

Working Wireless Solutions



Strategy, Business Development and Marketing
for Wireless Businesses

Java- Wireless Application Strategies Among US Mobile Operators

Written by
Derek Kerton and
Bin Hu



Sponsored by
Bouygues Telecom

May 2003

THE **KERTON GROUP**
www.kerton.com

The KERTON Group
1970 Mazey Street
Milpitas, CA 95035 - USA
+1 408 935 8702 Tel
+1 707 202 2434 Fax
info@kerton.com

Table of Contents

1. EXECUTIVE SUMMARY 1

2. APPLICATION PLATFORMS FOR MOBILE PHONES 3

 2.1. J2ME 3

 2.2. BREW 9

 2.3. BUSINESS MODELS OVERVIEW 13

3. CARRIERS: 20

 3.1. SUMMARY 20

 3.2. VERIZON WIRELESS 21

 3.3. SPRINT PCS 23

 3.4. CINGULAR..... 26

 3.5. AT&T WIRELESS..... 27

 3.6. NEXTEL 29

 3.7. T-MOBILE 31

 3.8. MVNOS 32

4. HARDWARE 34

 4.1. OEMS 34

 4.2. TERMINAL TESTING & QA 34

 4.3. JAVA PERFORMANCE & ACCELERATION 36

5. CONTENT 37

 5.1. APPLICATION DEVELOPERS 37

 5.2. DEVELOPER COMMUNITIES 39

 5.3. OTHER STAKEHOLDERS 39

6. ADDENDA..... 41

 6.1. METHODOLOGY..... 41

 6.2. ABOUT THE KERTON GROUP..... 42

Exhibit A: J2ME-Enabled Terminals and specs

Exhibit B: Acronym Glossary

1. Executive Summary

Downloadable wireless applications seem to be the wireless data product the U.S. wireless telecommunications industry has been seeking. From the early indications, BREW and J2ME wireless applications appear to **have the potential to offset the problem of voice commoditization and declining ARPU**. Based on the publicly available results from Verizon Wireless' "Get It Now" BREW service, it appears downloadable content can have a significant impact on data revenues – Verizon has seen an approximately 50% increase in data revenues since launching last autumn. Among the downloadable wireless applications, **gaming has proven to be the most popular**. According to DataMonitor, one of every six mobile phone users, 21.6 M people played games on mobile phones in 2001. Datamonitor forecasts **93 M users will be playing games in 2006**. Most carriers have indicated the enterprise as the second target segment for applets after gaming.

There are two main technologies for downloadable content, Java (J2ME) and BREW. Although this report was to look at Java in the U.S.A., Java is much better understood with the context of BREW. Qualcomm owns **BREW, and exercises rigid control of the value chain** (and extract too much cash, too). **Java** is attractive to stakeholders who prefer an **open ecosystem with more flexibility, and more globally accepted standards**.

The first operators to launch wireless application download services did so in 3Q 2002 (**AT&T, Sprint, Verizon**), while **T-Mobile** and some of the smaller operators have not yet launched 8 months later. Of the carriers who have launched, **all of them are claiming better than expected results**. Verizon wireless has shared more performance data than any other carrier, and their results show the potential for all operators in this market. Of the 33M Verizon subscribers, 3.2M now have BREW handsets, and Verizon said they downloaded an average of almost 2M BREW applications per month in the first quarter. According to Qualcomm, **Fifty percent of subscribers with BREW enabled phones are active downloaders. The average active Verizon BREW customer spends \$7.50/per month on downloads**. Verizon's expects 8M installed terminals by the end of 2003, as 40% of terminals sold now are BREW-enabled. BREW will absolutely have a positive effect on the bottom line, and positive ROI measured in months.

Carrier	Applet Content Breakdown
Alltel	BREW, 44 games, 19 info
AT&T Wireless	Java, 162 games, 12 info
Cingular	Java, 24 games
Nextel Communications	Java, 113 games, 64 business tools
Sprint PCS	Java, 125 games, 20 non game
Verizon Wireless	BREW, 87 games, 28 info, 11 software tools

Source: [The Kerton Group](#)

AT&T Wireless is particularly interesting because they plan to diverge from the J2ME standard by **using DoCoMo's DoJa**, which has many proprietary extensions. AT&T says they cannot wait for the upcoming MIDP 2.0 terminals, and want to take advantage of the DoCoMo

relationship to offer the future functionality now. Unlike AT&T who had the advantage of the DoCoMo partnership, **Sprint PCS has assembled a very complete J2ME ecosystem from scratch.** Sprint, although on CDMA and a likely BREW candidate, appears to be attracted to the control and ownership a Java solution offers, and they had the resources to execute it well. However, their **pricing** is so beneficial to the consumer as to be **detrimental to Sprint.** Flat rate pricing limits the upside of an operator, and this was just the problem the U.S. carriers were trying to mitigate in the voice sector. **Nextel's** defining characteristic is the **64 enterprise Java MIDlets** that they have available to their subscribers. True to their segmentation strategy, Nextel provides highly specialized vertical with low mass-market appeal, but huge value for the vertical industries. Nextel's solution is painfully **lacking a direct-to-bill** function, requiring a credit card purchase.

