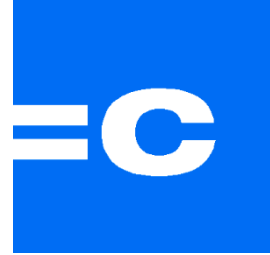




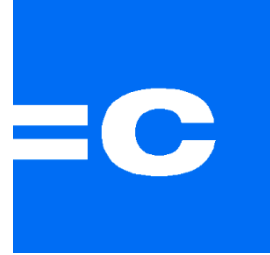
# Mobile Multiplattform-Apps

## 5. c-Akademie

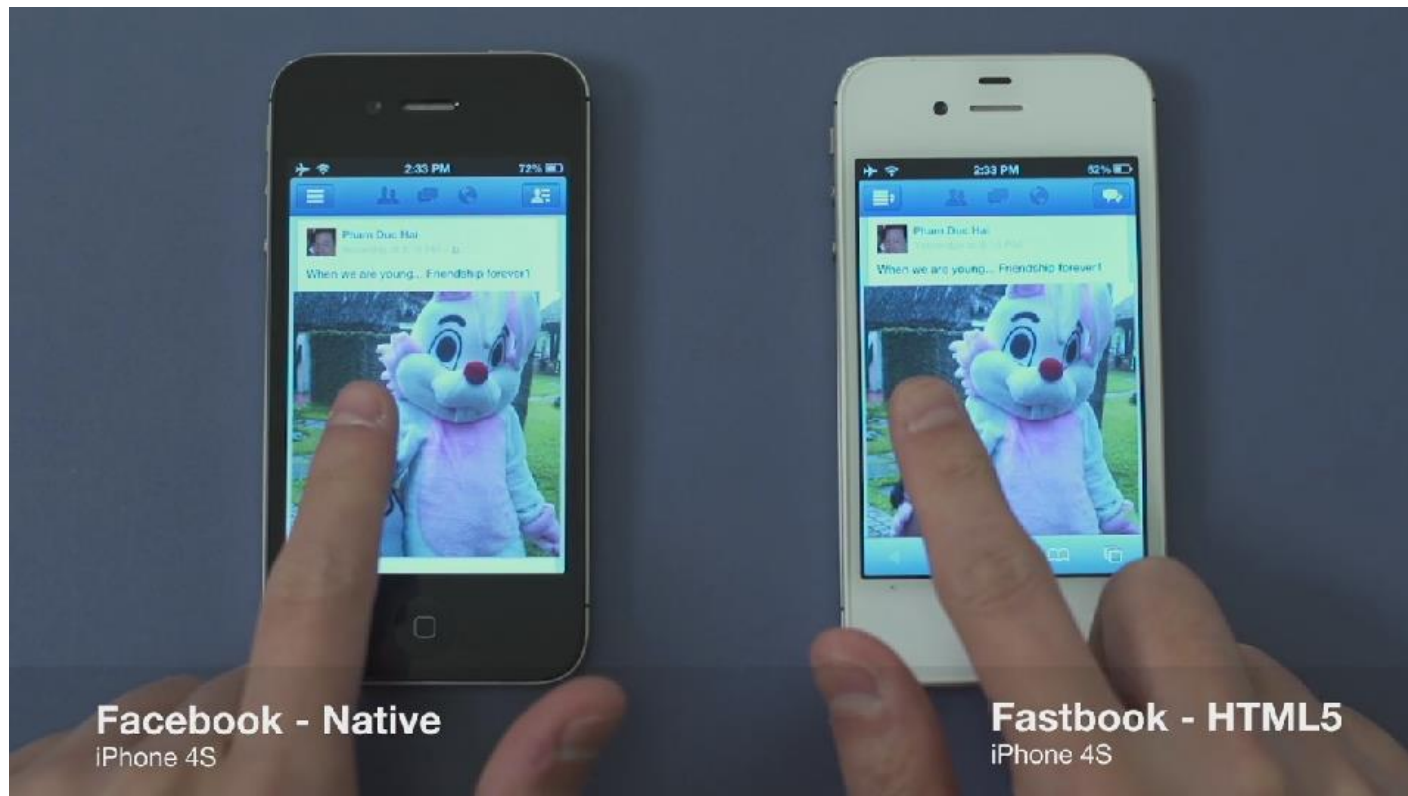
# Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5



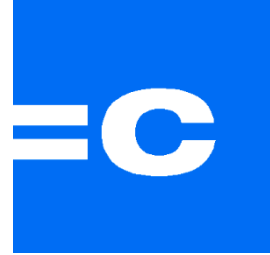
# Antwort Fastbook



- <http://fb.html5isready.com/>

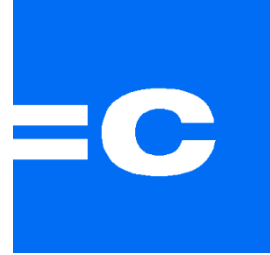


# Ablauf

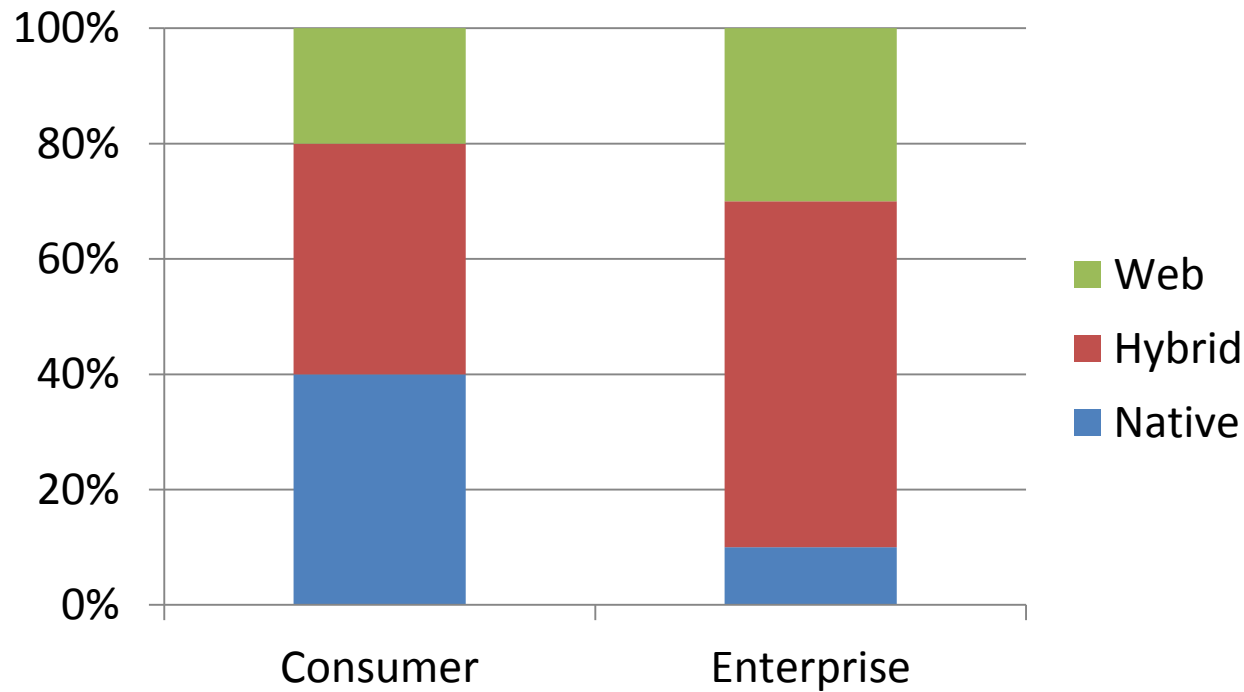


- Einführung Apps
- Architecture Frameworks
- UI Frameworks
- Hybrid Frameworks
- Demo
- Fazit und Ausblick

