COMPUTERWORLD

OSBC | OPEN SOURCE BUSINESS CONFERENCE

Cloud, Data, Mobile and Open Source: It Starts Here

MAY 21-22, 2012 | Hyatt Regency San Francisco
San Francisco, California

Cross platform app development using open source software

Peter Dickten / Marcus Ross

dcs-fuerth / zahlenhelfer

Cross platform app development

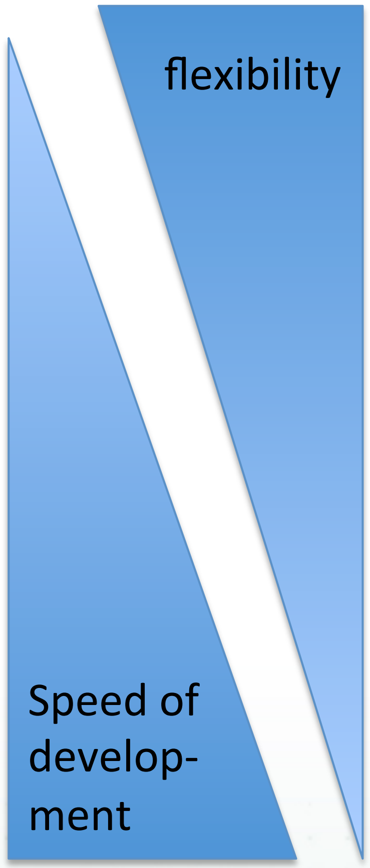
Support for multiple platforms is essential for both enterprise and end users

Relevant platforms:

- Apple iOS
- Google Android
- RIM Blackberry
- Microsoft Windows Phone 7 (?)

4 strategies to address multiple platforms

- Native development for every platform
- Cross-platform tools
- WebView wrappers
- MEAPs (mobile enterprise application platform)



=> M

Strategy 1: Native Development

good

(+) maximum flexibility

(+) mature development tools (e.g. Xcode)

(+) apps have the platform specific look & feel

Strategy 1: Native Development

bad

(-) which platforms should I support?

(-) high cost for the development of multiple clients (no code reuse)

(-) very few developers with knowledge of more than one platform are available

(-) adding one more platform is costly

Strategy 1: Native Development

summary / conclusion

deep pockets and lots of development time needed (even if client development is done in parallel => server side could be a development bottleneck)

=> no solution for our customers (market research and event management)

=> P

Strategy 2: Cross-platform tools

What's that?

- Development of a single source code in one language (e.g. JavaScript)
- Cross compiling/translation/embedding to several native source codes (e.g. Obj.C/Java)
- Usage of the native controls on every platform

Appcelerator Titanium, Rhomobile Rhodes

Strategy 2: Cross-platform tools

good

(+) very flexible

(+) one source code (with some duplication)

(+) one language and one toolset for all platforms

(+) good look & feel

Strategy 2: Cross-platform tools

bad

(-) case distinction needed for nice platform specific user interfaces (many resolutions/ aspect ratios)

(-) tool limits the number of supported platforms

(-) performance lower than purely native apps (not a good choice for 3d shooter games)

COMPUTERWORLD OSBC OPEN SOURCE BUSINESS CONFERENCE

Strategy 2: Cross-platform tools

summary / conclusion

If the platforms supported by the tool match the desired platforms and the apps consist of more that graphics this could be a solution.

Our customers wanted business apps for iOS and Android => good match.

=> M

Strategy 3: Webview wrappers

What's that?

- Development of a pure web application using HTML and JavaScript
- Frameworks could use styling to match the native controls
- Tools like Cordova encapsulate the result in a smartphone app running the web app in the browser of the smartphone

example: jQuery mobile (GUI) / Cordova (Pkg)

Strategy 3: Webview wrappers

good

(+) pretty flexible

(+) lots of platforms supported

(+) one source code

(+) one language and one toolset for all platforms

Strategy 3: Webview wrappers

bad

(-) the result doesn’t always feel “right” on all platforms

(-) slower performance, especially with large datasets

(-) we found very few examples of complex business applications

Strategy 3: Webview wrappers

summary / conclusion

Looks like a good solution if you need to support many platforms and the dataset is limited and the users are not too strict about adherence to the platform standards.

Support of Symbian, older Blackberry and WP7 makes this a good alternative.

