



# Exploring Mobile Applications Technology Landscape

Leveraging innovative solutions to achieve significant business value

Whitepaper

## Table of Contents

Introduction	1
Drivers of Mobile Applications	2
Entry Barriers to Mobile Applications Adoption	3
Approaches to Mobile Applications Development	4
Thin Client Application Development	5
Thick Client Application Development	5
Mobile Application Development Approaches – A Comparative Matrix	8
Why Mobile Application Framework?	9
Advantages for Mobile Developers	10
The Next Generation Pervagus Mobile Framework (PMF)	12
Conclusion	12
Glossary	13

## Introduction

In the present fast-paced business environment, the adoption of innovative mobile technologies play a crucial role in an organization's business strategy, delivering significant value. To gain competitive advantage, organizations are fast moving into wireless space to equip their workforce with access to vital information, anytime, anywhere.

Harnessing the power of mobile technologies, businesses automate and streamline their business processes with future-proof mobile applications that result in improved productivity and lowered operational costs.

While there are influential drivers for wireless technologies, the market is still in its infancy, in that, it calls for a cautious-and-strategic method in extending businesses onto mobile applications. Selecting the right mobile technology platform to develop mobile solutions catering to specific business need is of paramount importance.

The enterprise market for mobile computing is estimated to be \$ 30 billion – J.P. Morgan study.

This is possible through a unified approach that quickens integration processes and helps organizations stay ahead of the mobile curve. To achieve this, it is imperative to understand the existing and evolving approaches to mobile applications development architectures. An ideal mobile solution must balance between the present functional capabilities and extend them into the future technology advances.

This white paper explores the different approaches prevalent in the mobile applications development space, their challenges, and the advantages of embracing a comprehensive mobile application server to fulfill all of the mobile challenges.

## Drivers of Mobile Applications

Remote workers are faced with the inconvenience of accessing their resource information while on the move. Additionally, workers are unable to update their data and provide timely services to their customers. Mobile applications eliminate these labor-intensive and error-prone processes and improve operational performance.

Mobile workforce will grow from 15% - 38% of the total workers by 2005, predicts IDC research.

Realizing the plethora of opportunities available, organizations are making a radical shift towards embracing mobile computing in their business practices. Wireless applications enhance mobile workers productivity through improved decision-making capabilities, less paperwork and reduced cycle times for transactions and billings. The main drivers of mobile applications include sales executives, field technicians, maintenance workers, delivery staff, and workers in areas of healthcare, retail, manufacturing etc.

## Entry Barriers to Mobile Applications

While the mobile market is progressing at a rapid stride, the major components - devices, wireless networks and applications - are constantly evolving at varying speeds.

### Devices

The wireless market is developing with a myriad of pervasive devices and it becomes a Herculean task to deploy mobile applications that allow communication between multiple devices. This calls for the development of user-profiles performing intelligent routing capabilities, which help in determining the device configurations and communicate accordingly.

The following characteristics are to be determined while designing mobile applications :

- Form factor or viewing area size
- Browser capabilities, languages supported
- Available input methods
- Text coverage, graphics support etc (display performance)
- Expandability options and slots (peripherals and accessories)
- Push technology support
- Ruggedness
- Information storage capability; adding new data to devices (Memory)
- Device performance of calculations and logic (Processor speed)

### Networks

Wireless networks have their own standards in terms of connectivity and functionality aspects. These include gateways, towers and access links that are dependent on an organization's use of network protocols, application logic and device access. The characteristics for wireless networks are defined in terms of :

- Support to open Internet standards and protocols
- Extent of wireless networks coverage
- Cost factor
- Uniform / spotty coverage
- Data transmission speed, security concerns
- Time taken to service requests
- Authentication capabilities

