



eSIM With Secured Context Switching

By

K. M. Muzahidul Islam

(142-35-670)

A thesis submitted in partial fulfillment of the requirement for the degree
of Bachelor of Science in Software Engineering

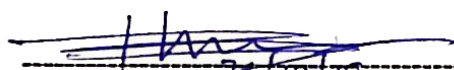
Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Fall – 2018

APPROVAL

This thesis titled on “eSIM With Secured Context Switching”, submitted by **K. M. Muzahidul Islam, (142-35-670)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Prof. Dr. Touhid Bhuiyan
Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman




Mohammad Khaled Sohel
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Md. Shohel Arman
Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2



Mr. Shafiul Hasan
Managing Director
Vivacom Solution, Dhaka

External Examiner

DECLARATION

It hereby declare that this thesis has been done by **me** under the supervision of **Kaushik Sarker, Assistant Professor & Associate Head**, Department of Software Engineering, Daffodil International University. It also declare that nithor this thesis nor any part of this has been submitted elsewhere for award of any degree.



K. M. Muzahidul Islam

Student ID: 142-35-670

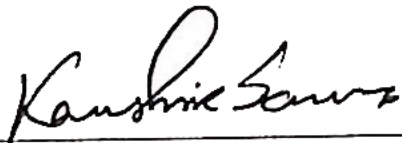
Batch: 14th

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:



Kaushik Sarker

Assistant Professor & Associate Head

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

ACKNOWLEDGEMENT

This report exhibits my Bachelor of Science Thesis "eSIM With Secured Context Switching". Gratefully, I got exhortation from different individuals to whom, I need to unequivocal my affirmation towards in this segment.

My sincerest thanks to my supervisor and mentor Kausik Sarker, Assistant Professor & Associate Head, Department of Software Engineering, Daffodil International University for giving us an opportunity to do this. Without his support, direction and inspiration we would not have the capacity to understand the structure and got a yield.

I also thankful to Professor Dr. Touhid Bhuiyan, Professor & Head, Department of Software Engineering, Daffodil International University.

I additionally thank scientists for their works which assist us with learning plan and actualize my exploration venture.

Overall, I need to express gratitude toward Almighty Allah for giving us tolerance. I am additionally appreciative to relatives and companions for giving their compelling help and supplications.

TABLES OF CONTENTS

APPROVAL.....	i
DECLARATION.....	ii
ACKNOWLEDGEMENT.....	iii
TABLE OF CONTANT.....	iv
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
ABBREVIATIONS.....	viii
ABSTRACT.....	ix
CHAPTER 1: INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Statement.....	1
1.3 Research Questions	2
1.4 Research Objectives	2
1.5 Scope.....	2
1.6 Thesis Organizations.....	3
CHAPTER 2: LITERATURE REVIEW.....	4
2.1 Background.....	4
2.2 Technical Specifications and Related Works.....	5
2.2.1 GSMA eUICC Architecture.....	5
2.2.2 Limitations of GSMA eUICC architecture.....	13
2.2.3 Virtual SIM (VSIM) Architecture.....	14
2.2.4 Limitations of Virtual SIM (VSIM) Architecture.....	14
2.3 Research Gap Analyzing.....	15

CHAPTER 3: RESEARCH METHODOLOGY.....	16
3.1 System Architecture.....	17
3.1.1 Context Switching Model.....	17
3.1.2 Profile activation/ deletion by network request.....	18
3.1.3 Request routing through Subscriber Profile Manager.....	19
3.1.4 Communication channel Stablishing Sequence Diagram...	20
3.2 Status Flag Details.....	21
3.2.1 Profile.....	21
3.2.2 Network.....	22
3.3 Profile Database.....	22
3.4 Description of Context Switching Model.....	22
CHAPTER 4: RESULTS AND DISCUSSION.....	25
4.1 Simulation proof of proposed Model.....	26
CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS.....	32
5.1 Findings and Contributions.....	32
5.2 Future Works.....	32
5.3 Recommendations.....	32
REFERENCES.....	33
APPENDIX – A.....	35
APPENDIX – B.....	45
APPENDIX – C.....	48
APPENDIX – D.....	49

LIST OF TABLES

Table 2.1: Conditions of eUICC provisioning and managements life cycle.....	6
Table 3.1: SPSC or SPM readable Flags with action.....	21
Table 3.2: Network readable Flags with action.....	22
Table 3.3: Actions for profile flag of active profile (SPM).....	24
Table 3.4: Actions of profile flag of requested profile (SPM).....	24
Table 4.1: System comparison with proposed model.....	25

LIST OF FIGURES

Figure 3.1: Context Switching Model Diagram.....	17
Figure 3.2: profile activation/ deletion by network request (in device).....	18
Figure 3.3: Request routing through Subscriber Profile Manager (Server).....	19
Figure 3.4: Sequence Diagram of Stablishing a channel for communication	20
Figure 4.1: Simulator Color Specifications.....	26
Figure 4.2: Simulator Initial Sate.....	26
Figure 4.3: Dialer one request through same connection.....	27
Figure 4.4: Dialer two made a request for another network.....	28
Figure 4.5: Server sends unable feedback.....	28
Figure 4.6: Request of dialer two are rejected.....	29
Figure 4.7: Dialer one has freed the channel again.....	29
Figure 4.8: Dialer two request again and server send a request to SPSC.....	30
Figure 4.9: SPSC has activated Requested profile to the GSM unit.....	31
Figure 4.10: After Profile switching, channel has established.....	31

ABBREVIATIONS

SIM	Subscriber Identity Module
eSIM	Embedded Subscriber Identity Module
VSIM	Virtual Subscriber Identity Module
BS	Base Station
MNO	Mobile Network Operator
M(V)NO	Mobile (Virtual) Network Operator
eUICC	Embedded Universal Integrated Circuit Card
EIS	eUICC Information Set
GSMA	GSM Association
ECASD	eUICC Certificate Authority Security Domain
SM-DP	Subscription Manager Data Preparation
SM-SR	Subscription Manager Secure Routing
ISD-R	Issuer Security Domain Root
SRID	ID of the relevant SM-SR
EID	eUICC-ID
ICCID	Integrated Circuit Card ID
EUM	eUICC Manufacturer
VSIM	Virtual SIM
SPM	Subscriber Profile Manager
SPSC	Subscriber Profile Switching Center

ABSTRACT

Users into telecommunication technology want to make safety, customization, personalization, quality channels including autonomy. This requirement increases the quantity of device and expenses. Context switching model reduces the quantity of device and reduces costs and establishes parallel connections with various client identity modules to meet significant demand. It assures device and SIM guard and stands against SIM cloning.

Keywords: eSIM, VSIM;

CHAPTER 1

INTRODUCTION

1.1 Background

Every day we are moving from heavy devices to small devices and our needs are increasing, mobile companies are trying to connect our needs in a small space [8]. The conventional removable SIM device contains a large amount of space and creates a limitation on our usage, it can be removed and cloned without user's permission, this vulnerability permits the device to steal and produce immoral activities. To overcome user removal vulnerabilities, discovered a new technology called eSIM, it is a rewriteable built-in secure hardware component, which stores the customer's identity profile and is installed from a central server [5]. On the other hand, VSIM concept is a revolution of user autonomy.

1.2 Problem Statement

After thieves steal a SIM-based device, eliminate the physical SIM, which can be used without the permission of the owner of the phone and the device is unable to find the device through the device, each device has an IMEI, the identification number of the device and this number is stored on the device unsafe. This number can be changed because IMEI Tracking Technology is not accessible to the general public and many of them do not save the IMEI number and they do not have any benefits due to the change of IMEI number or IMEI number masking, they will also lose the interest to use this technology. Although eSIM technology can to solve the SIM removal problem but not able to solve the problems caused by unsafe IMEI.

There is no good quality network in all places, competitive operator price is also different and people do not want to mix their personal and business life, so they need more than one SIM and to use multiple SIM they need multiple devices also. multiple radio and internet component are needed to enable multiple SIMs. As a result, the cost of the device also increases. Today, the customers do not want to rely on the third party, they themselves want to control everything themselves.

1.3 Research Questions

1. How eSIM and VSIM does work?

1.4 Research Objectives

The ultimate purpose of this research is to introduce the eSIM Context Switching model to expand ease of use and security level. So as to accomplish this ultimate purpose, some sub-objectives can be planned:

1. To develop a proper context switching model.

1.5 Scope

In this paper, main intention is introducing eSIM Context Switching model, which may able to increase the usability, reduce using one more device, cut of cost, improving physical and virtual security issues, and ensuring user autonomy.

1.6 Thesis Organization

This paper comprises of five sections which will cover the presenting and portrayal of eSIM Context Switching model. Here is an outline of the substance of each exhibited section:

- ◇ **Chapter 1:** This section presents the issue, gives an overview of the investigation and portrays the requirements of eSIM Context Switching model. This part likewise talks about the extent of the investigation, the exploration question and its destinations.
- ◇ **Chapter 2:** This section covers the literature review which is the past related works that been done previously. Additionally, this section speaks to (1) pertinent data (2) constraints of eSIM and VSIM (3) examine hole.
- ◇ **Chapter 3:** This section clarifies the subtleties of the proposed framework (eSIM Context Switching Model) design and flowgraph with methods.
- ◇ **Chapter 4:** This section talks about the model advancement and the execution including examines the assessment procedure of the proposed model.
- ◇ **Chapter 5:** This section examines the end, proposals and future attempts to enhance this investigation.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

eSIM is a rewritable built-in hardware component [5,6] which allows to provisioned SIM profile remotely over the air by a mobile network operator through universal discovery server. Other network operators SIM profile further can be installed by users. Multiple SIM profiles can be installed but it can active one SIM profile at a time. It has three categories namely (1) machine to machine (M2M), (2) machine to person (M2P), and (3) hybrid. This technology designed to reduces space, cost and energy consumption of the device. And makes possible to independently uses of a radio network and does not require the exchange of physical components. In this technology four groups of players involved, they are (1) eSIM vendor, (2) original equipment manufacturer (OEM), (3) Mobile virtual network operator, and (4) Independent profile manager [5].

To serve this technology to the consumer it has to maintain three network configurations namely (1) OEM-centered, (2) MNO-centered, (3) Independent party. Mobile network operator sells M2P categories eSIM [5].

GSMA has well defined eSIM technology by specifying eight technical connection of OTA profile management with five proprietary interfaces namely (1) Subscription Manager – Data Preparation (SM – DP), (2) Subscription Manager – Secure Routing (SM – SR), (3) Mobile Network Operator (MNO), (4) Certificate issuer, (5) eUICC vendor [16]. Three steps need to complete procedure of profile download and install, this procedure flowed by the profile enablement [16].

GSMA embedded SIM introduces to speed up the prosperity of M2M markets and operational competence for M2M ecosystem. It creates a new business model to prevent markets fragmentation by eliminating various and inconsistent technical solutions. This technology reduces integration, testing and handling costs for M2M SIM products including little influence on the existing system and network infrastructure. eSIM is not the virtual SIM; it uses the same hardware component and existing SIM form factors including MFF1, MFF2 for embedded and 2FF, 3FF for removable [6].

Weak network coverage can lead to interruption of communications, different network operators give special tariffs for call rates or internet including providing different bundles that encourage us to use multiple subscriber profiles [14]. The customer does not want to disconnect the old number [14], and this tendency increases the quantities of the device. Significant numbers of European citizens use one more mobile phone to differentiate between personal and office [14].

2.2 Technical Specifications and Related Works

2.2.1 GSMA eUICC Architecture

Embedded eUICC standardized by European Telecommunications Standards Institute (ETSI), they published eUICC requirements, definitions, roles, procedures in the year 2013 [16]. GSMA develops a standard specification based on ETSI to facilitate the standardized ETSI activities and needs of industry stakeholders. GSMA covers OTA installations, enablement, disablement and deletion processes of the profile to the eUICC.

