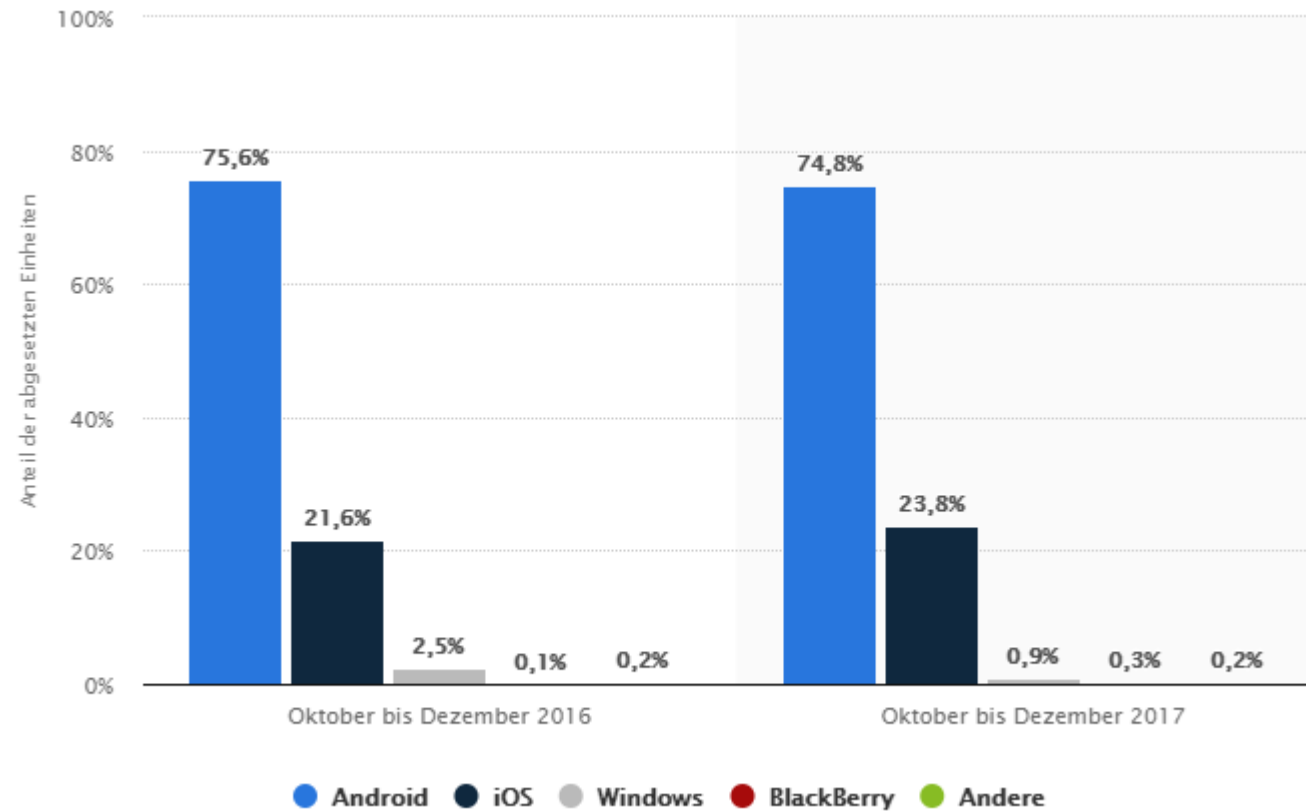


APACHE  
CORDOVA™

# Inhaltsverzeichnis

- Marktanteil Smartphone-Betriebssysteme
- Mobile Overview
- Was ist Cordova/PhoneGap?
- Architektur
- Nachteile/Vorteile
- Setup PhoneGap
- Aufbau eines PhoneGap Projekts
- Cordova Plugin API
- UI Frameworks
  - Ionic
  - Ionic Setup
- Performance
- Sicherheit
- Debugging & Testing
- Bauen der hybriden- App
- Aufgabe
- Quellen

# Marktanteil Smartphone-Betriebssysteme in Deutschland



Quelle: <https://de.statista.com/statistik/daten/studie/198435/umfrage/marktanteile-der-smartphone-betriebssysteme-am-absatz-in-deutschland/>

# Mobile Overview

- **Native App**
  - Für eine spezifische Plattform
  - Installation über den App Store
  - Kann alle Gerätefunktionen ansprechen
  - Entwicklung/Wartung muss für jede Plattform durchgeführt werden
- **Mobile Web App**
  - Laufen im Webbrowser des Geräts
  - HTML 5, CSS3, JavaScript
  - Spricht mehrere Plattformen anhand einer Code-Basis an
  - Benutzer muss mithilfe des Browsers zur Seite navigieren
- **Hybride App**
  - Mit mobilen Web-App-Technologien entwickelt aber wie eine native App in einem Wrapper ausgeführt
  - Spricht mehrere Plattformen anhand einer Code-Basis an
  - Zugriff auf Gerätefunktionen aber mit Begrenzungen
  - Installation über den App Store
  - Entwicklertools: Apache Cordova/ PhoneGap

Objective C oder  
Swift ...



Java

# Was ist Cordova/PhoneGap?

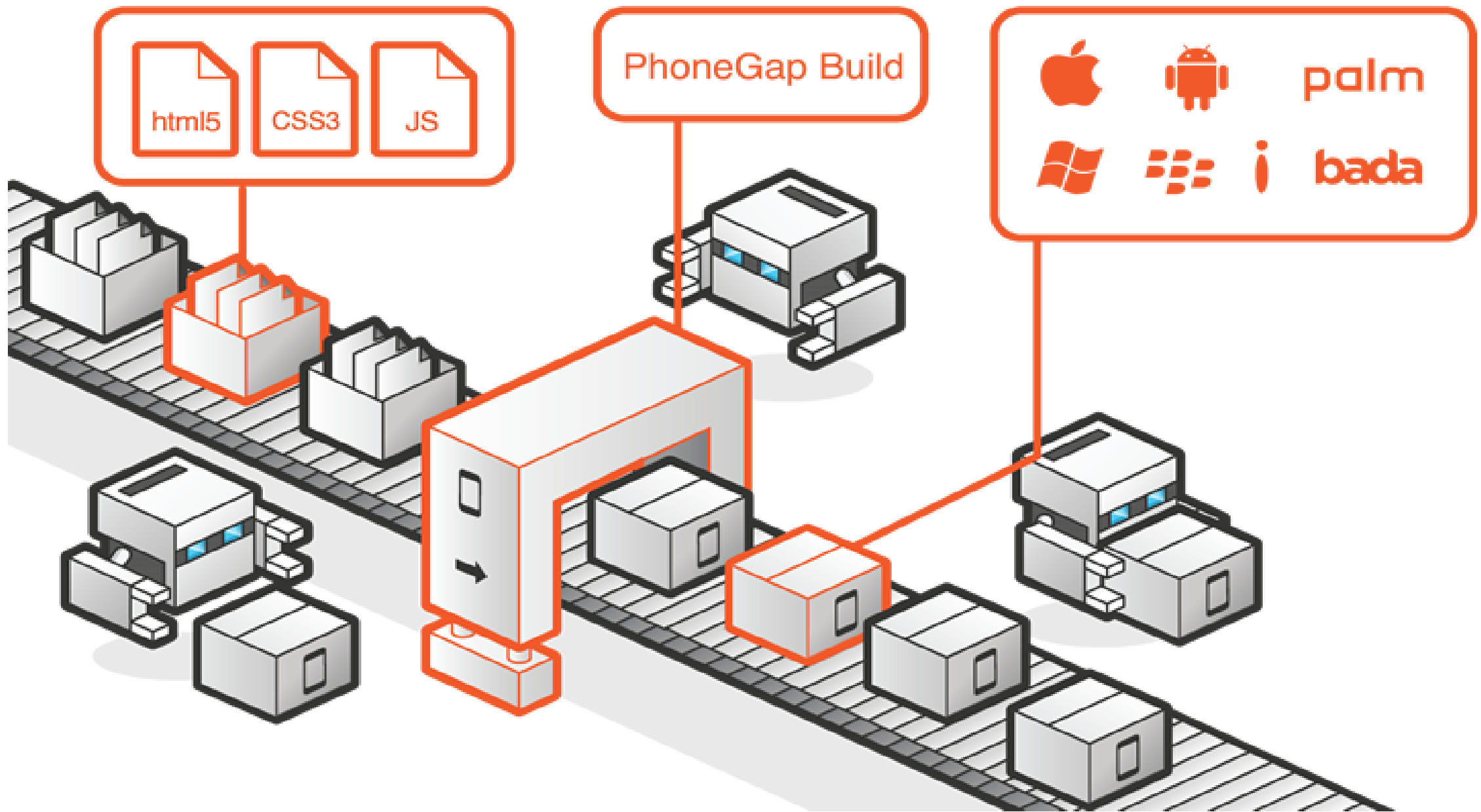
- Apache Cordova ist ein Open-Source-Entwicklungsframework für mobile Applikationen
- Ermöglicht das plattformübergreifende Entwickeln von mobilen Applikationen mittels CSS3, HTML5 und JavaScript
- Zugriff auf native Gerätefunktionen/-sensoren
- PhoneGap ist eine Distribution von Apache Cordova
- PhoneGap stellt Tools zur Verfügungen, die den Entwicklungsprozess vereinfachen/beschleunigen



