eclipse Plattform native Widget Funktionalität betriebsystemunabhängig zur Verfügung

2. JFace

- stellt Klassen zur Verfügung, die Routineaufgaben der Oberflächen Programmierung vereinfachen

3. Workbench

- ist verantwortlich für die Präsentation und Koordination der Benutzeroberfläche

Was ist RAP?

The screenshot displays the RAP Showcase application window. The interface is annotated with several labels:

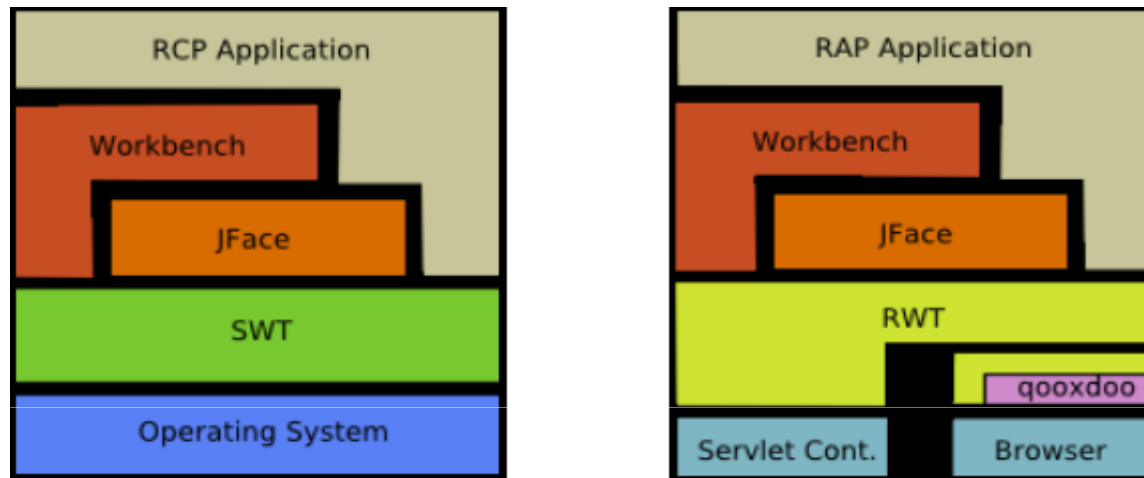
- Menu Bar:** Located at the top left, containing 'User' and 'Help' options.
- Workbench Window:** The main application frame.
- Workbench Parts:** A red box highlighting the top navigation area with tabs for 'Map', 'Talk Details', and 'Intro'.
- Workbench Page:** A purple box highlighting the main content area.
- Cool Bar:** A yellow box highlighting the left-hand navigation pane, which lists tracks like 'Business Track' and 'C/C++ Development Track'.
- Dialog:** A green box highlighting a 'Change User Data' dialog box. The dialog contains fields for 'Firstname' (Frank), 'Lastname' (Appel), 'Street', 'City' (Karlsruhe), 'Country' (Deutschland), 'Username' (fa), 'Password', and 'Confirm Password'. A message at the bottom states: 'The password and its confirmation are not the same.'

The background shows a map of Karlsruhe, Germany, with a red pin and a white callout bubble containing the text 'Karlsruhe, Deutschland'. The map includes street names like 'Kriegsstrasse' and 'Durlacher Allee'.

RAP – Wiederverwendung von RCP Plug-ins

1. 70% - 90% Wiederverwendung ist möglich
2. RAP implementiert nur eine Teilmenge von RCP!
3. Auslagerung inkompatiblen Codes in Fragmente
4. Anwendungen müssen Multi-User fähig werden

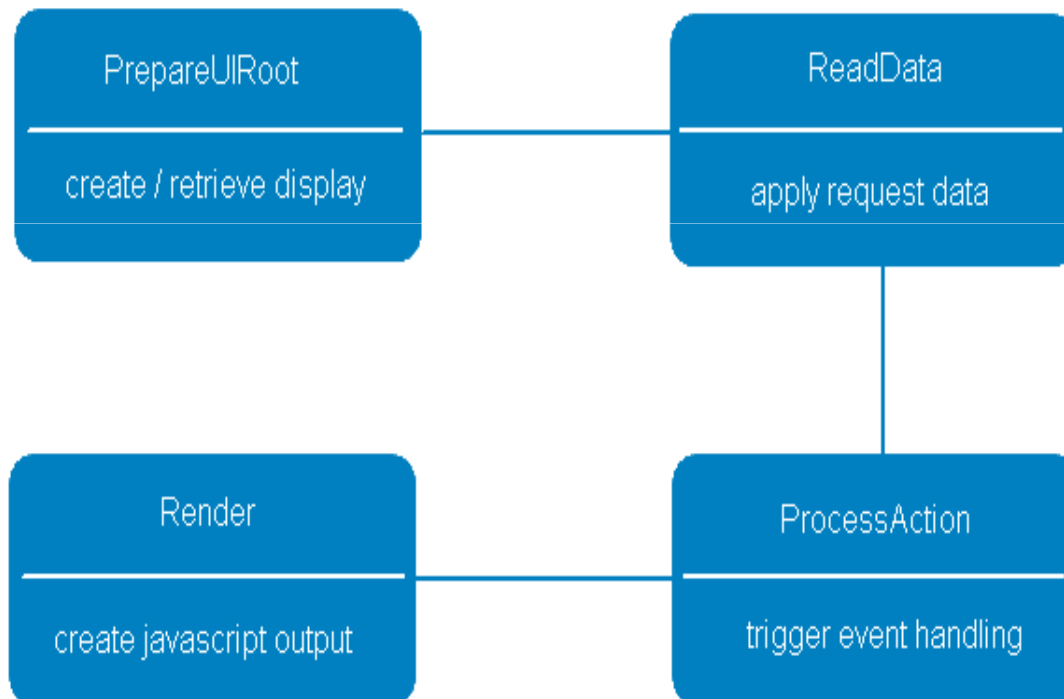
Wie funktioniert RAP?



