

## Building business applications for mobile devices



**Richard Lancaster**  
LANS Product Center

These days almost everyone has a mobile device, ranging from iPhone, Android, Blackberry, Symbian, or Windows phones to tablets such as Apple iPad, Motorola Xoom, or Samsung Galaxy. How do you design mobile applications and what tools do developers need? Will the existing methodology for application life cycle management be adequate for mobile applications? This Architects Corner is a primer on designing, developing and managing business applications for mobile devices and suggests issues to consider.

### Mobile Devices are Different

Corporate applications have been around for a long time. They typically run on servers that are mature, reliable, housed in a fixed location, fed with a constant power supply and have an almost unlimited upgrade capacity (memory, processors and storage).

Mobile devices have been on the market for a relatively short time. They have experienced rapid growth together with constant change in capacity and capability. Mobile devices are many and varied in

both manufacturer and operating system. Examples of manufacturers are: Apple, HTC, LG, Nokia, RIM, and Samsung. Operating systems include – Android (Google), Bada (Samsung), Blackberry OS (RIM), iOS (Apple), Symbian (Nokia), and Windows Phone (Microsoft).

The many combinations of devices and operating systems complicate mobile application development. One of the major obstacles is that development tools for native applications are incompatible across mobile

devices. Should you develop for all devices, or just a few, and, if just a few, which ones?

Some of the physical characteristics that make mobile devices different from laptops, desktops and servers are:

**The screen size** is smaller. Therefore you can show less information and you must provide a simple method for moving around the display area. **Keyboards** and on-screen touch keypads are small and unsuitable for sustained content creation and data manipulation. **Processing power** and the duration of operation are limited by battery capacity. Constant polling to a server will quickly drain a battery. **Memory capacity and local storage** are limited. **Communications** are slower and uncertain.

	Native Mobile Applications	Centralized Browser Application	Hybrid Browser Application
Cross-platform	Least potential for cross-platform portability. Development and testing is needed for each mobile device type and version.	Cross-platform operation, provided the application runs in a browser available on the mobile device.	Cross-platform operation, provided there is a container app available for the mobile devices you wish to use.
Data – external	Yes, when connected.	Yes, when connected, which is the only option.	Yes, when connected.
Data – local	Yes, local data access.	No local data.	Yes, local data access via the container.
Access to OS and hardware	Yes, can access OS and hardware such as camera and GPS	No direct interface, except for services provided by the browser.	No direct interface, except for services provided by the container.
Application Deployment	Requires download to and installation on the mobile device.	No deployment is required, except for a browser (already installed on the device).	Requires download to and installation on the device.
Development cost	Relatively high, especially when the app needs to run on multiple types of devices.	Relatively low, development occurs only once, with some additional work to ensure the app works with each browser.	Moderate, development occurs only once. The additional work is packaging the app into the container and testing it on each mobile device.
Development tools	Development tools are specific to the mobile device and operating system. Also, programming languages vary, e.g. Android uses Java and iOS uses Objective-C.	No development tools are required for the mobile device. Only Web development tools on the server.	No development tools are required for the mobile device. Developers build the application using Web development tools and use a container to package the app for each device type.
To develop an application for multiple types of devices you need to either: (a) Build a native application for each device; (b) Use a multi-platform application framework that generate a native application for each device; (c) Build the application to run in a browser. Building native applications that are portable across platforms can be a lot of work, due to multiple development tools/languages and few standard APIs.			
Not connected	Native applications can operate when the device is not connected to a network, especially when the data the application uses is stored on the device.	Browser applications only operate when the device is connected to a network.	Hybrid browser applications can operate when the device is connected to a network and also when disconnected.

## Mobile Applications are Different

The physical characteristics described above suggest that mobile applications need to be small, agile, and focused on discrete tasks to minimise the number of screens, local storage requirements and communications traffic.

Mobile applications can be browser-based, native to the device, or a hybrid of both.

**Native mobile applications** run directly on a mobile device's operating system, are fast and have access to all hardware features.