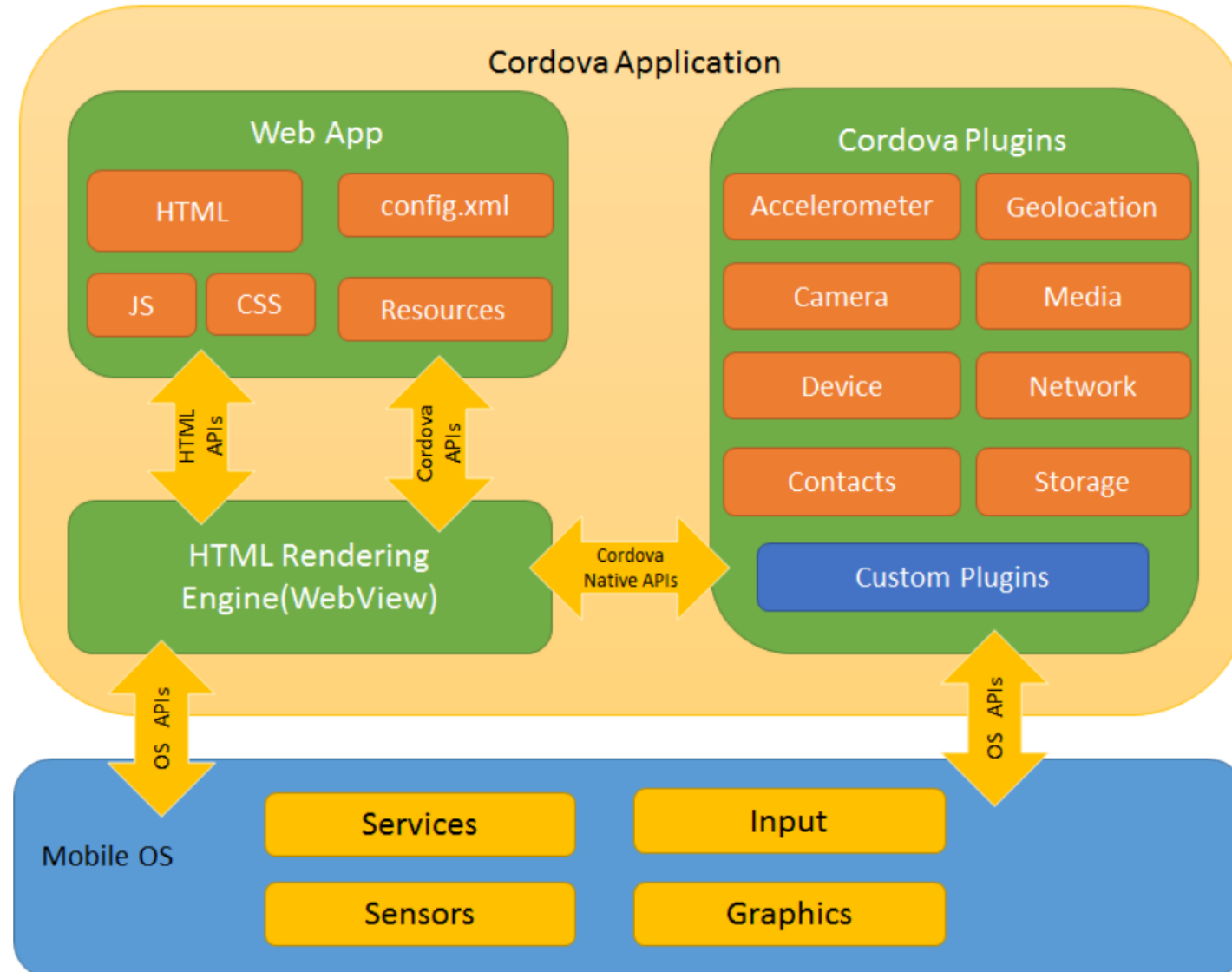
Cordova



Phone**Gap**



# Architektur



Quelle: <https://cordova.apache.org/static/img/guide/cordovaapparchitecture.png>



# Architektur

- WebView: Dient der Darstellung der mobilen Applikation
- WebApp: Hier befindet sich der Code der Applikation. Die Applikation selber wird wie eine *Webseite* implementiert. Genau wie bei einer Webseite wird in einer *index.html* *Css*, *JavaScript*, *Bilder* und andere Ressourcen referenziert.
- Plugins: Stellen ein Interface für Cordova und native Komponenten bereit, um auf die Geräte API zuzugreifen. Dies ermöglicht die Ausführung von *nativen Code* mittels *JavaScript*. Die von Cordova bereitgestellten *Core Plugins* ermöglichen den Zugriff auf allgemeine Gerätefunktionen/-sensoren wie Kamera, Beschleunigungssensor..
- Bei der Erstellung eines Cordova-Projekts sind standardmäßig *keine Plugins* integriert. Weiterhin stellt Cordova *keine UI-Widgets* oder *MV\* Frameworks* bereit sondern nur die *Laufzeitumgebung* in welcher diese ausgeführt werden.

## Nachteile:

- Performance-Probleme besonders bei rechenintensiven Anwendungen z.B. Spiele
- Können keine plattformspezifischen Interaktionen umsetzen
- Haben oft nicht das Look-and-feel der jeweiligen Mobilplattform

## Vorteile:

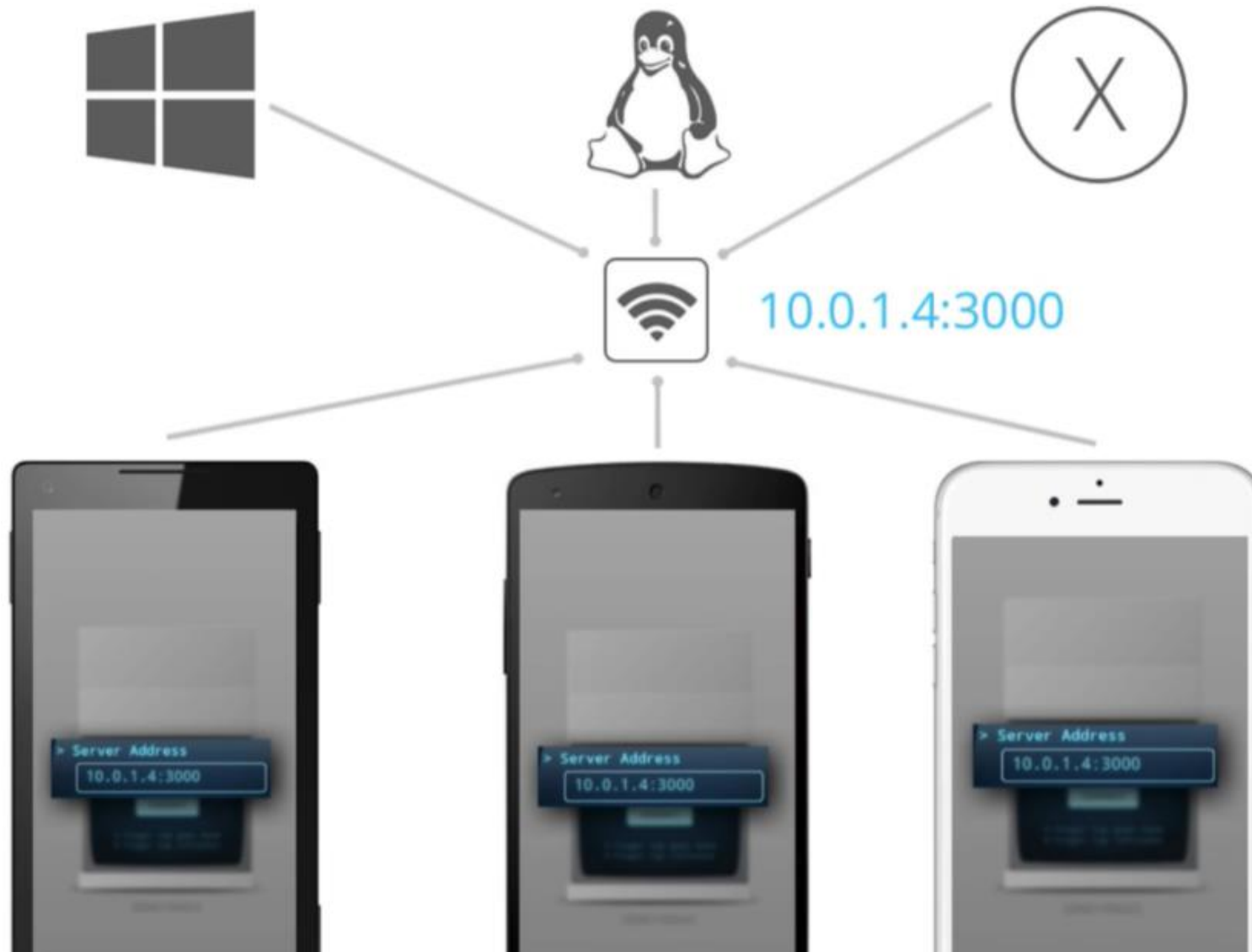
- Mit einer Entwicklung sind alle relevanten mobilen Plattformen abgedeckt
- Entwicklungs- und Testaufwand sinkt erheblich
- Geringere Entwicklungskosten



# Setup Phonegap

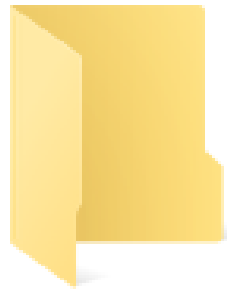
- 1) Installation von NodeJS
- 2) Installation von Git
- 3) Installation von Cordova (`npm install -g cordova`)
- 4) Installation von Phonegap (`npm install -g phonegap@latest`)
- 5) Erstellung eines neuen Projekts (`phonegap create MyProject`)
- 6) In das Verzeichnis „MyProject“ wechseln (`cd MyProject/`)
- 7) Der Befehl „`phonegap serve`“ startet einen kleinen Webserver, um MyProject zu hosten. Über die bereitgestellte IP kann das Projekt entweder mittels der PhoneGap Developer App (erhältlich im App Store) auf einem Smartphone oder im Desktop Browser (z.B. unter `http://localhost:3000/`) getestet werden.

