ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 8, No. 1, 2023

ENHANCING SIM CARD TECHNOLOGY: DEVELOPING AN ADVANCED OPERATING SYSTEM

Busra Ozdenizci Kose[1], *Alp Sardag*[2], *Vedat Coskun*[3] *and Haci Ali Mantar*[4]

[1, 4] *Gebze Technical University, Kocaeli, Türkiye*
[2] *Omreon, Istanbul, Türkiye*
[3] *Atlas University, Istanbul, Türkiye*
Authors' e-mails: [1] *busraozdenizci@gtu.edu.tr;* [2] *alpsardag@omreon.com;*
[3] *vedatcoskun@atlas.edu.tr,* [4]*hamantar@gtu.edu.tr*

https://doi.org/10.23939/acps2023.01.035

Submitted on 26.04.2023

***Abstract*: The emergence of smart cards is a significant achievement in the field of information technology. They are growing in importance as they offer secure data storage and authentication, improve security, simplify processes, reduce costs, and have a variety of applications. In particular, nowadays SIM cards play a vital role in the Telecom industry as an important type of smart card that allows a user to make calls, send messages, store details such as phone number and network authentication information, and provide various mobile data services. Currently, some leading companies take advantage of having their own smart card operating systems and have significant impact on the market on various issues (hardware on which the smart card operating system will run; security restrictions; control, permission and control of any application, etc.). It is important to work on creating a more flexible, adaptable and secure operating system infrastructure in order to overcome these limitations both in the customer perspective and in the ecosystem. The purpose of this research is to introduce the creation of a special Java Card operating system designed for SIM cards used in the Telecom industry. The study presents the software requirements and development steps of the proposed operating system and innovative architectural components necessary for its operation as a SIM card beyond being a Java Card operating system. The proposed new Java Card based SIM operating system is expected to pave the way for the development of eSIM operating system infrastructure as well as 5G technology and offer valuable opportunities.**

***Index Terms*: smart card; SIM; Java card; operating system; system architecture.**

I. INTRODUCTION

The advent of smart cards represents a significant accomplishment in the realm of information technology. They are becoming increasingly important due to their ability to provide secure data storage and authentication, enhance security, streamline processes, reduce costs, and be used in a wide range of applications [1–3]. Smart cards are small plastic cards that contain embedded microchips that can store and process data. Smart cards have proven to be a reliable and secure way to store and transmit data, and they can be used in a wide range of industries and applications, including banking, healthcare, transportation, loyalty programs and government [3–5]. They can also be used in combination with other technologies such as biometric authentication, making them even more versatile.

Basically the lifecycle of smart cards involves chip design and manufacturing, card manufacturing, personalization, issuance, usage, renewal or replacement, and end of life [1, 3]. The smart card operating system (OS) plays a crucial role in the lifecycle of a smart card by providing a secure platform for running applications and managing resources [3, 6–8]. A smart card OS as a software platform, runs on a smart card's embedded microchip and provides the ability to perform various functions, including data storage, encryption, authentication, and processing. It serves as the intermediary between the card's hardware and the applications that run on the card. It manages the card's resources and provides a secure environment for running applications, ensuring the confidentiality, integrity, and availability of the data stored on the card. It facilitates the development of a wide range of applications and services, such as payment systems, identification and access control systems, and secure storage of personal data.

The smart card OS provides a secure environment, a secure element for storing and processing data on the card, secure authentication using such as PINs, passwords, or biometric data, and cryptographic functions for enabling secure communication and data transfer [3, 9]. It is designed to protect sensitive information from unauthorized access and tampering, providing the card with the ability to securely perform a range of functions, such as electronic payment, access control, and healthcare applications. Also, the smart card OS manages multiple applications running on the card, ensuring that they are isolated from each other and preventing interference between them [6–8]. Some of the commonly used smart card operating systems include Java Card, MULTOS, Global Platform, and Microsoft Windows for Smart Cards. Each of these operating systems has its strengths and weaknesses, and selection of a smart card OS depends on the specific requirements of the application and the card.

Particularly, a SIM (Subscriber Identity Module) card as a smart card is a fundamental and indispensable component of Telecom ecosystems. SIM cards store a subscriber's information, including their phone number and network authentication credentials; and enable users to make calls, send messages, and access various mobile data services [10, 11]. A SIM card is a critical component of mobile communication, providing essential identification, security, and service provisioning features that enable subscribers to access mobile services and stay connected while on the go. Basically, the functional use of a SIM card is after the following stages: (a) firstly, a smart card is produced as hardware, (b) secondly, an OS is loaded on the hardware, (c) finally, applications are loaded on the OS depending on the purpose of use, which are called applets specific to Java Card.

