

WHITE PAPER

Key Criteria in Selecting Wireless Application Development Platforms

Sponsored by: RIM

Stephen D. Drake

Steve McClure

May 2007

INTRODUCTION

Understanding the Benefits of Wireless Applications

A few years ago, a panelist at a forward-looking technology conference, hosted by the Austrian government in Vienna, summarized his assessment of information technology (IT) trends with one phrase, "the future of the computer industry is not on the desktop." He quickly went on to explain that the progression of microelectronics to enable smaller and more powerful devices, combined with the Internet and other advances in networking, will allow substantial portions of applications to be run on handheld devices (e.g., phones, PDAs), appliances and other nontraditional platforms. If such devices are easily portable or if users can use distributed devices to access remote services, then IT enters the realm of wireless applications.

Before we can understand wireless applications, a brief explanation of mobility and wireless is critical to this discussion. Although mobile and wireless are often used in similar manners, the two terms are indeed different. Mobile represents the macro world and the large universe that everything wireless or not sit in. As an example, a worker could be using a laptop in a hotel without a wireless connection or any connection and would be considered mobile. Wireless, for example is a technology that allows such functions as communication to back-end systems and persistent synchronization. Such wireless efficiency leverages the power of the back-end systems and affords better battery life and a smaller footprint on the end device. The requirements to build an application that speaks to a wireless environment are many and specific. This white paper discusses some of the key criteria and best practices in seeking wireless application tools.

What Is Unique About Wireless Applications?

There are a number of attributes of mobile applications that place requirements on application developers beyond traditional client/server or standalone applications we run daily on our desktop computers. These attributes can be discussed individually and in combinations. All have something to do with exchanging information and services across a network between mobile devices and remote servers. Hence, connectivity and mobility account for most of the unique requirements. Some of the key aspects can be separated into three core areas: network, device, and security/standards:

Network

- ☒ The range of network bandwidth is varied and, in conjunction with latency, can seriously affect the user's overall experience with the application.
- ☒ The reliability of the network can compromise message delivery, which has important consequences for guaranteeing completion of database transactions.
- ☒ The user may not always be connected to the network, either by choice or because the network is not available. Some portion of the application may need to run locally on the mobile device, caching transactions for completion later. There are a number of wireless network challenges that must be overcome. Leveraging an always-on/always-connected environment delivers a more transparent experience than the continuing need to dial into a VPN. In addition, how such VPN-based solutions impact battery life and the nature of dynamic IP addresses across multiple networks pose multiple challenges to users and IT groups alike.

Device

- ☒ The user interface must be designed to take into account a large and growing range of potential devices all of which typically have more resource limitations than a desktop computer. Displays are smaller, processors slower, memory and storage limited, and battery power must be conserved with careful management. And, despite the advances of mobile devices, it is critical to point out that a huge gap exists between computing and mobile computing, and developers must take this into account regardless of evolution of the devices.

Security/Standards

- ☒ Security is a critical, overarching area of concern with corporate data being extended to mobile devices outside of the corporate firewall. In fact, these device-based security concerns are the largest concern for IT shops. Mobile devices are easily lost or stolen, the size of the device encourages sharing tendencies and the need for IT organizations to control, manage, and secure these devices like any other corporate asset is a daunting task.
- ☒ A broader array of "standards" comes into play and must be understood and incorporated into the application design.

SITUATION OVERVIEW

Challenges of Developing and Deploying Wireless Applications

Summary of Key Differences with Other Types of Applications

In order to obtain the business benefits mentioned earlier, an organization must develop and deploy the associated applications. Once deployed, the applications must then be maintained, enhanced, and their daily use managed. Before we go into a detailed discussion of the challenges presented by wireless applications, we will summarize what is unique about each.

