

# Workshop 6 (WS6C) – Mobile App Development

## Lektion 1: Einführung & Übersicht



# Was werdet ihr lernen?

- Generellen Überblick zur Entwicklung von Mobile Apps
- Client Entwicklung für iOS und Android
- Entwicklung mit:
  - Java → Android
  - Swift → iOS
  - HTML, CSS & JavaScript → Ionic

# Über mich

Christian Lüthold

2005 – 2008 Kantonsschule Frauenfeld

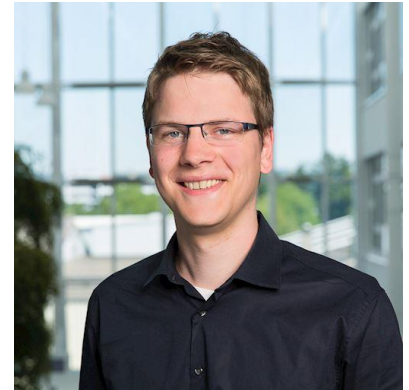
- Schwerpunkt Biologie & Chemie

2009 – 2015 Universität Zürich

- BSc Bioinformatics
- MSc Software Systems

Seit 2015 Advanced Software Engineer @ Zühlke Engineering AG

- Schwerpunkt: .NET, C#, AngularJS, JavaScript/TypeScript
- Projekte:
  - Web Platforms, Hybrid Apps
  - Rega, Zurich, Fifa



# Zühlke Engineering AG

## Facts & Figures

- Dienstleistungsunternehmen
  - Softwareentwicklung
  - Produktentwicklung
  - Management Consulting
  - Ventures
- Gegründet 1968
- Standorte in CH, DE, GB, AT, RS und bald CN
- 125 Mio. CHF Umsatz (2015)
- 730 Mitarbeiter(innen) (2015)



# Zühlke Team




Christian Lüthold  
[christian.luethold@zuehlke.com](mailto:christian.luethold@zuehlke.com)

Organisation




Christian Gauch  
[christian.gauch@zuehlke.com](mailto:christian.gauch@zuehlke.com)

Thema: 



Oliver Gepp  
[oliver.gepp@zuehlke.com](mailto:oliver.gepp@zuehlke.com)

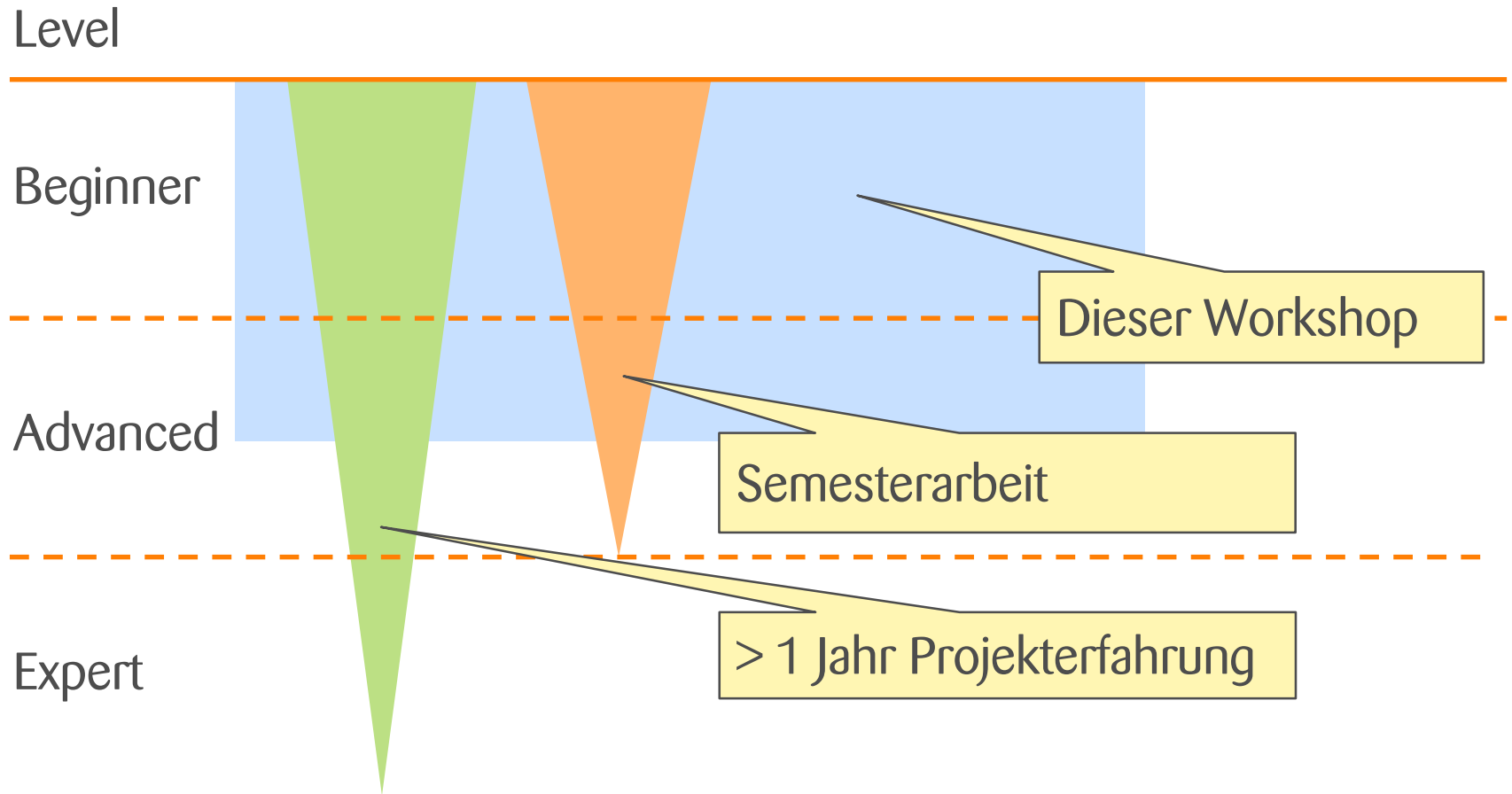
Thema: 