Recently, companies such as Thales, Giesecke & Devrient and NXP dominate the market by using the advantages of owning their own smart card OS; they determine the smart card hardware, OS and other features that will be introduced. Depending on the chip that these manufacturers recommend to their customers, the entire market uses the smart cards with the OSs of these companies together with the relevant chip. In addition, customer companies do not use smart cards according to their own wishes, but within the framework of the features offered to them. At the same time, the security content of the smart cards cannot be questioned. Companies can only have information about the smart cards they have purchased, limited to their own applications. The control, permission and control of any application that can run on a smart card can be done through these companies. In order to solve all these constraints within the ecosystem and to meet customer requirements, it is aimed to develop a more flexible and secure OS.

This study aims to present development of a unique Java Card OS that will work on the SIM cards used in the Telecom industry. After successfully completion and certification of Java Card based SIM OS, a company that needs a SIM card will be able to purchase smart card hardware that does not have an OS installed on it, that is, it is sold only depending on its hardware features, and many manufacturers sell it at a much cheaper price due to competitive conditions.

The rest of this paper is organized as follows: Section II presents the methods including the design considerations for smart card OS for SIM cards. Section III explains the development studies. Finally, the study is concluded in Section IV.

METHODS

The proposed Java Card OS, which will run on SIM cards to be used by mobile operators, will be developed in accordance with Oracle JCVM standards. There are certain international standards that must be compatible in order for the Java Card based SIM OS to be used.

First of all, a prototype of Java Card OS was developed for this study. In the first prototype, all modules indicated in Fig. 1 were developed and certified successfully – as it is shown in Fig. 2 – completing the Java Card Technology Compatibility Kit (TCK) tests of Oracle company. Java Card TCK is a portable, configurable automated test suite for verifying the compliance of an implementation with the applicable Java Card specification.

By using the prototype, additional, required modules in accordance with the mentioned Global Platform and ETSI standards will be analyzed and designed in detail; so that SIM Applets can run on the new, enhanced prototype OS version.

After the design work, microcontroller emulator, test cards and test program will be procured from suitable suppliers. Performance and stability tests of the test cards will be carried out with the provided emulator and it will be checked that these parts work in accordance with our development purposes.

Next, the required, designed modules on the Java Card OS High Level Architecture will be developed. As it is shown in Figure 1, the bottom layer that will be developed and run on Smart Card Hardware is the Micro Kernel module. This module includes I/O module, memory management module, encryption (crypto) module, module manager as well as basic libraries. It is only this part that is likely to change in terms of hardware; other codes will remain the same on different chips. Since the layered architecture will be used, the first mentioned layer above the hardware will be hardware dependent code. This part will be developed in Assembly and C language.

Above this layer, the Java Card Runtime Environment (JCRE) exists which includes ByteCode Interpreter and Applet Firewall. Java Card applications loaded with Global Platform standard commands will be able to run precompiled Java Card Applets with the developed "Bytecode Interpreter" module after they are connected to the Java Card API [12] addresses. Applet Firewall is a security module that allows applets to separate their memory space from each other. Thus, each applet runs only in its jurisdiction. In this way, no applet can access the area belonging to another applet. Again, in this section, there is a secure element containing sensitive data such as PIN and key, and access to this area is under the security control of Applet Firewall.

Native interface is the layer where C native methods are defined (function array) that can be called from the Java Card layer in JCRE. Since the Java Converted Applet (CAP) offered by Oracle does not allow the use of native functions, Oracle CAP converter software will be adapted so that the packages in the Java layer of the proposed OS can use native functions. In addition, the necessary multi-thread support will be added to implement the "context switch" mechanism in the ETSI TS 101 476 standard in JCRE. Running such a structure on a limited hardware resource requires an advanced software architecture.

The modules above the JCRE layer will be coded in Java language. Both modules of the OS such as Card Manager and third-party applications are in this layer. It is aimed to facilitate chip migration by moving as many modules of the OS as possible to the Java layer. At the bottom are the SIM Toolkit and Java Card core APIs standardized by Oracle. Basic crypto commands, basic language packages, classes, methods and security library are in this area.

On top of Java Card's core APIs, Card Manager application exist which have been determined by the Global Platform. Card Manager is the application that securely manages the smart card content in general and the life cycle of third party applications (installation of the application, activation, deletion, etc.). Card Manager is a Java Card applet in its own right, and because it is a part of the OS and has many privileged features on the card. Card Manager and utilities (Package/Applet Registry, Crypto, Loader, Deleter etc.) used by Card Manager will be developed.

Next to Card Manager, there are Java applets that provide SIM card master and remote management functionality on the same layer. The obligatory GSM/USIM applets, RAM/RFM applets are on this layer and will also be developed.

Apart from this, applets developed by third parties will be uploaded to this layer and run securely remotely or Over-the-Air (OTA) after the SIM card is given to the subscriber.

After the completion of the proposed OS development, Java Card Technology Compatibility Kit (TCK) and SIM API tests and certifications will be completed.