# PhoneGap Developer App



# Phonegap Projekt

Hooks werden zu einem bestimmten Zeitpunkt in den Erstellungsprozess der Applikation eingebunden



hooks

Unterstützte Plattformen



platforms

Support für native Funktionen



plugins

Die Applikation



www

App Parameter



config.xml

Plattform hinzufügen :

In das Verzeichnis der Applikation wechseln (z.B. cd MyProject) und Befehl für die gewünschte Plattform eingeben:

```
$cordova platform add android
```

```
$cordova platform add blackberry10
```

```
$cordova platform add firefoxxos
```

Plattform entfernen:

```
$cordova platform rm android
```

# Phonegap Projekt

## Config.xml

Spezifiziert den formalen Namen. Dieser wird im App-store und auf dem Smartphone angezeigt.

```
<widget id="com.example.hello" version="0.0.1">
  <name>HelloWorld</name>
  <description>
    A sample Apache Cordova application that responds to the deviceready event.
  </description>
  <author email="dev@callback.apache.org" href="http://cordova.io">
    Apache Cordova Team
  </author>
  <content src="index.html" />
  <access origin="*" />
</widget>
```

# Phonegap Projekt

## Config.xml

```
<widget id="com.example.hello" version="0.0.1">
  <name>HelloWorld</name>
  <description>
    A sample Apache Cordova application that responds to the deviceready event.
  </description>
  <author email="dev@callback.apache.org" href="http://cordova.io">
    Apache Cordova Team
  </author>
  <content src="index.html" />
  <access origin="*" />
</widget>
```

Spezifiziert Metadaten und Kontaktinformationen, welche im App-Store angezeigt werden

# Phonegap Projekt

## Config.xml

```
<widget id="com.example.hello" version="0.0.1">
  <name>HelloWorld</name>
  <description>
    A sample Apache Cordova application that responds to the deviceready event.
  </description>
  <author email="dev@callback.apache.org" href="http://cordova.io">
    Apache Cordova Team
  </author>
  <content src="index.html" />
  <access origin="*" />
</widget>
```

Definiert die Startseite der App



# Phonegap Projekt

## Config.xml

```
<widget id="com.example.hello" version="0.0.1">
  <name>HelloWorld</name>
  <description>
    A sample application that responds to the deviceready event.
  </description>
  <author email="me@apache.org" href="http://cordova.io">
    Apache Cordova
  </author>
  <content src="index.html" />
  <access origin="*" />
</widget>
```

Danger!! – cross site script (XSS) exploit risk

Definiert externe Domains, mit welcher die App kommunizieren darf.

„\*“ => Zugriff auf alle Domänen. Bevor die App live geht, sollten hier nur Domänen eingetragen werden mit denen die App kommunizieren darf.

z.B. <access origin=<http://google.com> /> . Für weitere Informationen siehe: Whitelist Guide Cordova

# Phonegap Projekt

Config.xml

Definiert diverse  
plattformspezifische  
Optionen

```
<preference name="DisallowOverscroll" value="true" />
<preference name="android-minSdkVersion" value="14" />
<plugin name="cordova-plugin-battery-status" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-camera" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-media-capture" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-console" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-contacts" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-device" source="npm" spec=">1.0.0" />
```

# Phonegap Projekt

## Config.xml

API Zugriff wie z.B. Kamera.  
Die Standard Phonegap  
APIs sind cross-platform

```
<preference name="DisallowOverscroll" value="true" />
<preference name="android-minSdkVersion" value="14" />
<plugin name="cordova-plugin-battery-status" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-camera" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-media-capture" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-console" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-contacts" source="npm" spec=">1.0.0" />
<plugin name="cordova-plugin-device" source="npm" spec=">1.0.0" />
```

# Phonegap Projekt

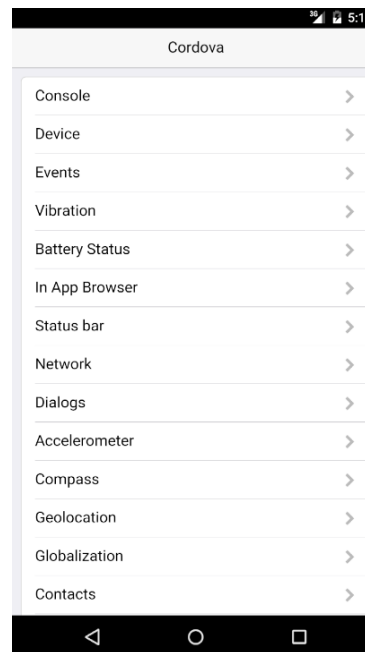
index.html

```
<body>
  <div class="app">
    <h1>PhoneGap</h1>
    <div id="deviceready" class="blink">
      <p class="event listening">Connecting to Device</p>
      <p class="event received">Device is Ready</p>
    </div>
  </div>
  <script type="text/javascript" src="cordova.js"></script>
  <script type="text/javascript" src="js/index.js"></script>
  <script type="text/javascript">
    app.initialize();
  </script>
</body>
```

Ist der „Kleber“ zwischen der Javascript Engine und dem nativen Code.

# Cordova Plugin APIs

- Um ein Plugin zu einer Applikation hinzuzufügen, muss in das Projektverzeichnis gewechselt werden (z.B. `cd MyProject`) und der Befehl „`cordova plugin add {Name des Plugins}`“ ausgeführt werden.
- Nachfolgend werden die wichtigsten Core Plugins aufgeführt. Der Code hierfür befindet sich in `Plugins.zip`.



# Console API

- Hilfreich für Debugging
- Fertige Apps sollten keine console.log Ausgaben enthalten

```
$('#Console3').on('touchend', function(){  
    console.log( {"object":"Yup", "Num": 10} );  
});
```

```
[phonegap] [console.log] { object: 'Yup', Num: 10 }  
[phonegap] [console.log] { object: 'Yup', Num: 10 }  
[phonegap] [console.log] { object: 'Yup', Num: 10 }  
[phonegap] [console.log] [ 1, 'String', { OBJ: 'YUP!' }, [ 1, 2, 3 ] ]
```

# Device API



- Gibt Information über das Gerät zurück

```
cordova plugin add cordova-plugin-device
```

```
$('#Dev1').html(device.cordova);  
$('#Dev2').html(device.model);  
$('#Dev3').html(device.platform);  
$('#Dev4').html(device.uuid);  
$('#Dev5').html(device.version);
```

```
Cordova : 5.1.1  
Model : Android SDK built for x86  
Platform : Android  
Universally Unique Identifier : ca32ff59191f1108  
Version : 5.1.1
```

# Event API

- Erkennung von User Interaktionen oder von Geräteänderungen
- Es ist nicht empfohlen die Funktionalität der Lautstärketasten zu überschreiben -> wird nicht im App-Store akzeptiert
- Die veralteten Android-Events sollten nicht verwendet werden

```
document.addEventListener('pause', pause, false);
document.addEventListener('resume', resume, false);
document.addEventListener("backbutton", backbtn, false);
document.addEventListener('volumeupbutton', volumeup, false);
document.addEventListener('volumedownbutton', volumedown, false);
// Android deprecated
document.addEventListener('menubutton', menubtn, false);
document.addEventListener('searchbutton', searchbtn, false);
```



# Vibration API

- Kann verwendet werden, um Aufmerksamkeit des Users zu erlangen
- Zum Testen braucht man ein Gerät
- Installation:

```
cordova plugin add cordova-plugin-vibration
```

```
navigator.vibrate(1000);
```

# Batterystatus API

- Erlaubt es auf Änderungen des Batteriestandes zu reagieren
  - z.B. Daten im Hintergrund speichern, wenn der Batteriestand niedrig oder kritisch ist

```
cordova plugin add cordova-plugin-battery-status
```

```
window.addEventListener("batterystatus", onBatteryChange, false);  
window.addEventListener("batterylow", onBatteryLow, false);  
window.addEventListener("batterycritical", onBatteryCritical, false);
```

# Batterystatus API

```
window.addEventListener("batterystatus", onBatteryChange, false);  
  
function onBatteryChange(info) {  
    $('#BatteryEvent').html('Battery Status');  
    $('#BatteryLevel').html( info.level );  
    $('#BatteryPlugged').html( info.isPlugged );  
}
```



```
{  
  level: 80,  
  isPlugged: false  
}
```

# In-App-Browser

- Ermöglicht das Öffnen einer Webseite innerhalb der App
- Dadurch wird die App nicht geschlossen, wenn ein Link innerhalb der App gedrückt wird

```
cordova plugin add cordova-plugin-inappbrowser
```

- Damit der In-App-Browser funktioniert muss die config.xml um

```
<allow-navigation href="http://*/*" />
```

ergänzt werden.

