



# Programmierung mobiler Geräte

SoSe  
2015

## Hybride Apps

**Markus Berg**

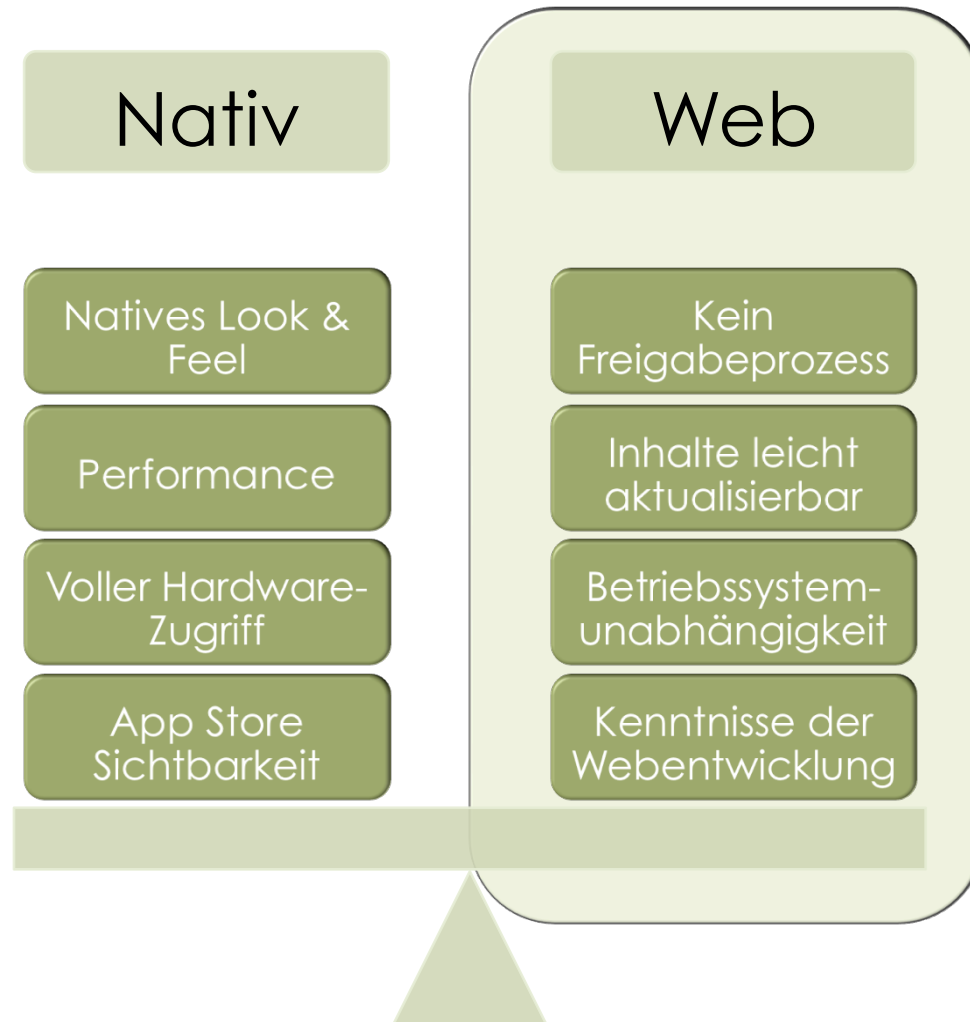
Hochschule Wismar

Fakultät für Ingenieurwissenschaften

Bereich Elektrotechnik und Informatik