Roman Rast  
[roman.rast@zuehlke.com](mailto:roman.rast@zuehlke.com)

Thema: 

# “Flughöhe” dieses Workshops



# Selbsteinschätzung

Was ist dein Erfahrungslevel in den folgenden Bereichen?

- Java
  - Swift
  - AngularJS
  - HTML, CSS & JavaScript
- 
- Android Entwicklung
  - iOS Entwicklung
  - Ionic Entwicklung

# Erwartungen

Was erwartest du von diesem Workshop?

- DO's
- DON'Ts

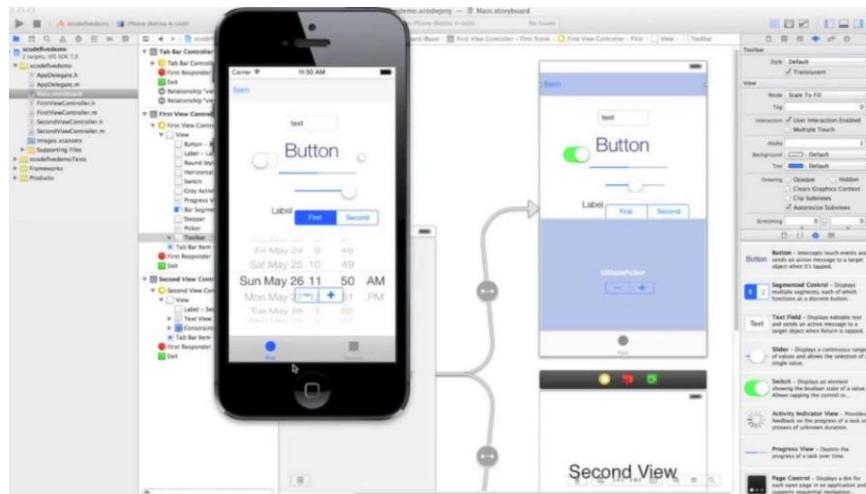


# Native iOS Development

- Objective-C (1980)
- Swift (2014)
- IDE: XCode

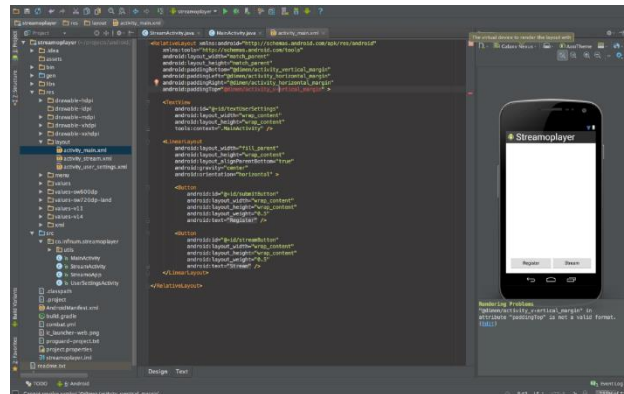
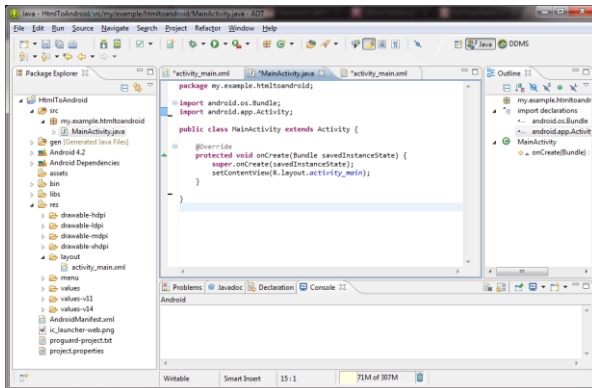


**Objective-C**



# Native Android Development

- Java
- XML für GUI Deklaration
- Nicht im Kurs: Kotlin, Go
- IDE:
  - Eclipse, IntelliJ IDEA
  - Android Studio



# Es gibt auch noch mehr Plattformen!

Ubuntu, Firefox, Windows Mobile...



