



Softwired

Introducing Wireless JMS

Dr. Silvano Maffeis, CTO, Softwired.

<mailto:silvano.maffeis@softwired-inc.com>

<http://www.softwired-inc.com/>

[more](#)

Softwired AG * www.softwired-inc.com * info@softwired-inc.com

phone +41 1 445 23 70 * fax +41 1 445 23 72 * Technoparkstrasse 1 * CH-8005 Zurich



Micro-Browsers Are Not Enough!

Wireless platforms based upon markup-languages such as WML have not proven very successful yet. For example, WAP users often complain about a poor user experience due mostly to lack of interesting content, to an awkward interaction model, and to errors occurring when a device loses network coverage for a period of time.

In my opinion, deploying WAP over a faster bearer, such as GPRS, will not solve the problem either. I believe the problem is more fundamental and relies in the fact that an interaction model that is used successfully on the wireline Web, was simply transferred to the wireless Web. This was done without taking into consideration that mobile devices are used in a completely different manner than PCs at home or at the office.

Wireless Java-Middleware to the Rescue

I believe that in the mobile world, user experience is greatly improved by providing an interaction model, in which the user does not have to go and search for content using a micro-browser, but where the user is alerted when something interesting happens. Such an alert should take the user's preferences and geographic location into consideration. This will ensure timely delivery of what an individual user considers to be critical or useful information without the need to actively search for it.

User experience is further improved by provisioning Java applications onto mobile devices, which allow the user to continue working with the mobile application even when disconnected from the network.

Those applications are best built using a "store-and-forward" communication paradigm, which is provided through appropriate messaging middleware (MOM) technology on the device.

To develop this type of mobile application, the approach of my company Softwired is to use Java technology straight through from the mobile devices to the "back-end" application servers. Our approach consists of deploying customized Java applications on the mobile device, and on the server computers.

Then, we use a packet-oriented wireless bearer such as SMS, GPRS, CDPD, or Mobitex to connect the mobile devices to the back-end servers in a flexible and scalable manner.



To make the interactions between the clients and the servers reliable and secure, we deploy standards-based Java messaging middleware technology on the mobile devices and on the servers. For that we rely on our iBus product family, industry's only end-to-end JMS (Java Message Service) middleware solution.

Application Areas

The application areas of such end-to-end, wireless enabled JMS middleware platform are manifold:

- **Mobile workforce automation**, and other **Business-to-Employee** (B2E) systems: These require customized client applications, offline and online operation, as well as high efficiency.
- M-Banking, M-Finance, M-Commerce, and other **Business-to-Consumer** (B2C) systems: Only wireless messaging middleware can guaranteed exactly-once delivery of transactions issued by a mobile user.
- Supply chain management, and other mobile **Business-to-Business** (B2B) systems.
- **Embedded systems**, remote-administration of machines, manufacturing control, and other **Machine-to-Machine** (M2M) **communication** systems: Messaging middleware supports the immediate delivery of signals and other important data.
- **Games**, chat systems, and other **Peer-to-Peer** (P2P) applications: Messaging allows for highly interactive communication.

Wireless JMS Deployment

A Wireless JMS (WJMS) deployment consists of Java-enabled mobile devices (PDAs, Communicators, JavaPhones, etc.) with a WJMS client library installed. Optionally, the devices provide a persistent database in order to implement store-and-forward communication.

