

# **AOTA-ADVANCE OVER THE AIR**

Project report submitted in partial fulfillment of the requirement for the degree  
of Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

Abhishek Bhardwaj(161312)

Under the supervision of

Mr Vikash Kumar

to



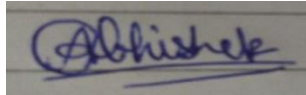
Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234,**

**Himachal Pradesh**

## DECLARATION

I hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.



Abhishek Bhardwaj(161312)

This is to certify that the work titled “AOTA – Advanced Over The Air” submitted by “Abhishek Bhadwaj of B. Tech of Jaypee University of Information Technology University, Solan has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.



Mr Vikash Kumar

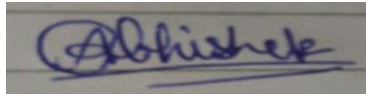
(Senior Technical Lead and Manager,Thales Group)

## ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Mr. Vikash Kumar, Senior Technical Lead and Manager, Thales Group for his generous guidance, help and useful suggestions.

I also wish to extend my thanks to my friends and colleagues for their insightful comments and constructive suggestions to improve the quality of this project work.

I will also like to thank Mr Wasin Khan, Senior Technical lead, and Mr Akram Raza, Senior Technical Lead, for there continuous support and also like to thank all the team member who always helped me when needed.



Signatures of Students:

Abhishek Bhardwaj (161312)

## TABLE OF CONTENTS

SNO.	Page No.
List Of Figures	v
List of Tables	v
Abstract	8
<b>1. Introduction</b>	<b>1-6</b>
1.1 General Introduction	
1.2 Problem Statement	
1.3 Objective	
1.4 Methodology	
1.5 Organization	
<b>2. Literature Survey</b>	<b>7-17</b>
<b>3. System Development</b>	<b>18-53</b>
3.1 Overall description of the project	
3.2 Functional and Non-Functional requirements	
3.3 Design and Diagram	
3.3 Implementation details and issue	
<b>4. Performance Analysis</b>	<b>54-59</b>
4.1 Agile Methodology	
4.2 Performance Analysis Details	
4.3 Improvements	
<b>5. Conclusion</b>	<b>60</b>
5.1 Future Scope	
<b>Reference</b>	<b>61</b>

## LIST OF FIGURES

Fig/Table no.	Contents	Page
Fig 2.1	UICC Structure	7
Fig 2.2	HTTP Connection Handshake in AOTA	11
Fig 3.3.1	Subscription management use case diagram	21
Fig 3.3.2	Use case diagram of LPM	21
Fig 3.3.3	Use case diagram of STH	22
Fig 3.3.4	RUM Targeting service grouped to same ICCID	22
Fig 3.3.5	STH detailed modules.	23
Fig 3.4.1.1	RUM Provisioning UseCase	24
Fig 3.4.1,2	Subscription Life Cycle	30
Fig 3.4.1.2(b)	Assign MSISDN	32
Fig 3.4.1.2(c)	Change IMSI	33
Fig 3.4.2	Rum Activation	36

Fig 3.4.2(b) Create Subscription	38
Fig 3.4.2(c) Register Service	39
Fig 3.4.2(d) Visit Flow	41
Fig 3.4.3 RUM Deactivation	44
Fig 3.4.4 RUM ReActivation	45
Fig 3.4.4(b) Class Sequence	47
Fig 3.4.6 RUM-MMGT Integration	50
Fig 3.4.6(b) Card Poll	51
Fig 3.4.6(c) Otaip triggers RUM	52
Fig 4.1 Agile Methodology	54

## **LIST OF Tables/Graph**

Graph/Table no.	Contents	Page no
Graph 4.2.1	Mobile Review	55
Graph 4.2.2	Overview of calculated averages	56

## **ABSTRACT**

The Advanced Over The Air (AOTA) enables the communication between the user and the card. It manages 2G, 3G and 4G card and it begins execution at the time of sim purchase and performs the activation operation in which it adds different services in the card and activate it for us and after that if card further wants to add an additional services then that must be done through a polling request. Aota interact with card with either of the one transport medium out of SMS and HTTP with push and pull mode.

RUM is the processing module of AOTA which carries out a lot of major functionalities of AOTA including the generation of APDU commands for UICC management. This is the principle objective of RUM and because it manages a lot of functions for remotely managing the cards, it is the central processing engine of AOTA. It is also used for other functions like in activating a card, updating file or application in card and send a force polling SMS.

LinQUs Provisioning Manager is part of Advanced OTA Solution, a fully-integrated software solution providing you with a robust, reliable and scalable infrastructure with which to manage your (U) SIM installed base. LPM allows automatic and concurrent provisioning of the following Advanced OTA Solution products following Advanced +OTA Solution products: OTA Manager, Device Detection Manager, SRM.



# Chapter 1

## Introduction

### 1.1 INTRODUCTION

A supporter personality module or endorser distinguishing proof module (SIM), known as a SIM card, is a coordinated circuit that mean to safely store the universal portable endorser character (IMSI) number and its related key, which are utilized to recognize and confirm the supporters on versatile communication gadgets, (for example, cell phones and PCs). It is additionally conceivable to store the contact data on numerous SIM cards. SIM cards are constantly utilized on GSM telephones; for CDMA telephones, they are just and just required for fresher LTE-proficient handsets. SIM cards are utilized in cell phones, satellite telephones, keen watches, PCs, or cameras.

The SIM circuit is a significant piece of the capacity of a general coordinated circuit card (UICC) physical brilliant card, which is normally made of PVC with installed contacts and semiconductors. SIM cards are the transferable between various cell phones. The first UICC savvy cards were the size of the credit and bank cards; sizes were diminished to a few times throughout the years, generally keeping electrical contacts the equivalent, with the goal that a bigger card could be utilized to chop down to a littler size.

A SIM card has its own exceptional sequential number (ICCID), global portable supporter character (IMSI) number, security validation and figuring data, impermanent data which are identified with the nearby system, a rundown of the administrations that the client approaches, and afterward two

passwords: an individual distinguishing proof number (PIN) is utilized for conventional use, and an individual unblocking code (PUC) for PIN opening.

The versatile environment is changing and getting perpetually open and associated step by step, with the quantity of various IoT gadgets and secure components available for use blasting.

Versatile system administrators (MNOs) need to progressively deal with the gadgets, UICCs and secure components over all systems, including 2G, 3G, 4G, 5G, Wi-Fi and CDMA, to ensure the best end-client that can encounter empowering new high-potential use cases, for example, improved portable broadband, gigantic IoT and M2M.

This arrangement is intended for ideal productivity for a superior MNO experience over all systems, and engages MNOs as the administration empowering agents, specifically on LTE and 5G systems. OTA empowers a Network Operator that acquaints new SIM administrations with alter the substance of SIM cards in a fast and financially savvy way. OTA depends on customer/server design where toward one side there is an administrator back-end framework (client care, charging framework and application server.) and at the opposite end there is a SIM card. The administrator's back-end framework sends some assistance solicitations to an OTA Gateway which at that point changes the solicitations into a Short Messages and afterward sends them onto a Short Message Service Center (SMSC) which transmits them to one or a few other SIM cards in the field. Along these lines, Over-The-Air (OTA) is an innovation that updates and changes the

information in the SIM card without reissuing it. Surely, the end client can get uncommon messages from the administrator, download or initiate new administrations on his phone, and considerably more, without coming back to a retail outlet.

Progressed OTA (Over the Air) is an answer for oversee a transient other Secure Elements, for example, This protected component is fit for executing security basic capacities and contains confirmation insider facts for associating with GSM/LTE systems. AOTA is initial an answer dependent on RoH (Remote administration over HTTP) in view of Pull mode. The activities which are can performed on the cards regularly are known as remote application the board (RAM) and remote document the executives (RFM) through push and pull mode. The push mode sends APDU orders promptly to the card which are activated by OTA and the force mode is activated by the card, where programmed demand is sent from the card to the server on normal interims for remote administration of the card. AOTA has numerous segments yet the focal segment is RUM. It deals with a great deal of functionalities for card the executives, for example, provisioning of the card, actuation and deactivation, dynamic retry and system location. RUM can likewise produce APDU orders for executing all the capacities which may likewise incorporate sending any update through various administrations. It likewise speaks with practically the entirety of the parts of AOTA.

## **1.2 Problem Statement**

The mobile connectivity is expanding day by day with the need of more

secure, more reliable, more accurate system to interact with smart cards remotely on the field. AOTA provides solution to all these needs via sending commands to the remote card for the application and file management. Earlier in case of OTA only push mode was available, which uses SMS as transport mode to send services to card on field. It has many disadvantages associated with. Pull mode in which the data is transferred through HTTP used is used by Aota. Out of many AOTA components, RUM is the like the CPU of Aota and it also helps in managing the card remotely through many functions. It is the need to help and communicate with many components and also to provide the services which would then update files or applications. RUM also helps in generation of a set of APDU command which are sent by the card to the server for requesting any service.

Until now, NORAM MNOs strategy was based on polling at switch on for urgent situation. If any problem occurs, MNO support team could launch services and just proceed with customer to a mobile equipment power cycle.

The drawback of this approach is that the number of empty polling reaches a level really too important. A lot of solution resources serve only empty polling use-case and the flow continues to growth with the number of subscriptions.

To resolve this problem introduction of new module called STH was introduced, main intention of this component, jointly with the SEE Emulator, is to re-conciliate all the OTA transport protocols around the “pull” paradigm.

As the pull mode is always started via sim card , so to solve that problem , STH was introduced to make the card come back to AOTA with polling request and execute the service after establishing a successful connection.

### **1.3 Objective**

The main objective of AOTA is to provide user services so that they can potentially communicate with anyone. SIM cards managed by Aota are managed and new services are added to the sim card when the card request for a service through the pull mode which was introduced newly to the AOTA.

### **1.4 Methodology**

The work of Aota Starts after user purchases a new sim. MNO passes the information to the AOTA

The first major step is provisioning of the card which is a time consuming process and in it AOTA include all the unique feature to the card like its ICCID, IMSI and MSISDN and also store the card information in itself

Provisioning can be done through various process, it can be done through the Linqous UI platform through Batch Load. Also it can be done using RUM through SOAP UI or it can be done through LPM.

After provisioning it complete next step is to activate the card and adding various services that the user has wished for. Also if user has not requested for any services empty activation is also there in which card is simply activated without any services being added to it

Admin services later end is done when Aota receives a polling request through the card. Polling request can be generated from card side(real polling) as well as from AOTA side(emulated polling)

STH(session trierring helper) helps in emulated polling and SEE as a emulator in emulating polling

Aota can also put card in various phase like Deactivation,suspended or delete

A card suspended or deactivated can be brought to activated state but a card deleted cannot be bring back to activated state

## **1.5 Organization of Report**

Chapter 1 gives a general introduction about the sim and how does the Aota preprocesses a sim .