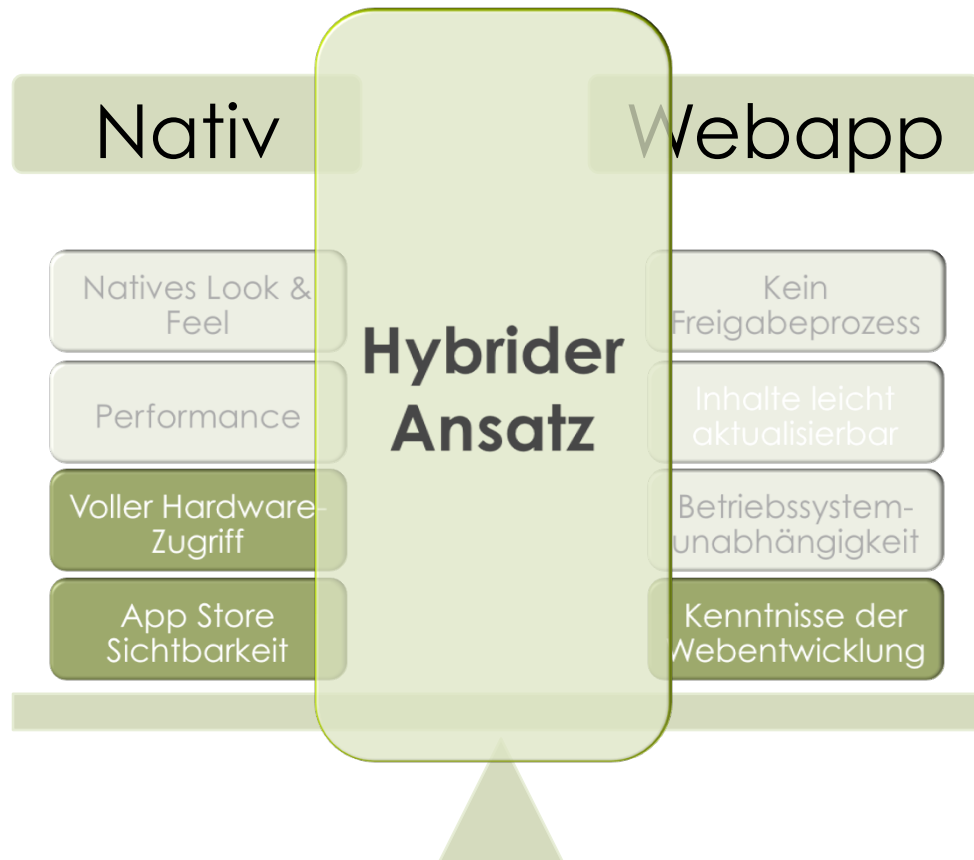
<http://mberg.net>

# Letzte Woche: Webapps



# Heute: Hybride Apps

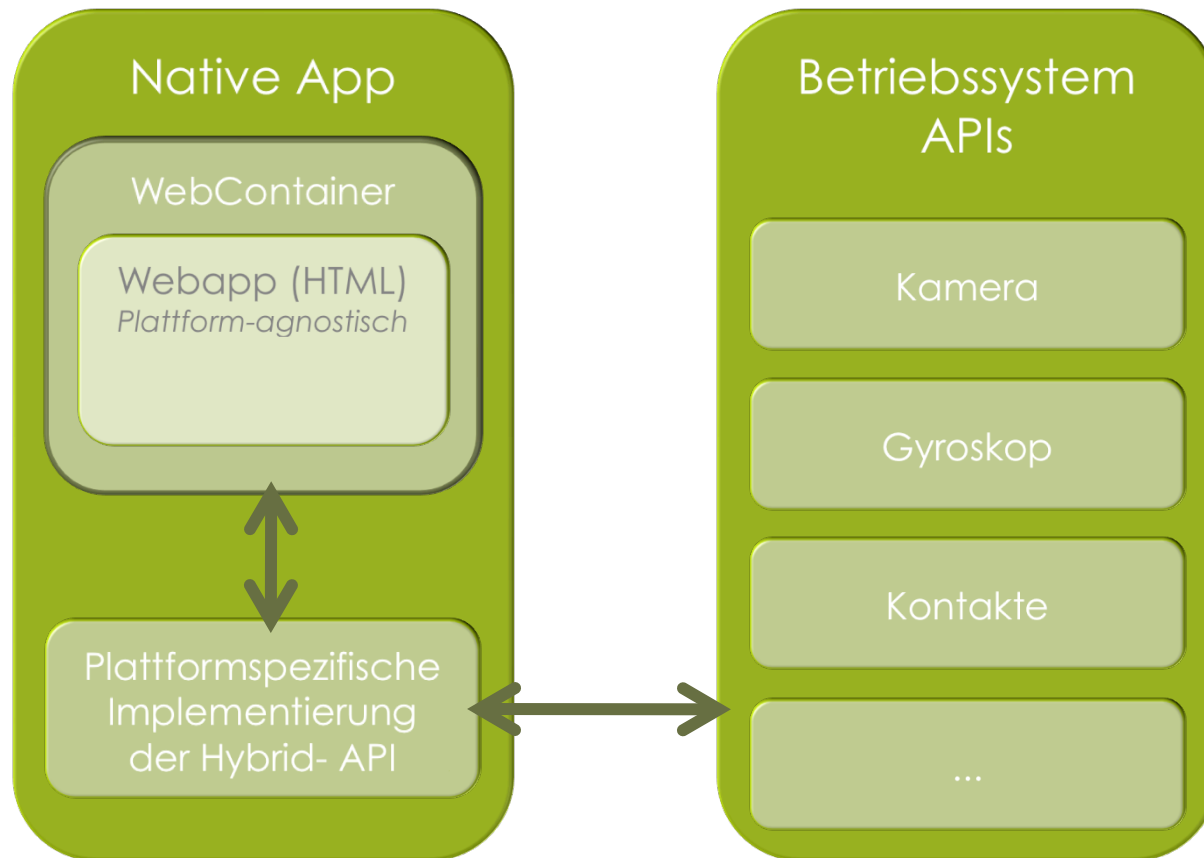
Nutzung von **Webtechnologien**  
in Kombination mit einem **nativen Container**



