

# The Definitive Guide to Choosing Your Enterprise Mobile Application Architecture

NATIVE vs WEB vs HYBRID

# The Mobile App Deluge

Mobile application development has become a critical business function as enterprises look to generate revenue and improve the customer experience through mobile apps.

As the demand for mobile apps grows, so does the development queue. According to a study by Opinion Matters, 85% of companies have a mobile backlog of between one and 20 applications, with half having a backlog of between 10 and 20 apps<sup>1</sup>.

You can't afford to have your competitive differentiator sitting in the development queue. If you know exactly what you want, it can be convenient to just outsource the work for a price, and simply build its cost into your budget. But developing a mobile application is not a one-time effort. Hiring a freelance developer or marketing agency to deliver a ready-to-ship mobile app is often a costly temporary fix, with long-term implications that are often overlooked.

According to MGI Research, most mobile apps will experience, in a two-year timeframe, at least four major update cycles stemming from operating system and device updates.<sup>2</sup> This means that buyers often find themselves in an unexpected "money pit" because they need to keep engaging with the original developer to fix things so their app remains compatible with each new wave of mobile operating systems and devices. Not to mention an inevitable, growing list of desired feature additions and functional tweaks.



## Solving the Million Dollar Question

The cost of building a mobile business app today typically ranges from \$50,000 to \$150,000, depending on its complexity. According to Forrester Research, the cost of building the first version of a native mobile app represents about 35% of the true two-year cost of the app!<sup>3</sup> This means the true cost of a native mobile app is between \$140,000 and \$425,000.

If you plan on building more than three apps during the next year, picking the right development strategy and architecture might be a million-dollar decision — or more. The long term financial implications of mobile app development are becoming quite clear. And while innovating to gain a competitive edge is more critical than ever, it's also clear that you need a financially viable and sustainable means of developing and maintaining mobile apps. This is where selecting the right mobile application architecture comes in.

But selecting the right mobile architecture is just the beginning. Many organizations that have gone down the mobile path have quickly discovered that traditional approaches to application development and delivery are fundamentally inadequate to keep up with the realities of enterprise mobility. Leading companies have found great success with integrated mobile and web platforms that not only allow them to solve their enterprise mobility problem but help them address the entirety of their application development and delivery challenges.

"The true cost of developing a native mobile app is between \$140,000 and \$425,000" Strester Research

<sup>1</sup> Opinion Matters, Mobile App Backlog Is Directly Damaging Revenue in the Enterprise <sup>2</sup> MGI Research, Mobile Enterprise Application Platforms (MEAPs): A Buyer's Guide

<sup>3</sup> Forrester Research, Your Company Needs A Mobile Organization

# Mobile Application Architectures At a Glance

The mobile application architecture you choose dictates how quickly and cost effectively you can release new and updated mobile apps to your user base. Many factors come into play when choosing the best mobile application architecture for your app. We'll go through those later. Here is a quick overview of the mobile application architecture types.

#### Native apps

- Run on a specific device and mobile operating system
- Built using native code (e.g. Objective-C or Swift)
- Downloaded from an app store
- Live on the device
  - **EXCEPTION ONLY IF REQUIRED**

#### Mobile web apps

- Run on any device with a web browser
- Built using standard web technologies
- Served from an application server

#### DEFAULT - MOST COMMON CHOICE

Hybrid Web Apps	Hybrid Mixed Apps	Packaged Hybrid Apps
Run on multiple mobile operating systems (after fine-tuning for each operating system)		
Built using a cross-platform framework		

- Downloaded from an app store
- Part lives on the device and part served from an application server

#### ALTERNATIVE - MOBILE WEB+

### How a Rapid Application Delivery (RAD) Platform Simplifies Mobile Architectures

It quickly becomes obvious after looking through the different characteristics of each of these architectures that one size definitely does not fit all. The reality is that any enterprise hoping to properly embrace mobility will be dealing with different apps that demand different architectures. Traditional approaches are simply not scalable enough to handle the level of variability in target platforms, form-factors, back-end integration, and technology (churn and skills) multiplied by the velocity of change requests across a portfolio of apps. The simple truth becomes increasingly evident, there must be a better way. As a companion to each architecture, we will augment this guide with how the enterprise Rapid Application Delivery platform from OutSystems tackles the entire scope of this amazingly complex problem.