Terminal OEMs are the ones most torn by the standards vs. proprietary decision. They make their earnings by having superior, differentiated handsets, but it is the differentiations that challenge standards and reduce mass-market economics. All OEMs must fight an internal battle between the drive to standardize, and the pressure to provide extra features. **There is, therefore, universal support for JTWI,** which raises the level of the minimum, but still allows plenty of room for proprietary extensions.

Developers are **agnostic about technologies.** They care about ROI. The developers are happy to develop in whatever language gets them access to the most customers. They are, therefore, **frustrated with the level of fragmentation,** and are eager for JTWI to ease the divergence. In the short term, they are **pleased with the real revenues BREW has earned** them, but in the long term, they are sanguine regarding the overall mass market economics of J2ME, and the easily accessible pool of Java development talent. There will be consolidation among development firms.

There are many other stakeholders in the downloadable apps ecosystem. There are **aggregators** who bring together multiple content providers, and have marketing channels with multiple carriers. There are **publishers** who aggregate, but also nurture development. There are hosting **ASPs**, Billing **ASPs**, third party **certification labs**, and more. In the **BREW model**, Qualcomm fills many of these roles, which makes the model **simpler and faster**, but also less flexible. In the **Java model**, a wealth of providers fills each role, making a **competitive, complicated ecosystem.**

While researching this report, it became apparent that many companies were holding back some announcements for the JavaOne conference in June, not the least of which was Sun Microsystems. **We expect Sun to come forward with a proposal for a complete Java ecosystem,** similar to what Qualcomm mandates for BREW. Of course, the Sun ecosystem will not be mandatory, and will be open. If Sun does this, it will be offering **Java community** exactly what it has wanted for two years, **leadership, but not dictatorship.**

Ultimately, either a BREW or Java choice both have valid arguments, and may be suited differently for different carriers. Every indication is that this market will offer a positive ROI to the various stakeholders, and provide some of the services the industry has been promising for years. The one thing that is becoming apparent is that **the biggest mistake would be to watch the download game from the sidelines.**

2. Application Platforms for Mobile Phones

2.1. J2ME

Prior to 1999, there were very few standardized application platforms in the micro devices industry. Micro devices such as mobile phones, set-top boxes, HVAC units, or microwave ovens, etc. all had proprietary platforms, and the software that ran on them needed to be coded using proprietary firmware of an extremely specific nature. This brought no economies of scale with respect to programming. Only a handful of in-house developers worked with any given language. Sun Microsystems believed in the advantages of having a consistent platform on which to develop software for these micro devices. If so, mass-market economics would emerge, development tools would appear, a pool of qualified programmers would be accessible, and costs would substantially decrease. Thus, in 1999, J2ME was released by Sun Microsystems to provide that consistent programming environment.

The opportunity to have a standard programming interface matched the arrival of more powerful mobile phone handsets, with better data connectivity. Using J2ME technology, a developer can write an application to run on one mobile terminal that also run on a terminal from a different OEM. The impact on the market is huge. This year, there are 100 M Java enabled mobile phones, over 56 different models (see Exhibit A), and over 34 carriers worldwide who have adopted J2ME.

J2ME Technology Overview

The Java 2 Platform Micro Edition (J2ME) is the Java technology architecture designed to address the specific technical requirements of new consumer and embedded information devices such as mobile phones, personal digital assistants (PDAs), TV set-top boxes, in-vehicle telematics systems etc. J2ME provides a modular and scalable architecture for the development and deployment of portable, dynamically downloadable, robust, and secure applications that is powered by and benefits from the Java technology in consumer and embedded devices. J2ME architecture defines a small set of core APIs that must be fully implemented in every J2ME compatible device, as shown in the figure **J2ME Architecture..**

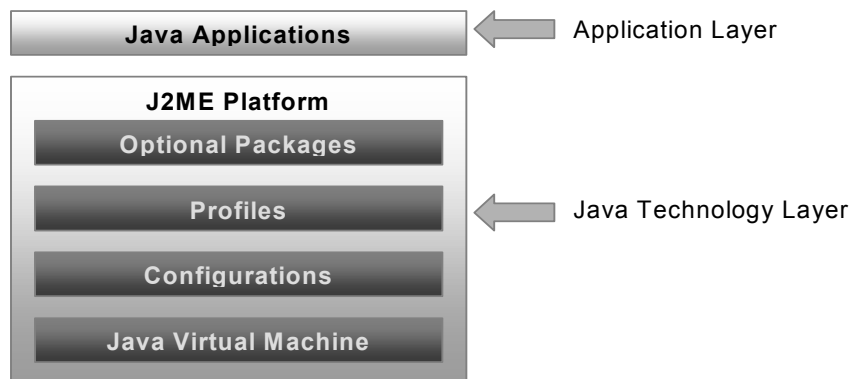


Figure – J2ME Architecture

Source: [The Kerton Group](#)