Development

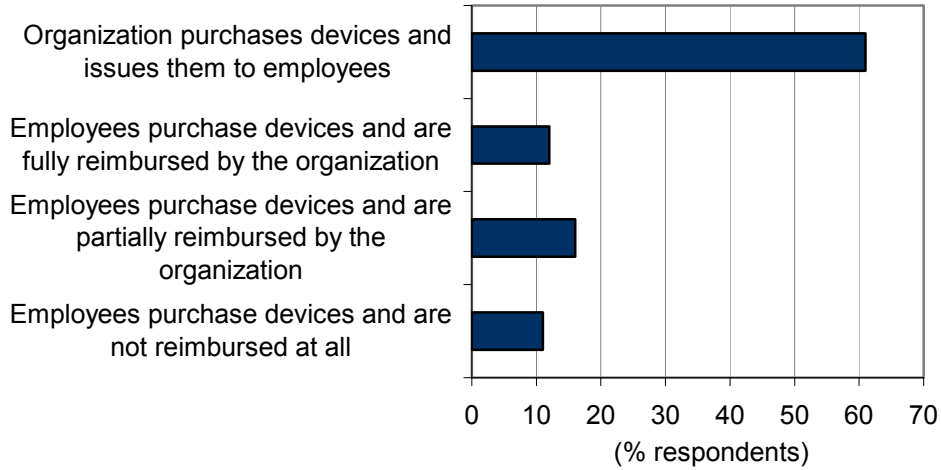
Requirements definition and application architecture design will require more time and resources because the application functionality must be partitioned and the user interface may need to run on a variety of heterogeneous devices, including the need to make provisions to anticipate future devices. However, demands from corporate IT to maintain more control over approved devices in the enterprise yield increasing control over which device types can and will be deployed and under which development environments.

Wireless applications may have to deal with multiple modes of input and output (bar code readers, touch screens, keyboard, character recognition, and voice recognition to name a few), as well as multiple operating systems or multiple nuances within the same operating system. Each device will have its own constraints, and once the application is coded, it must be tested on those devices either through simulation or actually porting the application to the device. Since the network or the user may not always be available, provisions will probably be required for information caching and asynchronous operation and subsequent replication and synchronization with server-side databases. Performance, especially as it affects the user, is always an issue and must be tested as well. As with ecommerce applications, wireless applications are often used without any user training. The user interface must be intuitive or self-explanatory, and as simple as possible. Despite many of these issues, IT organizations are beginning to take a much stronger role in the control of the mobile devices. Mobile devices today being deployed by organizations are increasingly being considered a corporate asset rather than a personal tool. Figure 1 demonstrates that while variation may exist in different geographies, in North America, we see a dominance of corporate-controlled deployments. As is evident from this report, the more controlled a deployment is, the easier it is to manage the application development requirements.

FIGURE 1

Employee Device Acquisition

Q. Which of the following statements best characterizes how MOST EMPLOYEES at your organization acquire mobile devices (i.e., mobile phones, PDAs, smartphones, BlackBerrys, etc.) that are used for work-related purposes?



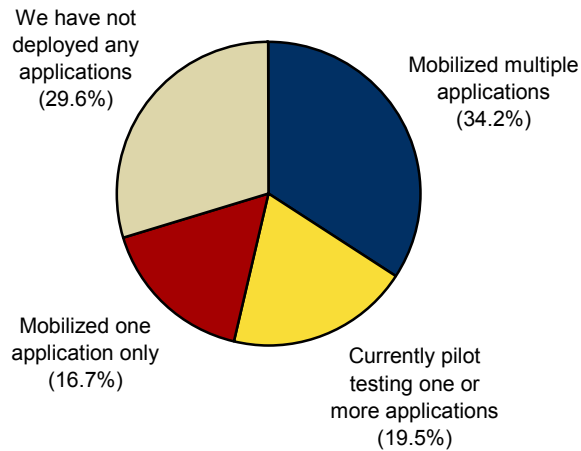
Source: IDC's *Mobilizing the Enterprise Survey*, 2006

Decisions affecting all of these issues are often solved as IT takes a stronger hand in the deployment of applications and solutions on devices that they approve. In fact, recent IDC data suggests that a substantial majority of respondents have already rolled out an application at some level. Figure 2 shows that at different stages, 70% of organizations have rolled out a wireless application.

FIGURE 2

Enterprise Stage of Wireless Application Deployment

Q. Which of the following, would you say, best describes your organization's current stage of mobility?



Source: IDC's *Mobilizing the Enterprise Survey*, 2006

Deployment and Ongoing Maintenance and Enhancement

Deploying the application requires software builds for multiple platforms and software quality assurance (QA) for each platform and for any combination of these platforms that will be interoperating. Various software must then be distributed and installed, more or less simultaneously, to all the servers and devices.