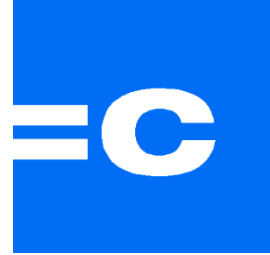
# Was Analysten sagen



- Gartner -> bis 2015 Grossteil aller Enterprise Apps werden als Hybrid entwickelt



# Möglichkeiten eine App zu erstellen



## Native Application

- Voller Zugriff auf Geräte API
- Programmiersprachen
  - Java
  - Objective-C
  - C#



## Web Application

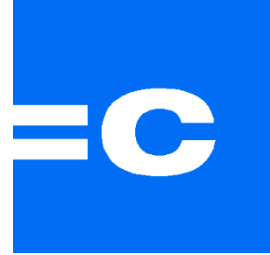
- Eine Codebasis
- Zugriff auf Hardware nur durch HTML5 Browserunterstützung (GPS, Camera)
- Programmiersprachen
  - HTML5
  - JavaScript
  - CSS 3



## Hybrid Application

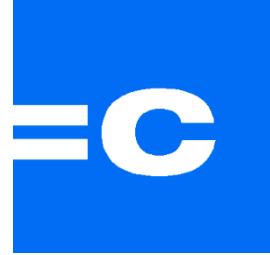
- JS-Native Communication
- App in einem Container verpackt
- Programmiersprachen
  - HTML5
  - JavaScript
  - CSS3

# Hybrid Application?



- Entwickelt mit HTML5; in einen nativen Container gewrapt
- Keine plattformspezifischen Sprachen wie Objective-C oder Java
- Kann über einen AppStore vertrieben werden
- Möglichkeit zum Zugriff auf native Komponenten (Kamera, Mikrofon, Kontakte oder Notification System)

# Pro & Contra - Native App



- + Bessere Performance
- + Flüssigere Animationen, Übergänge und schnellere Startzeit
- + Kann mehr Daten offline speichern
- + Voller Zugriff auf Hardware und OS Features
- Keine gemeinsame Codebasis
- Unterschiedliche API und Programmiersprachen für Zielplattform
- App Updates müssen jedes Mal den Approval-Prozess neu durchlaufen

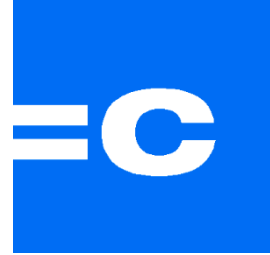


# Pro & Contra - Web App



- + Eine Codebasis, welche von jedem browserbasierten Mobilgerät aufgerufen werden kann
- + Performance Problem werden immer mehr minimiert, da Mobilegeräte und Javascript Engines schneller werden
- + Kein Approval-Prozess benötigt; Apps können sofort geupdatet werden
- Interpretierter Code
- Kein voller Zugriff auf alle gerätespezifischen Möglichkeiten
- Nicht im AppStore auffindbar

# Pro & Contra - Hybrid App



- + App kann über die bedeutenden AppStores vertrieben werden
- + Eine Codebasis für alle unterstützten Plattformen
- + Zugang über die API auf einige, oder sogar alle gerätespezifischen Features, welche über den Browser nicht aufrufbar sind
- Abhängig vom Store Approval-Prozess
- App Performance abhängig durch die Browsercontrol-Leistungsfähigkeit des Mobilgeräts



Users don't care if it's native or  
web as long as the app is fast and  
responsive

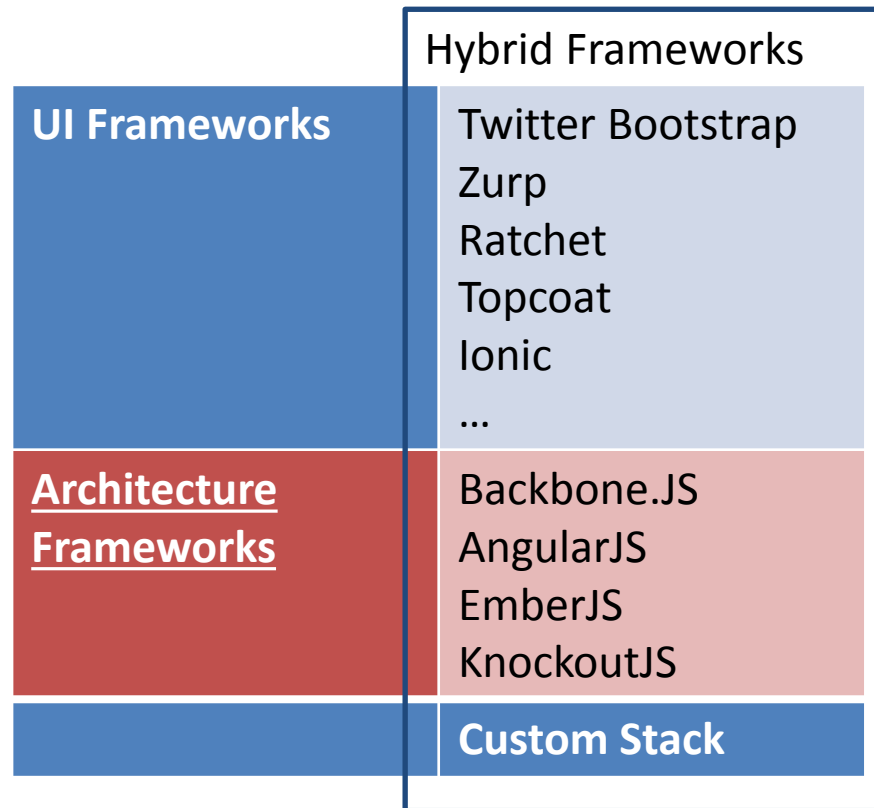
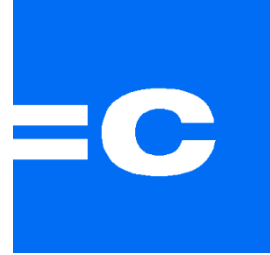
Kristofer Joseph, Javascript  
Frameworks Lead, Adobe