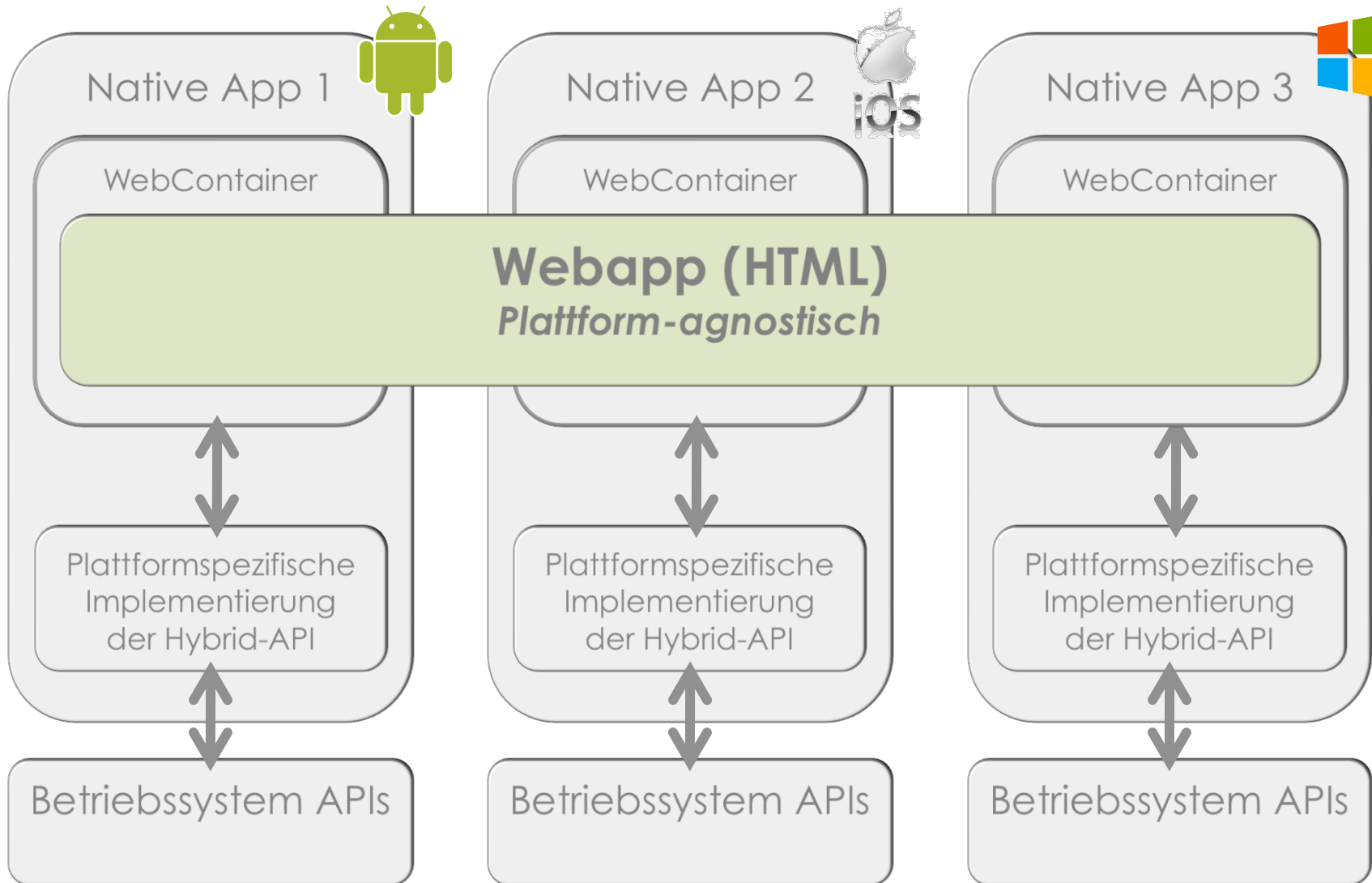
# Idee

- Native App mit einem WebContainer, der HTML-Seiten darstellen kann
  - Anzeige einer Webapp
  - Keine Darstellung von URL-Leiste etc. (Browser nicht erkennbar)
- Nativer Code zum Zugriff auf Hardware bzw. spezifische Funktionen des Betriebssystems
  - Betriebssystemspezifische Implementierungen einer API für hybride Apps
- Java Script Funktionen zur Interaktion der Webseite mit dem nativen Teil der App
  - Nativer Teil vermittelt zwischen dem Betriebssystem und dem Webcontainer (bzw. der Webapp im Webcontainer)

# Hybrid App



# Hybrid App



# Vorteile

- Store-Sichtbarkeit
  - Apps verkaufbar
- Eine einzige Quellcodebasis
- Im Gegensatz zur Webapp keine Internetverbindung notwendig
