M-Commerce Needs Middleware!

Author: Dr. Silvano Maffeis, CTO, Softwired AG, Zurich.
maffeis@softwired-inc.com

Dec.20. 2000

Silvano Maffeis is CTO at Softwired AG. Softwired is a privately held, venture capital funded enterprise messaging middleware software company headquartered in Zurich, Switzerland. Softwired has developed wireless messaging middleware infrastructure, such that highly scalable mobile commerce, wireless news distribution, location aware services etc. are a reality today. For more information, visit http://www.softwired-inc.com/

Introduction

Device manufacturers pushing new types of hardware, notably wireless PDAs, communicators and Smart Phones, into the market will characterize the future (FIGURE 1). Furthermore, Telecommunication firms will roll out packet-oriented, wireless bearers such as GPRS, EDGE and UMTS fairly quickly. This huge investment will be profitable only if a “killer application” materializes, propelling an explosive growth in demand for devices and bandwidth.

[FIGURE 1: Devices suitable for advanced m-commerce solutions.]

Just think for a moment about the history of the Internet. For the early nineties, the Internet was used predominantly for e-mail, file transfer, and accessing remote computers (telnet). The Web browser, along with the HTTP protocol, was the “killer application” that lead to the explosive growth of the Internet, to the proliferation of e-commerce, and to the establishment of many dot-com companies.

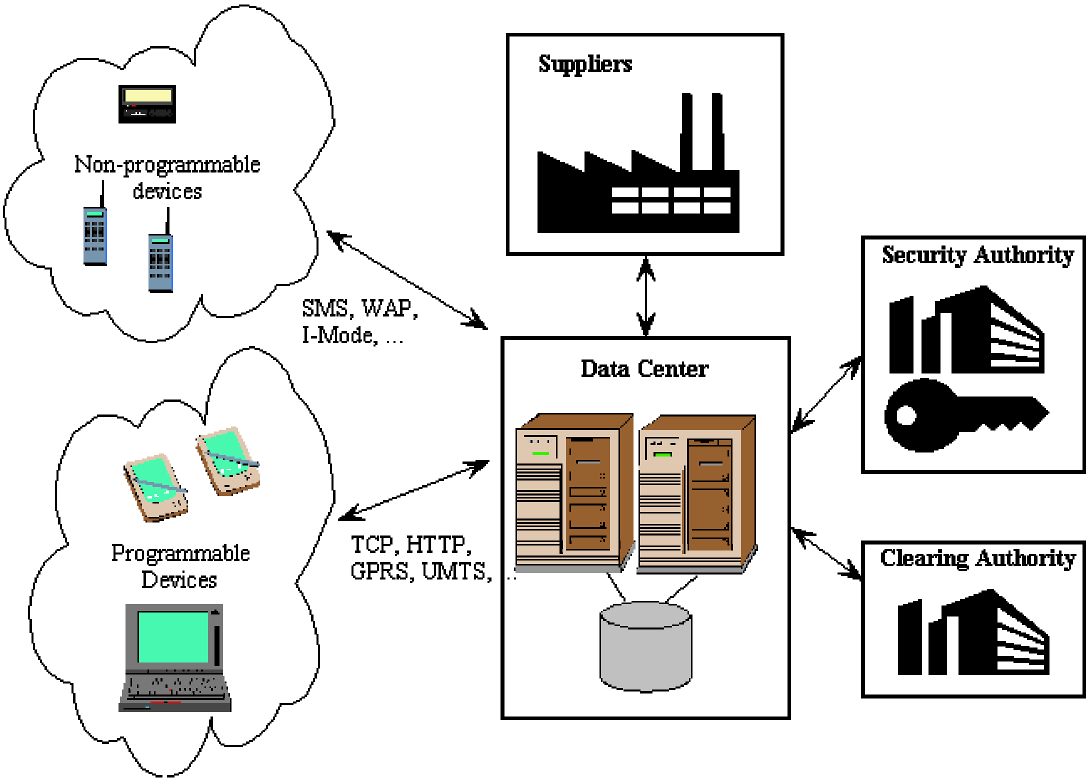
Something similar will likely occur for the Wireless Net which, is just emerging. However, I do not believe Web browsing will be the catalyst or “killer application” of the Wireless Net (for example, WAP hasn’t been a big screaming success yet!). So what “killer application” will make the huge investment into mobile devices and wireless communication infrastructure profitable? M-commerce could be one such important application. Others could be wireless enabled entertainment systems, local information, virtual communities, alerting and dispatching.

