

WHITEPAPER

# Enterprise Mobile Apps Are the Requirement. Do You Know the Solution?

Written by Michael Taylor, Rocket® Uniface Product Manager



# Table of contents

Introduction	3
What is an enterprise mobile application, and what makes it special?	4
What makes enterprise mobile application development especially complex?	6
Guidelines addressing enterprise mobile application challenges	9
Creating the right architectural structure	11
Front-end architecture	13
Rocket® Uniface and enterprise mobile applications	14
Conclusion	15

# Introduction

Enterprise mobile applications are currently one of the things keeping CIOs up at night. CIOs know that the right apps can engage customers and make employees more productive by enabling everyone to work and interact the way they prefer. However, they are all too aware of the risk of wasting time and effort by producing apps that never get used because they don't "feel" right to users. Or even worse, failing to deliver a mobile app that the business expects.

In a way, CIOs are right to be cautious. Experience has shown that enterprise mobile apps have requirements and implications over and above those of either consumer mobile apps or enterprise desktop apps. Yet companies can't afford to ignore the increasing expectations of mobility.

This paper aims to help readers understand the issues surrounding enterprise mobile applications and to suggest solutions so that they can reap the benefits and avoid the pitfalls. We outline our viewpoint on the challenges and make recommendations to help organizations that are either contemplating or are already involved with mobile development.

## 10 Ways Rocket® Uniface Helps You Succeed

- 1 Productivity
- 2 Reliability
- 3 Security
- 4 Integration & Reuse
- 5 Scalability
- 6 Agility
- 7 Sustainability
- 8 Technology Independence
- 9 Community
- 10 Partner Program



# What is an enterprise mobile application, and what makes it special?

We use the term “enterprise mobile application” to cover any application that runs on a mobile device and that interfaces with one or more enterprise systems. We’re therefore looking at applications such as online shopping and ticket booking applications for the general public, as well as at applications used by employees to access HR or sales functionality and information on the move—to name just a few examples.

In some respects, an enterprise mobile application is just a mobile application, but one with special requirements over and above those of a normal consumer app. It will help solve enterprise problems and will act as an extension to other enterprise applications rather than as a self-contained, standalone app in the way of consumer apps.

An enterprise app therefore has a greater requirement than a consumer app to be stable, available, and secure. Security is paramount because the app will probably make sensitive corporate data and functionality more easily accessible.

An enterprise app has other characteristics that distinguish it from most consumer apps. For example:

- The enterprise app should be more stable, as it typically has a longer life than a consumer app, which needs to change frequently if it is to retain user interest
- The enterprise app needs to work with large-scale enterprise databases, such as those associated with an inventory system
- The enterprise app does not have to carry out all the processing itself; it’s generally a small piece of front-end functionality that integrates with a more extensive system

## Characteristics of an Enterprise App vs. a Consumer App

- *Typically has a longer shelf life*
- *Needs to work with large-scale enterprise databases*
- *Generally a small piece of functionality integrating with a more extensive back end*



In terms of the user interface, the differences between enterprise and consumer apps might be less significant than they seem. Although an appealing “shiny” appearance may be less important for an enterprise mobile application than for a consumer app, enterprise apps still have to be highly usable. Like some consumer apps, an enterprise app may be heavily used, though perhaps not as heavily as an enterprise desktop application. And the enterprise app can be running on the user’s own device, just like a consumer app.

Inevitably, given these similarities, people will compare the enterprise app with the apps they use in their personal lives, such as Facebook and Twitter, and will expect the same high standards of usability, together with features like instant start-up.

## What are mobile moments?

// Mobile moments redefine every customer relationship. If a customer wants information or service in a mobile moment, that is your moment to shine. Be there and your customers will come to depend on you, deepening their loyalty and providing valuable information that your company can use to further improve the relationship.

To succeed in your customers’ mobile moments, you must understand their journey and identify their needs and context at each potential moment. Then design your mobile application to quickly provide just what’s needed in that moment. //

*Forrester Research*



# What makes enterprise mobile application development especially complex?

Enterprise mobile application development shares many of the complexities of enterprise desktop application development and has some additional ones of its own. Major challenges include:



**Security.** Although it may look like a desktop application in some respects, from a security standpoint a mobile app often behaves more like a website. You're putting your enterprise online for all to see, often including data that is confidential. Get this wrong and in the worst case you could greatly compromise company data.