=> P

COMPUTERWORLD OSBC OPEN SOURCE BUSINESS CONFERENCE

Strategy 4: MEAPs

Mobile enterprise application platform

What's that?

- Server side infrastructure delivers app logic and data to a generic client app
- Mostly for enterprise internal apps (e.g. support app for sales people)
- Server side components have interface to enterprise software (SAP, Siebel, RDBMS…)

Sybase Unwired Platform, Kony, Verivo, Syclo

Strategy 4: MEAPs

good

(+) very fast app ~~development~~ configuration

(+) lots of platforms supported

(+) works nicely with enterprise software

Strategy 4: MEAPs

bad

(-) very limited functionality

(-) server side infrastructure very complex and expensive

(-) client app user interfaces are often disgusting

(-) no solution for end users

(-) not open source

Strategy 4: MEAPs

summary / conclusion

If you have extremely deep pockets and want to push data from your existing SAP/Siebel/… infrastructure to 10.000 sales people around the world, this could be a solution.

No match for our use cases.

COMPUTERWORLD OSBC OPEN SOURCE BUSINESS CONFERENCE

Summary of solutions

- ~~Native development for every platform~~
- **Cross-platform tools**
- **WebView wrappers**
- ~~MEAPs (mobile enterprise application platform)~~

=> M

The evaluation project

Build a sample app …

- connecting to a server side application
- Reading and creating data
- GPS access
- Optional: push notification, ads
- (…)

We are Germans.

This is our chancellor

And yes … we drink a lot of alcohol.

Not only beer, but also cocktails.

Idea: MartiniApp ("yelp for martinis")

- User can create reviews of cocktail bars serving martini cocktails
- User can search for reviews based on their location
- Lots of ideas for revenue generation …
  - Happy hour catalogue
  - Coupons
  - Ability to buy someone a cocktail through the app

Dear VCs: Please contact us ;-)

=> M

Components of the project

- server side:
  - (Ruby on Rails-based) server implementing …
  - API to read and write reviews
- client side:
  - Smartphone app
    - One created with Appcelerator Titanium
    - Another created with jQuery Mobile / Apache Cordova

Apache Cordova overview

Development in pure HTML / JavaScript

Target platforms:

- iOS (iPhone, iPad, iPod touch, ...)
- Android
- Blackberry 5/6/7
- webOS
- Symbian
- Windows Phone 7

Android iOS

WebOS Windows Phone

Blackberry Symbian

Apache Cordova: concept

Write once, compile everytime

- One source (HTML5/CSS3/JS)
- Package to native through a WebView
- Separate tool chains for each platform (Native SDK of platform)

Apache Cordova API

The Cordova SDK delivers a JavaScriptAPI to access smartphone features

Accelerometer
Camera
Capture
Compass
Connection
Contacts
Device
Events
File
Geolocation
Media
Notification
Storage
+Plugins

Apache Cordova 101

- // load Cordova
  ```
  <script src=“cordova.js"></script>
  ```
- // wait for Cordova to init
  ```
  document.addEventListener("devicer
  eady", onDeviceReady, false);
  ```

Apache Cordova example

```
function onDeviceReady() {
   alert(‘Name:’+ device.name);
   alert(‘Platform:’+device.platform);
}
```

jQueryMobile: concept

- declarative UI (HTML5 data-attr.)
- target to touch devices
- progressive enhancement

  (A, B and C grade Browsers)

jQueryMobile 101

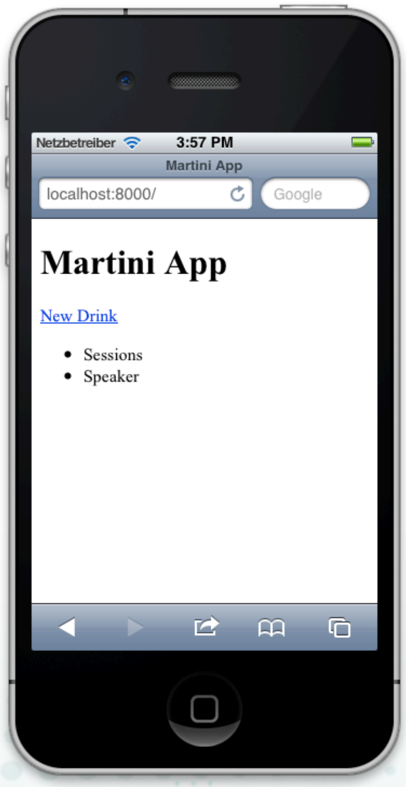
- // load jQM

```
<link rel="stylesheet“
 href=„jquery.mobile-1.0.1.min.css" />

<script src=„jquery-1.6.4.min.js">
</script>

<script src=„jquery.mobile-1.0.1.min.js">
</script>
```

