



# Developing Mobile Applications

*Eximius mBaze platform for effective mobile application development.*

**White Paper**

---

## **Abstract**

Eximius takes mobile application development seriously. Currently, Eximius offers a data-centric communication middleware for Java 2 Mobile Edition application development on mobile phones. In the coming months, Eximius will enhance the middleware with a new framework targeted at the mobile developer. Eximius will continue to innovate these mobile development tools and integrate them into the core Eximius® mBaze platform so that eventually there will be one core development environment to meet all your mobile application development needs.

This white paper will help IT decision makers and developers to understand the current and future landscape of Eximius mobile application development tools and help developers decide which offerings are most suitable for their needs.

© 2005 Eximius Corporation. All rights reserved.

To provide feedback on this white paper, please send e-mail to [info@EximiusMobility.com](mailto:info@EximiusMobility.com).

The information contained in this document represents the current view of Eximius Corporation on the issues discussed as of the date of publication. Because Eximius must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Eximius, and Eximius cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. EXIMIUS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Eximius, mBaze are registered trademarks of Eximius Corporation.

Other product or company names mentioned herein may be the trademarks of their respective owners.

Eximius Corporation • Toronto, ON • Canada

---

## CONTENTS

MOBILE APPLICATION PROGRAMMING	1
<b>Introduction</b>	<b>1</b>
<b>Welcome to the World of Mobile Development!</b>	<b>1</b>
<b>What are the Difficulties of Developing Mobile Applications?</b>	<b>1</b>
<b>Framework vs. Software Development Kit (SDK)</b>	<b>2</b>
Defining a Software Development Kit	2
Defining a Framework	2
<b>How Does mBaze Fit In?</b>	<b>2</b>
THE EXPLODING WORLD OF MOBILE SERVICES	3
<b>Interest in Mobile Services is Growing</b>	<b>3</b>
<b>Personal Information Management</b>	<b>3</b>
<b>New Application Frontiers</b>	<b>3</b>
<b>Content is in the Network</b>	<b>4</b>
DIFFICULTIES OF DEVELOPING MOBILE SERVICES	5
<b>Mobile Service Requirements</b>	<b>5</b>
<b>Limitations of Current Technologies</b>	<b>5</b>
WAP is Not Good Enough	5
SMS is Not Good Enough	6
Java (J2ME) is Almost There	6
<b>mBaze to the Rescue</b>	<b>7</b>
MBAZE MIDDLEWARE	8
<b>Data Centric Communication</b>	<b>8</b>
<b>An Aside: Publish/Subscribe</b>	<b>8</b>
<b>Features</b>	<b>9</b>
MBAZE FRAMEWORK	11
<b>Reinventing the Wheel... and Making it a Square This Time!</b>	<b>11</b>
<b>InfoGrazer</b>	<b>12</b>
<b>Lumina</b>	<b>12</b>
When to Use Eximius Lumina	13
<b>Eximius mBaze Framework vs. J2ME</b>	<b>13</b>
<b>Bookstore Service: an Example Application</b>	<b>13</b>
SUMMARY	15

## Introduction

This white paper will explain the Eximius mBaze mobile application platform that is available to you, why you should use the platform, and what you, a developer of mobile applications, need to get started with mobile development using the Eximius mBaze platform.

This paper will aid you in your decision-making when deciding how to develop applications for a particular mobile device. Along with an overview about the platform now at your disposal, you will also find useful information comparing platform features and how each affects your application capabilities.

Eximius has a strong commitment to you, the mobile application developer, and will continue to provide you an excellent platform that makes developing applications for mobile solutions an easy task.

Java 2 Micro Edition (J2ME) is one of the most successful software development kits for mobile phones today. That software development kit receives special focus in this white paper.

## Welcome to the World of Mobile Development!

Eximius offers a platform specifically for the needs of mobile developers. Although the current landscape of mobile development tools consists of an assortment of offerings, Eximius is the first to offer a framework for systematic development of mobile applications. Eximius will continue to innovate and integrate new functionality into the core Eximius mBaze development platform, so that eventually, developers will have one development environment for all their needs.

