

Warum und Wo

der Einsatz von JavaFX Sinn macht

Björn Müller

- 2001 HTML/JavaScript basierte Technologien
Casabac GUI Server
- 2007 Java-basierte Client Technologien
CaptainCasa Community
- Fokus auf betriebswirtschaftliche
Anwendungssystemen
 - „old economy applications“

CaptainCasa Community

- Community mittelständischer Softwarehäuser
 - Umfangreiche Geschäftsanwendungen
 - Kernuser: Sachbearbeiter
- Gemeinsame Rich Client Infrastruktur
 - Server-zentrischer Architektur
 - 2007 Entscheidung für Swing (!)
 - 2012 Entscheidung für JavaFX (Kompatibel!)

Es gibt wieder eine HTML5 Alternative!

- Werdegang bisheriger Alternativen nicht ganz so erfolgreich...
 - MS Silverlight
 - Adobe Flex
- ...oder „Geheim-Tipps“
 - Qt
- Wacker seit eh und je
 - Java Swing
 - Java SWT

Probleme beim Positionieren von JavaFX

- Architekturdefinition durch Slogans
 - Wir machen alles nur noch im Browser!
 - HTML5 ist unsere Strategie!
 - Wir erlauben keine Installation auf dem Frontend!
- Oder, etwas seriöser:
 - HTML5 ist „in“, Java(FX) im Client ist noch nicht „in“ und braucht Überzeugungsaufwand
 - JavaFX im Client muss echte Vorteile bieten, um Anwendung zu finden!
 - Der Einsatz von JavaFX ist kein Selbstzweck.

Laufzeitumgebung des Clients

HTML5

- Windows
- Linux
- MacOS
- Android
- iOS
- ...überall, wo es adäquate Browser gibt

JavaFX

- Windows
- Linux
- MacOS
- Android, iOS
„angedacht“, kein konkreter Zeitpunkt

Programmiersprache im Client

HTML5

- JavaScript
- Keine Typisierung
- Prototyping
- „Aufprägen von außen“
- Nutzung als Compilat...

JavaFX

- Java
- Starke Typisierung
- Vererbung, Interfaces
- „Struktur von innen“

JavaScript Prototyping

```
String.prototype.endsWith = function(suffix)
{
    return this.indexOf(suffix,
        this.length - suffix.length) !== -1;
};
String.prototype.startsWith = function(prefix)
{
    if (this.indexOf(prefix) == 0)
        return true;
    else
        return false;
};
```


Komponentenbibliothek

HTML5

- DOM
- Browserabhängigkeiten
- Einige Controls
- 2D
- Nicht erweiterbar
- Auf DOM aufbauende „Meta Bibliotheken“

JavaFX

- JavaFX
- Unabhängig v. Browser
- Viele Controls
- 3D, Animationen, ...
- Erweiterbar

Performance

HTML5

- JavaScript:
„nicht mehr
problematisch“
- DOM Operationen:
„langsam“
- Up to date Hardware!

JavaFX

- Java:
„schnell“
- Scene Graph
Operationen: „schnell“
- Up to date Hardware!

Tools

HTML5

- JavaScript Editoren
- Debugger
- Erste Umgebungen zum Profilen

JavaFX

- Entwicklungs-umgebungen
- Debugger
- Profiling
- Dump Analyse
- ...

Entwicklungseffizienz

HTML5

- Native JavaScript Verwendung:
„schlecht“

JavaFX

- „hervorragend“

Deployment

HTML5

- „Zero!“
 - „Zero?“

JavaFX

- Webstart, Applet
 - JRE Preinstallation
 - Administrator
- Bundle
 - setup.msi/ .exe
 - App-like
 - No Administrator

Einflussnahme auf Look&Feel

HTML5

- CSS - „sehr gut“

JavaFX

- CSS - „sehr gut“

Langfristige Stabilität „Framework Volatilität“

HTML5

- Kommen und Gehen von Hypes
- Volatilität im Browser-Umfeld

JavaFX

- Zumindest in einer Tradition der Langfristigkeit
 - Java Swing
 - Java SWT
- Geschlossene(re) Umfeld

HTML5 - Lösung durch Frameworks?

- JavaScript Frameworks
 - jQuery
 - Sencha Touch
 - Qooxdoo
 - ...
- JavaScript Cross Compiler
 - GWT
 - Dart

HTML5 - Lösung durch Frameworks?