## **Native Applications**

A native app is designed to run on one specific mobile operating system, such as iOS, Android or Windows Phone. It is built using the operating system vendor's technology and, typically, using development tools supplied by that vendor. For example, native Android apps are usually built in Google Android Studio using Java, while native iOS apps are built in Apple's XCode using the Swift or Objective-C programming languages. This approach allows developers to use device features and capabilities, such as a camera, GPS or 3D engine.

Native apps connect to external systems for data and functionality via standard API web technologies, such as REST, exposed by backend application servers or mobile back-ends as a service (mBaaS).

#### Benefits of developing a native app:

- Developers have access to all device features and functionality
- Developers can deliver a very complex user experience such as gaming or multimedia
- > The app can be distributed via the Apple, Windows or Google store

#### Disadvantages of developing a native app:

- Development costs are high, and timelines can be long, due to extensive hand-coding
- Organizations must build and maintain a different code base of the same app for each OS
- Developers must learn a different coding language for each version of the same app
- ▶ Version distribution can be slow due to the app store approval processes

### Bottom line

Build a native app when you want to deliver an extraordinary user experience at any cost. Games, like Angry Birds, are a good candidate for native apps.



# How does OutSystems Platform handle Native Apps?

Most native apps, especially business applications, are not self-contained on the device. Integration and communication with a myriad of back-end systems for data, process and workflow is a constant and challenging requirement. And in the case of many modern apps, these varied back-end servers - living both in corporate data centers and the cloud - may be leveraged concurrently.

OutSystems Platform provides the fastest and simplest way to build an enterprise grade mBaaS (mobile backend as a service) for your app. You can visually model support services that integrate with any corporate database, API or enterprise system. You can mash, cache, aggregate, orchestrate, integrate, add custom logic/process, and expose all this data via RESTful APIs - an easy way for native apps to connect with your servers - and scale it with one click, with no impact on your corporate systems. A wide range of application services (user management, social connectors, background jobs, etc.) support both your native app and a unified web presence, along with all your app's back office needs.

# **Mobile Web Applications**

A mobile web app can be accessed from any device running a web browser regardless of its underlying operating system. Mobile web apps are built using standard Web technologies such as HTML5, JavaScript and CSS.

Mobile web apps are built on top of web technologies. Screens (pages) and business logic are served from an application server.

#### Benefits of developing a mobile web app:

- Developers only need to maintain one code base
- The app is immediately available to users regardless of their device platform or underlying operating system (omnichannel app)
- Developers familiar with web app technologies can build this type of app
- New versions are instantly available
- Users do not have to install the app

#### Disadvantages of developing a mobile web app:

- Developers have limited access to device native features and functionality
- Web technologies offer limited performance in very complex user interactions, such as animations
- The app cannot be distributed via the app stores, where users are accustomed to looking for new apps

### **Bottom line**

Build a mobile web app when you want your application to be accessible from any device, without the need to install an application from an app store. An example of a mobile web application is the Financial Times web app. WEB APP BACKEND SERVER

# How does OutSystems Platform handle Mobile Web Apps?

OutSystems allows for a single authoring environment and codebase with the ability to create and publish a best-in-class user experience for all means of access, including mobile web applications. To maximize usability, mobile web

applications built in OutSystems Platform include support for responsive design, meaning that the user interface dynamically optimizes itself for the user's device, regardless of operating system, screen resolution or device orientation. The operational capabilities are similarly powerful, with one-click publishing directly from the development environment, which dramatically streamlines testing, production, and lifecycle management. For use cases where specific device capabilities need to be leveraged, the platform can separately (or concurrently) target a Hybrid App deployment model as well, which is detailed below.

# Hybrid - The Best of Both Worlds

The hybrid app approach is the fastest and most efficient way to deliver "real," device-savvy mobile applications to users with frequency and low development cost and overhead. After several years of painful and costly missteps with pure native apps, the emergent hybrid approach has recently swelled in popularity and continues to gain momentum. The hybrid approach minimizes the amount of custom code required for each supported operating system, while still giving developers the ability to incorporate native features and functionality.

Additional options within the hybrid category allow organizations to cost effectively fulfill application requirements while optimizing in-house resources.



**WEB** 

APP BACKEND

SERVER

# Hybrid Web Applications

A hybrid web application combines the best attributes of native and mobile web apps. The vast majority of the app is built using mobile web standards – HTML5, CSS and JavaScript – that are either served from an application server or are distributed with the app. The difference for a mobile web app is that, instead of running in a generic browser, the app runs inside a thin native shell. This allows the developer to access device sensors and functionality from within the web application.