1. SWT wird von RWT ersetzt - RWT stellt die Oberfläche im Browser dar
2. Die darüberliegenden Schichten bleiben aber größten Teils unverändert
3. RWT benutzt *qooxdoo* um Widgets im Browser darzustellen
4. Unterschiede:
 - Multi-User Umgebung: OSGi Bundles werden zwischen Sitzungen geteilt
 - Zusätzliche API für spezielle Aufgaben der Web-Programmierung

Wie funktioniert RAP?

The Request Lifecycle

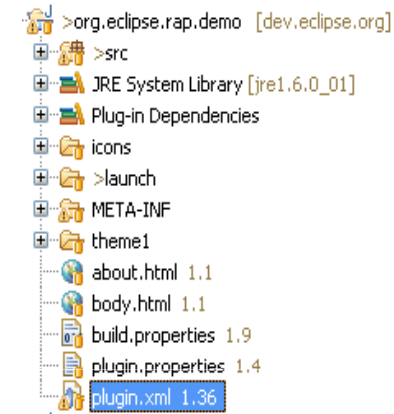


1. Am Ende der **ReadData** Phase sind alle Widget-Attribute mit dem Client synchronisiert
2. Diese Attribute werden zum späteren Vergleich gepuffert
3. Während der **ProcessAction** Phase können Änderungen der Widget-Attribute auftreten
4. Die **Render** Phase vergleicht die Attribute mit den gepufferten Werten und schickt nur das Delta an den Client

RAP aus Sicht des Entwicklers

- Der Applikationsstart beinhaltet die Ableitung oder Implementierung folgender Typen:
 - WorkbenchAdvisor
 - WorkbenchWindowAdvisor
 - ActionBarAdvisor
 - IPerspectiveFactory

- Diese sind RCP Entwicklern sehr vertraut und dienen dem gleichen Zweck wie ihre RCP Equivalente

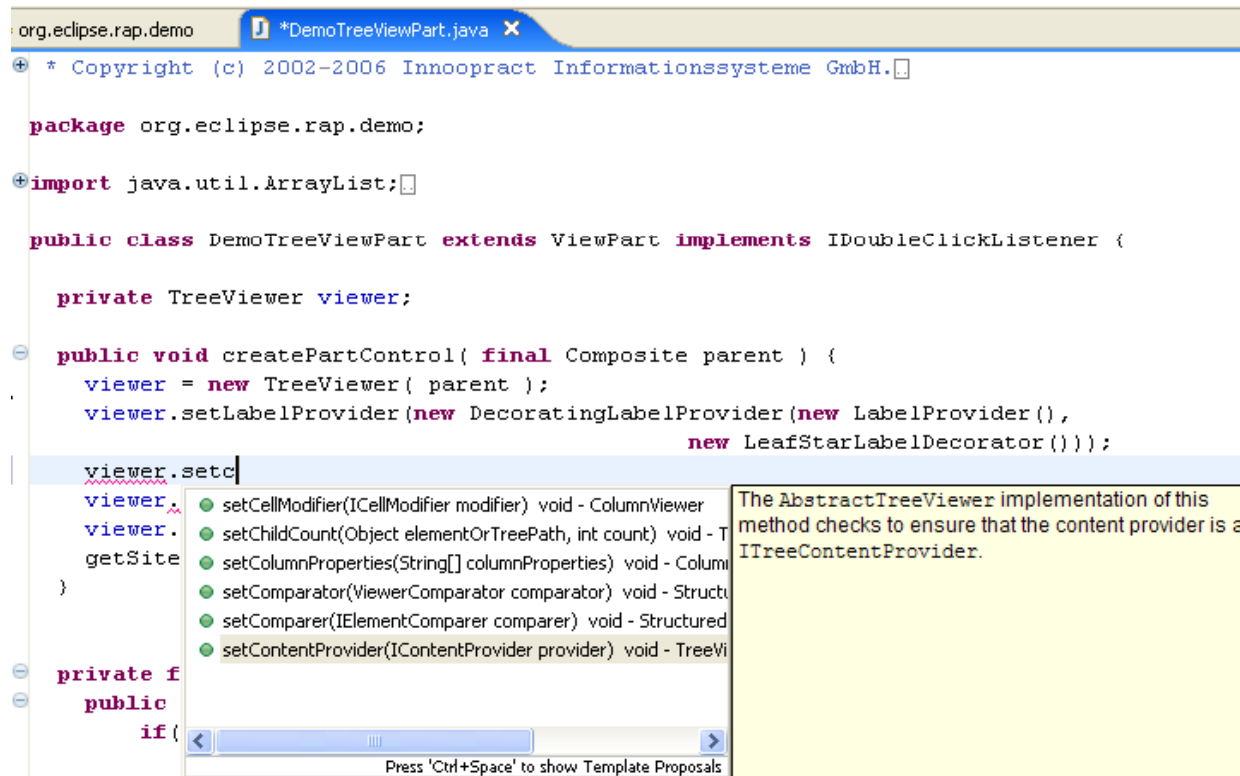


```
public class DemoWorkbench implements IEntryPoint {
    public int createUI() {
        Display display = PlatformUI.createDisplay();
        return PlatformUI.createAndRunWorkbench( display, new DemoWorkbenchAdvisor() );
    }
}
```

```
<extension
    id="org.eclipse.rap.demo.demoentrypoint"
    point="org.eclipse.rap.ui.entrypoint">
    <entrypoint
        class="org.eclipse.rap.demo.DemoWorkbench"
        id="org.eclipse.rap.demo.entrypoint1"
        parameter="default"/>
    </extension>
```

RAP aus Sicht des Entwicklers

1. JDT: Content Assist, Refactoring, etc., PDE, Javadoc...
2. Der Entwickler kommt nicht mit Javascript, CSS, HTTP, etc. in Berührung



