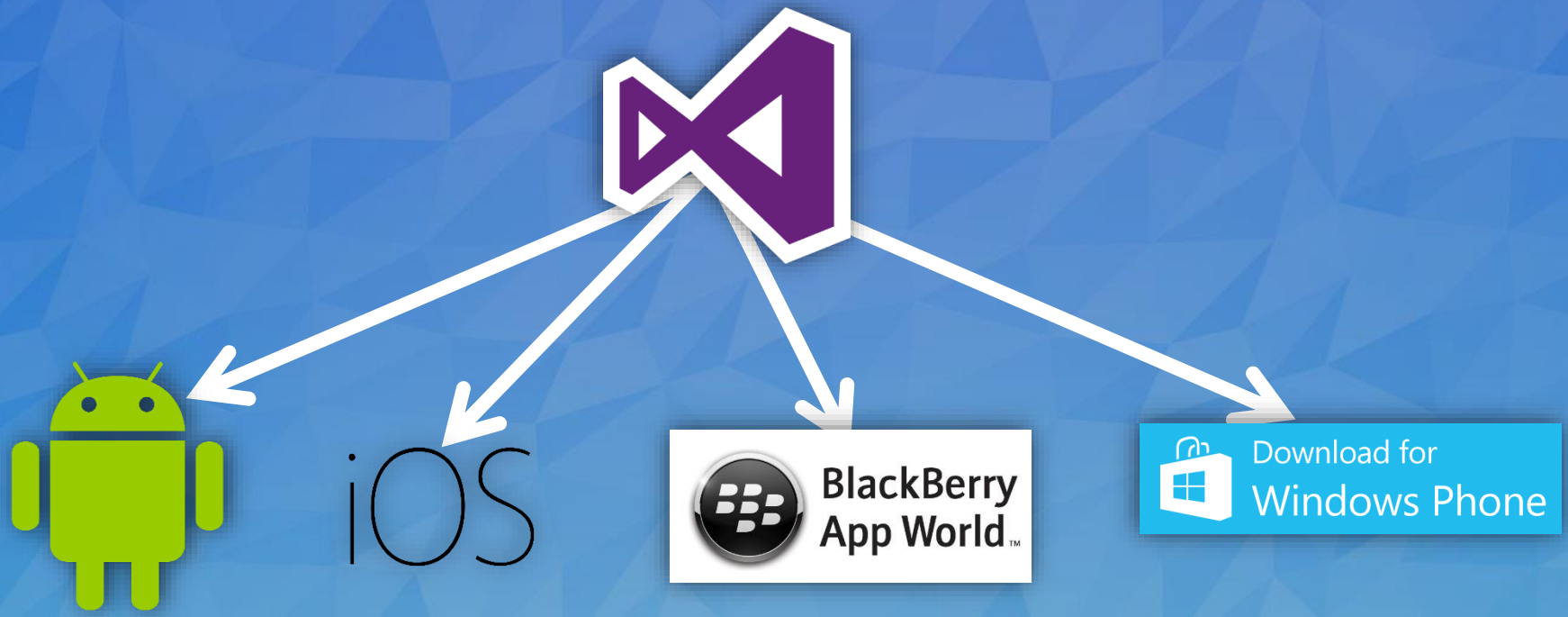
Crossplatform App Development

Gründe für Crossplatform-Entwicklung

Write once

Run everywhere


iOS
BlackBerry
App World
Download for
Windows Phone

Vorteile von Crossplatform-Entwicklung

- Geringere Entwicklungs- und Wartungskosten
- Gleiches „Look-And-Feel“ möglich
- Weniger plattform-spezifische Fehler
- Evtl. größere Reichweite wg. mehr unterstützten Plattformen

Web vs. hybrid vs. native

Web Apps

- Zugriff durch mobile Browser

Native Apps

Direkt Compilierbar C++ z.B. mit cocos2d-x	Über Zwischensprache z.B. Xamarin mit C#

Hybrid Apps

- Web App verpackt als native App

Web-basierter Ansatz

Vorteile:

- Sofort crossplatform (Jede Plattform mit Browser unterstützt)
- Für viele Entwickler einfacher
- Muss nicht im App Store gepublisht werden

Nachteile:

Keine direkte Kontrolle über UI

Verschiedene Browser und Versionen

Teilweise schwieriger Zugriff auf System-Features

Tendenziell schlechtere Performance

Nativer Ansatz

Vorteile:

- Zugriff auf alle System-Features problemlos möglich
- Bessere Performance
- Direkte Kontrolle, auch über UI

Nachteile:

Crossplatform komplizierter

Nativer Ansatz

Direkte Compilierung

Vorteile:

- Bessere Performance
- Nutzung spezieller Systemfeatures
- Direktzugriff auf die native API

Nachteile:

Aufwändige Programmierung

Compilerweichen notwendig

(GCC Compiler für Android, LLVM für iOS)

Compilerweiche Beispiel (cocos2d-x)

```
#if (CC_TARGET_PLATFORM == CC_PLATFORM_WINRT)
CCMessageBox("You pressed the close button","Alert");
 #else
CCDirector::sharedDirector()->end();
```

Zwischensprache

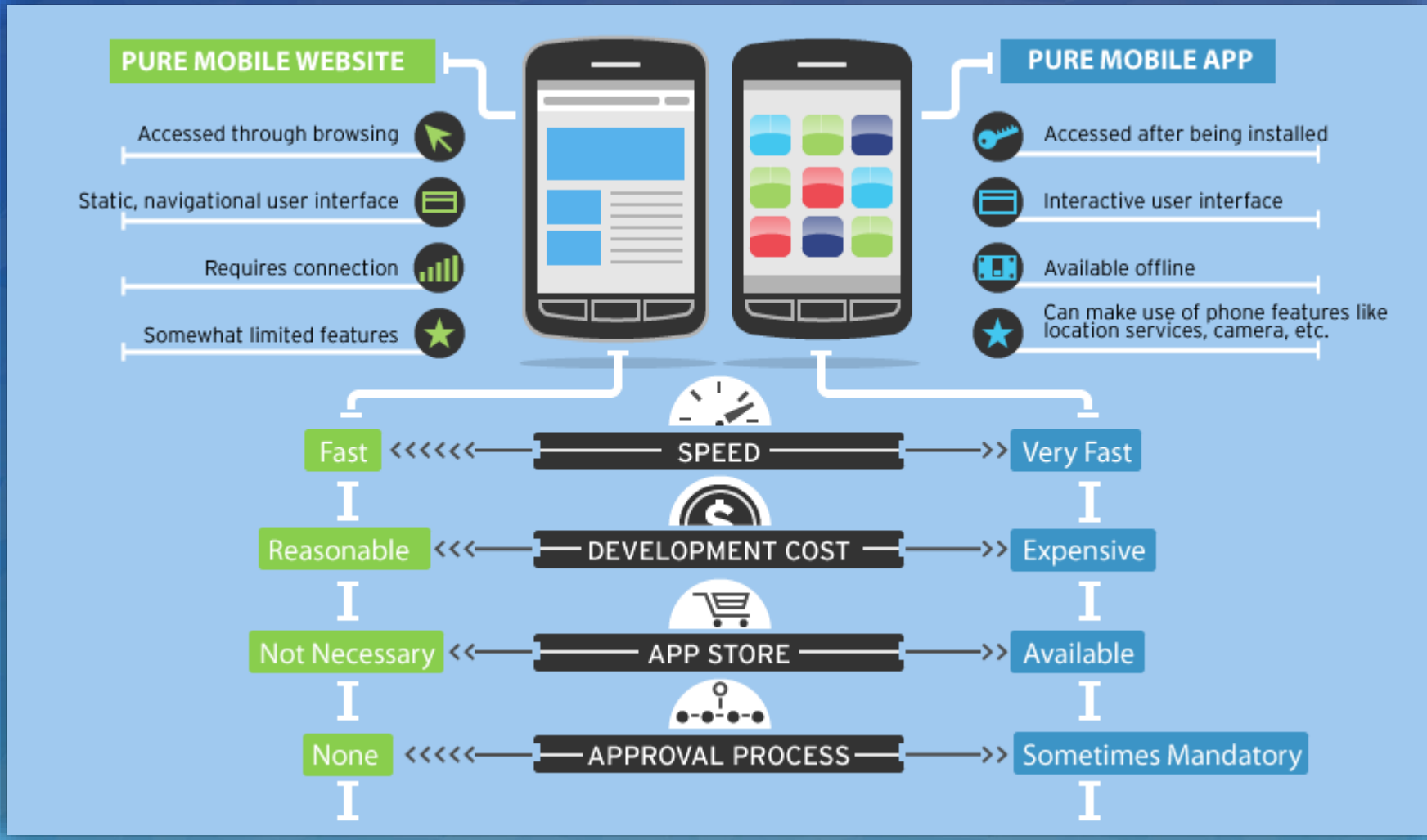
Vorteile:

„Write code once, run everywhere“

Nachteile:

Spezielle Platformfeatures nicht nutzbar wenn nicht in API des Crossplatformtools angeboten