# Consider a Framework



<b>UI</b>	Dojo Toolkit jQuery mobile Sencha Touch KendoUI	<b>Hybrid Framework</b> Twitter Bootstrap Zurp Ratchet Topcoat Ionic ...
<b>Architecture</b>		Backbone.JS AngularJS EmberJS KnockoutJS
	<b>Full Stack</b>	<b>Custom Stack</b>

# Architecture Frameworks

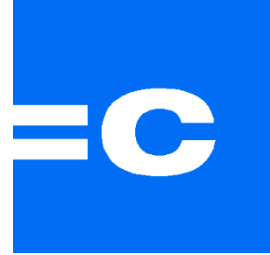


# Anforderungen an (Web-)App

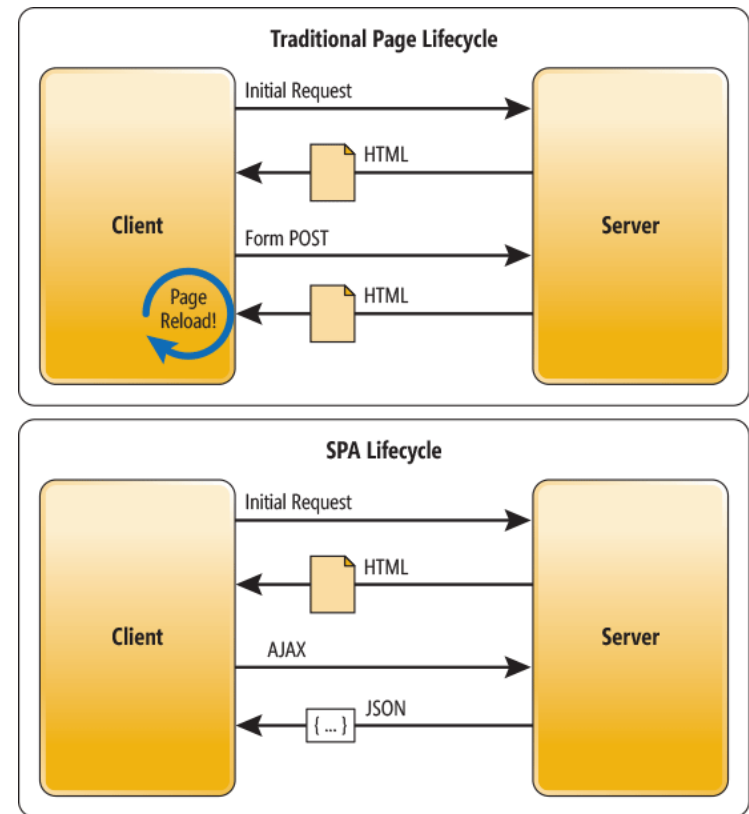


- Möglichst viel Logik auf dem Client
- Serverzugriff möglichst nur für Daten
- Andere Ansicht soll nicht zu komplettem Laden der Seite führen
  
- Gegenläufig zu klassischem HTML
- Kommunikation über AJAX
  - Single Page Application Frameworks

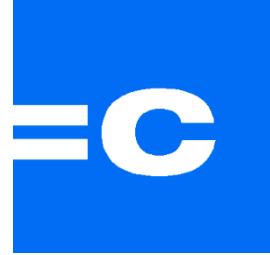
# Single Page Applications



- Nur benötigte Bereiche werden neu geladen
  - Ideal für responsive Designs
  - Komplexere User-Interfaces
  - Weniger Datenverkehr
- Clientseitiges MVC
  - **NUR** Client zuständig für:
    - Darstellung
    - App-Status
  - Klare Strukturen/Kapselung
- Vorgehen wie bei nativer App



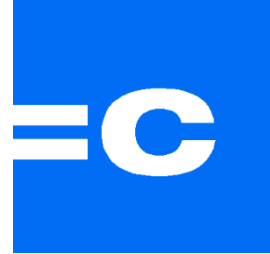
# SPA Frameworks Vergleich



	AngularJS	Backbone.js	Ember.js	Knockout
Dependency Injection	X		X	
Data Binding	X		X	X
RESTful Web Services	X	X		
Promises	X		X	
Localization	X		X	X
Template Engine		X	X	X
Routing	X	X	X	
Einfache JS-Objekte	X			
Benutzerdefiniertes HTML	X		tlw.	
Benutzerdefinierte Events	X	X	X	



# SPA Frameworks Vergleich



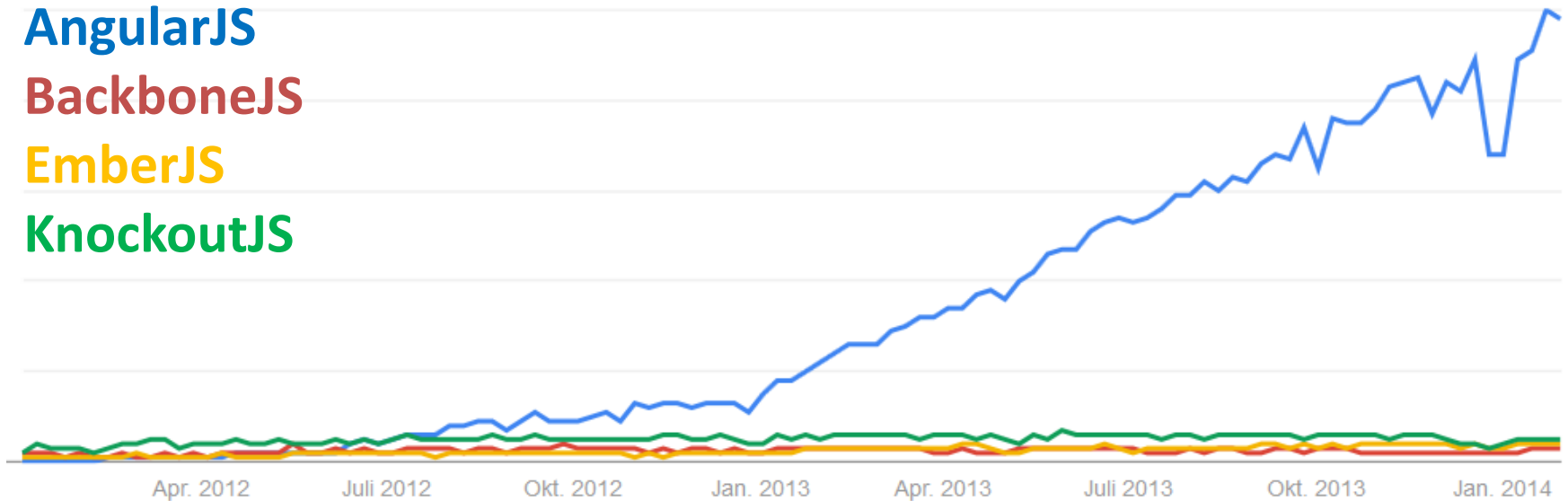
Suchanfragen:

AngularJS

BackboneJS

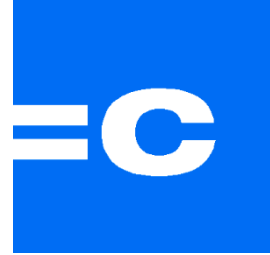
EmberJS

KnockoutJS



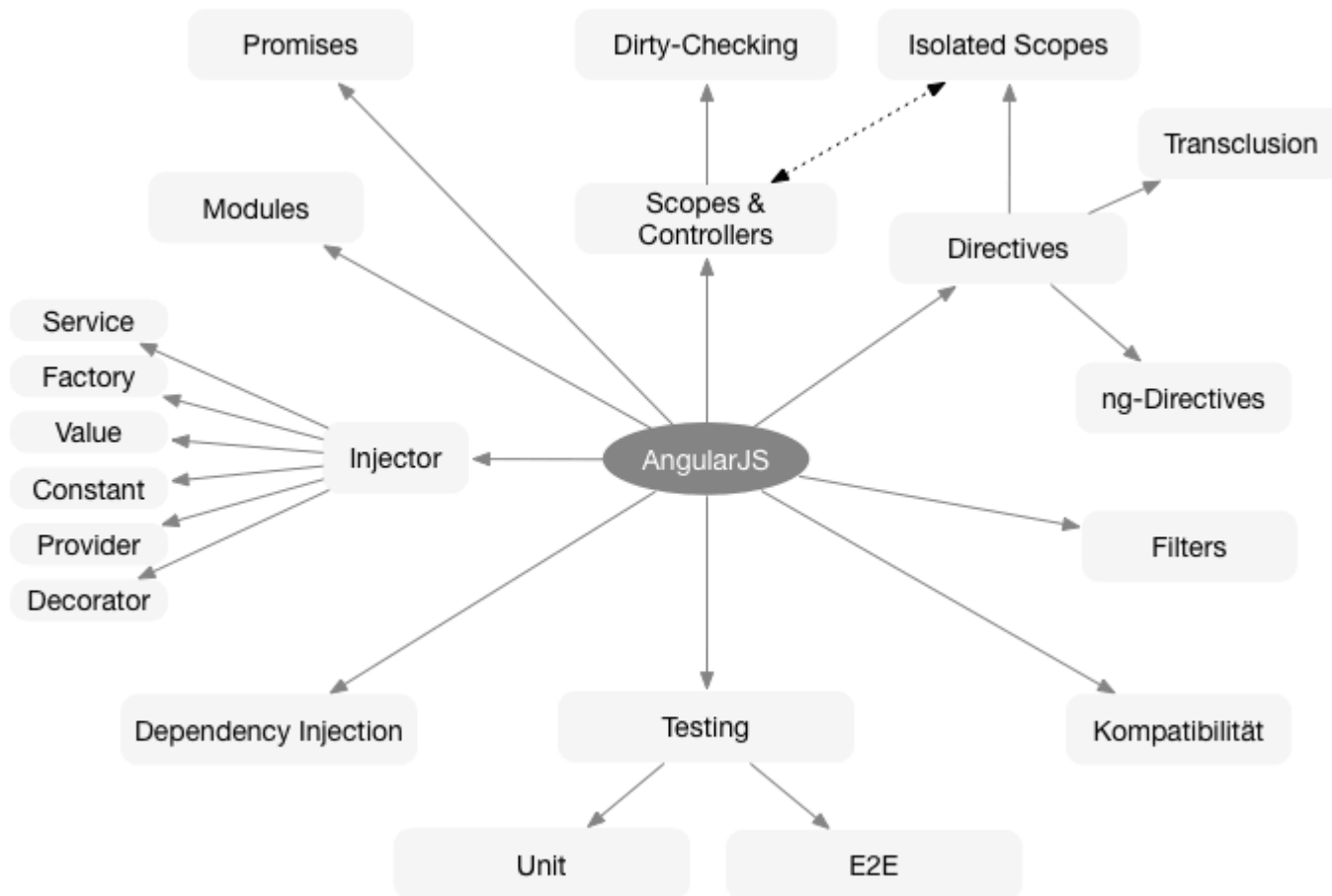
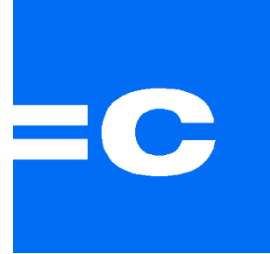
Quelle: Google Trends

# Was ist AngularJS



- JavaScript Framework für Single-Page Applikationen
- MVC/MVVM Framework
  - Komponenten basiert
  - Bidirektionales Binding
- Entwickelt und verwendet von Google
  - Auf Testbarkeit ausgelegt
  - Open Source Projekt

# AngularJS Aufbau



# AngularJS Vorteile



- Vereinfachung von wiederkehrenden Aufgaben
- Optimierte auf Test- und Wartbarkeit
- Arbeiten mit Komponenten
- Keine JavaScript Abhängigkeiten
- Rich-Client Architektur: Webserver wird nur zum Bereitstellen von Daten benötigt

# Bsp. Codereduzierung



- Ausgabe von:

```
var hello = "Hallo Welt!";
```

- „Klassischer“ Ansatz:

```
<span id=„msg“></span>
```

```
document.getElementById(„msg“)  
  .textContent = hello;
```

- The jQuery way:

```
<span id=„message“></span>
```

```
$("#message").text(hello);
```

- The Angular-way:

```
<span ng-bind="hello"></span>
```

# AngularJS

## Dependency Injection



- Services können über eindeutigen Namen global registriert werden
- Derselbe Service wird überall im Programm verwendet („Singleton“)
- Framework stellt bestimmte Services bereit
  - \$http                   AJAX requests
  - \$filter               Such und Sortierfunktionen
  - \$location           Navigation zwischen Views
  - \$q                     Promise API
  - ...