**Dealing with fragmentation.** You need to deal with multiple mobile devices and operating systems, each of which may have its own constraints and vulnerabilities. The app also needs to remain stable when a mobile operating system is updated (something you can't control). The various devices have different market shares in different geographies; For example, China and the Netherlands have shown a preference for Android, whereas in Japan and the U.K. we've seen something closer to a 50-50 split between Android and iOS.



**Design constraints.** The mobile app has to be optimized for a small screen size as well as limited memory, processing capability, and sometimes battery life—but must still deal with large volumes of data and provide enterprise-strength functionality.



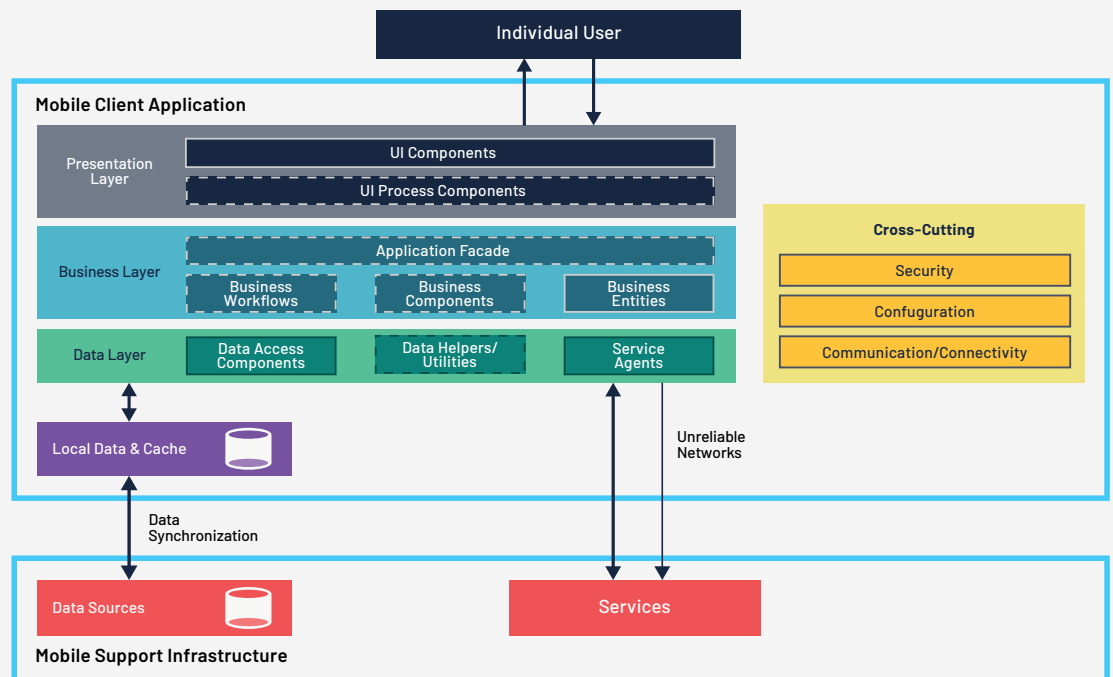
**Connectivity.** In some countries, network coverage is patchy, and in any case users may be offline some of the time (for example, when doing business in a remote area). Applications therefore need to be able to work offline as well as online. However, storing data on the device raises additional security concerns.

Other complexities are significant for some apps. These include:


- ✓ **Architectural complexity.** Mobile apps require an exceptionally deep stack of layers that have to be handled to get from device to data. These layers include the device, the network, the database, and the elements that process and present information. These can't be handled in isolation from one another: If one of these changes, then there is a risk of disintegration, or at least of security flaws resulting from working with inconsistent versions.

Figure 1

## Mobile App Architectural Complexity



- ✓ **Dependence on the wider environment.** There may be a trade-off between the level of integration with other enterprise systems and the stability of the new app. However you decide to balance these considerations, you can't create the app without taking into account the systems environment as a whole. This is one reason mobile apps typically require more effort than conventional apps.
- ✓ **Scalability requirements.** Your enterprise architecture was probably not designed with mobile in mind. For example, it may not be able to cope with the "chattiness" of a mobile app, or the back end may be too stateful to scale to accommodate large numbers of mobile users. A service layer can provide helpful abstraction (avoiding the need to duplicate functionality for mobile and other apps), but the device may constrain the amount of data passed or the speed of communication. The challenge is to keep the various elements as independent as possible while optimizing the overall picture for mobile.

