JARGON

White Paper:

Extending Enterprise Applications with Mobile Devices

Jargon Software Inc.
Minneapolis, Minnesota

http://www.jargonsoft.com

Updated October 2010
Updated August 2009
Updated January 2009
Updated September, 2007
May 2004

Overview:

This white paper is primarily intended for business and technical managers who are evaluating the opportunities and challenges involved in providing mobile workers with field access to enterprise applications.

We examine the business reasons for "going mobile," the usability and drawbacks of current mobile hardware devices, the pros and cons of various current software architectures, some desirable features of a mobile architecture, and how Jargon Software's technology tools address these needs.

The objectives of this white paper are to enable readers to understand:

- Pros and cons of traditional mobile application architectures, such as thin client and thick client approaches.

- The issues and considerations that must be addressed for successful mobile applications.

- How Jargon Software's mobile development and deployment tools incorporate these success factors for mobile applications.

Contents:

- **Why Go Mobile?**
- **"Heads Down" Mobile Applications**
- **Order Entry on a Cell Phone? (mobile hardware overview)**
- **Is your mobile application too thin or too thick? (or just right?)**
- **Jargon Software Architecture**
- **Summary**
- **References**

- **About Jargon Software**

Why Go Mobile?

Mobile devices offer many benefits to remote workers in field sales, field service, field inspection, warehouse operations, and similar "non-desk jobs." Foresighted organizations are deploying applications on these devices to their remote workers, as the speed, coverage areas and reliability of wireless data networks continue to grow.

Mobile applications provide easily proven ROI (Return On Investment), which has been well documented in many recent studies and articles and available separately from Jargon Software.

The many benefits of mobile applications include:
- Better access to customer and product information increases sales effectiveness, thereby increasing revenues.
- Mobile data access and validation reduces error rates, which lowers costs and provides better customer service.
- Mobile data access reduces the need for printed reports, which reduces costs, provides more timely information, and improves productivity.
- Mobile data access virtually eliminates the need for telephone calls to the home office to obtain information, which lowers costs and improves productivity both in the field and at the home office.
- Better customer service increases customer satisfaction and loyalty, resulting in repeat business, higher revenue, and greater strategic advantage.
- Automated data capture and activity tracking with mobile devices provides more complete real-time information for management decision-making, which increases strategic advantage.

"Heads Down" Mobile Applications

"Entry level" for mobility includes basic generic capabilities such as voice calls, voice mail, instant messaging (IM), email, web pages, and "push" content delivery for stock quotes, as an example.

While this level is the most easily deployed and provides many useful features, it only scratches the surface of the potential benefits that can be obtained by providing deeper levels of access to corporate information.

An often overlooked but real cost of voice calls, voice mail, email and instant messaging is that it always consumes two people's time, one to compose/send the message, and the second to receive/record it. This doubles the labor cost of accessing a specific piece of information.

What's more, voice calls and IM are intrusive in that they interrupt the person being called. If the person who needs information can access it directly, when (and only when) it is needed, without involving anyone else, this provides a clear advantage for improved productivity.

This discussion focuses on important enterprise activities that require mobile access to one or more back office database applications, such as order entry, customer service, CRM, inventory management, and similar core business functions. Examples include:

• **Sales Order Entry** extends across virtually all industries where sales reps are working outside of the office, frequently taking orders, and communicating them to the back-office order processing system. The sales rep downloads important customer and product information (including history) prior to a sales call and accesses it during the call without requiring a connection. When the order is complete a digital signature is obtained and uploaded along with the order directly into the backend system. This is all completed without needing in-office support and reduces delays in finding information. Call-backs to the sales rep are similarly reduced.

• **Route Sales and Direct Store Delivery**, example: the food distribution industry. In addition to the benefits of the mobile sales order entry, drivers also download call lists and route scheduling data.

• **Service industry** mobile applications include repair history, parts requisition, warranty status, and dispatch.

• **Chain of Custody** mobile tracking of sensitive, confidential information or products.

The technical challenge to implement these applications is to develop and easily deploy programs on mobile devices which work connected or disconnected and that interact with a back end enterprise application. Included are embedded SQL databases on the mobile device, which provide powerful local processing capabilities. These features are essential because service can be easily disrupted, as well as questionable operations in fringe areas or areas with no service or where service is prohibited (like a hospital).

Order Entry on a Cell Phone? (mobile hardware overview)

Mobile devices with adequate ways to enter information, and large enough screens to display more than simple text messages, are needed for field business applications. Small screens and limited input mechanisms of common cell phones can inhibit productivity. Smart phones, PDAs, netbooks, tablets, and notebooks provide the needed input and display features.