The goal of this article is to point out that we have just begun to understand how to build scalable, *wireline* information services for the Web. However, many big players intend to roll out *wireless* information systems, serving hundreds of thousands or millions of users simultaneously. Building this kind of information system is very challenging and barely understood today. There is a good body of knowledge, consulting companies, and books addressing today's *wireline* Web information systems. But it is much more difficult to obtain adequate help, supporting tools, and technical information when it comes to building *wireless* information systems.

This article outlines a very promising approach for building highly scalable and reliable m-commerce applications as well as other interactive wireless solutions. It is based on well-accepted industry standards framed around the Java and messaging middleware.

The Architecture of M-Commerce Systems

Without oversimplifying matters, we can say that an m-commerce system consists of Mobile Enduser Devices (Phones, PDAs, and Communicators), a Data Center hosting m-commerce applications, a Security Authority, a Clearing Authority (e.g., a bank), and Suppliers of merchandise. This is depicted in FIGURE 2.



[FIGURE 2: The architecture of M-Commerce Systems. Arrows denote data flows.]

Shown are:

- Mobile Enduser Devices: Those run a browser or a customized m-commerce application.
- Data Center: A Wireless Application Service Provider (WASP) running the server side of an m-commerce system on behalf of the Suppliers.
- Security Authority: provides security certificates, signs transactions, etc.
- Clearing Authority: allows the Data Center to check credit card information, to charge customers for their purchases, and to transfer funds to the Suppliers.

At the heart of this architecture is the Data Center. It orchestrates m-commerce transactions issued by the mobile devices, issues shipping requests to the Suppliers, charges credit cards, and so forth. The Data Center typically consists of a cluster of server computers running the "server side" of the m-commerce system, maintaining transaction information in a database.

Key Requirements of M-Commerce Systems

Device Independence

To be successful, M-Commerce systems typically need to run on a variety of mobile device platforms. Examples include conventional phones (WAP, SMS), Smart Phones (Running the Java MIDP platform), PDAs (such as the Palm), Communicators (Such as Symbian "Quartz" and "Crystal" devices). A browser-based architecture such as WAP might be sufficient for some m-commerce solutions, but others will require the deployment of a mobile application (for example a trading user interface) onto the mobile device.

Bearer Independence

This also means that *different wireless bearers* need to be supported: WAP, SMS, GPRS, and possibly UMTS.

Security

Identification, authentication (via the Security Authority), access control, and end-to-end data encryption must be supported for any m-commerce solution to be acceptable.

Reliability

What happens if a device has weak or intermittent network coverage while an m-commerce purchase is performed? *Guaranteed, exactly-once execution* of m-commerce transaction is another important requirement, especially when stocks or other expensive items are purchased from a mobile device. This feature is not supported by WAP, typically the deployment of a special piece of software on the mobile device is required. This special type of software, called *middleware*, will be addressed later.

Notifications

Another requirement is being able to send *notifications* to the customer to inform him/her that the merchandise has been shipped, that the supplier is out of stock, or that the credit card could not be charged. Such notifications typically occur after the user has issued a purchase transaction.

A Solution

Having listed the problems and requirements for building large-scale m-commerce systems, we can now outline an elegant technical solution addressing those issues.

Java Technology on the Devices and Servers

If special purpose m-commerce software, such as a stock trading application, is to be deployed on mobile devices, then we recommend that *Java* be used as the deployment language on the mobile device. By using Java, the amount of software that needs to be changed in order to adapt the application to various mobile platforms is minimized. Nowadays, viable Java execution environments are available for Palm PDAs, Smart Phones, Symbian, Windows CE and other platforms. We further recommend using server side Java technology, notably the Java-2 Enterprise Edition (J2EE) platform, in the data center. This allows for shorter time-to-market and avoids vendor lock-in.

Messaging Middleware to the Rescue

The Java platform or the mobile operating systems themselves do not provide the necessary tools and features necessary for developing scalable and secure m-commerce systems in reasonable time. By that I mean features necessary to ensure a guaranteed delivery of transactions, security, the timely delivery of notifications, integration of the data center software with the logistics systems of the suppliers, etc.

Here we propose utilizing so-called application-to-application *messaging middleware*. Messaging middleware has been used very successfully for many years in financial information systems, e-commerce systems, logistics solutions, etc.

Middleware denotes a class of "invisible" software residing between an application and the underlying operating system environment (FIGURE 3).