- 
- ✔ **Volatility of requirements in live running.** Applications must be designed to be robust enough to deal with gradual change while still being able to run even if an element is not up-to-date. For example, when the app has been running offline for a time and then connects back to the server, there may need to be a translation so that the “old” app on the device can still maintain its data.
  - ✔ **Requirements that change during development.** Because most users can’t easily define what they need until they’ve seen it, there may be frequent change requests. What further complicates this is that enterprise applications often have to change frequently to keep pace with legal requirements, new business needs, and so on. With a mobile app, you have additional elements to update: the app and its back end. While it’s easy to control updates of desktops and web portals, that’s harder to do with mobile devices, particularly if they’re not owned by the business.
  - ✔ **Volatility of environment.** Devices and operating systems are continuing to change frequently, and a new release can break an app. Arguably, life is starting to get easier in this respect as we are seeing fewer new features in each release. Instead, manufacturers are changing the way they expect the devices to be used—for example, emphasizing larger devices such as “phablets” instead of phones, and combining them with wearables such as smart watches and headsets.
  - ✔ **Market immaturity.** There is a lack of tools and components for enterprise mobile application development, simply because it’s a relatively new market. You may have to start developing at a low level and then add levels of abstraction as more tools become available. Sometimes these restrictions can be overcome by providing features such as cryptography and LDAP on the server to get round the fact that they are not yet available on the mobile device.
  - ✔ **Uncertainty about look and feel.** Developers have to choose between designing their mobile app to conform with the corporate standard across all platforms (as Facebook generally has) or giving it the native look and feel of the platform it’s running on (whether that’s Android, Windows, or iOS). Either way, you have to make sure that your app will work reliably, and in a broadly similar way, on all devices.
  - ✔ **Part of a wider picture.** Mobile development doesn’t replace your other development challenges; instead you must deal with it in addition to client-server, web, mainframe, and other challenges.



# Guidelines addressing enterprise mobile application challenges

Some of the challenges can be overcome by choosing the right development environment. Later we'll describe some of the features that such an environment should offer. However, important as the environment is, it can't dictate how to build your applications. You therefore need to define how your organization will approach developing mobile apps. Here are some topics to consider:

- **Meet customer and employee expectations about mobile support.** It's no longer optional to support mobile devices—it's essential. For example, shipping companies collect customer signatures on mobile devices and update the website so both sender and recipient can track delivery; they have to do this to compete. Some expectations, however, can be met with a mobile website. It doesn't necessarily have to be an app, at least initially, and that is something to keep in mind.
- **Provide existing functionality more conveniently.** Mobile apps should allow people to interface with enterprise systems easily and at a moment's notice: for example, an airline travel app that allows a traveler using a mobile device to check in, change a flight, and have the latest flight status information at hand.

---

Although these apps are not replacing existing solutions, they will often be preferred (consider IBM's "Mobile First" marketing strategy) because they are dynamic and because many things are more conveniently done on the move than while sitting at a computer. To support users' expectations, it may be necessary to improve on an existing process. For example, a sales application should indicate to the salesperson whether a product is available without the need to communicate with the sales department—which may not be expedient in a desktop application.

- **Segment the functionality into concise, clearly focused pieces.** It's important to be realistic about what's achievable, breaking down larger applications into user-friendly portions. This requirement can best be approached by thinking about the information that will be required, and the tasks that need to be performed for a particular situation outside of the office.
- **Ensure fast and easy navigation.** Consider whether a single app should combine a range of functionality within it or whether it's better to have a series of separate apps. Either way, users need to be able to meet a need quickly, without navigating through lots of different screens. Navigation can be facilitated by profiling users so that each role, such as sales or marketing, has its own flow through the functionality.



- **Design the app for usability.** Whatever functions you provide should ideally be easier to use than on the desktop or web counterpart. For example, a sales app could be integrated with your contacts so you can quickly add contacts to the back end. The user interface should be simple, in line with consumer applications to which people are accustomed.
- **Make the most of the device's functionality.** Harvest contextual information from the phone to reduce the onus on the user by defaulting as much as possible. For example, a sales application could use location information to present a screen with a map of all the customers in the area, as well as directions. The contextual information should help you put the right screen in front of the user without the user's involvement.
- **Ensure agility.** It should be fast and easy to update the app whenever business needs change. Reacting in a quick way to business change, new user requirements, and expanding functionality gives the business a sense of trust in its enterprise mobile apps.
- **Optimize performance.** If an application is sending and receiving large volumes of information, then performance will be affected, with network and processing being the primary bottlenecks. By starting at the interface and working back, performance can be optimized.
- **Please your users.** The user experience will determine whether end customers buy or use the mobile application. Although their choices ultimately depend on the emotions elicited by the app, those emotions are strongly influenced by technical considerations such as usability, performance, and reliability.