Wireless applications must be maintained and enhanced like any other application. The difference is that the portion of the application that resides on the device is best accessed by central IT management through mobile management controls. Updates usually occur over the network. Not all suppliers provide such an integrated functionality, some of these mobile components require a third-party software add-on. It is critical that organizations ask their suppliers about providing such a component as part of its mobile deployment. The good news is that this can be accomplished quickly once the user logs in. Sophisticated deployment solutions recognize user settings and can be independent of the device, easing new deployments or redeployments of applications and settings if the device is lost, stolen, or upgraded.

Management

Partly because of their inherent portability, mobile devices get damaged, lost, or stolen more easily. This may cause a security breach. There may be a loss of transaction information or other data that had been cached for later update. The number of users may scale up as the size of the mobile workforce expands. This means that monitoring the performance of the application day to day is critical. For many years, mobile-specific, device-level management has been available to address many of these issues and increasingly incorporate mobile security components. In addition to specific mobile device management solutions, management and security components within mobile middleware and other mobile software server solutions play important roles in supporting the management of wireless applications.

Furthermore, additional value in mobile device management can be realized through software measurement and optimization. Once a mobile solution has been rolled out, organizations can optimize the way the devices, applications, and updates are deployed. The ability to monitor not only the key components of a device and its software, but the users work process is critical. It allows IT organizations to get valuable feedback from these users relating to their working pattern, when applications need to be updated or enhanced, discover critical network, device, and other software patterns that assist in the overall mobile administration process.

What Wireless Application Developers Are Telling IDC

Challenges

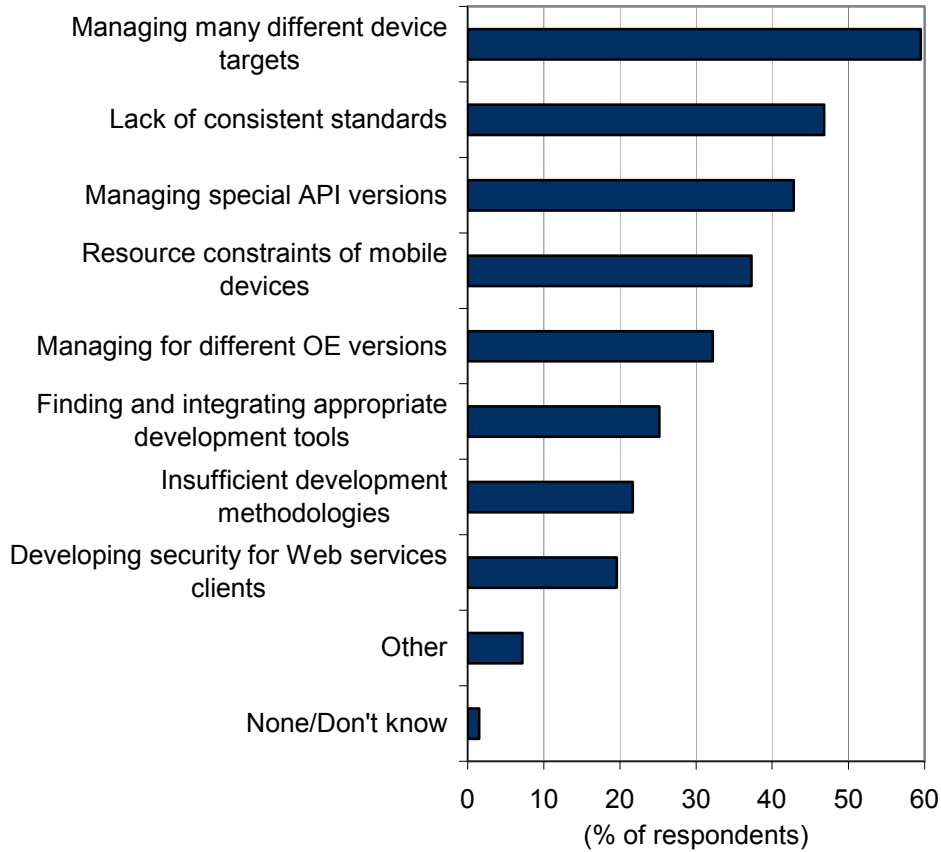
Respondents were asked to identify the top 3 challenges of wireless application development. The results are shown in Figure 3. At the top of the list is managing many different device targets, noted as a challenge by almost 60% of the respondents. That was followed in order by lack of consistent standards, managing special API versions, and resource constraints of mobile devices. Respondents downplayed security issues. Only one in five thought this was in the top 3 challenges. About a quarter noted finding and integrating appropriate development tools as a challenge, along with insufficient development methodologies.