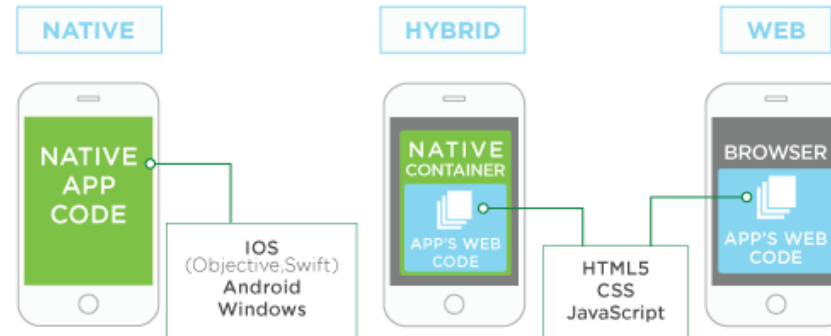
# App Typen

## MOBILE APPS AT A GLANCE NATIVE VS. HYBRID VS. WEB APPLICATIONS



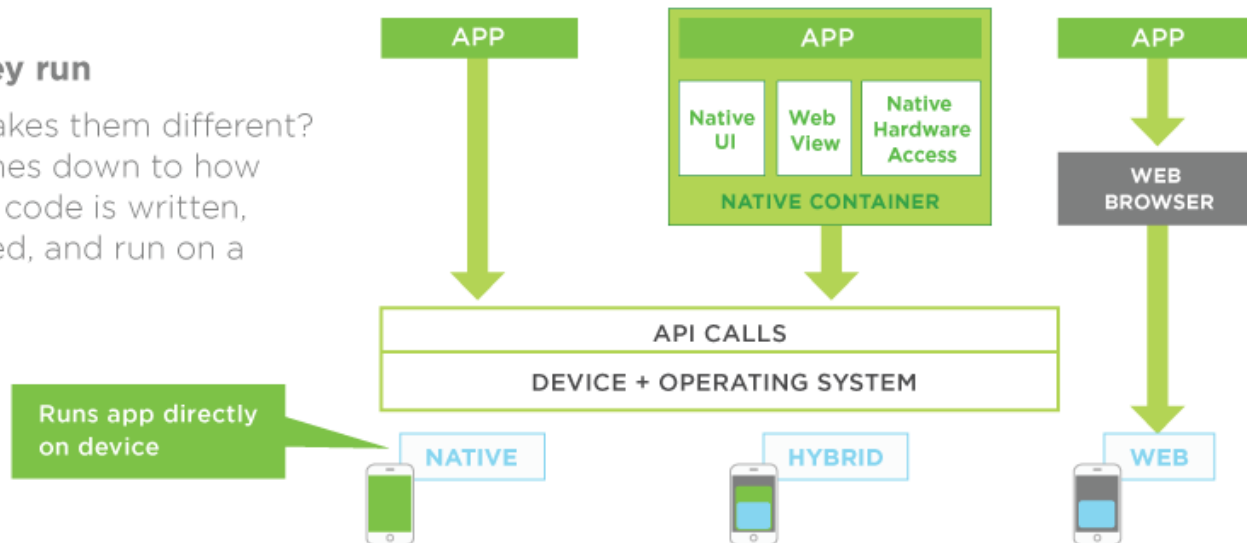
### How they're structured

When it comes to building your mobile app, you have three options: native, web, or hybrid.



### How they run

What makes them different? It all comes down to how an app's code is written, structured, and run on a device.



# Web Apps

## Mobile Version einer Website

- keine Installation, kein Speicherplatzverbrauch
- Entwicklung ist einfach und schnell
  - HTML5, CSS, JavaScript
  - Weitverbreitetes Skill-Set
- Vertrieb ohne App Store, unlimitiert
- Zugriff auf Hardware limitiert
- Offline nicht verwendbar
- Push Notifications nicht möglich