## Applications

While the wireless device market is constantly evolving in areas of field service and sales force automation, vendors are extending their applications to support these new devices. In this scenario, a mobile application architecture confined to a single standard or device will experience difficulty in adapting to new technology advancements. Essentially, the mobile application architecture should be robust and flexible in providing user-friendly mobile solutions that are secure, reliable, scalable and manageable. The different characteristics to be considered for mobile applications are:

- Support interoperable standards; adapt to new technology advances
- Address bandwidth constraints
- Coverage fluctuations, also known as spotty coverage
- Rapid development of mobile applications
- Extent of separation between presentation and business logic in delivering wireless data access

## Approaches to Mobile Applications Development

Organizations strive to meet the mobile consumer market demands with multi-access capabilities, improved wireless networks, enhanced data exchange standards (such as XML) and progressive application development procedures. With varied device capabilities and limited bandwidth, organizations need to separate mobile applications for data delivery between business logic and presentation.

The adoption of mobile practice involves any one of the following methods:

- 1) **Building new applications:** This approach allows using best-of-breed technology to build mobile applications from scratch. Challenges include manpower with the required skill sets, difficulty in “future proofing” and costly involving maintenance of data across multiple channels.
- 2.) **Extending legacy applications:** Leveraging the existing legacy systems, this approach embraces wireless capabilities using XML, Java, integration middleware etc, reducing deployment costs.

The two core approaches to mobile applications development are:

- Thin Client or Browser-based application Development
- Thick Client Application Development

## Thin Client Application Development

The first wave of mobile application development known as Thin Client or browser-based, uses markup languages such as WML or HTML for content and interactions (uses HTTP to communicate with a server). To provide solutions for managing an extensive variety of markup languages, techniques such as Transcoding and Transformations are in existence. Using Transcoding software, the existing Web page content (i.e., HTML) is converted into an alternate markup language such as WML, c-HTML etc. Also known as "Screen Scrapers", this software reformats the web page into parts to be used by specific mobile devices like cell phones, Palm handheld devices etc (with variations in display size and browser specification.)

### Pros:

- Leverages existing HTML web pages
- Easy to create thin-client applications
- Quickens development time

### Cons:

- Tightly integrated presentation and application logic increases maintenance costs
- Incapable of handling rich graphics in a WAP device
- Irrelevant HTML page parts reduce device speed and performance
- Personalize content specific to each device – increased cost and time
- Unreliable data connectivity, complex development cycles

Unnecessary transmission of Presentation tags - increased time and cost

## Thick Client Application Development

The Thick Client approach is broadly classified into three categories as:

- a) Native application development
- b) Smart Client application development
- c) Using Mobile application framework

### a) Native Application Development

Native applications are developed to address specific operating systems (OS) such as Palm OS, Windows CE etc using C or C++. The native solutions tightly integrate with corporate systems, which can be either directly from a native application or through an application server. Here, the native solution usually splits into a proprietary application for each device supported. Though capable of working offline, these applications are entirely dependent on the operating system of the device.

### Pros:

- Takes full advantage of OS features such as hardware management and interaction between the device and the application
- Network throughput optimized
- Native look and feel
- Presentation tailored to form factor

### Cons:

- Portability to other OS and devices is limited and requires major changes  
Doesn't adapt to generic approaches to development like database synchronization
- Extended development cycles

## b) Smart Client Application development using J2ME

This approach of application development utilizes a single application suite to control business and presentation logic, and delivers content specific to each mobile device.

For example, J2ME (Java 2 Mobile Edition) application automatically adapts to device configurations by using the device profile and configuration settings. These applications offer rich user interfaces that are not limited by the capabilities of WAP browsers.

The use of Java 2 platform overcomes the resource constraints of devices, thereby, allowing the best of both thin and native clients. This creates a new set of mobile applications known as Smart Clients.