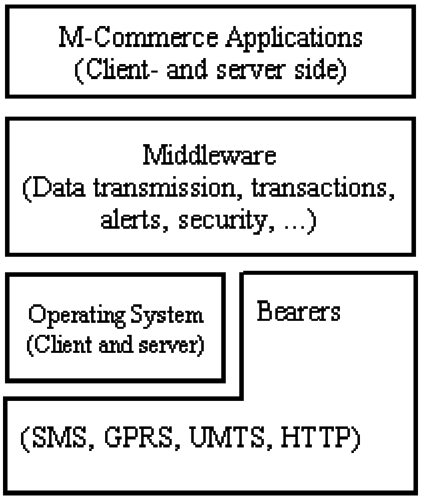


FIGURE 3: Mobile applications should be built atop *middleware*. The middleware adds important features to the underlying operating system environment and communications bearers.

Note: Developing an m-commerce system without viable middleware is much like developing word processing software directly atop the bare hardware, without taking advantage of an operating system such as Windows or Linux.

Messaging middleware ensures the reliable delivery of purchase orders and other m-commerce transactions, from a mobile device to the data center. The exactly-once delivery (and execution) of a transaction is guaranteed even when the user issues a "purchase" transaction whilst driving through a tunnel, or in other cases where the device might drop in and out of network coverage.

Messaging middleware also enables the integration of the m-commerce services in the data center with the logistics information system at the suppliers.

Another distinctive feature of messaging middleware is its ability to deliver information in "real-time", meaning immediately after an interesting event has occurred. For example, the data center can send a notification to a mobile device informing that an item that was ordered the day before has now shipped, or inform the user of the shipping date. This type of real-time communication is interesting for other mobile services as well: delivery of stock quotes and news, or for enabling "virtual communities" (e.g., chat rooms and buddy lists).

Furthermore, messaging middleware can be deployed efficiently atop a multitude of Internet standards (TCP/IP, HTTP), atop current-generation wireless bearers (GSM, SMS, TDMA), and also atop next-generation wireless bearers (GPRS, EDGE, UMTS).

In summary, for the development of viable m-commerce systems, we propose deploying the Java-2 Micro Edition (J2ME) platform on the mobile devices, the Java-2 Enterprise Edition (J2EE) in the data center, and using messaging middleware as a versatile “bridge” between mobile devices and the data center. The resulting m-commerce infrastructure is versatile (it supports many different devices and bearers), scalable, secure, and based on well-accepted industry standards.

We further propose using messaging middleware compatible with the Java Message Service (JMS) standard. By choosing a middleware that is compatible with JMS, a reduction of risk is achieved and the “time-to-market” can be decreased as well, as JMS fits seamlessly into the Java platform and is easier to use than other middleware standards in particular such as OMG and CORBA.

Where Do We Go From Here?

We have pointed out that messaging middleware plays a key role in mobile commerce solutions, it reduces time-to-market and the risk of depending excessively on a single proprietary vendor. Messaging middleware allows the developer of m-commerce solutions to focus more on the business problem, and less on issues related to deploying m-commerce software on various platforms, or on issues related to transmitting m-commerce transactions and other information reliably.

When choosing a messaging middleware vendor, you might want to ensure that:

- the middleware product runs both in the data center and on the mobile device
- the product fits seamlessly into the Java platform
- the product adheres to the JMS (Java Message Service) standard as well as to Internet standards such as XML and HTTP.
- the vendor can help you in architecting and developing your m-commerce system.

Besides a wireless-enabled messaging middleware, you will need to choose also the following tools and components:

- A J2EE application server. The server side of the m-commerce system runs on an application server in the data center. The application server must provide high scalability and fault-tolerance through "clustering".
- A WAP server, if you are planning to provide online catalogs and the like.
- A scalable database system to maintain customer account information at the data center.
- Monitoring and network management software for the data center. This helps you to ensure a smooth operation of the various software components and services running in the m-commerce data center.
- If you are planning to deploy customized software on mobile devices, such as a stock trading application, then you will also require development environments and emulators for the respective mobile device platform (e.g., Symbian, Palm, or Windows CE).

In summary, all the pieces of the puzzle (which we have found) for building a scalable m-commerce system are available today, without "locking" you into a single vendor. Those components can be evaluated individually, and "glued" together using Java messaging middleware creating an unparalleled m-commerce solution.

Resources

Java-2 Micro Edition: http://www.java.sun.com/products/j2me
Java-2 Enterprise Edition: http://www.java.sun.com/products/j2ee
Java Message Service (JMS) middleware: http://www.java.sun.com/products/jms
JMS for mobile devices: http://www.JavaMessaging.com/
Every dad needs a MOM (Messaging Oriented Middleware):
http://www.softwired-inc.com/pdf/technology/momdad-final.pdf
Softwired iBus//Mobile – Whitepaper by Dr. Silvano Maffeis CTO Softwired:
http://www.softwired-inc.com/products/mobile/getpdf.html