- ...zunächst ja:
 - Deutliche Erleichterung im Umgang mit HTML5 und JavaScript
- ...aber auch:
 - Beträchtlich hohe Frameworkstacks
 - Frameworknutzer müssen Frameworks verstehen, um Problemsituationen überhaupt begegnen zu können
 - Abhängigkeit von Framework und dessen Hype-Cycle
- ...und am Ende:
 - Kunde und Endanwender sehen das Endprodukt, nicht das Framework dazwischen...

Zusammenfassung

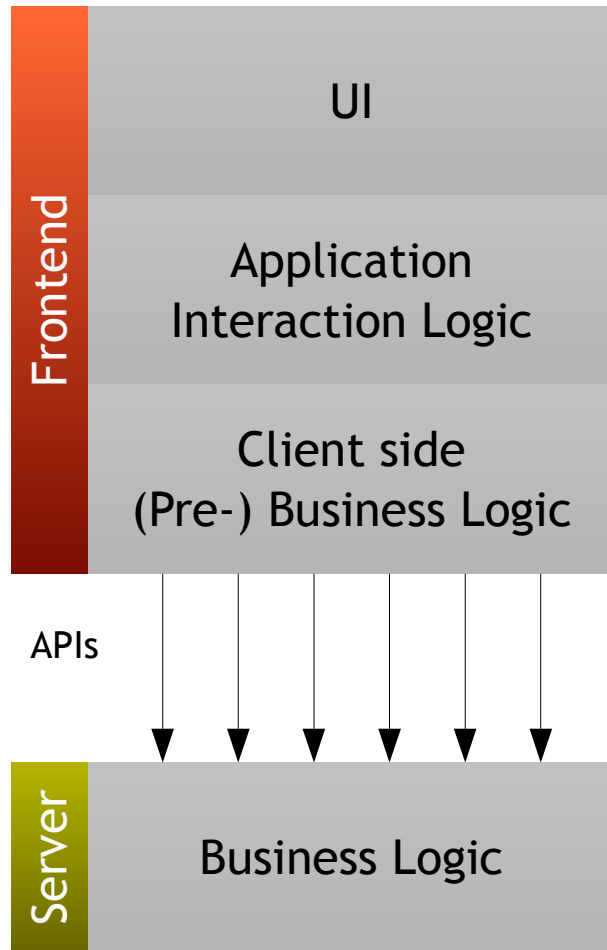
HTML5

- Endgeräte
- Zero Deployment

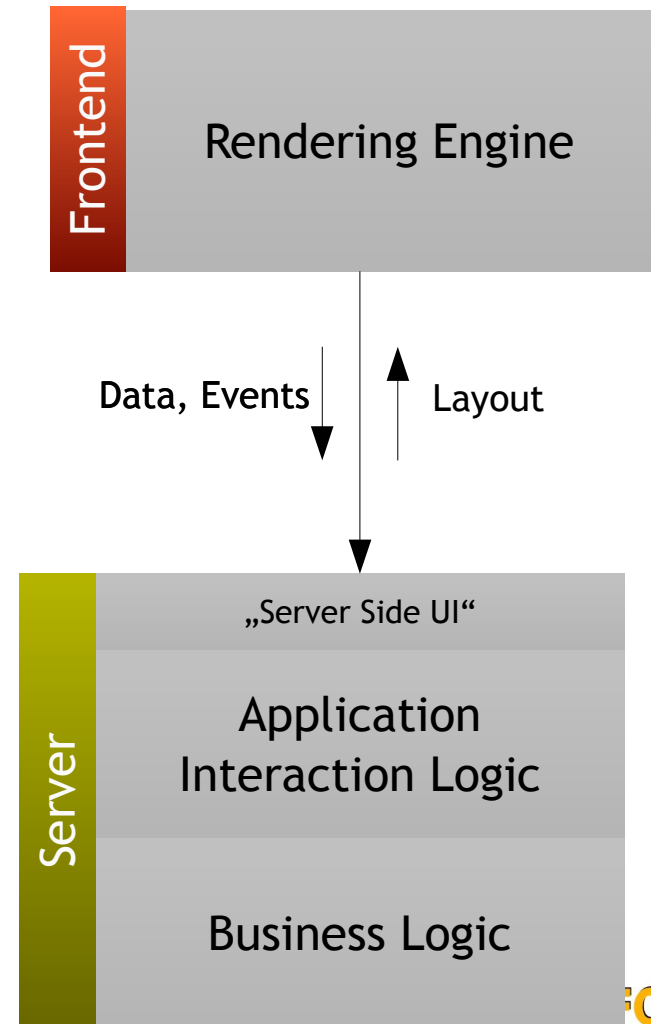
JavaFX

- Entwicklungseffizienz
- Komponenten
- Performance
- Tools
- Langfristige Auslegung