# Progressive Web Apps

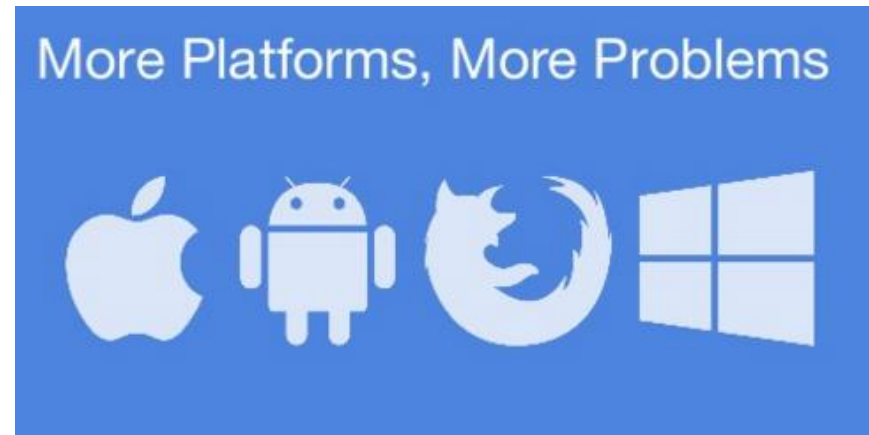
- Neuerungen im Browser
  - Service Workers & Cache → funktioniert (limitiert) offline
    - event-driven scripts (JavaScript)
    - haben Zugriff auf Domänen-Events
  - Push APIs → Push Notifications
- PWAs können direkt auf den Homescreen gepinnt werden
  - kein Umweg über den App Store nötig
  - einfache Installation und Maintenance
- Andere Web App Probleme bleiben bestehen:
  - Offline-Test
  - plattform-spezifische Navigation

Beispiel: <https://quickdraw.withgoogle.com/>

# Hybrid Apps

Mixtur von Native und Web

- Native App Container
- Webtechnologien
  - HTML 5 & CSS 3
  - JavaScript, TypeScript
- Plattform wechseln ist einfach
- Upload in die App Stores ist möglich
- Native Komponenten können verwendet werden:
  - Kamera, Mikrofon
  - Kontakte, Kalender, Fotos
  - Notifications



# Hybrid Mobile App Development

## Tooling



APACHE  
**CORDOVA**™



Phone**Gap**



**Sencha**



# Ionic

- Basiert auf AngularJS
  - GUI Komponenten = Angular Direktiven
  - nur wenige DOM Manipulationen
  - kein jQuery
- Plattform-spezifisches Styling mittels CSS
- Zugriff auf native Komponenten mittels Apache Cordova Plugins
- Ionic 2 (2017)