#### Benefits of developing a hybrid web app:

- Most of the app will be built using a single code base with web standards
- Developers are more likely to have some of the necessary web development skills
- The app can run on existing web infrastructures
- Most new versions can be released immediately, since the vast majority of changes will be on the web app
- The app can access device sensors and functionality
- ▶ The app can be distributed via the app stores

#### Disadvantages of developing a hybrid web app:

- Organizations must develop, maintain and deploy several code bases of the native shell to support multiple device and operating systems
- There are some limitations in terms of performance and user experience that make this approach impractical for some apps, like games.

### **Bottom line**

Build a hybrid web app when you need to access device capabilities or you want to distribute a branded app via the app stores. This is an ideal approach for apps that access, manipulate and display data as opposed to highly complex graphics or animations. An example of a hybrid web application is the Banana Republic app.



OutSystems did the heavy lifting of creating a multi-platform native shell, complete with Cordova integration that is maintained and extended as part of OutSystems Platform. This shell, called OutSystems Now, allows developers to build powerful hybrid mobile apps without having to understand any of the vagaries of the different individual native environments. It is open source, allowing customization to your specific needs (e.g. your logo, your native integrations, etc.) For the web part of the hybrid application, OutSystems Platform makes building integrations, business logic and user interfaces beautifully straightforward and fast. OutSystems also provides everything needed to leverage full use of device sensors from within the web app - no native code required.

# Hybrid Mixed Applications

A hybrid mixed app is similar to a hybrid web app but more code is written natively to take full advantage of device features and capabilities. Like hybrid web apps, hybrid mixed apps are primarily built with standard web technologies. Screens and logic are built using HTML5, CSS and JavaScript, which are either served from an application server or are distributed with the app. Hybrid mixed apps include an extra native component added to the native shell that enables highly interactive and device-intensive screens.

#### Benefits of developing a hybrid mixed app:

- Most of the app consists of a single code base for all devices, new and old
- Developers familiar with web app technologies can build that portion of the app
- Core app components can be created using existing web infrastructure
- New versions can be released quickly
- The app can use device features and capabilities
- The app can be distributed via app stores

#### Disadvantages of developing a hybrid mixed app:

- Organizations must develop, maintain and deploy multiple versions of the native shell to support multiple operating systems
- Building native shells is less work than building fully native apps, but this is still very significant overhead as new devices and versions of mobile operating systems proliferate
- Each new native component added needs to be maintained for multiple code bases

### Bottom line

Build a hybrid mixed app when you require one or two screens with complex user interactions that need to work on both old and new devices. An example of such a hybrid mixed app app is LinkedIn.



# How does OutSystems Platform handle Hybrid Mixed Apps?

We did the heavy lifting of creating a multi-platform native shell, complete with Cordova integration that is maintained and extended as part of OutSystems Platform. This shell, called OutSystems Now, allows developers to build powerful hybrid mobile apps leveraging access to device sensors and apps without having to understand any of the vagaries of the different individual native environments. For the web part of the hybrid application, OutSystems Platform makes building integrations, business logic and user interfaces beautifully straightforward and fast. OutSystems also provides everything needed to leverage full use of device sensors from within the web app - no native code required.

OutSystems Platform provides the fastest and simplest way to build an enterprise grade mBaaS (mobile backend as a service) for the native portion of your app. You can visually mash, cache, aggregate, orchestrate, integrate, add custom logic/

process, and expose all this via RESTful APIs - an easy way for native apps to connect with your servers and scale it with one click, with no impact on your corporate systems. A wide range of application services (user management, social connectors, background jobs, etc.) support both your native app and a unified web presence, along with all your app's back office needs.

# Packaged Hybrid Application

A packaged hybrid app consists of a native shell and a mobile web app very much like a hybrid web architecture. The difference is that the native shell is maintained by a third-party vendor, further reducing the need for in-house maintenance work on the native code bases.

#### Benefits of developing a packaged hybrid app:

- > Organizations only need to maintain one code base for all devices
- Developers familiar with web app technologies can build this type of app
- The app can run on existing web infrastructure
- New versions can be released quickly
- ▶ The app can use device features and capabilities
- The app can be distributed via the app stores

#### Disadvantages of developing a packaged hybrid app:

- Organizations cannot brand or customize the user interface
- Some performance and user experience limitations
- Access to the device is limited to what's supported by the vendor

### Bottom line

Use a packaged hybrid app when you need to access standard device's features and capabilities from the app and you don't want to maintain any native code base. An example of a vendor supplied app is Salesforce1 Platform.



