

Giving Currency to Java Card API

Subtitle: A Developers Guide to Opportunities on a New Java Platform

Summary: Java Card API is the newest Java platform. It is designed to run on 8-bit microprocessors with as little as 256 bytes of RAM and 14K of ROM. Java Card has been adopted by all manufacturers of smart cards worldwide because it provides the first opportunity to run single applications on cards from any vendor, because it enables the first true multi-application cards, and because it speeds and simplifies application development in these very small memory environments. This article is designed to introduce smart card concepts to Java developers and provide some insight into revenue opportunities represented by these new platforms.

Main Body:

The computer industry, more than any other, is built on the fundamental assumption of constant change. Moore's law, for example, states that every eighteen months the number of transistors that can fit in a specific area doubles. Inexorably and with almost assiduous precision, the industry experiences the obsolescence of an entire spectrum of products, and the birth of a new generation based on the latest generation of microprocessors. The newest generation of processors has yielded many well recognized devices, including sub-notebooks, personal digital assistants, and intelligent cell phones. It has also yielded a new generation of devices which have not been as well known - until now - of which smart cards are only one example.

But smart cards, because they have existed in a variety of forms since 1974, have had substantial success in parts of Europe (France and Germany notably), and have had some substantial companies driving their development (e.g. Bull/CP8, Gemplus, and Schlumberger, among others) have received more attention than other new devices. In many ways, smart cards have now reached critical mass - not only in Europe, but in the United States and Asia, as well. Today, 95% of all smart cards are sold in Europe. By 2001, analysts at Data Monitor predict that well over 25% of the worldwide market for the *3 billion smart cards sold in that year* will be outside of Europe. Much of the success that smart cards will experience in the

next few years, and their increasing worldwide adoption, is directly related to one fundamental invention that is changing the nature of competition in the smart card industry: Java Card API

Java Card API is the smallest virtual machine specification for Java. The specification is designed to allow Java to run on an 8-bit microprocessor, with 14 Kbytes of electrically erasable and programmable read only memory (EEPROM), 8 Kbytes of read only memory (ROM), and 256 *bytes* (no we did NOT forget the K) of random access memory (RAM). The specification was first released in November, 1996 at the Cartes 96 show in Paris. By the release of Java card API 2.0 one year later, every major vendor of smart cards in the world had licensed the technology. All these vendors are in the process of building working implementations. Both Schlumberger and Gemplus, companies which combined sell something like 75% of all smart cards, already have existing implementations and development toolkits for sale. By 2001, over 700 million of the 3 billion cards sold will run the Java Card virtual machine.

This rapid adoption rate for Java technology on a platform with which most Java developers have little familiarity has left independent software developers and system integrators wondering what smart cards, and Java Cards in particular, mean to their business. Is this an area they should be building competence in and if so, what numbers and types of resources should be developed to take advantage of this new opportunity? Equally important, the more creative are asking what Java on small devices means to their business, and are beginning to make plans to succeed in the world of *individualistic computing*.

Individualistic computers are portable devices containing a microprocessor that contain some specific functionality that is under the consumer's control. Devices include something as large as a personal digital assistant or cell phone, and something as small as an intelligent watch or ring.

This article provides some answers to these questions. The article begins with a brief review of smart cards, the benefits of Java Card API, and an explanation of why Java Card API has taken this industry by storm. This is followed by some ideas of the types of applications and services that will be demanded by companies trying to implement Java Card technologies. Lastly, we discuss how Java Card is part of a larger trend in computing, and suggest ways for you to begin thinking about developing the skills needed to succeed in an era of individualistic computers.

What is a Smart Card?

A smart card is a card embedded with either a microprocessor and a memory chip or only a memory chip with nonprogrammable logic. The microprocessor card can add, delete, and otherwise manipulate information on the card, while a memory-chip card--the most common example of which are prepaid phone cards--can only undertake a predefined operation. Smart cards, unlike magnetic stripe cards, can carry all necessary functions and information on the card. Therefore, they do not require access to remote databases at the time of the transaction.