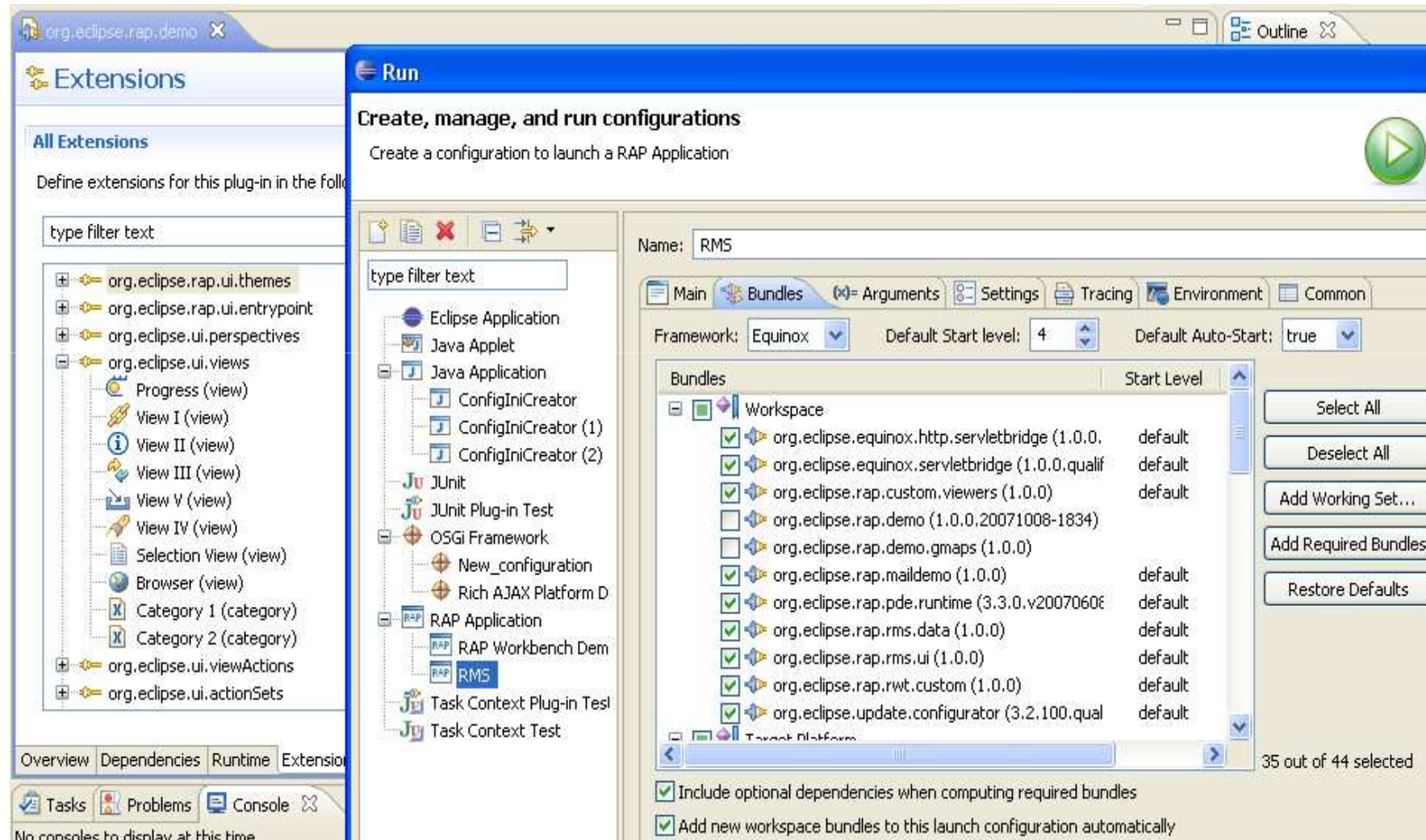
```

org.eclipse.rap.demo  *DemoTreeViewPart.java x
+ * Copyright (c) 2002-2006 Innoopract Informationssysteme GmbH.
package org.eclipse.rap.demo;
+import java.util.ArrayList;
public class DemoTreeViewPart extends ViewPart implements IDoubleClickListener {
    private TreeViewer viewer;
- public void createPartControl( final Composite parent ) {
    viewer = new TreeViewer( parent );
    viewer.setLabelProvider( new DecoratingLabelProvider( new LabelProvider(),
                                                         new LeafStarLabelDecorator() ) );
viewer.setC
viewer.
viewer.
getSite
}
- private f
- public
    if(
Press 'Ctrl+Space' to show Template Proposals

```

The AbstractTreeView implementation of this method checks to ensure that the content provider is an ITreeContentProvider.

RAP aus Sicht des Entwicklers



The screenshot shows the Eclipse IDE interface with the 'Run' dialog box open. The dialog is titled 'Create, manage, and run configurations' and is used to launch a RAP Application. The configuration name is 'RMS'.

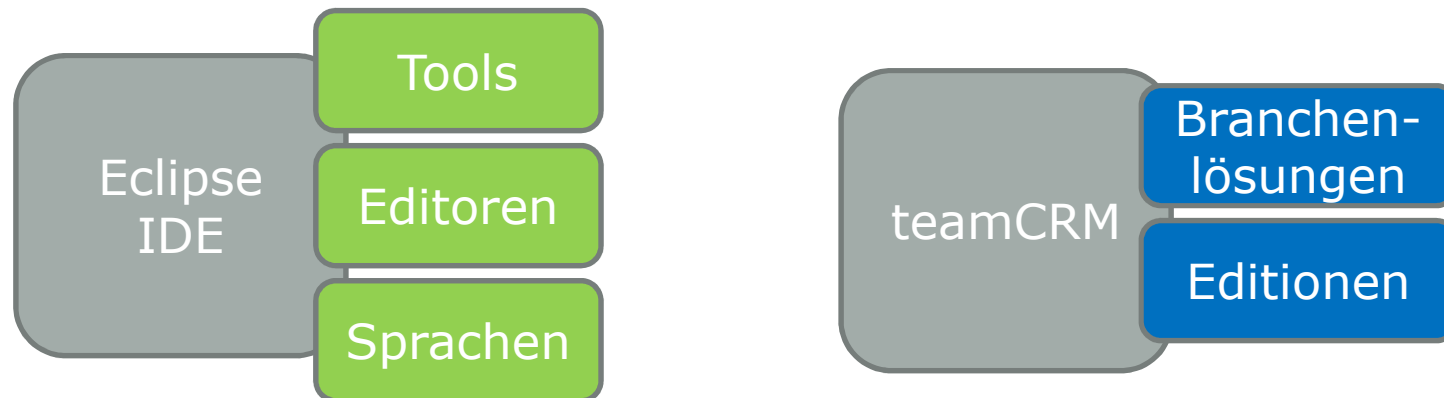
The 'Bundles' tab is selected, showing a list of bundles to be included in the launch configuration. The bundles are:

Bundle Name	Start Level
org.eclipse.equinox.http.servletbridge (1.0.0)	default
org.eclipse.equinox.servletbridge (1.0.0.qualif	default
org.eclipse.rap.custom.viewers (1.0.0)	default
org.eclipse.rap.demo (1.0.0,20071008-1834)	
org.eclipse.rap.demo.gmaps (1.0.0)	
org.eclipse.rap.maildemo (1.0.0)	default
org.eclipse.rap.pde.runtime (3.3.0.v20070600	default
org.eclipse.rap.rms.data (1.0.0)	default
org.eclipse.rap.rms.ui (1.0.0)	default
org.eclipse.rap.rwt.custom (1.0.0)	default
org.eclipse.update.configurator (3.2.100.qual	default

The 'Include optional dependencies when computing required bundles' and 'Add new workspace bundles to this launch configuration automatically' options are checked.

Anwendungsarchitektur

- Modularisierung mittels Plug-Ins
 - Lose Kopplung
 - Definierte Schnittstellen
 - Leichte Austauschbar- und Erweiterbarkeit
- Analogie zur Eclipse IDE

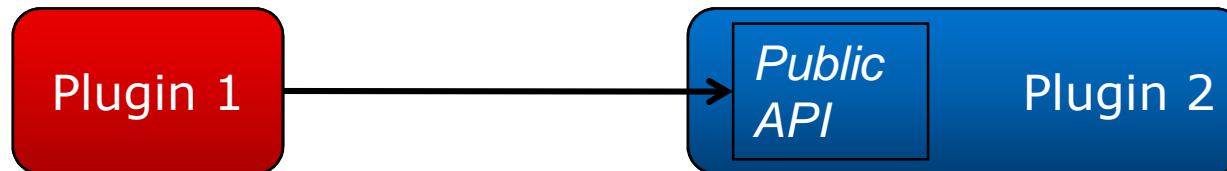


Plug-In-Abhängigkeiten

- Zwei Arten von Abhängigkeiten

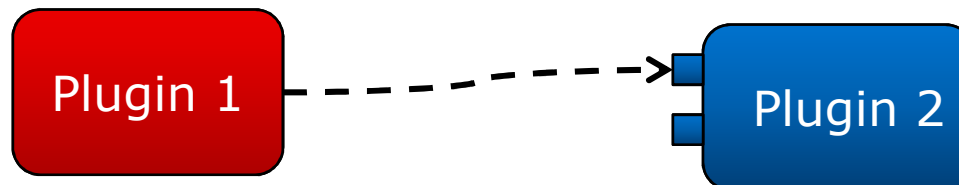