Table 2.1: Conditions of eUICC provisioning and managements life cycle [16,3]

No	Name	Condition	
		Start	End
01	eUICC Registration at SM-SR	eUICCs are produced and a Provisioning Profile is loaded and active in the Provisioning operator's network. They are tested and ready for shipment. Each eUICC has a corresponding EIS.	The eUICC is registered at the SM-SR and ready for Profile download. It can now be shipped to the machine to machine Device manufacturer
02	Un-personalized Profile Verification	<ul style="list-style-type: none"> i. The Profile Description has been provided by the MNO to the SM-DP and the Un-personalized Profile has been generated by the SM-DP in a separate process. ii. The SM-DP has sample eUICCs of a specific type. 	The Un-personalized Profile is valid and is now ready for the Profile ordering procedure for an eUICC type.

03	Profile Ordering	<ul style="list-style-type: none"> i. An Un-personalized Profile has been created by the SM-DP based on the Profile Description provided by the MNO. ii. The MNO has a demand for a quantity of eUICC Profiles. iii. The Un-personalized Profile has been validated on the target eUICC type using the Un-personalized Profile verification procedure in section 	<p>The ordered quantity of Profiles is now ready for the Profile download procedure. Related Operator Credentials are available to the MNO.</p>
04	Profile Download and Installation	<ul style="list-style-type: none"> i. A Customer has subscribed to a selected MNO. ii. The EID of the target eUICC and the SRID are known by the MNO. iii. A Profile ordering procedure has been 	<p>An ISD-P has been created in the eUICC for the MNO, containing a Profile in disabled or enabled state. The SM-SR has updated the EIS for this eUICC accordingly. The MNO may activate the related</p>

		<p>completed with a selected SM-DP.</p> <p>iv. The target eUICC is integrated into a machine to machine Device and is associated to an SM-SR.</p> <p>v. The MNO may activate the related Subscription in the network by the ICCID.</p>	Subscription in the network.
05	Master Delete	<p>i. There is an Orphaned Profile on a eUICC which is, for example, blocking the loading of another Profile.</p> <p>ii. The Orphaned Profile cannot be deleted using the normal ISD-P deletion procedure.</p>	The target Profile is deleted from the eUICC. The EIS in the SM-SR is up to date.

		<ul style="list-style-type: none"> iii. The Initiator decides to delete the Orphaned Profile on the eUICC. iv. The Orphaned Profile is disabled. 	
06	Profile Enabling	<ul style="list-style-type: none"> i. The target Profile is disabled on the eUICC. Another Profile is enabled. ii. The Subscription associated with the target Profile is active in the MNO's network. iii. The EID of the target eUICC, the SRID associated with the target Profile and the ICCID of the target Profile are known by the MNO. 	The target Profile is enabled on the eUICC. The previously Enabled Profile is disabled. The EIS is up to date.
07	Profile Enabling via SM-DP	<ul style="list-style-type: none"> i. The target Profile is disabled on the eUICC. 	The target Profile is enabled on the eUICC. The previously Enabled Profile

		<p>Another Profile is enabled.</p> <p>ii. The Subscription associated with the target Profile is active in the MNO's network.</p> <p>iii. The EID of the target eUICC, the SRID and the ICCID of the target Profile are known by the MNO.</p>	is disabled. The EIS is up to date.
08	Profile Disabling	The target Profile is enabled on the eUICC.	The target Profile is now disabled on the eUICC, and the Profile with Fall- back Attribute set is enabled.
09	ISD-P Deletion	The MNO decides to permanently delete a Profile on a eUICC.	The target Profile is deleted from the eUICC. The EIS in the SM-SR is up to date.

10	ISD-P Deletion via SM-DP	The MNO decides to permanently delete a Profile on a eUICC.	The target Profile is deleted from the eUICC. The EIS in the SM-SR is up to date.
11	SM-SR Change	<ul style="list-style-type: none"> i. The EID of the eUICC is known ii. The SRIDs of SM-SR1 and SM-SR2 are known. iii. The ISD-R is personalized with the keys of SM-SR1. iv. The change of SM-SR is allowed. 	<ul style="list-style-type: none"> i. The ISD-R is personalized with the keys of the target SM-SR (SM-SR2). ii. The eUICC is registered within the target SM-SR (SM-SR2). iii. The EIS and EID reside within the target SM-SR (SM-SR2). iv. SM-SR1 is no longer related to the eUICC. v. The MNO owner of the Profile(s) is aware of the change.

12	Fall-back Mechanism	<ul style="list-style-type: none"> i. The machine to machine Device reports network loss to the eUICC. ii. The eUICC is configured to perform the Fall-back mechanism if certain network connectivity issues are reported by the machine to machine Device. iii. The Profile with Fall-back attribute set is not the presently Enabled Profile. 	The eUICC has enabled the Profile with Fall-back Attribute set and the EIS of the SM-SR is up-to-date.
13	eUICC Certificate Check	eUICC has to designed according to the present specification	The eUICC certificate and EUM certificate has been checked by the MNO.

2.2.2 Limitations of GSMA eUICC architecture

To find limitations of Embedded Universal Integrated Circuit Card (eUICC) we considered all procedure sections of eUICC they are (1) eUICC Registration at SM-SR, (2) Un-personalized Profile Verification, (3) Profile Ordering, (4) Profile Download and Installation, (5) Profile Enabling, (6) Profile Enabling via SM-DP, (7) Profile Disabling, (8) ISD-P Deletion, (9) ISD-P Deletion via SM-DP, (10) Master Delete, (11) SM-SR Change, (12) Fall-back Mechanism, (13) eUICC Certificate Check with initial and end state condition of circuit [table 1]. We found many issues like security and lack of requirements [14]. Many of the security issues already solved by reference [16], authors of this paper considered only eUICC procedure security issues and they did not consider any lacking requirements and performance. So here we are trying to bold these sections lacking's and we found out -

1. **Multiple profile activation or artificial switching is not possible:** The industry has the intention to make a slim device and people do not like the limitation on usage. eSIM reduced space but creates a limitation on usage, although it can install multiple sim profiles there is no way to activate all profile or active/disable automatically any profile by network request. Automatically enable and disable should be happening when activate profile does not handle any necessary request and coming request want to interact with the user.
2. **Inactive profile cannot share contact list to the device:** There is no way to partial inactivation for the profile, disabling profile [table 1] perform full disable. The inactive profile cannot share the contact list with the device. This problem creates data redundancy because customer needs existing numbers but eSIM stores contact list separately into the profile memory which does not share any partially contacts with the device when it does disable profile.

3. **Response time is long for unable connection:** If any profile disabled and has a request from network against this profile then need to flow all procedures end user to end user. request response time may be less, if the system saves the state of the profile to the network.
4. **User cannot manage own profile remotely:** eSIM doesn't gives user to profile management autonomy. If anyone need to change profile or anything with profile then they need to contact with MNO that makes user unsatisfactory.

2.2.3 Virtual SIM (VSIM) Architecture

Virtual Subscriber Identity Module (VSIM) which is introduced in the year 2012. This technology provides different framework and technique epitomes for holding and exchanging individual information contained inside the memory of portable handset gadgets. A few exemplifications furnish versatile handset clients with the capacity to download individual information to a server after validation and Verification steps are finished. Different encapsulations enable versatile handset clients to arrangement new portable handsets remotely by transferring individual information from a server. A few exemplifications use alphanumeric passwords for client confirmation and check purposes. Different exemplifications actualize biometric sensors for client validation and Verification reason [13].

2.2.4 Limitations of Virtual SIM (VSIM) Architecture

VSIM is the first concept that considered user autonomy they give all legal management powers to the users but it has some lacking's on security, performance and requirements also.

1. **SIM cloning:** It can copy profile from old SIM and VSIM doesn't specified anything about old SIM after doing copy. If they do not delete profile from old SIM or disable SIM serial then it may lead SIM cloning [13].

2. **Multiple profile activation:** In this technology has multiple profile installation system but at a time one profile can be active. And they do not specify anything about offline profiles or any artificial technology that can solve this problem [13].
3. **Response time:** VSIM server doesn't remain status of any profile [13]. so, VSIM server unable to give waiting, busy, closed feedback without request device. This fault can to increase response time.

2.3 Research Gap Analyzing

This paper directly related with eSIM and VSIM technology but both technologies has some common critical issues for both technologies like only single profile activation, contact or others required data cannot be shared to the device without profile activation, Response time high for different cases. eSIM has a unique issue about user autonomy. On the other hand, VSIM also has a major unique issue about old profile back up from device which may lead SIM cloning problem.

CHAPTER 3

RESEARCH METHODOLOGY

eSIM with required context switching model is the combinations of eSIM [5,6,3] and VSIM [13] model with few modifications where reduced cost, space and increases usability by overcoming all the limitations of eSIM and VSIM model that already indicated to 2.2.2 and 2.2.4. it has many features, some of them-

1. **Handling multiple SIM:** It is the main purpose of this model. Multiple profile can be active according to user observations by integrating M2M [5] artificial methodology.
2. **Sharing:** If any SIM is closed or offline then it can share necessary data (such as contact) can be shared with device.
3. **Response time will be half:** If channel request is come and profile is not active at eUICC or busy or waiting then it can to send feedback without requesting to end device. So, it can reduce time then traditional Network system.
4. **Ensuring user autonomy:** People always want self-maintenance ability and this model gives remotely SIM management such as add new SIM or delete SIM or modify SIM status without contacting MNO.
5. **Preventing sim cloning:** To reduce SIM cloning, in this system has not given any IMEI modification autonomy to the user and server continuously update its IMEI of profile database.
6. **All pros of eSIM and VISM are present:** All good requirements (authentication, profile installation/deletion, network access, channel establishment. Etc.) of eSIM and VSIM are presents that give more reliable model.

