mobileOne

Helping the world goMobile™ ™

JUNE 2005

Mobilize your business: How technology makes that work

MobileOne Infocom Private Limited
Marketing Department
1102, Paigah Plaza, Basheerbagh,
Hyderabad 500029, AP
India

Table of Contents

Introduction

The rising demand for mobile solutions has created a unique situation between Information Technology (IT) departments and the companies they support. While IT has historically been the one pushing for innovation, mobile and wireless applications have turned the tables: now executives, sales staffs, and many other employees want mobility as soon as possible. But IT is putting on the brakes, asking "Do mobile applications make sense?" and "How can we deploy them securely and reliably?"

Given the massive stakes, IT should be careful. The burdens placed on IT—to optimize processes and create new ones, to integrate existing applications and link disparate systems, and to otherwise unlock new value from SFA, CRM, ERP and other systems without major new expenditures or massive retooling—have managers pondering their next step. Their best chance of keeping up is to start over with a new architectural level. Using web services to combine functionality from several applications paves the way for a service-oriented architecture (SOA) that enables a flexible structure of applications with the ability to change quickly at the lowest cost. The danger for IT is to view mobile solutions as a sideshow.

But when the architectural elements of mobility are examined, it's apparent that mobility and SOA require nearly identical underpinnings and features. Mobile applications require secure wireless connectivity, disconnected operation, centralized and synchronized data, configuration and management of applications, and security. The fundamental argument of this paper is that applications based on SOA that cross company boundaries possess many characteristics of mobile applications and that IT should address SOA and mobility in a harmonized roadmap. Gaining experience with mobile solutions will bring immediate benefits to users and later reduce risk during the ultimate implementation of SOA. A key implication of this argument is that a mobile architecture is actually more robust, reliable, scalable, and secure than the current one and will become the standard paradigm in five to ten years.

MobileOne understands this and have created technology for IT to unlock the full value of a mobile end-to-end architecture. The challenge for IT is to embrace mobility as an architectural issue rather than an add-on to individual applications.

In explaining this proposition, this white paper takes a comprehensive tour of mobile solutions, from the enterprise application to the chipsets running on mobile devices. The topics covered include:

•The architecture of mobile solutions. How should they be built and deployed?
•The technology. What are the essential components?
•Case studies demonstrating successful mobile solutions
•A roadmap for the design and deployment of mobile solutions.

The expanding promise of mobile solutions

The state of mobile applications today is analogous to the first corporate web pages a decade ago: curiosities with some value but isolated and lacking in real functionality. The web pages evolved to Internet-aware applications and now to web services, which 10 years later may have the most profound effect of any of these changes. Mobile solutions are following a similar trajectory.

The benefits are already tangible, whether it's the CEO receiving need-it-right-this-second data on his smart phone or the roaming sales force clinching deals in the field rather than at their desks. Users are already educated about and hungry for the productivity and efficiency improvements mobility provides. Decisionmaking is speeded by the delivery of data and applications as needed. Wireless networks grant the opportunity to synchronize disconnected devices. Sales can be concluded on the spot by a salesperson equipped with a device that records the transaction and passes along data to begin its fulfillment. And once mobile solutions are integrated into ongoing business processes, previously unrecognized distractions and delays are gradually eliminated as the solution is continually refined and improved. Companies that understand how to build and deploy mobile solutions—and how to do so to maximize process improvement—will gain a significant edge over their stilltethered competitors.

Consider these examples:

At the terminals of the Frankfurt, Germany Airport, an SAP system tracks the inspection of the 22,000 fire shutters that close to ensure smoke and hot gasses are not transmitted through the air-ventilation system in the event of a fire. RFID tags identify each shutter and record the last date of inspection. PDAs read the tags and record inspection notes about problems. Paperwork is gone for good and technicians spend more time improving safety. Eventually maintenance requests will be processed in real time, meaning shutters will be fixed in minutes or hours instead of days, which will ultimately save lives.

An insurance company increased its revenue by 30 percent after introducing a mobile solution to replace the paperwork formerly needed to close a sale. This allows deals to be closed immediately. The company used to send customers home with a paper application, and many never came back. The time between the customer's signature and the final underwriting of the policy was slashed by 50 percent. Similar reduction of the time between initial contact and closing the sale can have an impact on a wide variety of businesses.

At one U.S. Air Force base, the time spent on routine maintenance of an aircraft shrank from six months to just four-and-a-half after technicians were equipped with wireless notebooks. Using the notebooks allows them to search for parts and instructions while still in the bellies of planes instead of back at their desks. This mobile solution made aircraft maintenance 25 percent more efficient. A mobilized workforce is one that's in the field, in front of customers and suppliers, solving problems and closing sales at the point of contact. Mobile solutions eliminate barriers between the user and the task at hand and amplify the value of backend applications. While retreating to a desktop PC to seek help or to report a task's completion can be useful, the true value of mobile applications—and of IT in general—to the labor force is directly related to their ability to use IT at the exact moment they need it.

The different roles of mobile users

IT must plan to provide mobile solutions in every corner of the company. Potential mobile users fall into several categories:

The occasional user has a cell phone or basic smart phone with a small, text-only interface.