Most of the rules and guidelines for developing Web applications for Enterprise Application Servers also apply generally to mobile phones based on J2ME and mBaze. However, there are a couple of important differences between Web application development and mobile phone development.

## What are the Difficulties of Developing Mobile Applications?

Any given mobile phone can be very different and unique from the next even though both have J2ME. Even similar devices from the same manufacturer can be very different because of varying hardware capabilities.

Therefore, a very important part of development for mobile phones is to have the means to isolate your application from these variations and to be able to configure your application easily for different devices.

---

## Framework vs. Software Development Kit (SDK)

The words “software development kit” and “framework” appear often in this white paper. There is a fundamental difference between those two terms. Since they are so important, here is a brief explanation of what each means.

### Defining a Software Development Kit

You use a software development kit to develop for a particular device such as a mobile phone. Even notebook PCs and desktop PCs generally require an SDK to build software that will run on those particular computers. Most modern development tools provide the SDK for you.

You might be surprised, but generally you do not develop using an SDK only. The SDK alone without a framework isn't enough. When developers build applications they always develop using a framework.

Existing popular frameworks are:

- Apache Jakarta Struts
- Spring
- J2EE
- Microsoft .NET

### Defining a Framework

A framework is a set of tools and application architectures that helps you build you applications in a systematic way. If your application requires fast time to market, rapid application development, is user interface-centric, or needs to run on different mobile phones, a framework helps achieve these goals.

### How Does mBaze Fit In?

Eximius has introduced the mBaze Framework and the new concept of personal application servers. Personal application servers allow for easy development of a new generation of applications for mobile phones.

### Interest in Mobile Services is Growing

As mobile phones are increasingly becoming more capable, the opportunity for providing new and exiting services is growing. The number of users of mobile applications continues to grow indicating that people are increasingly more willing to use applications on mobile phones.

### Personal Information Management

The common themes for all of these applications are personal information management and communication. Many phones come with Personal Information Management (PIM) software built-in. Commonly, these PIM applications allow users to read e-mail, and they provide calendaring, task management and contact management. A user is expected to synchronize her mobile PIM with her desktop PIM in order to keep information on the mobile device up to date. However, this is not always possible, either because of synchronization software incompatibilities, but far more frequently, because synchronization is simply not convenient. Users forget to do it, and if they are constantly on the move, they do not have time to do it. Consequently, although users have an interest in using mobile PIMs, they usually end up abandoning them due to synchronization being inconvenient.

Moreover, built-in PIMs make it very difficult to share information, particularly scheduling, which is a basic prerequisite for enterprise-level PIMs. This is why hand-held devices such as RIM's Blackberry are very popular; they take advantage of always-on connections, and offer the convenience of an up-to-date synchronized at any time.

J2ME capable phone open up the possibility of having the same kind functionality on any mobile phone. Moreover, with J2ME it becomes possible to have applications that can better interoperate and that can work with any desktop PIM by being more open then the built-in PIMs.

### New Application Frontiers

New kinds of applications become possible with J2ME capable phones that can benefit from always on connections, such as exchange of business cards, real-time surveys, up-to-the-minute bus schedules, or browsing the latest news headlines.

More advanced applications also become possible such as location-based services for tourist attraction. When at an unknown location, users perceive the mobile phone as a quick and familiar way of obtaining information.

Mobile ticketing is growing steadily in Europe and Asia. With programmable mobile phones it becomes possible to buy and use e-tickets for a variety of services very conveniently. What has been, so far, primarily a domain of airfare e-tickets now

---

becomes possible for other kinds of services as well.

Entertainment involving mobile dating and games such as tag are particularly interesting for youth market.

### Content is in the Network

Common to all of these applications is that content is dynamic and is located in the network. J2ME technology provides a good foundation for developing these types of applications, but the development process is difficult, slow and error prone.

