San Francisco, CA 94118

free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to surf more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a list of available dates. The visitor may select one of those dates, and then begin surfing on an archived version of the Web. The links on the archived files, when served by the Wayback Machine, point to other archived files (whether HTML pages or images). If a visitor clicks on a link on an archived page, the Wayback Machine will serve the archived file with the closest available date to the page upon which the link appeared and was clicked.

4. The archived data made viewable and browseable by the Wayback Machine is compiled using software programs known as crawlers, which surf the Web and automatically store copies of web files, preserving these files as they exist at the point of time of capture.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL]. Thus, the Internet Archive URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). A web browser may be set such that a printout from it will display the URL of a web page in the printout's footer. The date assigned by the Internet Archive applies to the HTML file but not to image files linked therein. Thus images that appear on a page may not have been archived on the same date as the HTML file. Likewise, if a website is designed with "frames," the date assigned by the Internet Archive applies to the frameset as a whole, and not the individual pages within each frame.

6. Attached hereto as Exhibit A are true and accurate copies of printouts of the Internet Archive's records of the HTML files or PDF files for the URLs and the dates specified in the footer of the printout (HTML) or attached coversheet (PDF).

7. I declare under penalty of perjury that the foregoing is true and correct.

DATE: 12/18/18

Christopher Butler

accuracy, or validity of that document.

State of California
County of San Francisco

Subscribed and sworn to (or affirmed) before me on this

18 day of December, 2018, by

Christopher Butler,

proved to me on the basis of satisfactory evidence to be the person who appeared before me.

Signature:

Exhibit A

Develop on the Java platform of the future

Summary
This article, the first in a new **Java Developer** series on smart cards, will introduce you to smart card hows and whys. All you need is a smart card, a card reader, and software that lets you communicate with the card, and you can begin developing a real-world application. This article includes: a package to manipulate smart cards using ISO 7816; a demonstration of how to read and write data to a memory smart card using the Gemplus reader and a Gemplus smart card; and a discussion of some of the applications that make use of smart card memory features. Future articles will use cards by different manufacturers. We'll also touch on smart card standards.

With this first article in the series, we will lay the groundwork for future articles with a discussion of the new and emerging standard called OpenCard. Future articles will deal with security cards and electronic purse cards. Finally, this article will teach you some of the basics about smart card software architectures. (3,500 words)

Smart cards have been getting a lot of buzz lately on the Web, at the JavaOne conference last April (four sessions dealt with the technology), on the big network news stations, and on CNN. In this article we'll bring the smart card to life with a real-world smart-card example. The techniques presented here will allow you to start building Java applications that are smart-card enabled.

We'll focus on two types of smart cards: memory smart cards, which can be viewed as minuscule removable read/write disks with optional security; and processor cards, which can be viewed as miniature computers with an input and output port. Future articles will cover processor cards in greater depth.

As the meat of the article, we'll develop a simple prototype for reading and writing data to a smart card. We will discuss a drug prescription card, which keeps a list of all your prescriptions and tracks your insurance, prescription plans, and other useful info. Later articles will expand on the idea of the prescription card.

You'll notice that a recurrent theme that runs throughout this series on smart cards is the need for a security framework to prevent rogue plug-ins, ActiveX components, and so on from getting at your personal and/or corporate info-goodies. To this end, the demonstration of how to read and write data to a smart card included in this article will provide you with persistent, secure (and portable) storage.

What is a smart card?

You can think of the smart card as a "credit card" with a "brain" on it, the brain being a small embedded computer chip. This card-computer can be programmed t perform tasks and store information, but note that the brain is little -- meaning that the smart card's power falls far short of your desktop computer.

Smart cards currently are used in telephone, transportation, banking, and healthcare transactions, and soon -- thanks to developers like you -- we'll begin to see th used in Internet applications. Smart cards are already being used extensively in Japan and Europe and are gaining popularity in the U.S. In fact, three significant events have occurred recently in the smart card industry in this country:

PC/SC
Microsoft and several other companies introduced PC/SC, a smart card application interface for communicating with smart cards from Win32-based platforms fo personal computers. PC/SC does not currently support non-Win32-based systems and may never do so. We will discuss this in greater detail later on.

OpenCard Framework
OpenCard is an open standard that provides inter-operability of smart card applications across NCs, POS, desktops, laptops, set tops, and so on. OpenCard promi to provide 100% pure Java smart card applications. Smart card applications often are not pure because they communicate with an external device and/or use libraries on the client. (As a side note, 100% pure applications could exist without OpenCard, but without it, developers would be using home-grown interfaces t smart cards.) OpenCard also provides developers with an interface to PC/SC for use of existing devices on Win32 platfroms.

