



# Java GUI - Quo vadis?

Björn Müller, CaptainCasa



# Björn Müller

- SAP, 1991
  - Anwendungsentwicklung
  - Architekturentwicklung
  - Java Application Server
- Casabac Technologies, 2001
  - AJAX „Pionier“
  - 2005 Übernahme durch Software AG
- CaptainCasa Community, 2007
  - Corporate Community im Bereich „Frontends für Unternehmensanwendungen“

# CaptainCasa Corporate Community

- Softwarehersteller
  - D / CH / A
  - 30 - 400 Mitarbeiter
  - Logistics, Financials, Human Resources, Administration, ...
- Definition und Entwicklung einer gemeinsam genutzten Rich Client Plattform

# CaptainCasa Community



# Die ganze Welt geht in Richtung HTML5. Oder?



# Ein „typischer“ Entwickler

**HTML5!!!**



# Ein „typischer“ Entwickler

**HTML5!!!**



Eclipse!

Mail

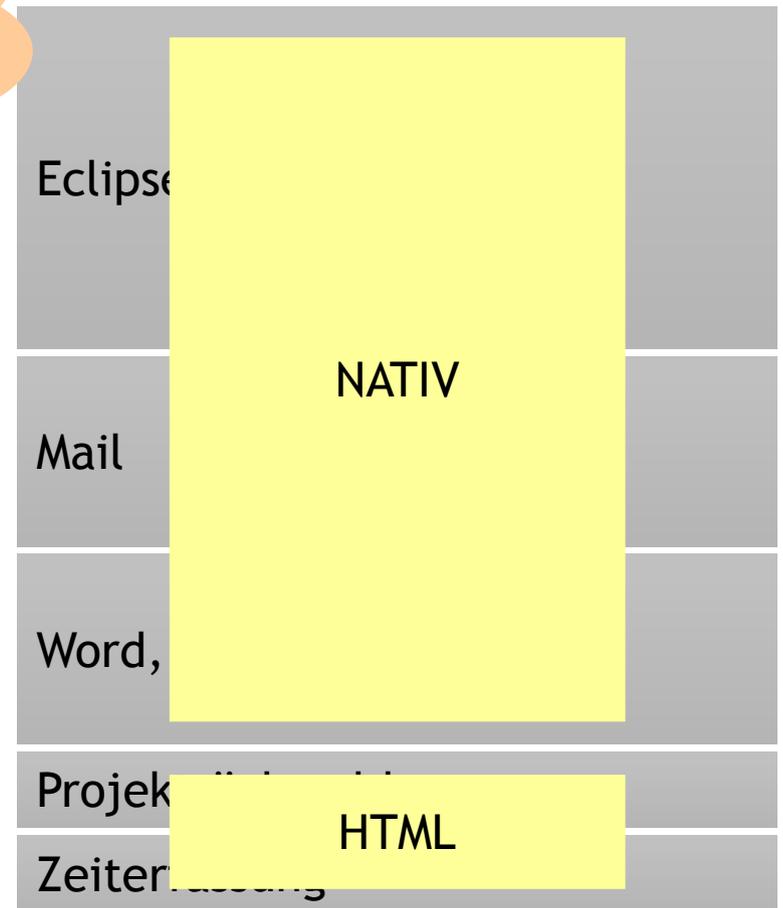
Word, Visio, Powerpoint

Projektrückmeldung

Zeiterfassung

# Ein „typischer“ Entwickler

HTML5!!!



