

JAVA CARD

Hemant V. Badgujar*1, S.P. Patil*2

*1,2Department Of MCA, Yashoda Technical Campus, Satara, Dist. - Satara (415001),
Maharashtra, India.

DOI : <https://www.doi.org/10.56726/IRJMETS53361>

ABSTRACT

Java Card technology provides a secure and portable platform for developing smart card applications. It is widely used in many applications such as payment, ID card and access control due to its simplicity, security features and interoperability. In this lesson we will explore the interesting field of Java Card development. The aim of the training is a better understanding of Java Card technology, including the design and development process for creating a secure smart card.

I. INTRODUCTION

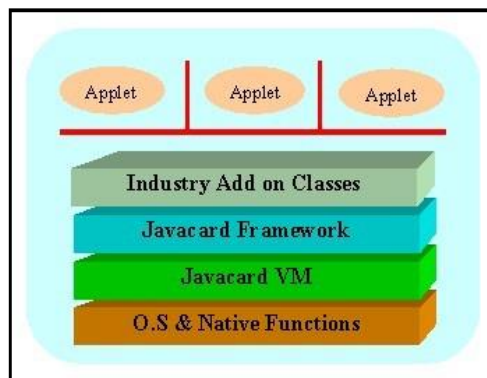
A Java card is a smart card that supports Java program execution. Sun's JavaSoft division launched the Java Card 2.0 API specification for Java programming on smart cards. This API is used in smart cards by many licensees. To create Java Card applications that comply with the 2.0 standard, developers need to have a clear understanding of the card's architecture, key groups and processes, and application development. This article provides guidance to developers about the system architecture, application programming interface (API), and Java platform and is useful for developers. By exploring Java Card vulnerabilities, developers gain the skills needed to develop applications for the platform.

WHAT IS A JAVA CARD?



Java card refers to the smart card that can run Java programs. The Java Card 2.0 specification provides general guidance for creating application programming interface (API) for smart cards. Minimum specifications for the Java Card include 16 KB ROM, 8 KB RAM. Below is an example of a Java card.

SYSTEM ARCHITECTURE CARD



(VM) that runs on anis always running. The JVM layer acts as an intermediary by providing a language and communication protocol that hides the developer's technology. The Java Card framework contains API classes for building applications and accessing services. Businesses or businesses can provide additional libraries to enhance services and achieve security and performance standards. Java Card applications (called applets) can

card, each card is covered by ISO 7816, as part of this it is important to understand that smart cards (such as Java Cards) are not equivalent to personal computers. Java Card 2.0 should not be considered a removal of the Java Development Kit (JDK).

THE LIFETIME OF A JAVA CARD

The lifecycle of a Java card begins with writing the native operating system, class library, and selected applications to read-only memory (ROM). Java Cards must be initialized and personalized before use. Initialization involves loading general information onto the memory card without any glitches. This information is not consistent across multiple cards and is not specific to an individual, such as the name of the manufacturer or producer. office, the continuously without interruption. To achieve this, the Java Card VM provides an electronic, to store persistent data. The full is equal to itself. When power is off, loop.

The lifecycle of a Java card begins with writing the native operating system, Java Card VM, API class library, and selected applications to read-only memory (ROM). Java Cards must be initialized and personalized before use. Initialization involves loading general information onto the memory card without any glitches. This information is not consistent across multiple, such as the name of the manufacturer or producer.

Office, the continuously without interruption. To achieve this, the Java Card VM provides an electronic, to store persistent data. The full equal to itself. When power is off the VM runs in an endless loop The lifecycle of an application begins when the application is installed and registered in the login system, and deleted from the registry. Whether the address in deleted applications can be used again depends on the garbage usage on the card. Applications will remain disabled until selected from the terminal. Objects environment are created in non-volatile memory (such as EEPROM). If these products are not regularly transferred from other products, they may be at risk of loss or wastage. But (about a thousand times) than writing to Some items are entered multiple times and their contents in the field do not need to be carefully checked. To achieve this, the Java Card supports the concept of mutable objects residing in RAM.

JAVA CARD 2.0 LANGUAGE SUBSET

Java and can be compiled using a standard Java compiler. The following features are not included :

Therefore, the keyword related to this feature has been removed from the Java Card language. However, it is worth noting that VM users may choose to support additional features of the version after release, such as the 32-bit code format or traditional methods. This change is available on more smart cards with more memory resources.

THE JAVA CARD 2.0 FRAMEWORK

We looked at ISO 7816. What is standard series with additional sector. ease of use. and provides a simple and understandable programming interface.

JAVA CARD SECURITY

Java applications but the Java Card system's security standard differs from the Java standard in several ways. security classes. The security code is used by virtual machines. Java applications belongs to the application that created it. Although an application can refer call methods on the object unless it contains the object or explicitly specifies the object. An application can share one of its features with a specific application or with all applications. component in a Java application. Its options, execution and operation applications.

HOW THINGS WORK TOGETHER INSIDE A JAVA CARD

In (JCRE) classes provided by application Card is associated with a unique Application ID. Once the application is successfully installed on the non-volatile memory card and linked card libraries, JCRE starts the installation process of the application. As the final step of installing an application, JCRE calls the application's installation method. This method should be implemented as in the application class. Responsible for creating the application and registering it with JCRE. This ensures efficient memory usage and performance of applications environment.

DEVELOPING A JAVA CARD APPLLET

Once the created, the next step is to prepare it that supports This preparation process includes a few simple steps to ensure the app works properly. This step involves converting the application to the appropriate

operating mode and thoroughly testing it in a different simulation environment. These tests help analyze the compatibility, functionality, and application ecosystem. By completing these necessary preparations, you can ensure that your smart card distribution, provides the required functionality and complies with the specifications.