II. SYSTEM DEVELOPMENT

Essential developments to be made on the first prototype of Java Card OS are defined as follows: software and hardware configurations; Java Card Operating System “Native API Call”, file system that meets ETSI TS 151 011, ETSI TS 101 220, ETSI TS 102 221, ISO/IEC 7816 standards; Card Manager applet; new GSM and RFM applet enhancements; GSM SIM Toolkit libraries conforming to ETSI TS 102 241 standard; SIM card event based applet triggering; adding multithreading feature; native additional function development: detailing additional commands; Oracle Converter customization for GSM file system and mask generation; multiple applet lifecycle management and garbage collector; mobile network authentication.

Currently the architectural components mentioned have started to be developed on the first prototype system. Some of the improvements are presented in detail below.

A. SOFTWARE AND HARDWARE CONFIGURATIONS

Primarily, SIM applications (SIM Toolkit Applet) to be installed on the Java Card OS must provide the necessary infrastructure to operate in accordance with some important specifications. These specifications are listed hereunder:

- ETSI TS 151 011: Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface.
- ETSI TS 102 225: Secured packet structure for UICC based applications.
- ETSI TS 102 221: UICC-Terminal interface; physical and logical characteristics.
- ETSI TS 102 223: Smart Cards; Card Application Toolkit.
- ETSI TS 101 220: ETSI numbering system for telecommunication application providers.
- ETSI TS 102 241: UICC Application Programming Interface for Java Card.
- ETSI TS 102 222: Integrated Circuit Cards; Administrative commands for telecommunications applications.
- ETSI TS 123 048: Security mechanisms for the (U)SIM application toolkit.
- ETSI TS 101 476: GSM API for SIM toolkit stage 2.
- GSM 11.14 : (SIM – ME) Interface.

Accordingly, function set and specifications in Java and Native C layers that will support these requirements; data, algorithm and messaging requirements for Java and Native C layers of the OS for SIM applications are determined.

As an important issue, in order for the OTA managed applets to work safely, accurately and using limited resources on the SIM, the following areas must be supported by the proposed OS: concurrency, event based application triggering, flash management and authentication on the network.

Within the scope of this work, the developed first prototype of the Java Card OS runs on HUADA’s CIU98428F MCU. This MCU is especially produced for Java Card based SIM OSs used in the Telecom industry. During the development phase of our Java Card OS, HUADA's MCU emulator with JTAG interface was used for debugging, which is shown in Figure 3.

For the development of the Java Card OS, three different programming languages (Assembly, C and Java) are used. Developments are being made on two different IDEs for these three different programming languages: (1) Keil Uvision version 5.24 for assembly and C programming languages, (2) Eclipse version Luna for Java programming language.

B. ARCHITECTURAL COMPONENTS

As it has been mentioned, in the first prototype of Java Card OS, the basic modules and architectures in Fig. 1 were developed and certified successfully. The modules and libraries that need to be added in Native C and Java layers in order for the Java Card OS to fulfill the requirements of the SIM card OS, are shown in gray boxes in Fig. 4. The significant architectural components of the proposed Java Card based SIM OS are described hereunder:

1) *Native API Call:* Java Card applications developed by third parties cannot use the "native API call" feature. The "Javac" compiler and the "CAP converter" converter provided by Oracle do not have this feature. Development should be done in the bytecode interpreter module in order to be able to call functions in the Native C layer from the layer written in the Java programming language with the "Native API Call" method. In addition, a new converter is currently being developed using the "Javac" compiler and CAP converter tools. Thanks to this new Converter, native functions can be compiled from coding in the Java layer and converted into a form that the bytecode interpreter can interpret.

2) *GSM Applet:* In the normal Java Card OS, the OS stands up with the Card Manager application selected by default. This is because when standard cards (e. g., credit/debit cards, meal cards, transportation cards) are processed with POS/ATM machines, the relevant application on the card OS is first selected by the POS/ATM and after this selection step, no action is taken on all of the other commands sent. It is directed to the applet selected in the first command by the OS without any action. It is seen that the default selected application for SIM cards must be GSM Applet. Because instead of selecting an application on the SIM card while the phone is booting, it selects the first GSM files in accordance with the GSM specifications and tries to connect to the network according to the configuration information in these files. The proposed GSM Applet handles the first incoming file operations, and when it does not recognize commands such as applet selection, it can direct them to the Card Manager Applet. Owing to this method, the first commands are processed quickly without the need for applet changes, and it is aimed to reduce the error rate that may occur in the code, since minimum update is required for the Card Manager Applet that has been developed.