3.1 System Architecture

3.1.1 Context Switching Model

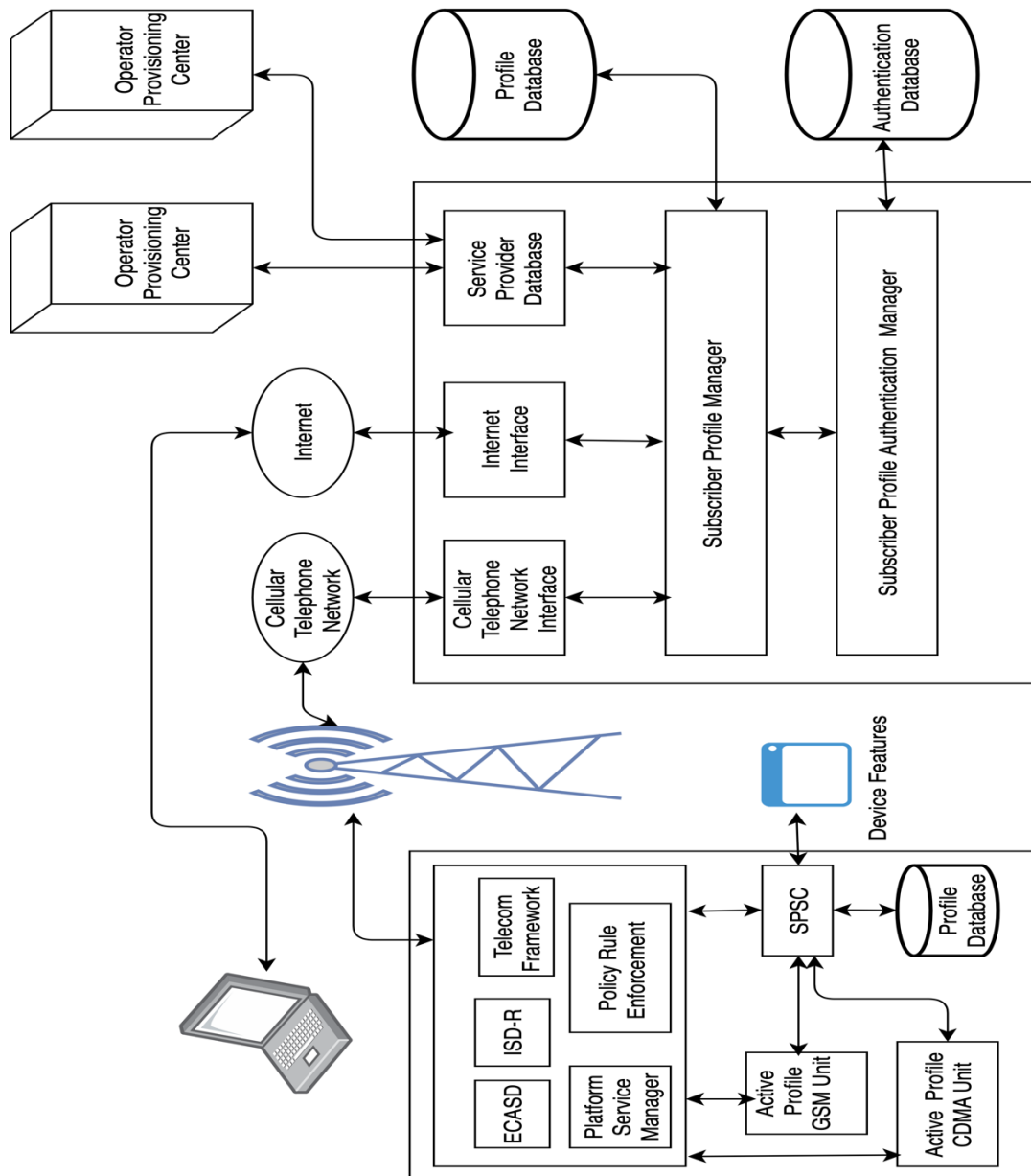


Figure 3.1: Context Switching Model Diagram [13,3]

3.1.2 Profile activation/ deletion by network request

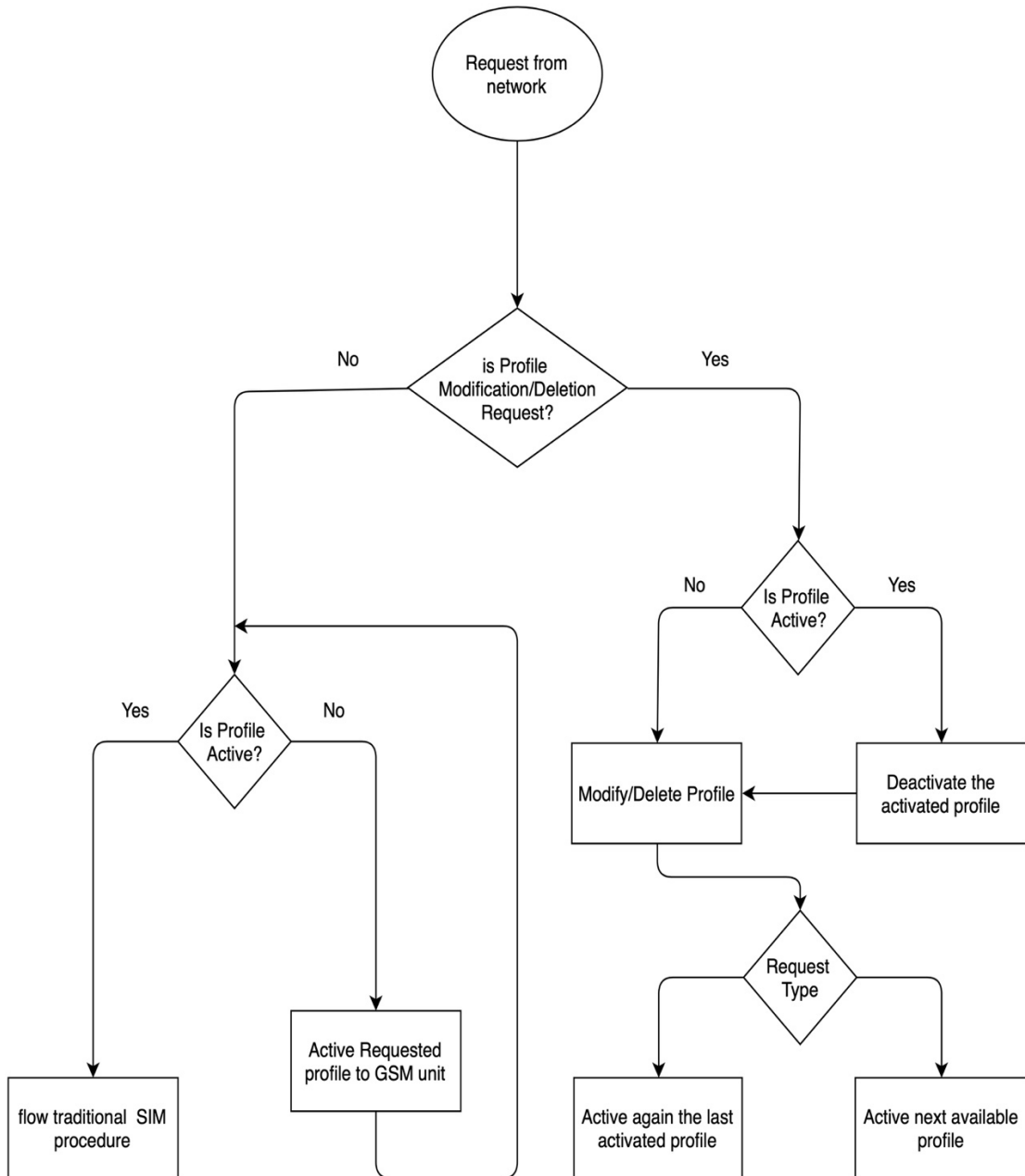


Figure 3.2: profile activation/ deletion by network request (in device)

3.1.3 Request routing through Subscriber Profile Manager

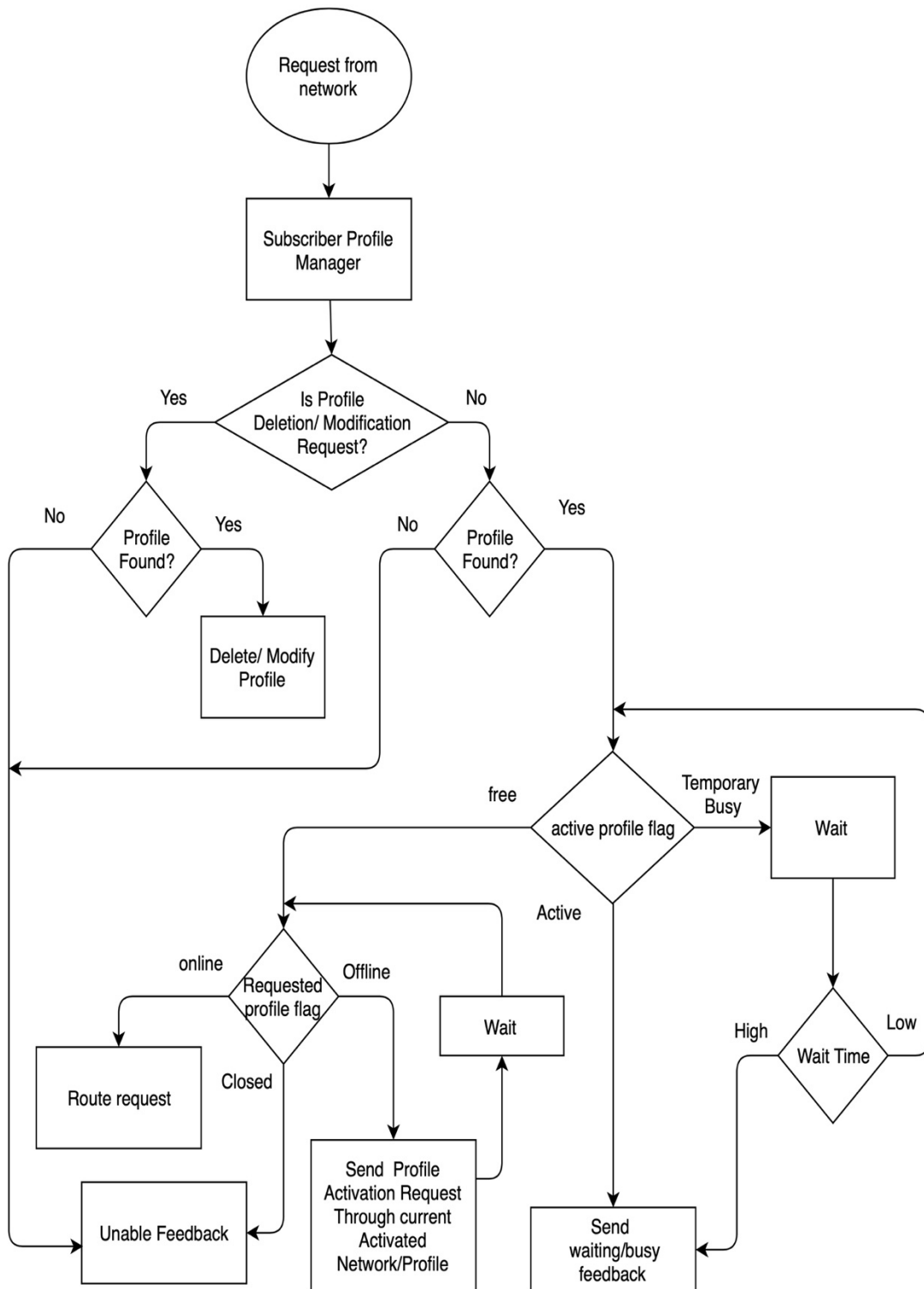


Figure 3.3: Request routing through Subscriber Profile Manager (Server)