# Native Java UI Technologien

- Es macht Sinn, sich mit nativen Java UI Technologien zu beschäftigen
  - Große Menge an funktionierenden nativen Anwendungen, die existieren und die weiterentwickelt werden
  - Große Anzahl an Entwicklern, die Java beherrschen und eine passende UI Technologie nutzen wollen
  - Konkrete Szenarien, in denen es auch zukünftig Sinn macht, nicht in Richtung HTML(5) zu gehen



# Java UI Technologien

- Java Swing
- Java SWT (Eclipse)
  
- Java auf Android
- Java FX



# Java Swing und Java SWT

# Java Swing und Java SWT

- Schnell
- Stabil
- „Programmierung!“ von Oberflächen
- Auf allen Desktopplattformen
  - Nicht auf mobilen Plattformen!
- Langfristig supported
- ...es tut sich nicht viel...!



# Java Swing versus Java SWT

- SWT ist schneller. - Stimmt nicht!
  - Eclipse <==> NetBeans
- Augenfälligster Unterschied
  - Swing: eigenes Control-Rendering
    - Sieht anders aus als OS
  - SWT: Controls aus Betriebssystem
    - Sieht so aus als OS



# Java Swing versus Java SWT

- Einsatz von Java SWT
  - Im Eclipse Umfeld!
  
- Einsatz von Java Swing
  - Industrieanwendungen
  - Utilities
  - Applets / Webstart
  - ...
  - (NetBeans)

# Java Swing und Java SWT

- Java 1.6 Update 10
  - Der „Hammerrelease“ für Java Frontendentwicklung!
  - Wesentliche Verbesserung der Lauffähigkeit von Applets
    - Eigener Prozess
    - Eigene Konfiguration
    - Individuelle Versionierung



# Java Swing Umfeld

- Überschaubarer Markt an hochwertigen Controls
- Eine Reihe Swing unterstützende Frameworks
  - JFreeChart
  - Jasper Reports
  - Swinglabs-Output
    - PDF Renderer
    - OSM Viewer



# Java Swing „Pain Points“

- Browserintegration
  - Teilweise recht „nette“ Lösungen

Swing  
SWTBridge  
SWT  
Browser

- Animationen

# Java SWT Umfeld

- Überschaubarer Markt an hochwertigen Controls
- Eine Menge Frameworks im Eclipse Umfeld
  - BIRT
  - EMF
  - ...“Eclipse Projects“



# Java Android

# Java Android != Java...

- Sprachumfang - OK, voll umgesetzt!
- Bibliotheken - Grrr.
  - java.lang.\*, java.util.\*, ... - OK.
  - Swing - Nicht umgesetzt, eigene UI Bibliothek.
  - Generell: Immer wieder ärgerliche Lücken.
- Fazit: komplett eigene Java UI Implementierung notwendig.

# Android Demo

- Demo.



# Java Android UI Entwicklung

- Einfach...
  - Einfache APIs.
  - „Guter“ Entwicklungsprozess.
  - „Gutes“ Tooling.
  - Einfache Animationen
- Kommentare zum Einsatz
  - Deskriptiv & per Programm
  - Einsatz von Styles
  - Eine Reihe von Restriktionen
- Fazit
  - Umstieg von Swing: KEIN PROBLEM, aber...

# Java Android UI Entwicklung

- Restriktionen, „Pain Points“
  - Android Java != Java
  - Memory Konfiguration der Anwendung
  - Begrenzte Höhe des Control Trees
  - Multiple Scroll Level
  - Styles nicht zur Laufzeit änderbar
  - Nur ein Listener für Events
- ...wird den Stallgeruch „komme von Mini-Devices“ noch nicht ganz los.



# Java FX

# Java FX

- Java 1.\* - Katastrophe
    - Eigene Scriptingsprache...
  - Java 2.0 - Wunderbar!
    - Zurück zu Java!
    - Java FX als neue UI Control Library
  - „Swing Pain Points“ wurden adressiert
    - Browser Integration
    - Styling
    - Animationen
- ...und in ansprechender Weise gelöst!