# AngularJS

## Dependency Injection



Deklaration an einer Stelle im Programm:

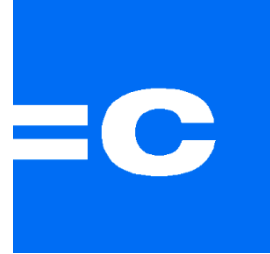
```
moduleA.factory('valueConverter', function() {  
  this.convert = function() { ... };  
});
```

Verwendung an anderer Stelle:

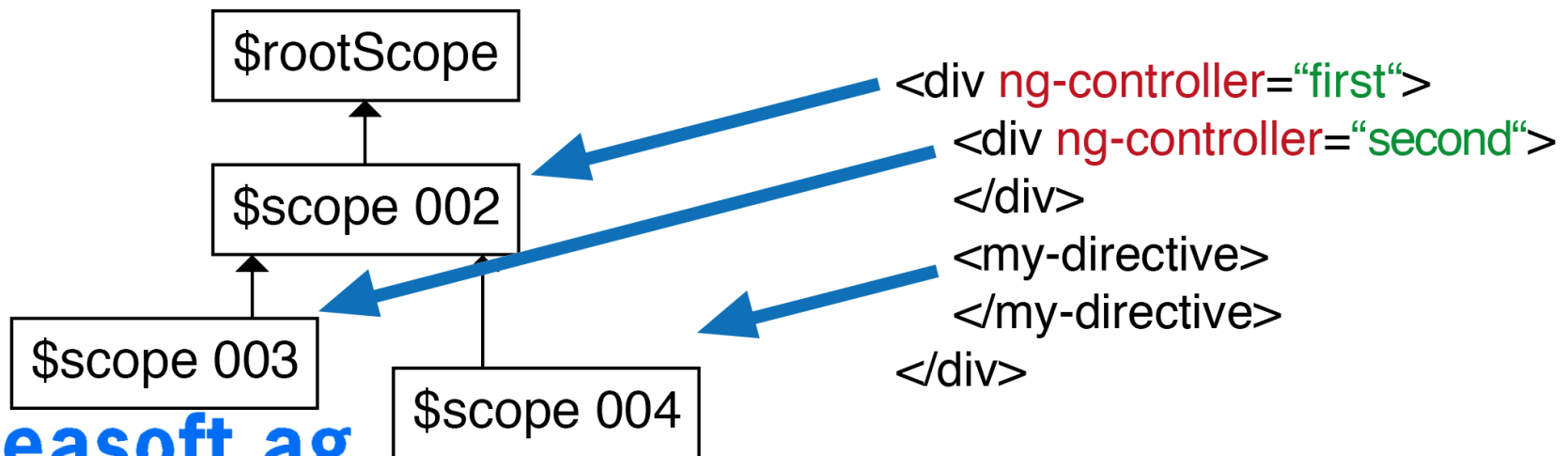
```
moduleB.factory('testService', ['valueConverter', function(vc) {  
  this.abc = function() {  
    return vc.convert();  
  }  
});
```

# AngularJS

## Scopes und Events



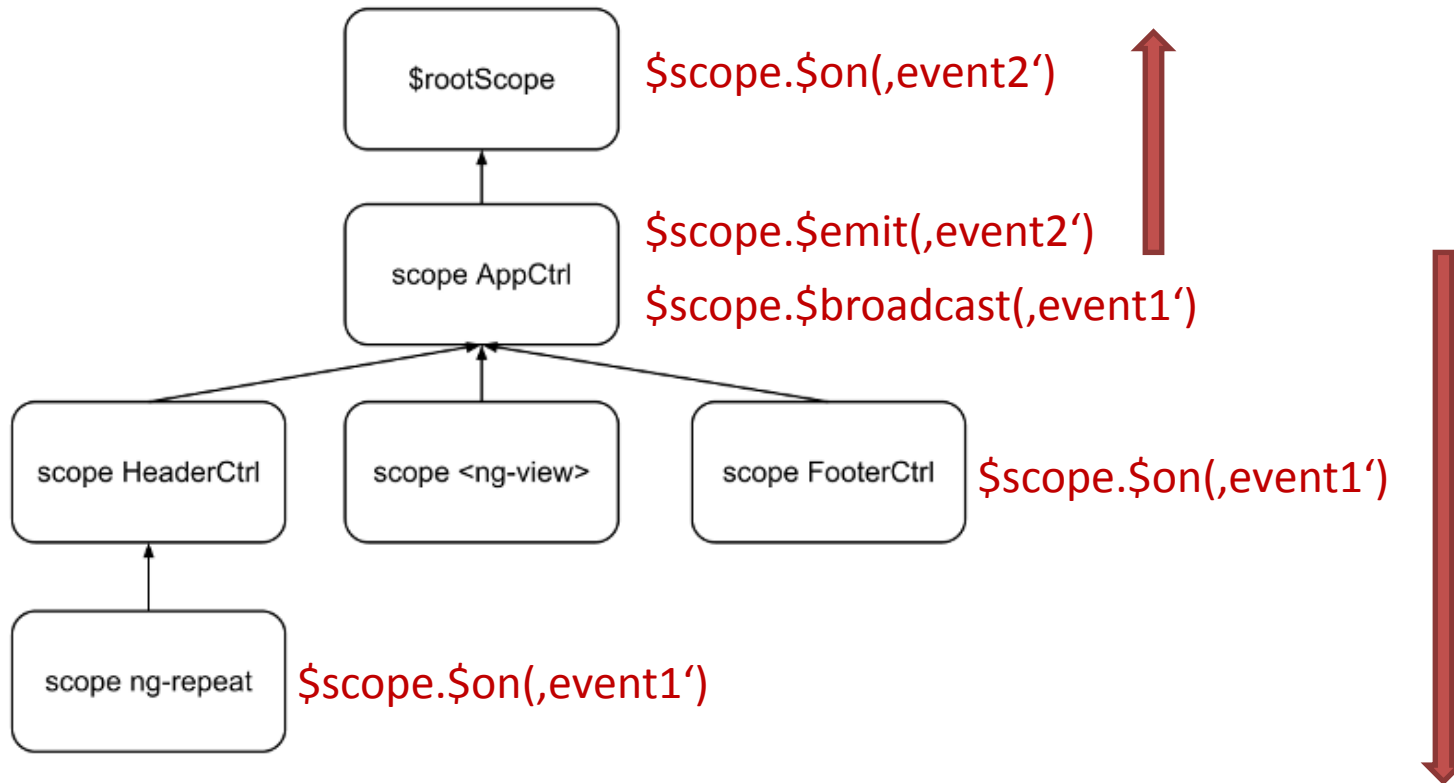
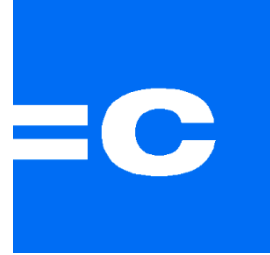
- Bindeglied zwischen Controller und View
- Definiert Methoden und Properties die in View bereitstehen
- Setzt Änderungen an Properties in View um
- Erlaubt es Änderungen zu überwachen
- Hierarchisch organisiert