What is important to note, however, is that typical to many organizations, the top challenges registering in this survey may actually not be challenges at all once a company has made a particular platform choice to support. That is, once a company has begun to treat the mobile device as a corporate asset and has made a move to standardize on a particular platform, areas such as device, application, and OS management, as well as dealing with standards and API versioning become less of an IT headache. In fact, the strongest solutions offering the development of applications on a particular platform typically include the management and administration of much of these issues as part of an overall integrated offering.

FIGURE 3

Wireless Application Development Challenges

Q. *What do you find most challenging in connection with developing and deploying mobile applications? Choose up to three.*



Source: IDC's 1Q06 Software Developers' Collaborative Study, 2006

What to Look for in a Comprehensive Platform for Developing, Deploying, and Managing Wireless Applications

Key Requirements for a Wireless Application Development and Deployment Platform

With the above as an introduction, we can look at the four key requirements for a comprehensive platform for developing, deploying, and managing wireless applications. Such a platform should not stand in isolation, but must be complemented with a rigorous set of development processes and best practices across the development life cycle and be part of an integrated package.

The application capabilities listed obviously constitute a super set of what any individual project or application may require based on its particular unique environment:

- Development tools
- Manageability
- Transport layer
- Security and standards

Development Tools

Ease of Development

A good development platform should strive to facilitate the myriad of tasks that must be performed to complete the development project. This is often accomplished with a tool framework, such as Eclipse that allows for the integration of various tools in the form of plug-ins, as well as a common integrated development environment (IDE) with a graphical user interface. The platform should do more than provide role-based editors and tools for architects, designers, developers (i.e., coders), testers, and the like, but it should also provide a high level of integration within the tools. One tool should anticipate the needs of the next across the life cycle, where possible using industry-standard formats. It is best to have a common repository for the artifacts created by each successive step in the development process. Metadata management should not be an afterthought.

Ease of use is also obtained by introducing extra levels of abstraction at various points in each process or architectural layer. This allows development to be more declarative and relieves the developer of the need to deal with every little nuance associated with different deployment choices, be it device or OS type.

Choice of Client

Not only should the platform support both smart and thin client user interfaces, but it is imperative that the platform understand exactly what the limits are of the device resources in terms of supporting a thin client. Building applications leveraging a robust local store on the device versus those leveraging a browser-based solution has specific requirements as to what type of data is to be deployed and how it is delivered between device and back-end. Comprehensive platforms will allow support for multiple client types.

The client choice brings up the trade-off IT is faced with regarding how to treat mobile devices. Are the devices under corporate liability or personal devices? And, if there is a mix, how does IT address administration and security policies and the costs to support those devices? Is it necessary for IT to give up some control for potential enhanced usage patterns and improved productivity?

Permitting a Limited Set of Preconfigured Architectures

An organization always has a variety of architectures to choose from when implementing new software applications. An organization should have the option of using the development platform with a set of architectures that have been evaluated and chosen by it to be used by its developers as the organization's standard architectures. The various assets, artifacts, patterns, libraries, etc. for each architecture can be packaged with the platform in a way that the various architectural alternatives are fully supported and made available to the developer to select whichever is appropriate to the task at hand. In this way, there will be a more consistent approach to software design and a more manageable approach to software asset life-cycle management.

Compatibility with Existing Development Resources/Practices

Very few organizations have the luxury of ignoring its existing development infrastructure. The typical organization is dealing with what IDC has labeled the software complexity crisis. This is the unfortunate proliferation of IT technologies that has been accreted over time through the adoption of successive generations of technology or, in some cases, because of mergers and acquisitions. Most organizations are not going to abandon these investments just because they are starting a wireless application development project. Ideally, any additional investment that might be required should be compatible with its existing investments. This extends to its development skill mix and to its development policies, processes, and practices. One way to do this is to support widely adopted development frameworks like Eclipse or NetBeans. Another is to support commonly used languages like Java and JavaScript, C#, and JScript.

