Hybrid apps for rapid mobile development

Dr Alex Jonsson
Co-founder, Evothings

Eindhoven High Campus, June 3, 2014
New role(s) for mobile devices in IoT

Layers/nodes in IoT stack:
- APIs
- Cloud
- Fog (local cloud)
- Gateway, controller
- Sensors, hardware

Communication with the IoT stack

Visualization
Remote control
Storage
Analytics
Sensor

Remote control
Storage
Analytics
Sensor
Companies opening up their platforms and products to the developer community

- ARDUINO
- Raspberry Pi
- Texas Instruments
- beaglebone
- TI Sensor Tag
- Ford Developer Program
- GM For Developers
- SmartThings
- Ninja Blocks
- nest
- hue
- PHILIPS
- NIKE+ API
- UP
- fitbit
Examples of IoT software technology – APIs, libraries, protocols, standards

- **Weightless**: Wireless long range, low power, low cost communication to connect devices
- **MQTT**: Lightweight, high reliability messaging transport protocol
- **Bluetooth**: Aka BLE, BT4.0, wireless low power, short range communication
- **XMPP**: Real-time messaging communication protocol
- **IBEACON**: BLE based positioning and “push” communication technology from Apple
- **ZigBee**: Communication protocol for personal area networks with mesh capabilities
- **Z-Wave**: Wireless low power communication protocol for home automation
- **WiFi Direct**: Wifi standard that enables devices to connect directly without a router
- **OpenXC**: API to access a vehicle’s internal network & data, initiated by Ford
- **DMX512**: Communication protocol for control of stage lighting and more
- **DDS**: Standard to enable scalable real-time high performance data exchange
- **Canbus**: Communication standard for Automotive and vehicles

- New technology, often developed specifically for IoT/M2M, is crucial for scalable solutions
- We expect a lot more new IoT specific technology to enter the market going forward
Three development paths for apps (three slides)

Native
- Advanced UI interactions
- Fastest performance
- App store distribution

Full capability

Hybrid
- Web developer skills
- Access to native platform
- App store distribution

Multiple platforms

HTML5
- Web developer skills
- Instant updates
- Unrestricted distribution

Single platform

Partial capability

Source: http://svitla.com
Native development

Mobile OS is a two-horse race between Google’s Android and Apple’s iOS.

iOS development
• xCode, Interface Builder (Apple)
• Native 3:rd party libraries and frameworks
  • 3twenty (UI), SQLite Admin (db)

Android development
• Eclipse + Android developer tools (java)
• Native Android Development Kit (C/C++)
• 3:rd party tools
  • App Inventor for Android (graphics)
  • The Simple project (import from basic)

There are also other systems, e.g. Windows Phone, developed using the Visual Studio IDE (Microsoft).
Mobile web apps

- **Web apps** are not “real” applications
- **They are html web pages** which *look and feel* similar to native application
- Run by an on-board browser on the mobile device, written in HTML5.

**Graphical Tools**
- Drag-n-drop tools for download, or running as a service in the browser.
- Simpler app, strongly templated apps with fixed functionality.

**Code-based web tools**
- More freedom than the graphically oriented tools.
- Third-party javascript libraries
- Supporting functions
  - access to device APIs
  - cloud-side functionality
  - persistence
Hybrid apps are proper applications, and can be published on app stores.

- Part native
- Part HTML5, running in a web view

Template systems
- Native parts are prepared ahead of time
- UI, navigation, logic using scripting
- Signing w. Apple’s and Google’s tools
- Hybrid tool suites;
  - Adobe Phonegap
  - IBM Worklight,
  - AppMobi (Intel)
  - Evothings.

Plug-ins for adding functionality
- Ready-made functions, per feature, per platform
- For IoT quite early days; Low Energy plug-in, iBeacon, XMPP
- Way faster than one app for each platform.
Hybrid processes on mobile

Source: neebal.com blog
<table>
<thead>
<tr>
<th>SUMMARY</th>
<th>Native apps</th>
<th>Hybrid apps</th>
<th>Web apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>App are developed in...</td>
<td>Objective-C in xCode, or Java in Eclipse, Appbuilder</td>
<td>HTML + Javascript + text editors and via 3:rd party frameworks like Phonegap/Cordova</td>
<td>HTML/Javascript via text editors, graphical environments, browser-based editors</td>
</tr>
<tr>
<td>Advanced functionality is enabled by...</td>
<td>Writing code for each platform, 3:rd party tools for UI, structure like 3sixty</td>
<td>Via pre-written plug-ins and creating bespoke plug-in functions for special purposes.</td>
<td>Built-in functions in the mobile browser, while fairly limited and non-modifiable</td>
</tr>
<tr>
<td>Apps are tested with...</td>
<td>iOS emulator, Android emulator, real devices</td>
<td>Using real devices, also in emulation on the computer.</td>
<td>In a web browser, on the PC or a phone.</td>
</tr>
<tr>
<td>Apps are signed/certified using...</td>
<td>Apple’s and Google’s signing tools</td>
<td>Apple’s and Google’s signing tools</td>
<td>No signing required</td>
</tr>
<tr>
<td>Apps are published on...</td>
<td>Apple’s App Store, Google play and other stores</td>
<td>Apple’s App Store, Google play and other stores</td>
<td>Any web server, or cached locally on phone</td>
</tr>
</tbody>
</table>
When you edit the code, any changes are pushed to connected phones!

Instantly, w/o compile time
Evothings Studio
Develop together with customers
EvoThings Studio

Snappy good-looking IoT apps made easy

Create UIs using proven high-performance HTML5 technologies – compatible with open standards!
Evothings Studio

Native performance under the hood

Pre-built JavaScript libraries, examples and templates – all ready to use!
Evothings Studio

Example customer scenario

Youtube link to video

Courtesy of fuffr.com
Want to add mobile? Come talk to us!

Dr Alex Jonsson
Co-founder, VP Community
alex@evothings.com
+46 704 331 312

Tomas Uppgård
Co-founder, CEO
tomas@evothings.com
+46 709 201 020