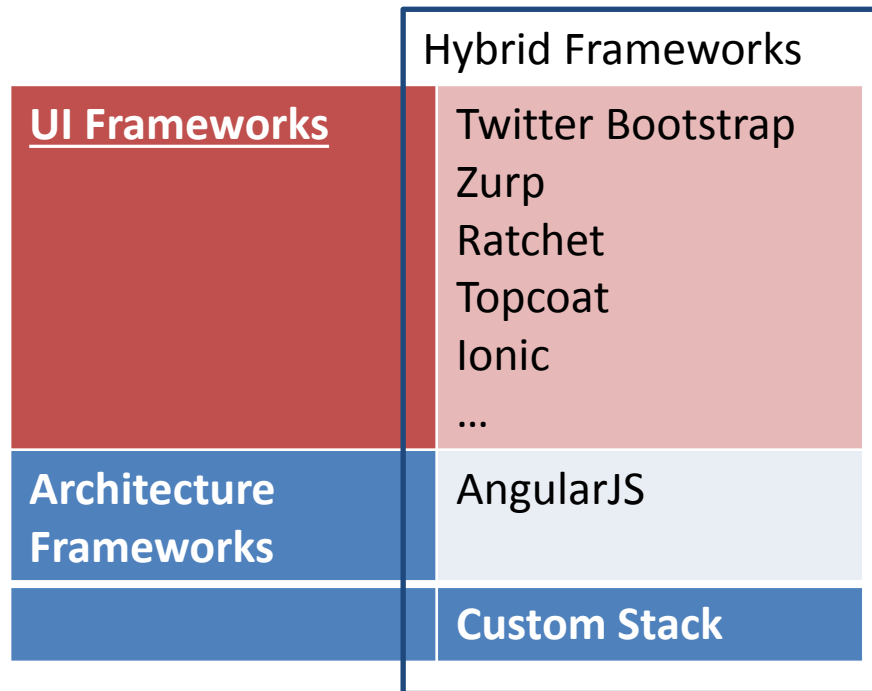
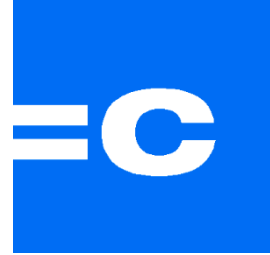


# AngularJS

## Scopes und Events



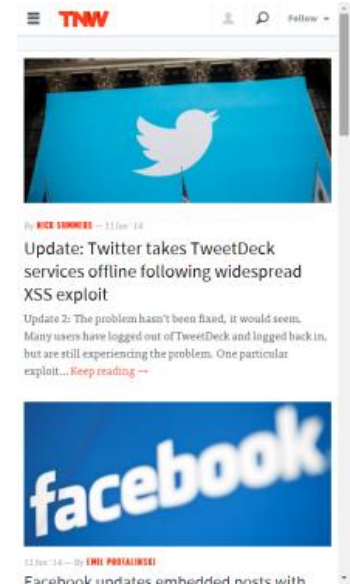
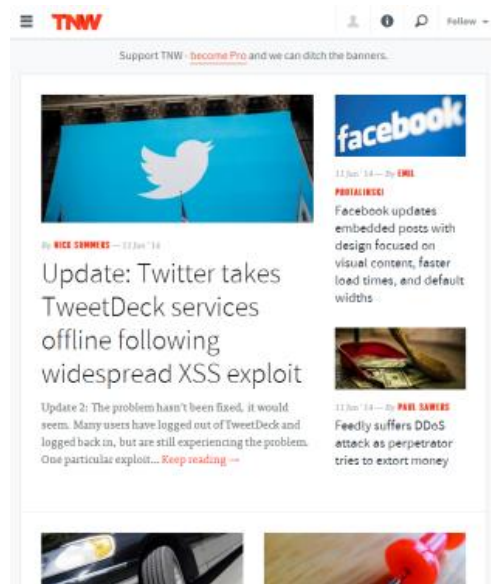
# UI Frameworks



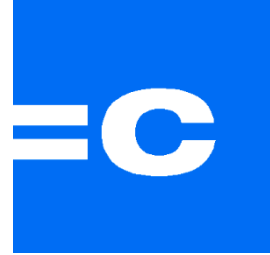
# UI Frameworks



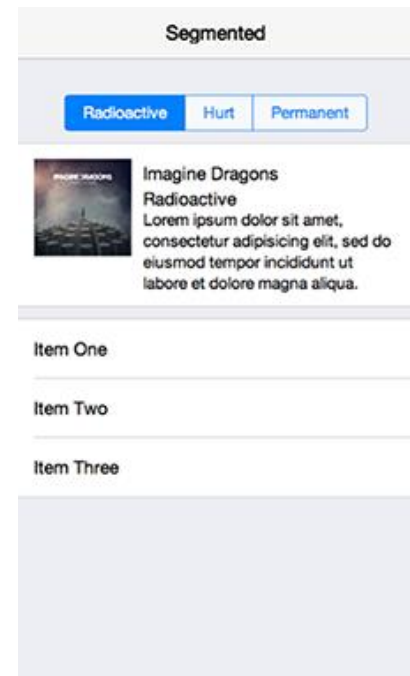
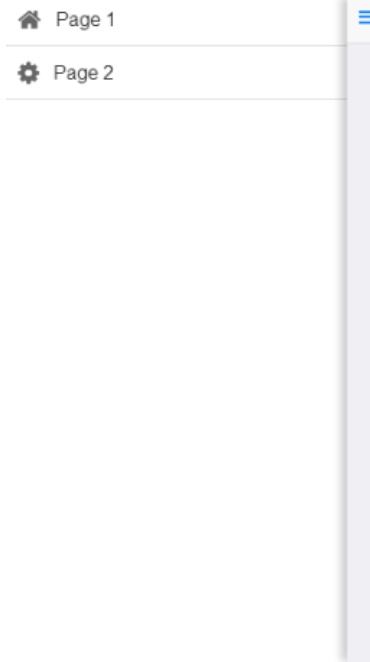
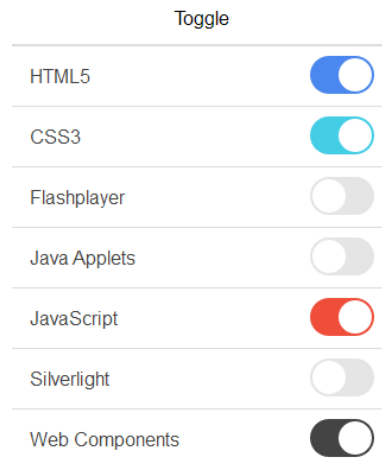
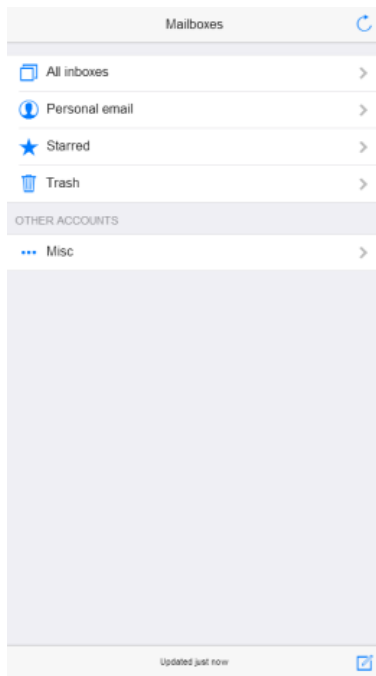
- Responsive CSS3 Frameworks