Chapter 2 gives the literature survey related to sim management

Chapter 3 deals with how the various algorithm works

Chapter 4 deals with the type of model used for making the product and what are the performance of the product and how can futhur improvement be made on thatproduct

Chapter 5 finally conclude this report with a conclusion and also specifying the future vision for the product

## CHAPTER 2

### Literature Survey

Smart card sent request to the servers for requesting any service in the form of Application Data Protocol Unit(APDU) and on receiving these APDU server decode the APDU and sent the response back in form of APDU. There are 2 types of APDU, one is response APDU and other one is command APDU. Command APDU contains the request made by the card to the server and it contains 4 mandatory bytes which are CLASS, INSTRUCTION, PARAMETER1 and PARAMETER2 and these APDU are ranging from 0 to 65 535 bytes of data. A response APDU is sent by the card to the reader to give information about the status of message send – it also contains data ranging from 0 to 65 536 bytes of data, and 2 the mandatory one are status bytes (SW1, SW2).

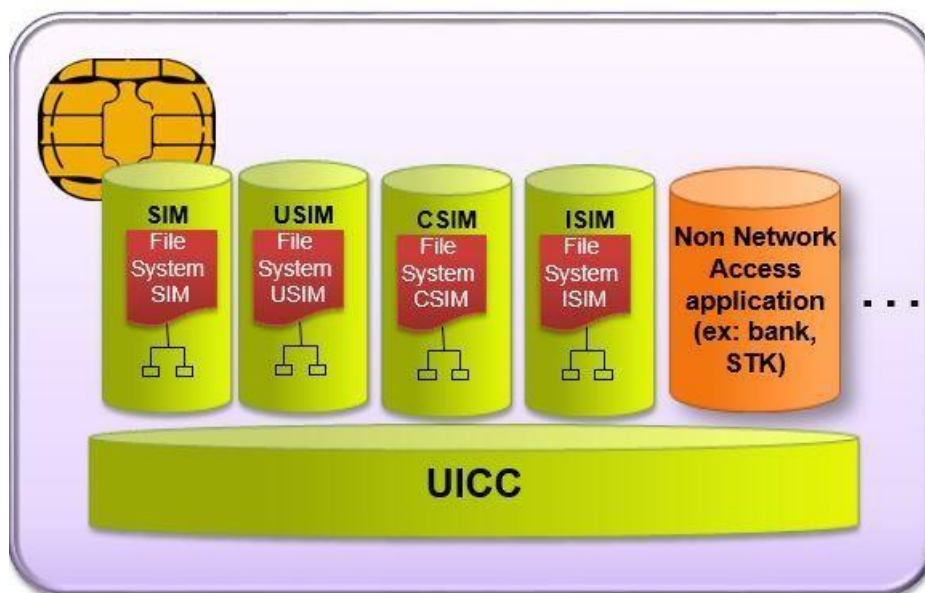


Fig 2.1 UICC Structure

UICC is needed to be provisioned first before doing any operations. This pre-provisioning mainly means to provide card profile, attachment information, 03.48 security keys with 03.40 header and IMSI, ICCID which can uniquely identified the card info regarding 2G/3G cards.

For 4G cards, additional security information are added TLS-PSK (UICC card is present at operator's premises but it has neither been sold, changed or used). An MSISDN will be assigned by the backend operator to the existing subscription to activate the subscription on platform side.

### **Subscription Life Cycle**

#### **(Devicechange)**

A subscription has different states. A card can be Lost/Damaged any time and in that case AOTA need to be informed so that It can cancel the subscription of that card and the person who found that card will not be able to misuse it.

Another situation may be that the mobile phone of the user has been badly damaged and is in no situation to be used again. In those situation Aota needs to be informed so that it can make the required changes in the subscription. This use case must have a DDE applet installed and must must activate the UCII whatever the protocol is used(Http or SMS)

### **Single Campaign Management in Push Mode (via SMS)**

In Advanced OTA Solutions V6.5.3, the administrator first builds a scenario that it is puts all the services that needs to be updated on the card and after the scenario is created, target group is created containing ICCID of cards on which the given services specified in the scenario needs to be updated and start and end time is defined for the running of the campaign. When the campaign reach its start time then all the necessary update are done and end user should



get notified about it through SMS.

1. Define all the necessary services and the respective card profile you want to target.
2. Creating the campaign and then giving the required starting and ending time variation .Now we need to perform the update for the pending RFM and RAM operation
1. These update are then received by device and it notify the EMSE that T is correctly received.
2. Now you have to keep a track of the percentage number of subscriber in which the campaign has been correctly received.

There is a Last Polling date in the post campaign for all the user who has not polled any services during the defined time.

1 Operationals Campaigns widget in the pull mode, Manages the Target card Groups pages.

2 Now you have to create various kind of service based scenario in Campaign management and also need to define the validity period in which the campaign is not invalid.

3 Receive the relative update.

Now you have to keep a track of the percentage number of subscriber in which the campaign has been correctly received.

Device Detection: Device Detection Management are used to find the information of the device that the user belonging to specific group are using. Since sims are uniquely identified through ICCID or IMSI or MSISDN. Which device is being used should be added in the subscription of SIM.

**HTTP:** Hypertext Transfer Protocol

HTTP is the transport medium that is being used for the polling in case of real polling or force polling. In terms of real polling, the polling is initiated from the card side and the AOTA sends the APDU through HTTP medium while in case of force polling, STH triggers the polling with the help of SEE and forces card to poll. HTTP has an advantage over SMS that it can be used to send a large amount of data.

A status of the requested service is being sent back to the client in the form of APDU command. When the client sends a request to server:

- The security of the request is defined in the header while data is being present in the body.
- The targeted server is being identified by a special value that is defined in the header

Main Advantages of HTTP in AOTA :

- The HTTP protocol is chosen because it can send a large amount of data.
- Also the Internet is available everywhere.
- It reuses the existing tools hence cost reduction is there.
- It can be developed on the existing network easily..

Main Principles are :

The underlying lists show the necessary conditions for the successful operation needed on OTA over HTTP:

- The UICC card must be an HTTP client.
- The AOTA server is the HTTP server.

- The protocol with the remote servers is based on HTTP POST.
- Connection is established when the initiative of the UICC card is done.

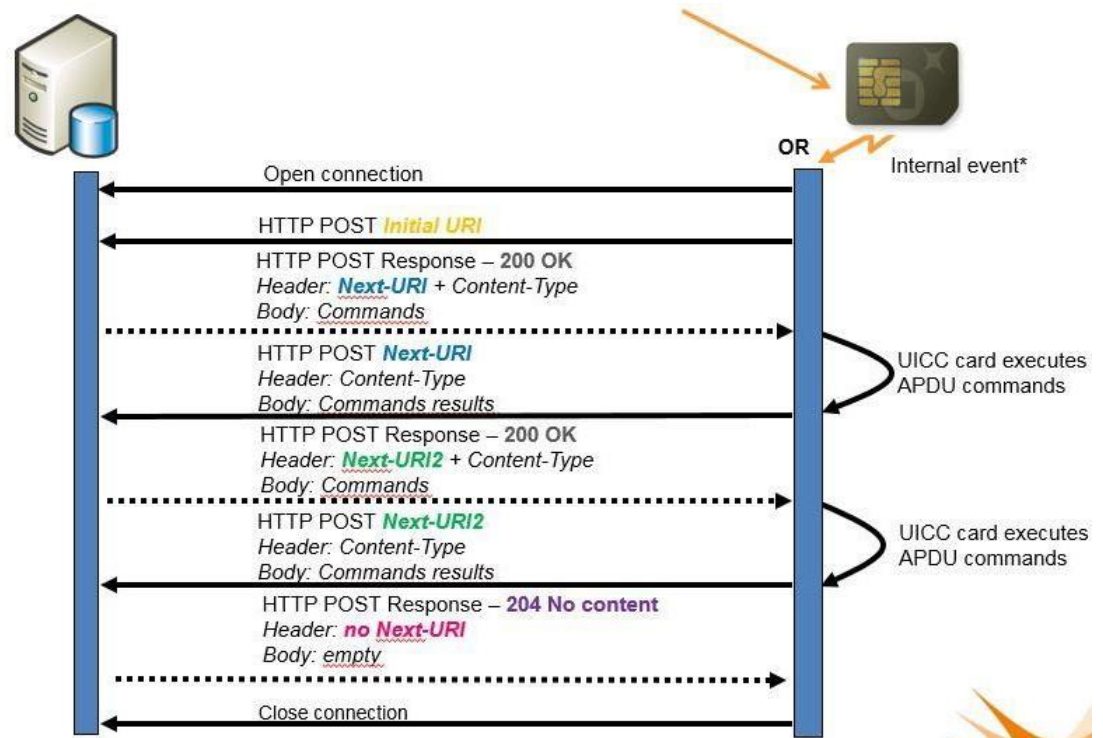


Fig 2.2 HTTP Connection Handshake in AOTA

- The general dialog says that the UCCI card must send the APDU request for the AOTA for an service update they want to do.
- After that HTTP POST messages are then sent by the Admin Agent from the AOTA server requesting the commands to be executed by the Admin Agent. The same HTTP POST request is then used to return the results of the command that has being succesfully executed.
- The Content-Type field is present that specifies the type of message that are been used to indicate the RFM/RAM

commands or responses that are being transported.

- The body likewise contains the information that are being related to the sort of message. The substance of the body is additionally reliant on the Content-Type of its correspondence. The APDU and POR(proof of receipt) are additionally remember for the body of the message.
- The Next set of URI is also being provided by AOTA to Agent in the form of HTTP POST response if the additional commands needs to be executed.

Next URI Mechanism: The Next-URI instrument has permitted the utilization of POST strategy as it were. The accompanying has indicated the procedure stream of how the association is creatures built up between the servers and the card. As the card being the customer, it is capable to open these association. It can likewise be utilized to activated by an inward occasion from an application, for example, a surveying demand or by an outside occasion, for example, a SMS. The card then opens the connection.

- 1 After the connection is established the HTTP request is then sent to the server loaded with the address of that and also the name of application that is being target.
- 2 After the request is received by the server, it respond back to the client with the set of APDU command that specifies the status of request.
- 3 The card read the APDU response send by the server and then response back to it.
- 4 The server then executes the request and sends the APDU command back to the client with a POP(proof of Receipt) if it is asked by the client

- 5 This sequence of exchanging keeps on going until the servers sends back status as 204 which specifies that there are no more command to be exchanged

OTA over HTTPS: In order to secure the communication between the AOTA server and the UICC card, the cryptographic protocol has been used—Transport Layer Security Pre-Shared Key(TLS/PSK).

Note: Only the message body is been encrypted.

1. The card stores the ICCID and PSK key to the Admin Agent. The master key and ICCID are used to compute for the PSK.
- 2 The ICCID and PSK ID are then being sent to the OTA server.
- 3 AOTA server stores only the master key. For each connection established, the AOTA server computes the TLS session key with ICCID, master key and a random key sent by the card.Hence the secret key used for communication is always unique and is found out by A3 or A8 Algorithm.

### **Retry Mechanism**

The Admin Agent is also a liable association with the AOTA server and for the purpose of achievement of the meeting.

On the off chance that a correspondence blunder happens during the preparing of the organization stream, the Admin Agent will attempt to reconnect as per a card backer explicit retry approach:

The Admin Agent makes a few endeavors for continuing the organization meeting. A holding up period between two endeavors and the most extreme number of endeavor is indicated by the retry strategy. On the off chance that the correspondence is restored, the

Admin Agent attempts to continue the HTTP

Exchange by exploring the last URI of this organization meeting .