Smart cards have three unique benefits that have made them popular: increased data storage, security, and portability. A typical mag-stripe card can hold about 140 bytes of data--just enough space for a PIN number and critical data needed to log into a server-based system. In many situations, such as automating drivers' licenses, this is not enough memory to meet the application's storage needs. Mag-stripe cards have two other weaknesses. First, they are not very secure. It is easy enough today to purchase the tools needed to hack into confidential data on a mag-stripe card. Any card--a phone card, for example--that requires storing value (money) on the card requires substantially more secure technology

Second, mag-stripe cards require a host system to store and process all data. For mag-stripe cards to be useful, your card must always be connected to an online system. This is not necessarily a problem in the United States, where telecommunications costs are low. However, in other countries (France, for example), the costs of telecommunications make it uneconomical to keep a reader continually on-line to process a credit-card or debit-card transaction. Taking the reader off-line and having a dial-up process for approval is not an option either--it takes too long. Thus, smart cards--in this case, chip cards--offer a unique solution. The transaction can be processed and stored locally with a chip card, and then the entire batch of transactions can be downloaded to the bank's system at the end of the day.

Third, portability becomes even more important for certain types of data - for example, personal identity data (also called your *personal digital identity*), contact information, among others - in a world of

multiple devices where data must be accessed on an anytime, anywhere basis. Some devices, like kiosks, will be networked. Others, such as your personal digital assistant or a data pad for viewing secure information, may not. Consumers will want certain types of information with them at all times, no matter which device is used, where it is, or whether it is networked. Smart cards, with their processing capabilities and increased memory capacity, provide a convenient solution for making data portable and universally accessible.

Java Cards Offer Extra Benefits

Although smart cards have benefits over mag-stripe cards that make card issuers willing to spend substantially more for an individual card, smart cards had some limitations--portability of applications, the flexibility to download applications into a card, and a small universe of knowledgeable programmers. The Java Virtual Machine's unique design offered the smart-card industry a solution to these limitations. Java's most fundamental feature--the ability to write applications on one operating system platform and have them run on all platforms--allows a single application to run on all smart cards.

Just as important, Java's design around applets--small code objects that are designed to be downloaded onto a client machine from a remote host--is perfect for allowing multiple applications to reside on a single card. Applets are small enough so that several can fit into the small amounts of memory available on smart cards. Because they are downloadable, applets can also be installed "in place" via card-accepting devices such point-of-sale (POS) terminals. Java Cards' functionality can be continually updated as new applications or updates become available.

Java provides a security model that lets applets from multiple sources reside safely on the same card. This is important, because applets on smart cards may attempt access to data intended to be private only to that

applet. For example, you would not want to allow one applet to get at critical purchasing histories from another application that might come from a competing company.

Java speeds and simplifies the development of smart-card applications. Because of its many time-saving features--automated memory management and automatic garbage collection, for example--Java can reduce coding time over traditional languages such as C++ by as much as 60 percent. The results are even more impressive when compared to the proprietary, assembly-language-like environment for smart card development. Applications compiled into machine code typically take many months (up to a year) to debug and deploy, because they need to be masked in ROM for security reasons. Applications written in Java, however, can be debugged much more easily and deployed almost immediately through Java's dynamic applet download capability.

Part of the reason for this improvement has to do with the tools that can be used. Instead of having to learn a new proprietary tool, Java Card developers can build applications by using standard, off-the-shelf integrated Java development environments such as Symantec Cafe. Code written this way then runs through a post-processor, which shrinks the applet as much as possible and ensures that the code will run correctly in the limited Java Card environment.

Java Solutions

Although it is still too early to know whether the ultimate model will be for individuals to carry a single card that holds all their private information or whether they will carry a variety of cards, there are some clearly defined markets that will adopt smart cards in the next few years. The most immediate areas where Java Cards will find a use include:

* **Financial Services.** Financial institutions are looking to use smart cards to deliver higher value-added services to businesses and consumers at a lower cost per transaction. These services include money on a card, corporate card programs, and targeted marketing programs based on analysis of consumers' buying patterns.