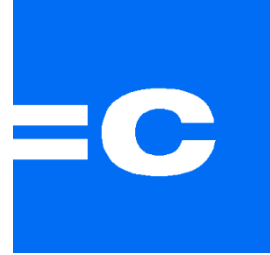
# UI Frameworks



- Mobile optimierte CSS3 Frameworks

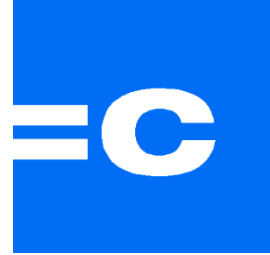


# Vergleich Mobile UI Frameworks

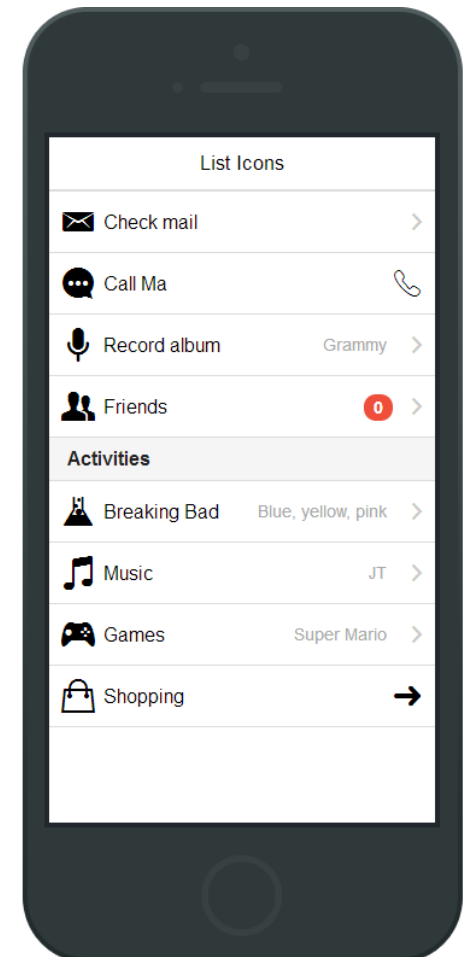


Framework	Android	iOS	WindowsPhone	Blackberry OS
ChocolateChip-UI	+	+	+	
KendoUI	+	+	+	+
Ratchet	+	+		
Ionic	+	+		
Topcoat	+	+		
PhoneJS	+	+	+	
OnsenUI	+	+		

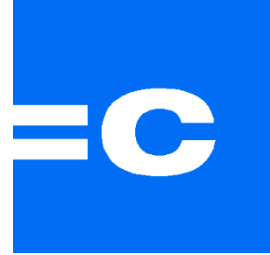
# Ionic



- Designed für leichtgewichtige Mobile Apps
  - Keine Web Pages
- Built mit LESS
- Basiert auf AngularJS

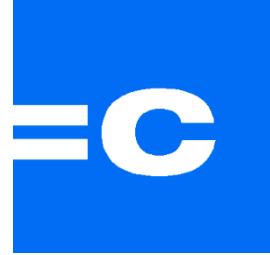


# Hybrid Frameworks



<u>Hybrid Frameworks</u>	
UI Frameworks	Ionic
Architecture Frameworks	AngularJS
	Custom Stack

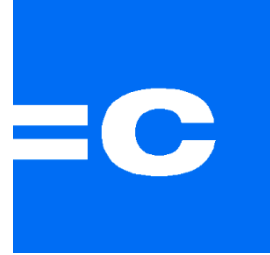
# Cross Mobile Möglichkeiten



<b>Native</b>	Android, iOS, Windows Phone	Skills wird für jede Plattform und Sprache benötigt
<b>Cross Compiled</b>	AppLause Mono/Xamarin	Non-UI Code kann wiederverwendet werden UI muss teilweise neu geschrieben werden
<b>Interpreted</b>	Appcelerator RhoMobile	App wird gegen eigene API entwickelt
<b>Hybrid</b>	PhoneGap Cordova	Web App wird innerhalb eines nativen Container gehostet
<b>Web Apps</b>	Twitter Bootstrap	Responsiver Aufbau der Website Kein «Native»-Look&Feel



# Vergleich



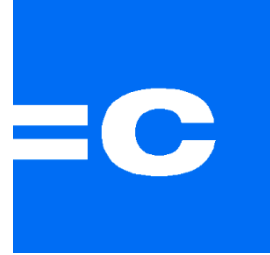
	Phonegap	Titanium	Mono/Xamarin
Zugriff auf Gerätefunktionen	X	X	X
Natives User Interface		~	X
Programmiersprache	JavaScript, HTML, CSS	JavaScript	C#
Plattformen	iOS Android Blackberry Windows Phone Ubuntu Phone Firefox OS Tizen	iOS Android Blackberry	iOS Android Windows Phone

# Phonegap Geschichte



- Entwickelt von Nitobi 2009
- Aufgekauft durch Adobe 2011
- Aktuell unter Apache Software Foundation
  - Apache Cordova
- Phonegap == Apache Cordova

# Phonegap

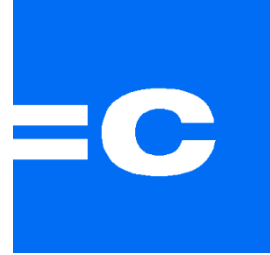


- Container für Web-Apps
  - Startet Webanwendung wie eigene native App
  - Icon im Menü
  - Keine Internetverbindung nötig
  - Zusätzliche Funktionalität wird bereitgestellt
- Genauso als Website einsetzbar



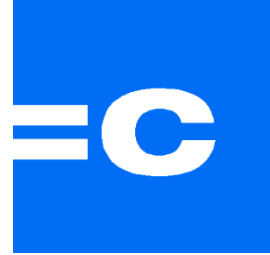
	amazon- fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp7 (Windows Phone 7)	wp8 (Windows Phone 8)	win8 (Windows 8)	tizen
<b>cordova CLI</b>	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓ Windows	✓	✗
<b>Embedded WebView</b>	✓ (see details)	✓ (see details)	✗	✗	✓ (see details)	✓	✗	✗	✗	✗
<b>Plug-in Interface</b>	✓ (see details)	✓ (see details)	✓ (see details)	✗	✓ (see details)	✓	✓ (see details)	✓ (see details)	✓	✗
<b>Platform APIs</b>										
<b>Accelerometer*</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Camera*</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Capture*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
<b>Compass*</b>	✓	✓	✓	✗	✓ (3GS+)	✓	✓	✓	✓	✓
<b>Connection*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
<b>Contacts*</b>	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
<b>Device*</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Events</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
<b>File*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
<b>Geolocation*</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Globalization*</b>	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗
<b>InAppBrowser*</b>	✓	✓	✓	✗	✓	✓	✓	✓	(partial)uses iframe	✗
<b>Media*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
<b>Notification*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
<b>Splashscreen*</b>	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
<b>Storage</b>	✓	✓	✓	✗	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓

# Phonegap Entwicklungsprozess



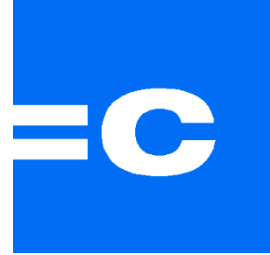
- App nur mittels HTML, CSS und JavaScript erstellen
- Konfiguration mittels XML-Datei
- Wenige spezielle Anpassungen für einzelne Plattformen vornehmen
- Install-Pakete kompilieren und installieren
  - Je nach Plattform unterschiedlich
  - Teils abhängig vom Entwicklungs-PC
  - Oder...