  - HTML-Datei wird mit ausgeliefert (statt sie auf einem Server bereitzustellen)
- Trotz Verwendung von Webtechnologien Zugriff auf Gerätefunktionen möglich

# Nachteile

- Geringere Performance
  - Kommunikation über Zwischenschicht (Webkomponente)
- Kein natives Look & Feel
  - Fühlt sich nicht immer „echt“ an (subjektiv), d.h. Nutzern fällt der Unterschied zu nativen Apps auf
- Nach wie vor interpretierte Sprache
  - Fehler fallen erst zur Laufzeit auf
  - Limitierte Unterstützung durch IDE
- Browser interpretieren Code teilweise unterschiedlich
  - Wird meist vom Framework kompensiert

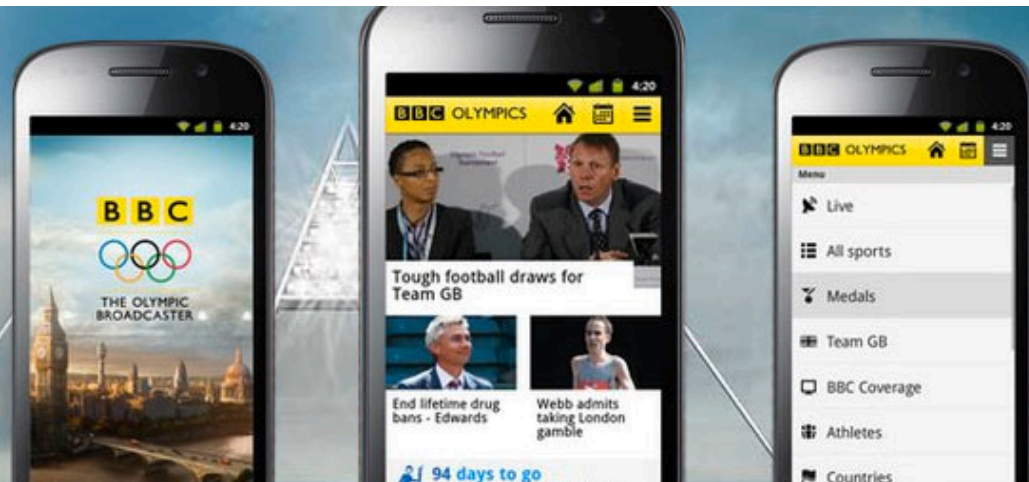


# Prognose

■ Laut Gartner werden 2016 **50% aller Apps hybrid** sein  
[<http://www.gartner.com/newsroom/id/2324917>]