# Status Bar

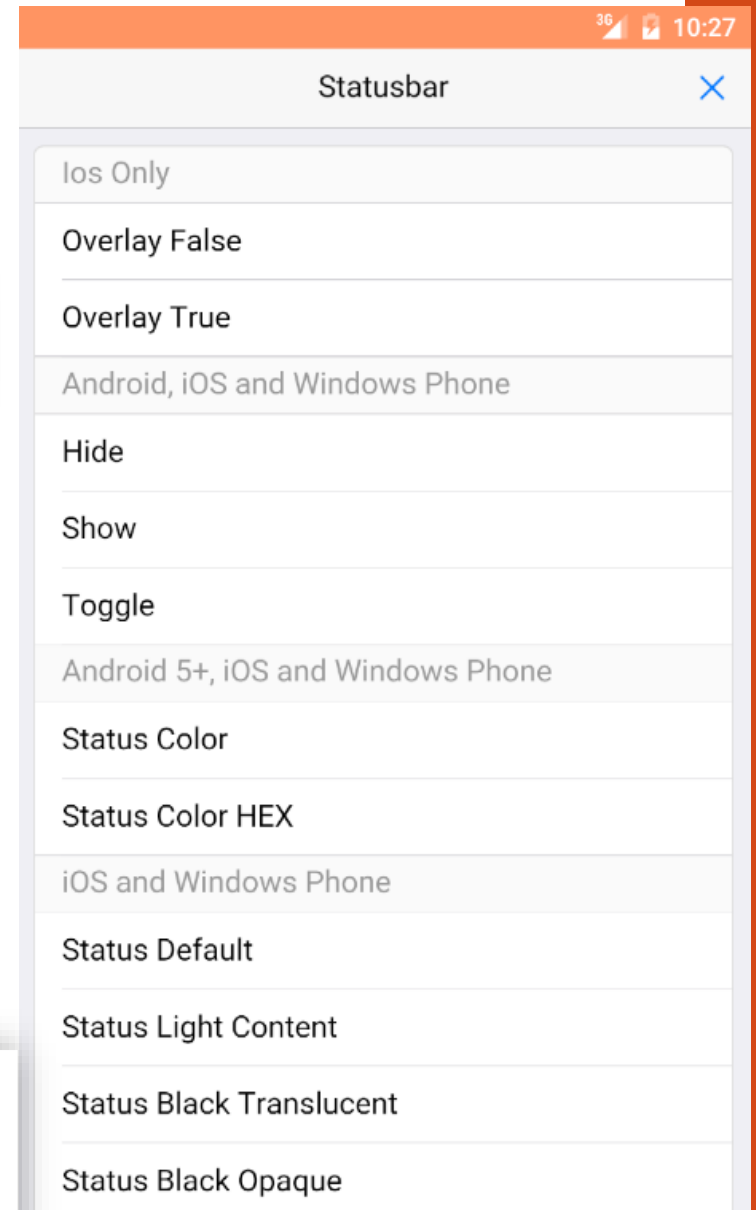
- Konfiguration des Styles und der Sichtbarkeit der Statusbar

```
cordova plugin add cordova-plugin-statusbar
```

```
StatusBar.overlaysWebView  
StatusBar.styleDefault  
StatusBar.styleLightContent  
StatusBar.styleBlackTranslucent  
StatusBar.styleBlackOpaque  
StatusBar.backgroundColorByName  
StatusBar.backgroundColorByHexString  
StatusBar.hide  
StatusBar.show
```

config.xml:

```
<preference name="StatusBarOverlaysWebView" value="true" />  
<preference name="StatusBarBackgroundColor" value="#aaaaaa" />  
<preference name="StatusBarStyle" value="default" />
```



# Dialog API

- Eingabe von Daten, Auswahl von Optionen

```
cordova plugin add cordova-plugin-dialogs
```

Methoden:

```
navigator.notification.alert  
navigator.notification.confirm  
navigator.notification.prompt  
navigator.notification.beep
```

```
navigator.notification.alert(message, alertCallback, [title], [buttonName])
```

# Networking API

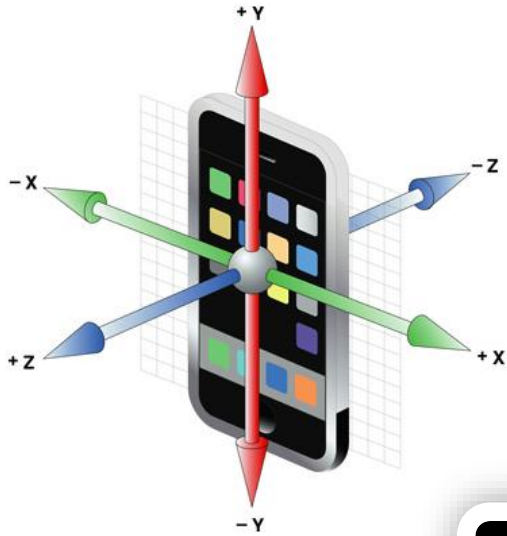
- Ermöglicht die Netzwerkanbindung zu prüfen
- z.B. Meldung an den User, wenn keine Netzwerkverbindung vorhanden
- Voraussetzung um im App-Store akzeptiert zu werden

```
cordova add cordova-plugin-network-information
```

```
var networkState = navigator.connection.type;  
  
document.addEventListener("online", onOnline, false);  
document.addEventListener("offline", onOffline, false);
```

```
Connection.UNKNOWN  
Connection.ETHERNET  
Connection.WIFI  
Connection.CELL_2G  
Connection.CELL_3G  
Connection.CELL_4G  
Connection.CELL  
Connection.NONE
```

# Accelerometer API



- Ermöglicht den Zugriff auf die Daten des Beschleunigungssensors
- Erkennt Bewegungsänderungen relativ zu der aktuellen Geräteorientierung
- Anwendung z.B. für Spielsteuerung

```
cordova plugin add cordova-plugin-device-motion
```

```
navigator.accelerometer.getCurrentAcceleration(onSuccess, onError);  
navigator.accelerometer.watchAcceleration(onSuccess, onError, { frequency: 100 } );  
navigator.accelerometer.clearWatch(watchID);
```

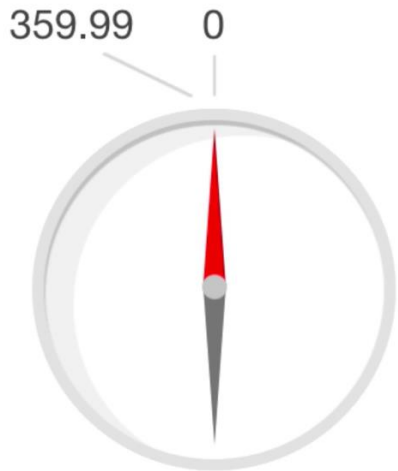
```
acceleration.x  
acceleration.y  
acceleration.z
```



# Compass API

- Ermöglicht das Auslesen des Magnetometers
- Ausgabe: 0 - 359.99 -> Gibt an wohin die Kompassnadel zeigt