The classic field force requires mobile solutions that mirror their step-by-step procedures with devices and applications designed to speed up and streamline their day-to-day processes.

Engineers and expert users need larger, more flexible hardware—large screen PDAs, notebooks, and the like—for more complicated tasks and at times always-on connectivity.

Executive users want smart phones with a specially tailored subset of high-level functions for monitoring key areas of company performance.

Technology Trends

Mobile solutions are finally coming into their own thanks to a confluence of technology trends, most notably the rise of the Internet and wireless networking along with standard application platforms.

Wireless networks

The growth in wireless networks and the evolution of their architecture have expanded the possibilities for developers by creating, based on standards, flexible conduits for data synchronization and real-time connectivity.

Bluetooth is designed for use in the immediate area surrounding the device. RFID scanners, printers, barcode readers, and other devices can communicate with a mobile device like a PDA or a smart phone.

WiFi, sometimes called **WLAN**, is based on the 802.11 series of standards and provides wireless connectivity in homes, offices, or small areas.

GPRS, the General Packet Radio Service, is used with millions of GSM mobile phones to provide Internet connectivity to smart phones and PDAs.

WiMax, an emerging standard for wireless connectivity in a metropolitan area, will most likely connect users within a three to-five mile radius but has the capacity to reach up to 30 miles. With Wi-Fi as a workhorse, Bluetooth for peripherals, and WiMax emerging for power users in the field, developers already have the tools they need for creating mobile solutions tailored for a specific business process in almost every context.

Standard application platforms for devices

Another enabler of mobility is the creation of standard hardware and software platforms.On the software side, Java* (including PersonalJava*, Java 2 Micro Edition (J2ME*), and Brew*) provides a standard set of Application Programming Interfaces (APIs). Using these standards, the same application can be deployed to a PDA used by the sales staff or a heavy-duty device used in the field, and devices can be upgraded or replaced without affecting the applications. On the hardware level, the standardization also has two main platforms: Intel® Centrino™ mobile technology for notebooks and tablet PCs, and Intel XScale® technology for handheld devices and smart phones. Both hardware layers are optimized to enable wireless connectivity and great battery life,+ while providing maximum computing power.

From the first to second generation mobile solutions

The second generation of mobile solutions learned from the mistakes of the first, including a lack of true functionality and the false assumption that wireless networks can be dependably always-on. The failures of the first generation have left the impression, however, that mobile solutions are more of a toy than a reliable business tool.

Early solutions overstated the promise of the available technologies, like the Wireless Access Protocol (WAP) and miniscule interfaces for wireless devices. Entire enterprise applications were shoehorned into a tiny browser, ignoring both the performance constraints of those devices and the users' individual needs.

The overwhelming question "How will we benefit?" was rarely asked or else framed incorrectly during the first generation of mobility, which exhibited a tendency to embrace the technology for the technology's sake.

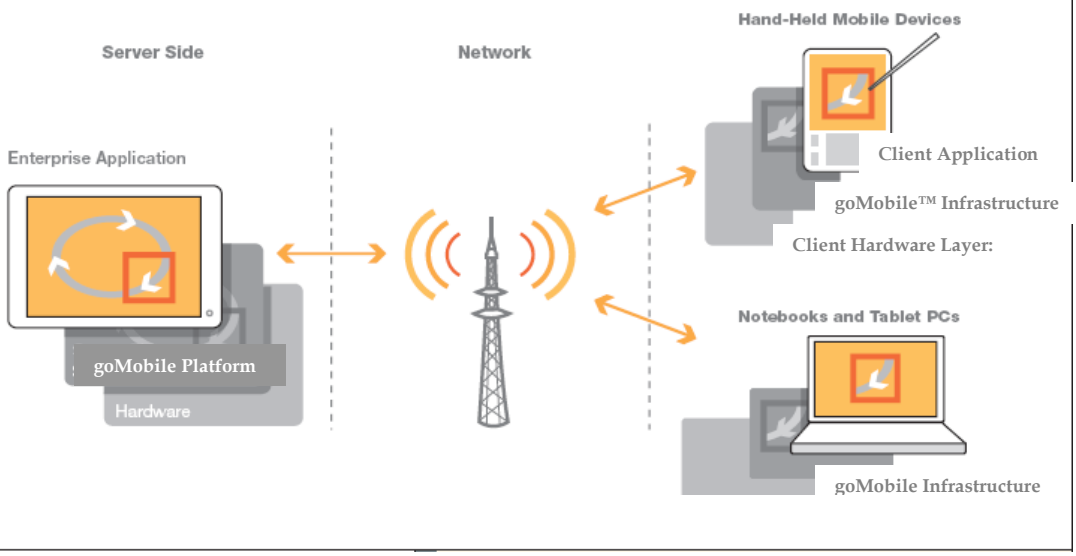
The design of second-generation mobile solutions now begins from the assumption that the user context is often disconnected and that he or she only occasionally connects to a reliable network to exchange data. Data isn't collected and handed off in real-time simply because connectivity is not always available even with today's networks. A disconnected application is one that has precisely enough functionality to meet the needs of its user and complete the part of the process that has been mobilized, which is, in fact, all that it needs to do. In order to be selective about moving down portions of the process, a layer like goMobile™ Infrastructure is needed to manage the translation.