### Pros:

- Supports standalone, offline applications
- Distributed processing for better load
- Efficient bandwidth usage/balancing
- Enhanced services such as end-to-end security and data synchronization

## Cons:

- Security not currently native to the platform
- User interface API is primitive
- Current MIDP specifications do not support listening and needs extension to enable listening for server push

## c) Development Using Mobile Application Framework

The mobile application framework, also known as Wireless Middleware (built on open-standards like Java etc), provides a development platform that extends an organization's mission critical applications (data delivery in real-time) onto multiple devices and channels. The platform tightly integrates with an enterprise application server, and any changes on the enterprise application are directly updated on the client device using user-interface (GUI) and business logic. This offers the advantages of performing a twin role of being a thin-client (server-controlled updates) and thick-client (offering reliability, usability and offline capabilities).

Unlike generic platforms like J2ME etc, wherein, organizations need to spend a lot of time in developing generic components (also overcoming the challenges faced by developers); mobile application frameworks with their set of semi-built components will jump start the mobile solution development, thus allowing an organization to concentrate on the business problem at hand, rather than dealing with data-packets etc.

The use of middleware provides the ability to work offline or execute application logic on the client. This is possible with the use of smart client devices, wherein, a small mobile client application sits on the device. Middleware provides a set of application programming interfaces (APIs) across all networks and operating systems offering flexibility and scalability aspects.

## Pros:

- Offline usage, strong security, deployment flexibility etc
- Seamless integration of enterprise systems with the existing business applications like SCM, ERP etc
- Open-standards platform supports wireless data, voice and web service channels

- Extends existing applications for multiple devices
- Utilizes full advantage of device features for management and interactions
- Highly customizable and scalable; reduces development time and costs
- Provides out-of-the-box functionalities of device management, application management, synchronization, rules-based development and generic data synchronization to multiple devices
- Provides smart-connect mode operation, shielding applications from having to worry about connectivity

### Cons:

- Extensive interface and management support requirements
- Application developers require high level expertise in integrating legacy systems

## Mobile Application Development Approaches

### - A Comparative Matrix

Features / Functionalities	Thin Client Applications	Native Applications	Smart Client Applications	Mobile Application Framework / Mobile Middleware
Examples of Mobile Application development	WML / HDML Applications, Transcoding etc.	Writing custom applications on Palm, Pocket PC or other devices using C or C++	EVB, Brew, EVC++, J2ME etc.	Pervagus, iConverse, Sybase etc.
Diverse Platforms	Limited to few platforms	Applications are increasingly becoming multi-platform	Restricted to some platforms	Widely applicable to various platforms
Security	Vulnerable at gateway	Secure but not integrated with other components in the platform	Secure but not integrated with other components in the platform	Highly secure as the integrated Security management is in-built
Usage of existing Infrastructure / Application	High - Requires WAP-enabled browsers	Moderate - Applications leverage the existing infrastructure	Low - To install new infrastructure	Very High - As Middleware offers plug-in capability
User friendliness (Look-and-Feel)	Low - Applications largely consist of text	Moderate - Depends on the specific device form factor	High - Customized to form factor	High - Few keystrokes and form factor customization
Time for Deployment	Rapid	Extensive	Extensive	Rapid

Features / Functionalities	Thin Client Applications	Native Applications	Smart Client Applications	Mobile Application Framework / Mobile Middleware
Total Cost of Ownership	Low	High	High	Moderate
Return on Investment (ROI) realization	Realizes at a medium pace	Applications are increasingly becoming multi-platform	Restricted to some platforms	Widely applicable to various platforms
Scalability	Cannot be ramped up drastically	Can be extended to predefined levels	Scalable to certain predefined levels	Highly scalable
Maintenance Effort	High	Low	Moderate	Low
Personalization	Moderate	Low	Moderate	High - Uses Profile Management
Data access speed (from local database in the device)	Low and the database operations are few	Moderate	Online-Offline	Optimized for data manipulation operations
Graphics usage in Presentation	Less (emphasis is more on text)	Less	Moderate	High
Network Connectivity	Unreliable	Reliable	Reliable	Highly reliable
Online/Offline Operations	Mostly Online	Mostly Offline	Online-Offline	Online-Offline-Spotty connectivity
Integration with other systems/ applications	Low	Moderate	Moderate	High
Device/Carrier/ Enterprise Application Independence	Poor	Good	Good	Very Good
Support for multiple devices and networks	Less	Less	Less (slowly improving)	Supports multiple devices and networks