# Phonegap Build

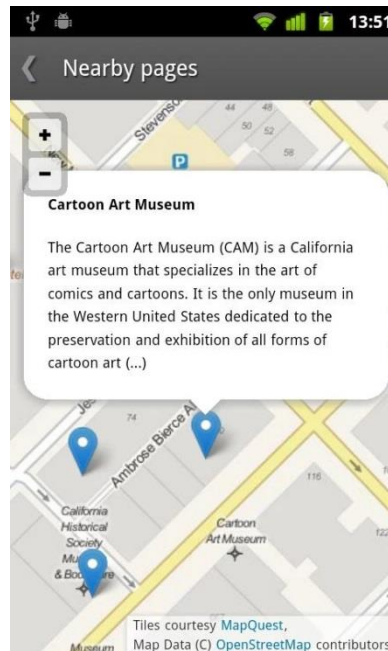
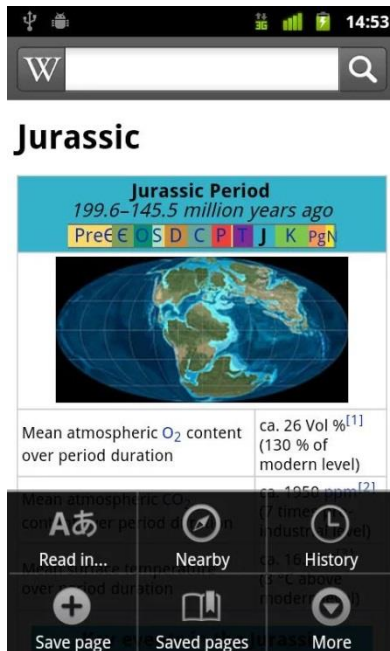


- Online Compiler
- Erstellen von Apps für versch. Plattformen
  - Ohne spezielle Hardware
  - Entwickler Lizenzen weiterhin benötigt
- „Hydration“
  - Neue Version wird online erstellt
  - Bei App-Start wird automatisch nach Update gefragt

# Phonégap Beispiele

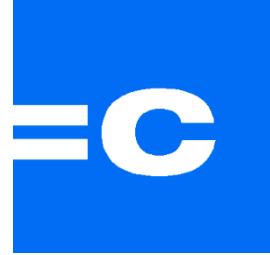


- Wikipedia

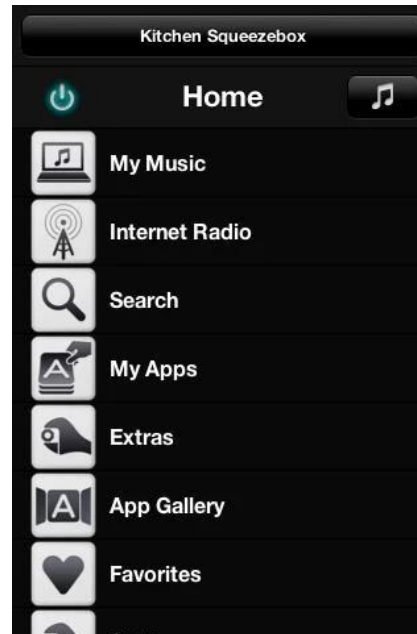


**Water** is a chemical substance with the

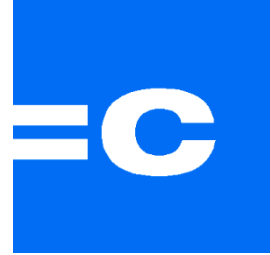
# Phonégap Beispiele



- Logitech Squeezebox Controller

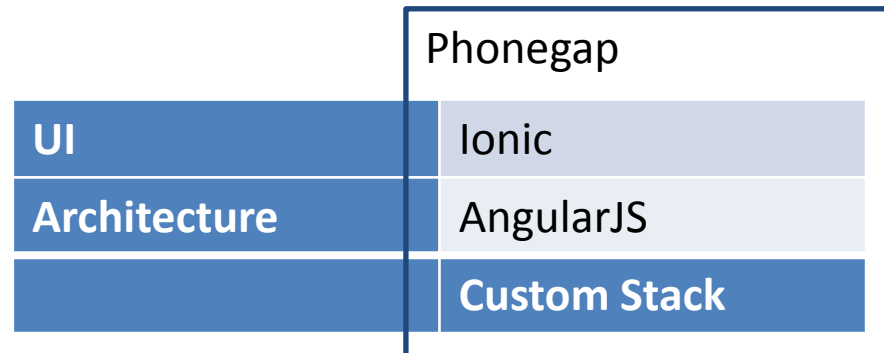






# Demo

## TodoApp mit AngularJS, Ionic und Phonegap

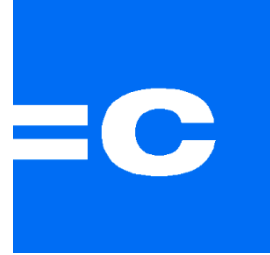


# Fazit



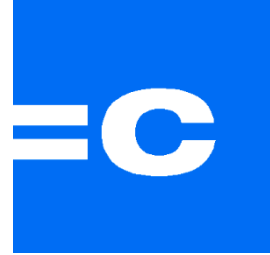
- Soll meine App als Hybrid erstellt werden?
  - Nein, wenn bereits in native Technologien investiert wurde und kein Bedarf vorhanden ist auf neue mobile Plattformen zu wechseln
  - Nein, wenn die App komplex und hoch interaktiv ist
  - Ja, wenn die App einfache Funktionen unterstützen soll
  - Ja, wenn die App mehrere Plattformen unterstützen soll

# Fazit

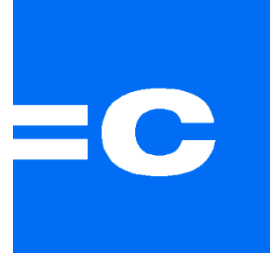


- Hybrid Ansatz weil,
  - Wartungskosten
  - Zugang zu Entwicklern
  - Time to market

# The Future



- Besserer Support für HTML5
- Zunahme von optimierten JS Engines
- Optimiertes Rendering
- Leistungsstärkere Mobilgeräte
- Ease of development
- Cross-Plattform



Fragen?

Vielen Dank für Ihre Aufmerksamkeit