First-generation solutions had no such layer. Imagine the driver of a Pepsi or Coca-Cola delivery truck. In this scenario, he begins his day with a preloaded truck and a handheld device containing data on the day's delivery route. At each stop, as cases of soda are unloaded at the customer site, he uses the device to print receipts, check inventory levels, or change orders on the fly, perhaps filling extra orders from the inventory on the truck. On his return to the warehouse, the device is placed in its cradle and it is synced with the appropriate backend applications. Why doesn't he synchronize this data on the fly using a wireless or cellular network? Mobile carriers claim only 80 percent or 90 percent coverage at most. Even with 99 percent coverage, an online-dependent solution fails the moment the driver encounters that one percent. Using an offline solution, process steps can be streamlined and improved without introducing a technological risk into the scenario. This is why field force management and sales scenarios supported by goMobile™, for example, build in the flexibility to work in an offline mode.

Architectural challenges of mobile solutions

FIGURE 1

Mobile solutions overview



A second-generation solution like this one is informed by both a more sophisticated understanding of how to mobilize business processes as well as advances in networking and mobile devices. Business processes are analyzed and the data and functionality that need to be mobile are carefully transplanted to the mobile device. The challenge for mobile solutions is to seamlessly move the data back and forth during times of connectivity so that, when disconnected, the user can accomplish his or her goals without thinking about whether the device is connected. As we will see in the rest of this paper, the mechanics of the goMobile™ Infrastructure to manage this interaction are quite sophisticated, as are the technologies Intel has delivered for enabling seamless wireless connectivity, high mobile performance and enabling great battery life.

IT architects will recognize in the second-generation architecture of mobile solutions many of the problems involved in supporting IT processes that span company boundaries. For example, instead of always-on connectivity, the reliability of the services provided by third parties must be managed. If the failure of critical third-party services stops a key business process, perhaps the right architecture can cache data or transactions to enable the business process to proceed. Such mechanisms are identical to the architectural techniques used in disconnected mobile applications.

Architectural challenges of mobile solutions

So far we have explained the larger issues surrounding mobile solutions: how they fit into larger trends in IT, why users are clamoring for them, and how technology has opened the door to rapid expansion.

But for those in IT managing the design, deployment, and operation of mobile solutions, often with the help of consultants and vendors, this level of detail is not enough. The key to crafting effective mobile solutions is a detailed understanding of the implementation problems and the structure of the solution. An IT department armed with this knowledge and keeping a sharp eye out for threats has an excellent chance of success. The rest of this paper addresses these issues, starting with a detailed look at the challenges inherent in a second-generation solution.

The big picture

The requirement for disconnected operation to support a mobilized portion of an enterprise application is the key driver behind most of the challenges of a second-generation mobile solution. Figure 1, which shows the broad outline of a mobile solution, will be used as a starting point for our discussion.

Figure 1 shows the elements on both the server side and the client side of the solution: On the server side, the goMobile™ Infrastructure manages data that needs to travel back and forth from the mobile device. This creates a gateway to the functionality of the enterprise application from the mobile device and provides a central point for management and administration of thousands of mobile devices.

The network transmits data back and forth from the server and mobile device via the Internet, using any of the wireless networking technologies or direct network connectivity for synchronization.

The client side includes the hardware and software layers of the mobile device. Here Intel® Centrino™ mobile technology or Intel XScale® technology works with the goMobile™ client-side services and the mobile application, either native or webbased, to help the user meet his or her goals. Simple enough. However, along with the advantages of a combined connected and disconnected computing environment, come the complexities and challenges. Successful deployment requires careful consideration to all of the following—especially security.

Integrating with enterprise applications

To make a mobile solution work, the goMobile™ layer must reach into an enterprise application and create a live copy of the data needed to support the part of the application being moved to the device. If the device is used to collect a transaction for the enterprise application, such as the creation of a new sales order, all of the information needed from the user must be collected and transmitted back to the server. The goMobile™ server layer then creates and executes the transaction by synchronizing its copy of the newly changed data with the original application, thus creating the new sales order. Enterprise applications are not yet built to support mobile applications. If they were, they would have services ready-made to support this activity. As it is, some sort of integration platform must be used to do this work. In all MobileOne Solutions, this work is done with goMobile™, the proven, reliable platform for application integration built with scalable, enterprise-class components, making it perfect for creating mobile and any other IT solution. goMobile™ comes out of the box with mappings for the data and services of many ERP solutions including but not limited to SAP, Oracle, Peoplesoft, Siebel, Salesforce.com, etc. that make creating mobile solutions that much easier.

Scaling and designing servers

For certain applications, the volume of data or the volume of users or both is too much for a single load-balanced complex of servers to handle. In these cases, the data or users must be partitioned across several different complexes of servers. For example, one set of servers could handle North America while the other set handles Europe. Scalability is also ensured because goMobile™ Solutions are built for distributed and heterogenous networks and can be in use at thousands of high-performance data centers worldwide.