On the off chance that the greatest number of endeavors has been reached, the organization meeting solicitation will be deserted.

On the off chance that the TLS meeting neglected to set up the association for security/approval reason, the organization meeting will be quickly disposed of.

**MMGT Pull Mode:** AOTA is the business' first stage to empower the force mode for crusade refreshes whereby an armada of UICCs are refreshed without a moment's delay to reflect membership changes. For instance, to invigorate wandering inclinations.

As it were, the UICC in the cell phone likewise starts the association for an update consequently, in light of a preplanned plan. This conveys considerably more productive update battles, another significant advantage to administrators.

The draw approach advances the achievement rate by arriving at all dynamic supporters for all intents and purposes. It decreases the weight on OTA server and dispenses with the need to make a huge battle group. It focuses server assets on associated gadgets, dissimilar to the push mode which frequently endeavors to start the update to cell phones that are detached which brings about the misuse of server assets and causing numerous futile retries.

Furthermore, AOTA depends on an associated mode that essentially lessens the quantity of issues caused when a cell phone loses its system association in an update, and recoups flawlessly when it happens. The highlights in this arrangement are intended for ideal proficiency for a superior MNO experience over all systems.

**Battle Management Push Mode:** AOTA is very much dependent on crusade

administrators to perform push battles on cards that are not utilizing pull mode.

AOTA offices the blend of both approaches all together for mass administration of dynamic reports on cards.

Brought together CMM: furthermore of the two crusade motors depicted in the past segments, Advanced OTA Solution. Mass Manager permits the joining of both modes in a solitary battle to deal with a heterogeneous arrangement of cards utilizing different bearers (SMS and HTTP) exploiting pull mode as regularly as could reasonably be expected.

This assuagement presents the instrument of programmed carrier determination streamlining tasks as far as observing and detailing.

WebLogic is a sort of server programming application that runs over center level, between back-end databases and related program based slim customers. WebLogic is one of the main web based business online exchange handling (OLTP) stage, created to associate all clients in a conveyed registering condition and to encourage the joining of centralized computer applications with the dispersed corporate information and applications.

WebLogic server depends on Java 2 Platform, Enterprise Edition (J2EE), the standard stage which is utilized to make Java-based multi-level venture applications. J2EE stage innovations which were created through the endeavors of BEA Systems and different merchants likewise as a team with the fundamental designer, Sun Microsystems. Since these applications are the normalized modules, it can computerize numerous framework level errands that would be in any case have requested programming time.

Subscriber Care Interface – It is the GUI interface for AOTA. This workspace contains several pages: From this interface, a customer can handle various operations like provisioning of UICC, running campaign, creation of template (Template is a Service with pre-defined parameters which can be used for

running campaign and launching unitary services).

### **SMS solicitation to card.**

Provisioning has been utilized for cards and also there is a security related provisioning which permits you to make cards in bunches by means of SCI/CCI.

Endorser Repository is utilized for information reference, (for example, MNO, model and brand) the executives and supporter seeing which permits you to see gadgets have a place with the specific endorser and activities made

UICC-SE Repository is utilized for card example seeing, card profile review and creation.

It incorporates both the repo and CCI card the board parts too.

Applet Repository is utilized for applet the board which permits you to see and make applets.

OTA Service Repository has been utilized for administrations definition and the board which permits you to make/evacuate administrations.

RHS is used to store the history of the command executed and is also utilized for HTTP activity. Gadget Tracking is utilized for Device Detection Manager which records the DM occasion, for example, changing portable of memberships.

RUM Template Manager is a kind of GWAF-based application utilized by overseers to effectively deal with the RUM layouts. Contingent upon the entrance rights, the client can make, erase, or update formats. A layout contains a lot of foreordained parameters to be applied remotely to the SIM card records by utilizing the chose RUM assistance.

The layout characterized by the Template Manager is utilized in MMGT situation.



Administrations: To make a layout, from the Operation Template select a Management page, trailed by RUM-Template Manager. The Template Manager Widget is shown.

The motivation behind RUM Services are that it send data in large number and also it create APDUs for Sim Cards.

One RUM Service is answerable for overseeing one element (record or application) in the card.

- A RUM Service mainly has two primary assignments.
- Produce most APDUS demands for cards
- Process reaction APDUs got from cards

In code level, RUM assistance is a Java class that customer can execute

- Customer can give certain number of parameters to the administration
- Generation of these APDU changes relying upon the gave parameters
- Response from the card is likewise prepared by a similar assistance
- Service may or maynot perform adjustments to the card record picture on the AOTA database so it is in accordance with the document on the card.

## **Chapter-3**

### **System Development**

#### **2.1 Description of the project:**

Thales AOTA management ensures the secure management of all types of sim card remotely. It ensures a secured management. It was basically OTA meaning over the air and later it became AOTA where A referred to as Advanced. It uses SNMP to establish network connection to exchange data between mobile and card to external server. There are 2 types of transport mechanism which are mainly used name HTTP and SIM. First SIM was the main type of transport mechanism which was used over the RCA, later with the introduction of RUM there felt a need of a transport mechanism that can transport a large number of data effectively, so HTTP was introduced. Now, we have both the transport mechanism readily available with us and we can use any one of them according to the need. If the need arises we can also use both .

AOTA is Thales latest version of the over-the-air platform which is being enabled mobile network operators to manage all the 2G, 3G and the LTE benefits of the push(RCA) and pull(RUM) modes of the the cards. Pull mode is one of the important innovation that was introduced by AOTA in which all the UICC initiates the dialogue.

The solution delivers excellent:

- Unwavering quality – for the crucial application and for membership

the executives

- Security – for the open IP world
- Success rate – on UICCs or gadgets with its surveying instruments  
programmed update
- Performance – with wise mass administration of different powerful  
updates

### Advantages

#### For end-clients

- Faster access to substance and applications on account of higher  
transmission capacity
- Seamless actuation of new administrations
- Access to reliable administrations

#### For administrators

- A superior stage: high achievement rate and accessibility of empowering  
transporter grade SLA
- Scalable and future-confirmation stage: adaptable organizations and  
administration advancement
- A helpful stage: convenience, instinctive and shrewd
- Addressing numerous gadgets and secure components: with the most

efficient channel (SMS, HTTPs)HTTPs)

## **3.2 Functional and Non Functional Requirement**

### **3.2.1 Functional Requirement**

Functional requirements for AOTA (since AOTA V6.1).

- Oracle or MySql any one of these database
- Introduction of some new SMS features—Subscription Activation mode, Push Profile, and the Active Retry (since AOTA V6.3).
- SOAP UI or Linqus UI
- Eclipse:

### **3.2.2 Non functional Requirement**

1. Software/System Architecture: The use of programming engineering will be made to acknowledged in a manner that empowers the client to do their support occupations, similar to investigating to settle mistake and so forth in an exceptionally proficient manner.
2. Virtual Environment: Moving the virtual machine on the fly to a very surprising equipment. Moving the pre-owned circle space of the virtual machine on the fly to some other information store. Programmed circulation of virtual machines are absolutely on an alternate hosts for the heap streamlining. The accompanying requests must not be utilized for applications.
3. Service Monitoring:
  - a. Performing the Measurement Interval

- b. PM the Data Alarming
- c. Specifying the Counter for KPIMeasuring
- d. Explanation of the PM Data & Counters