# Java FX Demo

- Demo



# Java FX UI Entwicklung

- Einfach...
  - Einfache APIs.
  - „Guter“ Entwicklungsprozess.
  - „Gutes“ Tooling.
  - Einfache Animationen. Einfaches Styling.
- Kommentare zum Einsatz
  - Deskriptiv & per Programm
  - „Sauberer“ Einsatz von Styles.
  - Schnell.
  - Viele Controls verfügbar, aber doch noch ein paar fehlende Grund-Controls
- Fazit
  - Umstieg von Swing: KEIN PROBLEM

# Java FX UI - Fragezeichen...

- Politische Positionierung
  - Konkurrent 1: HTML5
  - Konkurrent 2: Swing
- Plattform Support außerhalb der Desktop Welt
  - Android
  - iOS



# Positionierung

# Java native <=> HTML5

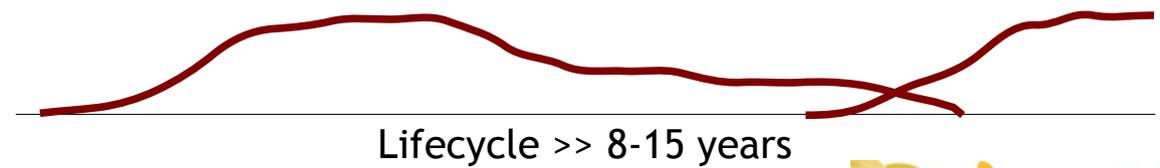
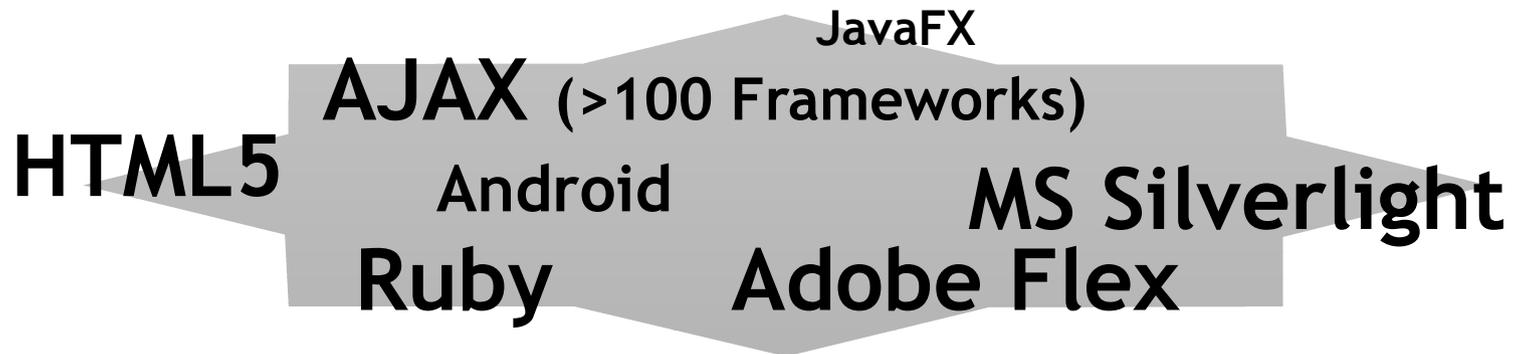
- Klare Positionierungen für „Java native“
  - Eigenständige Client Anwendungen.
  - Android > autarke Apps
  - Desktop > lokal laufende Clients
- Klare Positionierung für „HTML5“
  - Server basierte Anwendungen für anonyme Endnutzer
    - „Reach“ ist entscheidend!
      - Installationsaufwand
      - Plattformabdeckung
  - Zusammenarbeit mit Designern

# „Java native“ im Umfeld von Unternehmensanwendungen

- Merkmale von Unternehmensanwendungen
  - Viele Dialoge (>> 100)
  - Bandbreite der Dialoge:
    - Super-komplex / flexibel / ...
    - Einfach
  - Server-seitige Logik / Regeln
  - Typische Nutzer
    - Sachbearbeiter - massive Nutzung
    - Management - Reporting
    - Mitarbeiter - Self Service



# „Java native“ im Umfeld von Unternehmensanwendungen



# „Java native“ im Umfeld von Unternehmensanwendungen

- Mittelständische Softwarehersteller
  - Kleine Teams stemmen große Anwendungen.
  - Keine Zeit und erst recht keine Lust auf
    - Browserinkompatibilitäten
    - Javascript Fummeleien
    - ...
  - Schuld ist immer der Softwarehersteller.