jQueryMobile example

- // show a List

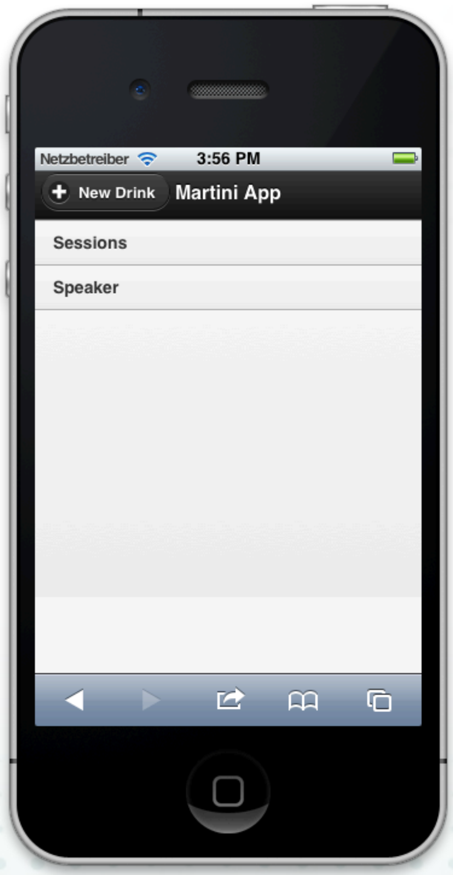
```
<div data-role="page" id="home">
  <div data-role="content">
      <ul>
          <li>Sessions</li>
          <li>Speaker</li>
      </ul>
  </div>
</div>
```



Martini App

New Drink

- Sessions
- Speaker



Sessions

Speaker

=> P

Demo time!

Appcelerator Titanium overview

Development in pure JavaScript (*), compilation/ embedding to/in native languages (ObjC/Java)

Target platforms:

– iOS (iPhone, iPad, iPod touch, ...)
– Android
– Mobile Web (HTML5)
– beta: Blackberry

(*) you can write/use platform-specific addons in ObjC/Java

Appcelerator Titanium: concept

Write once, ~~run~~ adapt everywhere

- One source code, but case distinctions for UI:
  - If android ... then ... else ...
  - Relative/percentage positioning
  - JSS (similar to CSS) for each platform/resolution
- Separate image folders for each platform/ resolution

Appcelerator Titanium API

The appcelerator SDK delivers a javascript API to access smartphone features like *Accelerometer, Analytics, Notifications, Contacts, Database, Facebook, Filesystem, Geolocation, Gestures, GUI, Locales, Maps, Media, Networking, Settings, XML, Yahoo-APIs* (...)

- Some APIs are platform specific
- APIs are appstore-safe.

Titanium UI example

UI is build completely in code: No HTML/GUI designer (*)

```
var btnLogin = Titanium.UI.createButton({ title : 'login',
        top : 160, left : 10, right : 10, height : 40});

btnLogin.addEventListener('click', function() {
        // do something
}
```

(*) but there‘s at least one 3rd party tool

Titanium API example

```
var db=
   Titanium.Database.open(‘martiniDB');
```

DML-Kommando:

- db.execute(
  'INSERT INTO cocktails(beverage) VALUES(?)', 'dry martini');

Demo time!

COMPUTERWORLD OSBC OPEN SOURCE BUSINESS CONFERENCE

Some extras:

- You can extend applications by native modules (written in ObjC / Java)
- a very active marketplace for 3rd-party-extensions (“modules”)
- Commercial support/training/certification is provided by Appcelerator
- Appcelerator offers cloud-based services (e.g. push notifications) as optional services
- Codestrong: Conference in October in SF

=> M+P

Final Winner?

- Cordova+jQueryMobile
  - Freedom of HTML5
  - many supported platforms (6)
  - JavaScript knowledge not required
- Titanium Mobile
  - fast & native UI of the target platform
  - more APIs
  - commercial support (SLA)

Q&A

Thank you !

Peter Dickten (@Pe_D)

and

Marcus Ross (@zahlenhelfer)