Demo: <http://ionicframework.com/docs/v2/components/#tabs>

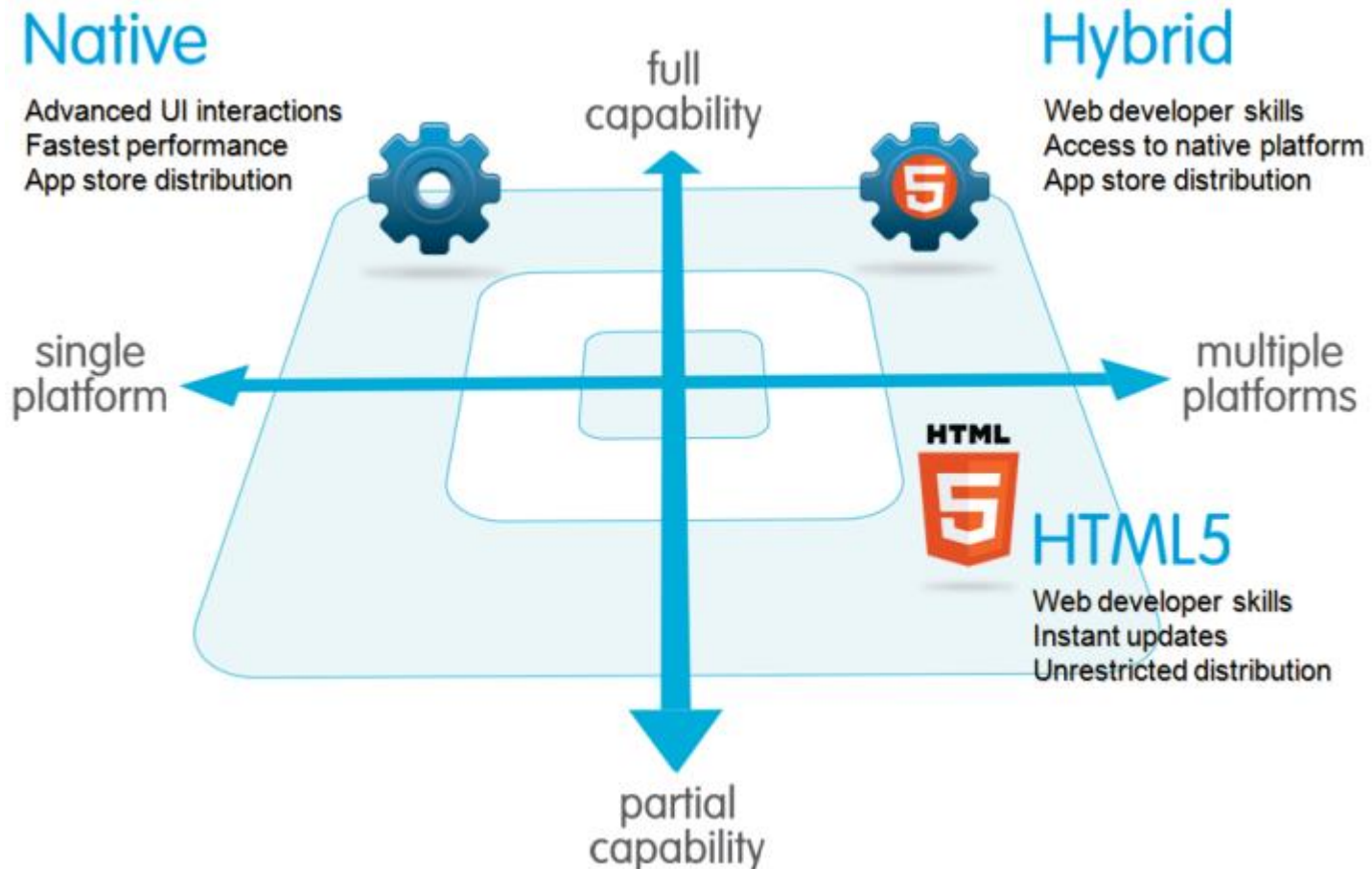
# Übersicht

## Hardware Zugriff, Programmiersprachen

- Native
  - Voller Zugriff auf Hardware, via nativer Schnittstelle
  - Programmiersprachen: Java, Objective-C / Swift, C#
- Web
  - Kein/limitierter Zugriff auf Hardware, via HTML5
  - Programmiersprachen: HTML, CSS, JavaScript
- Hybrid
  - Zugriff auf Hardware via JavaScript, native Kommunikation
  - Apps sind in nativem Container eingepackt
  - Programmiersprachen: HTML, CSS, JavaScript

# Und was ist jetzt die beste Lösung?

Well, it depends...



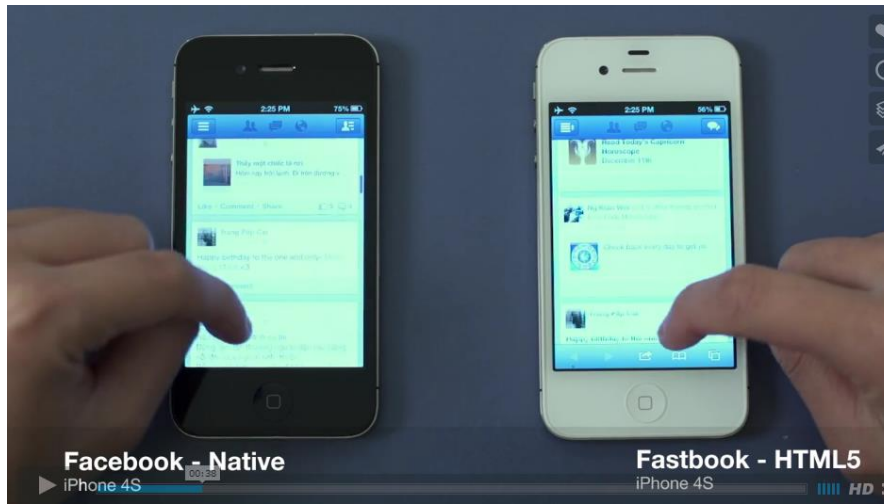
# However...



Mark Zuckerberg:  
Our Biggest Mistake Was Betting Too Much On HTML5  
(2012)

Link: <https://techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/>

# But then again...



Sencha Team:

A side-by-side comparison of how well the HTML5 app performs against both the native iOS and the native Android Facebook apps (2012)

Link: <https://vimeo.com/55486684>

# Gartner, Inc. Analysis (2013)

The battle between HTML 5 web apps and native mobile apps is over

*“Hybrid apps, which offer a balance between HTML5-based web apps and native apps, will be used in over 50% of mobile apps by 2016”*

Links:

<http://www.gartner.com/newsroom/id/2429815> (2013)

<http://www.gartner.com/newsroom/id/2324917> (2013)

# Was denkt ihr?

- Wann verwendet man was?
- Wieso entwickelt man überhaupt etwas anderes als “native”?

# “Na gut, dann also eine Hybrid App...”

Moment! Wie sieht's denn in ein paar Monaten aus?