- Package-Imports

- Analog zur Verwendung von Bibliotheken

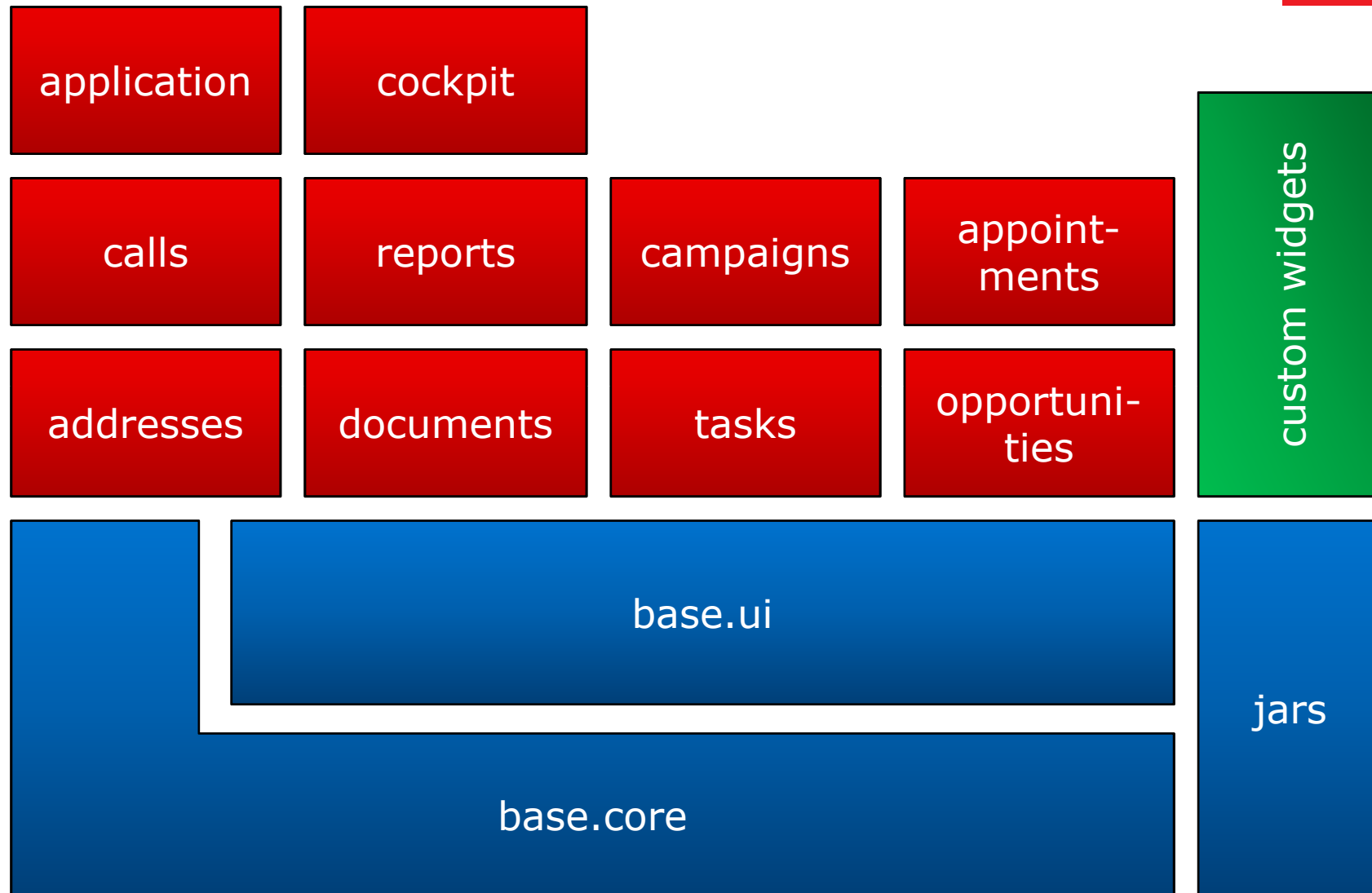


- Extension Points

- Lose Kopplung
 - Abhängigkeit wahlweise optional



Anwendungsarchitektur – Plug-Ins



Integration von Add-Ins

- In teamCRM: lokale Add-Ins für Komfortfunktionen
 - Dokumentenhandling (Internet Explorer, Firefox)
 - Emails (Outlook, Thunderbird)
 - Serienbriefe (Word)
- Demo
- Kommunikation zwischen lokalen Add-Ins und teamCRM mittels Service-Handler
 - Service-Handler laufen im Session-Kontext
 - Server-getriggerte UI-Updates möglich (Beispiel: Neue archivierte E-Mail)

Custom Widgets

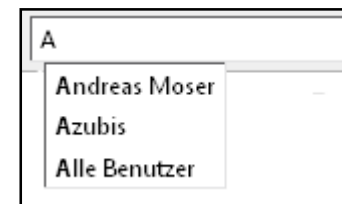
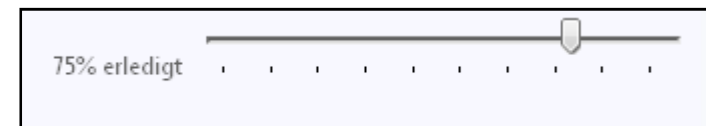
- Beispiele

- Kalender
- Toolbar
- Datepicker
- Slider
- Textfeld mit Vorschlagsliste

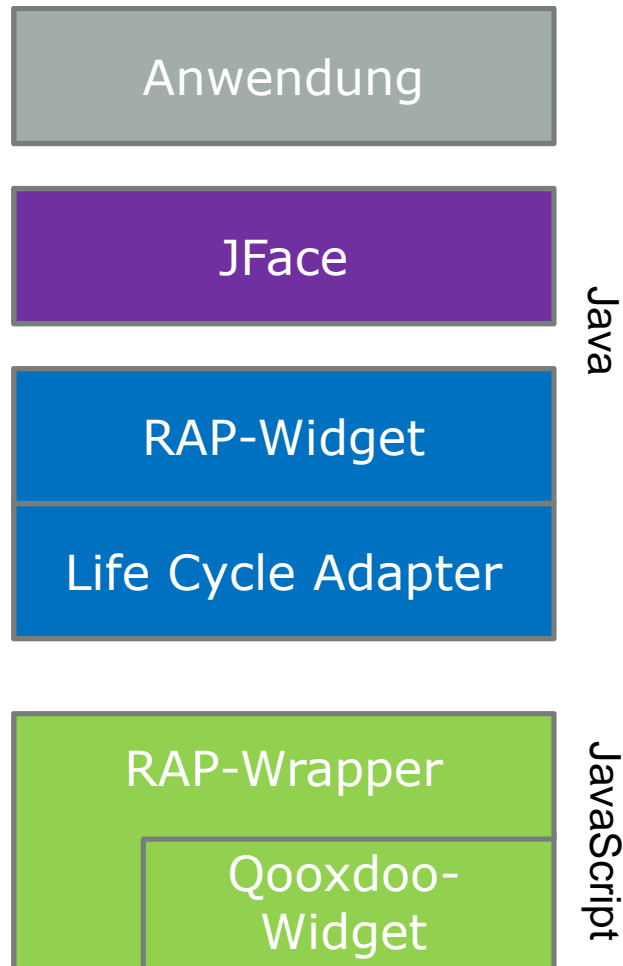


- Bestandteile

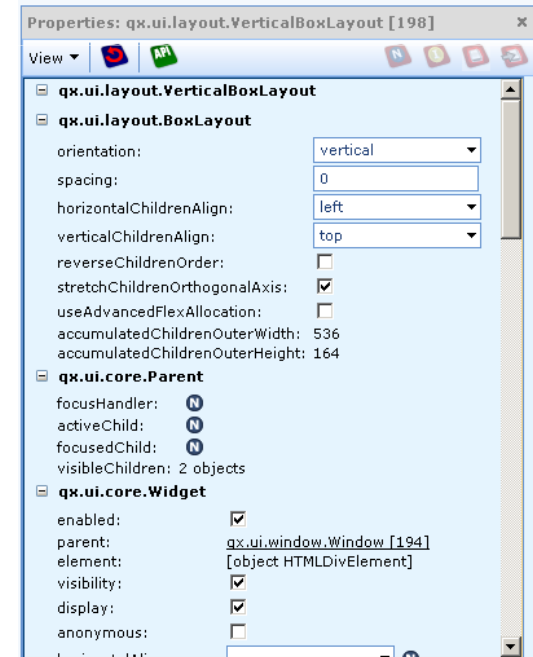
- Qooxdoo-Implementierung
- Java-SWT-Widget
- RAP Life-Cycle-Adapter



Custom Widgets cont.