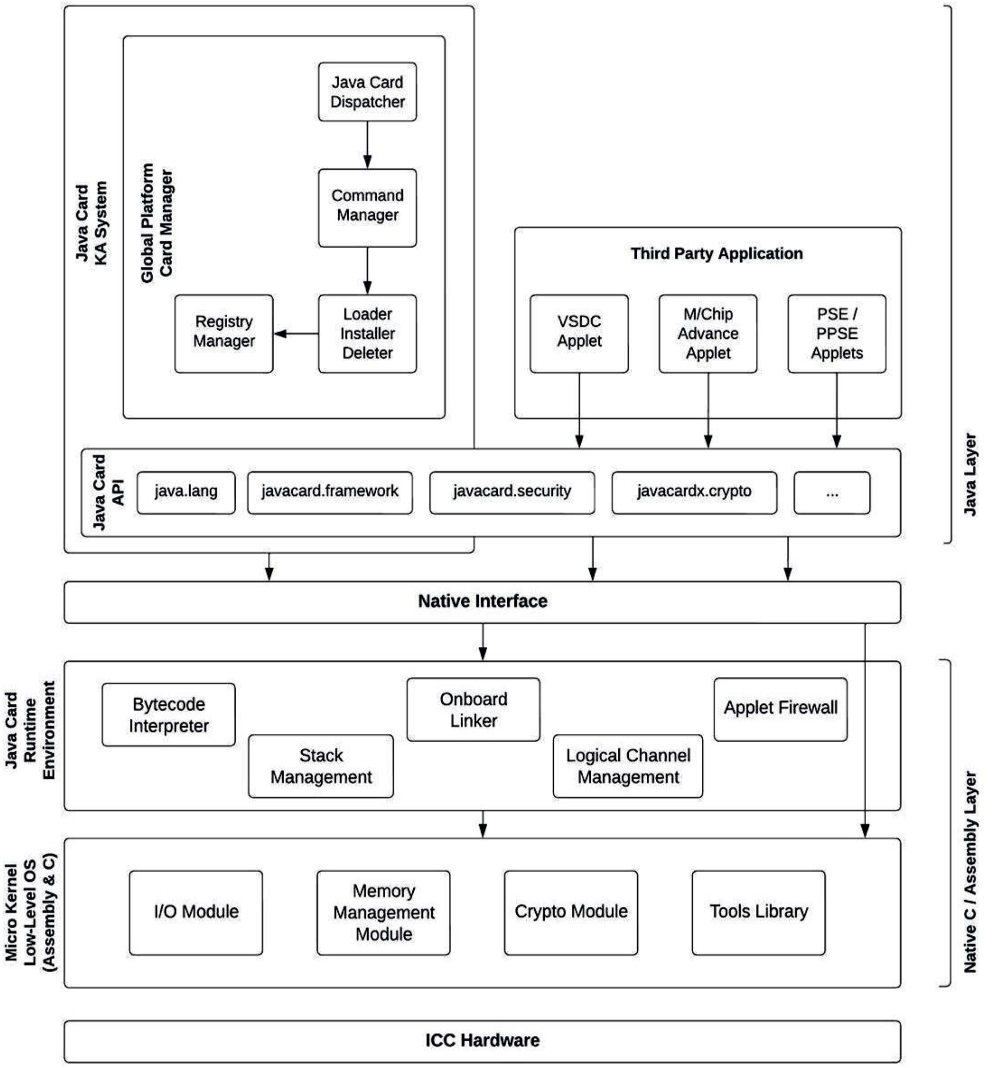
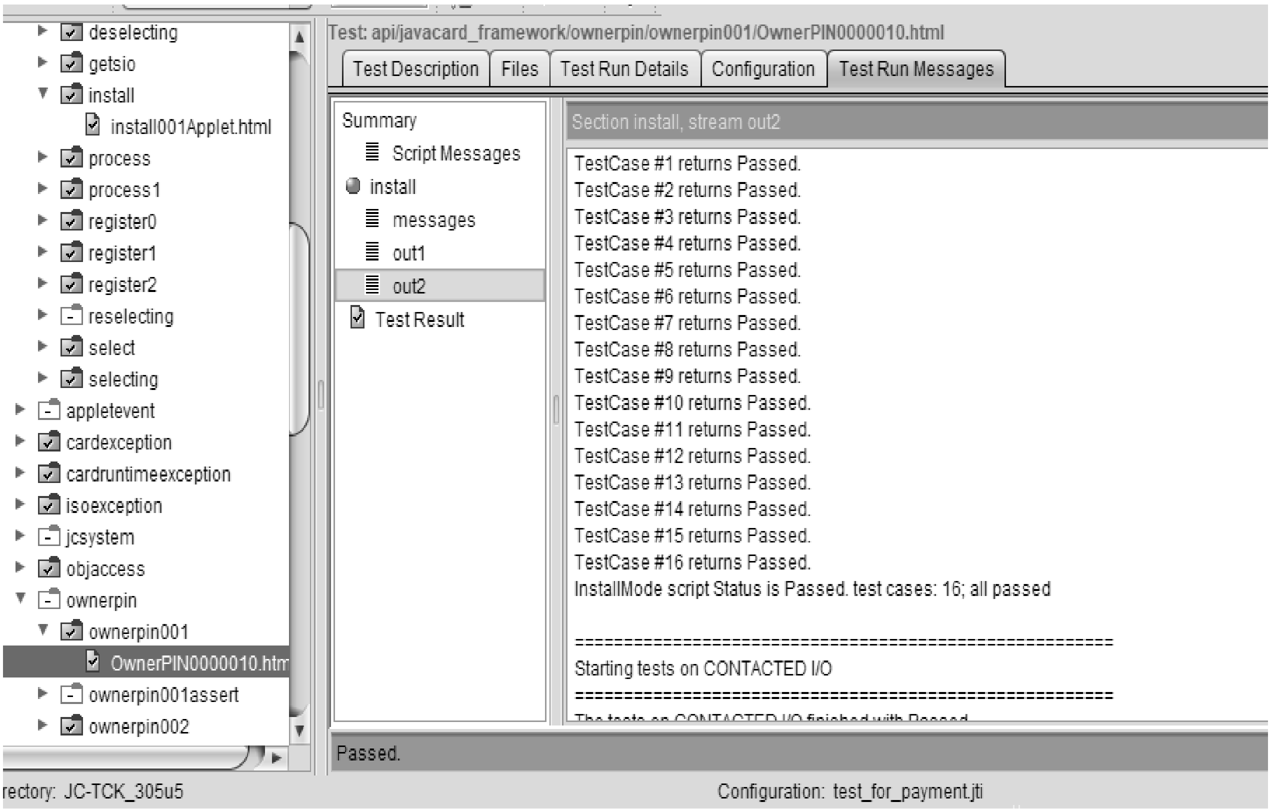