II. USING THE JAVA CARD DEVELOPMENT KIT

These components and tools include:

Java Card Framework class: This class is required to create Java Card applications. They provide the necessary functions and APIs for small-scale development. Allows developers to test and evaluate their applications in a simulated environment.

APDUTool Utility: This utility helps send APDU (Control Panel Data) commands to JCWDE. Command APDU is used to provide a service request for the smart card.

Converter Tool: The used to convert Java Card applications to the format required for preservation or installation. For masking, implementation class and package classes are converted into archives. When installed, the application and its components are converted into a Off-Card Verification Tools: These tools verify the authenticity of the data generated by the Converter tool. They ensure that converted files are valid and suitable for export.

Mask Generator: The mask generator creates mask files that can be added to masks in.

Non-card installer: Non-card installer is used to install Java Card applications on smart cards. It facilitates the deployment of applications to physical smart cards. Using these groups and tools developers can track specific improvements. They compile the application, optionally test and debug it in the JCWDE environment, and then proceed to convert the application to the appropriate format. The converter tool creates JCA or CAP files depending on the intended use case (protection or installation) and can create export files if a package is required. This information may be further processed or transmitted as necessary for the development and distribution of the Java Card. The diagram below shows.

COMPILING A JAVA CARD APPLLET

When you create Java Card applications, you write them using language. However, due to the limited storage space of smart cards, applications are coded using special techniques optimized for the smart card environment. To compile the Java

board application, you can use versions 1.2.2, 1.3, or later.

A commonly used Java compiler is "javac" The compiler includes Java 2 SDK 1.3 or later. Don't forget to include the "api21.jar" class file before calling the compiler to ensure a successful compilation. This JAR file contains the Java Card API classes required for application development.

TESTING A JAVA CARD APPLLET IN THE JCWDE

Testing an application may be an optional step and may occur at a later stage in the development process, such as converting the application into a block or configuration mode. early testing of your application, you can choose to test. This feature allows you to use the Java Virtual Machine to simulate the execution on a desktop application to a specific format. This provides an easy way to check and verify the functionality of the application before moving on to the next step

STARTING THE JCWDE

Small tasks to be executed as if hidden inside a read-only smart card. Install the application on the real smart card.

The input for thisspecifies one or more applications to add. These applications are then wrapped in a JCWDE environment and simulated to exist in a read-only smart card execution environment.

In the profile, the application is identified by its unique application identifier (Aid). Once the application is loaded the environment can make direct requests in the form of APDUs to the required applet for further processing. This allows you to interactively test and verify your application's behavior and functionality in the simulated Java Card environment provided by JCWDE.

CONVERTING A JAVA CARD APPLLET

the process of integrating the Java Card application into the mask or downloading the application to the smart card after production is not fair. Instead, many steps need to be taken.

The first step for preservation is to convert the implementation class and all package into archives. This JCA file contains the bytecode and metadata needed by the application and related classes. When the JCA archive is created, it is converted, along with other JCA archives from other packages.

This is a modification made specifically for the mission environment and then attached to the mask. Mask refers to the non-volatile where applications and related files are permanently stored.

Similarly, when creating a smart card, downloading the Java Card application to the card does not occur directly. . Instead, the application and its necessary files are converted to a format suitable for installation on the smart card. These conversion formats, such as conversion application (CAP) files, are used in the installation process.

Follow the steps below and the application and its associated files will be properly converted and ready for deployment on the smart card, ensuring compatibility with the target complementing the runtime environment and compliance with the

RUNNING THE OFF-CARD INSTALLER

In Java Card , "scriptgen" command to complete the card extraction process. This command runs scripts on Solaris workstations or batch files on Windows NT platforms. The input of the command is the path to the conversion application (CAP) file to be loaded onto the smart card.

By issuing the "Scriptgen" command and taking the input, the board loader creates application. By default the installer redirects the to provide you with appropriate instructions.

However, the "scriptgen" command also has the option to specify output file names. This allows you to save design settings in a special file for future use or for automation purposes.

By using the 'Scriptgen' command and providing the CAP file as input, script for a Java Card application that can be loaded using appropriate equipment and procedures specific to the intended environment.

III. CONCLUSION

A versatile platform for various applications in the smart card industry. It can be used in ID cards, medical cards, debit cards, electronic wallets and other applications. The simplicity allows multiple applications to be combined on a single card, called multi-Java Card applications.

The launch of the J C expands the application of Java into the field of smart cards. However, commercialization and further development and localization of applications and tools for widespread use and distribution require time. However, as the demand for smart solutions continues to grow, the number of Java Cards in circulation is expected to grow exponentially in the coming years.

The simplicity and portability of J C technology means that personal data storage and downloading applications can be done with ease. Use a small card to carry in your wallet or purse for security checks. As technology continues to evolve, Java Card has the potential to revolutionize the industry and improve the functionality and security of smart cards.

IV. REFERENCE

- [1] <https://www.devotion.greenvest.co.id/index.php/dev/article/download/416/799>
- [2] <https://en.wikipedia.org/wiki/InternecoCThings>
- [3] <http://www.w3schools.com/html/htmIntro.asp>
- [4] <http://www.cisco.com/web/solutions/trendslit/overview.html>
- [5] Usman, Uke k. (2021). Mengenal Teknologi 5g. 2017, 345–348.
- [6] Wijanto, Eddy. (2017). Analisis Kesiapan Teknologi Dalam Penerapan Teknologi Telekomunikasi Generasi Kelima (5g). Jurnal Teknik Dan Ilmu Komputer.
- [7] Wijaya, Anggie. (2021). Perkembangan Teknologi 5g. Universitas Pendidikan Indonesia, 1(1), 2–5.