3.1.4 Communication channel Stablising Sequence Diagram

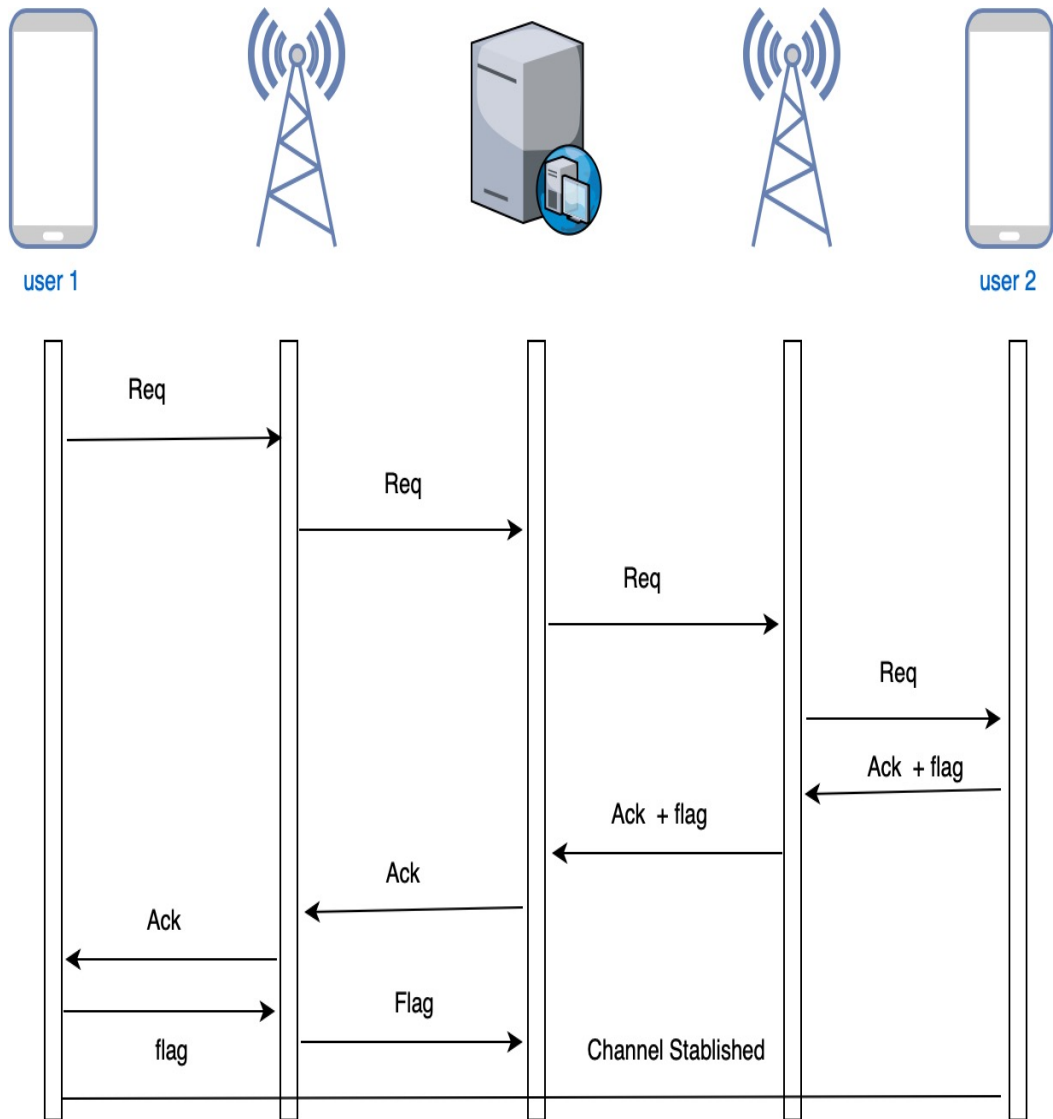


Figure 3.4: Sequence Diagram of Stablising a channel for communication

3.2 Status Flag Details

Flag is a predefined bit to remember something or to leave a sign for others program or to communicate M2M [4]. To communicate between SPSC and SPM we need different flags.

3.2.1 Profile

These types of flags can be set within internal data for subscriber profiles. SPSC and SPM has only right to show these flags.

Table 3.1: SPSC or SPM readable Flags with action

No	Flag Name	Action
1	Active	When any communication channel has to established between two end device then active profile can be set.
2	Online	It does indicate that Profile is active and it has established network.
3	Offline	This flag denoted that currently profile not active or has not established network but it can be active by SPM request.
4	Closed	it does indicate unable to activate this profile
5	Temporary Busy	If activate profile receive any data or signal for a while then this flag can be set to the profile.
6	Busy	This flag denoted that it can handle only message request.

3.2.2 Network

These types of flag can be set on the request header to send data between server and device. MNO has no right to open its internal data. This type of flags ensures high priority.

Table 3.2: Network readable Flags with action

No	Flag Name	Action	Directions
1	Server	This flag can be set by SPSC or SPM on header of the data packet.	i. Server to Server ii. Device to Server

3.3 Profile Database

Profile database [13] contains data table with at least four attributes they are (1) serial no, (2) profile raw data, (3) flag, (4) IMEI. This table's name will depend on the user primary key. Profile flag and IMEI will be updated continuously through device with help of any network and IMEI manually update should be stopped. Only SPSC and SPM has database writing ability.

3.4 Description of Context Switching Model

1. ECASD, ISD-R, Telecom Framework, Platform Service Manager, Policy Rule Enforcement will work like eSIM procedure [11] but it has to use SPSC as media to communicate with Subscriber Identity Profile, as shown in Figure 3.1.
2. SPSC will active profile into CDMA activation unit only by own device request not by the network request. Procedure 3 to 14 only for GSM network.
3. When procedure 1 handover works to the SPSC then it has to check request type. Two type of request can be sent by GSM network one of them Profile related which is devoted to profiling modification, As shown in Figure 3.2. Another type of request is user related which procedures like traditional SIM system [9].

4. If SPSC gets any request without profile related request to the SPM then, it will flow procedure of eSIM, as shown in Figure 3.2.
5. When SPSC will send device/flag information to the SPM about every Subscriber Profiles then SPSC must have to send encryption key. Although, SPSC will send all of these data through the ONLINE flagged profile. SPSC will add a SERVER flag to transfer these types of data.
6. If any data header contains SERVER/ DEVICE flag then network must have to proceed and MNO has no right to view these types of data.
7. Cellular Telephone Network Interface, Internet Interface, Service Provider Database and Subscriber Profile Authentication Manager will work like VSIM procedure [13] but it has to use to SPM as media to handle SERVER flagged request or any database related request. And
8. Without SERVER/ DEVICE flag network will flow traditional network procedure [9].
9. Profile installation and deletion from eUICC procedure like eSIM [11] but it has to handover to the SPSC before final stage of procedure. And profile deletion and installation to the server site procedure like VSIM [13] but it has to handover to the SPM before final stage.
10. When SPM get any request then it has to check request type. There is a two type of network requests for SPM one of them is Profile related and another is routing related. Both request first check profile presents from profile database if profile not found then it produced unable feedback, as shown in Figure 3.3. And if profile present then it flows procedure 10 or 11.
11. If network request for profile deletion/ modification it send again deletion request to SPSC to make ensure and then delete or modify profile to the server side.

12. If network request for routing then it has to consider profile status, there are three types of profile status available that must have to consider for active profile to take further actions. Profiles flag with action-

Table 3.3: Actions for profile flag of active profile (SPM)

Profile Flag Name	Condition Check	Action
Temporary Busy	Wait Time High	wait few moments and check again its profile
	Wait Time Low	Send busy feedback
Active		Send waiting/ Busy feedback
Free		Flow procedure 12

In this table “Free” means active profile is TEMPORARY BUSY or ACTIVE flag free.

13. In this stage need to route the request, it has to check three types of flags for requested profile to take further actions. Profile flag with actions-

Table 3.4: Actions of profile flag of requested profile (SPM)

Profile Flag Name	Action
Online	Route the current request
Offline	First send profile activation request to the SPSC and wait and check again.
Closed	Send unable feedback

14. If need to established any channel between two devices then network need to communicate with server, as shown in Figure 3.4.

CHAPTER 4

RESULTS AND DISCUSSION

In this paper, we are trying to solve some eSIM issues and providing proper context switching model by merging two system [3,5] with quite modifications, one of them eSIM which has some pros on security and policy but it has also some limitations such as activating profile parallely, user profile management autonomy, etc.

Table 4.1: System comparison with proposed model

No	Features	eSIM	VSIM	Proposed Model
01	Multi profile active	No [5,16,3,6]	No [13]	Yes
02	Offline profile can share contacts	No [3]	No [13]	Yes
03	Profile backup from device	Yes [3]	Yes [13]	No
04	Unable or waiting feedback	From Device [3]	From Device [13]	From SPM
05	Unable or waiting feedback processing time	Long [3]	Long [13]	Short
06	User Autonomy	No [3]	Yes [13]	Yes
07	Device owner authentication Required To add profile	Not Specified	Not Specified	Yes
08	Profile owner authentication required to add profile	Yes [3]	Yes [13]	Yes
09	Master password for modification	Not Specified	Yes [13]	Yes
10	CDMA and GSM Active at a time	No [3]	No [13]	Yes

Another one is VSIM where has user profile management autonomy but it can copy old SIM which may lead SIM cloning, and it has no ability to active profile parallelly like eSIM [13]. In table 4.1 has shown the comparison between eSIM, VSIM and Proposed Model to understand its necessity easily.

4.1 Simulation proof of proposed Model

To prove proposed model, we made a virtual simulator where we considered the only context switching part. In the device, we used eSIM Constant as a transmitter which is a box of all eSIM and device necessary components. In the VSIM server, we used only internet interface because of other interfaces are not related to context switching. To test this model perfectly we used three cellular networks tower and two Dialer devices. Dialer devices have context switching feature also. Here we added some screenshots of our simulator experiments with action description-

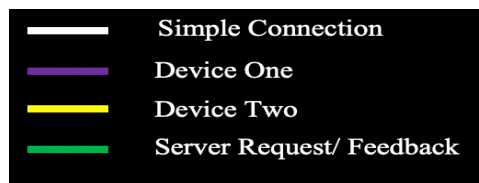


Figure 4.1: Simulator Color Specifications

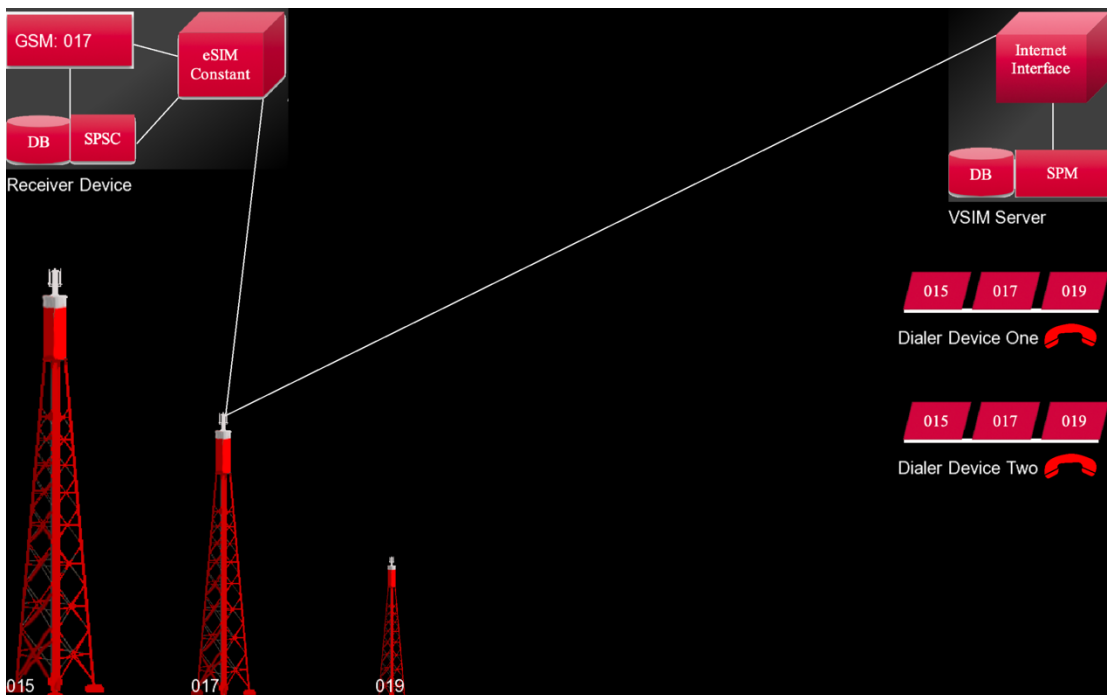


Figure 4.2: Simulator Initial State

In figure 4.2, GSM activation unit, SPSC and eSIM constant has made a triangle normal connection within receiver device one the other hand, Internet Interface and SPM also made a normal connect between them within VSIM server site. At this time eSIM constant of Receiver Device has made a normal connection with Internet Interface of VSIM server through GSM network tower 017. Normal connection denotes online flag for profile.