Native applications can operate connected and disconnected from a network. Several of the features that make mobile devices unique (e.g. location awareness) require a network connection. Building an application that uses these features will need thoughtful design when disconnected operation is a requirement.

Native applications that collect data to send to server-based applications and additionally operate in disconnected mode, must synchronize the data periodically with the server-based applications. This requirement adds to the complexity of the mobile application.

You should use a native application when either, (a) you need access to hardware features on the mobile device; or (b) the application needs to run in disconnected mode or use local storage; or (c) communication costs are high.

**Centralized browser applications** are typical Web or rich Internet applications. This style of application is server-centric, as

the database, business logic and Web serving is managed by one or more servers. The browser manages the user interface with HTML, JavaScript and CSS. These applications have no direct interface with the mobile device hardware or operating system, except for services provided by the browser.

Browser applications designed for larger screens may 'work' on a mobile device, but they become unwieldy when viewed on a small screen. In such cases you need to design the user interface for a small screen.

You should use a centralized browser application when you wish to offer the widest reach across devices with the least development effort.

**Hybrid browser applications** are browser applications packaged into a container (or framework) that is a native application on the mobile device. The browser application runs in the container using HTML, JavaScript and CSS like a typical browser application. In contrast to server-centric browser applications, hybrid browser applications can run when devices operate in disconnected mode, as the container provides access to local storage.

Hybrid browser applications do not require the server-side infrastructure of server-centric browser applications.

You should use a hybrid browser application when developers are more skilled with HTML, JavaScript and CSS, than with development tools available for native mobile applications.

## Design Considerations

When designing mobile applications, architects must consider the limitations of the device and the effort required from users to perform tasks with the device. Think about how to best use the device resources.

**Limited local data storage:** You should use the centralized browser model for mobile applications that need access to large amounts of data. When disconnected operation is required, minimize and optimize the information that's stored locally. Also, ensure that your application manages out-of-storage conditions.

**Power constraints:** Avoid power intensive features, such as excessive graphics activity and constant communication with servers. Design for minimal power usage.

**Reduced memory and processor speed:** Design mobile applications that are modular and focused on discrete tasks. Information should be returned to the user in the least possible time and progress updates should be provided while the user waits for completion of a task.

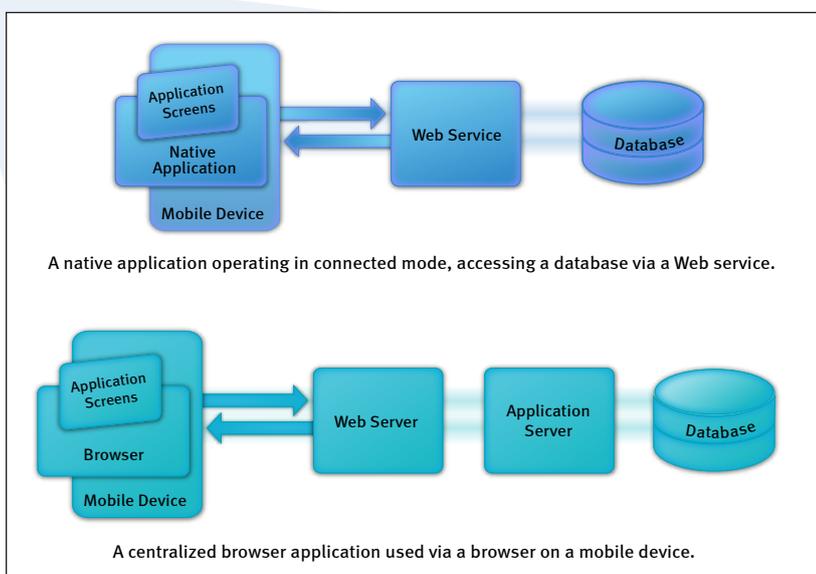
**The user experience:** Scrolling from screen to screen is confusing and typing information for long periods on small keyboards is cumbersome. Users expect immediate response from mobile applications, with information that is understandable in one view. Architects should design mobile applications that accomplish one, or at most a few, small discrete tasks. Single task focused applications will run quickly, consume less power and simplify the user interface.