More often than not, these issues are mitigated by the right choice of platform, especially if that platform abstracts the complexities and provides utilities to minimize any skills gaps (e.g., by translating .NET artifacts to Java). C#, C++, and Java are all object-oriented languages, so most of the experience gained from using C# or C++ can be employed with minimal effort when programming in Java. It also helps that a development in recent years has been moving to a thinner client/server architecture, which places more emphasis on a browser-based user interface, especially for wireless applications.

Delaying Implementation-Specific Choices

A good development and deployment platform should defer implementation-specific choices for later in the development cycle whenever possible. This allows the application designers to concentrate on meeting the business and architectural requirements as a first priority, while not getting bogged down in a myriad of detail prematurely. In essence this is a form of virtualization. The designer can specify what the application must be able to do, and postpone exactly how that will be implemented. This trade-off is only possible if there are appropriate code generators to convert the more abstract design specification into the binary code that must run on each server, database, or device. The same can be done with content if the environment has multiple rendering engines that tailor the base content to each device automatically.

This also helps to future-proof wireless applications, making them less vulnerable to changes in devices or network protocols. The underlying design of the application is compromised because a developer hard-coded functions to one specific implementation.

In the end, wireless applications are a mission-critical business tool and provide an opportunity for IT to leverage existing resources on back-end systems. Such projects go well beyond a technical function, but their demonstrated ROI can help position IT as a valuable contributor to the organization. With this growing trend, it is important for IT organizations to choose platforms with strong business cases, solid partners, demonstrated deployments, and measurable returns.

Globalization and Localization

The network may or may not be private, but if the application being developed is for a multinational global enterprise, then globalization will become an issue sooner or later. Since any changes are always more costly to make after an application is released to production, globalization requirements should be part of the overall development process. Much of this support will likely stem from the operating environments of the servers and devices, so this is not a development platform issue per se. Issues like this surface in the user interface, the database, and content management systems.

Manageability

Ease of Deployment, Including Incremental Upgrading

The platform should be comprehensive enough to support all deployment environments required by the application. This includes servers and devices. With a knowledge of which servers and devices the wireless application will require, the developer should be able to easily identify them, partition the software assets appropriately, and have the build and packaging of the software for distribution and installation be as automated as possible. Once deployed, the process of releasing incremental updates should also be facilitated incrementally and automatically.

Accommodation of Current and Future Device Types

The platform should recognize and support critical form factors and functionality critical to the organization and end-user requirements. That support should be extensible to new devices as allowed into the enterprise. This support extends beyond understanding the capabilities of the device operating environment, but rather looks at functionality such as Bluetooth, GPS, WiFi, smart card, barcode, ruggedized, camera, etc. Other device considerations include performance, storage capacity, startup time, display resolution, other input/output feature configurations. The platform should be able to keep configuration profiles for each device used and apply that metadata to the automation of the various development and deployment tasks, especially software QA.

Workflow and Assignment

The platform should have built-in support for managing a workflow between various participants in a business process, including interruptible session management and the concept of long transactions in which customer, partners and employees are engaged in a multi-step, multi-day succession of tasks, some of which may require approvals.

Integration

The platform must provide some form of integration with the various other applications and databases in the organization's application portfolio. Integration with back-end systems is a very different environment in mobile than with desktop. It is vital that most tasks performed on the mobile device are eventually sent back to the multiple back-end systems in a useable manner. How the data is optimized for delivery to the user in a disconnected environment and synchronizing that data from a mobile device to a back-end system are key components in an integrated mobile offering.

Life-Cycle Support

From requirements gathering to post-deployment optimization, various tools and editors should recognize the unique aspects of each type of developer and be specific as possible in its support of that role. In addition the platform should have common life-cycle tools like version control and change management. Test should not be an afterthought, but a first-class participant with support for a broad array of testing, including functional testing, performance and load testing, GUI testing, and even security testing. Ideally, all these tools will be managed through a common repository.