## ■ Prominente Beispiele

- LinkedIn
- Netflix
- BBC (BBC Olympics)



# Phonegap vs. Cordova

- 2009 erfunden als „Phonegap“ von Natobi
  - Entwickelt sich schnell zu einem der bekanntesten Frameworks für Hybride Apps
- 2011 aufgekauft von Adobe
- Adobe übergibt Phonegap an Apache
- Aus rechtlichen Gründen Apache Open Source Version umbenannt in „Cordova“
- Das „neue“ Phonegap basiert nun auf Cordova
  - Beinhaltet zusätzliche Tools
  - U.a. Build Service (Cloud Compilation für andere Plattformen)

# Phonegap: Features

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	Windows Phone 8	Ubuntu	Firefox OS
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	✓	✓	X
Network	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓

<http://phonegap.com/about/feature/>

# Setup: Android

- Wenn nicht vorhanden, Java (JDK 7) installieren
- Android SDK installieren und Systemimage herunterladen
  - Am besten Android Studio installieren (benötigen wir später ohnehin)
    - <http://developer.android.com/sdk/index.html>
  - SDK Manager starten und `tools`, `platform-tools`, `build-tools` sowie eine Android Version (Plattform SDK + System-Image) hinzufügen
    - Für Phonegap benötigen wir API 19 (Android 4.4.2)
    - Wir wählen das SDK, und sowohl das x86- als auch das ARM System Image
- Evtl. HAXM (Intel Hardware Accelerated Execution Manager) installieren
  - Hardware-unterstützte Virtualisierungseine (Hypervisor)
  - In Verbindung mit x86 Android-Images
  - Achtung: nicht auf alten Prozessoren ohne Intel Virtualization Technology

# Setup: Phonegap

- Siehe: [http://docs.phonegap.com/en/4.0.0/guide\\_platforms\\_android\\_index.md.html#Android%20Platform%20Guide](http://docs.phonegap.com/en/4.0.0/guide_platforms_android_index.md.html#Android%20Platform%20Guide)
- Apache Ant installieren und Umgebungsvariable hinzufügen
- Node.js installieren
  - <https://nodejs.org/>
- Phonegap hinzufügen
  - <http://phonegap.com/install/>
  - ```
sudo npm install -g phonegap
```
- Umgebungsvariablen anpassen (Bsp.: Mac OS)
  - ```
export PATH=$PATH:/Users/markus/Library/Android/sdk/tools  
export PATH=$PATH:/Users/markus/Library/Android/sdk/platform-tools
```
- Virtual Device hinzufügen (AVD Manager)
  - Mit SDK 19

# Phonegap: Projekt anlegen

## ■ Projekt anlegen

```
phonegap create HelloGap
cd HelloGap
phonegap platform add android
phonegap build android #only build android...
phonegap build #...or build all added platforms
```

## ■ Starten im Emulator

- Erst Emulator per AVD starten, dann:

```
phonegap run android --emulator
```

## ■ Starten auf Gerät

- Unter Windows müssen erst entsprechende USB-Treiber installiert werden
  - <http://developer.android.com/tools/extras/oem-usb.html>
- USB Debugging auf Gerät aktivieren
- Gerät verbunden? `adb devices`
- App ausführen: `phonegap run android`

# Phonegap

- Code liegt im Ordner **www**
- Startseite der Anwendung wird in **config.xml** eingestellt

```
<content src="quotes/index.html" />
```

Referenziert z.B. */HelloGap/  
www/quotes/index.html*

- Nativer Code wird im Ordner **platforms** erzeugt
- APK wird im Ordner **out/production** angelegt