### 3.3 Design Diagram

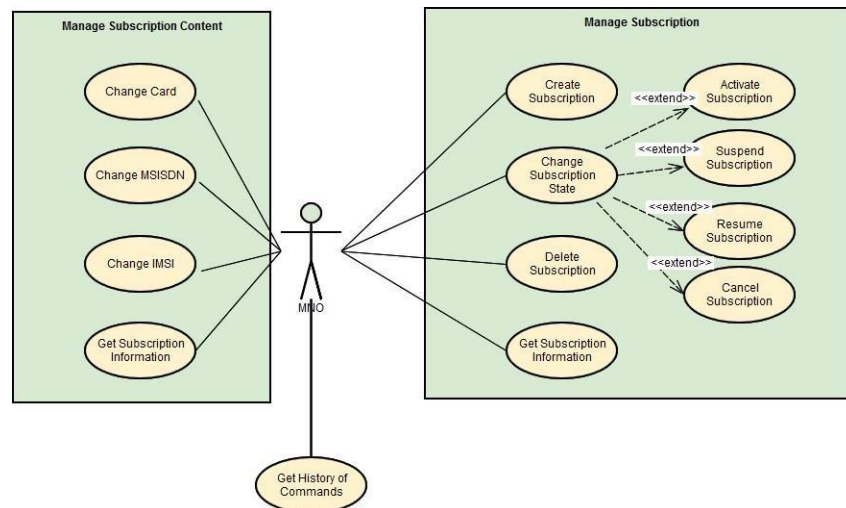


Fig 3.3.1 Subsrcption Life Cycle

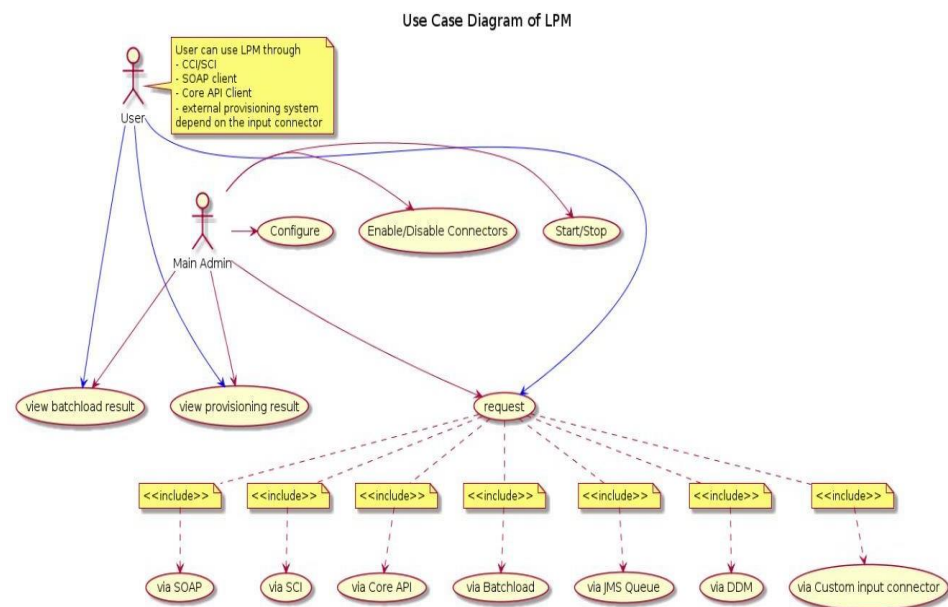


Fig 3.3.2 LPM Use Case

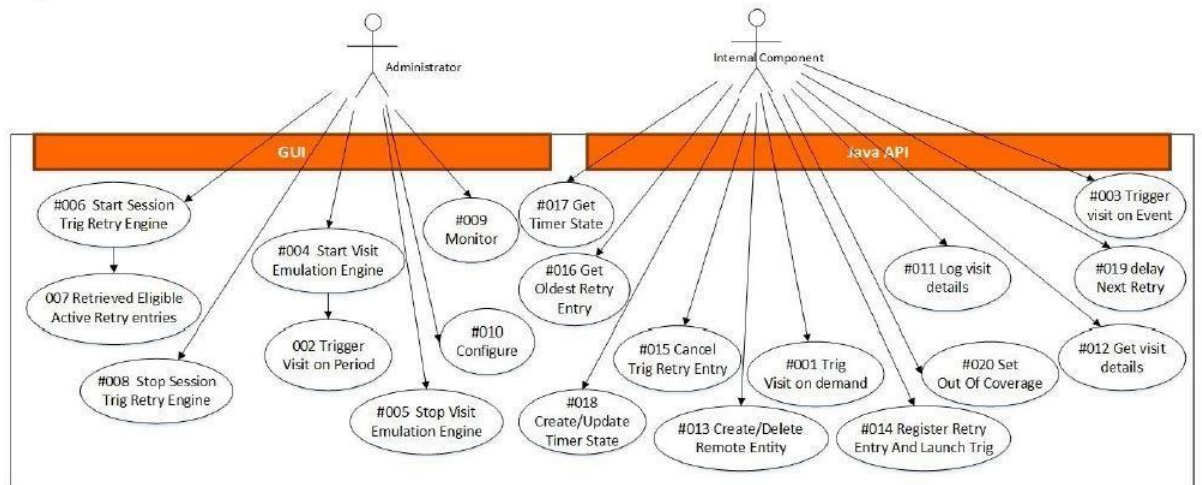


Fig 3.3.3 STH Use Case

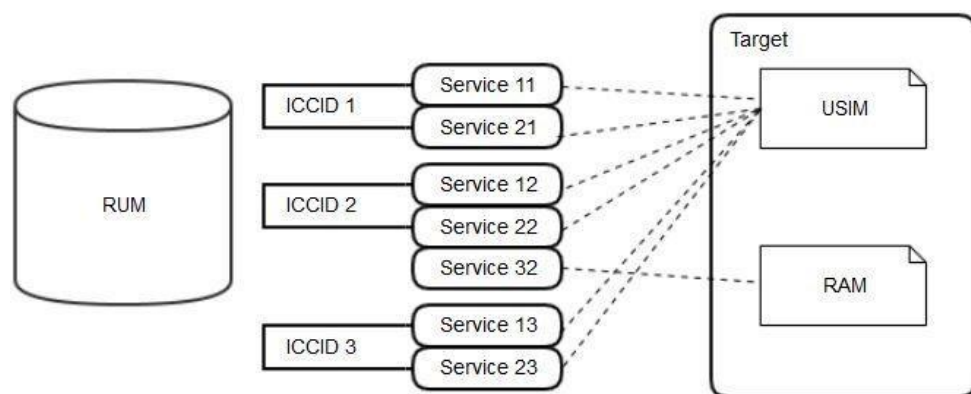


Fig 3.3.4 RUM Triggering service group to same ICCID

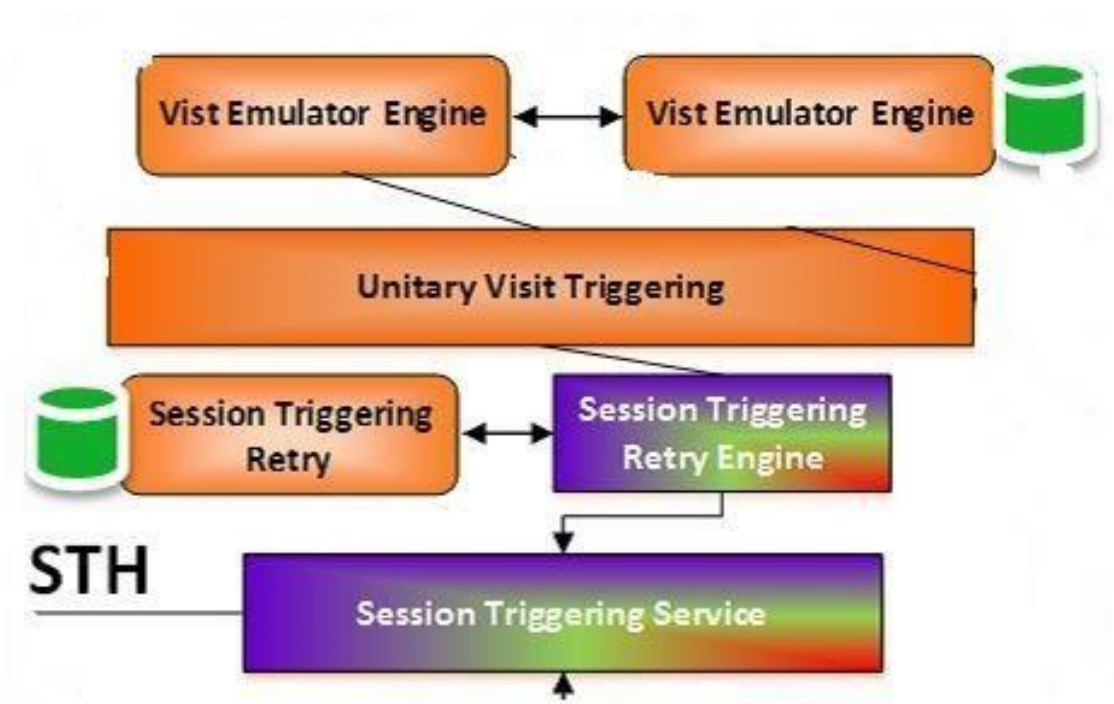


Fig 3.3.5 STH detailed module

### 3.4 Implementation Detail and Issue

#### 3.4.1 Implementation Detail

Advanced OTA mainly ensures that the secure remote management of the 4G and 5G SIM and eSIM file and its application lifecycle. The first thing that needs to be done when Sim is sold is

**3.4.1.1 Provisioning:** It is to Add Important information to the card and store its unique characteristic like ICCID, MSISDN And IMSI. After Provisioning next step is card

Activation. It can be Done With the help of RUM

#### 3.4.1.1.2 RUM-Provisioning:

##### Usecase overview:

The provisioning can be performed after the MNO purchased cards. The subscription as well as card instances will be created for each card. All these information will be required by the **Activation use case** which will occur once the card is connected to the MNO network for the very first time

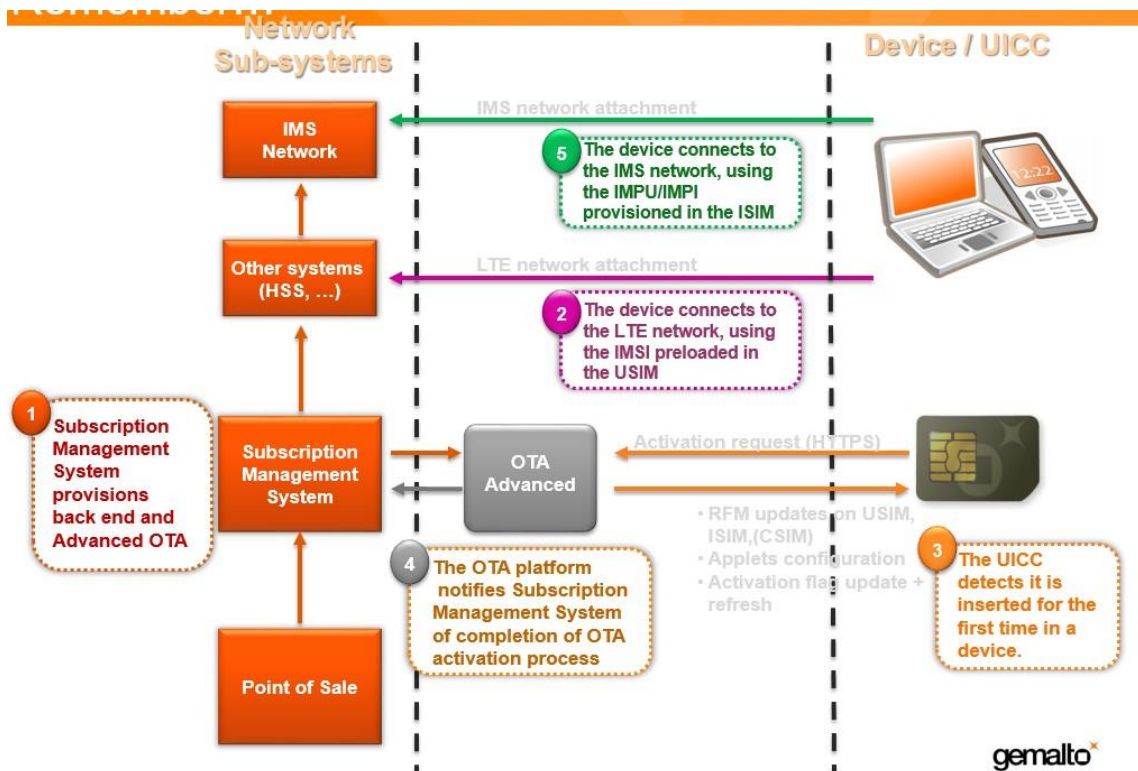


Fig 3.4.1.1.2



## Subscription Management:



Users are able to

- a. Manage Subscriptions
  - Create a Subscription
  - Resume Subscription
  - Suspend Subscription
  - Delete a Subscription
  
- b. Manage Subscription Content
  - Assign MSISDN
  - Invoke Ota Services
  - Change Card
  - Change IMSI
  - Change MSISDN

## Provisioning Flow

The main class is `com.gemalto.rum.api.InternalSubscriptionManagerBean` under RUM-Core project.

There are 2 ways to perform creation of subscription. They are:

- Batch - This is done via LPM. LPM would invoke the different components via connector modules to perform the provisioning.
- Unitary - This is done via RUM. A SubscriptionManager webservice API is used to create the subscription.

## Create Subscription

Steps:

1. Create Card Security  
RUM would invoke SEKMS to provision the card security. The key info can be provisioned to MB or KMS depends on the API called.
  - a. RCA MB - The card security data is provisioned to MB for createSubscription API.
  - b. KMS - The card security data is provisioned to KMS for createSubscriptionWithGSEPMLSecurity API.

GSESPML will be provisioning first. GSESPML data if exist will be created first to avoid the case when subscription is created but SEKMS cannot process the GESPML file

2. Create Subscription record in SRM  
The actual subscription data. During provisioning, it is possible to have ICCID only, and when activated, it should contain the IMSI and MSISDN.  
If there is an existing subscription in SRM, it will be clear first

3. Create Card Security - RCA MB

If there is no GSESPML data , the RCA security data will be created by calling SEKMS API.

4. Create card instance in CRM

The card instance will be created in CRM by calling CRM API.

Before version AOTA 6.1, RUM will directly create card profiles in RCA\_CARD\_PROFILE, but it's obsolete now.

5. Create visit log record in STH

A visit log record which contains ICCID of the subscription and AID of the ISD (Issued Security Domain) will be created through STH API by RUM. This record will be managed by STH for polling process such as **last polling date** information.

When subscription is in "CREATED" state, this record will not be activated. After subscription is in "ACTIVATED" state, RUM will activate the STH record. During the campaign, this data will be used by STH.

6. Create the history data.

The history data creation is optional. It depends on the value of *invokers.without.history* parameter in OAPS. If this parameter value is "true" then the history data will be created.