Managing distributed data

In order to work properly as a disconnected solution, the appropriate data must find its way from the application to the mobile device. Any data that is changed must then find its way back. Of course, when the same data resides in two places, the possibility for a collision exists, and mobile solutions must have a way to resolve such problems.

Choosing the appropriate device

One of the most common mistakes in deploying mobile devices is choosing devices without the cooperation of users. Designers of mobile solutions frequently assume they understand the users and the way the device will be used but find out during deployment that their understanding was wrong. Of course, the right mobile device for most applications balances functional cost, computing power, battery life, and wireless connectivity. Through goMobile™'s use of Java and other standards, the same application can run on different devices, tailored for different users or different groups of users. Alternatively, each class of user might have the same mobile device (which is frequently optimal and simplifies maintenance, administration, and security) but have a different set of mobile applications depending on their job.

Managing devices and applications

Mobile devices tend to be used in mass quantities. It is not uncommon for thousands or even tens of thousands of devices to be part of a single deployment. Some sort of automated administration and maintenance system is required to handle so many devices and track which users have which applications. An upgrade to an existing application, a bug fix, or deployment of a new application cannot realistically entail 10,000 visits to devices.

Implementation and deployment

More than most sorts of IT projects, mobile solutions involve moving parts from a variety of third parties. The enterprise applications, the integration technology, the network, the client device, and the client software may all come from different vendors. Orchestrating cooperation in such an environment can be challenging. Implementation requires setting up a complex environment. Mobile solutions are often tricky to deploy. Sometimes the mobile and non-mobile solution must be run in parallel for a time, or some sort of big bang roll-out must be pulled off. One of the most challenging parts of deploying mobile solutions is ensuring performance. Portions of large enterprise applications, potentially with large amounts of data, must be squeezed through a narrow wireless pipe and then run on a mobile device with a fraction of the power of a server in a data center.

Keeping an eye on performance from the start is crucial to successful deployment of a mobile solution (including usability, enabling great battery life, and optimizing synchronization). Poor performance, for whatever reason, is a surefire way to kill a mobile solution.

Security

The most common difficulty most IT departments have with mobile solutions is managing all of the different aspects of security that come into play. The data must be secured as it moves from the enterprise application through the server-side technology, across the network to the device. Generally this problem is well understood. But once the data gets on the device, IT executives start to worry. Who will be using that device? How can they be validated on the device level? How does the device identify itself to the server? How is the data secured on the device? What happens if the device is lost? The fact of the matter is that it makes IT executives nervous to think of all of the data on all of the mobile devices traveling across the world. The only remedy is a comprehensive vision of security that includes mobile solutions. It is a matter of taste how much of this is provided by vendors and how much an IT department implements itself. The issues just discussed must be confronted when deploying a second-generation mobile solution. To create and roll out enterprise-grade solutions, an IT department must meet these challenges. The question is: “How?” As the next section explains, goMobile™ has an end-to-end architecture aimed precisely at creating robust, enterprise-quality mobile solutions. In a mobile solution, there is no reason that the data and functionality must come from just one enterprise application. With goMobile™, it is possible to construct a mobile solution using portions of many different enterprise applications. Substitute a role-based interface for the mobile device in the discussion above, and it becomes clear that mobile solutions are really another flavor of composite application. The same set of problems just mentioned for mobile solutions also must be solved in much the same form for composite applications, applications built out of services from other applications that are one of the key benefits of a service-oriented architecture. As a serviceoriented infrastructure is built, with only a little care and attention, almost all the work done to create services will serve to support mobility as well.

The technology of mobile solutions

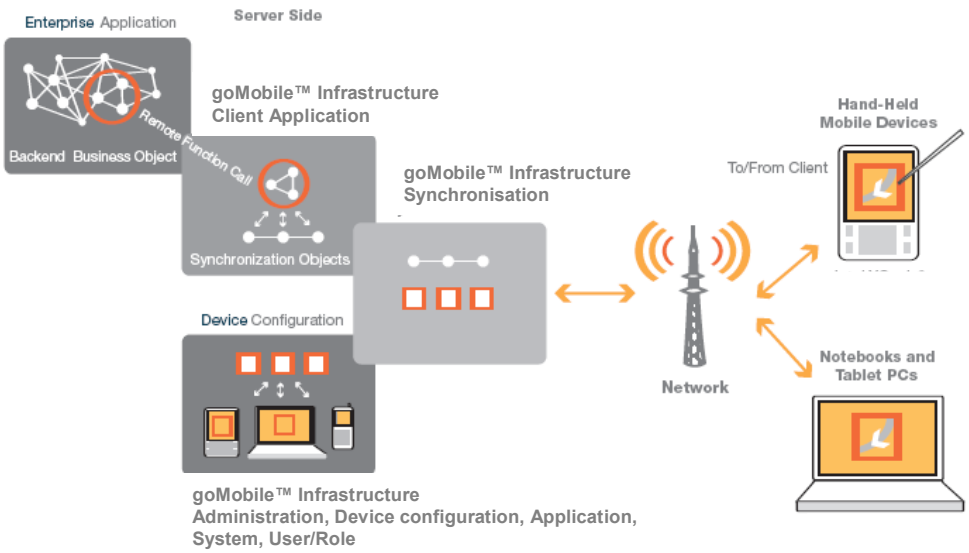
Mobile solutions are fascinating because of the way that an application is distilled to its essence in becoming mobile. Complex data on the server must be precisely sifted and simplified. The networks that extend to mobile devices are expanding their coverage but are still a narrow pipe. Bandwidth must be used sparingly. Mobile devices, although powerful, are designed to meet the needs of a mobile workforce. In turn, some features of traditional desktop networks must be sacrificed. Therefore, maximizing the benefits of mobile solutions presents a great design challenge.