* **Affinity Programs.** Airlines, retailers, and other companies that offer a range of ancillary services and loyalty programs along with their basic product want use smart cards to deliver these programs with a higher level of service, improved ease of use, and at a lower cost. For example, airlines want to use smart cards not only as a vehicle for issuing and carrying tickets--even though the single benefit of being able to securely order/provide a ticket directly to chip cards via the Internet is substantial. Airlines want to use the cards to provide tie-ins to their frequent-flyer programs and to cross-marketing deals with auto rentals and hotels, as well as to provide simplified access to private airline lounges.

* **Cellular Phones.** Cellular phone services in the United States are losing \$1.5 million per day to fraud. Although smart cards offer a mechanism to secure cellular phones against fraudulent use, only Java Cards offer the ability to download new functions into a phone in real time.

* **Set-Top Boxes.** Subscription satellite and cable services suffer from fraud problems similar to those in the cellular phone business. Once again, Java Cards offer security and the ability to add/update customer functions available to consumers in real time.

* **Secure Network Access.** Smart cards can carry an individual's digital signature. As such they provide a special mechanism to secure access to computer networks within a corporation, help ensure that only

individuals with the proper authority can get access to specific network resources, and reduces the likelihood that hackers can break into a system.

Figure 1 shows projected sales of smart cards by vertical market over the next few years. The figure shows that applications for phones, banking/finance, games, and security will create the greatest demand for cards. 1998 will be the year for pilots, as working versions of Java Card 2.0- enabled cards finally come to market in large numbers from multiple vendors. 1999 will see more pilots, but also some of the pilot projects from the previous year moving into full production. 2000 and beyond? Well, that's the hockey stick. That's when demand for smart card programmers, applications, and services is going to potentially far outstrip supply.

FIGURE 1
Annual Worldwide Smart Card Sales (millions), By Industry

Market Segment	1994	2000	Annual Compound Growth Rate
Phone Cards	310	1,400	29%
Bank and Loyalty Cards	20	500	71%
Games Cards	1	500	182%
Health Cards	62	400	36%
ID Cards	1	400	171%
Transport Cards	1	200	142%
Vending Cards	4	200	92%
Pay TV Cards	10	100	47%
GSM Cards	9	50	33%
Meter Cards	2	50	71%
Access Control Cards	-	-	0%
TOTAL	420	3,800	

Source: GemPlus, 1996

Reaping the Java Card Bounty

So, what opportunities exist in this looming bonanza for application developers, custom software developers, and application integrators? The answer is somewhat schizophrenic: lots and very few. Lots of opportunities will exist for those companies that spend the time to understand the unique features of the smart card market, identify trends in that market that they can exploit, and build the skills needed to succeed. Very few opportunities will exist for the average Java programmer who treats Java cards - and for that matter any of the low-end Java platforms (e.g. Personal Java, Embedded Java) - as just an extension of the desktop. Put it this way: "I can write a Java applet. Java Card is written in Java. Therefore, I can write Java Card applets" is not a formula that guarantees success. Substantial success with the low-end Java platforms is possible. But like most business opportunities, success will require a reasonable investment on the part of any newcomer.

Let's consider the smart card market for a moment to understand why this is true. Today, the smart card market consists of about 10 smart card manufacturers (mostly French), a host of terminal vendors (who make the card readers), about 50 smart card issuers worldwide, a handful of system integrators like Cards, Etc. in Australia, and (in the case of the financial services industry) card associations like VISA, Mastercard/Mondex, American Express, Europay, and JCB. About 200 smart card programmers worldwide, split between the manufacturers, the issuers, and the integrators, serve this market. This is a relatively small universe of participants who interact through an existing set of corporate relationships and individual networks. Since the technology is so detailed, because the development environment is so arcane and difficult, and since security is one of the key benefits of smart cards, issuers tend to turn to the well-established experts in the field, in combination with their own internal staff, to build their software and back-end systems.

Just because Java Card API suddenly allows any Java programmer to write an applet doesn't mean that the industry is suddenly going to change its approach to business. As with all new technologies, it will take time for industry thinking to catch up with the implications inherent in a new paradigm.