# Long-term Unterschiede



# Users don't care!

*“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**

# Pros & Cons

## Native App

Pros	Cons
<ul style="list-style-type: none"><li>• Beste Performanz:<ul style="list-style-type: none"><li>• Startzeit</li><li>• flüssigere Animationen</li></ul></li><li>• Voller Zugriff auf die Geräte Hardware</li><li>• UI Komponenten sind garantiert konsistent mit dem Plattform Look-and-Feel</li></ul>	<ul style="list-style-type: none"><li>• Signifikante Entwicklungskosten</li><li>• Keine gemeinsame Code Basis zwischen den verschiedenen Plattformen</li><li>• Vertrieb nur über den App Store möglich</li></ul>

# Pros & Cons

## Web App

Pros	Cons
<ul style="list-style-type: none"><li>• Schnelle und einfache Entwicklung</li><li>• Niedrigere Entwicklungskosten</li><li>• Code Basis kann von jedem mobilen Browser geöffnet werden</li><li>• Benötigt kein App Store Approval</li></ul>	<ul style="list-style-type: none"><li>• Kein Zugriff auf die Hardware</li><li>• Browser-basierte User Experience<ul style="list-style-type: none"><li>• Langsam (JavaScript Engines!)</li><li>• Browser-spezifische Probleme</li></ul></li><li>• Nicht für das Gerät optimiert</li></ul>

# Pros & Cons

## Hybrid App

Pros	Cons
<ul style="list-style-type: none"><li>• Bessere Integration mit der Hardware</li><li>• Schnellere Entwicklung</li><li>• Niedrigere Entwicklungskosten</li><li>• Code Basis geteilt zwischen verschiedenen Plattformen</li><li>• App Store kann verwendet werden</li></ul>	<ul style="list-style-type: none"><li>• Performanz ist abhängig vom Web-Container der jeweiligen Plattform</li><li>• Browserkompatibilität mit Hilfe des Containers verbessert</li><li>• UI Komponenten sind schwer an das Look-and-Feel der Plattform anzupassen</li><li>• Benötigt platform-spezifische Skills</li></ul>

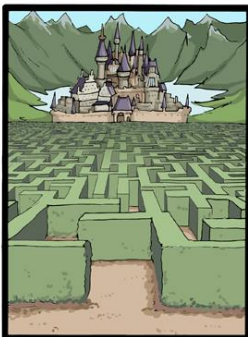
# Wie entscheide ich mich?

Einige wichtige Fragen:

- Was für (plattform-spezifische) Funktionen werden benötigt?
- Komplexität der App?
- Time-to-market?
- Budget?

## The dilemma of mobile apps development

Develop a native app for each device and maintain several projects



Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project



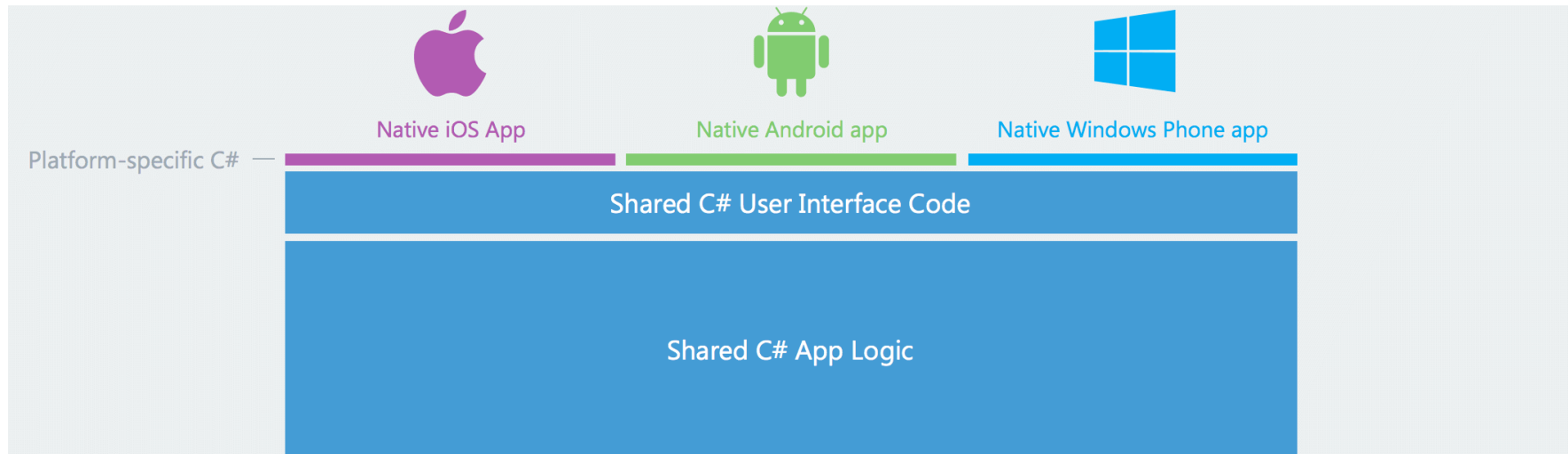
CommitStrip.com

## WHEN DECIDING BETWEEN NATIVE AND HTML5, A GOOD APP STRATEGY SHOULD CONSIDER:



# Übrigens...

Schon mal was von Xamarin oder “Cross Platform Applications” gehört?