Follow the conventions and display rules inherent in the operating system. Also, it's worth reading design guides published by the device manufacturer.

**Communications and networks:** Although wireless communications have increased in speed over the past few years, they offer lower data transfer rates and are less reliable than fixed networks. Wireless networks face variables, such as transmitters and receivers on the device and geographic coverage of mobile networks.

A mobile application that expects a synchronous-like connection to a server will fail when reception is intermittent or poor. Continuous polling to servers will drain the battery quickly and may incur excessive communications costs.

Design mobile applications to use



communications services in short bursts. Network activity should be assigned to a separate thread to avoid locking the user interface and inhibiting other applications.

### Your First Mobile App

Once you decide to build applications for mobile devices, how do you choose an appropriate first application? The choice of a mobile application ought to be a business requirement that uses features of the device to gather information and improve business processes. Choose an application that is not mission-critical, but will produce a useful business outcome and a realistic development experience.

Duplicating existing functionality is the easiest transition into mobile application development, but try to improve on it by taking advantage of the mobile device's features. For example, a restaurant chain can provide a list of locations on its Web site and customers can use their mobile device to view that list. From a business perspective, this functionality provides limited incentive to encourage customers to come to a restaurant. A mobile application with access to the GPS for location awareness may direct the customer to the nearest location, show what's on special and offer a discount. The extra features are more likely to drive customers to the restaurant chain.

The take-away message is to consider how a mobile application can respond to the needs of the user and provide improved service.

### Ongoing Management

In addition to issues associated with developing applications for mobile devices, you must plan for the ongoing management of both the mobile devices and applications. Deployment and ongoing management costs will be a significant proportion of the overall cost of mobile applications. Some of the contributing factors are: who owns the mobile device, security requirements, information protection, support, configuration and application deployment, and communication costs.

**Ownership:** The first policy to determine is whether the company supplies mobile devices to employees, or allows employees to bring their own mobile devices. When a company owns the mobile devices it can



control and lock down device features. With a bring-your-own policy, IT administrators lose control of when operating system upgrades occur and what other applications are installed on the device. You need to weigh up the pros and cons of each option as it applies to your context, especially for data security.

**Security** is a primary concern when mobile devices carry sensitive corporate information. Mobile devices are easily lost or stolen. So, a security plan must include facilities to remotely destroy or lock information. Two of the tools available are remote wipe and remote lock. Remote wipe is an appropriate action when the company owns the mobile device. However, since remote wipe will remove the employee's personal information as well, that may cause problems. The situation is even more complicated when employees purchase devices that do not have a remote wipe feature.

To avoid this problem, do not store corporate information on mobile devices. Instead use centralized browser applications. The trade-off is that these applications are unavailable when the mobile device is not connected to a network. The sensitivity of the information will determine how drastic a measure you need to take to secure corporate data.

**Support costs:** Mobile applications and mobile devices will increase the workload

and cost of IT support. Support costs are likely to be more manageable when IT administrators have control of and can limit the range of devices and operating systems.

**Deploying applications:** Especially in the case of native apps, configuration and application deployment require careful planning to avoid excessive cost and to ensure the current version of an application is in use. The complexity of the deployment task increases with the range of mobile devices and operating system versions a company has to support.

**Communications and network** costs are more predictable when mobile devices operate on unlimited data plans, or when using public or private Wi-Fi. However, telecommunications companies seem to be moving away from unlimited data plans and, as a consequence, communication costs may increase. The administrative overhead of examining monthly bills in order to separate company usage from private usage may be prohibitive. Companies may bear the full communication costs, or may determine a policy to reimburse a fixed amount or percentage of the monthly bill.

This article is an excerpt of a white paper that can be found at: [www.lansa.com/resources/whitepapers.htm](http://www.lansa.com/resources/whitepapers.htm)