For one thing, security continues to be a concern, especially in the case of financial institutions issuing stored-value cards. These institutions have financial liability if a rogue applet is installed on their cards and ultimately manages to steal money or other confidential data from their systems. Therefore, you should not expect to be able to buy a blank card at your local computer warehouse store and install an applet, at least not in the next few years. In the near future, cards will continue to be distributed by issuers, who will act as the controlling bodies for what applications are loaded onto their cards. They will test and certify each applet before it is loaded. Therefore, the early opportunities around Java card will involve working hand-in-hand with the smart card issuers to develop their specific applications and systems.

I advise firms and individuals interested in beginning to build expertise in smart cards to follow a four step process between now and the end of the millenium. First, begin playing with the technology and learning the market. Buy the toolkits that are in the marketplace and develop some initial applications to learn the process of writing software for this unique environment. Study the marketplace by subscribing to appropriate publications and talking to experts in the field (some of whom are listed at the end of this article).

Second align yourself with existing partners who have substantial expertise in the field and who will be needing many more technologists in the next few years than they can hire. Don't plan to make money on your first couple of projects, but do them well and establish your reputation with your early partners. Once you have established a good reputation with a single vendor, lots us business will begin to flow.

Third, build your brand identity in the marketplace. Once you have succeeded with one partner or small set of partners, begin to develop a brand. Invest in a good logo, develop a clear statement of your brand

values and market focus, put together a good set of collateral. Combine this with a good marketing plan and some investment in marketing activities such as booths at trade shows, public relations, and seminars.

Fourth, put in good recruiting and training programs. Like your customers, your growth will be limited by the dearth of technologists who know anything about smart cards and other low-end devices. In order to grow, firms will need to recruit smart people with related experience (e.g. embedded systems development) and retrain them to their specific needs. The firms that combine good marketing with an ability to maximize their human resources will ultimately become the new generation of successful smart card technology vendors.

Conclusion

Smart cards and Java Card API represent a relatively new set of technologies with a great deal of promise. Smart cards provide a secure, portable platform for “any time, why where” computing that can carry and manipulate substantial amounts of data, especially an individual’s personal digital identity. Java Card API allows applications written for one card to be downloaded “in place” into any other card. Java Card thus allows smart cards to become a general purpose computing platform and creates a potentially huge market for application software and development - especially for financial, telecommunications, television, and security applications.

By the year 2000, the demand for Java Card API applications and services will begin to outstrip the current supply of resources available to the industry, thus creating a potentially lucrative market for individuals and companies who wish to specialize in smart card application development. However, succeeding in this new market will require working today to establish your company’s expertise and reputation with the key players in today’s market - including smart card manufacturers, terminal vendors,

issuers, system integrators, and others. Companies that make this investment in the next 24 months will be positioned to take advantage of the rapid market expansion that will occur around the year 2000. By that time, the market will have evolved to a point where many new opportunities - standardized applets for cards, enterprise systems software, integration of card capabilities into “boxed” software, systems integration, to name a few - will become available. Those who are prepared will prosper and have the psychic satisfaction of knowing they correctly anticipated and participated in the birth of an entirely new generation of computing.

Resources

1. JavaSoft Resources

Type of Resource	Contact Information
Java Card API Web Site	http://java.sun.com/javacard. This web site contains links to all the companies that are currently licensed to issue Java Card-based technologies
Java Card Developer Programs	1-800-JAVASOFT or (512)-434-1591 from international locations
Java Developer Connection	http://java.sun.com/???? (to be filled in)
Java Card Business Development	javacard@java.sun.com

2. List of industry resources to contact for further information

Type of Resource	Contact Information
Java Card Forum	

Smart Card Forum	
Schlumberger	
Bull/CP8	
Dallas Semiconductor	
De la Rue	
Geisecke & Devrient	
GemPlus	

3. List of publications on smart cards (Bank Technology News, Card Technology Journal, etc)

Publication	Subscription Information
Card Technology Journal	
Bank Technology News	
Retail Delivery Systems News	www.ectoday.com Phillips Business Systems Information 800-777-5006 \$595/year