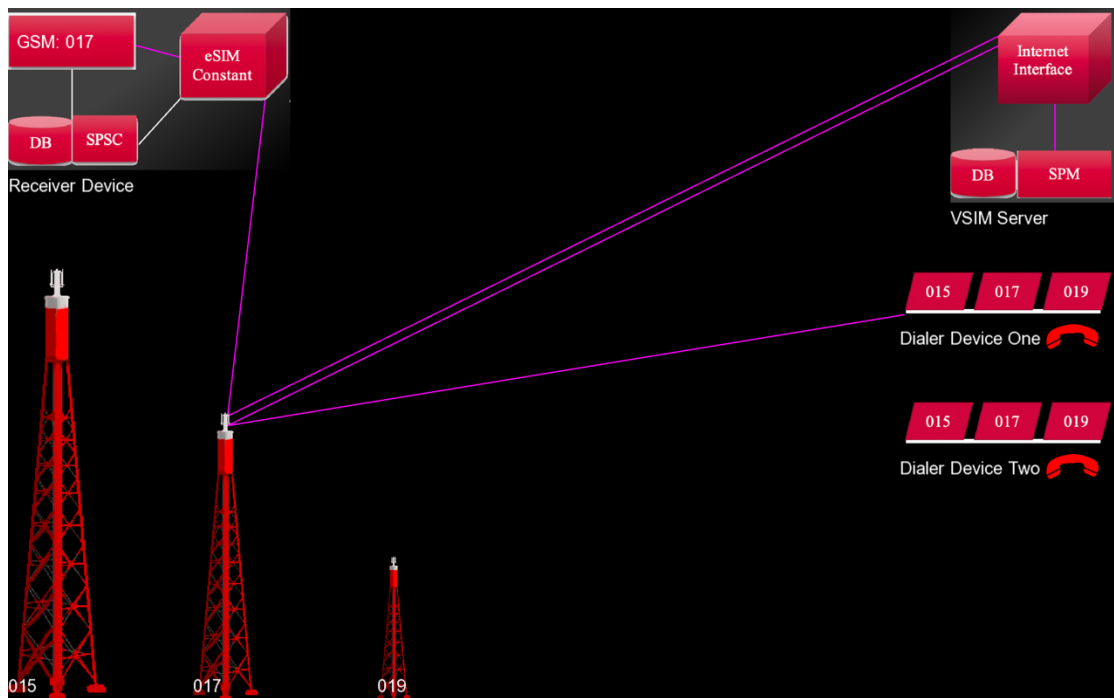


Figure 4.3: Dialer one request through same connection

In figure 4.3, Dialer Device One wanted to established a voice channel with Receiver Device profile 017. To established this channel, it does some sequential action,

1. Dialer Device One send a request to the GSM network 017
2. GSMA network 017 route this request to the Internet Interface
3. Internet Interface handover the request to the SPM
4. SPM analyze profile flags of requested profile and it got online flag [Figure 4.2], after getting this flag SPM route this request to eSIM constant through Internet Interface and GSM network tower 017.
5. eSIM constant communicate directly with GSM activation unit to establish a voice channel.

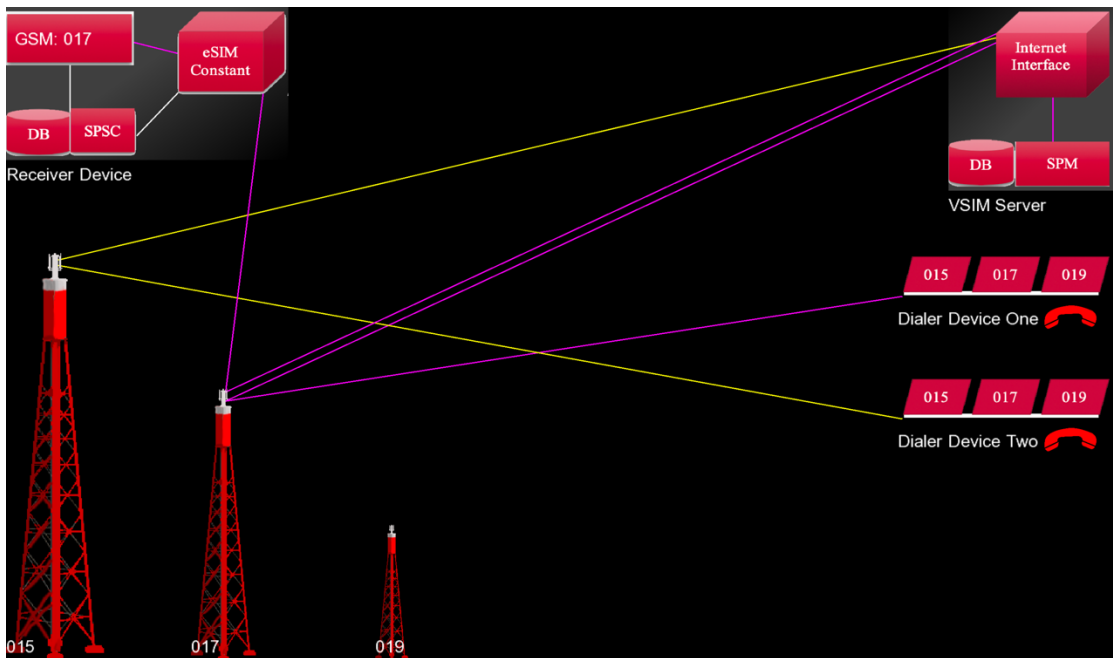


Figure 4.4: Dialer two made a request for another network.

In figure 4.4, Dialer Device Two wanted to established a voice channel with Receiver Device profile 015.

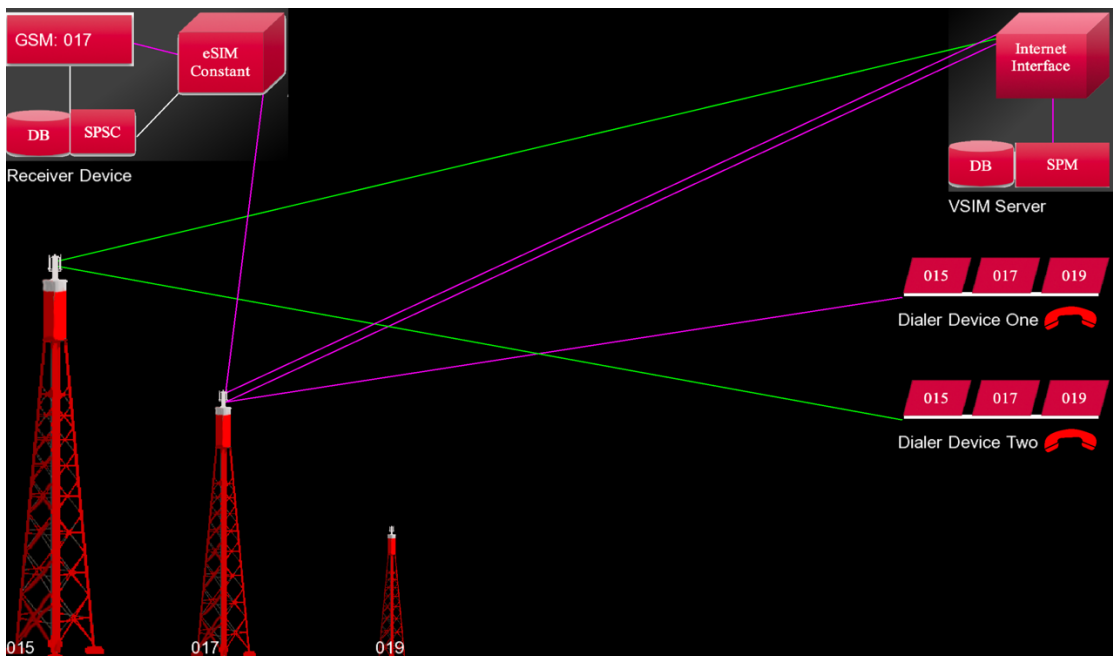


Figure 4.5: Server send activation feedback

In Figure 4.5, But device one has already an established voice channel (Active flag) so SPM returns a busy feedback to the dialer device.

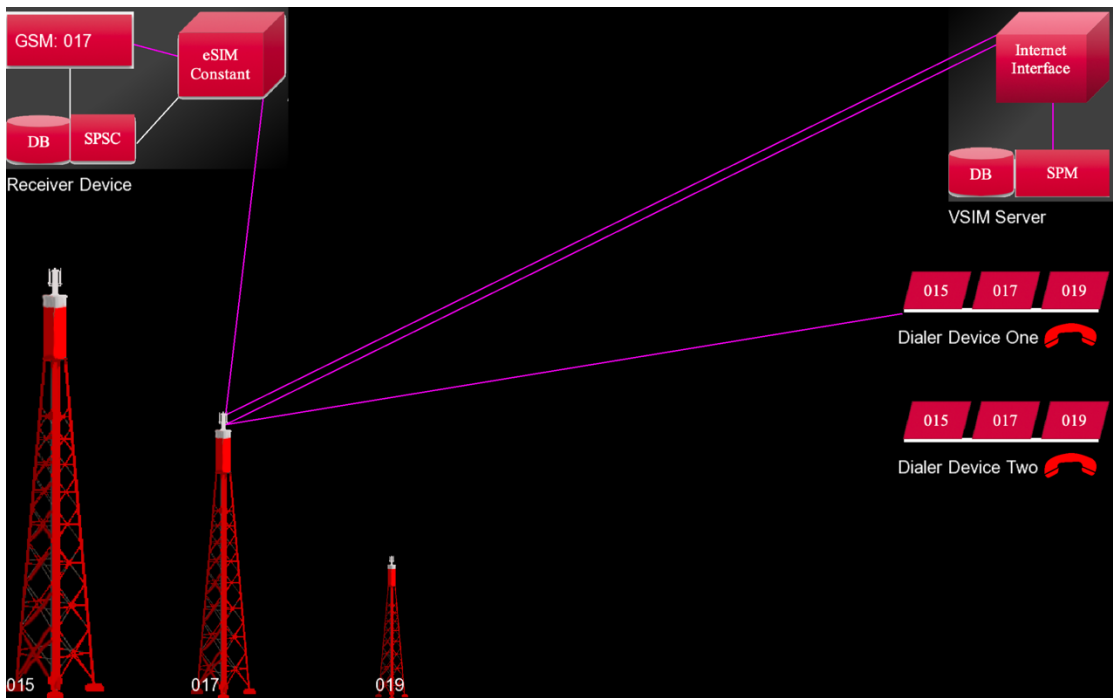


Figure 4.6: Request of dialer two are rejected

In figure 4.6, after getting busy feedback from SPM Dialer Device Two reject request automatically.

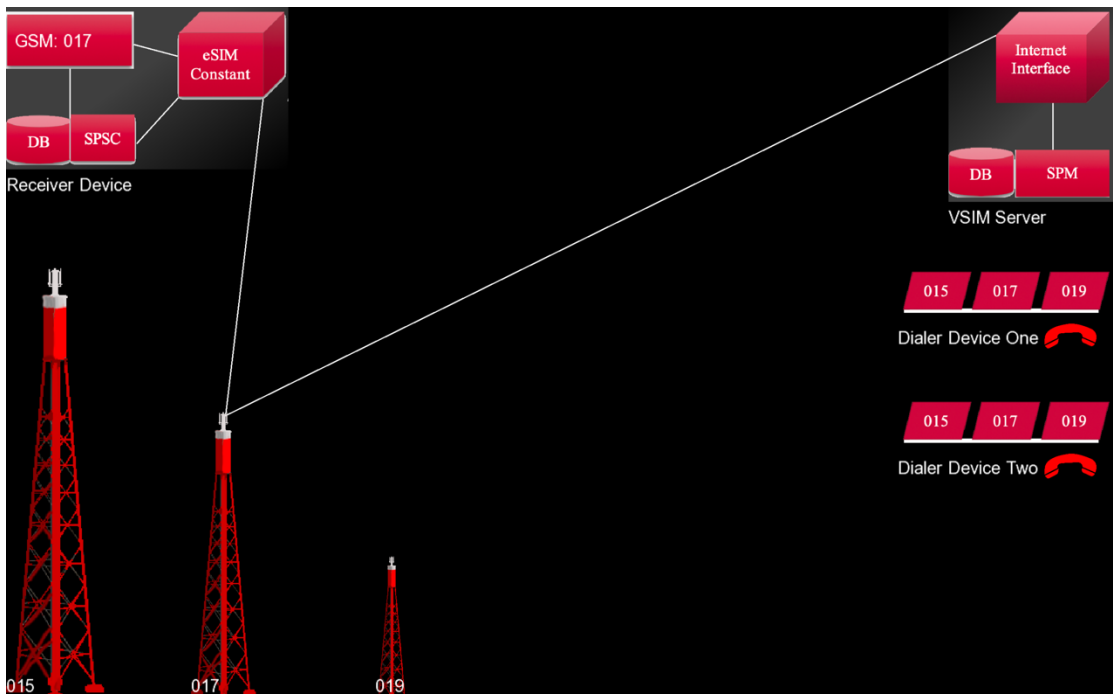


Figure 4.7: Dialer one has freed the channel again

In figure 4.7, Dialer Device One has removed the established voice channel.