# Wo schlägt das Herz der Anwendung?



Abb. 4: Auszubildender

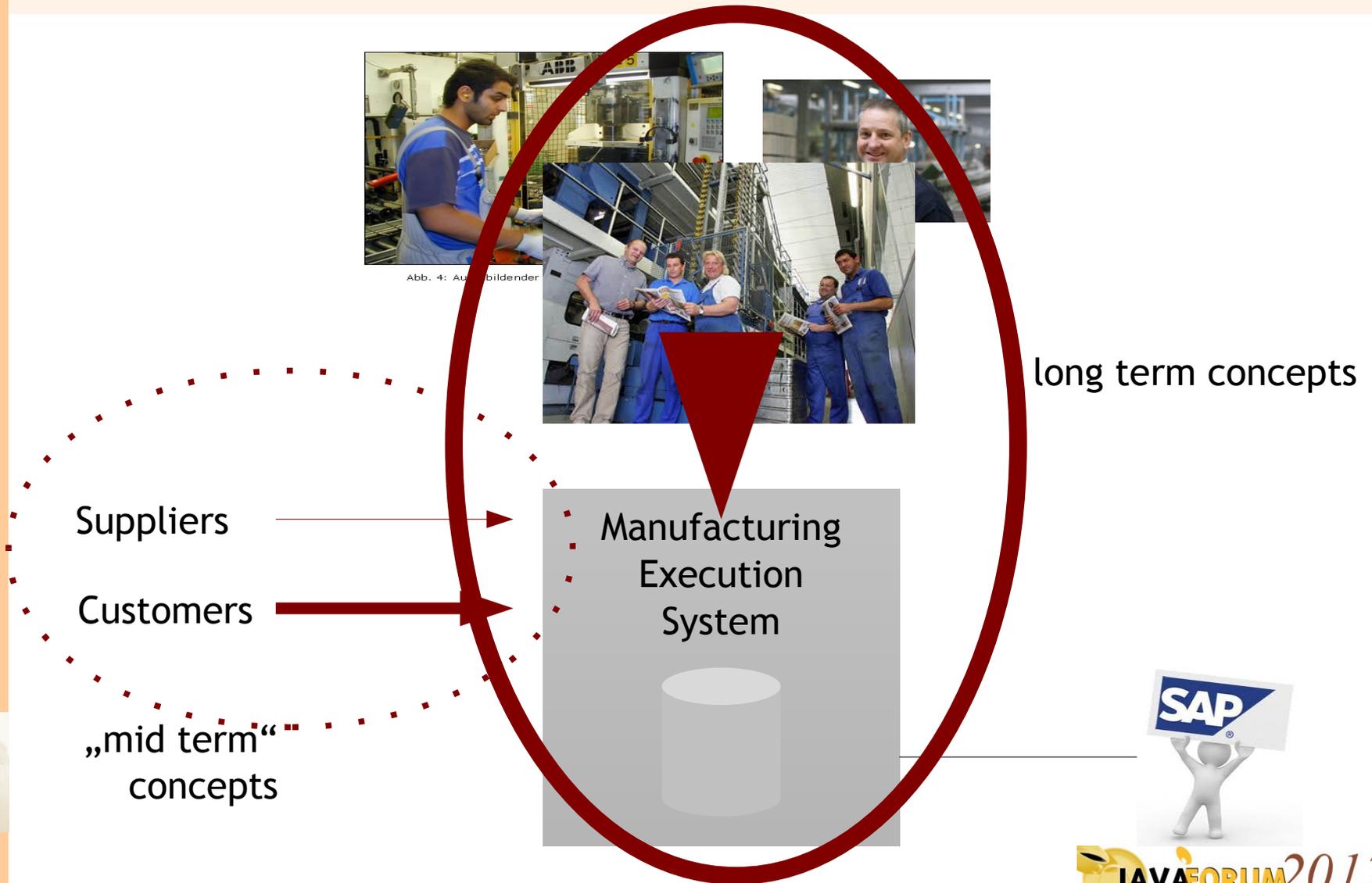
Suppliers →

Customers →