Web vs. Nativ

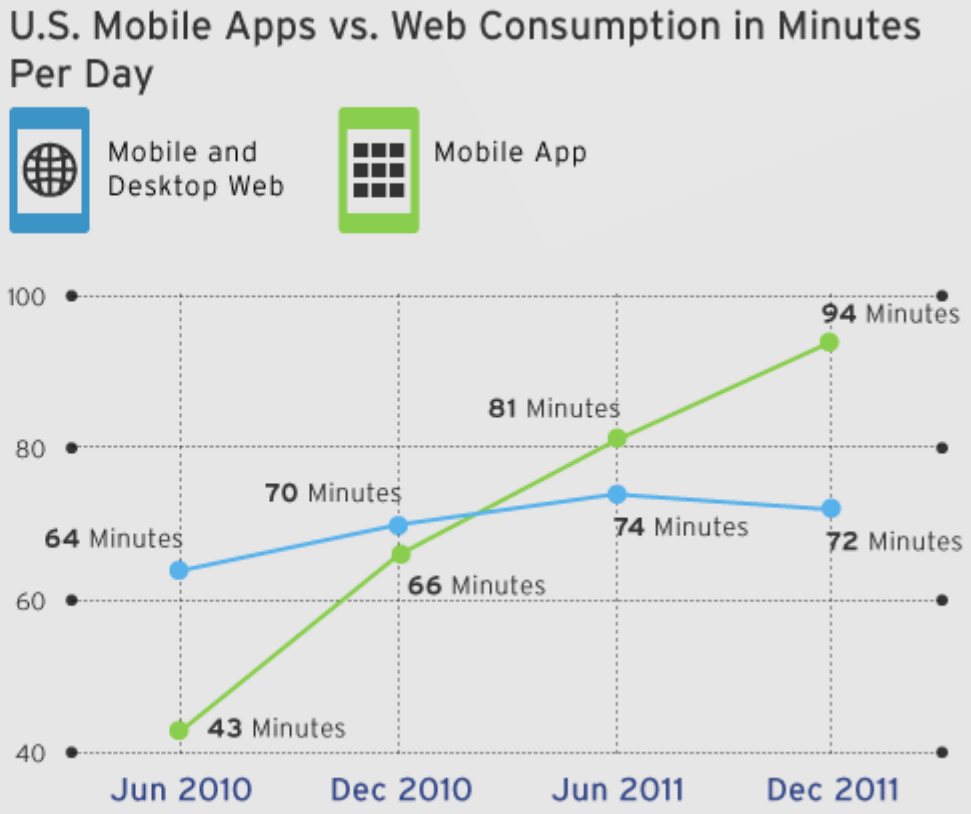


Quelle: http://www.mdgadvertising.com/blog/wp-content/uploads/2013/06/should-you-build-a-mobile-app-or-mobile-website.png