Terminology

PDA - Personal Digital Assistant, a handheld computer
WPDA - Wireless PDA using Wi-Fi, broadband, or cellular network connectivity
Smart phone – a WPDA using a cellular connection including cell phone features
Netbook- A compact and inexpensive notebook with limited power and local storage, but excellent for use with thin client applications
Tablet PC - Specialized notebooks with touch screens and small form factor

Mobile Operating Systems

Windows: CE, PocketPC, Mobile 2003, Mobile 5, and Mobile 6
Linux
Startup OS platform from Google
Palm OS – relegated to a small retail segment

Mobile Hardware Options

SD memory cards (same as used with digital cameras)
Bar code scanners
RFID readers
Credit card swipers
Bluetooth printers

Image and video capture
GPS
Other Bluetooth devices, including headsets and microphones

Connections

Wired: Traditional telephone modem plugs into wall outlet and provides point-to-point (PPP) connection.

Wireless:

- Wi-Fi hot spots
- WAN
- Commercial broadband
- Cellular

Wireless service is not uniform
- Spotty in rural areas, especially in mountains and rough terrain.
- Good in and near major cities.
- Good on and near major highways, like the Interstate and federal roads.
- Inconsistent even in urban areas due to obstructions, like basements, and areas with massive metallic objects, like girders and beams.
- For the mobile user a continuous connection is both expensive and causes decreased battery performance and life.

Is your mobile application too thin or too thick?

Nearly all mobile applications to date have been implemented using either a "thin client" or "thick client" approach.

Thin Clients

A "thin client" is defined as an application that stores little or nothing locally and is small and compact. All programs and data are stored on the server.

For mobile devices, this means that no special software or data storage resides on the device, except whatever came with the device originally. Thin clients often utilize a mobile web browser (like Pocket IE with its reduced functionality). Sometimes a thin client can be implemented using remote desktop, telnet, or Citrix (like GotoMyPC).

The primary advantage of thin-client designs is that there is no deployment requirement. The devices are able to provide real-time access to data "out of the box."

However, the disadvantages are **numerous and more serious**:
- If the software is too thin, functionality will be lost.
- Browser, remote desktop, telnet, and Citrix based applications only work if you are connected. If you cannot connect or lose your connection, you're stopped in your tracks.
- To support a variety of devices with different pocket web browsers, expensive server-based "transformation middleware" products are used and integrated into existing back-office applications. More moving parts, more to break.

Industry analysts have recognized these issues:

> *"Let's face it, in terms of developing friendly, responsive user interfaces, browser-based programming is a huge step backwards. HTML and the browser were not designed to be a platform for application user interfaces. They were designed for hyperlinked documents."*
> > *MSDN Column "Death of the Browser?" Billy Hollis - October 14, 2001*

> *"Ever talk to a major call center or field sales deployment of any of the various Internet-based Customer Relationship Management (CRM) applications? Well, if you haven't, it's not that pretty*
>
> *... (M)ost end users are struggling with the decreased usability and performance of the applications in Internet mode.*
>
> *... (U)sers report serious deterioration of application performance due to the numerous roundtrips, cumbersome navigation, and bogged-down networks. For sales reps, there are three scenarios: either they have lost any disconnected use of the system, run a small data center on their laptop,* or have a slimmed down application that limits functionality."
> > *AMR Research column "The Overselling of Internet-Based Architecture and the Rebirth of Client/Server" - Rod Johnson - May 19, 2003*

Many problems with current web apps are noted by Forrester Research:

- **Primitive functionality:**
  "HTTP was designed for document browsing, not complex application interactions."
- **Bandwidth gluttony:**
  "Each time a server needs to update a few bytes on a Web page it sends the entire page."
- **Terrible usability:**
  "Forrester has evaluated more than a dozen browser-based apps to see how well they meet end-user needs. Not one application has earned a passing score."

Thick (Fat) Clients

A thick client (often called "fat client") can be defined as software in a client/server environment that performs most or all of the application processing with little or none performed in the server.

For mobile devices, this means most or all of the application software and data storage resides on the device. The programs are written in a 3rd-generation language such as C, C++, C#, Visual Basic or Java. The data storage often uses a scaled-down version of an enterprise database such as those offered by Oracle, Microsoft, and Sybase, and require server-based synchronization software products. Again, there are more moving parts on the server and more to break.

The primary advantage of thick-client designs is that they do not require a continuous ("always-on") connection. They are designed to work in a disconnected (off-line) mode. This overcomes one of the main weaknesses of thin client designs.