Mobile solutions also provide a significant challenge to vendors for the same reason. The software on the server must provide a toolkit for streamlining data on the way to the device and enriching it on the way back. Imagine a piece of data's attributes as the luggage it takes with it on its trips across the corporate network. The elaborately detailed, rich data of enterprise applications might be visualized as steamer trunks packed with granular detail. To send that data to a mobile device, it's necessary to pare the detail back to the equivalent of a weekend bag—a careful selection of detail needed for the task at hand and nothing else. Upon its return from the device, the data is enriched; its changed attributes are unpacked and repacked into the trunks. Thousands of devices must be managed. The software on the device must have a small footprint that makes efficient use of scarce storage, memory, processing speed and power. The chips on the device have the same challenges and must pack the most computer power and wireless connectivity into a package that enables great battery life. MobileOne has worked to make sure that their platform for second-generation mobile solutions addresses many of the general problems associated with the creation of mobile applications. This allows those who design and build applications to focus on the work the applications must do, instead of solving infrastructure problems. goMobile™ Infrastructure helps a mobile solution reach into the complex data structures of the enterprise application for just the right data to pass to the part of goMobile™ that helps manage data and build applications on the client side. goMobile™ also provides a framework for managing thousands of mobile devices.

The following discussion explains goMobile™ and the benefits of mobile solutions by describing the trip that data takes, starting with the enterprise application on the server through the distillation of the data, its transmission of the network, and its use on the device.

FIGURE 2

Server-side architecture of mobile solutions



The server side

As Figure 2 shows, the server side of a mobile solution has three main jobs:

• Providing a gateway to the application on the device for data or information synchronization.
• Managing the deployment and configuration of applications on the device.
• Managing server-side security, authentication, and access. goMobile™ approaches these jobs with a set of general purpose abstractions such as synchronization objects and frameworks for performing crucial tasks such as device administration. The goal of goMobile™ is to provide a toolkit so powerful that application developers write only the smallest possible amount of custom code.

Integration with enterprise applications

A mobile application starts with an understanding of the data and functionality that must be moved to the device. In most of the first-generation WAP-based mobile solutions, not much thought was given to this issue. The entire server-based application was shoveled through the narrow pipe to a mini-browser. The devices were also crude and had little functionality beyond a web browser with limited functionality. Few victories were achieved in this manner.

The key to the second-generation mobile solution is to take a close look at the goals the mobile user is trying to achieve and then to move just the information required to meet those goals to the device. The most robust architecture for a mobile application never requires synchronous connectivity—that is, a live connection with the server. As a practical matter, most applications take advantage of synchronous connectivity when it is available but don't rely on it too much. The first interesting problem for application architects using goMobile™ begins once the data that must travel to the mobile device has been identified.

Data models in enterprise applications are generally incredibly complex and abstract so that they can be repurposed to meet different needs and stored efficiently. The customer record in an ERP system, for example, might be stored in 10 or 20 different tables. What goMobile™ does is reach into the enterprise application through the APIs and creates a two-way pipeline for the data needed for the mobile application and moves it to an intermediate database called the replica database. Two points help make it clear how this extraction takes place. The first is that for both ERP and non-ERP applications, goMobile™ is used to help extract and reformat the data from the enterprise applications. The second point is that the data need not be extracted from only one enterprise application. It is possible that a mobile application might use processes or web services from an ERP, CRM, and SCM system. In this way, mobile applications have the same power as composite applications: that is, applications that are built out of services. Building composite applications is one of the key reasons companies are pursuing service-oriented architectures, and, as we mentioned earlier, mobile applications are similar in their structure.

Once the data has been extracted, it is stored in a simplified form in what is called a synchronization object. The replica database is actually a collection of synchronization objects. Synchronization objects are very simple objects, usually with flat data structures, that are used to send data to the mobile device through one of several synchronization methods which we will discuss shortly. All of the data extracted from that customer object in ERP might be stored in one table in the synchronization object. One implication of this simplification is that when data is changed on the mobile device and sent back to the replica database, it must then be enriched and turned back into the complex form so it can be stored in the enterprise application.