Fig. 1. Java card operating system high-level architecture



```
▸ ☑ deselecting
▸ ☑ getsio
▾ ☑ install
    install001Applet.html
▸ ☑ process
▸ ☑ process1
▸ ☑ register0
▸ ☑ register1
▸ ☑ register2
▸ ⊟ reselecting
▸ ☑ select
▸ ☑ selecting
▸ ⊟ appletevent
▸ ☑ cardexception
▸ ☑ cardruntimeexception
▸ ☑ isoexception
▸ ⊟ jcsystem
▸ ☑ objaccess
▾ ⊟ ownerpin
  ▾ ☑ ownerpin001
      OwnerPIN0000010.htm
  ▸ ⊟ ownerpin001assert
  ▸ ☑ ownerpin002
```

Test: api/javacard_framework/ownerpin/ownerpin001/OwnerPIN0000010.html

Section install, stream out2

```
TestCase #1 returns Passed.
TestCase #2 returns Passed.
TestCase #3 returns Passed.
TestCase #4 returns Passed.
TestCase #5 returns Passed.
TestCase #6 returns Passed.
TestCase #7 returns Passed.
TestCase #8 returns Passed.
TestCase #9 returns Passed.
TestCase #10 returns Passed.
TestCase #11 returns Passed.
TestCase #12 returns Passed.
TestCase #13 returns Passed.
TestCase #14 returns Passed.
TestCase #15 returns Passed.
TestCase #16 returns Passed.
InstallMode script Status is Passed. test cases: 16; all passed

=======================================================
Starting tests on CONTACTED I/O
=======================================================
The tests on CONTACTED I/O finished with Passed
```

Passed.

rectory: JC-TCK_305u5

Configuration: test_for_payment.jti

Fig. 2. Oracle TCK Java Card operating system test tool

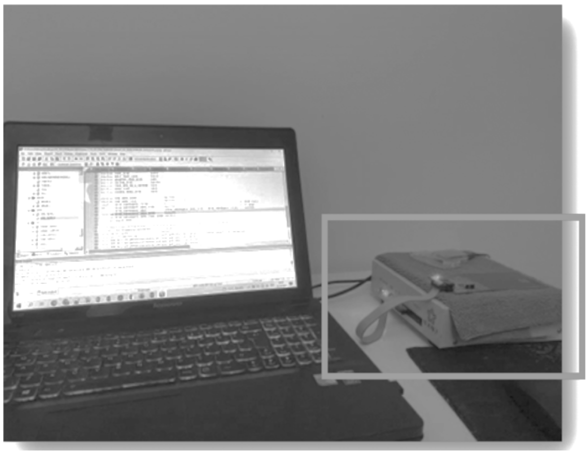


Fig. 3. HUADA's MCU emulator

3) *RFM (Remote File Management) Applet:* Another important component currently under development is the RFM Applet. The main functions of the RFM Applet are to receive the secure command and remote transmission protocols specified in the ETSI TS 102 225 standards, to run the file update operations on the file system in the order determined by the GSM operators, and to send the results as a short message to the GSM operator.