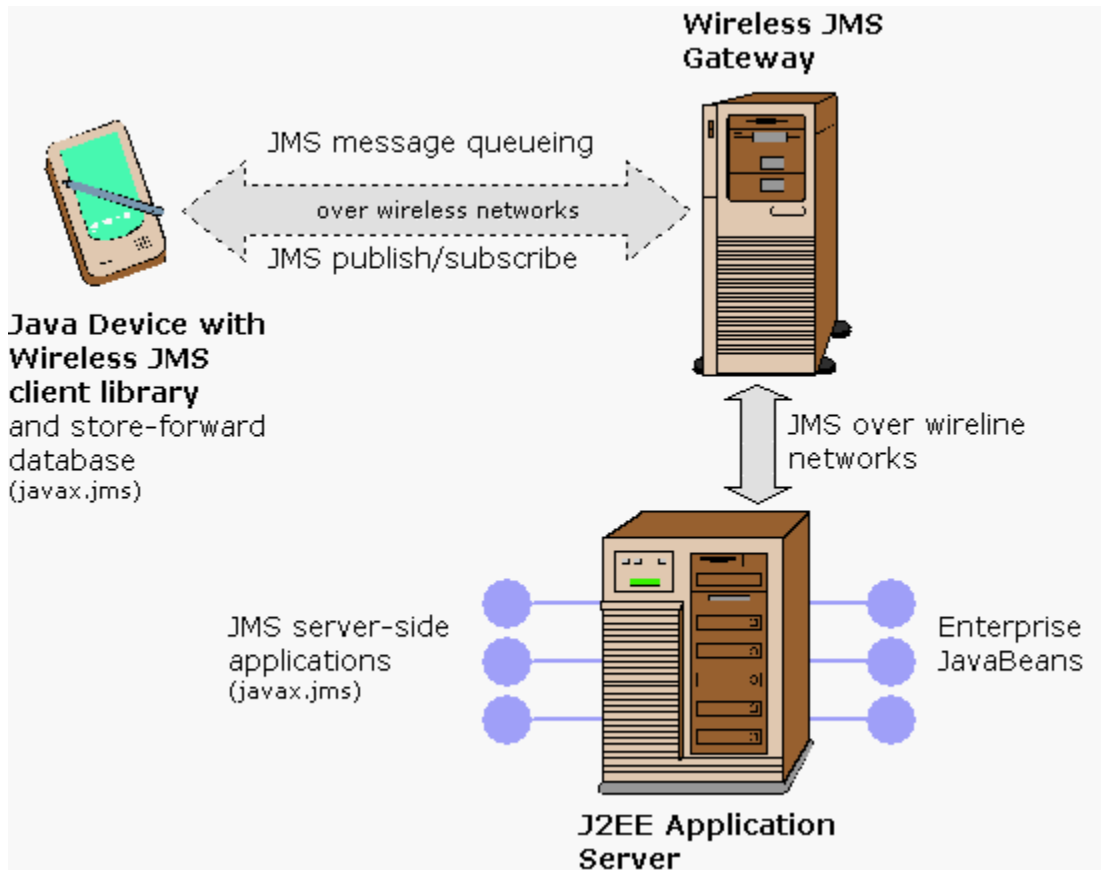


Figure 1: A Standards-Based Mobile Java Platform.

The devices use a wireless network to communicate with a WJMS Gateway service. The Gateway is a software component able of receiving WJMS messages from a wireless network, and of relaying these messages to a server-side JMS provider. The JMS provider either is a stand-alone JMS messaging server such as Softwired's iBus//MessageServer product, or a full-fledged J2EE application server such as BEA's WebLogic Server product.

The server-side application logic of the mobile system is implemented in the form of stand-alone JMS applications, of EJB MessageDrivenBeans, or of both.



Common services such as location, profiling, and billing can also be run under control of the applications server.

This results in a standards-based end-to-end solution, in which we use one powerful communication abstraction straight-through from the clients to the servers: JMS. No need for format transcoders in the form of transcoding engines, Servlets, or server pages.

Wireless JMS Resources

For further information about using Java middleware technology for building mobile Java applications:

- Software Download: iBus//Mobile Standards Based Mobile Java Applications Platform:
<http://www.softwired-inc.com/>
- Documentation: "JMS for Mobile Applications and Wireless Communication".
http://www.softwired-inc.com/people/maffeis/articles/softwired/profjms_ch11.pdf
- Documentation: Chapter 12 of the Book "Professional Mobile Java Programming", Wrox Press, ISBN: 1861003897

or contact the author of this article at silvano.maffeis@softwired-inc.com, Phone: +41-1-445-2370.