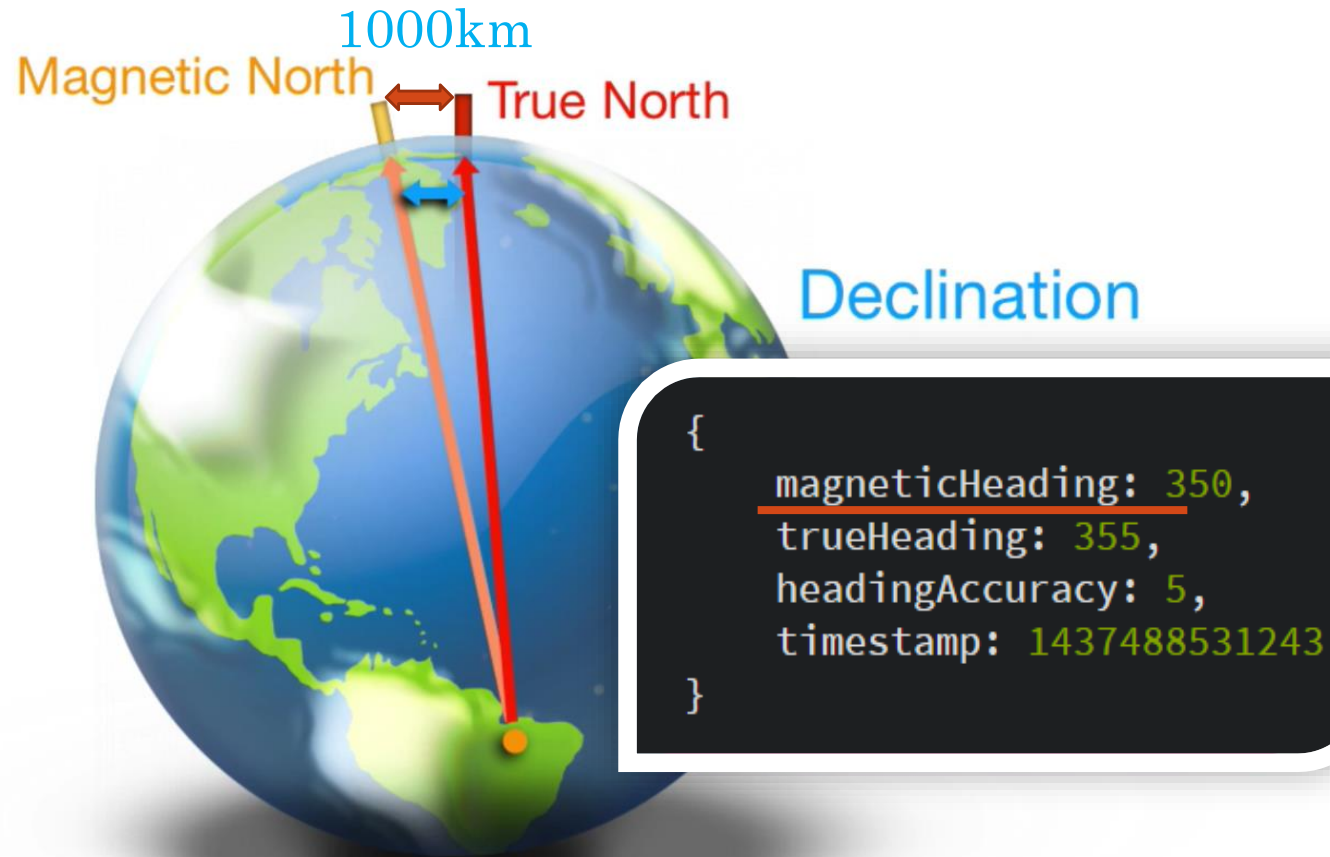
```
cordova plugin add cordova-plugin-device-orientation
```



```
{  
  magneticHeading: 350,  
  trueHeading: 355,  
  headingAccuracy: 5,  
  timestamp: 1437488531243  
}
```

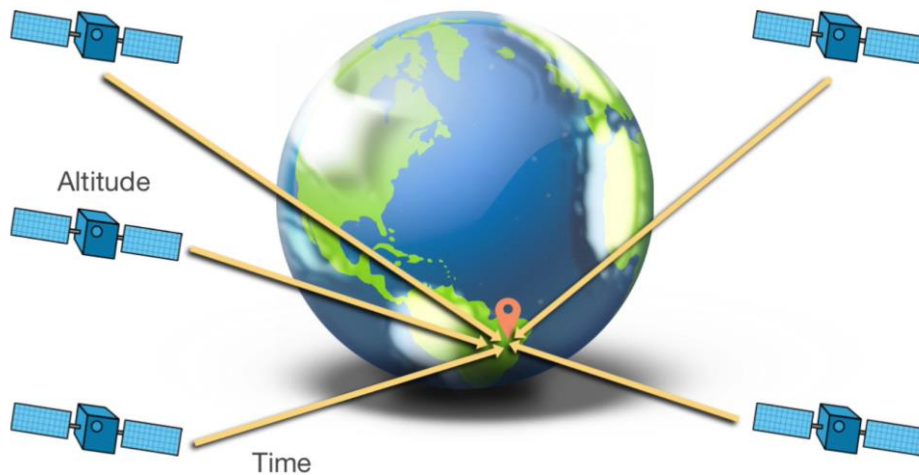
```
navigator.compass.getCurrentHeading  
navigator.compass.watchHeading  
navigator.compass.clearWatch
```

# Magnetic North vs True North

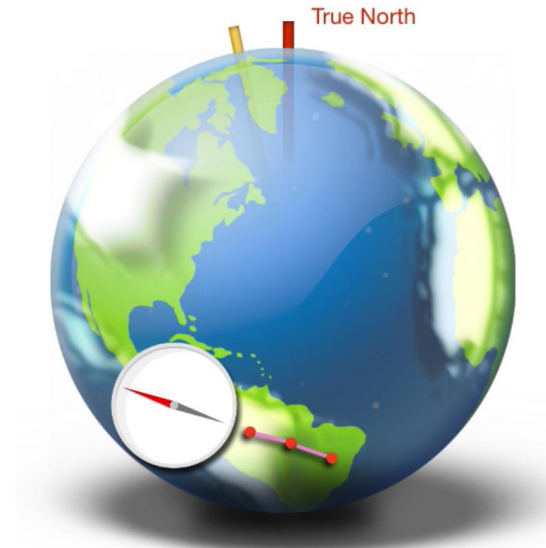


# Geolocation API

- Ermöglicht die Bestimmung der derzeitige Position des Gerätes auf dem Globus
- Der GPS- Sensor benötigt die meiste Energie und ist der langsamste von allen Sensoren



Position



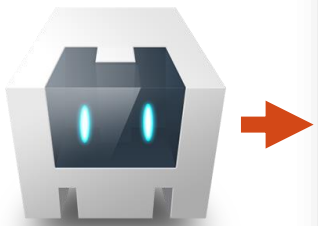
Richtung in die das Handy  
sich bewegt



Geschwindigkeit

# Geolocation API

- Funktioniert nur für Cordova-Versionen > 5.0
- Positionsdaten sind sensible Daten. In der Datenschutzerklärung sollte dargelegt werden, inwiefern die App die Positionsdaten verwendet. Weiterhin sollte der Nutzer darüber informiert werden, wenn die App anfängt Positionsdaten abzufragen.



```
{
  coords: {
    latitude: 53.0,
    longitude: -1.2,
    accuracy: 21,
    altitude: 254,
    altitudeAccuracy: 10,
    heading: 152,
    speed: 0,
  },
  timestamp: 1438116637404
}
```

```
cordova plugin add cordova-plugin-geolocation
```

Methoden:

```
navigator.geolocation.watchPosition( geoSuccess,
navigator.geolocation.getCurrentPosition;
navigator.geolocation.clearWatch( geoWatchID );
```

# Globalization API

- Ermöglicht die Formatierung von Daten abhängig von der Region des Benutzers

```
cordova plugin add cordova-plugin-globalization
```

```
navigator.globalization.getPreferredLanguage  
navigator.globalization.getLocaleName  
navigator.globalization.dateToString  
navigator.globalization.stringToDate  
navigator.globalization.getDatePattern  
navigator.globalization.getDateNames  
navigator.globalization.isDayLightSavingsTime  
navigator.globalization.getFirstDayOfWeek  
navigator.globalization.numberToString  
navigator.globalization.stringToNumber  
navigator.globalization.getNumberPattern  
navigator.globalization.getCurrencyPattern
```

# Contacts API

- Vor dem Zugriff auf die Kontaktinformationen des Benutzers, muss dieser benachrichtigt werden -> App wird sonst nicht im App-Store akzeptiert
- Aus der Datenschutzerklärung der App muss klar hervorgehen, dass Zugriff auf die Kontaktdaten besteht und inwieweit diese Daten verwendet werden

```
cordova plugin add cordova-plugin-contacts
```

```
navigator.contacts.create  
navigator.contacts.find  
navigator.contacts.pickContact
```

# Camera API

- Ermöglicht das Aufnehmen von Bildern und das Auswählen von Bildern aus der System Bilderbibliothek
- Aus der Datenschutzerklärung muss klar hervorgehen, inwiefern die Bildern verwendet werden z.B. Speicherung auf einen Server



<https://>

Wichtig: Sicherer  
Upload von User  
Daten



Benutzer mitteilen,  
dass die Bilder auf  
einen Server  
hochgeladen  
werden



Bei keiner  
Benachrichtigung  
wird die App vom  
App-Store  
abgewiesen

# Camera API

```
cordova plugin add cordova-plugin-camera
```

- Es können nur Bilder aufgenommen werden – keine Videos
- Methode:

```
navigator.camera.getPicture(camSuccess, camError, options);
```

- Wichtig: In den übergebenen Optionen den „destinationType“ auf FILE\_URI oder NATIVE\_URI setzen – bei DATA\_URL kann die App abstürzen

```
'destinationType' : navigator.camera.DestinationType.FILE_URI
```



# Media API

- Ermöglicht das Aufnehmen und Abspielen von Audio

```
cordova plugin add cordova-plugin-media
```

- Zugriffspfad auf eine Datei innerhalb des www-Verzeichnisses abhängig vom OS

```
if( device.platform === 'Android' ) {  
    sourceToPlay = '/android_asset/www/Warriors.mp3';  
} else {  
    sourceToPlay = 'Warriors.mp3';  
}
```

# Media API

- Methoden:

```
var media = new Media( sourceToPlay, null, mediaError, mediaStatus );  
media.getCurrentPosition  
media.getDuration  
media.play  
media.pause  
media.release  
media.seekTo  
media.setVolume  
media.startRecord  
media.stopRecord  
media.stop
```

# Media Capture API

- Ermöglicht den Zugriff auf die Audio, Video und Bilder Aufnahmefähigkeiten

```
cordova plugin add cordova-plugin-media-capture
```

```
var capture = navigator.device.capture;  
capture.captureAudio  
capture.captureImage  
capture.captureVideo
```

```
mediaFile.getFormatData
```

Nicht von Android  
unterstützt

```
{  
  "height": 0,  
  "width": 0,  
  "bitrate": 0,  
  "duration": 0,  
  "codecs": ""  
}
```

Nicht von IOS  
unterstützt

# File API

- Ermöglicht Schreib-und Lesezugriffe auf Dateien im Gerät

```
cordova plugin add cordova-plugin-file
```

Siehe Beispielprojekt: file-system.js

# Whitelist

Mittels dieses Plugins können Ausnahmelisten (Whitelists) für die Navigation erstellt werden.