# Übung: Demo

- Wir starten die Beispiel-App



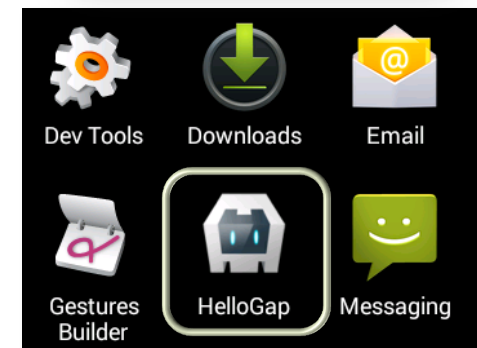
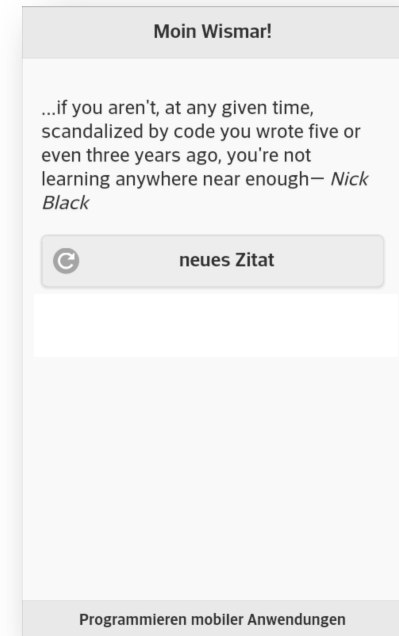


# Übung: Eigene Inhalte

- Wir kopieren unsere Zitate-App der letzten Woche in den Ordner www (Pfad sollte nur aus Kleinbuchstaben bestehen) und starten sie per Phonegap

```
phonegap run android --emulator
```

- Wir haben nun eine hybride App erzeugt!
  - Sie liegt als apk vor
  - Sie lässt sich im PlayStore veröffentlichen
  - Wir sehen keinen Browser
  - Wir mussten keinen nativen Code verwenden
  - Wir haben ein Icon auf dem Homescreen



# Phonegap & AJAX

- Oft kein Zugriff auf Inhalte aus dem Web möglich
  - Z.B. REST-Interfaces
- Daher im ersten Schritt der Anwendung Eventhandler für *mobileinit* definieren und dort:
  - **CORS aktivieren** (Cross Origin Resource Sharing)
    - AJAX Requests zu fremden Servern ermöglichen

```
$(document).bind("mobileinit",function() {  
    $.support.cors = true;  
    $.mobile.allowCrossDomainPages = true;  
});
```

# Zusammenfassung: Hybride App erzeugen

- Wir erzeugen eine hybride App aus unserer Zitate-Webapp

```
phonegap create QuotesApp
cd QuotesApp
phonegap platform add android
phonegap build android
```

- Kopieren der Quelldateien nach www
  - Ggf. Anpassen der config.xml (Startseite)
- Ggf. aktivieren von CORS
- Starten

```
phonegap run android --emulator
```

# Plugins

- Ein großer Vorteil von hybriden Apps ist das Ausführen von nativem Code
- Hierzu müssen Aufrufe in der „Web-Welt“ an Android weitergereicht werden
  - Kommunikation der WebView mit der nativen Plattform
- Dies geschieht mit Hilfe von Phonegap Plugins
  - z.B. Kompass

```
phonegap plugin add org.apache.cordova.device-orientation
```
  - z.B. Accelerometer

```
phonegap plugin add org.apache.cordova.device-motion
```
  - Fügt Code im Ordner **plugins** hinzu
    - Nativer Code
    - Javascript

# Phonegap-JavaScript

- Phonegap-JS laden (in HTML-Datei, z.B. index.html)

```
<script src="../../cordova.js"/>
<script src="../../cordova_plugins.js"/>
```

- *Device-Ready-Listener* hinzufügen (vorher kein Zugriff auf Phonegap/Gerätefunktionen möglich/garantierbar)

```
$(function() {
    document.addEventListener("deviceready", onDeviceReady, false);
});
```

- *EventHandler* definieren

```
function onDeviceReady() {
    //do Phonegap-related work here
};
```

# Übung: Beschleunigung

- Wir erweitern die Zitate-App um folgende Funktionalität:
  - Bei hoher Beschleunigung in z-Achse soll das nächste Zitat geladen werden
    - In Ruhelage ist der Wert 9,81
    - Wird kleiner im Fall
  - X und Y sind in Ruhelage 0
    - Wenn Handy flach auf dem Tisch liegt
  - Lässt sich nicht im Emulator simulieren