The Eximius mBaze platform addresses these issues helping you develop applications that are more convenient, have pleasing user interfaces and allow applications to easily take advantage of an always-on connection of a mobile phone.

---

## DIFFICULTIES OF DEVELOPING MOBILE SERVICES

Before describing the Eximius mBaze platform, you need to understand the limitations of existing tools and technologies in developing the next generation of mobile services. In this section we enumerate the requirements of these exciting new services, and explain how current technologies do not meet these requirements.

### Mobile Service Requirements

The next generation of mobile services will bring desktop-like usability to mobile devices. Current applications built using SMS or WAP browsers simply do not meet the usability levels that users have come to expect from other computing devices.

New mobile applications will require, among other things, the requirements outlined in the table below.

Requirement	Description
<b>Local processing capabilities</b>	Transferring more processing to the phone allows for more responsive and intelligent user interfaces.
<b>Local storage capabilities</b>	Similar to local processing, storing information on the phone allows for more responsive applications.
<b>Asynchronous communication</b>	Mobile applications today typically use a request/response model that requires constant attention from the user. Allowing applications to be notified of interesting events opens the door to a whole new class of mobile applications
<b>Secure communication</b>	Sensitive data, such as financial transactions must be transferred in a secure manner.
<b>Aesthetic, convenient user interface</b>	Usability and aesthetics are important in mobile devices with limited display and input capabilities.

### Limitations of Current Technologies

The most common technologies currently used to develop mobile services—WAP and SMS—cannot be used to develop mobile services with the above requirements.

#### WAP is Not Good Enough

Wireless Application Protocol (WAP) is a standard that allows mobile phones to access services in a similar manner as a Web browser. However, WAP fails to meet any of requirements above for the next generation of mobile services.

WAP interfaces are notorious for being aesthetically unattractive and support only the most basic user interactions. WAP also does not provide any facilities for local processing or local storage on mobile devices. Instead, WAP can only display simple “pages” of information, with each user interaction requiring communication with a

---

server. This communication is both expensive and time consuming, and impedes the responsiveness of the user interface.

WAP also does not provide any clean mechanism for asynchronous communication with a mobile application. While the WAP Push specification tries to address this limitation, its implementation and availability leaves a lot to be desired. For instance, messages sent with WAP Push require attention from the user who must decide how to act on the message. It is instead desirable that applications first process the message and decide how to notify the user (if at all). In addition, the availability of WAP Push is not guaranteed as it has not been deployed by all cellular service providers.

Furthermore, frequently WAP communication is not secure. WAP messages are sent without any encryption and there are no authentication mechanisms provided.

#### SMS is Not Good Enough

Short Message Service (SMS) allows mobile phones to send and receive short text messages. SMS was designed for human-to-human communication and is simply not appropriate for developing mobile services. As with WAP, SMS also fails to meet the requirements for mobile services.

SMS user interfaces are extremely limited, restricted to a short unformatted text stream. The interaction is even more limited than WAP; the only action that can be performed on an SMS message is reply to it. In addition, SMS messages are limited in size to about 160 characters.

While SMS does allow for asynchronous communication to a mobile device, SMS messages force user attention to process, instead of allowing an application to process the data first.

As well, SMS messages are not secure; they provide neither encryption nor authentication.