One key question in any mobile application is: How does the data move from the enterprise application to the replica database? In most applications, goMobile™ does some sort of polling to see which data has changed so it can update the replica database. It is also possible that some sort of application trigger on the enterprise application can notify goMobile™ when some data has changed and push the data out to the replica database. The right approach depends on the nature of the application or what the business process requires and is not prescribed by goMobile™. So far so good. But what about functionality such as the ability to execute a transaction to create a new order or to invoke some process? How can the mobile device execute such a transaction especially in a disconnected architecture? The way that this works is similar to the relationship between the objects in the enterprise application and the synchronization objects. What happens is that the information needed to perform a transaction is collected on the mobile device in a synchronization object which is then brought back to the replica database. goMobile™ then uses that synchronization object to enrich the simplified description of the transaction on the device with additional information so the transaction can be executed against the enterprise application.

Data synchronization

Now the fun begins. The replica database contains all of the data that might be delivered to any device that is running the particular application linked to the replica database. To make a mobile solution work, goMobile™ must keep track of what data in the replica database is supposed to be delivered to each device. To make this happen, at some point goMobile™ must be told what section of the replica database should be sent to each individual device. This is part of the device configuration, the administration of which will be covered shortly. Once goMobile™ knows what part of the replica database should be on each device, it patiently waits for that device to log in through the user sign-on mechanisms. When the device is recognized, synchronization is initiated using one of several techniques. The data is transferred to the device or from the device so the information can be entered in its rightful place in the enterprise application. There are two techniques and two flavors of synchronization. Generic synchronization sends all the data on a one-way trip. This synchronization technique is used for reference data that will never be modified, such as the values of drop-down menus and such. If the data in generic data is changed, a whole new copy is sent to overwrite the older data. Smart synchronization is the technique most people are familiar with when they synchronize their PDA with the data on their notebook. Data can be changed in both places, on the server or on the device. The synchronization must figure out what the updates are in both directions and send the right data along. One of the great things about smart synchronization is being able to change the data in both places. But this comes at the risk of collisions; that is, when the same data is changed in both sides of the synchronization. Smart synchronization must have some mechanism of resolving collisions, either automatically by having one side win or by the intervention of an administrator. The two flavors of synchronization are sending all the data during a synchronization or only sending the changes, also called deltas. Sending only deltas is one of the most important techniques used to minimize bandwidth and ensure high-performance synchronization. One of the ways that goMobile™ makes the system more robust is by tunneling the synchronization protocol through HTTP, a port that is sure to be open on most networks.

Scaling a mobile solution

In practice, a goMobile™ server might be load-balanced across several servers to increase performance and reliability. goMobile™ has another feature that can support even more scaling by partitioning the data across many different groups or nodes of goMobile™ servers. For example, let's say that an enterprise application has so much data and so many users that one large server complex would be overwhelmed. goMobile™ allows the data for North America, for example, to be channeled from the enterprise application to one node, the data for Europe to go to another, and the data for Asia to still another. This approach allows for virtually unlimited scalability of a mobile solution. goMobile™ does this without forcing the same partitioning on administrative functions, which can still be performed centrally against one database for all users.

Device administration

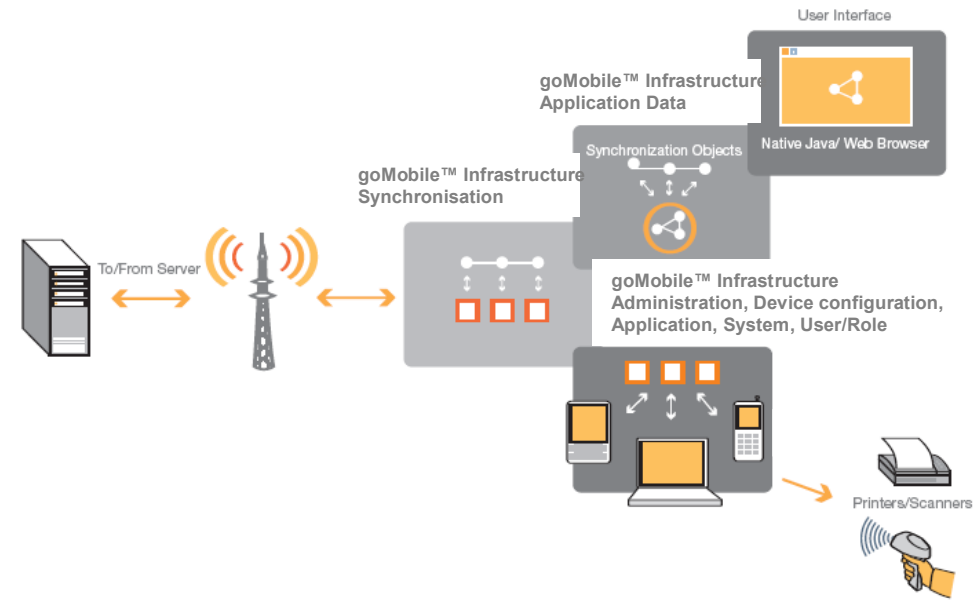
One of the challenges of mobile solutions is that the number of devices can quickly run into the thousands, potentially creating an administration and maintenance nightmare. goMobile™ simplifies this problem by mapping thousands of users into a small number of roles. Each role is associated with a specific set of applications cations that should be installed and maintained on a particular device.

Users can be grouped in other ways as well to manage their access to certain types of data. The profile for each user is synchronized to the device at the same time data is. In this manner, new applications can be rolled out or maintained and devices can be configured for the first time or restored if data is lost. Device administration is handled simply via a web console, and configuration data can be backed up and restored separately from the replica databases on the server.