In all cases the history data is created in SRM. In addition to this, it is depending on the OAPS configuration value for, *useRHSForSubscriptionCommands* parameter, RUM will create the history record in RHS as well

## **Delete Subscription**

Records/Info from the following components are removed when this operation is performed

1. Delete provisioned records in SRM, CRM, SEKMS (MB or KMS)
2. Delete ota services and polling history from RUM
3. Delete infra history records for OTAIP
4. Delete records that correspond to the card instance from MMGT
5. Delete records from STR/STH
6. Notify LUOTAC to cancel any invocations that corresponds to this target
7. Create RHS history record and SRM history record to indicate subscription deletion if invoker is not in the list specified in OAPS parameter "invokers.without.history".

However, if invoker is in the list specified in OAPS parameter "otaa.specific.invokers", delete the RHS history record and SRM history record.

## **Resume Subscription**

1. Validate the subscription data
2. Resume the subscription in SRM
3. Deprecated : Update state of the RCA\_SMART\_CARD object.
4. Update the STH record.
5. Logging RHS and Audit Trails.

## **Suspend Subscription**

1. Check the Input Parameter coherence

## MSISDN

If it is provided then RUM will retrieve the subscription data from SRM based on MSISDN.

The ICCID and IMSI inputs, if they are provided, should be same as the ICCID and IMSI in the corresponding subscription

## ICCID

If the MSISDN is not provided as input and ICCID is not null then RUM will retrieve the subscription data from SRM based on ICCID  
The IMSI input if it is provided should be same as the IMSI in the corresponding subscription

## IMSI

If both MSISDN and IMSI are not provided then IMSI is a mandatory input parameter.

RUM will use the IMSI to retrieve subscription data from SRM

### **InconsistentSubscriptionInfoException**

Any failure when checking the parameter input will cause InconsistentSubscriptionInfoException

2. Check if current subscription's state is applicable for suspension

### **Suspension Eligibility**

Only CREATED, ACTIVATED and SUSPENDED subscription is applicable for suspension, otherwise RUM will throw IllegalSubscriptionStateException exception

3. Suspend the subscription in SRM
4. Deprecated : Suspend the RCA\_SMART\_CARD object.
5. Update the STH record. If visit log record does not exist, it would be created.
6. Logging RHS and Audit Trails.

### 3.4.1.2 Subscription Lifecycle

The Subscription Lifecycle is described by below diagram

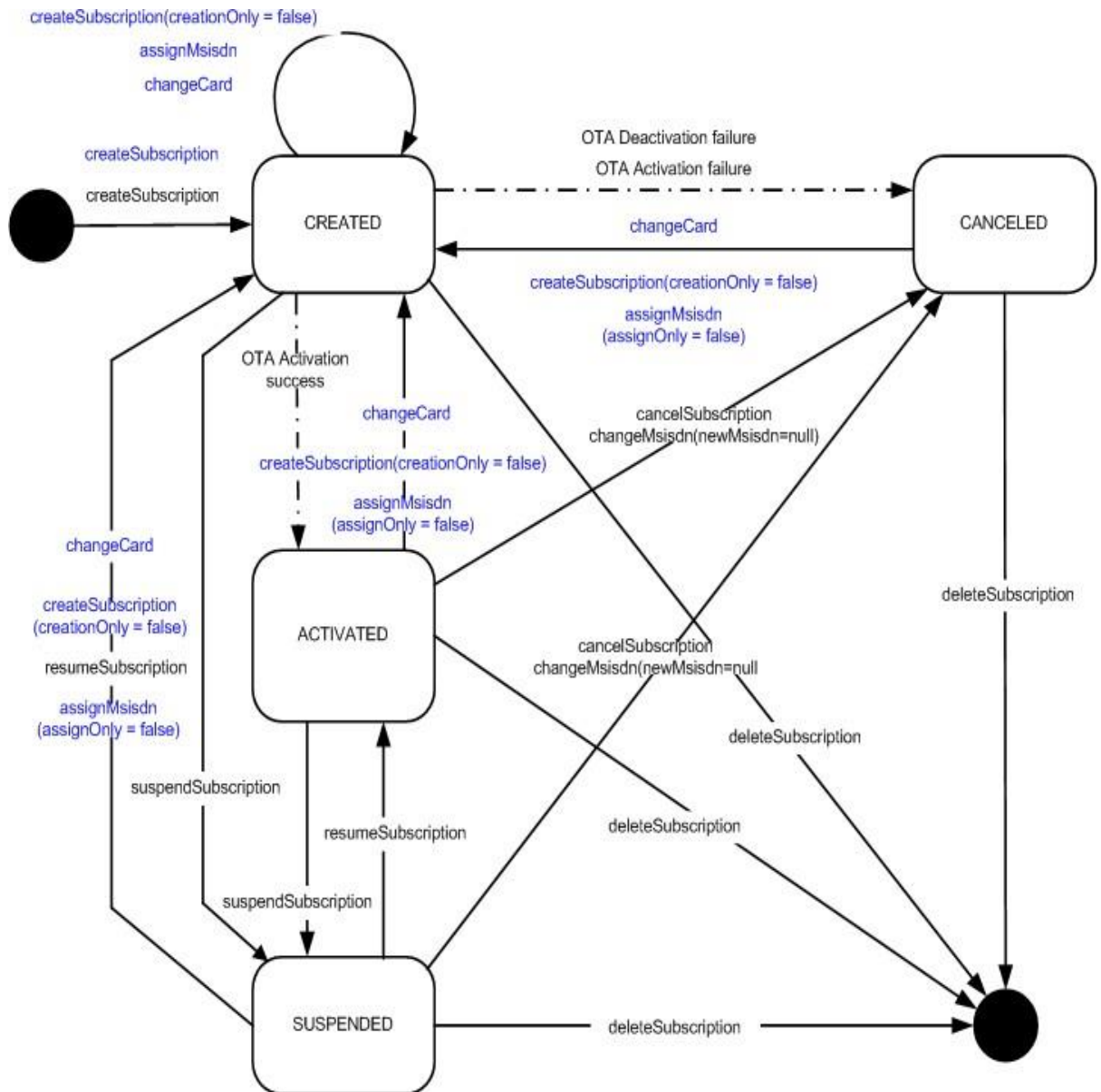


Fig 3.4.1.2

Different states of subscription can be seen as below :

#### Created

The subscription is created. Even if the subscription is not yet active, the msisdn can already be allocated to the subscription at this stage. At this stage, a subscription can be ACTIVATED, CANCELLED or SUSPENDED. The subscription may be deleted at any time.

#### Activated

The OTA activation process has been successfully performed; the subscription is fully operational. At this state, a subscription can be CANCELLED or SUSPENDED. During the Change Card Use case, the ACTIVATED subscription should be passing through CREATED state before a new card assigned.

#### Suspended

The Subscription is suspended. The subscription may be resumed or deleted at any time.

This state may be set because of loss, robbery, no payment...

#### Canceled

In this state, the OTA activation process has failed, due to an “external” problem that the Advanced OTA server cannot resolve by itself  
(For example, bad OTA keys value)

### **Manage Subscription Contents**

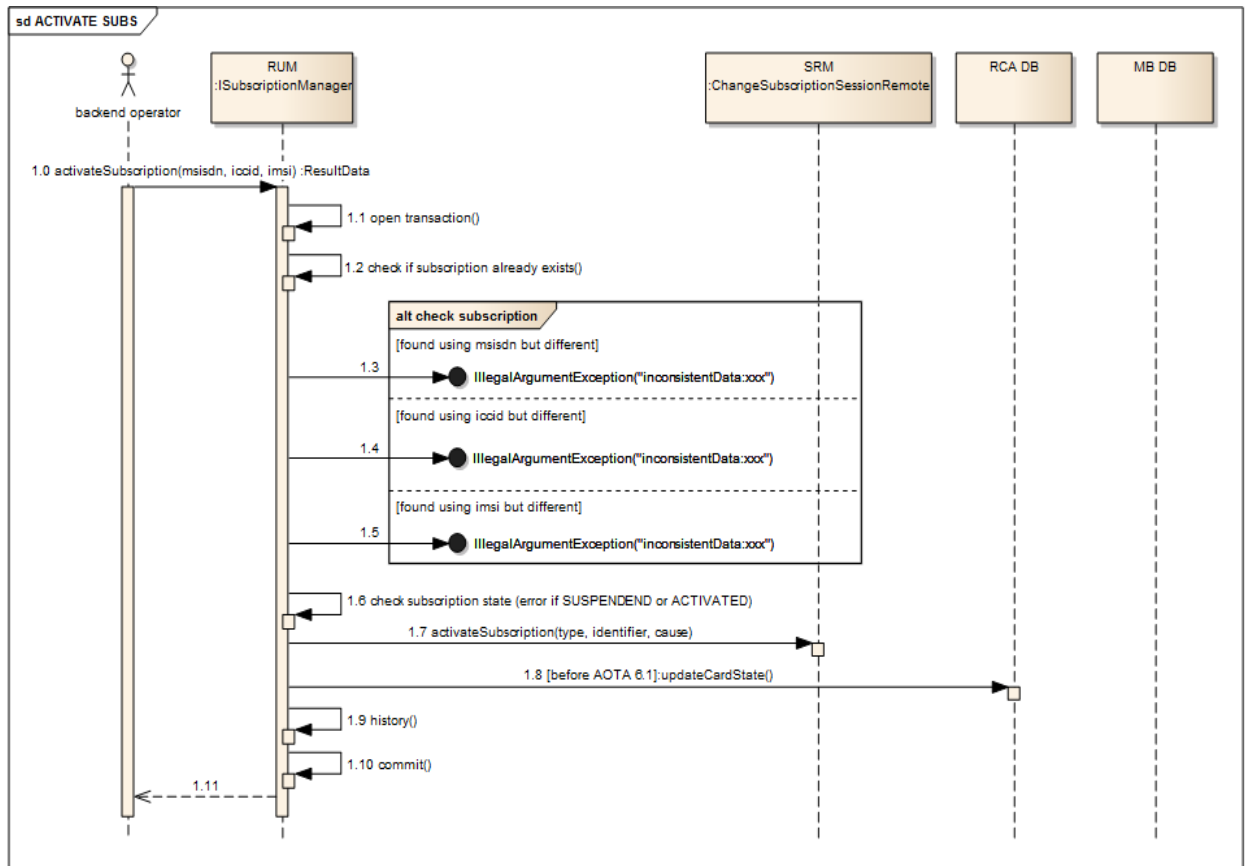


Fig 3.4.1.2(b)

#### Note:

Before step 1.9 (create SRM and RHS history), the visit log record in STH is created if it does not yet exists, and this remote entity is activated.

#### Change Card

##### Note:

For step 1.10 (deleteDevice for old ICCID), a new device record is added for the new ICCID that is associated with the subscription created.



For step 1.18 (delete OTA services), terminate use case to MMGT for the old ICCID is also done.

LUOTAC may also be notified to cancel the invocations submitted to OTA for the old ICCID.

The visit log record for the old ICCID in STH is deleted and a visit log record for the new ICCID is created.