#### Java (J2ME) is Almost There

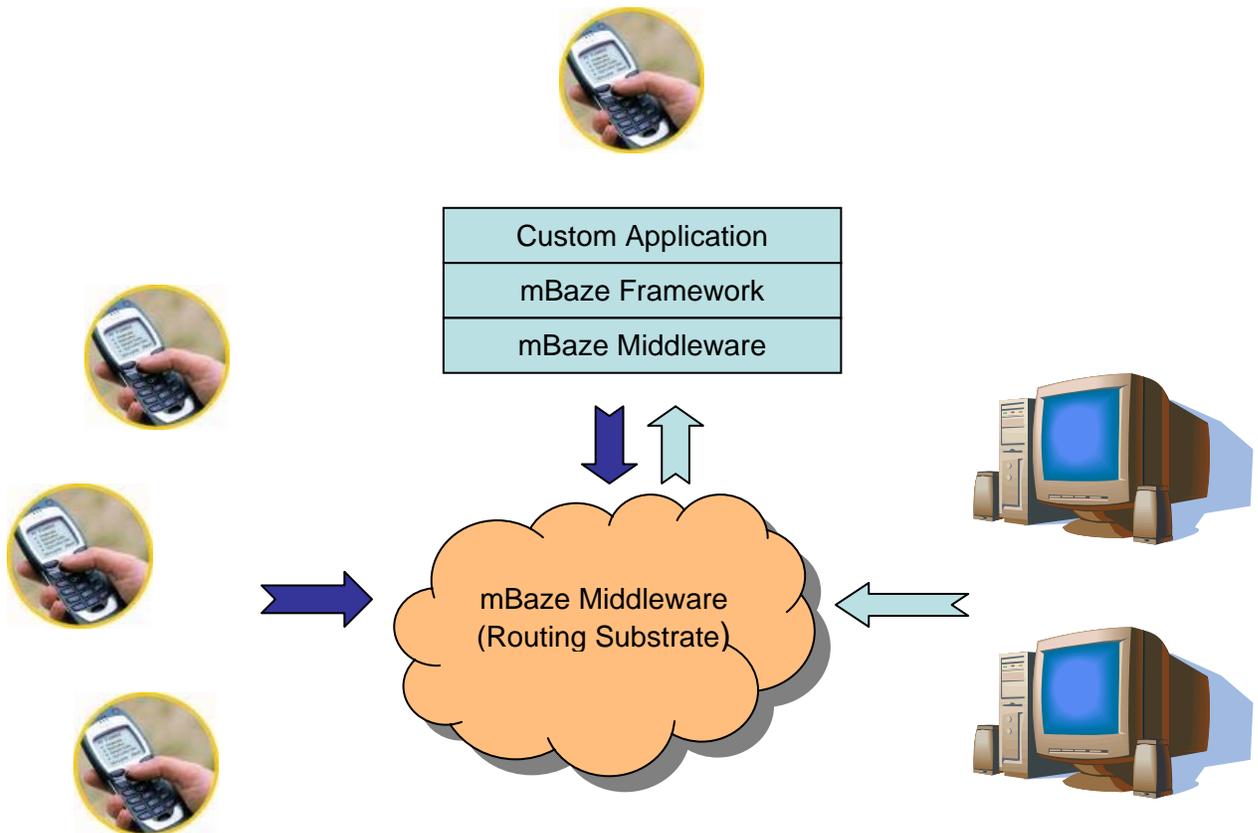
Increasingly, mobile devices support applications written using the J2ME Java standard. Java and J2ME address the requirements for next generation mobile services. However, as we will see, while J2ME makes it possible to develop such services, it is difficult to develop these applications.

Java places no limitations on the user interface of mobile applications, other than those imposed by the physical hardware itself. Java games on mobile phones illustrate the rich user interfaces and interaction methods supported by this technology. In addition, Java applications can perform local processing and have access to persistent storage on the mobile device.

Furthermore, Java allows applications to develop custom communication protocols. Protocols can be developed that are as secure as needed. Unfortunately, the J2ME specification does not support asynchronous communication. However, as described below, the mBaze framework works around these limitations and provides facilities to support asynchronous communication.

### mBaze to the Rescue

The Eximius mBaze platform helps you develop advanced mobile services. The mBaze platform consists of **mBaze Middleware**, a routing substrate designed for the easy deployment of scalable mobile services, and **mBaze Framework**, a set of development tools and techniques to help develop mobile services quickly and with minimal effort. The figure below illustrates a high level view of the mBaze platform. mBaze Middleware and mBaze Framework are described in more detail in the following sections.



mBaze Middleware facilitates the development and deployment of mobile services in a heterogeneous network. This section describes the communication primitives provided by the mBaze middleware.