Client Centric

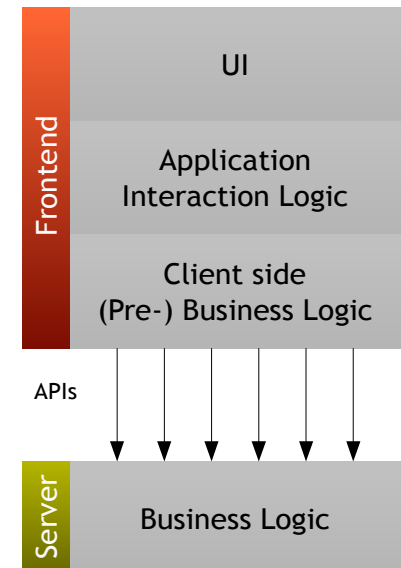


Server Centric



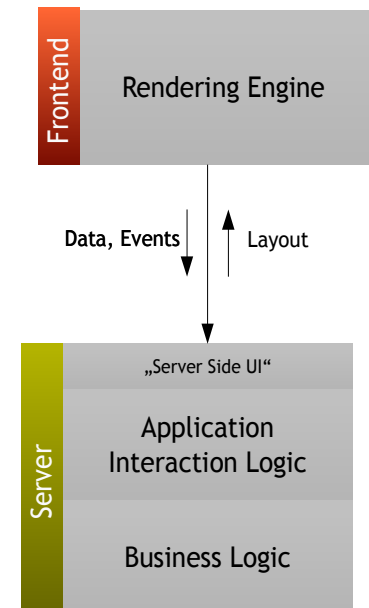
Client Centric

- Pro
 - Direktes Loslegen
 - Voller Zugriff auf UI Möglichkeiten
 - Lokale Datenhaltung möglich
- Contra
 - Viele Schnittstellen zum Server
 - Immer größer werdender Client
 - Frontend und Backendentwicklung
- Sinnvoller Einsatz
 - Kleine(re) Szenarien
 - Hohe Disziplin in der Entwicklung



Server Centric

- Pro
 - Nur ein Ort der Entwicklung
 - Nur eine Schnittstelle
 - Einfaches Deployment
 - Abstraktion vom UI
- Contra
 - Kein direkter Zugriff auf Client
 - Synchronisation mit Logik nur über Roundtrip
- Sinnvoller Einsatz
 - Umfangreiche Anwendungen