- Entwicklung Layer-spezifisch
 - Test und Demos je Schicht
- Umfangreiches Tooling für JS-Schicht
 - Property-Editor
 - Statische Code-Analyse
 - Build-Tools



Fehlersuche

- Einfacher:
 - Bugs in Anwendungscode.
 - Debugging lokal in Eclipse (Jetty) oder Remote (Tomcat)
- Schwieriger:
 - Bugs in Custom-Widgets (JavaScript)
 - Debugging mit Firebug (Firefox-Plug-In) relativ komfortabel
 - Tests in allen Browsern notwendig
 - In teamCRM: Umfangreiches Tooling für JS-Widgets (Objektinspektor, Code-Anomalien, etc.)
 - Qooxdoo-Know-How unabdingbar
- Aussichtslos:
 - Bugs im RAP-Framework ☺
 - www.eclipse.org/rap/bugs.php



Theming

- Großteil des Aussehens von Widgets über Themes anpassbar
 - Farben, Ränder, Hintergrund, Fonts, etc
- Realisierung recht mühselig und zeitaufwändig
 - Themes werden statisch geladen → Anwendungsneustart, um Änderungen zu sehen
 - Varianten für einen Widgettyp erst seit kurzem möglich
- Definition über Property-Dateien (in Zukunft CSS)

```
# Space between image and text
# default: 2px
button.spacing: 5px

# Font for buttons with style PUSH or TOGGLE
# default: 11px "Segoe UI", Corbel, Calibri, Tahoma, "Lucida Sans Unicode", sans-serif
button.font: 12px "Segoe UI", Arial, Helvetica, sans-serif
```

Theming



teamCRM Angemeldet als: Michael Grün 03.06.2008 13:43

Kontakte Startseite | Alle Einzelkontakte

Ansichten Bearbeiten

- Alle Kontakte
- Alle Einzelkontakte**
- Alle Organisationen
- test.meintag
- test

Letzte Suchen

- linked
- blue, !Kontakt ist eine Organisation
- blue
- *
- a
- 0xD8334586DAC8361396E201A3D4496
- fees

Navigation: Codepit, Suche, **Kontakte**, Kalender, Dokumente, E-Mails, Kampagnen, Verkaufschancen, Telefonate, Berichte, Aufgaben

Toolbar: Neu, Öffnen, Duplizieren, Löschen, Aktualisieren, Speichern, Drucken, Zugriffsrechte erweitern, Bericht erstellen, Überprüfen, Anreichern, Dubletten, E-Mail senden, Telefonieren, Brief schreiben, Folgeaktion, Neue Verknüpfung, Verknüpfen mit

Suche: Schnellsuche, Schlagwortsuche

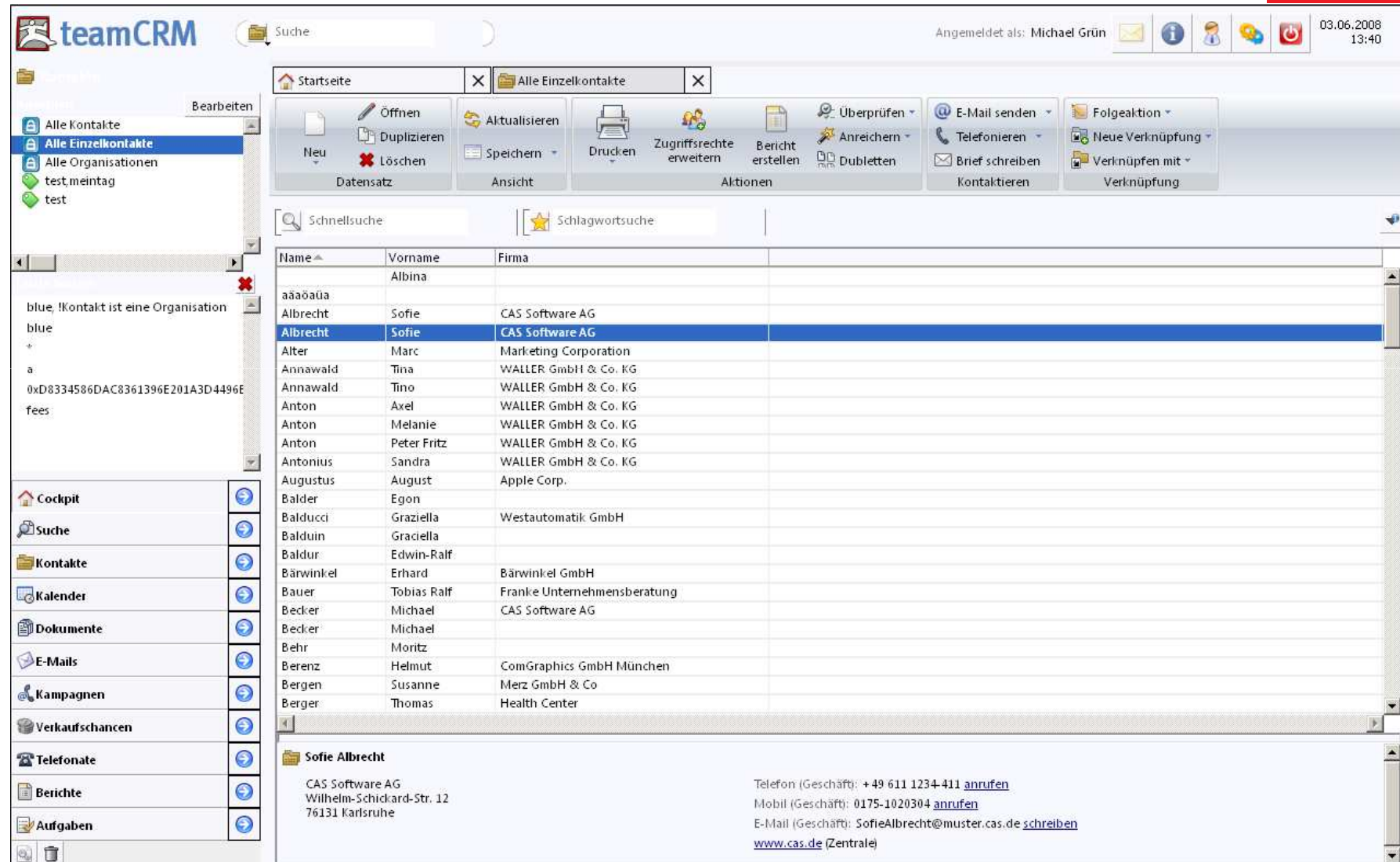
Name	Vorname	Firma
Albina		
Albrecht	Sofie	CAS Software AG
Albrecht	Sofie	CAS Software AG
Alter	Marc	Marketing Corporation
Annawald	Tina	WALLER GmbH & Co. KG
Annawald	Tino	WALLER GmbH & Co. KG
Anton	Axel	WALLER GmbH & Co. KG
Anton	Melanie	WALLER GmbH & Co. KG
Anton	Peter Fritz	WALLER GmbH & Co. KG
Antonius	Sandra	WALLER GmbH & Co. KG
Augustus	August	Apple Corp.
Balder	Egon	
Balducci	Graziella	Westautomatik GmbH
Balduin	Graciella	
Baldur	Edwin-Ralf	
Bärwinkel	Erhard	Bärwinkel GmbH
Bauer	Tobias Ralf	Franke Unternehmensberatung
Becker	Michael	CAS Software AG
Becker	Michael	
Behr	Moritz	
Berenz	Helmut	ComGraphics GmbH München