## Change IMSI

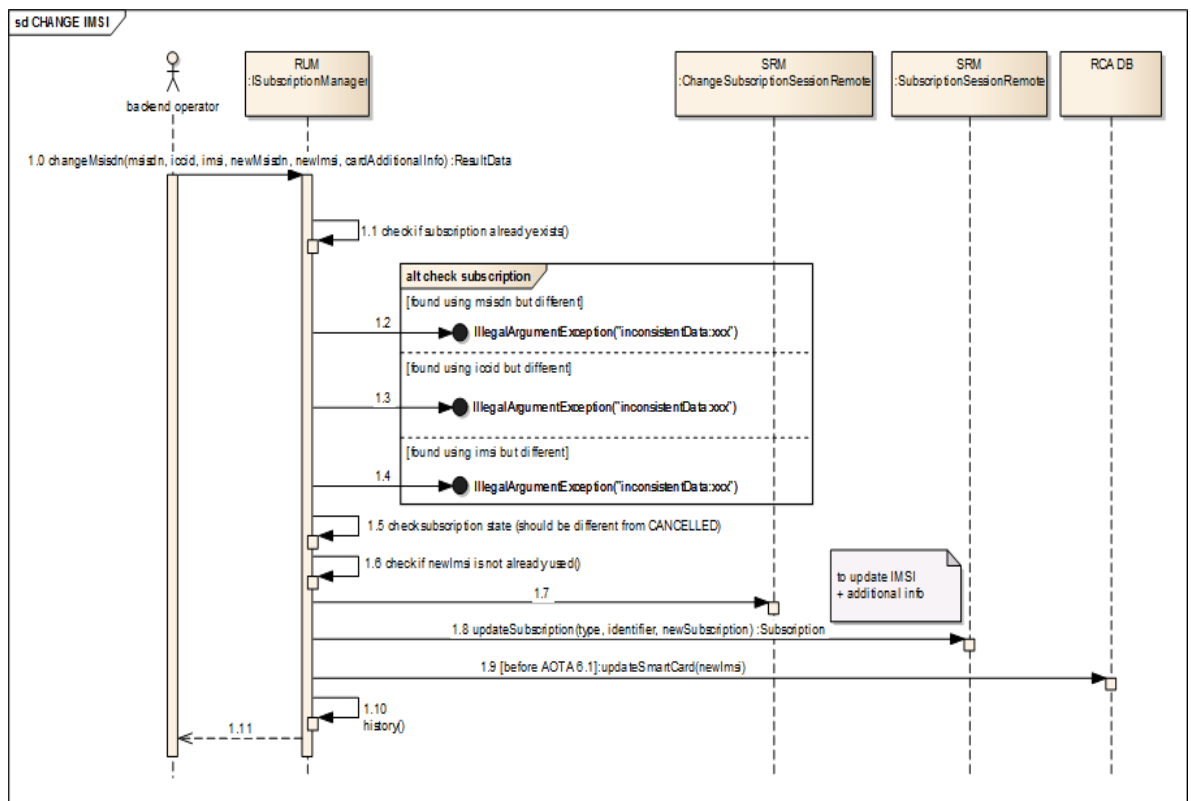


Fig 3.4.1.2(c)

## Change MSISDN

### Note:

Before step 1.9 (changeMsisdn), if new msisdn is not null and subscription for existing

msisdn is activated, terminate use case to MMGT is done in addition to the delete of ota services and polling history from RUM. LUOTAC may also be notified to cancel the invocations submitted to OTA. The visit log record in STH is created if it does not exist, and this remote entity is activated/deactivated depending on the subscription state.

### **Change IMEI**

To handle IMEI update on existing subscription, RUM has added a new API.

#### **API parameters are:**

- Choice between three identifiers, only one is mandatory : <msisdn>, <iccid>, <imsi>
- IMEI tag is mandatory : <imei> value can be empty
- Optional parameters are: <handset> and <invokerName>
- If Handset tag is present, <model> tag is mandatory along with <name> and <manufacturer> tags

By default handset creation is not allowed on AOTA platform. It is restricted by SRM Parameter **NO\_MODEL\_AUTOCREATE**. To enable the handset creation, please use below mentioned command:

### **3.4.2 RUM-ACTIVATION**

During the Activation use-case AOTA platform executes several mandatory operations on the card to make it become ACTIVATED Card in the system. These operations will be

executed in a sequential fashion. If an error is encountered at any step, activation will halt. Activation use-case is available only via HTTP transport.

The steps:

1. Get device information from the card and update them in the SRM tables.
2. A user can attach any other RUM services to be executed at this time, if such services are attached they will also get executed at this point.
3. Send the DDE acknowledgement to the card so that it doesn't trigger Activation use-case to the server anymore.
4. Set the next polling date on the card.
5. The Activation use-case is now completed.

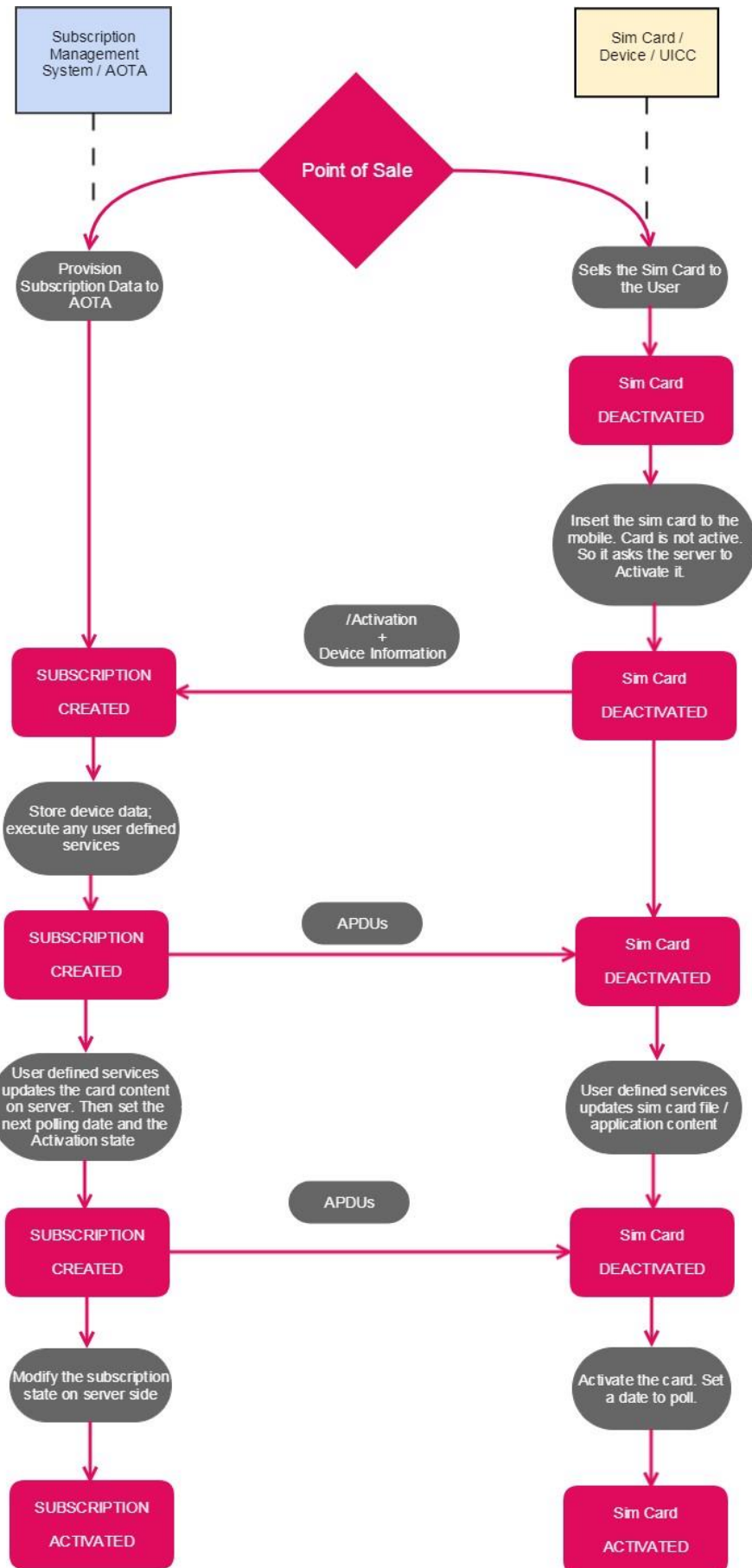


Fig 3.4.2

Visit As you can see from the above diagram, Activation flow has two major steps:

1. Provisioning Flow.

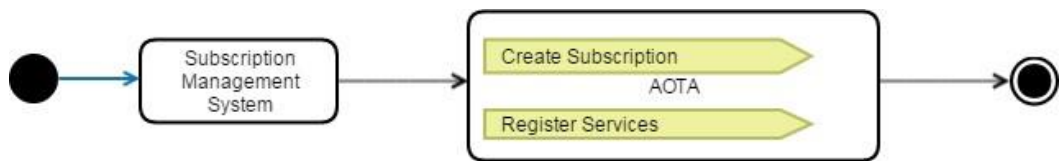
MNO creates the subscription data in AOTA server.

2. Flow

End user switches on the device for the first time, and card sends a request to the server, demanding to be activated

### Provisioning Flow

Provisioning step includes creating the subscription information in AOTA.



Provisioning Flow consists of 2 steps :

- Create the subscription if it does not exist.
- Register the use-case of services for the subscription activation. These services will be executed during the activation time on the card.

### Create Subscription

Here is the sequence diagram.



## Register OTA Services

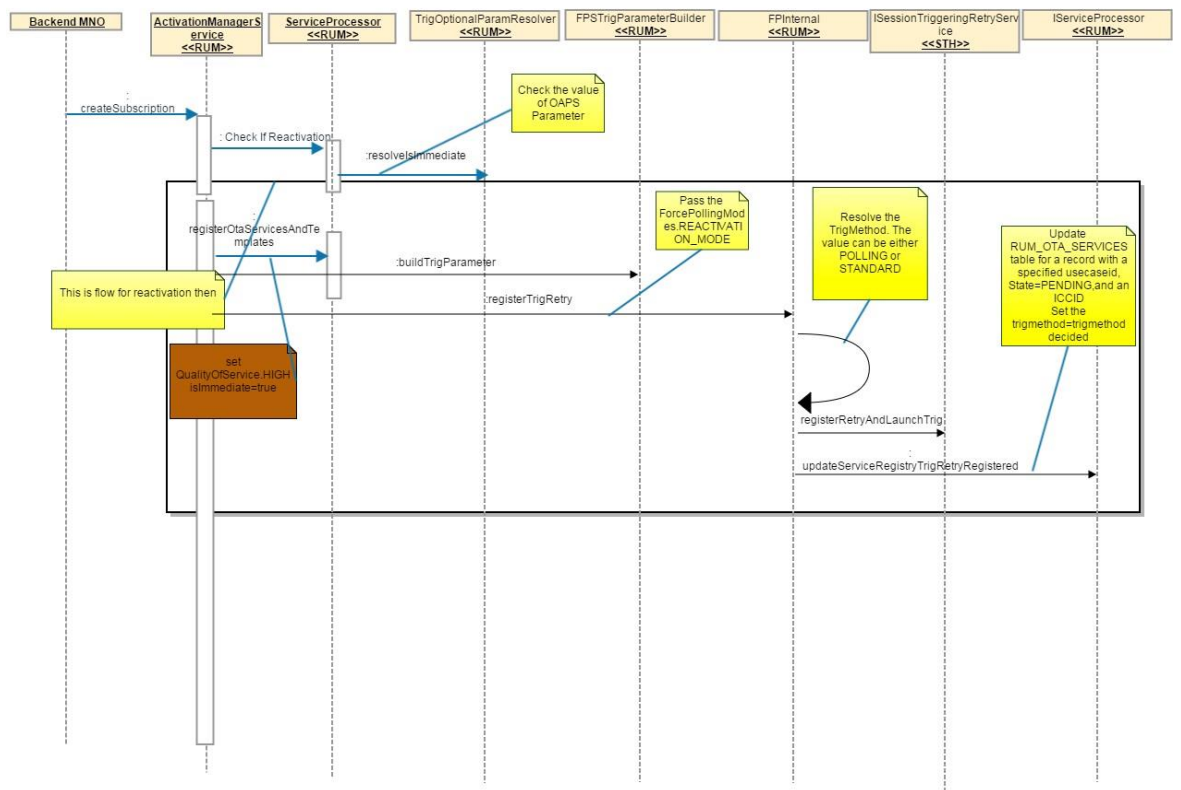


Fig 3.4.2(c)

For CreateSubscription API, services can be registered for execution during /Activation request from UICC. There are two types of service groups that can be specified as parameters for the CreateSubscription API. They are **otaServiceGroupList** and **otaPollingServiceGroupList**.