Manufacturing  
Execution  
System



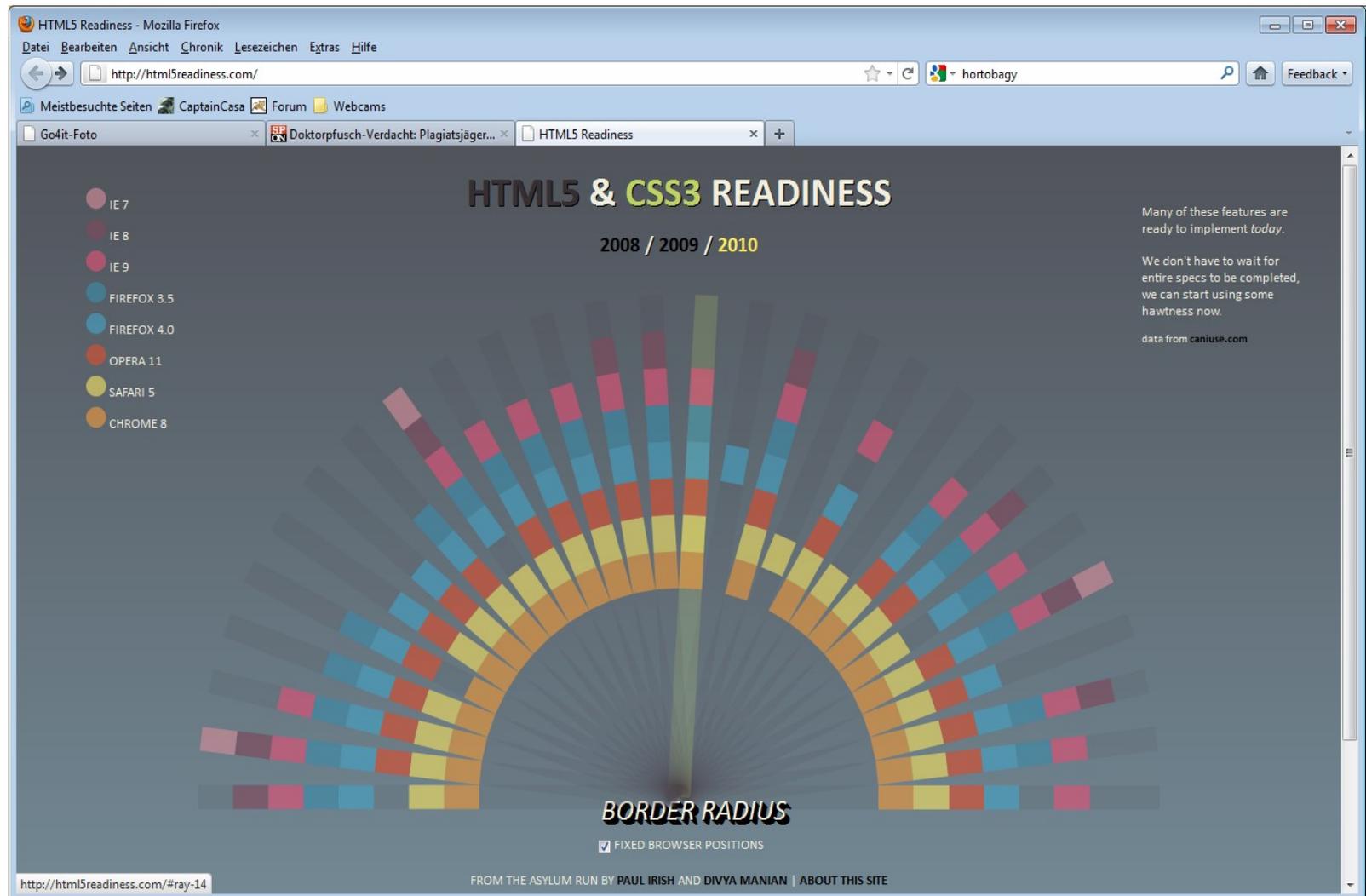
# Wo schlägt das Herz der Anwendung?