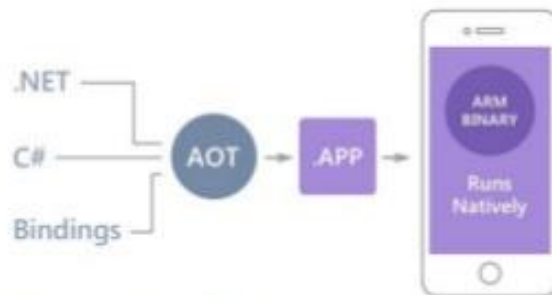


- Geteilte Code Basis für alle Plattformen mittels C#
- 100% native API Zugriff
- Hohe Performance, Apps laufen native

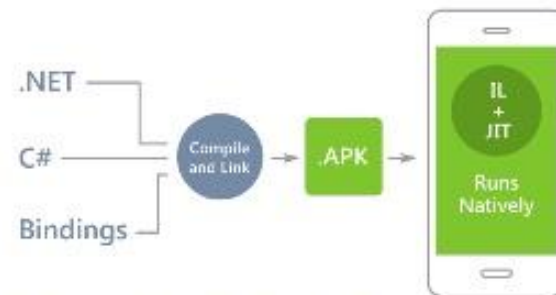
# Wie funktioniert denn das?

Mittels “Cross Compilation”

## Native Performance



**Xamarin.iOS** does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.



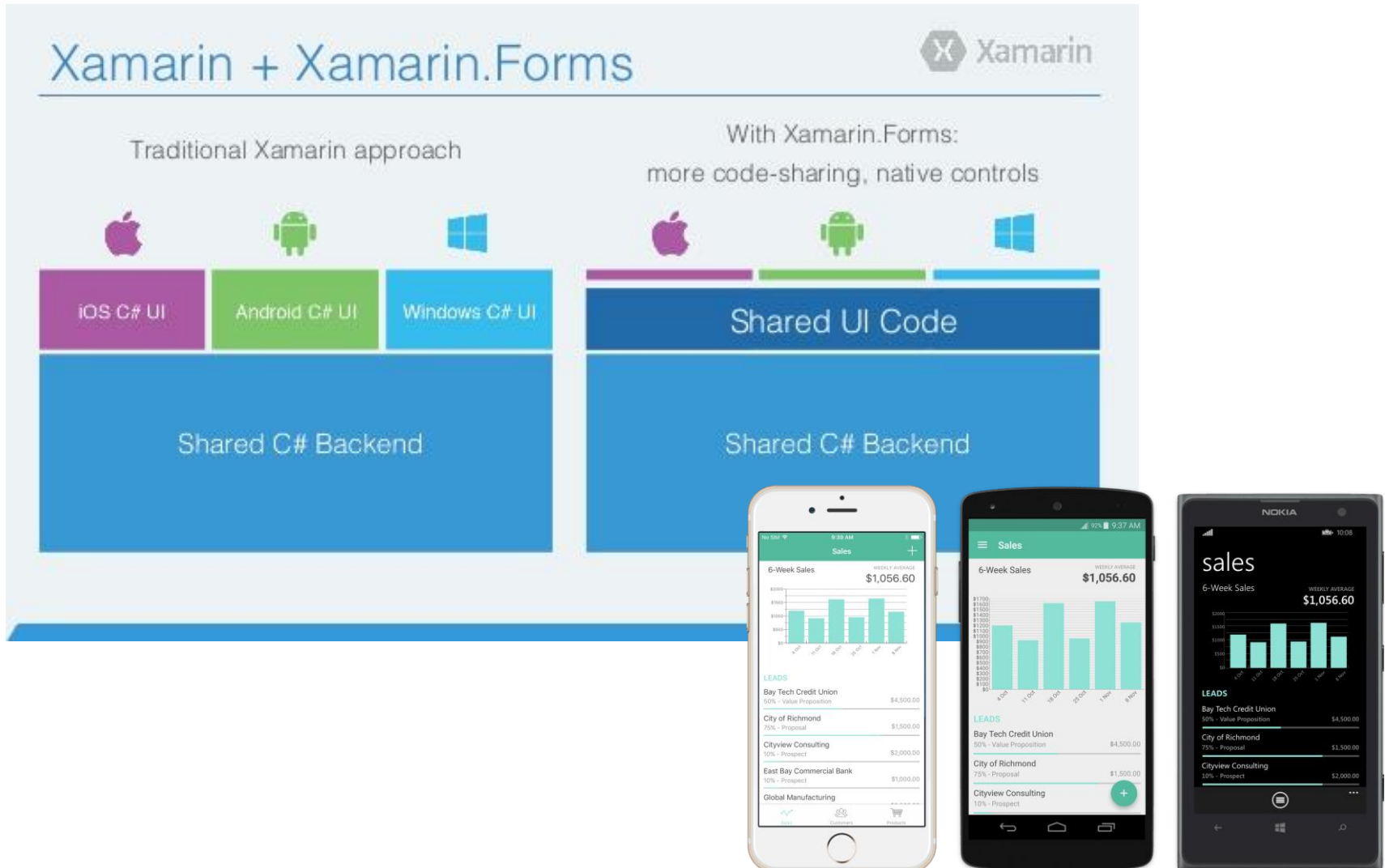
**Xamarin.Android** takes advantage of Just In Time (JIT) compilation on the Android device.