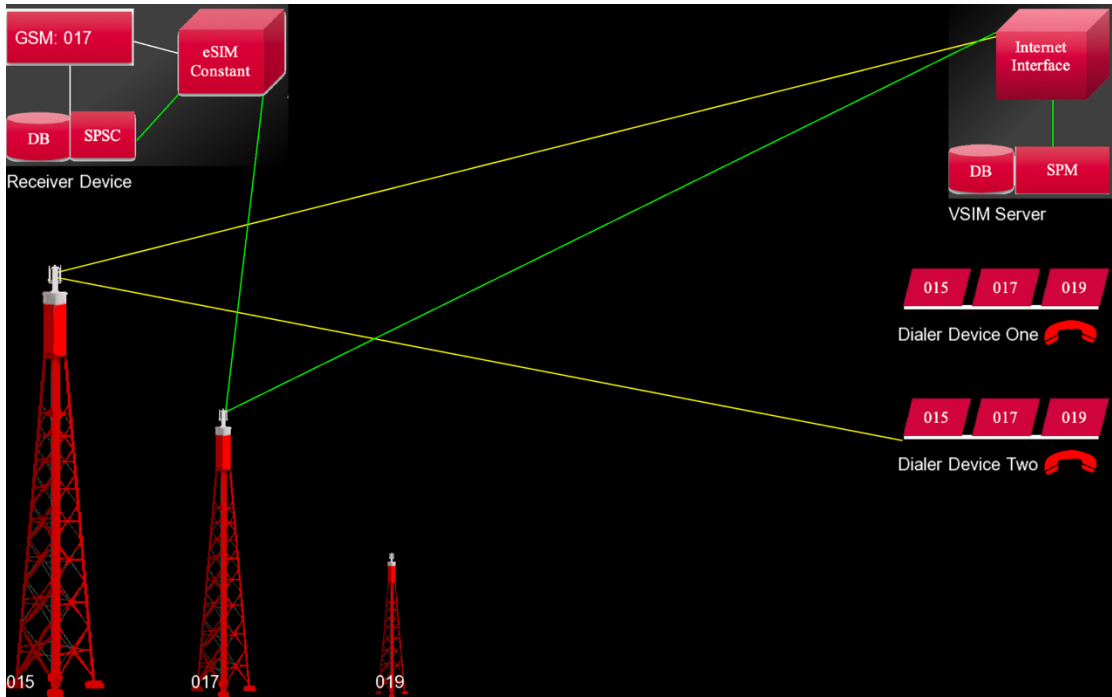


Figure 4.8: Dialer two request again and server send a request to SPSC

In figure 4.8, Dialer Device Two wants to establish a voice channel with Receiver Device Profile 015 but Receiver Device has active different profile with online status. To establish this channel, it does some sequential task.

1. Dialer Device Two VSIM server Internet Interface through GSM network tower 015.
2. Internet Interface handover this request to the SPM as usual.
3. SPM analyzed the requested profile and got offline status of requested profile. it also got different online flagged profile which is located same Receiver Device that wanted by Dialer Device Two. At this situation SPM send profile switching request to the Receiver Device eSIM constant through current online profile connection. SPM does wait for a while and recheck again.
4. eSIM constant checked again is profile active or not and reroute to the SPSC.

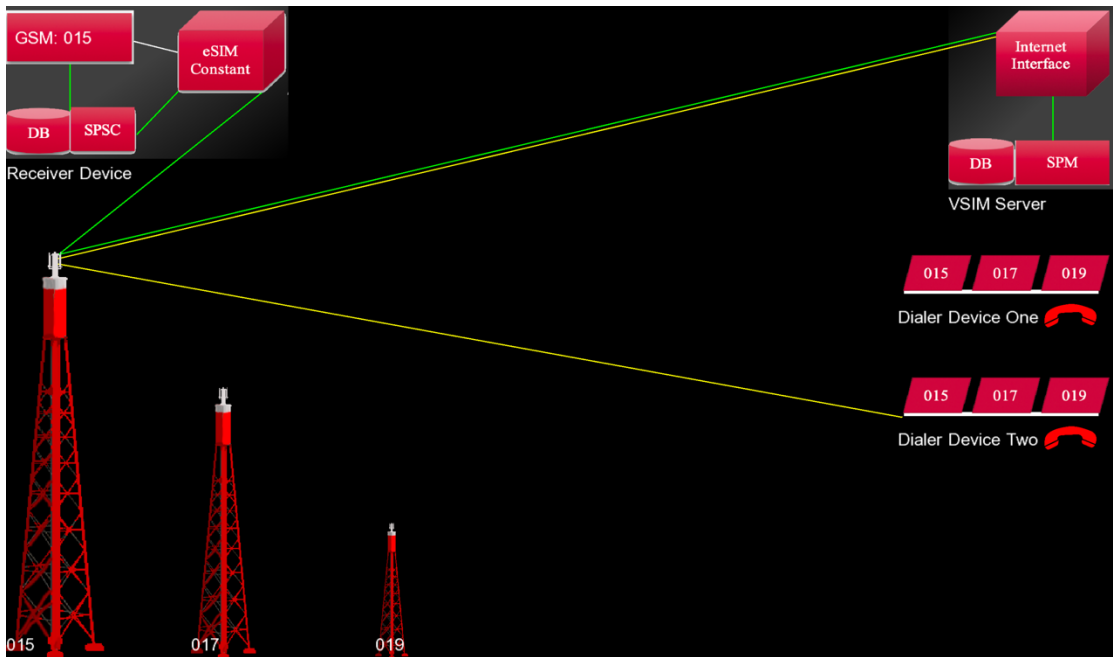


Figure 4.9: SPSC has activated Requested profile to the GSM unit

- In figure 4.9, SPSC removed current activated profile from GSM profile activation Unit and active requested profile.

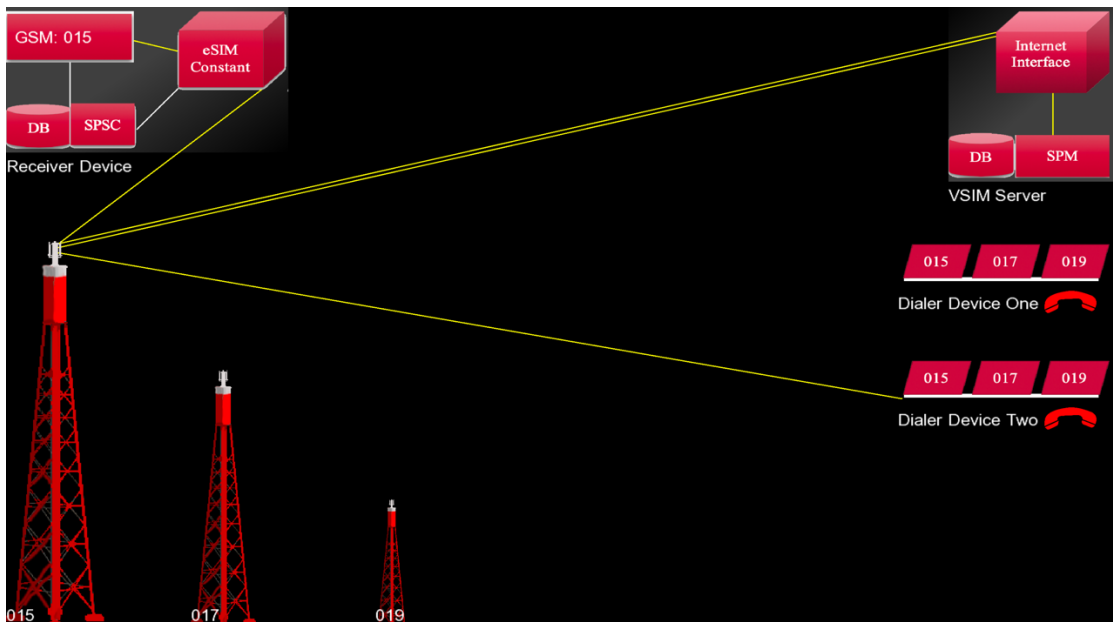


Figure 4.10: After Profile switching, channel has established.

- In Figure 9, a voice channel is established between Receiver Device and Dialer Device Two.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Findings and Contributions

Here the main focus was finding limitations of eSIM (Embedded Subscriber Identity Module) and VSIM (Virtual Subscriber Identity Module) to introduce proper context switching model. Finally, in this paper has been established “eSIM with Secured Context Switching” model which is a combination of eSIM and VSIM model with quite a modification, and this model has mainly two intention one of them using the same hardware components to reduce cost and space, another one is parallel activating all installed profiles within the eUICC according to the network request. in this model also has been considered all pros of these existing model.

5.2 Future Works

In this paper has solved context switching of subscriber’s profile in eUICC, user autonomy and some security issues such as SIM cloning, etc. In this paper main goal was defining context switching model so, we did not concentrate at SPM to another SPM communication.

5.3 Recommendations

To include SPM to another SPM communication with “eSIM with Secured Context Switching” model, have to consider at least three criteria they are (1) SPM unique identity (2) more than one SPM should not create for the state or the country (3) use of country code or state code as an identity number of SPM may be beneficial to route between two SPM. SPM routing may solve the roaming problem also.

