4.7 JAVA CARD

Java Card is a smart card with Java framework. Smart card was developed in 1974, by Roland Moreno. Smart card is a plastic card with intelligence and memory. Smart cards are becoming popular as identity module and wireless security devices. In many countries driving licenses are being issued on smart cards. The SIM card on a GSM mobile phone is a smart card as well. The importance of smart card made ISO to standardize all its interfaces. These are done through ISO 7816 standards. These ISO standards define the physical characteristic of the card (ISO 7816-1: Physical Characteristics), locations and dimensions of the contacts (7816-2:: Dimensions and Locations of the Contacts), signals and transmission interfaces (7816-3:: Electronic Signals and Transmission Protocols), and command interfaces (7816-4:: Interindustry Commands for Interchange). A smart card is embedded with either (i) a microprocessor and a memory chip or (ii) only a memory chip with non-programmable logic. A microprocessor card can have an intelligent program resident within the card which can add, delete, and otherwise manipulate information on the card. A memory card on contrast, can store some information for some pre-defined operation. Smart cards are capable of carrying data, functions, and information on the card. Therefore, unlike memory strip cards, they do not require access to remote databases at the time of the transaction.

Microprocessor based smart cards which used to be used for some specific application areas are becoming quite common. Smart cards have now emerged as multi function cards. To allow interoperability, Java was chosen as the vehicle for interoperability. All the microprocessor based smart cards now offer Java API framework on the smart card. This is why smart cards with Java framework are also called Java Cards. 3GPP has decided to use Java Card as the standard for USIM and ICC (Integrated Circuit cards). Java Card technology preserves many of the benefits of the Java programming languages such as: productivity, security, robustness, tools, and portability. For Java card, the Java Virtual Machine (JVM), the language definition, and the core packages have been made more compact to bring Java technology to the resource constrained smart cards.

A smart card of a GSM SIM card supporting Java Card functionalities may typically have 8 or 16 bit microprocessor running at speed between 5 MHz to 40 MHz with 32K to 128K bytes of EEPROM (Electronically Erasable Programmable Read Only Memory). Though Java card works in a master/slave mode, using the proactive SIM technology of GSM Phase 2+, it is possible for the application on the SIM card to get activated in an automated fashion. Also, Java card technology supports OTA (Over The Air) downloads. In OTA download, a Java applet (through SMS) can be downloaded by the network operator proactively or by the user interactively over the wireless media. Applications written for the Java Card platform are referred to as applets.

The development framework in Java card is different from that on a desktop computer. The major challenge of Java Card technology on smart card is to fit Java system software in a resource constraint smart card while conserving enough space for applications. Java Card supports a subset of the features of Java language available on desktop

computers. The Java Card virtual machine on a smart card is split into two parts (Figure 4.13): one that runs off-card and the other that runs on-card. Many processing tasks that are not constrained to execute at runtime, such as class loading, bytecode verification, resolution and linking, and optimization, are dedicated to the virtual machine that is running off-card where resources are usually not a concern. Thle e on-card components of Java Card include components like the Java card virtual machine (JCVM), the Java card runtime environment (JCRE), and the Java API. Task of the compiler is to convert a Java source into Java class files. The converter will convert class files into a format downloadable into the smart card. Converter ensures the byte code validity before the application is installed into the card. The converter checks the classes off-card for,

- Well formedness
- Java Card subset violations
- Static variable initialization
- Reference resolution
- Byte code optimization
- Storage Allocation.
- The java card interpreter
- Executes the applets
- Controls run-time resources
- Enforces runtime security

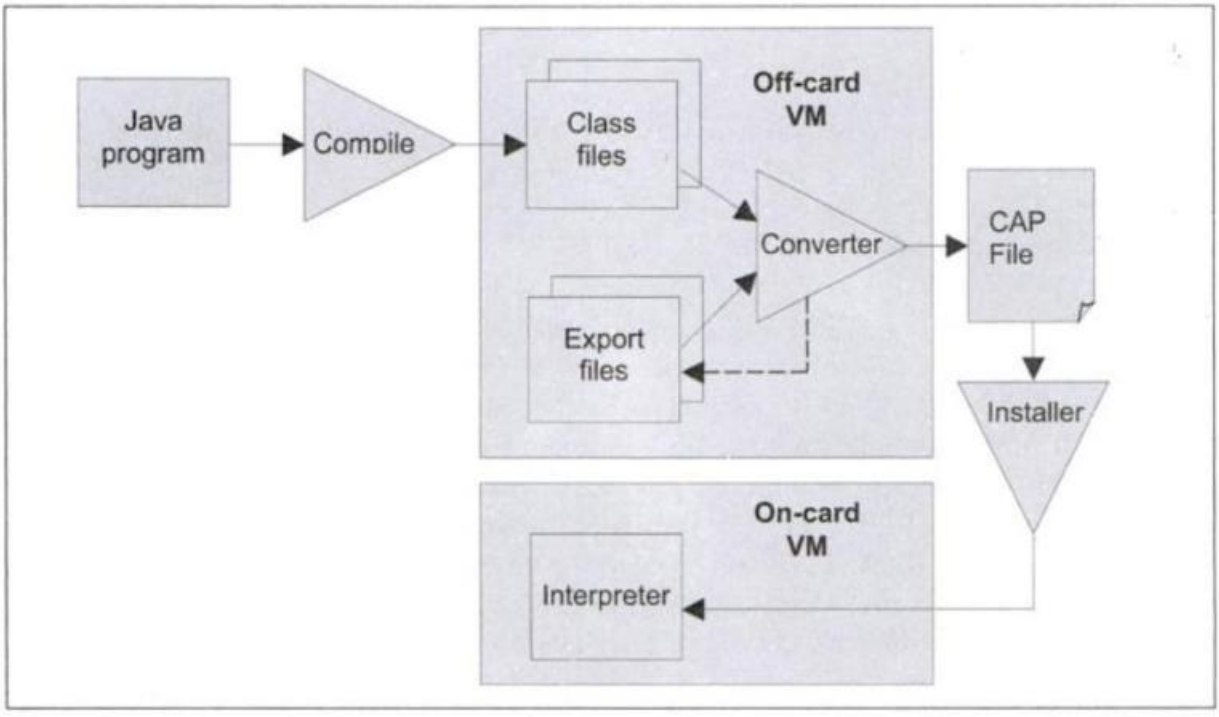

Figure 4.13 Architecture of Java card applications development process

Following conversion by the off-card VM into CAP (Converted APlet) format, the applet is transferred into the card using the installer. The applet is selected for execution by the JCRE. JCRE is made up of the on-card virtual machine and the Java Card API classes. JCRE performs additional runtime security checks through applet firewall. Applet firewall partitions the objects stored into separate protected object spaces, called contexts. Applet firewall controls the access to shareable interfaces of these objects. The JCVM is

a scaled down version of standard JVM (Java Virtual Machine). Elements of standard Java not supported in JCVM are,

- Security manager
- Dynamic class loading
- Bytecode verifier
- Threads
- Garbage collection
- Multi dimensional arrays
- Char and strings
- Floating point operation
- Object serialization
- Object cloning

As mentioned above Java applications for a Java Cards are called Applets. Java Card applets should not be confused with Java applets on the Internet. A Java Card applet is not intended to run within an Internet browser environment. The reason for choosing the name applet is that Java Card applets can be loaded into the Java Card runtime environment after the card has been manufactured. That is, unlike applications in many embedded systems, Java Card applets do not need to be burned into the ROM during manufacture.