# HTML(5) Hemmnisse

- Heterogene Browserwelt
  - Desktop \*x, Android \*x, iOS
- JavaScript
- Unmenge an Framework und Hypes
- Höhe der Frameworkstacks
  
- ...ist HTML(5) wirklich die passende Technologie für den Kernteil der Anwendung?

# http://html5readiness.com/



# Valide Option

- Java Native für den Kernteil der Anwendung
  - Sachbearbeiterdialoge
  - Datenerfassung
  - Konfiguration
  - ...langfristige Stabilität!
- HTML(5) für die Außenbereiche der Anwendung
  - Partnerdialoge
  - Self Service Dialoge
  - Portalintegration
  - ...kurzfristige Reaktion!

# Zweifelhafte Option

- Ein UI Framework als „Eierlegende Wollmilchsau“
  - Überforderung der Frameworks!
- HTML5
  - Design von einzelnen Dialogen
  - Scripting
  - Einfach!
- Java native
  - Style Guide für viele Masken
  - Programmierung
  - Flexibel!

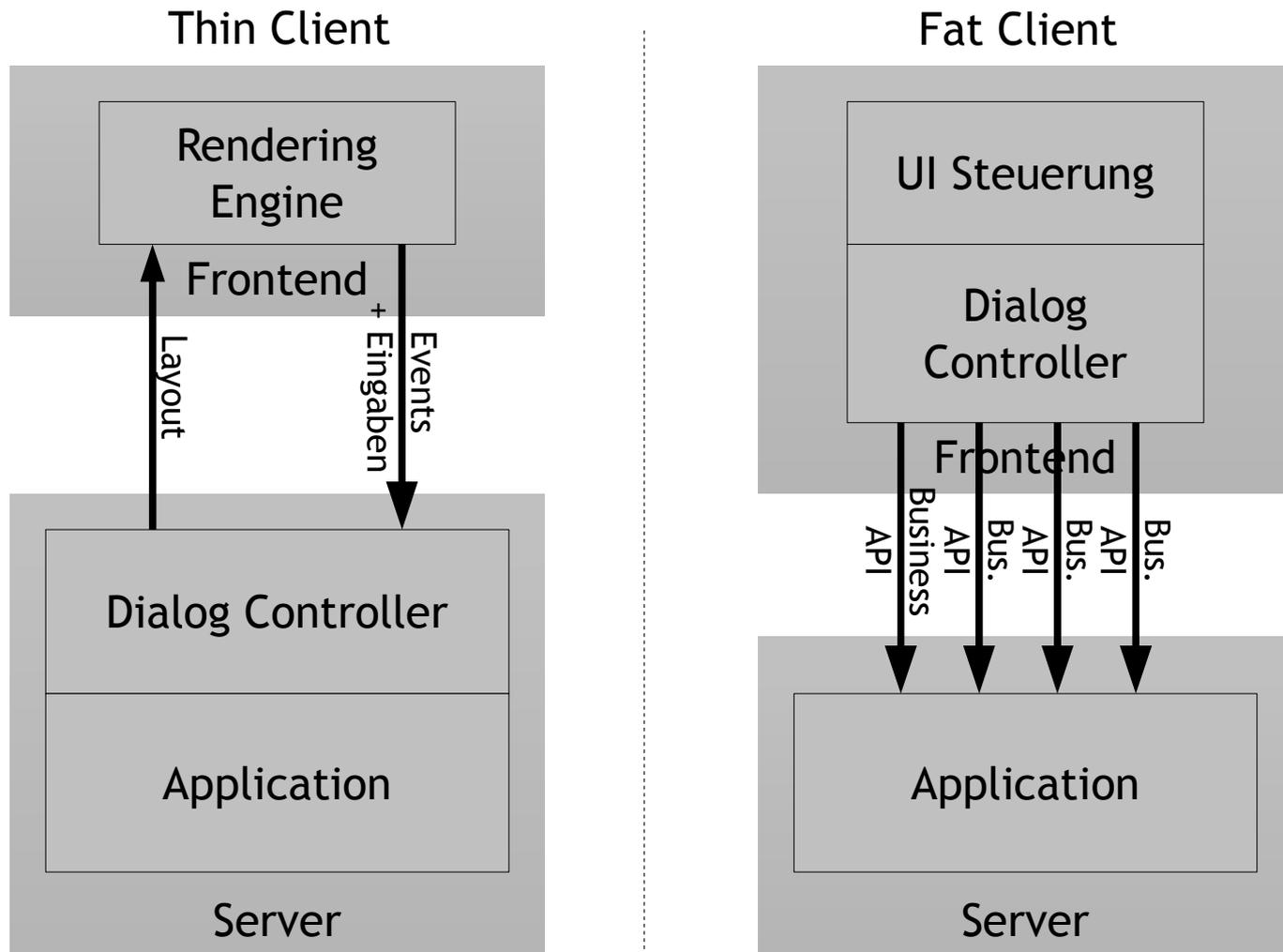
# Bessere Option

- Nutzung von jeweils adäquaten Technologien
- Native Java
  - „Kern“
  - Sachbearbeitermasken
  - Operative Nutzung
- HTML5
  - Massennutzung, Self Service
  - Mobile Szenarien
  - Partnereinbindung