# Übung: Beschleunigung

## □ Plugin installieren

```
phonegap plugin add org.apache.cordova.device-motion
```

## □ Plugin nutzen

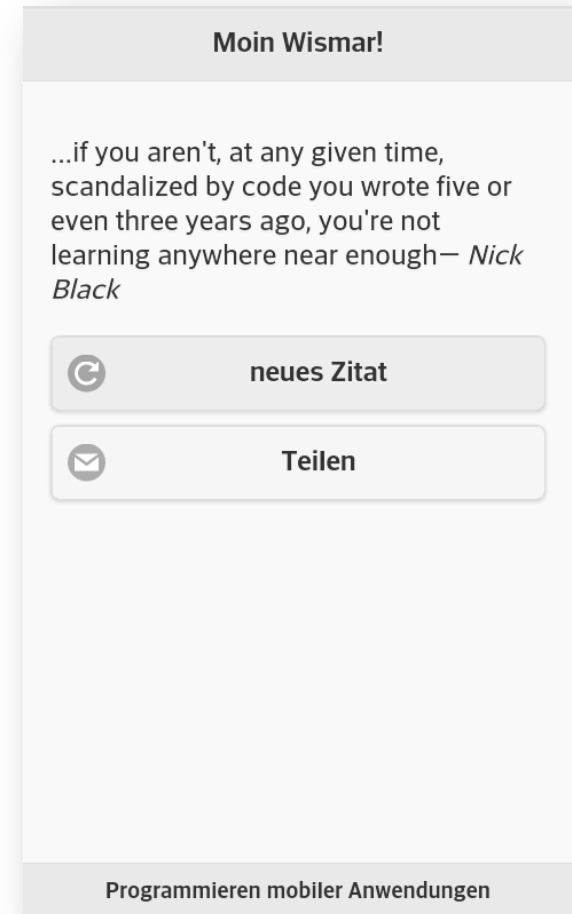
```
navigator.accelerometer.watchAcceleration(onSuccess, onError,  
options);
```

## □ Naives Ermitteln von Werten, sodass Aktion nur ausgelöst wird, wenn Gerät kurz und kräftig nach unten/ schräg hinten bewegt wird

```
if(acceleration.x<5 && acceleration.z<5.5 && acceleration.y<7){  
    ...  
}
```

# Kleine Demonstration

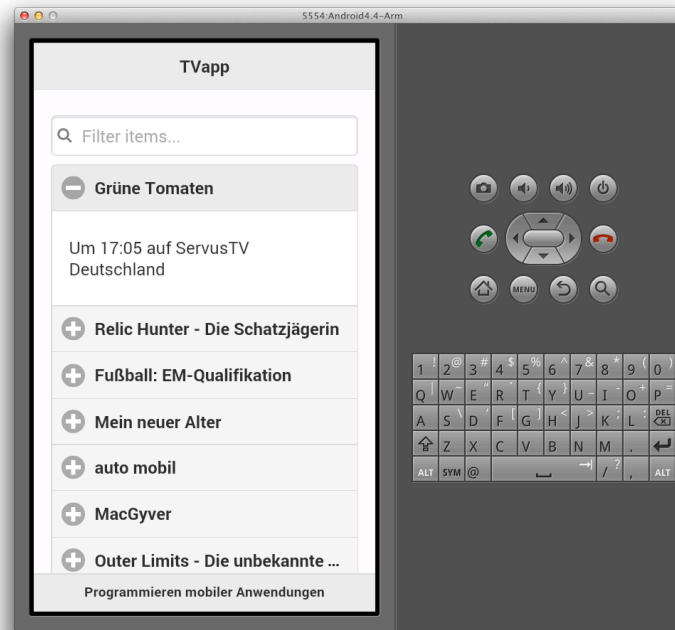
- Integration des Beschleunigungssensors nicht sichtbar (logisch!)
- Integration eines Plugins zum Teilen des Zitates per SMS, Mail etc.
  - Nutzen wir auch im Praktikum





# Praktikum: Teil 1

- Schritt 1: Erzeugen Sie eine hybride App aus der TVApp des letzten Praktikums!



# Praktikum: Teil 2

- Erweitern Sie die App nun um die Funktionalität, beim Anzeigen der Details kurz zu vibrieren
- Fügen Sie einen Button zum Teilen der ausgewählten Sendung (z.B. per Whatsapp) hinzu

```
> phonegap plugin add https://github.com/EddyVerbruggen/  
SocialSharing-PhoneGap-Plugin.git  
> phonegap prepare
```

- Abgabe bis zum 28.04.2015

# Nächste Woche

- Eine kurze Java-Wiederholung als Vorbereitung zu Android