### Data Centric Communication

mBaze Middleware is based on a revolutionary paradigm that allows for faster and easier development of more powerful distributed applications.

Most distributed applications are based on a point-to-point message sending primitive. The limitations of this are that it requires the sender to know the address of the receiver, and makes it both difficult and expensive to send the same message to multiple recipients.

Instead, mBaze Middleware provides a decoupled, data centric communication primitive in which receivers simply specify what they are interested in and senders indicate what they are sending, and mBaze takes care of routing messages to all interested receivers based on the content of the message. This deceptively simple primitive is a powerful feature that makes it much easier to develop complex distributed service-oriented applications quickly.

### An Aside: Publish/Subscribe

The mBaze Middleware's data centric communication is based on the publish/subscribe communication model. In the publish/subscribe model, the sender of a message is referred to as a publisher, and the receiver is called a subscriber. In addition, one or more brokers act as mediators to route messages from publishers to subscribers.

In traditional distributed applications, a sender sends a message directly to a receiver. In the publish/subscribe model, receivers (subscribers) first specify the messages they are interested in receiving by sending a subscription message to the broker. Senders (publishers) send messages in the form of a publication to the broker. The broker examines the content of the publication, finds subscriptions that match the publication (i.e. subscriptions whose interest is satisfied by the publication), and routes the publication to the interested subscriber(s).

A key advantage of the publish/subscribe model is the decoupling of senders and receivers. Notice that in the above description, the publishers and subscribers are not aware of each other; the brokers mediate all communication between them. This allows the development of the communicating parties in a service to be developed and deployed independently. It also allows the senders and receivers to be transparently deployed on drastically different platforms, from limited cell phones to powerful Unix servers. For example, a server can communicate with clients running on a cell phone or a desktop computer without any modification whatsoever.

---

Furthermore, since messages are addressed by their content instead of fixed IP addresses, it is easy to add or change the set of senders or receivers. For example, a client does not need to know the addresses of the set of servers and have to decide on the server it should communicate with for a given protocol. Instead, it simply publishes its message, and the broker will determine the server(s) that should receive the message.

## Features

mBaze Middleware provides many benefits to application developers. The table below outlines those benefits while a later section will discuss the benefits of mBaze Framework.

mBaze Feature	Description
<b>Based on well-known application server model</b>	Significantly decreases learning curve for developing on mobile devices by mimicking a software development environment that most developers are already familiar with.
<b>Decoupling</b>	Applications developed using mBaze Middleware allow communicating parties to be decoupled and hence be developed and deployed independently.
<b>Data addressing</b>	Applications no longer need to worry about issues related to finding and keeping track of the addresses of nodes in the network. Instead, the receivers of a message are implicitly determined from the content of the message. This greatly simplifies application protocol development.
<b>Incremental deployment</b>	The mBaze Middleware substrate allows new services to be deployed incrementally over the same messaging infrastructure.
<b>Simplicity</b>	The messaging model provided by mBaze is easy to understand and facilitates rapid application development.
<b>Lightweight</b>	The mBaze Middleware's interface is lightweight and can be used by applications on limited mobile devices, yet is powerful enough to develop complex services.
<b>Ease of integration</b>	The messaging model makes it easy to integrate with existing services built on a messaging model such as existing enterprise applications that follow a Web Services architecture or the Java Messaging Service (JMS) specification.
<b>Security</b>	mBaze Middleware provides secure communication primitives including password-based authentication and strong encryption.
<b>Reliability</b>	mBaze Middleware substrate is built on leading research on reliable distributed systems.
<b>Asynchronous communication</b>	mBaze Middleware supports asynchronous communication to mobile devices even when they are behind a firewall.
<b>Centralized administration</b>	mBaze Middleware allows the centralized configuration and monitoring of the distributed mBaze Middleware substrate.