Frameworks

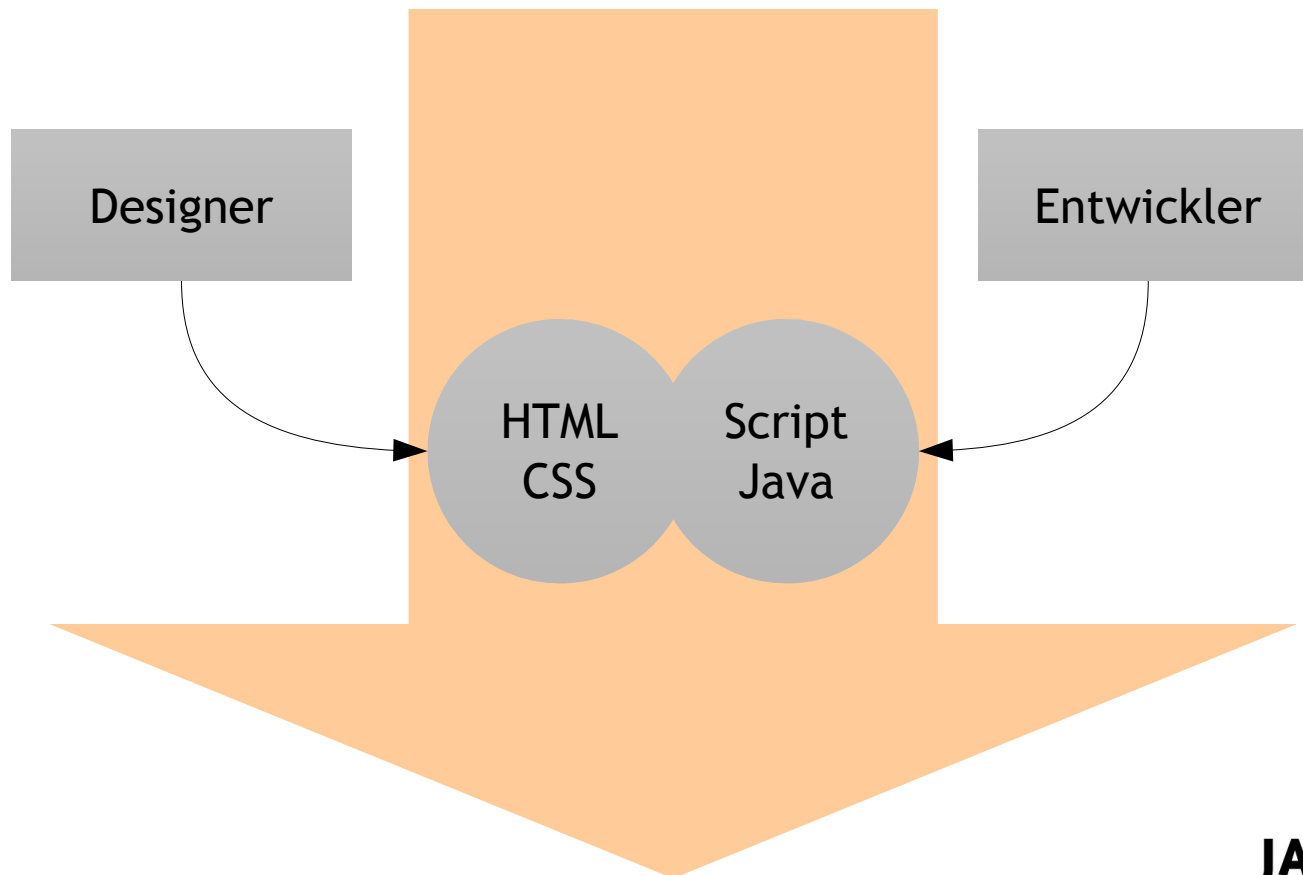
	HTML5	JavaFX
Client centric	GWT jQuery Qooxdoo Sencha Touch	JavaFX Libraries
Server centric	JSF *Faces Vaadin Tapestry	CaptainCasa - JSF Canoo - Dolphin

Szenarien

- Generell
 - Nicht mit Windmühlen kämpfen!
 - Man mag von JavaFX angetan sein - aber es gibt Szenarien, in denen man nicht an HTML5 vorbeikommt
 - Hier gilt: Mehraufwand einkalkulieren!
 - „Zero Installation!“ und „Run everywhere!“ hat seinen Preis
 - Ansatz „Ein Framework für alles!“ passt in der Regel nicht!
 - ...intelligente Architektur für ein Szenario definieren!