# How does OutSystems Platform handle Packaged Hybrid Apps?

We did the heavy lifting of creating a multi-platform native shell, complete with Cordova integration that is maintained and extended as part of OutSystems Platform. This shell, called OutSystems Now, allows developers to build powerful hybrid mobile apps leveraging access to device sensors and apps without having to understand any of the vagaries of the different individual native environments.

OutSystems Now is the fastest way to get started with hybrid, without all the effort of building, testing, deploying and maintaining a native shell throughout the multiple native operating system updates, not to mention the painstaking approval processes of the different platforms' app stores.

Unlike pure packaged hybrid, it is open source, allowing customization to your specific needs (e.g. your logo, your native integrations, etc.) For the web part of the hybrid application, OutSystems Platform makes building integrations, business logic and user interfaces beautifully straightforward and fast. OutSystems also provides everything needed to leverage full use of device sensors from within the web app - no native code required.

# How to Choose a Mobile Application Architecture

You can see that many factors come into play when choosing a mobile application architecture. The number of options can further complicate the decision making process. That's why it's important to understand your mobile app's requirements. Specifically, consider the following:

### 1 Who will use the app?

Organizations typically build mobile apps for either their customers, partners or employees. Image and branding requirements are typically more demanding for customer-facing apps. Although the usability requirements are equally as strong for both user groups, you can usually keep the presentation simple for business-to-employee apps.

#### 2 Will the app consist of a single screen with a lot of animations or sounds?

The classic example of such an application is a game. Other examples include augmented reality applications or applications with heavily customized UIs and hardware-intensive animations.

#### 3 Do you need access to the device's sensors?

Some applications rely on data collected via device sensors. Capabilities like the camera or GPS are available to mobile web apps based on HTML5. Anything more elaborate however, like reading a barcode, requires native functionality.

### 4 Does the app require a few screens with a lot of animations or sounds?

Some apps consist of a combination of screens with standard functionality and screens with very complex user interactions such as heavy animations or sound. Standard screens, like a catalog of cars featuring technical specs, are a good candidate for a mobile web application. The one screen with the interactive 3D model of the car is a good candidate for native development.

### 5 Do you need access to custom native functionality?

If the functionality you need has been implemented by a third party, then you can cut development costs by using a proprietary native shell. However, if what you need is not available out of the box, you'll have to invest in developing a shell that supports your requirements.

### 6 Do you need to distribute your app via the app stores?

Users typically find apps via their device app store, but there are other ways to distribute an app. For example, users can access an app from a link sent via email or shared through social networks.

#### 7 Do you need to support devices with poor performance?

You can't count on every user having the latest and greatest mobile device, so you need to consider which models you will support. Older devices may not support animation or complex user interfaces running in the web browser. If these are important, you'll need to implement them natively. Alternatively, you can deliver different experiences for people with high-end versus low-end devices.

### 8 Do you need to brand your app?

Customer-facing apps generally need to represent your brand, but it's generally not a requirement for employee or partner apps. If branding is not an issue, you can choose a third-party hybrid shell. It will not feature your icon or logo, but it will save you the time and effort required to build and maintain your own shell.

#### 9 Do you need to support multiple operating systems?

The cost of native development quickly grows with each operating system you support. And it's not only about iOS, Android and Windows Phone. There are also differences between the various versions of each vendor's operating system. In other words, developing for iOS 7 is different from developing for iOS 8.

With the answers you just gave, you can navigate the following decision tree to determine the best mobile application architecture for your project.



## Decision Tree NATIVE VS WEB VS HYBRID







© Copyright OutSystems 2015. All rights reserved.

#### About OutSystems

OutSystems provides the enterprise Rapid Application Delivery (RAD) platform that makes it easy to develop apps once and deliver seamlessly across iOS, Android, Windows Phone, and Web - deeply integrated with existing cloud and on-premises databases and systems of record. OutSystems Platform enables the rapid delivery and effortless change of large application portfolios and is available as a public cloud, private cloud and on-premises solution. Over 450 enterprise organizations in 25 countries across 22 industries use the Platform to deliver beautiful mobile and web apps in record time. For more information visit us at www.outsystems.com or follow us on Twitter @OutSystems.