Web vs. Nativ: Zeit

TIME USAGE

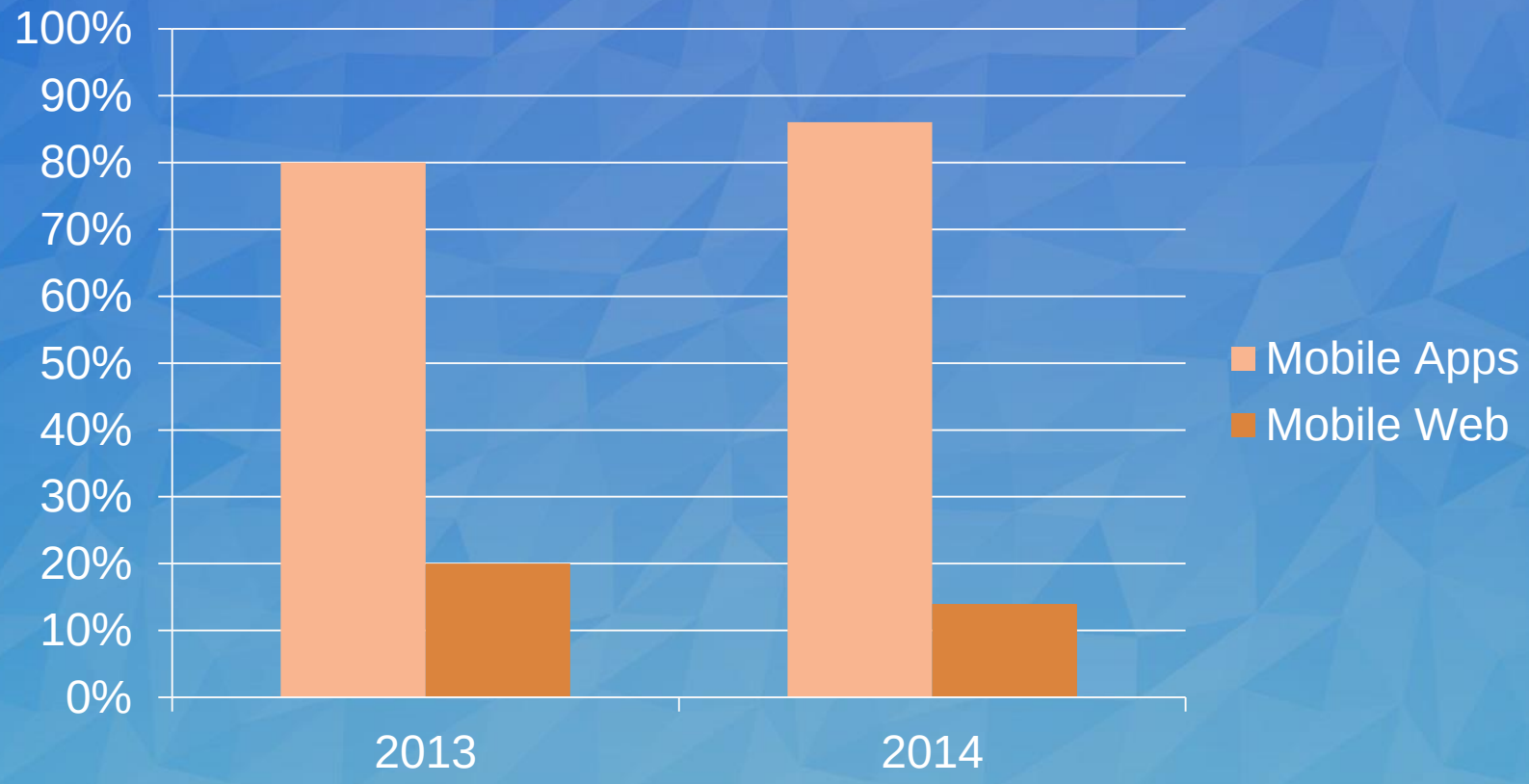
Mobile apps trump entire web (mobile and desktop) in terms of how much time users spend with each. According to comScore and Alexa data compiled by Flurry Analytics, mobile apps are taking up a greater share of users' time each day compared to mobile and desktop web usage, which was dominant just two years ago.



Quelle: http://www.mdgadvertising.com/blog/wp-content/uploads/2013/06/should-you-build-a-mobile-app-or-mobile-website.png