4) *File System Manager:* File system must meet ETSI TS 151 011, ETSI TS 101 220, ETSI TS 102 221, ISO/IEC 7816 standards. The File System Manager component will be organized for secure file management as well as secure and recognized verified applications. Since GSM file system and Java Card applications will be located on the same NVM (Non-Volatile Main Memory) Heap area, and new files will be added/deleted frequently or Java Card applications will be added/deleted via OTA; it has been found appropriate to use the same Garbage Collector algorithm for the application and the file system. Therefore, the same data structures as the Java Card objects in the prototype will be used in the files and directories of the GSM file system.

5) *GSM SIM Toolkit Libraries:* SIM Toolkit Applets registered to GSM events specified in the ETSI TS 102 241 specification should be triggered by the OS according to their priorities. The proposed SIM OS requires two features that differs from this standard: (1) Keeping priority information of applets in the "registry" table, unlike the standard Java Card OS; (2) Event registration and priority-based applet selection and execution. The proposed SIM OS has to be able to run applets running on a SIM card simultaneously. For this reason, the "Context Switching" or "Priority Based Context Switching" algorithm, which will be one of the important outputs of this study, is currently being developed. The context switch to be developed will provide a very fast response to the requests of the person connected to the network, such as speaking, receiving SMS, and emergency calls.

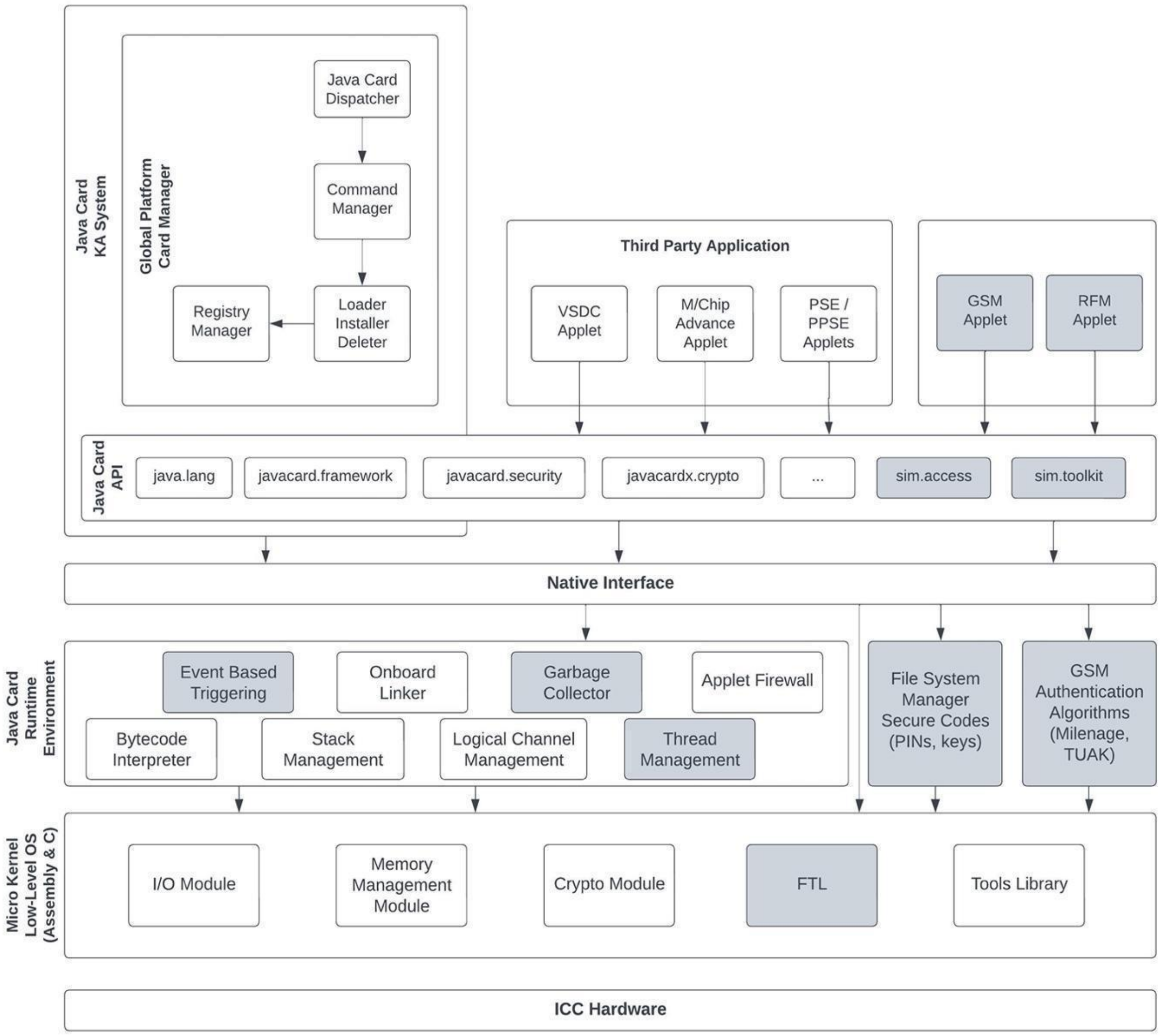


Fig. 4. High-end architecture of the proposed Java Card based SIM operating system