### Top considerations for enterprise mobile app development

- *User experience*
- *Security*
- *Usability*
- *Deployment across multiple platforms and operating systems*
- *Reliability—particularly in certain environments, e.g., clinical*
- *Maintainability*
- *Availability—should be running all the time, even with no data connection*

# Creating the right architectural structure

In addition to the guidelines, it helps to define what the overall architecture of enterprise mobile applications should look like. While some of the decisions will be organization-specific, we believe certain architectural features should always be present.



## Data driven architecture

An enterprise mobile application's architecture should be data-driven. That is to say it should be about quickly obtaining specific data from the large amount that's available to help the user carry out a particular task—to deal with what analysts have defined as a "mobile moment."

Adopting a data-driven architecture also helps your apps to work across multiple devices. Once you separate data from presentation, it becomes easy to create new user interfaces for each device.

The app's architecture should be designed starting from the data structures that are already in place, so that it fits in easily with those.



## Dealing with multiple data sources

An enterprise mobile application often has to deal with data from multiple sources, which might include some or all of a website, an existing client-server application, various legacy systems, and third-party data such as that provided through Google. Usually some of these sources are internal while others are in the cloud. Whatever data the app requires is likely to be used in situ wherever it is already stored; it doesn't usually make sense to duplicate it for the mobile app.

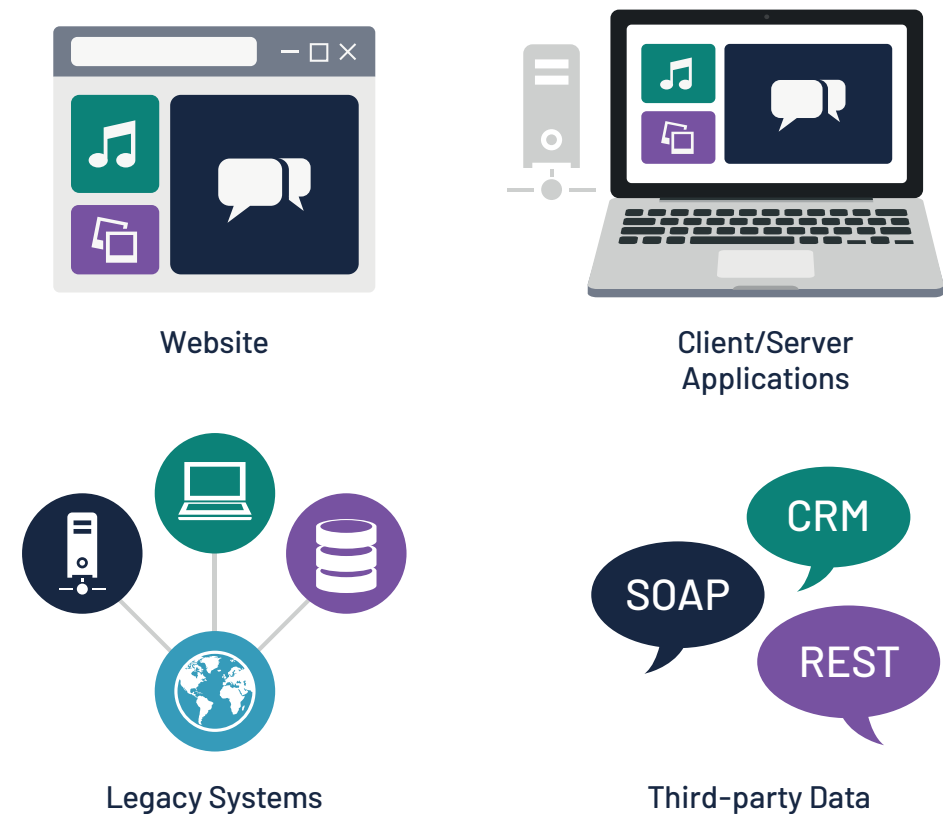
In addition, information from the mobile device (for example, locations) can be used, although it's important to make users aware that this is going to happen and to get their consent when appropriate.