Services indicated in **otaServiceGroupList** are executed during /Activation from UICC. Services indicated in **otaPollingServiceGroupList** are executed during the first /Polling from UICC, after the activation is completed. These services are registered as high priority so that they will be executed first.

Please refer [RUM - Provisioning](#) for complete provisioning guide.

## **Visit Flow**

When the end user inserts his fresh SIM card and switches on the handset, card sends a request for Activation. This is the trigger for the visit flow. The following steps during the visit:

1. During the device start-up, card detects that it is not ACTIVATED. This ACTIVATED state is managed by the DDE applet.
2. The DDE applet in the card calls the server with using /Activation URI.
3. When the request is received by the server, it detects whether a subscription exists in CREATED state. If the subscription is in ACTIVATED state, AOTA will execute either the Deactivation or the Reactivation use-case (depending on OAPS configuration).
4. If the card is in CREATED state, AOTA will attempt to send all activation associated services.
5. Activate the card, send acknowledgement to DDE applet and set the next polling date.

For RFM/RAM updates, if certain steps in the workflow fails, the subscription will be set to "CANCELLED" because for Activation use-case it is always STOP\_ON\_ERROR.



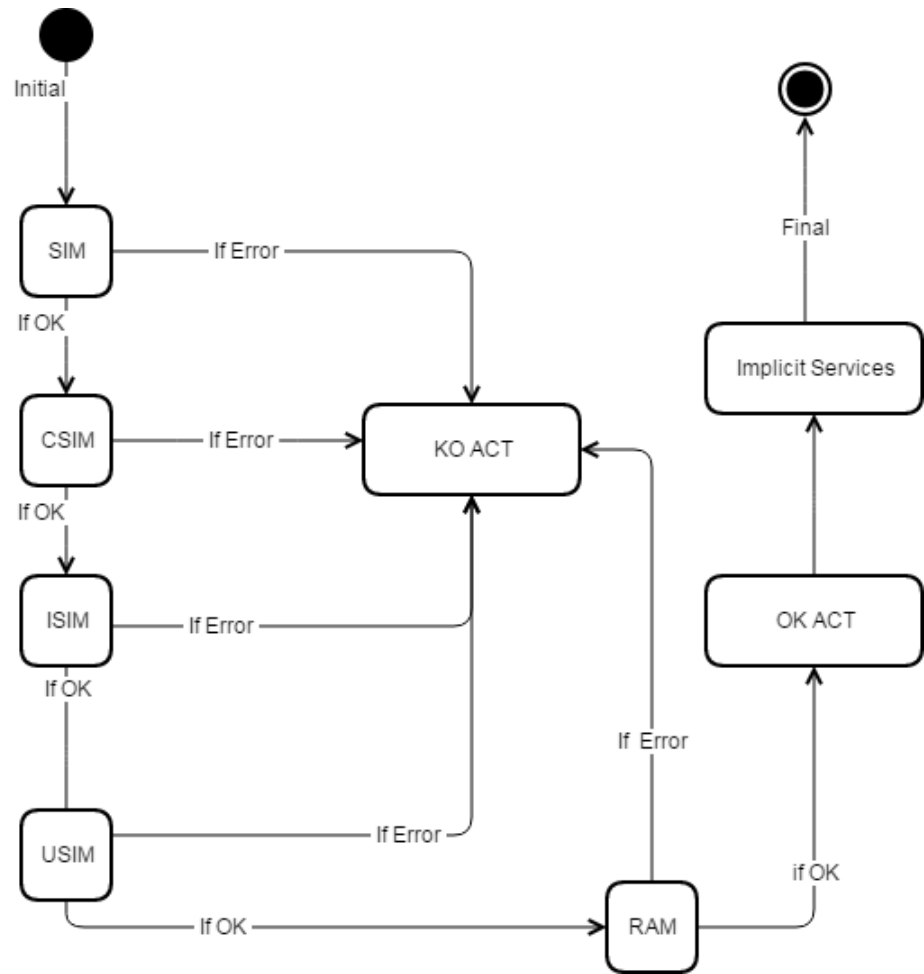


Fig 3.4.2(d)

When the entire activation process which includes the RFM/RAM updates is completed successfully, the subscription state will be updated to "ACTIVATED". There are some implicit services to be executed after successful activation use-case. These services are the commands to:

- call the refresh applet.
- call the update on activation status in DDE Applet.
- call the update on polling date in Polling Applet.

visit flows are discussed in RUM - Workflow section.

## ATT Activation use-case

One of our customers, ATT, has a different definition of activation use-case.

For all other customers, the subscription state is **not yet ACTIVATED until the first polling** by the SIM Card.

For ATT, the subscription state is **already ACTIVATED even before the first polling** by the SIM Card. The subscription is created and subsequently activated by the custom application developed specifically for ATT, i.e. ASAP

ATT is very important step in card proviosining and should be done very carefully and an unnotices mistake in the execution during the time of ATT would let to the failure of proviosning so ATT is really important step and must be done with full focus and should be activated as soon as possible.

### **3.4.3 RUM - Deactivation**

#### **Introduction**

This is a sub use-case of the Activation. When a card is not activated in AOTA platform, but it is really activated in the network, AOTA uses this use case to deactivate the card in the network.

#### **Deactivation Flow**

Following conditions should be satisfied for deactivation scenario to be triggered:

1. There is a subscription is in CREATED state in AOTA.
2. The same card reaches AOTA (RUM component) with a /Polling request.

In Deactivation flow the following steps will be executed:

1. Deactivate the card.
2. Deactivate the polling applet.

After these two steps are executed successfully, the real SIM card is at DEACTIVATED state and the subscription in AOTA is in CREATED state. So the next step of the card is to call the server with /Activation request. This call will trigger the Activation work flow to start.

# DeactivationUseCase

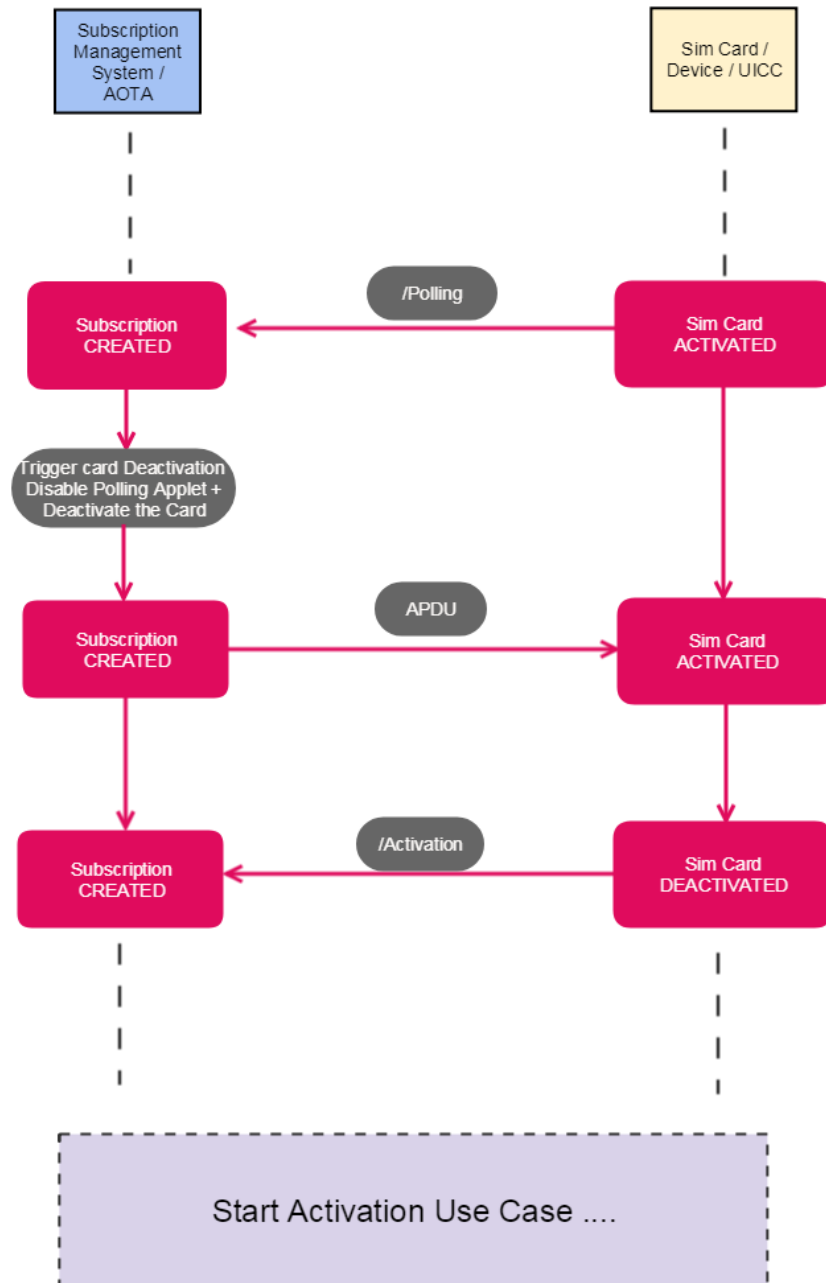


Fig 3.4.3

### 3.4.4 RUM - Reactivation

#### Introduction

This is a sub use-case of the Activation. When a card is not activated in AOTA platform, but it is really activated in the network, AOTA uses this use-case to re-activate the card. This feature is used in Verizon. .

#### Reactivation Flow

Following conditions should be satisfied for deactivation scenario to be triggered:

1. There is a subscription in CREATED state in AOTA
2. Subscription should have some pending Activation bounded services; only non-implicit services are considered.
3. The reactivation feature should be enabled from OAPS (reactivation.mode.enabled).

