Porting Existing PhoneGap Apps to Tizen OS
- Development Story

Anil Kumar Yanamandra
Thomas Mitchell

ProKarma

TIZEN
DEVELOPER
CONFERENCE
2013
SAN FRANCISCO
About ProKarma...
Who am I?

• Anil Kumar Yanamandra
• Mobile Architect & Head – CoE for Mobility
• @ProKarma
  • Mobile Application Development: B2C, B2B
  • Native (iOS, Android, BlackBerry), Hybrid, Mobile Web apps/sites
  • Custom MDM solutions
  • Mobile Dev Governance: Coding standards, Best practices
  • Mobile Consulting: Native or Hybrid or Web? Requirements driven architecture
  • R & D
Agenda

• What is PhoneGap?
• Our porting experiment
• Changes and issues
• Demo
• Lessons learnt
What is PhoneGap BTW?
PhoneGap

- Open-source
- Cross-platform
- Web standards based
- Mobile App Development Framework
- Access native features via Web APIs
- Pure backend framework: Works with any web UI framework
- Extensible: Add custom features
Typical Mobile App Architecture – Native

Native View-1 ➔ Network Manager ➔ Device OS ➔ iOS/Android/Tizen

Native View-2 ➔ Persistence Manager ➔ Device OS ➔ iOS/Android/Tizen

. . .

Native View-X ➔ Geo-Location ➔ Device OS ➔ iOS/Android/Tizen

Native API calls - PS (UDID, Contacts, Camera)

PS – Platform Specific

iOS – Objective-C
Android – Java
Tizen – C/C++

Native app package for distribution (.ipa, .apk, .tpk)
Typical Mobile App Architecture – PhoneGap

Native WebView

JavaScript/AJAX (PI)

Native API calls handled by PhoneGap library. These would be perceived as JavaScript API calls by the HTML views. (And the developers!)

Native app package for distribution

PhoneGap Library

Network Manager
Persistence Manager
Geo-Location
PIM

PS

Backend

Custom Plugin

Custom Plugin

PI – Platform Independent
PS – Platform Specific

Device OS

iOS/Android/Tizen

HTML views (PI)

CSS styling (Partially PS)
Interesting. But why bother when I have Tizen Web API?

- **Code Reuse!**
- **Single codebase across platforms**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Native</th>
<th>PhoneGap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>TelephonyManager mTelephonyMgr = (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE); String deviceUid = mTelephonyMgr.getDeviceId();</td>
<td>var deviceUid = device.uuid;</td>
</tr>
<tr>
<td>iOS</td>
<td>NSString deviceUid = [[UIDevice currentDevice] uniqueIdentifier];</td>
<td>var deviceUid = device.uuid;</td>
</tr>
<tr>
<td>Tizen</td>
<td>String deviceUid; SystemInfo::GetValue(<a href="http://tizen.org/system/duid">http://tizen.org/system/duid</a>, deviceUid);</td>
<td>var deviceUid = device.uuid;</td>
</tr>
</tbody>
</table>
Our porting experiment
Automobile shipment app

- **Use-case:** Ship vehicles from the comfort of your mobile device
- **Features:**
  - Role based User Registration
  - Scan VIN barcode to auto input Year, Make, Model
  - Get Quote
  - Save Quote
  - Place an Order
  - Pay by Credit Card
  - Track Orders
  - Map shipment locations
  - Review orders
Tech Details

• **Platform support:**
  ✓ iOS: iPad and iPhone
  ✓ Android: Tablet and Smartphone
  ✓ BlackBerry 10

• **Tech Stack:**
  • Native shell
  • PhoneGap
  • HTML5 stack: HTML/CSS3/JavaScript
  • AJAX
  • Only “Native” feature: VIN scanner
JavaScript libraries used

- Backbone.js
- iScroll 4
- Require.js
- jQuery mobile
- less
- Mobiscroll
- Modernizr
Screencaps: Tablet and Smartphone apps
Porting to Tizen OS: Changes

• Extract web codebase from the Android/iOS PhoneGap project
• Import to new Tizen project
• Define config.xml settings
• Whitelist all URLs the app would use
• Remove the JS hooks to “native” VIN barcode scan plugin
• No more!
Debug

- **What worked?**
  - Almost everything!
  - Responsive UI: media-queries, all the JavaScript libraries

- **What needed “fixes”?**
  - “font-face” CSS code that loads custom font using OTF files did not work
  - “Pull-to-Refresh” feature in iScroll plugin

- **Issues: Tizen dev device RD-210**
  - Performance
  - Load times
  - Refresh rates
Demo Time!
Final result
Lessons learnt
Summary: Why was this porting successful?

- Stick to standard web technologies: Avoid Third party frameworks with custom JavaScript translation layers.
  - Keep up with the latest supported Libs!
- Employ Responsive Web Design principles:
  - Build for varying screen sizes
  - Build for varying device capabilities
- Tizen’s outstanding HTML5 compliance makes porting HTML5 based apps easier and economical