## Why Mobile Application Framework?

The mobile application framework / middleware is witnessing continuous growth. Demanding agility in the applications space, the market is maturing from generic middleware into a new generation of multi-channel middleware paradigm. Application vendors are finding means to diversify access to multiple devices and networks with intermittent connectivity.

New techniques are falling in place such as the provision of standardized synchronization mechanisms to enable offline communication across heterogeneous platforms. However, the success is largely dependent on the underlying principle of the business process being supported in an enterprise.

## Advantages for Mobile Developers

Middleware overcomes the challenges faced by mobile application developers while implementing disparate applications such as:

- Multiple mobile-device features such as OS, form factors, language-usage etc.
- Diverse nature of wireless network technologies and protocols
- Intermittent connectivity spreads across a wide range of enterprise systems like desktop PCs, servers etc.
- Adopts multiple application development approaches such as “always-on” thin-client, “off-line” thick-client etc.

The first wave of Mobile Application Framework surfaced mainly to support Web applications. The framework handled only a Web browser-processing HTML through a thin-client device. Realizing the need for multi-channel access with rich user-interface capabilities, the current wave extends into product architectures to support multi-client devices.

Presently, most application framework vendors are turning to provide wireless application gateways (WAGs) and other mobile-enabling features such as synchronization, event notification etc.

A typical mobile environment scenario, wherein, middleware plays a key role involves:

**Optimized Communication:** Operates through a common set of APIs making it easy to develop and deploy mobile applications that optimize communication over IP-based networks.

**Security:** Encryption/decryption functionalities using PKI technologies guarantee secure data exchange over wireless networks.

**Bandwidth:** Features such as compression and transport optimization help in effective data transmission.

**Industry-standard protocols:** Optimal utilization of TCP/IP protocol to reduce airtime expenses, and to support wireless communications.

**Compatibility:** Offers compatibility between industry-standard TCP/IP protocols and all wireless networks through its common set of APIs.

**Spotty coverage:** Automatically adjusts to spotty/fluctuating coverage by minimizing connection loss (possible by slowing down and speeding up), as and when required.

**Offline capability:** One of the key features of middleware is its ability to support offline functions. Here, the middleware assigns some of the processing capabilities on the wireless device so that the user can be connected for a shorter amount of time and manipulate the data on the wireless device while offline.

“Push” Technology: The core functionality to mobile applications is its ability to “Push” data to mobile users’ devices as they operate in disconnected and out of coverage modes. This is possible in different ways such as connected, disconnected and prior filtering process capabilities.

Device agnosticism: Allows multiple devices to communicate without the need to know device-specific coding.

Network Connectivity: Facilitates increased geographic coverage across multiple mobile devices that are independent of network providers and carrier technologies.

Personalization: Middleware solutions offer personalization through scripting, markup language, user-interface etc. This includes message alerts, the look-and-feel aspect, historical usage patterns etc.

Application systems integration: Tight integration of traditional databases, legacy systems and voice systems with mobile applications. XML is proven to be the widely accepted choice to operate across multiple systems.

Manageability: As part of the middleware suite, the management software handles clients (i.e., devices) via multiple networks, by updating and synchronizing information from enterprise back-end systems.

Session management: Maintains session integrity across multiple devices and channels by embracing the concept of mobility, transporting a session from one device to another etc.

Database Synchronization: Offers connectivity from any back-end system / database to any device that executes only those specified data transfers. Connectivity is also extended within existing enterprise applications like SCM, ERP etc. Apart from this, the wireless communication component in middleware provides access to other peripheral devices like camera, POS, GPS, barcode scanners, fax etc.