Sofie Albrecht

CAS Software AG
 Wilhelm-Schickard-Str. 12
 76131 Karlsruhe

Telefon (Geschäft): +49 611 1234-411 [anrufen](#)
 Mobil (Geschäft): 0175-1020304 [anrufen](#)
 E-Mail (Geschäft): SofieAlbrecht@muster.cas.de [schreiben](#)
www.cas.de (Zentrale)

Anwendung ohne Theming



The screenshot shows the teamCRM application interface. At the top, there is a search bar and a user profile for Michael Grün, logged in on 03.06.2008 at 13:40. The main area displays a list of contacts with columns for Name, Vorname, and Firma. The contact 'Sofie Albrecht' is selected and highlighted in blue. Below the list, the details for Sofie Albrecht are shown, including her address and contact information.

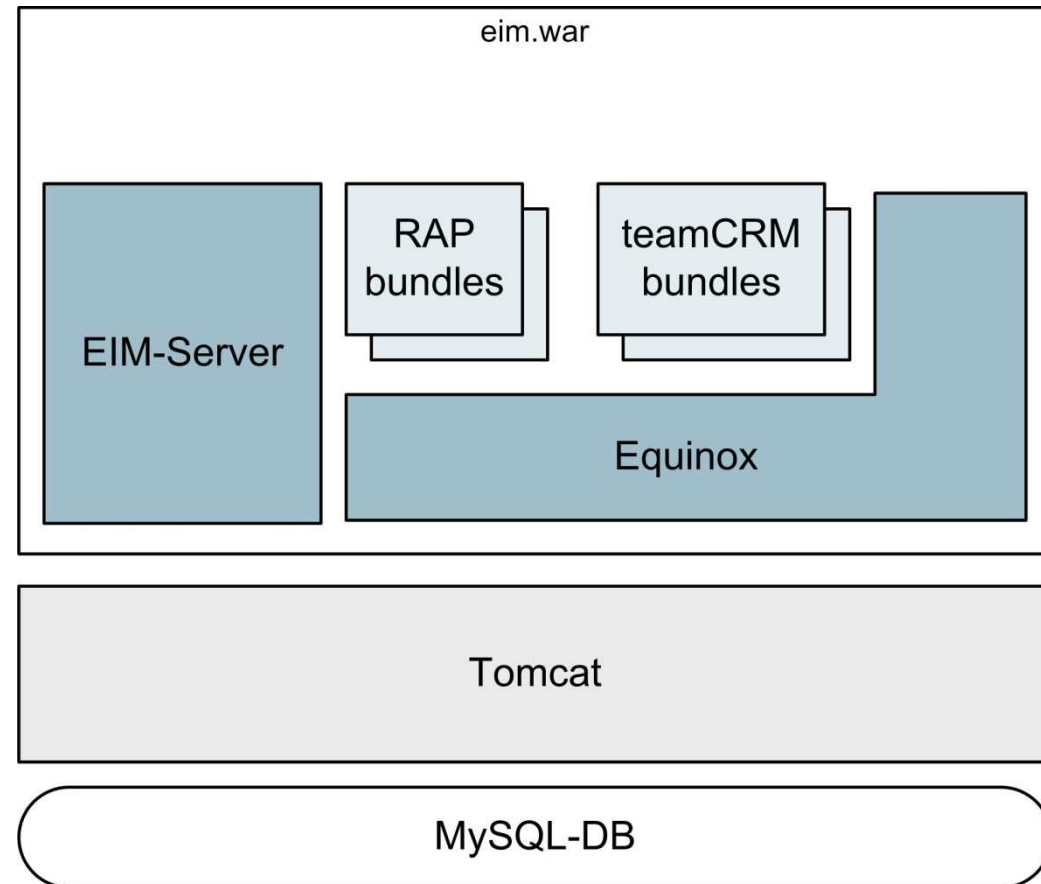
Name	Vorname	Firma
	Albina	
Albrecht	Sofie	CAS Software AG
Albrecht	Sofie	CAS Software AG
Alter	Marc	Marketing Corporation
Annawald	Tina	WALLER GmbH & Co. KG
Annawald	Tino	WALLER GmbH & Co. KG
Anton	Axel	WALLER GmbH & Co. KG
Anton	Melanie	WALLER GmbH & Co. KG
Anton	Peter Fritz	WALLER GmbH & Co. KG
Antonius	Sandra	WALLER GmbH & Co. KG
Augustus	August	Apple Corp.
Balder	Egon	
Balducci	Graziella	Westautomatik GmbH
Balduin	Graciella	
Baldur	Edwin-Ralf	
Bärwinkel	Erhard	Bärwinkel GmbH
Bauer	Tobias Ralf	Franke Unternehmensberatung
Becker	Michael	CAS Software AG
Becker	Michael	
Behr	Moritz	
Berenz	Helmut	ComGraphics GmbH München
Bergen	Susanne	Merz GmbH & Co
Berger	Thomas	Health Center

Sofie Albrecht
 CAS Software AG
 Wilhelm-Schickard-Str. 12
 76131 Karlsruhe

Telefon (Geschäft): +49 611 1234 411 [anrufen](#)
 Mobil (Geschäft): 0175-1020304 [anrufen](#)
 E-Mail (Geschäft): SofieAlbrecht@muster.cas.de [schreiben](#)
www.cas.de (Zentrale)