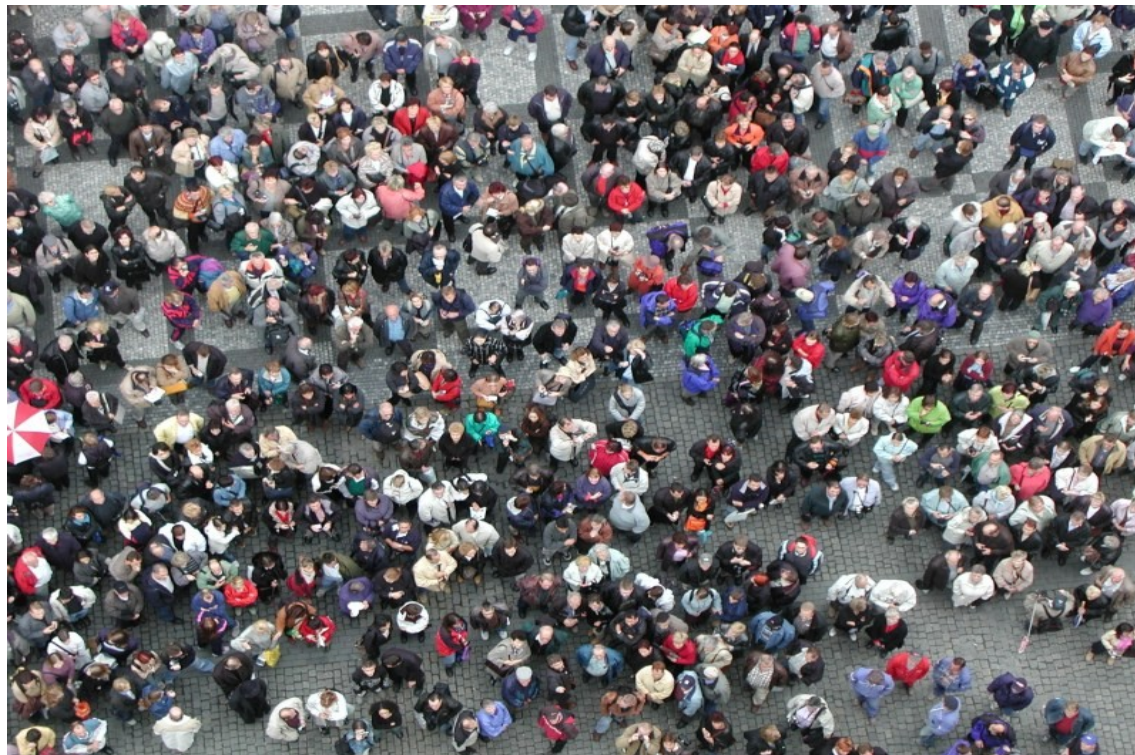
Klare HTML5 Positionierung

- Design-first Szenarien“



Klare HTML5 Positionierung

- Anonyme Endnutzer in großer Zahl
 - Eventuell einmal Konkurrenz durch JavaFX Bundle-App-Installation - je nach Wertigkeit der Anwendung




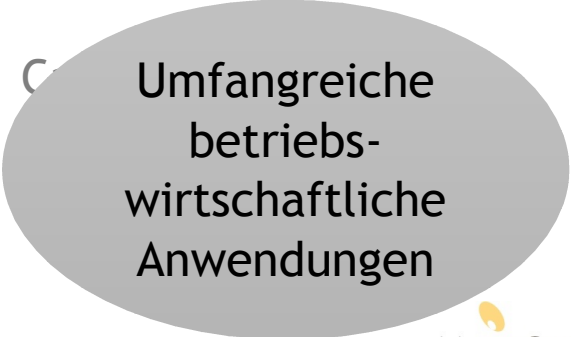
Client-zentrische Anwendungen

	HTML5	JavaFX
Client centric	GWT jQuery Qooxdoo Sencha Touch	JavaFX Libraries
	●	●
Entwicklungseffizienz		
	●	●
Umfang		
	●	●
Desktop		
	●	●
Endgeräte		
	●	●

Server-zentrische Anwendungen

	HTML5	JavaFX
Server centric	JSF *Faces Vaadin Tapestry, ...	CaptainCasa - JSF Cannoo - Dolphin
	●	●
Entwicklungseffizienz	●	●
Installationsaufwand	●	●
Langfristiger Betrieb	●	●
Hype Resistenz	●	●

Sinnvolle Positionierung von JavaFX

	HTML5	JavaFX
Client centric	GWT jQuery Qooxdoo Sencha Touch	 <p>Umfangreiche Clients für Desktop</p> <p>Clients mit besonderer Interaktivität</p>
Server centric	JSF *Faces Vaadin Tapestry	 <p>C</p> <p>Umfangreiche betriebswirtschaftliche Anwendungen</p>

Fazit (I)

- JavaFX ist da!
 - Es gibt eine Alternative zu HTML5 - im Bereich von Desktop Endgeräten
- JavaFX ist effizient(er)
 - Client-centric: ganz klar!
 - Server-centric: keine Nebenkriegsschauplätze!
- JavaFX bedeutet Installation

Fazit (II)

- Gute Einsatzgebiete von JavaFX
 - Funktional mächtige Desktop Clients
 - Möchten Sie wirklich Ihre Logik in JavaScript formulieren?
 - Desktop Clients mit besonderer, grafischer Interaktion
 - Langfristig stabile Desktop Clients für Anwendungssysteme mit Sachbearbeiter/Werker-Bezug

Fazit (III)

- Derzeitige Haupthemmnisse
 - Dominanz von „HTML5 everywhere“
 - Installationsaufwand
 - **Plattformstrategie**
- Irgendwo spielt HTML5 immer eine (berechtigte!) Rolle!
 - „Nur JavaFX“ macht i.d.R. keinen Sinn: wer JavaFX einsetzt, muss gleichzeitig auch beantworten können, wie er/sie HTML basierte Masken bereitstellt

CaptainCasa Community

- Server-zentrische Architektur!
 - Java Server Faces basierte Server-Verarbeitung
- Kernmasken der Anwendung
 - JavaFX Client
- Self-Service / Portal / Mobile
 - Simple HTML5