Support policies for mobile devices must be carefully designed. When an end user has a problem, do they mail the device in? Do they go to a service center? What happens when the device is broken or stolen? What happens when the battery dies? Does the end user call the administrator? How can administrators have access to the device? Must you be at a cradle to support the device? When in the user's day can you support the device, once a day in the office or at any time in the field? It is crucial that training on mobile solutions also includes clear communication about how to get support. It is important to point out that goMobile™ can simply do all of the jobs mentioned so far.

FIGURE 3

Client-side architecture of mobile solutions



The client side

Development of applications on the client side is far trickier than one might first suspect. The interplay between application architecture and the operation of the hardware of the device can turn into a ballet of efficiency or a rugby scrum of contradictory forces. Here's a simple example to make this point. One of the principles in our upcoming discussion of how Intel architecture enables great battery life is that much of the memory, I/O ports, supporting chipsets, and the display are powered down when not in use. Let's say, for example, that a client-side application decides to poll every so often to see if any new data is ready on the server. This structure would eat up the battery power of the device by turning on many chips that could be turned off if the application were changed so the server notified the device of changes by sending an alert. Client applications must be written with an awareness of the device from top to bottom. goMobile™ applications are streamlined for maximum exploitation of the client.

Figure 3 shows all of the moving parts of the client:

In any given business scenario, the "client device" may refer to a smart phone, a PDA or handheld PC, a notebook, or possibly a custom device with a specific dedicated use. Whatever the client device, the goMobile™ architecture supports Java, ensuring a common platform for any developer wishing to build tailored mobile applications that are platform-independent.

Data on the client side

The data on the client side is a collection of synchronization objects just like on the server side. The only difference is that on the server side the replica database has all of the data for the application and all users while the client side has only the data needed by the applications and users of that particular device. goMobile™ on the client side performs the same synchronization duties as the server, sending and receiving data using whichever synchronization mechanism the application designers selected.

Client-side application architecture

Application designers have to make the choice between two styles of applications on the client side. Applications can either be based on the web server/browser paradigm or be written in Java, the so-called native approach. The web server/browser architecture is an innovative example of software miniaturization. In this approach, a stripped-down web server runs on the device. Java Server Pages are used to create the user interface for a browser also running on the device. The Java application code resides inside the web server and communicates with the user through the local browser. This approach allows server-side skills to be repurposed to create client-side applications. In the native approach, Java code is written to perform the work of the application and to create a custom user interface using the Abstract Window Toolkit (AWT) that is part of the Java API. In both approaches, the Java application code uses API libraries provided by goMobile™ to access data on the device, to perform synchronizations, to access the server through a synchronous connection and to perform other utility functions.

Device and application configuration

The client side is on the receiving end of the synchronization of configuration information and applications. In most cases, anything about the device can be changed except the operating system, which requires maintenance by IT personnel at a workstation.

Client-side hardware and peripherals

Client-side devices are frequently connected to peripheral devices such as printers, UPC barcode scanners, RFID readers, and other such equipment. Such devices are key to providing a complete solution to many different kinds of applications. goMobile™ provides a library of drivers for a wide range of barcode scanners, RFID readers, printers, and other special purpose devices that enhance the productivity of mobile solutions.

Client-side security

User security presents its own development dilemmas on the client side. Mobile devices are at risk of loss or theft as they travel around the world with users. Once lost or stolen, the devices could be vulnerable to extended periods of hacking. At the theoretical limit, a complete remedy is not available. If a device falls into the hands of an expert hacking team, the data on it will be vulnerable. The same is generally true of serverside security. The best hackers generally can find a way into almost any system.

Data on a mobile device can be protected in many ways. Data stored on the device may be encrypted. Data moving to and from the device may be encrypted as well. The device can have its own layer of authentication including advanced techniques such as biometrics. The device can delete its data if it has been inactive or not synchronized for a period of time. The bottom line is that security on a mobile device is one aspect of a comprehensive approach to security. Mobile devices should be made part of a company's security policy and should not be a special case. When users sign on to server applications using goMobile™, all of the same server-side authentication used for enterprise applications is used to ensure the right person is there and that he or she has access only to appropriate data and functionality.

Design and deployment roadmap

A generic roadmap that meets the needs of all potential deployments would be quite a valuable document and, unfortunately, impossible to create. What can be offered, however, is advice collected from the pioneers in deploying mobile solutions of the sort that is summarized in Figure 5.

Most IT departments will deal with mobile technology first as an experiment with an important application, then as an increasing number of requests, and finally as an extension of the supported computing platform.
One of the first problems faced is similar to what happened with web sites in the late 1990s. Usage spontaneously exploded outside of the management of IT. Early implementers recommend a three-stage approach to the initial problem of bringing mobile devices into the fold:

•Connect devices that may not be under corporate management to the network.
•Start managing those devices by putting a standard layer of application and utility functionality on them.
•Secure the devices and control what may be done with them.