---

## MBAZE FRAMEWORK

The mBaze Framework is software that helps you develop advanced mobile services quickly and easily. mBaze is ideal for the following scenarios:

- Applications that take advantage of XML Web services.
- Applications that must work well in either a connected or disconnected environment.
- Developing applications using Java.
- When the same tools need to be used for desktop, server, and device programming.
- Rapidly developed mobile applications.
- Software that provides a reliable and secure environment.
- Applications that may be moved to different phones.

### Reinventing the Wheel... and Making it a Square This Time!

It is possible to develop great applications using only J2ME. But only with mBaze Framework, is it possible to develop great applications quickly and easily.

mBaze Framework is a reusable design for mobile applications. It is based on the well known and tried concept of application servers, but adapted for the resource constrained mobile devices.

mBaze Framework provides an alternative to reinventing the wheel for each new mobile application. It provides an extensible software architecture that has been shown in practice to be very successful. mBaze Framework frees you to work on building your core application functionality instead of spending time and resources on designing and developing the supporting infrastructure and having to worry whether your design is good.

By choosing mBaze framework your applications can immediately start reaping the benefits of good design practices for mobile devices.

mBaze Feature	Benefit
<b>Based on the well-known application server model</b>	Significantly decreases learning curve for developing on mobile devices by mimicking a software development environment that most developers are already familiar with.
<b>Designed on sound software engineering principles</b>	Application server architecture has proved itself in practice. By choosing mBaze Framework, you can capitalize on the knowledge and experience you gain by developing on application servers.
<b>Lets you concentrate on application-specific functionality</b>	With mBaze, you can concentrate on developing you application. The framework takes care of the tedious and error prone plumbing for you.
<b>Enhances application portability and reuse</b>	Because all differences between devices are captured in mBaze framework, the applications you develop are easily portable to all J2ME capable devices.
<b>Clean separation between user interface and application logic</b>	By developing using mBaze framework, your application logic can stay UI-independent. This not only increases the portability of your application but also significantly reduces the complexity and time to develop.

## InfoGrazer

InfoGrazer is a lightweight application server. The integral part of InfoGrazer is the grazelet container. Grazelets are similar to Java Servlets but whose primary application area is very constrained environments such as those on mobile phones. InfoGrazer relies on mBaze Middleware to provide network connectivity. A lightweight implementation of HTTP is provided as a familiar communication mechanism. In addition, grazelets have direct access to mBaze Middleware for efficient multicast and unicast communication.

InfoGrazer also includes a lightweight and efficient persistence for Java objects. This persistence support can save and restore objects locally (on the device) or remotely (on the network) in a way that is transparent to the application.

## Lumina

Eximius Lumina complements InfoGrazer. When applications are developed using InfoGrazer, some way of interacting with the user is needed. Alternatively, using Lumina, the majority of the UI handling is already done for you and all you have to do is describe you application's UI at the application server and Lumina will take care of UI display and interaction logic for you.

Eximius Lumina empowers developers using InfoGrazer to build a single mobile application that automatically adapts its display for a wide variety of mobile phones.

Lumina contains a set of controls that intelligently and asynchronously renders UI descriptions for different types of mobile phones and creates a UI based on J2ME High-level UI components. Lumina does not require any special handling by the

---

application besides the UI description.

In addition to adaptive J2ME High-level UI rendering, Lumina provides a rich framework for customization and extensibility. This flexibility enables developers to easily extend the controls to take advantage of new and different devices that will enter the market. In addition to being highly adaptable for future devices, Lumina enables developers to control the rendering for a specific device or class of devices within the same programming model.

Moreover, the device extensibility model also allows developers to easily add support for new devices without re-authoring their mobile applications. Lumina supports the control extensibility model with which developers can build user controls, composite controls, and brand new mobile controls to be used in mobile software projects.