Operations Management, Tracking, and Reporting

If not provided by the development platform, then consideration should be given to some level of interactivity between the development environment and the operations environment. This will facilitate tracking and reporting of day-to-day experience with the use of the application once it is in production. This is especially useful when developers are informed of bugs or operational issues and have some additional information about the context of the problems identified. On the flip side, the operations management personnel need to know which changes were made to address previously reported problems. Deploying a number of features, such as group rules, IT policy controls, OTA updates and multidevice tracking, can provide such an opportunity to address these needs.

Transport Layer

Transactions, Content, and Messages

The development platform needs to provide some standard services for synchronization and replication of messages and data. This level of support should be easily incorporated into the application, hopefully with minimal effort on the part of the developer, who should be able to assume the service is automatically incorporated. The platform should guarantee that data will be kept consistent between the device and the database on the server. If a transaction fails, the platform should manage the necessary recovery.

The platform should have built-in support for publish/subscribe content delivery, should be declarative as possible, and not require the developer to knit together a lot of extra code in order to push content to the application user using mobile data service channels.

Infrequent Link or Immediacy

The platform should accommodate both infrequently connected devices and those that are always on, always connected, or never connected. Beyond the device structure, platforms should also accommodate the reality of instable/intermittent mobile networks. It should know which case corresponds to each device being used with the wireless application and respond appropriately, both at the server and within the device. A persistent connection is critical for a compelling mobile experience to the end user, whereas multiple requirements to connect via VPN and multiple password lookups is a drain on the network and the battery and creates a disruptive and inefficient work environment.

The development platform should recognize and take advantage of persistent storage capabilities of each device in use and employ strategies to avoid a security risk from a lost, stolen, or misappropriated data.

Security and Standards

Security

The platform should provide several types of security features, including but not limited to role-based access control with authentication and validation, encryption/decryption, in the context of industry standards. These features need to be compatible with the various devices and their user interfaces. It should provide or at least work effectively with existing firewalls, tiers, ports, database authentication and encryption and communications encryption. Regardless of the platform leveraged, security is of the utmost importance and, depending on the platform environment, may be all-encompassing within a more complete end-to-end infrastructure or may require the adoption of third-party products to incorporate within the platform to bring it to a level of security that is accepted by the adopting organization.

Today, the largest emphasis and what is top of mind to IT organizations is device-based security. Specifically, password protection, device wipe/device lockdown, and encryption. Since mobile devices by nature are away from a primary work environment and because of their size and sometimes being considered as more of a personal tool than a corporate asset, the ability to have some level of control on the device is of utmost importance. Devices can be lost or stolen with an increasing amount of sensitive data residing locally. Users may leave a company or be off of the corporate network for days, weeks, or more. The combined need for such functionality and policies in place to execute these critical functions is driving mobile security adoption in the enterprise.

Lastly, it is critical to reiterate the significance of securing the network. Organizations must not only feel secure that their devices are being protected, but that the information traveling over the transport layer is also secure. Understanding your suppliers' ability to provide a secure transport layer for the delivery of your mobile data down to your device is a must for all organizations.

Standards Support

Another way for an organization to extend its development skill mix and its development policies and practices is to support widely adopted industry or de facto standards.

The range of standards that come into consideration is broader for wireless applications development. Certainly, those on the server side are more familiar than the plethora of networking and device-specific standards. These standards are not only still evolving, but vary geographically. The more the development platform supports, the better.

CONCLUSION

Although email sometimes dominates much of the landscape in the wireless space, the reality is that wireless applications beyond email are here today. Much of the work done around education and understanding of wireless deployments and wireless technology (largely starting with email) over the last 5–10 years has paved the way for wireless applications to be adopted in a much more rapid fashion. For many organizations, much of the technology acquisition and corporate buy-in has already taken place and these companies are already beginning to build and deploy wireless applications.

The improvements in wireless application software development and the increased movement toward the adoption of standards-based technology afford an advantageous environment today to roll out wireless applications. Although device and networks continue to push forward and improve, any limitations in these areas are easily advanced with efficient wireless applications. Such advances in the development of wireless applications should overcome any hesitancy of an organization seeking to deploy a wireless application. Real solutions are being deployed today by deploying wireless applications, and it is critical for organizations to take advantage of such opportunities to deliver improved worker productivity, cost efficiencies, and a strategic advantage.

Copyright Notice

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2007 IDC. Reproduction without written permission is completely forbidden.