## Navigation Whitelist

Kontrolliert zu welchen URLs innerhalb des WebViews navigiert werden darf. Standardmäßig werden nur file:/// URLs erlaubt. Um andere URLs zu erlauben, muss der Tag <allow-navigation> in die config.xml eingefügt werden.

## Intent Whitelist

Kontrolliert welche URLs innerhalb der Applikation geöffnet werden dürfen. Diese Ausnahmeliste wirkt sich nicht auf Plugins sondern nur auf Hyperlinks und window.open() Aufrufe aus. In die config.xml müssen <allow-intent> tags eingefügt werden.

```
cordova plugin add cordova-plugin-whitelist
```

# Whitelist

## Navigation Whitelist

```
<!-- Erlaube Links zu example.com -->
<allow-navigation href="http://example.com/*" />

<!-- Wildcards können verwendet werden, um das gesamte Netzwerk zu whitelisten.
     *Nicht empfohlen* -->
<allow-navigation href="*" />
```

## Intent Whitelist

```
<!-- Erlaubt Links, die eine Messenger App öffnen -->
<allow-intent href="sms:*" />

<!-- Erlaubt Links, die die Nummereingabe öffnen -->
<allow-intent href="tel:*" />

<!-- Erlaubt Links, die eine Karte öffnen -->
<allow-intent href="geo:*" />

<!-- Erlaubt alle URLs von installierten Apps
     *Nicht empfohlen* -->
<allow-intent href="*" />
```

# Third-Party-Plugins

- Installation von github oder von npm z.B. PhoneGap BarcodeScanner

## PhoneGap Plugin BarcodeScanner

=====

build error

Cross-platform BarcodeScanner for Cordova / PhoneGap.

Follows the [Cordova Plugin spec](#), so that it works with [Plugman](#).

### Installation

This requires phonegap 5.0+ ( current stable v3.0.0 )

```
phonegap plugin add phonegap-plugin-barcodescanner
```

Older versions of phonegap can still install via the **deprecated** id ( stale v2.0.1 )

```
phonegap plugin add com.phonegap.plugins.barcodescanner
```

It is also possible to install via repo url directly ( unstable )

```
phonegap plugin add https://github.com/phonegap/phonegap-plugin-barcodescanner.git
```

Intel® XDK

 React Native

 ionic

  
NativeScript

# UI Frameworks

  
Onsen UI

 **jQuery**  
mobile

 **Sencha**



**Kendo UI**  
THE ART OF WEB DEVELOPMENT





- Frontend-Framework
- Business-Logik-Schicht mit Angular
- MVVM/MVC-Pattern in einer hybriden App (Ionic 3 verwendet das auf TypeScript basierende Angular 4)
- Sämtliche Komponenten liegen für unterschiedliche Plattformen (iOS, Android und Windows) angepasst vor -> Design-Guidelines sind so umgesetzt, dass ein Unterschied zu den nativen Kontrollelementen fast nicht erkennbar ist.



- Ionic installieren

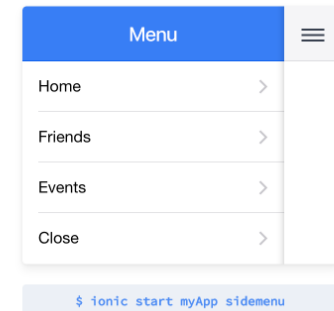
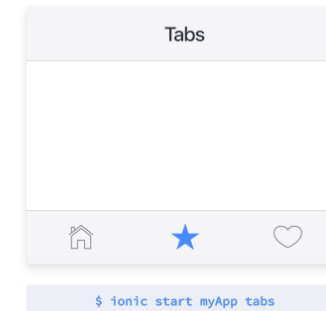
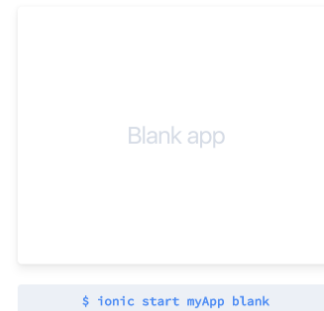
```
npm install -g cordova ionic
```

- Erstellen einer neuen App

```
ionic start myApp tabs
```

- Anzeigen der App im Browser

```
cd myApp  
ionic serve
```



# Performance



# Performance

- Browser bzw. Betriebssysteme spielen eine entscheidende Rolle
- Je schneller der Browser, desto schneller die hybride App
- Ab iOS 8 und Android KitKat (4.4) : leistungsstarke Render-Engine WebKit + WebGL
  - Großer Performancezuwachs für Hybride-Apps
- Ab Android 5.0 kann der WebView über den Google Play Store aktualisiert werden
- Bei Apple bringt jedes Betriebssystem Update neue JavaScript und CSS Engine Updates



# Performance

## Probleme:

- Große Plattformvielfalt auf dem Markt
- Android Versionen unter 4.4 profitieren nicht von WebKit
  - Hybride Apps weisen eine schlechte Performance auf

## Lösung:





- Erlaubt es Web-Applikationen in einer eigenen dedizierten Laufzeitumgebung auszuliefern

Dies bedeutet:

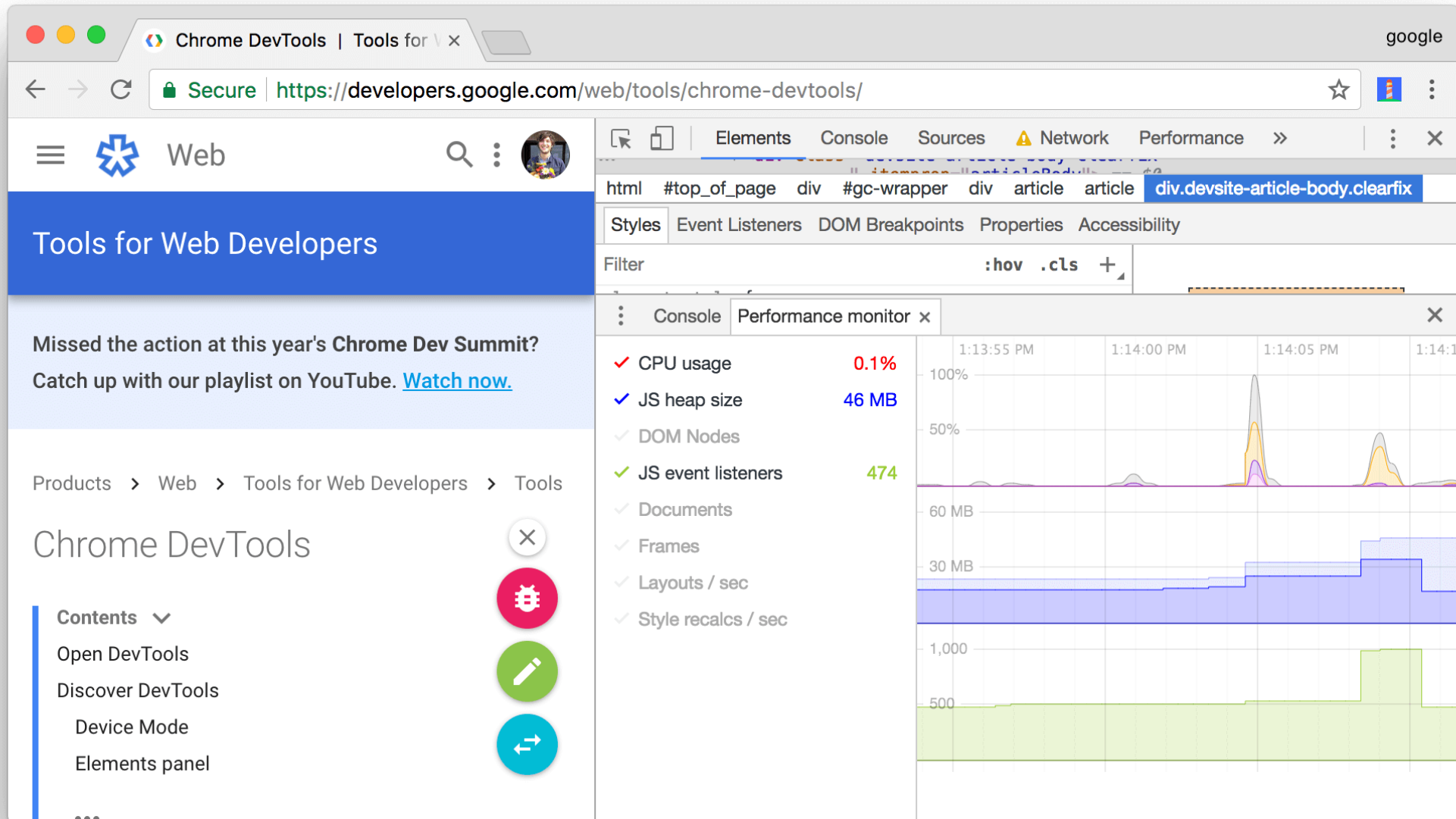
- Die Applikation wird performant laufen, egal wie alt der WebView oder Browser des Geräts ist
- Keine Laufzeitunterschiede zwischen einzelnen Geräten

```
cordova plugin add cordova-plugin-crosswalk-webview
```