Deployment

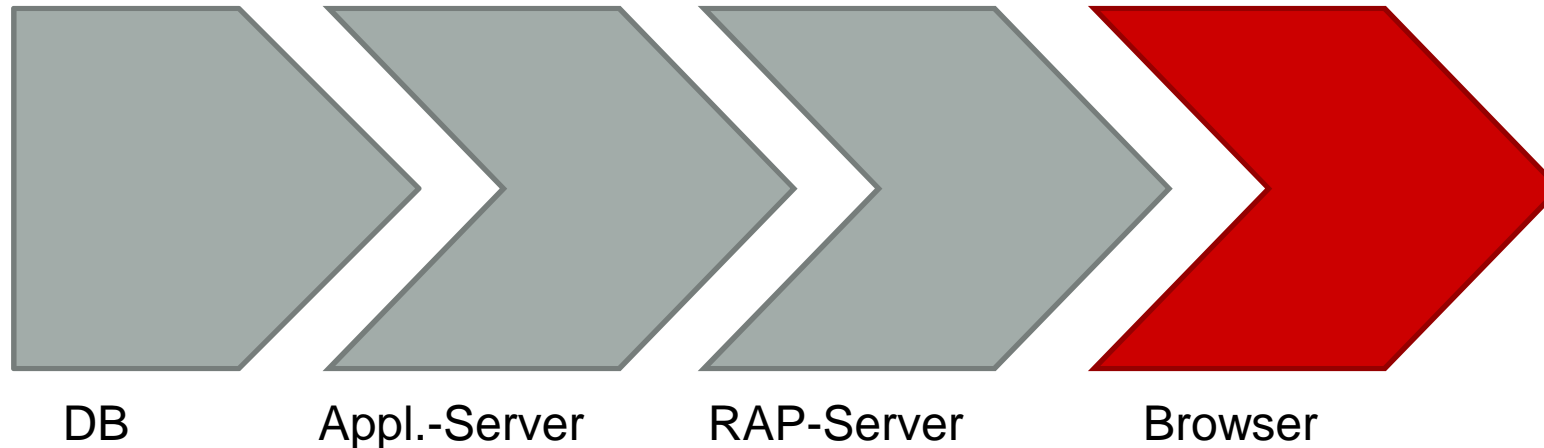
- Presentation-Layer innerhalb von Equinox
- Business und Data Layer außerhalb
- ➔ Integration von Nicht-OSGi Libs notwendig
 - Applikationsserver
 - Javax-Servlet
- Kommunikation über ServletBridge
 - Externe Packages in Equinox-Kontext verfügbar
 - Direkter Zugriff auf Java-Klassen möglich



Build-Prozess

- Vollautomatisierter Daily-Build
 - Eclipse Release Engineering (Releng)
 - Abrufen der Sourcen
 - Kompilieren
 - Feature erstellen
 - War-File packen
 - Auf Test-Rechnern verteilen
 - Der Deployment-Prozess ist relativ stabil...
 - Aber: Bundle-Abhängigkeiten werden erst zu Laufzeit aufgelöst
 - Fehleranalyse oft langwierig und mühsam






Performance

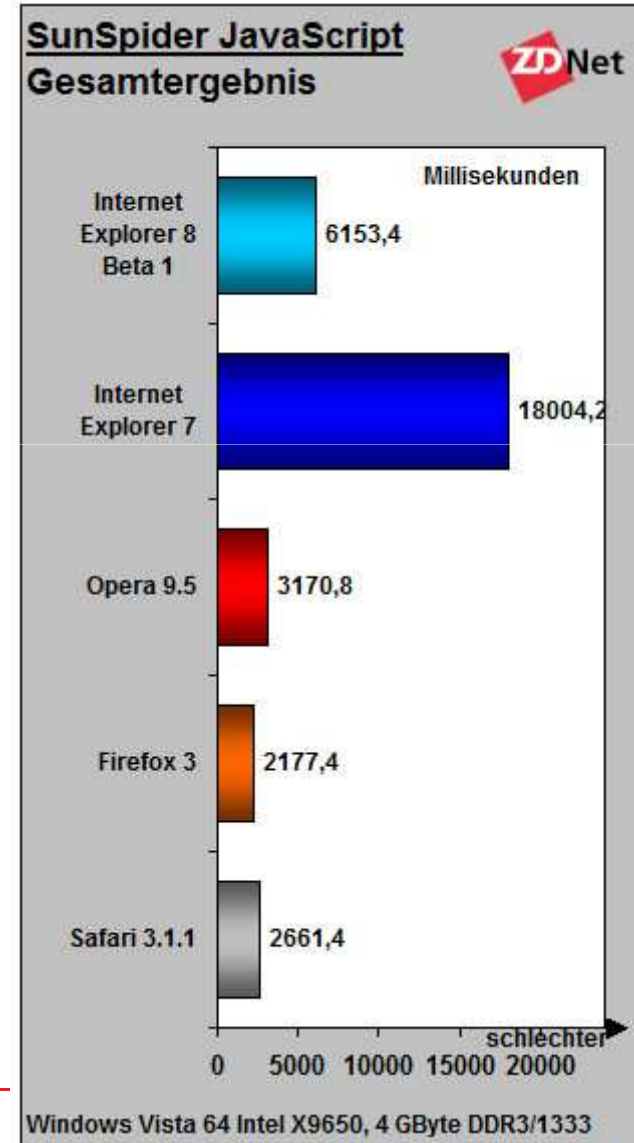


- Anforderungen an Browser enorm
 - 30.000 Zeilen JS-Code in Startseite
 - ~ 4000 <div>
 - Viele kleine Requests
- Geschwindigkeit
 - Viel RAM & Multi-Core ein Plus
 - Firefox 3, Safari spitze, IE7 eher zäh

Speicher & Performance

- Browser

	Browser	RAM nach Start	RAM nach Demo
	Firefox 2	16 MB	68 MB
	Firefox 3	35 MB	112 MB
	Safari RC1		
	Safari	40 MB	125 MB
	IE 7	34 MB	318 MB
	IE 8 Beta	26 MB	278 MB



Fazit

- RAP bietet mit SWT, JFace, Workbench solides Fundament für Anwendungen
 - Saubere Anwendungsarchitektur möglich
 - Modularisierung mittels Bundles
 - MVC mit JFace-Viewern
 - Lose Kopplung dank SelectionService, etc.
- Einschränkungen
 - RCP-Komponenten manchmal nicht flexibel genug
 - Theming recht aufwändig
 - Browser als Laufzeitumgebung nur bedingt geeignet

The end

- Demo: Plug-ins austauschen

Backup