While external data from third parties can be best integrated on the mobile client, integration of internal data will usually take place on a back-end server (which could be in the cloud). This back-end element can also carry out most or all security checks, acting as a boundary between the enterprise and the outside world. Service-oriented architecture can provide a mechanism for communication between the mobile device and the back-end server.

Enterprise apps have to exist within a remarkably diverse architectural environment; the various data sources can have vastly different internal architecture. It is important to superimpose a common data model to bring together and coordinate the app and the back end, and to then enforce this model. A product such as those offered by Rocket® Software can act as a hub to ensure this happens. With this arrangement, the mobile app functions in a way very similar way to that of a regular enterprise app that communicates via APIs with a back end.

Figure 2

## Multiple data sources for enterprise mobile apps







## Front-end architecture

Although a lot of the security can be handled at the interface to the back end, the app itself needs to provide a secure, robust interface that is not easily abused.

It also has to be adaptable enough to run in a range of conditions; for example, if the network is slow or unavailable, the app should be able to continue working.

The architecture of the front end needs to take into account the characteristics of mobile devices. Displays need to be optimized for small screens, and processing needs to be able to run with a limited CPU or limited memory. This may require compromise. In addition, a range of devices need to be considered, not just one specific device.

# Rocket® Uniface and enterprise mobile applications

The right development environment is crucial to success in enterprise mobile application development. Rocket® Uniface enables organizations to overcome the challenges we've outlined here. Below we address some of the areas that with which our clients say they need the most support at the moment, as well as how we can help.

▶ **Responsive.** We can insulate you from the differences of screen size among devices—for example between a 4-inch phone and a 10-inch tablet—which would otherwise take up programming time even if both are running the same operating system. Write your app with Rocket and it will automatically adapt to any mobile platform. You can create screens made out of different sub-screens and select which ones will be passed through to a given platform based on screen size. (Without Rocket® Uniface you'd need a whole array of tools to do that.)

▶ **Uniformity across front and back ends and a way to integrate them.** Rocket® Uniface gives you a single mode for writing the front and back ends and for interfacing between them and with other services—all in one place and around a consistent model. It makes it easy to be data-driven—to focus on the data you need and on how your app should work with it.

▶ **Security.** Rocket products have security engineered into them. In addition, Rocket® Uniface provides a wide range of security features your developers can use; we can support them in following best practices. Moreover, because we handle all the elements in the stack, from mobile front end to database, we can make sure processing is handled in the way that gives you the best protection from hackers.

▶ **Reliability and availability.** Developers have been using products from Rocket to create enterprise applications for many years. Our architecture has been enhanced and optimized over time so that it scales to support large numbers of concurrent users, maintaining maximum reliability and availability.

▶ **Maintainability.** Our model-driven architecture makes it easy to put through changes to an app, whether you're adding new functionality, updating validation in line with new legislation, or simply improving the look of it to keep users interested. Adjust the model and everything else is automatically updated to match; users don't have to do anything, and they aren't disrupted.

▶ **Flexible development strategy.** You might want to start by developing a mobile-enabled website—getting a feel for how useful mobile is going to be to your specific organization at a relatively low cost—and then create an app later. This is easy with Rocket® Uniface because the same functionality can be used in conventional applications, on websites, or in mobile apps. You can make the choice between website and app late in the development cycle, if necessary.

## Rocket® Uniface's Mobile Solution

- *Responsive*
- *Uniformity across front and back ends*
- *Security*
- *Reliability and availability*
- *Maintainability*
- *Flexible development strategy*

## Conclusion

Clearly, enterprise mobility has specific requirements and can't be approached in the same way as one would approach consumer mobile apps. You can certainly incorporate some lessons from consumer apps, and also from conventional enterprise applications, but there are additional considerations that are different from either. IT needs to get into the mindset of "thinking mobile," compromising where necessary to fit in with the limitations of the platform and to ensure that the result has the "mobile feel" from which everyone will benefit.



# About Rocket® Uniface

Rocket® Uniface, the most productive, reliable development tool in the industry, provides a model-driven environment for the rapid development of scalable-enterprise mission-critical applications. Learn more at [www.rocketsoftware.com](http://www.rocketsoftware.com).



© Rocket Software, Inc. or its affiliates 1990–2021. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.

Rocket\_Uniface\_Whitepaper\_v5