Business Rules: Applying business logic and rules, middleware leverages the existing work and code that simplifies workflow, eliminates duplication effort, and saves time and money.

## The Next Generation Pervagus Mobile Framework (PMF)

Pervagus, the end-to-end mobile solutions provider, offers a comprehensive Next Generation Pervagus Mobile Framework (PMF) for wireless applications development and deployment that supports multiple devices and networks across enterprises.

Built from the ground up on future-proof technologies, PMF (also a Wireless Middleware), utilizes open-standards like Java 2 Micro Edition (J2ME) and Java 2 Enterprise Edition (J2EE) to deploy applications onto mobile devices that increase portability to other Palm OS devices like mobile phones, Windows CE based devices etc.

PMF's Profile Manager offers high level of personalization at device and user levels. Profile Manager maintains a set of objects that can store a cross section of user preferences (e.g., personal user interface settings) and device features (e.g., color image support). Users can create a "Persona" which is a single concrete characterization of their preferences and profiles.

PMF is one among the few providers that support applications for both thick-client (offline capability) and thin-client (browser-based) multiple devices. In a thick-client based development, a small footprint software package sits on the device. In a disconnected mode, PMF facilitates automatic 2-way data synchronization through store-and-forward message queuing and push messaging.

## Conclusion

mBusiness is witnessing phenomenal growth as new wireless technologies offering improved bandwidth, rapid transmission and cost-effective mobile devices hit the marketplace.

Selecting the right application development approach is imperative to create wireless solutions that address all of the challenges being experienced in mobile space. The mobile application framework / middleware technologies coupled with a small footprint software embedded on the client device serves as THE perfect combination utilizing the features of the device in its entirety.

# Glossary

A glossary of wireless and mobile terms used in this whitepaper:

- HTML (Hyper Text Markup Language) - An authoring markup language used to create Web pages and hyperlinks.
- XML (eXtensible Markup Language) - XML is emerging as the global method of choice for creating web content. It allows for industry-specific language definitions and the ability to operate over multiple devices and network platforms.
- WAP (Wireless Application Protocol) - WAP is a specification for a set of communication protocols to standardize the way wireless devices (such as cellular telephones and radio transceivers) can be used for Internet access, including e-mail, the World Wide Web, newsgroups, and Internet Relay Chat (IRC).
- XML (Extensible Markup Language) - XML is a flexible way to create common information formats and share both the format and the data on the Internet, intranets etc.
- Java - A platform-independent programming language that abstracts data on byte codes so that the same code runs on any operating system.
- J2ME (Java 2 Platform, Micro Edition) - A technology that allows programmers to use Java programming language and related tools to develop programs for mobile wireless information devices such as cellular phones and personal digital assistants (PDAs).
- WML (Wireless Markup Language) - WML is a markup language that allows the text portions of Web pages to be presented on cellular phones and personal digital assistants (PDAs) via wireless access.
- HTTP (Hyper Text Transfer Protocol) - The Web server and the client browser use the protocol to communicate and transfer data across the Internet in the form of requests and responses.
- API (application programming interface) - API is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another
- PKI (Public Key Infrastructure) - PKI is a specification that enables users of a basically unsecure public network such as the Internet to securely and privately exchange data and money. This is possible through the use of a private cryptographic key pair that is obtained and shared via a trusted authority.

## About Pervagus, Inc.

Pervagus, Inc. has extensive experience in deploying end -to-end mobile services for the Transportation industry, Field Service, Healthcare, Financial Services etc. Our domain knowledge and technical proficiency set on par with industry standards offer highly tailored solutions that befit your current and future requirements.

**For more information, please contact :**

Pervagus, Inc, 10, Corporate Place South, 2nd Floor, Piscataway, NJ 08854 (USA),  
Ph: (732) 465 - 9800, Fax: (732) 981 - 1414, E-mail : [info@pervagus.com](mailto:info@pervagus.com)