Once a standard platform for mobile computing is offered by an IT department, early implementers recommend first offering horizontal applications such as access to email, calendar, and address book and then later follow up with appropriate vertical mobile solutions based on ERP, CRM, and SCM applications. Early implementers caution that for mobile solutions, one device does not fit all. Usability is key—consideration should be given to where the device will be used, how it will be used and who will be using it (matching form factors to user groups). It's also important for rollout to occur in stages; pilot users are generally forgiving and will give a solution some time to improve, but end users expect perfection and quickly lose interest in the face of difficulties.

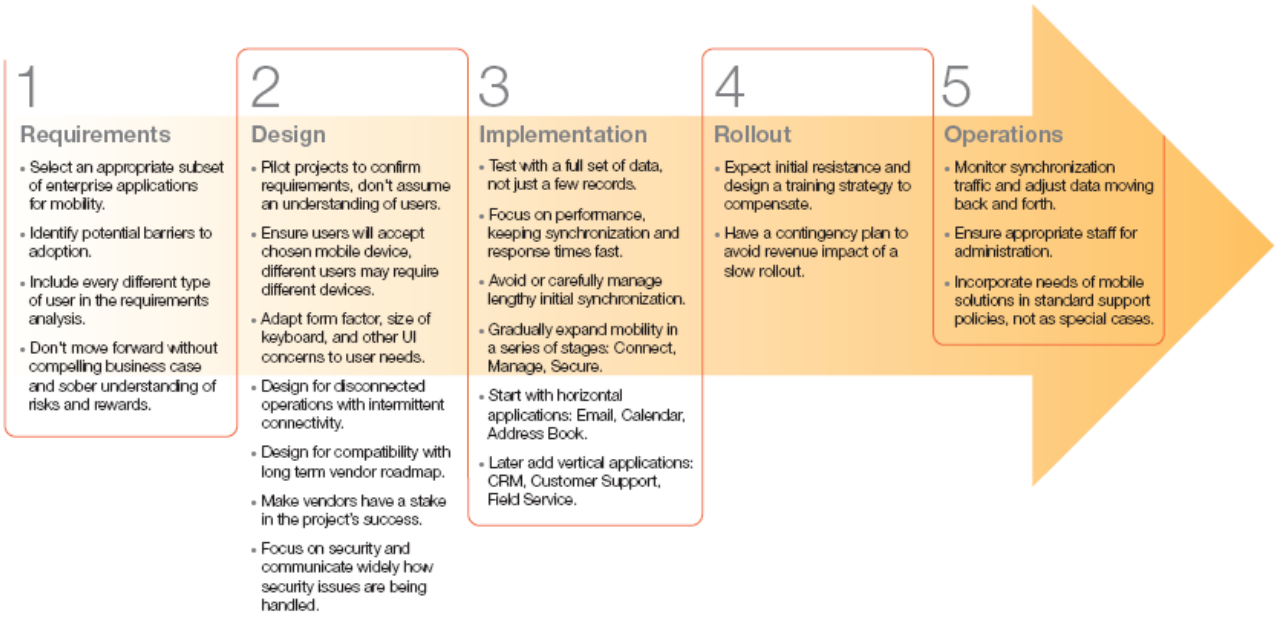
During implementation, some of the most common problems include making incorrect assumptions about bandwidth, network speed, device capabilities, the amount of data required on the device, and the time required to synchronize the data. Planning for synchronization is one of the most problematic areas. Early implementers point out that capacity planning must account for the bursty nature of synchronization. For disconnected devices used in the field, all users may synchronize at about the same times, right at the beginning and end of shifts, for example, which can have a dramatic impact on the need for capacity.

IT departments often underestimate the time needed for the initial sync, which can result in users waiting hours for the initial synchronization of the device to complete over a narrow wireless pipe. Frequently, the initial sync is best accomplished through a wired network. Testing with enough data to really represent the application instead of just a few records goes a long way to finding potential synchronization problems.
One irony of mobile computing is that by planning for a bad network you get a better computing environment. Smart mobile computing reduces latency and enables users to make progress regardless of network connectivity.

FIGURE 5

Roadmap for design and implementation of mobile technology

A guide to best practices and common pitfalls.



Service-oriented architecture revisited

Now that every facet of mobile solutions has been explained, it is possible to return to the initial argument of this paper with a new perspective. Mobile solutions have a variety of characteristics designed to accommodate the unreliability of wireless connectivity. What has not become clear yet to the larger market is how the same techniques used for mobile computing will be applied to the creation of a service-oriented architecture.

But if we look at the borders of the enterprise and imagine how we will support key cross-company integrations, isn't this a similar case to the server side of a mobile solution? Don't mobile applications resemble composite applications in the way that they can use services from anywhere? Shouldn't the prudent IT department plan to keep a cache of all data that is being used from the applications of remote partners so that when connectivity is lost, operations in the company can continue? Shouldn't all server based applications be able to partition themselves to scale? Won't the security demands of a cross-company integrated application be higher than that demanded by the current architecture? While a mobile solution is not the equivalent of a fully-realized service-oriented architecture, implementing mobile solutions provides the rare opportunity to make users happy right away, provide a significant ROI, and prepare for the future at the same time.