# Performance

- Kompakter Code => Minify CSS und JS ( [jscompress.com](http://jscompress.com) , [cssminifier.com](http://cssminifier.com) )
- Verwende, wenn nicht nötig, keine großen Libraries, Frameworks oder Plugins. Statt jQuery kann auch zepto.js verwendet werden
- Ein minimalistisches UI Design spart Ressourcen
  - Optimierung von Bildern mittels CSS Sprite Sheets
  - Skalierung von Bildern in HTML vermeiden
  - Verwende Bilder die von der Größe optimal passen
  - Begrenze Schatten und Gradienten auf ein Minimum
  - Generiere die UI nicht auf dem Server, sondern in JS auf dem Klienten
- Verwende lokales Caching, um Netzwerkzugriffe zu minimieren
- Bevor die App auf den Markt kommt, sollte jeder Aspekt des Designs, Ladegeschwindigkeit und Performance getestet werden (end-to-end Automation Testing )
  - Performance Monitor von Chrome (ab Version 64)
  - Unit Testing Frameworks: Jasmine, Mocha und Karma
  - Tools um Unit Tests auszuführen: Istanbul
  - Mit Browser-Perf/ New Relic kann die Performance direkt im Browser getestet werden

# Chromes Performance Monitor





# Bessere Performance durch Node.js und Ionic

- Ionic
  - Cleveres Speichermanagement
  - Komplex Listen werden nur in Teilen im HTML5-Code geladen
    - Performantes Scrolling
- Node.js
  - Geschicktes Anfrage-Handling ans Backend
  - Asynchrone Anfragen
  - Aktive Kommunikation zwischen Server und App



# Sicherheit



# Sicherheit

- Keine alte Version von cordova verwenden



```
sudo npm update -g cordova
```

# Sicherheit

- Regelmäßig überprüfen, ob Plugins und Plattformen aktuell sind



cd to/your/app  
cordova plugin update



cd to/your/app  
cordova platform update ios android

# Sicherheit

- Keine Plugins verwenden, die man nicht braucht.
- Ausnahmelisten (Whitelists) sind wichtig => hindert die App daran mit nicht authentisierten Webseiten zu kommunizieren



`<access origin="*" />`



`<access origin="http://example.de" subdomain="true"/>`

# Sicherheit



# Sicherheit

- Bei Banking-Apps oder anderen Apps, die mit sensiblen Daten arbeiten, sollte das Cordova-HTTP-Plugin verwendet werden
  - Dadurch werden Man-in-the-Middle Angriffe verhindert

The screenshot shows the GitHub repository page for 'wymsee / cordova-HTTP'. The repository description is 'Cordova / Phonegap plugin for communicating with HTTP servers. Allows for SSL pinning!'. It has 83 commits, 2 branches, 8 releases, and 11 contributors. The 'master' branch is selected. Below the repository information, there is a list of files and their commit history:

File	Commit Message	Time
src	updated http request and removed some android code that is now in js....	2 months ago
www	fixes for useBasicAuth and setHeader for angular use	2 months ago
.editorconfig	Fixed ios code to work with AFNetwork 3.0.x	2 months ago
.gitignore	updated readme and added gitignore	2 years ago
CHANGELOG.md	updated the changelog	2 months ago
LICENSE	updated the readme	2 years ago
README.md	updated install instructions in readme	2 months ago
package.json	1.0.2	2 months ago
plugin.xml	fixed some issues with responseSerializers and moved some functions o...	2 months ago
zedconfig.json	Fixed ios code to work with AFNetwork 3.0.x	2 months ago

# Sicherheit

- Daten auf einem gesicherten Server speichern
- Lokaler Speicher wird nicht verschlüsselt => Web Cryptography API
- Niemals sensible Daten cachen



Web Cryptography API



# Content Security Policy

- Ist ein Sicherheitskonzept, um Cross-Site-Scripting und andere Angriffe durch Einschleusen von Daten in Webseiten zu verhindern
- Die index.html muss um den gewünschten meta-Tag erweitert werden

```
<!-- Gute standard Deklaration :  
* gap: wird nur für iOS benötigt (wenn der UIWebView verwendet wird) ->native Kommunikation  
* https://ssl.gstatic.com wird nur für Android benötigt, damit TalkBack korrekt funktioniert  
* Unterbindet die Verwendung von eval() and inline Skripten, um Cross-Site-Scripting zu erschweren. Um dies zu ändern:  
  * Aktiviere inline JS: füge 'unsafe-inline' zu default-src hinzu  
  * Aktiviere eval(): füge 'unsafe-eval' zu default-src hinzu  
-->
```

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self' data:  
  gap: https://ssl.gstatic.com; style-src 'self' 'unsafe-inline'; media-src *">
```

```
<!-- Erlaube alles vom selben Ursprung und foo.com -->
```

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self' foo.com">
```

```
<!-- Erlaubt alles (z.B. CSS, AJAX, object, frame, media, etc) außer das:
```

- \* CSS nur vom selben Ursprung und inline Style,
- \* Skripte nur vom selben Ursprung und inline Skripte , und eval()

```
-->
```

```
<meta http-equiv="Content-Security-Policy" content="default-src *; style-src 'self' 'unsafe-inline';  
  script-src 'self' 'unsafe-inline' 'unsafe-eval'">
```

# Sicherheit

- Keine iFrames verwenden



`<iframe src="*"></iframe>`

- Verwendung des In-App-Browser-Plugins
- Eigene App vor Veröffentlichung versuchen zu hacken

# Sicherheit

- Alle Benutzereingaben validieren - vor allem auch serverseitig
- Niemals die App für Android 2.3 Gingerbread bauen => viele Sicherheitslücken
- Niemals Authentifizierungsschlüssel in JS-Dateien ablegen



validate

# Debugging & Testing



Quelle: [https://www.iconexperience.com/g\\_collection/icons/?icon=debug](https://www.iconexperience.com/g_collection/icons/?icon=debug)

# Setup Android Emulator

- Installation von JDK 7 oder höher
- Installation von Android Studio ([developer.android.com/sdk](http://developer.android.com/sdk))
- Systempfad ergänzen:

`C:\Program Files\Java\jdk1.7.0_75;`

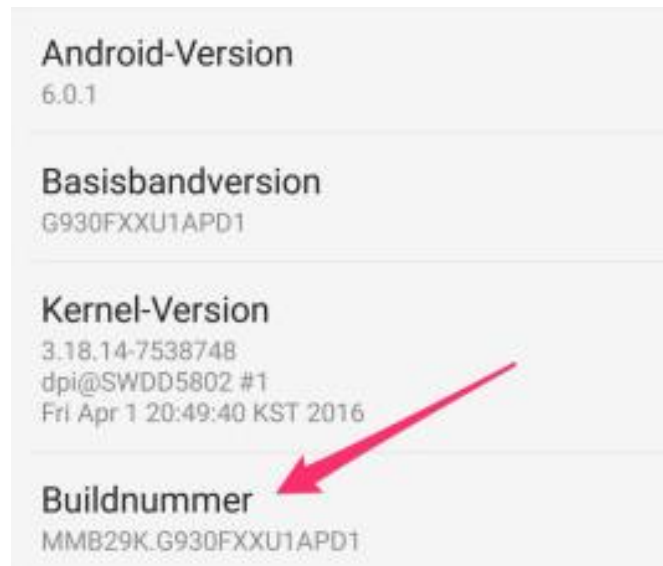
`C:\Users\<USER>\AppData\Local\Android\sdk;`

`C:\Users\<USER>\AppData\Local\Android\sdk\platform-tools`

- Download Android(API-Level 16-25 bzw. Version 4.1 - 7.1.1)
- Setup AVD
- Emulate Projekt (`cordova run --emulator`, `cordova run android --target="Nexus_6"`)

# USB-Debugging

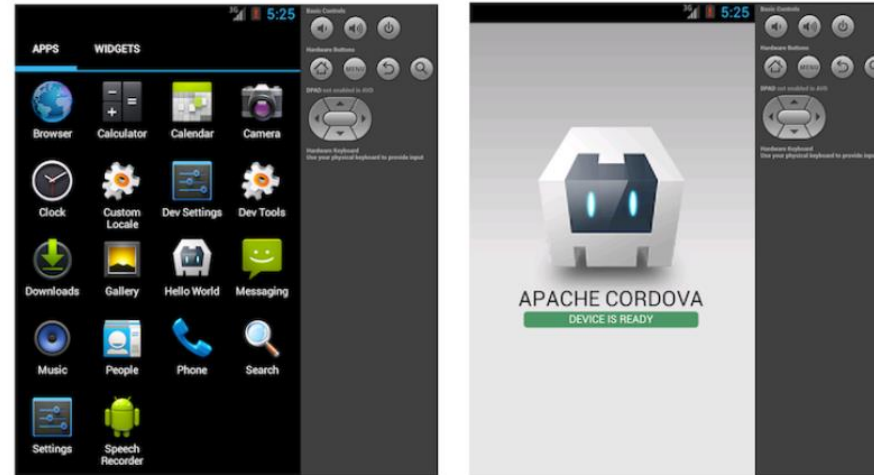
- Aktivierung des USB-Debuggings für Android:
  - Einstellungen des Geräts öffnen
  - Menüpunkt „Geräteinformationen“ bzw. “Softwareinfo“ auswählen
  - Solange auf die „Buildnummer“ tippen bis eine Meldung erscheint