However, the disadvantages of a thick client (like a thin client) are **numerous and more serious**:

- If the application is too thick, deployment and maintenance become backbreaking projects that consume large amounts of time and money.
- Local applications typically do NOT provide any online access.
- 3GL-based applications are the most expensive type to develop.
- Thick clients usually require elaborate batch client/host synchronization procedures to upload/download data between the client database and the central server database, requiring expensive server software and often pricey "mobile databases" on the server. These databases in turn require synchronization to the back office application's database, adding additional complexity on the server.

A note on .Net

Microsoft's .NET architecture is a thick client, although with better connectivity. It requires a very large client footprint, programming in a 3GL, and a very complex architecture with large overhead.

The .NET client side CLI (Common Language Interface) competes with the Java Runtime Environment (JRE). It does NOT solve deployment problems. On the host side, its Messaging Servers use a SOAP interface which competes with middleware vendors and with messaging server vendors such as IBM and Oracle.

The "X Internet" client, a "rich client" and a size that's just right

The term "X Internet" client was coined by Forrester Research, to refer to an eXecutable Internet client with software that falls between the thin and thick models. Such a client was defined in two Forrester articles:

The X Internet Revives UI Design [1]
This article discusses the problems with clients that are too thin. The mobile environment magnifies the consideration of these issues:

- Simplistic Web UIs disappoint users and designers.
- New smart clients and UI servers aim at richer experiences.
- X Internet apps bring back split-second performance.
- Designers must use cached data to anticipate user actions.

X Internet Clients Save Enterprise Apps [2]
Forrester spoke with 30 IT execs at $1 billion-plus firms. Thin client UIs are favored by 73% of respondents, citing ease of deployment, simplified access to applications, and lower maintenance costs as top advantages. More than half say that poor usability plagues their enterprise apps.

X Internet tools let firms build Net-based apps that:

- Boost user experience.
  "improved functionality and responsiveness"

- Trim bandwidth consumption.
  "1 KB to 2KB XML files replace 50K to 200K Web pages"
- Leverage current application investments
  "support XML-based Web services"
- Unleash designers.
  "With executable Internet apps, designers can take advantage of real-time data manipulation and rich client-side functionality"
- Utilize Microsoft .NET capabilities, but without the traditional development overhead.

Jargon Software Architecture

We believe (with Forrester) that the future of business applications includes the Internet architecture, X Internet (i.e. "rich clients"), and mobile platforms. To this end Jargon Software has created a rich mobile client application engine called **Jargon Reader**. We utilize XML files to define the client user interface (UI), and we embed JavaScript which is run on the mobile device. Developers create applications that directly manipulate individual client components from any HTTP-compliant middleware, or via JavaScript.

Our rich client application engine downloads XML from a host, similar to reading web pages. Since these XML pages are hosted, deployment overhead is eliminated. Successful mobile applications must be able to run connected or disconnected. This results in the need to store local data. **Jargon Reader** stores low volume data automatically in text, table and other components. Embedded SQL databases are used for high volume storage requirements. Synchronization is achieved at the application level, resulting in more efficiency and eliminating the additional expense of mobile middleware and databases.

The **Jargon Reader** rich client also supports mobile devices such as Bluetooth printers, bar code scanners, video capture, etc.

All of this is built into our mobile solutions.

Summary

Jargon Software's applications incorporate an X Internet (i.e. "rich client") mobile application engine and Internet architecture making possible mobile applications that are easy to modify, extend and deploy, thereby enhancing and extending the life of existing back-office enterprise applications. Jargon's mobile approach delivers a cost effective mobile solution that is partially connected, uses fewer moving parts, fewer security considerations and open standards.

The benefits provided by Jargon Software technology include:

- **Connected or disconnected mobile applications**

- **Applications run on handheld and mobile netbooks, notebooks, and tablets**

- **Self deploying using Internet technology**

- **Connects to multiple host environments**

- **Applications are easy and fast to develop and deploy using open standards.**

References

1. <u>The X Internet Revives UI Design</u>
   Randy Souza
   Forrester Research Inc.
   April 2002

2. <u>X Internet Clients Save Enterprise Apps</u>
   John P. Dalton
   Forrester Research Inc.
   December 2002

About Jargon Software

Jargon Software Inc. is a Minnesota based software development company, founded in 1997, that provides innovative solutions for the mobile e-business applications and software tools market.

Contact Information:

Jargon Software Inc.
8700 W. 36th St. Suite 209
Minneapolis, MN 55426

(+01) 952 426-0858
(+01) 866 568-4291 Fax

Clark Bain, President
clark@jargonsoft.com
www.jargonsoft.com