REFERENCES

1. Bender, H., & Lehmann, G. (2012). Evolution of SIM provisioning towards a flexible MCIM provisioning in M2M vertical industries, in *Intelligence in Next Generation Networks*. In *2012 16th International Conference on Intelligence in Next Generation Networks* (pp. 57 - 64). Papers (5) | Patents (4): IEEE. Retrieved from <https://ieeexplore.ieee.org/document/6376034>
2. Casey, T., Smura, T., & Sorri, A. (2010). Value Network Configurations in wireless local area access. In *2010 9th Conference of Telecommunication, Media and Internet* (pp. 1-9). Papers (13): IEEE. Retrieved from <https://ieeexplore.ieee.org/abstract/document/5557719>
3. Embedded SIM Remote Provisioning Architecture. (2013). Retrieved from <https://www.gsma.com/iot/wp-content/uploads/2014/01/1.-GSMA-Embedded-SIM-Remote-Provisioning-Architecture-Version-1.1.pdf>
4. Flag Definition. (2018). Retrieved from <https://whatis.techtarget.com/definition/flag>
5. Gerpott, T., & May, S. (2017). Embedded Subscriber Identity Module eSIM. *Business & Information Systems Engineering*, 59(4), 293-296. doi: 10.1007/s12599-017-0474-4
6. GSMA Embedded SIM Specification Remote SIM Provisioning for M2M. (2018). Retrieved from <https://www.gsma.com/iot/wp-content/uploads/2014/10/Embedded-SIM-Toolkit-Oct-14-updated1.pdf>
7. LEADING M2M ALLIANCES BACK THE GSMA EMBEDDED SIM SPECIFICATION TO ACCELERATE THE INTERNET OF THINGS - Newsroom. (2015). Retrieved from <https://www.gsma.com/newsroom/press-release/leading-m2m-alliances-back-the-gsma-embedded-sim/>
8. Mobile network operator on-demand subscription management study. (2018). Retrieved from [http://www.ey.com/Publication/vwLUAssets/EY-mobile-network-operator-on-demand-subscription-management/\\$FILE/EY-mobile-network-operator-on-demand-subscription-management.pdf](http://www.ey.com/Publication/vwLUAssets/EY-mobile-network-operator-on-demand-subscription-management/$FILE/EY-mobile-network-operator-on-demand-subscription-management.pdf)
9. Mouly, M., & Pautet, T. (2018). The GSM System for Mobile Communications. Retrieved from <https://dl.acm.org/citation.cfm?id=573838>
10. Park, J., Baek, K., & Kang, C. (2013). Secure Profile Provisioning Architecture for Embedded UICC. In *2013 International Conference on Availability, Reliability and Security* (pp. 297 - 303). Papers (1) | Patents (11): IEEE. Retrieved from <https://ieeexplore.ieee.org/document/6657256>
11. Remote Provisioning Architecture for Embedded UICC Technical Specification. (2017). Retrieved from https://www.gsma.com/newsroom/wp-content/uploads//SGP.02_v3.2_updated.pdf
12. Richarme, M. (2008). The virtual SIM - a feasibility study. Retrieved from http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6812

13. Shi, G., Tangirala, V., Durand, J., & Dudani, A. (2018). Google Patents. Retrieved from <https://patents.google.com/patent/US8200736B2/en>
14. Sutherland, E. (2009). Counting Mobile Phones, Sim Cards & Customers. *SSRN Electronic Journal*. doi: 10.2139/ssrn.1468847
15. Vahidian, E. (2013). Evolution of the SIM to eSIM. Retrieved from <https://brage.bibsys.no/xmlui/handle/11250/262765>
16. Vesselkov, A., Hammainen, H., & Ikalainen, P. (2018). Value networks of embedded SIM-based remote subscription management - IEEE Conference Publication. Retrieved from <https://ieeexplore.ieee.org/abstract/document/7347220>

APPENDIX – A

```
/*Programming Language: Kotlin
```

```
IDE: IntelliJ Idea*/
```

```
class Home : ApplicationAdapter() {
```

```
/*
```

```
* This class is the core class for all device
```

```
* Here i implement design parts of eSIM context Switching
```

```
* As render() method do work as thread so, i did not add any extra thread
```

```
* i did only access static variable from object class to avoid memory leakage
```

```
* */
```

```
lateinit var batch: SpriteBatch
```

```
lateinit var shape:ShapeRenderer
```

```
lateinit var stage: Stage
```

```
var network: ArrayList<Image> = ArrayList()
```

```
lateinit var font: BitmapFont
```

```
lateinit var image_network_interface:Image
```

```
lateinit var image_spsc:Image
```

```
lateinit var image_db_vsim:Image
```

```
lateinit var image_db_device:Image
```

```
lateinit var image_reject_one:Image
```

```
lateinit var image_reject_two:Image
```

```
lateinit var image_spm:Image
```

```
lateinit var image_esimCons:Image
```

```
lateinit var image_activationUnit:Image
```

```
private var line_Width=3f
```

```
val contactS= ArrayList<Image>()
```

```
override fun create() {
```

```
    stage = Stage()
```

```

Gdx.input.inputProcessor = stage
shape = ShapeRenderer()
batch = SpriteBatch()
font =
FreeTypeFontGenerator(Gdx.files.internal("font/arial.ttf")).generateFont(FreeTypeFontGen
erator.FreeTypeFontParameter().also {
    it.size = 20
    it.color = Color.WHITE
})

image_network_interface = Image(Texture("network_interface.png"))
image_scale(image_network_interface,1.1f)

image_spsc = Image(Texture("spsc.png"))
image_scale(image_spsc,1.1f)

image_db_device = Image(Texture("db.png"))
image_scale(image_db_device,1.1f)

image_db_vsim = Image(Texture("db.png"))
image_scale(image_db_vsim,1.1f)

image_activationUnit = Image(Texture("activationUnit.png"))
image_scale(image_activationUnit,1.1f)

image_esimCons = Image(Texture("esimCons.png"))
image_scale(image_esimCons,1.1f)

image_spm = Image(Texture("spm.png"))
image_scale(image_spm,1.1f)

image_reject_one = Image(Texture("reject.png"))
image_scale(image_reject_one,.2f)

image_reject_two = Image(Texture("reject.png"))
image_scale(image_reject_two,.2f)

```

```
addContacts("019.png")
addContacts("017.png")
addContacts("015.png")
```

```
for (i in 1..3){
    val img: Image = Image(Texture("radio_tower.png"))
    img.width = (.8f * i)* img.width
    img.height = (.8f * i)*img.height
    network.add(img)
    stage.addActor(network[i-1])
}
```

```
stage.addActor(image_network_interface)
stage.addActor(image_db_vsim)
stage.addActor(image_spm)
```

```
stage.addActor(image_esimCons)
stage.addActor(image_db_device)
stage.addActor(image_spsc)
stage.addActor(image_activationUnit)
```

```
stage.addActor(image_reject_one)
stage.addActor(image_reject_two)
```

```
for (i in 0 until contactS.size){
    stage.addActor(contactS[i])
}
```

```
//Action
```

```
for (i in 0 until contactS.size) {
    contactS[i].addListener(object : ClickListener() {
        override fun clicked(event: InputEvent?, x: Float, y: Float) {
            super.clicked(event, x, y)
        }
    })
}
```

```

if (i%2==0 && !DailerDevice_one.Device_Locked){
    DailerDevice_one.Device_Locked = true
    if (i==0) {
        DailerDevice_one.NetworK_Index = 0
        SPM.RequestTo_connect(MNO_List.mno_list[0].id, "one")
    } else if (i==2){
        DailerDevice_one.NetworK_Index = 1
        SPM.RequestTo_connect(MNO_List.mno_list[1].id, "one")
    } else if (i==4){
        DailerDevice_one.NetworK_Index = 2
        SPM.RequestTo_connect(MNO_List.mno_list[2].id, "one")
    }
    DailerDevice_one.ContactS_Index = i
} else if (i%2==1 && !DailerDevice_two.Device_Locked){
    DailerDevice_two.Device_Locked = true
    if (i==1) {
        DailerDevice_two.NetworK_Index = 0
        SPM.RequestTo_connect(MNO_List.mno_list[0].id, "two")
    } else if (i==3){
        DailerDevice_two.NetworK_Index = 1
        SPM.RequestTo_connect(MNO_List.mno_list[1].id, "two")
    } else if (i==5){
        DailerDevice_two.NetworK_Index = 2
        SPM.RequestTo_connect(MNO_List.mno_list[2].id, "two")
    }
    DailerDevice_two.ContactS_Index = i
}
}
})
}

```

```

image_reject_one.addListener(object :ClickListener){
    override fun clicked(event: InputEvent?, x: Float, y: Float) {
        super.clicked(event, x, y)
        DailerDevice_one.rejectCall()
    }
})

```



```

image_reject_two.addListener(object :ClickListener){
    override fun clicked(event: InputEvent?, x: Float, y: Float) {
        super.clicked(event, x, y)
        DailerDevice_two.rejectCall()
    }
})

}

override fun render() {

    Gdx.gl.glClearColor(0f, 0f, 0f, 0f)
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
    if(Gdx.input.isKeyPressed(Input.Keys.ESCAPE)){
        Gdx.app.exit()
    }
    shape.begin(ShapeRenderer.ShapeType.Filled)
    shape.rect(image_db_device.x,image_db_device.y -
10f,(image_activationUnit.width+image_esimCons.width+100),(image_db_device.height+i
mage_esimCons.height+150), Color.DARK_GRAY, Color.CLEAR, Color.DARK_GRAY,
Color.DARK_GRAY)
    shape.rect(image_db_vsim.x,image_db_vsim.y -
10f,(image_db_vsim.width+image_spm.width+100),(image_network_interface.height+ima
ge_spm.height+150), Color.DARK_GRAY, Color.DARK_GRAY, Color.DARK_GRAY,
Color.CLEAR)
    shape.rect(stage.width-485f,895f,450f,10f)
    shape.rect(stage.width-485f,595f,450f,10f)

    //network interface to MNO

    if (DailerDevice_one.Device_Locked){
        device_MNO_write(DailerDevice_one.Network_Index,
DailerDevice_one.ContactS_Index, DailerDevice_one.Signal_Color)

networkINF_MNO_write(DailerDevice_one.Network_Index,DailerDevice_one.Signal_Col
or,-20f)

```

```

    }

    if (DailerDevice_two.Device_Locked){
        device_MNO_write(DailerDevice_two.Network_Index,
DailerDevice_two.ContactS_Index, DailerDevice_two.Signal_Color)

networkINF_MNO_write(DailerDevice_two.Network_Index,DailerDevice_two.Signal_Color,-8f)

    }

    //SPSC start

networkINF_MNO_write(MNO_List.mno_list.indexOf(SPSC_Runtime.profiles.mno),SPSC_Runtime.Signal_color,2f)

eSIMconst_MNO_write(MNO_List.mno_list.indexOf(SPSC_Runtime.profiles.mno),SPSC_Runtime.Signal_color)

    SPSC.SendSFlagToSPM()

    eSIMconst_SPSC_write(SPSC_Runtime.SPSC_to_eSIMconst_Signal_color)

    eSIMconst_ActivationUnit_write(SPSC_Runtime.eSIM_to_activeUnit_Signal_color)

    activationUnit_SPSC_write(SPSC_Runtime.SPSC_to_activeUnit_Signal_color)

    networkINF_SPM_write(SPSC_Runtime.Signal_color)

    //SPSC End
    // Interface boundaries start

    //end
shape.end()

stage.draw()
    setPosition()

```

```

batch.begin()
for(i in 0..2) {
    // val networkTopPos = Vector2(network[2].x, network[2].height)
    font.draw(batch, MNO_List.mno_list[i].id, network[i].x, 50f)
    font.color = Color.WHITE
    font.data.setScale(2f)

}

font.draw(batch, "GSMA: "+SPSC_Runtime.profiles.name,
image_activationUnit.x+20, image_activationUnit.y+80f)
font.draw(batch, "Receiver Device", image_db_device.x, image_db_device.y-30f)
font.draw(batch, "VSIM Server", image_db_vsim.x, image_spm.y-30f)
font.draw(batch, "Dialer Device One", image_reject_one.x -
(image_reject_one.width+220f), image_reject_one.y+40f)
font.draw(batch, "Dialer Device Two", image_reject_two.x -
(image_reject_two.width+220f), image_reject_two.y+40f)
batch.end()

}

fun device_MNO_write(networkIndex:Int, contactS_index:Int, color: Color){
    shape.rectLine( contactS[2].x-contactS[2].width,(contactS[ontactS_index].y-
10),network[networkIndex].x+(network[networkIndex].width/2)+5,(network[networkIndex]
.height+network[networkIndex].y-30), line_Width,color, color)
}

fun networkINF_MNO_write(networkIndex:Int, color: Color, extraY:Float){

shape.rectLine(image_network_interface.x,(image_network_interface.height+image_net
work_interface.y-
57+extraY),network[networkIndex].x+(network[networkIndex].width/2)+5,(network[netwr
okIndex].height+network[networkIndex].y-10+extraY),line_Width, color, color)

}

```

```

fun eSIMconst_MNO_write(networkIndex: Int, color: Color) {
    shape.rectLine(image_esimCons.x+image_esimCons.width-
58,(image_esimCons.y+8),network[networkIndex].x+(network[networkIndex].width/2)+5,(
network[networkIndex].height+network[networkIndex].y-10),line_Width, color, color)
}