6) *Garbage Collector:* Applets from many different developers are running on a networked SIM card. The lifecycle of each applet can be managed remotely: (1) many operations (e. g., version update, configuration parameters management, adding, deleting) should be done without interrupting the device; (2) after applet erase/write operations with different sizes over time, the disk space should be integrated so that the unreferenced areas can be reused; (3) applets – that are distributed in different storage areas according to the defragmented structure – should be able to work with performance; and (4) garbage collection structures should be constructed. The preparation of all these structures with the principles resistant to “Wear Leveling” and “Anti Tiering”, as in the first prototype, and with high performance, requires a very efficient algorithmic design. In some OSs, it is not possible to accept the analogy of garbage collection/defragmentation services in the range of hours for SIM cards. Within the scope of this study, it will not be enough to develop a garbage collector and defragmentation routine; versions that can work very quickly and atomically (i. e., ending in seconds) will be developed. Also, defragmentation processes should take up very little space due to space constraints in SIM applications.

7) *Thread Management*: More than one applet should be able to be run together without overflowing each other's fields. Many GSM operations operate asynchronously as per the standards. For example, when an application sends a text message to the network, it takes a certain amount of time for the network to receive and approve this message. After the commands sent to the network by the SIM card applets are prepared, they are sent with the *ProactiveHandler.send()* command specified in ETSI TS 102 241 standard and presented to the applets by the OS. A “context switch” is mentioned in the description of this command in ETSI 102 241. Having a context switch feature is important as it allows other SIM card applications to be processed on the CPU

while asynchronous response is awaited after the transactions are made due to the asynchronous nature of many GSM operations such as SMS sending. As might be expected, this feature (e. g., making context transitions between applets, multithreading) is not available on standard cards and therefore operating systems.

8) *GSM Authentication Algorithms:* 3G/4G Milenage (i. e., 3GPP authentication and key generation functions) mobile network authentication algorithm is being used in the first prototype. For the proposed Java card based SIM OS, in addition, TUAK algorithm will be developed in accordance with ETSI TS 135 231 V12.1.0 specifications. TUAK is a mobile network authentication algorithm based on the SHA-3 (Secure Hash Algorithm) algorithm and an alternative to the Milenage algorithm. Messaging and data structures will be designed for TUAK support.

9) *FTL (Flash Translation Layer):* The most important request of GSM operators for SIM cards is to be able to write more than 500,000 times on the flash memory. The reason for this arises from the need to store some information on flash due to the workflows of the applets as well as many flash erase/write operations of subscribers via OTA on the SIM cards. As a result of some conducted tests with GSM operators, it has been observed that these write and erase operations cause the Garbage Collector algorithm to perform mandatory defragmentation/compaction on the SIM card several times a year. However, it is seen that the selected MCU for this study guarantees only 100,000 erase/write operations, as in almost all MCUs sold as SIM card in the market. Hence, a FTL algorithm will be developed in accordance with the given constraints.

III. CONCLUSION

This study presents the design and development issues of a unique Java Card based operating system for SIM Cards. With this proposed new SIM OS, many innovative modules and features such as memory addressing, context switch, garbage collection and defragmentation, multithreading, secure file management, FTL methods on smart card will be provided.

At the end of the development process, as in the first prototype, Java Card TCK 3.0.5 tests will be conducted on Java Card based SIM OS, and these tests are expected to be 100 % successful. After these tests, it is also aimed to conduct pilot tests in various regions with a local operator with 300 different smartphone models. Moreover, these tests are aimed to be completed with a 100 % success rate.

The proposed unique Java Card based SIM OS will make important contributions to the development of 5G technology and the infrastructure of eSIM OS technology, as well as being an OS that can run on different hardware structures.

Acknowledgement

This work is funded by TUBITAK (The Scientific and Technological Research Council of Turkey, www.tubitak.gov.tr/en) under TUBITAK project Grant Number 1507– 7220353.

REFERENCES

[1] Mohammed, L. A., Ramli, A. R., Prakash, V., & Daud, M. B. (2004). Smart card technology: Past, present, and future. International Journal of the Computer, the Internet and Management, 12(1), 12–22.

[2] Markantonakis, K., & Mayes, K. (2003). An overview of the Global Platform smart card specification. Information Security Technical Report, 8(1), 17–29. DOI: 10.1016/S1363-4127(03)00103-1

[3] Deville, D., Galland, A., Grimaud, G., & Jean, S. (2003, February). Smart card operating systems: Past, present and future. In Fifth USENIX/NordU Conference.