# Architektur von nativen Java Clients

Native != Fat

# Thin or Fat Architecture

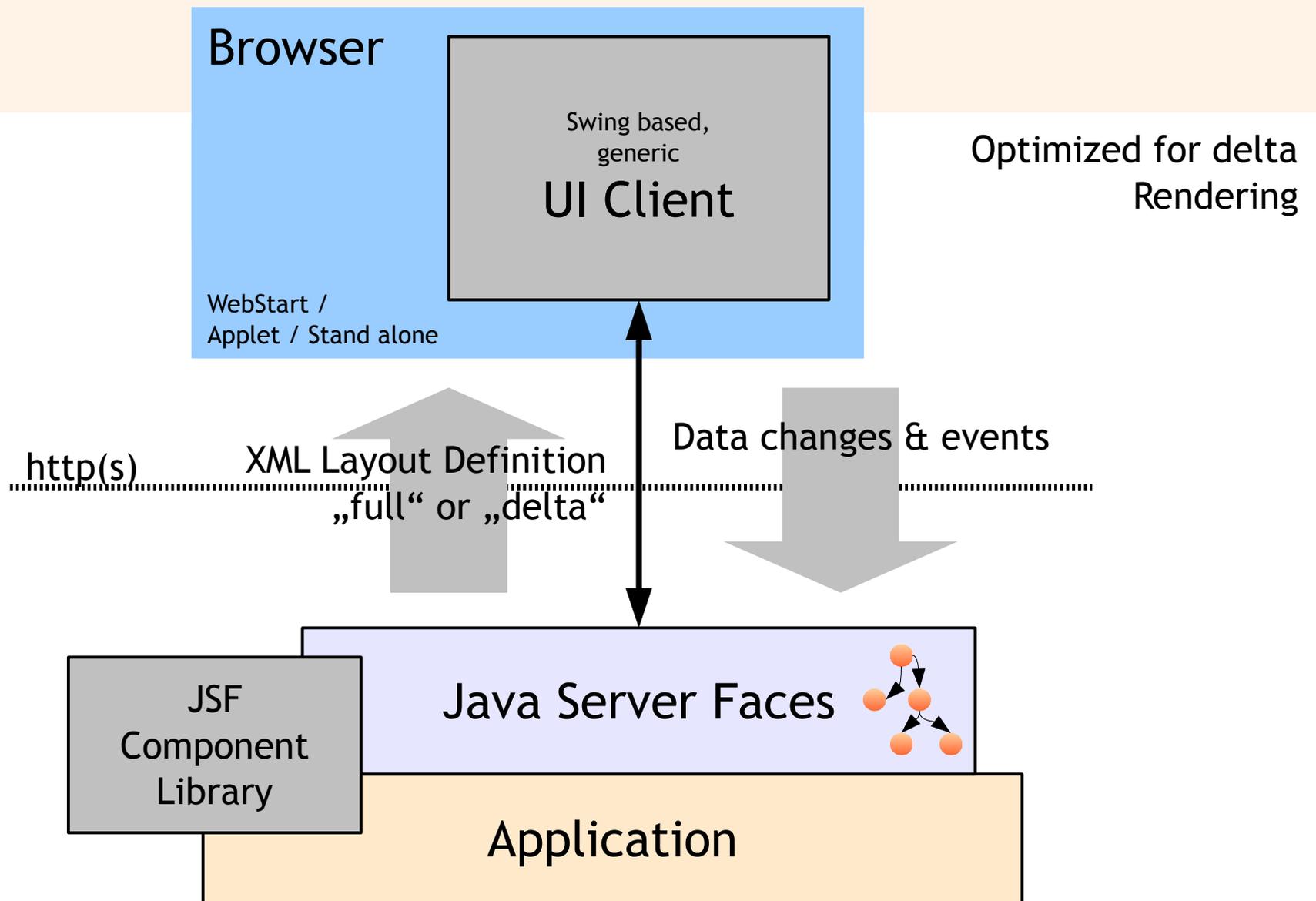


# Thin or Fat Architecture

- „Thin Client“ - Vorteile
  - Nur Server-side Entwicklung
  - EIN Interface zwischen Client und Server
  - EIN Roundtrip pro Interaktion
  - Kein Client Rollout bei Änderungen/Korrekturen
  
  - Architektonische Entkopplung des Clients vom Server
- „Fat Client Falle“
  - Unbewusste Entscheidung für Fat Client durch fehlende Architektur



# CaptainCasa



# CaptainCasa Enterprise Client

- Mini Demo...





# Fazit

# Fazit



# Fazit

- Untersuchen Sie Ihre Anwendung / Ihr Vorhaben
  - Wo liegt der Kern der Anwendung?
  - Wie langfristig ist meine Entwicklung in diesem Kern?
  - Wer sind die Nutzer?
  - Wo muss ich langfristig denken? Wo will ich flexibel denken?
- Schließen Sie von Ihrer eigenen Toolumgebung auf das, was Sie Ihren Anwendern „antun“.

# Fazit

- Gestatten Sie sich ein Recht auf Ihre eigene Meinung!  
Die UI Welt ist „volatil“ und wird es bleiben...
- Ihre Anwendung braucht kein „UI Dogma“ sondern ein stabiles User Interface!



# Conclusions III

- Interested in CaptainCasa Community?
- Visit our booth outside...
- Visit <http://www.CaptainCasa.com>
  - Downloads
  - Demos
  - Infos
- Contact us: [info@CaptainCasa.com](mailto:info@CaptainCasa.com)