# Testen/Debuggen der App

- Testen der App im Emulator

```
$ cordova emulate android
```



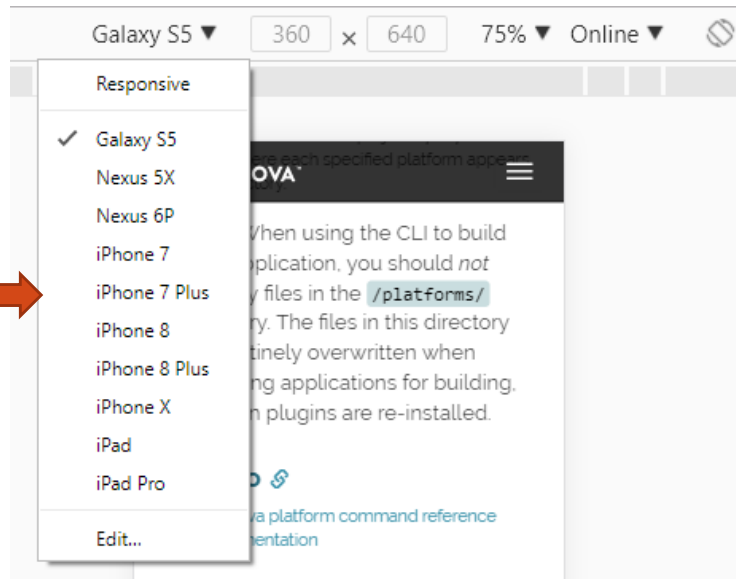
- Testen der App auf dem Smartphone (Smartphone mit dem Rechner verbinden und USB-Debugging auf dem Smartphone aktivieren)

```
$ cordova run android
```

# Testen/Debuggen der App

- Testen/Debuggen der App in Chrome
  - Chrome bietet JavaScript Debugging, Netzwerk-Performance-Profiling, Emulation von Sensoren ...
  - Befehl „**phonegap serve**“ ausführen und angezeigte Adresse in Chrome eingeben (z.B. `http://localhost:3000/`). Wenn statt localhost 127.0.0.1 eingegeben wird, sperrt Chrome aus Sicherheitsgründen mehrere Funktionen wie z.B. die Bestimmung der aktuellen Position
  - F12 öffnet die Entwicklerkonsole

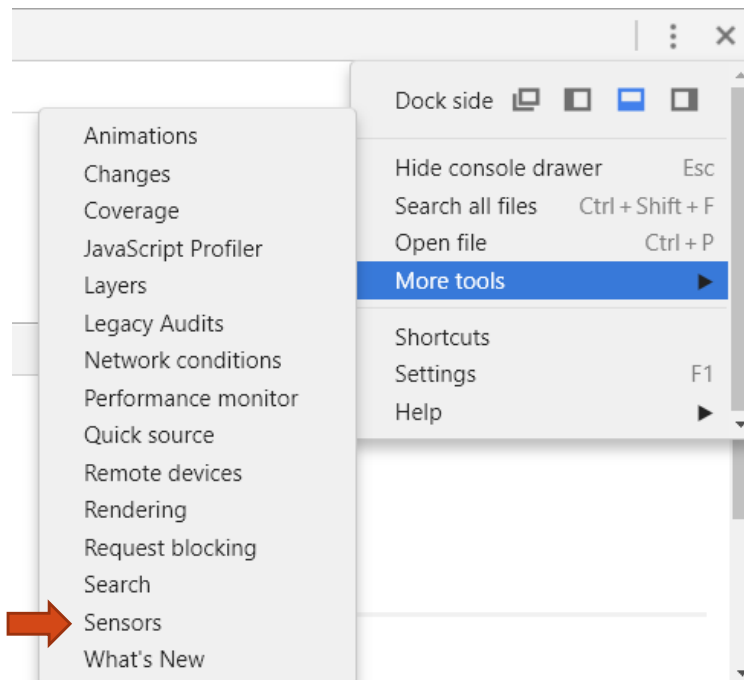
Gerätespezifische Viewports



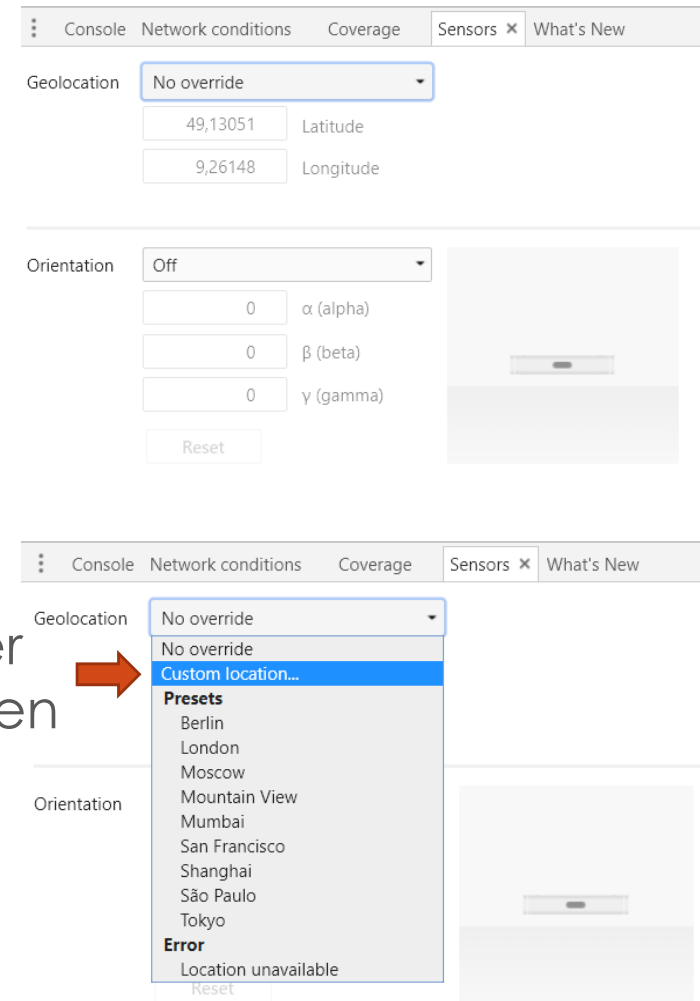


# Testen/Debuggen der App

- Mittels Chrome können Sensoren emuliert werden (Geräteposition/ -orientierung)



Eingabe der  
gewünschten  
Position



# PhoneGap Developer App

```
$ phonegap serve
[phonegap] starting app server...
[phonegap] listening on 192.168.1.11:3000
[phonegap]
[phonegap] ctrl-c to stop the server
[phonegap]
```



Angezeigte Adresse in die  
PhoneGap Developer App  
eingeben

# Bauen der hybriden-App



# Bauen der hybriden App

- iOS Apps können nur auf einem Mac oder in einem Cloud Service gebaut werden

App bauen :

```
$cordova build ios  
$cordova build android
```

Nach einem erfolgreichen Build für Android wird die apk-Datei standardmäßig hier abgelegt:  
{MyProject}/platforms/android/build/outputs/apk/android-debug.apk

# Phonegap Build

- Für die Veröffentlichung muss die App kompiliert werden
  - lokal: cordova build android ios
  - cloud: Auf einem Server kompilieren - > Phongap Build Service
- Phongap Build Service: Github oder zip-File
  - www- Verzeichnis + config.xml



css



fonts



images



img



js



spec



config.xml



index.html



# Hello World

Hello World sample application that responds to the deviceready event.




 Install



 Update code

 Rebuild all

 Builds

 Plugins

 Collaborators

 Settings

App ID	Version	Owned by	PhoneGap (iOS / Android / Windows)	Source	Tag / Branch	Commit	Last built (2)
2038636	1.0.0	sarahgrimm1@gmx.de	cli-5.2.0 (3.9.1 / 4.1.1 / 3.8.1)	<a href="https://github.com/phonegap/phonegap-template-hello-world.git">https://github.com/phonegap/phonegap-template-hello-world.git</a>	master	57e6a47	1 minute

iOS

No key selected ▾

 Rebuild

 Log

 Error



No key selected ▾

 Rebuild

 Log

 apk



No publisher ID s... ▾

 Rebuild

 Log

 xap

# Aufgabe



- Erstellt ein neues PhoneGap Projekt
- Tauscht das www-Verzeichnis mit dem in Moodle abgelegten www-Verzeichnis des Schnitzeljagd-Projekts (task.zip) aus.
- Erweitert das Grundgerüst der App um die Funktionalität, dass bei Erreichen einer Station die Meldung „Du hast die Station ? erreicht“ dem Benutzer angezeigt wird.
- Weiterhin soll die Möglichkeit bestehen einen QR-Code einzuscannen.

# Quellen

- <https://cordova.apache.org/>
- <http://docs.phonegap.com/>
- <https://ionicframework.com/>



**Fragen könnt Ihr an [Sarah.Richter1@oulook.de](mailto:Sarah.Richter1@oulook.de) schicken ☺**