When to Use Eximius Lumina

Use Lumina if you want to create mobile applications that target the broadest range of mobile devices.

### Eximius mBaze Framework vs. J2ME

The following comparison chart will give you a good reference on the capabilities of the different tools and what they support.

	<b>J2ME</b>	<b>mBaze</b>
<b>Java</b>	X	X
<b>Message-based Communication</b>		X
<b>Scriptable UI</b>		X
<b>Works across many phones</b>	X	X
<b>Familiar Development Environment</b>		X
<b>Rapid Development</b>		X

### Bookstore Service: an Example Application

We briefly illustrate the ease of mobile application development with the mBaze platform. We look at a use case scenario for a location-based service application.

Your company is contracted to develop a mobile service for bookstores that allows the bookstores to inform their customers about newly arrived items. The major requirement imposed by the bookstores is that customers are only notified of items that are of interest to them. Otherwise, the bookstore is concerned that the customers perceive the service as spam. The potential user base for this application is large as several bookstore chains have expressed interest in participating in this service.

You decide that you want to implement this application using the mBaze platform. The application is composed of two main components: one for the mobile devices and the other for the bookstore.

The first component allows the customer to create a book profile, stored on their phone, to specify their interests. To keep the application simple and intuitive, you allow the users to express their interest using book title, author name or subject.

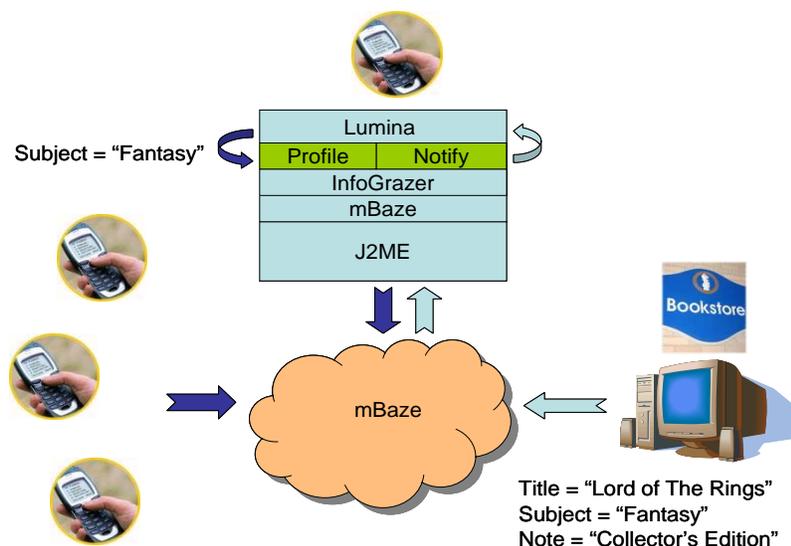
The second component allows bookstores to send notifications about new items that they receive. Each item is described in terms of title, author and subject. In addition, other information describing the characteristics of the book can be sent.

Developing the mobile component using mBaze Framework is simple since all you need to do is develop two grazelets. The first one collects book interests from the user, stores it locally using the mBaze persistence support and subscribes according to users profile using mBaze Middleware.

The second component simply receives notifications for new items and displays them to the user using Lumina.

You are done!

The example illustrates that developing applications for mobile phones does not need to be any harder than developing traditional applications. Hence, you can leverage your existing application development knowledge and experience for mobile phones.



---

## SUMMARY

Now you know more about the tools, technologies and techniques that you can rely on in your future mobile application projects, as well as what you can expect from Eximius in this area.

The Eximius mBaze platform, consisting of mBaze Middleware and mBaze Framework aid you in your various projects. Hopefully the information presented here will guide you further and make the framework selection process and your approach to mobile solutions and application development even easier.

If you have feedback about this document, or have questions about mobile application development please e-mail [info@EximiusMobility.com](mailto:info@EximiusMobility.com).