Xamarin The future of apps



# Native UI

## Geteilter Code oder Xamarin.Forms



# Stundenplan

#	Datum	Track 1	Track 2	Raum	Dozent
1	20.02.	Einführung		1.021	Christian Lüthold
2	27.02.	iOS		1.021	Oliver Gepp
3	06.03.	iOS		1.021	
4	13.03.	iOS		1.021	
5	20.03.	iOS		1.021	
6	27.03.	Android	Ionic	1.333 / 1.021	
7	03.04.	Android	Ionic	1.333 / 1.021	Christian Gauch / Roman Rast
8	10.04.	Android	Ionic	1.333 / 1.021	
	17.04.	Osterferien			
9	24.04.	Android	Ionic	1.333 / 1.021	
	01.05.	Tag der Arbeit			
	08.05.	Projektwoche			
10	15.05.	Ionic	Android	1.333 / 1.021	
11	22.05.	Ionic	Android	1.333 / 1.021	
12	29.05.	Ionic	Android	1.333 / 1.021	
	05.06.	Pfingstmontag			
13	12.06.	Präsentationen		1.021	Alle

# Projektarbeit

## Mobile App Projekt

- Es werden 2er/3er-Gruppen gebildet
- Jede Gruppe entwickelt eine App
- Jede Gruppe präsentiert ihre App am 12.06.2017
- Die Apps können mit den folgenden Technologien entwickelt werden:
  - Java (Android, native)
  - Objective-C / Swift (iOS, native)
  - Ionic (Android & iOS, hybrid)
- Das App Projekt wird gezippt und komplett abgegeben
  - **Deadline: Sonntag, 11.06.2017, 24:00 Uhr**
  - An: [christian.luethold@zuehlke.com](mailto:christian.luethold@zuehlke.com)

# Präsentation

- **Zeit:** 5 Minuten pro Gruppe
- **Inhalt:**
  - Vorstellung der App
  - Begründung:
    - Wieso wurde diese Plattform/Technologie gewählt?
    - Was sind die Vor-/Nachteile dieser Plattform/Technologie?
    - Was war eine technische Herausforderung?
    - Was läuft gut und was nicht so besonders?
    - Was würdet ihr beim nächsten Mal anders machen?
- **Bewertung:**
  - Die Präsentation wird direct durch die Jury bewertet.
  - Die Jury kann auch Fragen stellen.
  - Es gibt Preise zu gewinnen. 😊

# Wie bestehe ich diesen Workshop?

Bedingungen:

- Mindestens 80% physische Teilnahme an der Vorlesung
- Teilnahme in einer Gruppe am App Projekt:
  - App wird pünktlich abgegeben
  - App funktioniert
  - Präsentation wird erfolgreich gehalten

# Are you ready?

Nächste Woche geht's nämlich los mit iOS Development!

Ihr braucht dafür:

- Apple Macintosh Laptop
  - Mindestens 1 Laptop pro Gruppe
  - Wer keinen hat, kann sich bei Herrn Adam zur Ausleihe melden:  
[urs.adam@fhnw.ch](mailto:urs.adam@fhnw.ch)
- Installiert XCode (via App Store)
- Optional: iOS Gerät



# Gruppenbildung – Track 1

- Gruppe 1
  - Andreas Gasser
  - Martin Lüpold
- Gruppe 2
  - Jason Dimitratos
  - Albane Beltzung
- Gruppe 3
  - Heimo Paffhausen
  - Matthias Langhard
- Gruppe 4
  - Marco Agovino
  - Line Stettler
- Gruppe 5
  - Remo Koch
  - Melanie Stucki
- Gruppe 6
  - Lea Boesch
  - Marco Bibrich
- Gruppe 7
  - Joel Blumer
  - Ivana Stoilova
- Gruppe 8
  - Sandra Hohl
  - Luzian Seiler

# Gruppenbildung – Track 2

- Gruppe 9
  - Vijeinath Tisaveerasingham
  - Irina Terribilini
- Gruppe 10
  - Bettina Burri
  - Mario Winiker
- Gruppe 11
  - Nicolas Novak
  - Simon Rissi
- Gruppe 12
  - Sascha Schmid
  - Rinesch Murugathas
- Gruppe 13
  - Tobias Baumgartner
  - Dominic Bär
- Gruppe 14
  - Joel Meiller
  - Thierry Bitz
- Gruppe 15
  - Luca Ingold
  - Alina Liburkina
- Gruppe 16
  - Tim Nüchel
  - Kha Dac-Wing
  - Walter Dominique