```

```

fun eSIMconst_SPSC_write(color: Color) {

shape.rectLine(image_esimCons.x+3,(image_esimCons.y+8),image_spesc.x+(image_spesc.w
idth),((image_spesc.height/2)+image_spesc.y-10),line_Width, color, color)
}

```

```

fun eSIMconst_ActivationUnit_write(color: Color) {

shape.rectLine(image_esimCons.x+3,(image_esimCons.y+(image_esimCons.height/2)),im
age_activationUnit.x+(image_activationUnit.width),((image_activationUnit.height/2)+ima
ge_activationUnit.y-10),line_Width, color, color)
}

```

```

fun activationUnit_SPSC_write(color: Color) {
    shape.rectLine(image_activationUnit.x +
image_activationUnit.width/2,image_activationUnit.y,image_db_device.x+(image_activat
ionUnit.width/2),(image_db_device.y+image_db_device.height-10),line_Width, color,
color)
}

```

```

fun networkINF_SPM_write(color: Color) {
    shape.rectLine(image_network_interface.x +
image_network_interface.width/2,image_network_interface.y,image_network_interface.
x + image_network_interface.width/2,(image_db_vsim.y+image_db_vsim.height-
10),line_Width, color, color)
}

```

```

fun setPosition(){

network[0].setPosition(stage.width/3,0f)
network[1].setPosition(stage.width/6,0f)
image_network_interface.setPosition(stage.width-10-
image_network_interface.width,stage.height-10-(image_network_interface.height))
    image_spm.setPosition(stage.width-10-image_spm.width,(stage.height-10-
(image_network_interface.height + image_spsc.height + 100)))
    image_db_vsim.setPosition(stage.width-70-(image_spsc.width*2),(stage.height-10-
(image_network_interface.height + image_spsc.height + 100)))
    image_activationUnit.setPosition(0f,(stage.height-10-(image_activationUnit.height))
    image_db_device.setPosition(0f,(stage.height-100-(image_activationUnit.height*2))
    image_spsc.setPosition(image_db_device.width+ image_db_device.x-10,(stage.height-
100-(image_activationUnit.height*2)))
    image_esimCons.setPosition(image_activationUnit.width+100,(stage.height-10-
(image_esimCons.height)))

    var device_one_width:Float = 0f
    var device_two_width:Float = 0f

    for (i in 0 until contactS.size){
        if (i%2==0){
            contactS[i].setPosition(stage.width-
(10+contactS[i].width+device_one_width),stage.height-10-(contactS[i].height+600))
            device_one_width+= contactS[i].width
        }else{
            contactS[i].setPosition(stage.width-
(10+contactS[i].width+device_two_width),stage.height-10-(contactS[i].height+900))
            device_two_width+= contactS[i].width
        }
    }

    image_reject_one.setPosition(stage.width-(40+image_reject_one.width),stage.height-
10-(image_reject_one.height+720))

    image_reject_two.setPosition(stage.width-(40+image_reject_two.width),stage.height-
10-(image_reject_two.height+1020))

```

```

}

fun image_scale(image: Image, size:Float){
    image.width = size * image.width
    image.height = size * image.height
}

fun addContacts(url:String){
    for (i in 0 until 2){
        contactS.add(Image(Texture(url)).also {
            image_scale(it,1.1f)
        })
    }
}

override fun dispose() {
    batch.dispose()
    stage.dispose()
    font.dispose()
    shape.dispose()
}
}

```

APPENDIX – B

```
/* Server side*/

enum class Flags {
    Online,
    Offline,
    Active,
    Busy,
    Free
}

object SPM_DB {
    var profiles= ArrayList<Profile>()
}

object MNO_List {
    val mno_list:ArrayList<MNO> = ArrayList()

    //Test Purpose
    fun addNetwork(){

        mno_list.clear()

        mno_list.add(MNO().also {
            it.id = "019"
            it.name = "Banglalink"
        })

        mno_list.add(MNO().also {
            it.id = "017"
            it.name = "Gp"
        })

        mno_list.add(MNO().also {
            it.id = "015"
            it.name = "T-Talk"
        })
    }
}
```

```

class MNO{
    //according to VSIM
    //I am here adding only network id and name to test
    lateinit var id:String
    lateinit var name:String
}
object SPM{
    /*
    * This class manages all process of SPM which located to the VSIM Server
    * I added a thread instead of presents of render() thread because of long process loop
    * and multiple data transfer to increase its performance
    * I made this class as static class to avoid memory leakage
    * */
    val Signal_Color = Color.GREEN
    val WAIT_TIME:Long = 2
    fun RequestTo_connect(profile_identity:String,deviceId:String){
        if (SPSC_Runtime.Connected_Device=="") {
            SPSC_Runtime.Connected_Device = deviceId
        }
        Thread(Runnable {
            for (i in 0 until SPM_DB.profiles.size){
                if (profile_identity==SPM_DB.profiles[i].name
                    && SPM_DB.profiles[i].flag == Flags.Active.name){
                    TimeUnit.SECONDS.sleep(5)
                    //route back
                    if (SPSC_Runtime.Connected_Device=="one") {
                        DailerDevice_two.Signal_Color = SPM.Signal_Color
                        TimeUnit.SECONDS.sleep(WAIT_TIME)
                        DailerDevice_two.Signal_Color = DailerDevice_two.Signal_Color_default
                        DailerDevice_two.rejectCall()
                    }else if (SPSC_Runtime.Connected_Device=="two"){
                        DailerDevice_one.Signal_Color = Signal_Color
                        TimeUnit.SECONDS.sleep(WAIT_TIME)
                        DailerDevice_one.Signal_Color = DailerDevice_one.Signal_Color_default
                        DailerDevice_one.rejectCall()
                    }
                }
            }
        })
    }
}

```



```

} else if (profile_identity==SPM_DB.profiles[i].name
    && (SPM_DB.profiles[i].flag == Flags.Free.name
        || SPM_DB.profiles[i].flag == Flags.Online.name)){
    if (SPSC_Runtime.Connected_Device=="one") {
        SPSC_Runtime.Signal_color = DailerDevice_one.Signal_Color
    } else if (SPSC_Runtime.Connected_Device=="two"){
        SPSC_Runtime.Signal_color = DailerDevice_two.Signal_Color
    }
    SPSC_DB.profiles[i].flag = Flags.Active.name
    SPSC_Runtime.profiles.flag = Flags.Active.name
    SPM_DB.profiles[i].flag = Flags.Active.name
    SPSC_Runtime.eSIM_to_activeUnit_Signal_color =
SPSC_Runtime.Signal_color
    SPSC.SendSFlagToSPM()

    //route
} else if (profile_identity==SPM_DB.profiles[i].name
    && (SPM_DB.profiles[i].flag == Flags.Offline.name)){
    //profile activation request send
    if (!DailerDevice_one.Device_Locked || !DailerDevice_two.Device_Locked &&
SPSC_Runtime.Connected_Device!="") {
        SPSC_Runtime.Signal_color = Signal_Color
        SPSC_Runtime.SPSC_to_eSIMconst_Signal_color = Signal_Color
        //wait()
        TimeUnit.SECONDS.sleep(WAIT_TIME)
        SPSC.ActiveRequested_Profile(profile_identity)
        SPSC_Runtime.Signal_color = SPSC_Runtime.Signal_color_default
        SPSC_Runtime.SPSC_to_eSIMconst_Signal_color =
SPSC_Runtime.Signal_color_default

        //recursive
        SPM.RequestTo_connect(profile_identity, deviceId)
    }
}
}
}).start()
}
}

```

APPENDIX – C

```
/* Device Context switching module*/

object SPSC_Runtime {
    val Signal_color_default= Color.WHITE
    var Signal_color= Signal_color_default
    var profiles:Profile = SPSC_DB.profiles[0]
    var Connected_Device = ""
    var SPSC_to_eSIMconst_Signal_color= Signal_color_default
    var SPSC_to_activeUnit_Signal_color= Signal_color_default
    var eSIM_to_activeUnit_Signal_color= Signal_color_default
}

object SPSC_DB{
    val profiles = ArrayList<Profile>().also {
        it.add(Profile("019",Flags.Online.name,MNO_List.mno_list[0]))
        it.add(Profile("017",Flags.Offline.name,MNO_List.mno_list[1]))
        it.add(Profile("015",Flags.Offline.name,MNO_List.mno_list[2]))
    }
}

class Profile{
    lateinit var name:String
    lateinit var flag:String
    lateinit var mno: MNO
    lateinit var profileRawData: eSIM_Profile

    constructor(name: String,flag:String, mno: MNO){
        this.flag = flag
        this.name = name
        this.mno = mno
    }
}
}
```

```

class eSIM_Profile{
    //According to eSIM
    //we are not going to change here
}

object SPSC {

    fun ActiveRequested_Profile(profileIdentity:String){

        for (i in 0 until SPSC_DB.profiles.size) {
            if (SPSC_DB.profiles[i].name != profileIdentity
                && SPSC_DB.profiles[i].flag==Flags.Online.name){
                SPSC_DB.profiles[i].flag = Flags.Offline.name
            }
        }
        for (i in 0 until SPSC_DB.profiles.size) {
            if (SPSC_DB.profiles[i].name == profileIdentity) {

                SPSC_Runtime.SPSC_to_activeUnit_Signal_color = SPM.Signal_Color
                SPSC_DB.profiles[i].flag = Flags.Online.name
                SPSC_Runtime.profiles = SPSC_DB.profiles[i]
                SendSFlagToSPM()
                TimeUnit.SECONDS.sleep(SPM.WAIT_TIME)
                SPSC_Runtime.SPSC_to_activeUnit_Signal_color =
                SPSC_Runtime.Signal_color_default
            }
        }
    }

    fun SendSFlagToSPM(){

        SPM_DB.profiles.addAll(SPSC_DB.profiles)

    }
}

```

APPENDIX – D

```
/* additional dialer device, Testing purpose*/

object DailerDevice_one {

    val Signal_Color_default = Color.MAGENTA
    var Device_Locked = false
    var NetworK_Index = 0
    var ContactS_Index=0
    var Signal_Color = Color.MAGENTA

    fun rejectCall(){
        DailerDevice_one.Device_Locked = false
        if (SPSC_Runtime.Signal_color == DailerDevice_one.Signal_Color_default) {
            SPSC_Runtime.Signal_color = SPSC_Runtime.Signal_color_default
        }

        if (SPSC_Runtime.Connected_Device=="one"){
            SPSC_Runtime.Connected_Device=""
            SPSC_Runtime.profiles.flag = Flags.Online.name
            SPSC_Runtime.eSIM_to_activeUnit_Signal_color =
SPSC_Runtime.Signal_color_default
        }
        DailerDevice_one.Signal_Color = DailerDevice_one.Signal_Color_default
    }
}

object DailerDevice_two {

    val Signal_Color_default = Color.YELLOW
    var Device_Locked = false
    var NetworK_Index = 0
    var ContactS_Index=0
    var Signal_Color = Color.YELLOW

    fun rejectCall(){
        DailerDevice_two.Device_Locked = false
```

```

if (SPSC_Runtime.Signal_color ==DailerDevice_two.Signal_Color_default ) {
    SPSC_Runtime.Signal_color = SPSC_Runtime.Signal_color_default
}
if (SPSC_Runtime.Connected_Device=="two"){
    SPSC_Runtime.Connected_Device=""
    SPSC_Runtime.profiles.flag = Flags.Online.name
    SPSC_Runtime.eSIM_to_activeUnit_Signal_color =
SPSC_Runtime.Signal_color_default

}
DailerDevice_two.Signal_Color = DailerDevice_two.Signal_Color_default
}
}

```