Web vs. Nativ: Zeit

Percentage of time spent on mobile web and apps on connected devices in the US

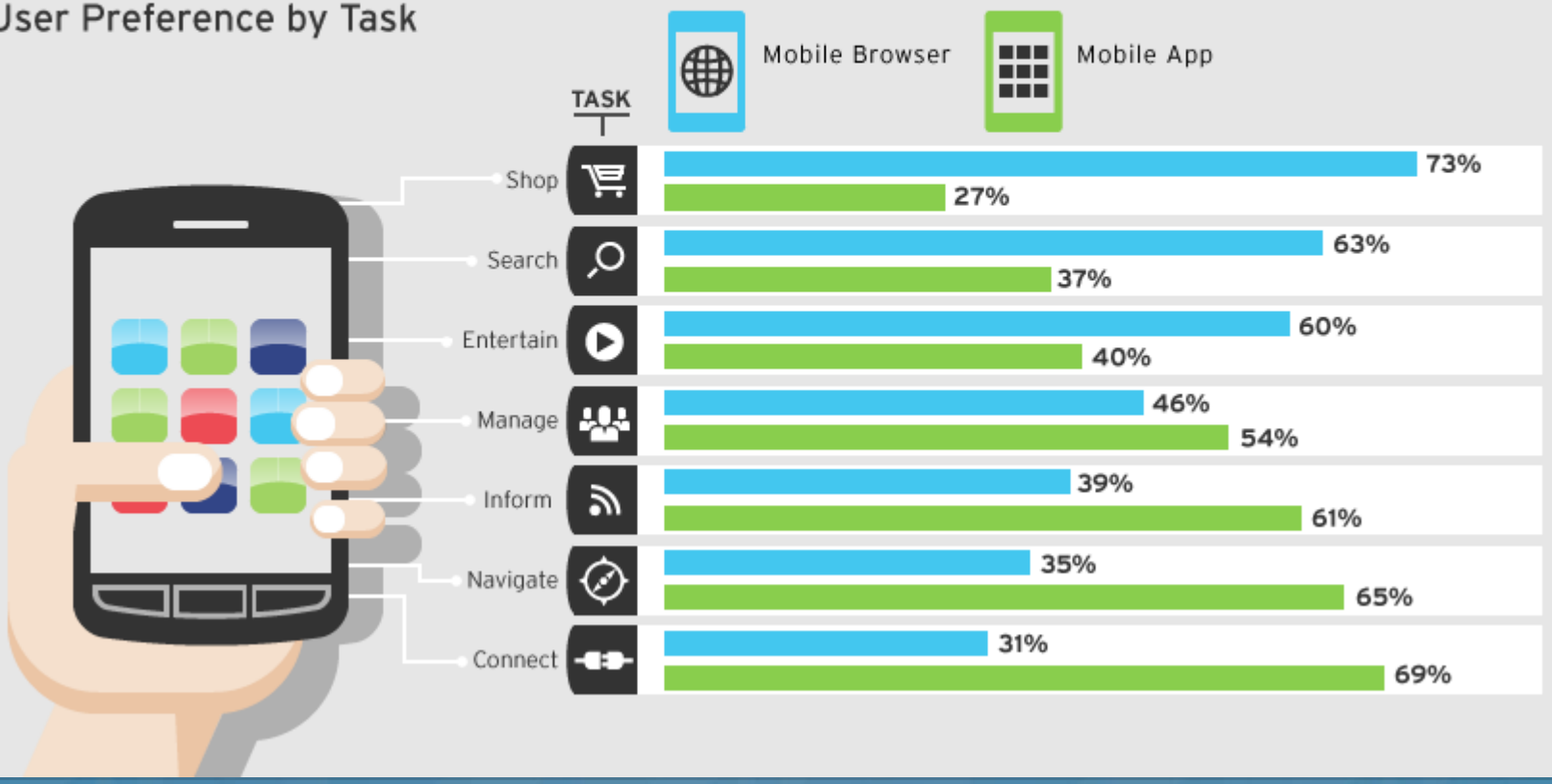


Nutzerpräferenz nach Aufgaben

CONTENT USAGE

For many, the choice of whether to pursue a mobile-optimized site or mobile app will come down to content. How are people going to interact with you on mobile devices? A breakdown of which platform mobile users prefer for different tasks can help.

User Preference by Task



Quelle: http://www.mdgadvertising.com/blog/wp-content/uploads/2013/06/should-you-build-a-mobile-app-or-mobile-website.png

Wähle Native App, falls…

- Spezielle Smartphone-Funktionen benötigt
- Hohe Personalisierung
- Komplexes Design und UI
- Monetarisierung
- Offline verfügbar

Wähle Web App, falls…

- Begrenztes Budget
- Suchmaschinenoptimierung
- Benötigt keinen App Store
- Häufige Updates
- Von überall zugänglich

SDKs zur Entwicklung von Crossplatform Mobile-Apps

- Xamarin
- Cordova / Phonegap
- Appcelerator
- Qt
- Sencha
- RhoMobile
- Alpha anywhere
- MoSync
- Codename One
- Widgetpad

- Ausgeschlossen: Game Engines / Frameworks

Vergleich verschiedener SDKs

	Xamarin	**Cordova**	**Appcelerator**	**Sencha**	**Qt**	**RhoMobile**	**MoSync**	**Codename One**
Kosten	Kostenlos	kostenlos	39$ bis 259$ pro Monat	3855$ bis 4825$	0$ - 175$ pro Monat	0$ bis 999$ pro Monat	kostenlos	0$ bis 79$ pro Monat
Hybrid / Nativ?	Nativ	Hybrid	Nativ	Hybrid	Nativ	Hybrid	Beides	Nativ

- Wichtige Frage:
  - Gleiche UI für alle Plattformen oder
  - UI an jede Plattform einzeln anpassen?

Vergleich verschiedener SDKs

- Cordova: lokal laufende HTML5-App
  - Große Bandbreite an Webframeworks nutzbar (Bootstrap, JQuery e.t.c)
  - **Vorsicht** Gefahr hoch das am Ende eher eine Webseite als eine App rauskommt: Nutze Frameworks die ein App Look and Feel nachbilden z.B. Sencha Touch
- Xamarin: gleiche native Codebasis, aber getrennte Uis
- Appcelerator: komplett gleicher nativer Code

Probleme hybrider Crossplatform-SDKs (z.B. Cordova)

- Alle UI-Elemente werden in HTML gerendert, Look-And-Feel nativer UI wird lediglich simuliert.
- Responsiveness und Qualität nativer UI-Elemente ist unerreicht.
- Qualität des Web View Renderings ist plattformabhängig.