JavaCard
JavaCard was introduced by Schlumberger and submitted as a standard by JavaSoft recently. Schlumberger has the only Java card on the market currently, and th company is the first JavaCard licensee. A smart card with the potential to set the overall smart card standard, JavaCard is comprised of standard classes and APIs that let Java applets run directly on a standard ISO 7816 compliant card. JavaCards enable secure and chip-independent execution of different applications.

Note:
Although this article focuses on smart cards, it is important to note that you are not limited to these kinds of devices. Personally, I prefer the "Ibuttons" device being produced by Dallas Semiconductor. It is small and portable like a credit card, but so much handier. Why? You don't have to dig out your wallet in search of a card; Ibuttons is right there, on your finger. Yes, it's a ring!

While contactless versions of the smart card do exist (see below for more information on this), I think the Ibuttons, functional-jewelry type of device could be quite profitable. For more information on Ibuttons, see the Resources section. By the way, the Java Commerce Team demonstrated a

"JavaRing" at Java Internet Business Expo (JIBE) in New York last August. You can read about this in the article in Fortune magazine (again, see the Resources section).

hy use a smart card?

hat are the advantages of using a smart card? Well, a smart card:

- is more reliable than a magnetic stripe card
- currently can store a hundred times more information than a magnetic stripe card
- is more difficult to tamper with than mag stripes
- can be disposable or reusable
- can perform multiple functions in a wide range of industries
- is compatible with portable electronic devices such as phones, personal digital assistants (PDAs), and PCs
- is constantly evolving (after all, it incorporates a computer chip)

ypes of smart cards

s mentioned above, this article will focus on two types of smart cards -- memory and process. In all, there are five types of smart cards:

1. memory cards
2. processor cards
3. electronic purse cards
4. security cards
5. JavaCard

mart cards are a personal piece of hardware that must communicate with some other device to gain access to a display device or a network. Cards can be plugged to a reader, commonly referred to as a card terminal, or they can operate using RF radio frequencies.

mart cards can communicate with a reader or receiver (see the section on readers below for more on these two terms) in one of two forms:

ontact smart cards -- The connection is made when the reader contacts a small gold chip on the front of the card.

ontactless smart cards -- These can communicate via an antenna, eliminating the need to insert and remove the card by hand. With a contactless card, all you have do is get close to a receiver, and the card will begin communicating with it. Contactless cards can be used in applications in which card insertion/removal may be npractical or in which speed is important.

ome manufacturers are making cards that function in both contact and contactless modes.

reate a development environment for building smart card apps

order to develop smart card applications, you need a few things, namely: a smart card reader; software to communicate with the reader as well as some software communicate with the card that has been plugged into the reader; and, of course, smart cards and smart-card hardware.

mart card reader

o communicate with a smart card or develop an application that is smart-card capable, you must have a reader. The reader provides a path for your application to nd and receive commands from the card. There are many types of readers on the market, the most prevalent being the serial, PCCard, and keyboard models. Keyboard models pop up here and there; expect them to be directly available from the large PC makers by June 1998.)

his article uses serial readers to support the devices. A serial reader connects to a computer's serial port. Note that the code provided also supports a PCCard ader; most laptops come with PCCard slots built in.

ach manufacturer provides a different protocol for speaking to a reader. Once you can communicate with the reader, there is one protocol for communicating with smart card: Communication with a smart card is based on the APDU format. (The APDU format is discussed below.) For information on purchasing your own ader, see the "Gemplus smart card readers" heading in the Resources section.

oftware for communicating with the reader

number of object-oriented classes are needed for the smart card example included in this article. These are:

- ISO command classes for communicating with 7816 protocol
- Classes for communicating with the reader
- Classes for converting data to a manufacturer-specific format
- An application for testing and using the cards for the purpose for which the application was designed

mart cards and smart card hardware

s noted at the beginning of the article, in order to develop the smart card application here, you need smart card hardware and some smart cards. You can purchase mart card development kits from a number of companies, including Gemplus and Schlumberger.

or those of you who already have readers, you should be able to use your reader by supplying an implementation of an interface class that we will discuss later. As entioned above, before we can communicate with the card, we must be able to communicate with the reader, and just as there are many different cards, there are any different readers.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.