[4] Coskun, V., Ozdenizci, B., & Ok, K. (2013). A survey on near field communication (NFC) technology. Wireless personal communications, 71, 2259–2294. DOI: 10.1007/s11277-012-0935-5

[5] Ozdenizci, B., Ok, K., & Coskun, V. (2013). NFC loyal for enhancing loyalty services through near field communication. Wireless personal communications, 68, 1923–1942. DOI: 10.1007/s11277-012-0556-z

[6] Eletriby, M. R., Sobh, M., Eldin, A. M. B., & Fahmy, H. M. (2014, June). High performance Java Card operating system. In 2014 Eighth International Conference on Software Security and Reliability (SERE), pp. 30–39. IEEE. DOI: 10.1109/SERE.2014.16

[7] Bouffard, G., & Lanet, J. L. (2014). Reversing the operating system of a Java based smart card. Journal of Computer Virology and Hacking Techniques, 10, 239–253. DOI: 10.1007/s11416-014-0218-7

[8] Mesbah, A., Lanet, J. L., & Mezghiche, M. (2019). Reverse engineering Java Card and vulnerability exploitation: a shortcut to ROM. International Journal of Information Security, 18, 85–100. DOI: 10.1007/s10207-018-0401-9

[9] Ozdenizci, B., Ok, K., & Coskun, V. (2016). A tokenization-based communication architecture for HCE-enabled NFC services. Mobile Information Systems, 2016. DOI: 10.1155/2016/5046284

[10] Koshy, D. G., & Rao, S. N. (2018, September). Evolution of SIM Cards – What's Next? In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1963–1967. IEEE. DOI: 10.1109/ICACCI.2018.8554774.

[11] Ok, K., Coskun, V., Yarman, S. B., Cevikbas, C., & Ozdenizci, B. (2016). SIMSec: A key exchange protocol between SIM card and service provider. Wireless Personal Communications, 89, 1371–1390. DOI: 10.1007/s11277-016-3326-5.

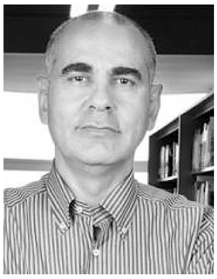
[12] Java Card API, Classic Edition. Available at: https://docs.oracle.com/javacard/3.0.5/api/overview-summary.html

Assoc. Prof. Dr. Busra Ozdenizci Kose received M.Sc. degree at the Department of Information Technology in Isik University and Ph.D. degree at Informatics Department in Istanbul University. She co-authored the books titled ‘‘Near Field Communication: From Theory to Practice’’ published by John Wiley & ion Technologies and Blockchain.

Sons, Inc., 2012 and ‘‘Professional NFC Application Development for Android’’ published by Wrox, 2013. Her research areas include Near Field Communication, Smart Cards, Mobile Communicat

Dr. Alp Sardağ is a Computer Engineer, Entrepreneur, Researcher and Lecturer. He is the founder and director of Metamorfoz ICT delivering innovative R&D projects in the finance and telecom sector. He is working as a consultant for Turkish Scientific Institute (TÜBİTAK) on smart card operating systems for eID, passport and e-signature.

He specializes in and lectures on a wide range of topics including Secure Coding, Embedded Systems, Artificial Intelligence. He also holds several patents on telco products. He is currently researching using AI techniques for breaching and prevention of breaches for secure embedded systems.



Prof. Dr. Vedat Coskun is a Comuter Scientist, Academician, Reseacher, and Author, who founded and directs the NFCLab@Istanbul research laboratory. Dr. Coskun is a Professor of Software Engineering at Istanbul Atlas University, and he has had the privilege of teaching and lecturing at various uniersities around the world. His areas of expertise include Blockchain, Cybersecurity, Payment Ecosystem, and Near Field Communication. Dr. Coskun strongly believes in the importance of bridging the gap between academia and industry in Information Technologies. As a humble Enabler, he has had the opportunity to work with national and international companies, helping them to adopt the latest technological advancements and improve their operations.

Prof. Dr. H. Ali Mantar received his B.Sc. degree in Electronics and Telecommunication Engineering at Istanbul Technical University (1993), received his M.Sc. and Ph.D. degrees in Electrical Engineering (1998) Computer Science (2003) at Syracuse University. He worked as a lecturer at Syracuse University between 1997 and 2000. His research work was supported by the Graduate Students Research in Wide Area Network Management project which was funded by the National Science Foundation (NSF) and was involved in a Computer Resilience Project funded by DARPA, USA (2003–2005). He worked as a Professor of Computer Engineering at Gebze Technical University (2004–2006), Director of TÜBİTAK – BİLGEM (2015–2020), and Istanbul Technical University Vice Rector (2021–2022). He is acting as the Rector of Gebze Technical University.