The reactivation flow executes following steps.

1. Terminates the polling flow immediately
2. Asks the card to comeback with reactivation URI
3. Execute the remaining activation services
4. Reactivation is not going through the DDM. So RUM shall explicitly registers a readIMEI services to get device information.

Please find the flow diagram below.

## ReactivationUseCase

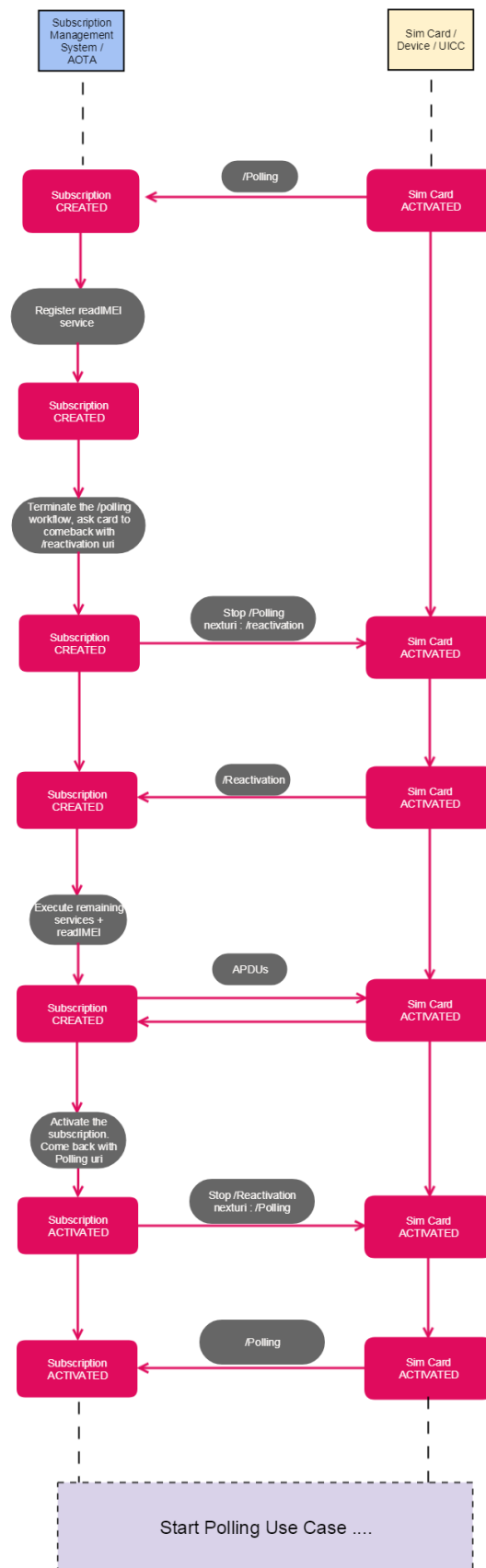


Fig 3.4.4

## Class Sequence Diagram

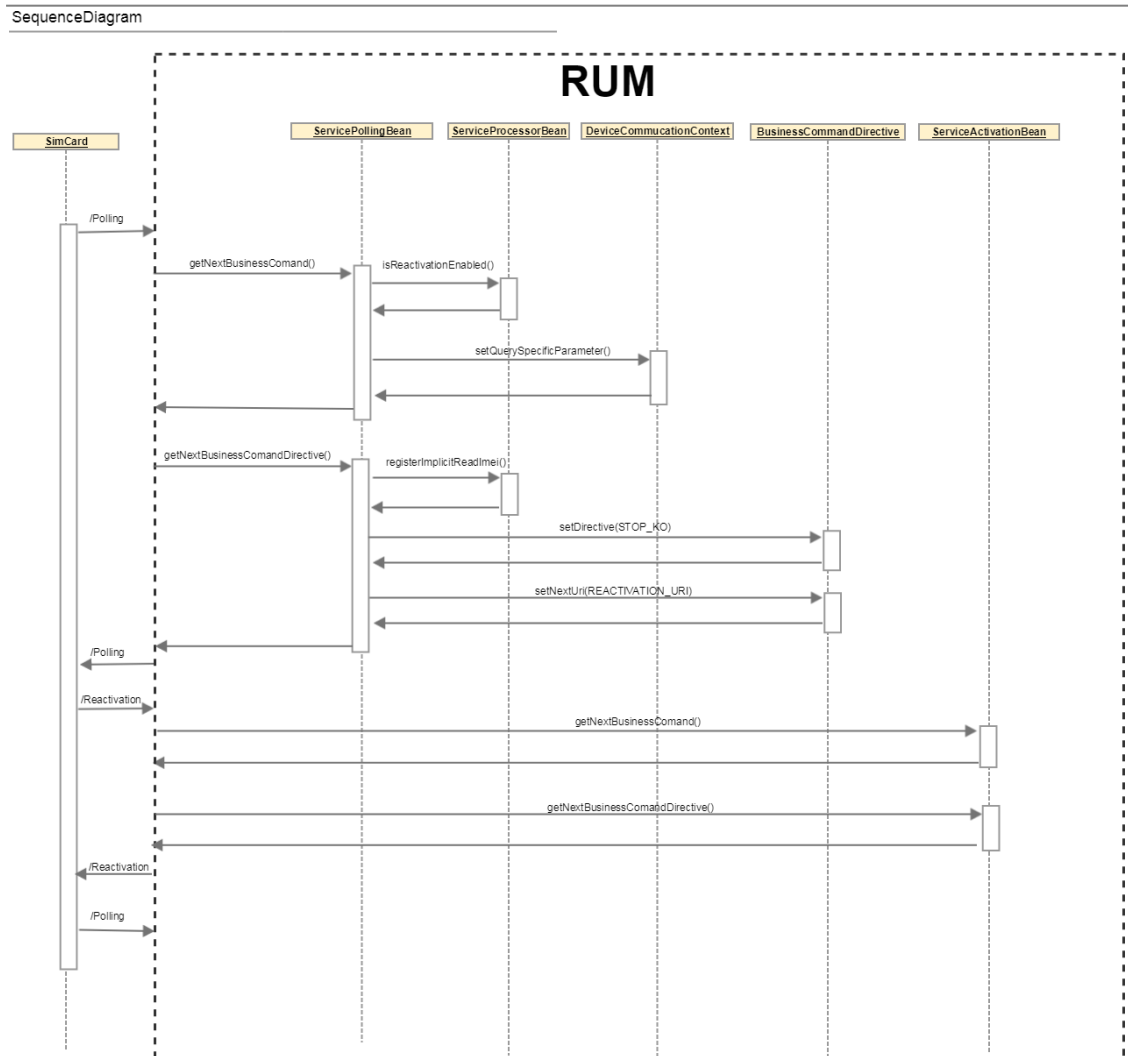


Fig 3.4.4(b)

### 3.4.5 Polling

Polling is updation(execution) of pending services for a card.

## 1) SessionType(RUM)-

TYPES OF POLLING- There are two types of polling:

### a)Real polling

- When polling is initiated by card.

- pull mode.

- HTTP

(forced polling-forcing card to do a real polling)

Using <isImmediate> Tag to trigger card to initiate forced polling.(target polling applet or admin agent)

### b)Emulated polling

When polling is initiated by AOTA(RUM,STH)

## 2) SessionSource(SEE)

TYPES OF POLLING-

### a)on\_demand

RUM or any other component for this service execution.

### b)onevent

NPD(Network Presence Detection)



When the card is out of coverage but the services are activated(pending) . So when the card is in network

coverage ,getNotification() ---Polling.

c)onperiod

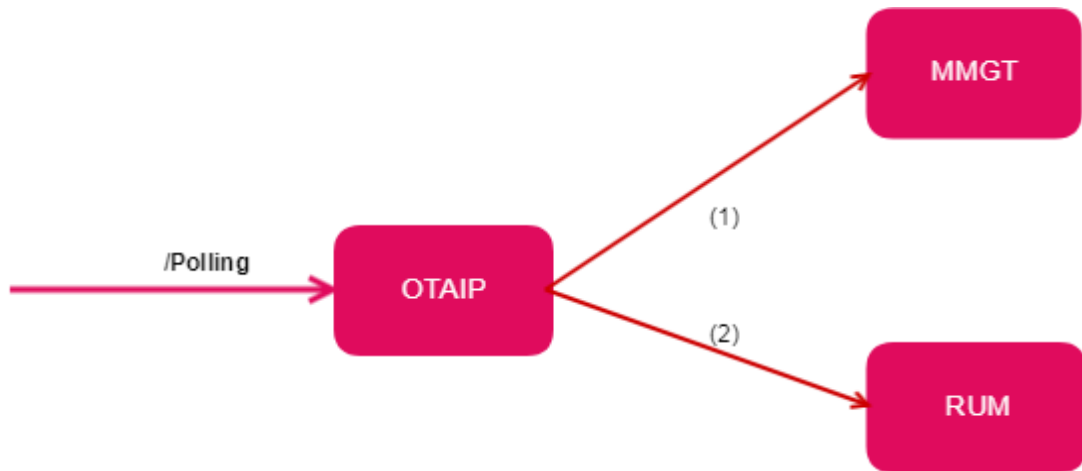
STH -table-STH\_Remote\_Entities\_Registry-It contains instance of cards.(We periodically loop this table to ensure that all cards MUST poll after some time, if they haven't we initiate on\_period polling).

STH(is Session triggering Helper. provides periodic polling, polling retry) and SEE is stimulator of card.

Since RUM is passive,SEE and STH have APIs to initiate polling.Then RUM sees it as polling.

Now we have done both push+pull.

### 3.4.6 RUM-MMGT Integration



**Fig 3.4.6**

#### **Introduction**

MMGT is the campaign manager in AOTA. Purpose of MMGT is a facility to execute services targeting multiple subscriptions. In OTAIP configurations, a Polling request is chained in such a way that each Polling request first triggers the MMGT module first, and then RUM. This chaining has been done as part of the MMGT - RUM integration. Both MMGT and RUM modules are working together to execute campaign use cases.

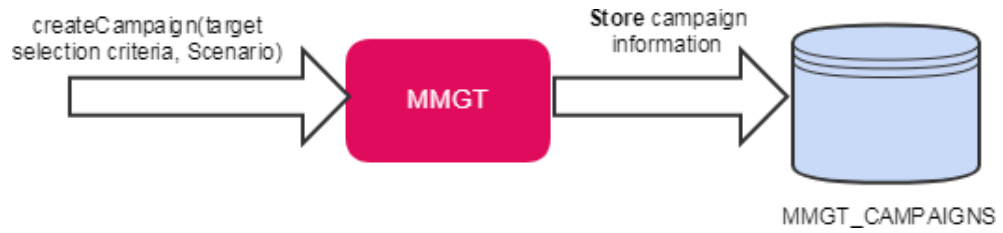
**From the card perspective**, it only visits one URI: /Polling.

**Internally**, OTAIP calls MMGT's EJB first, and then calls RUM's EJB subsequently.

#### **AOTA Campaign Flow**

Following diagrams show the different steps that AOTA goes through to execute a campaigns.

#### **Step 1 - User Create a Campaign**



## Step 2 - Card Polls

Because of the Polling chaining, OTAIP first triggers the MMGT module.

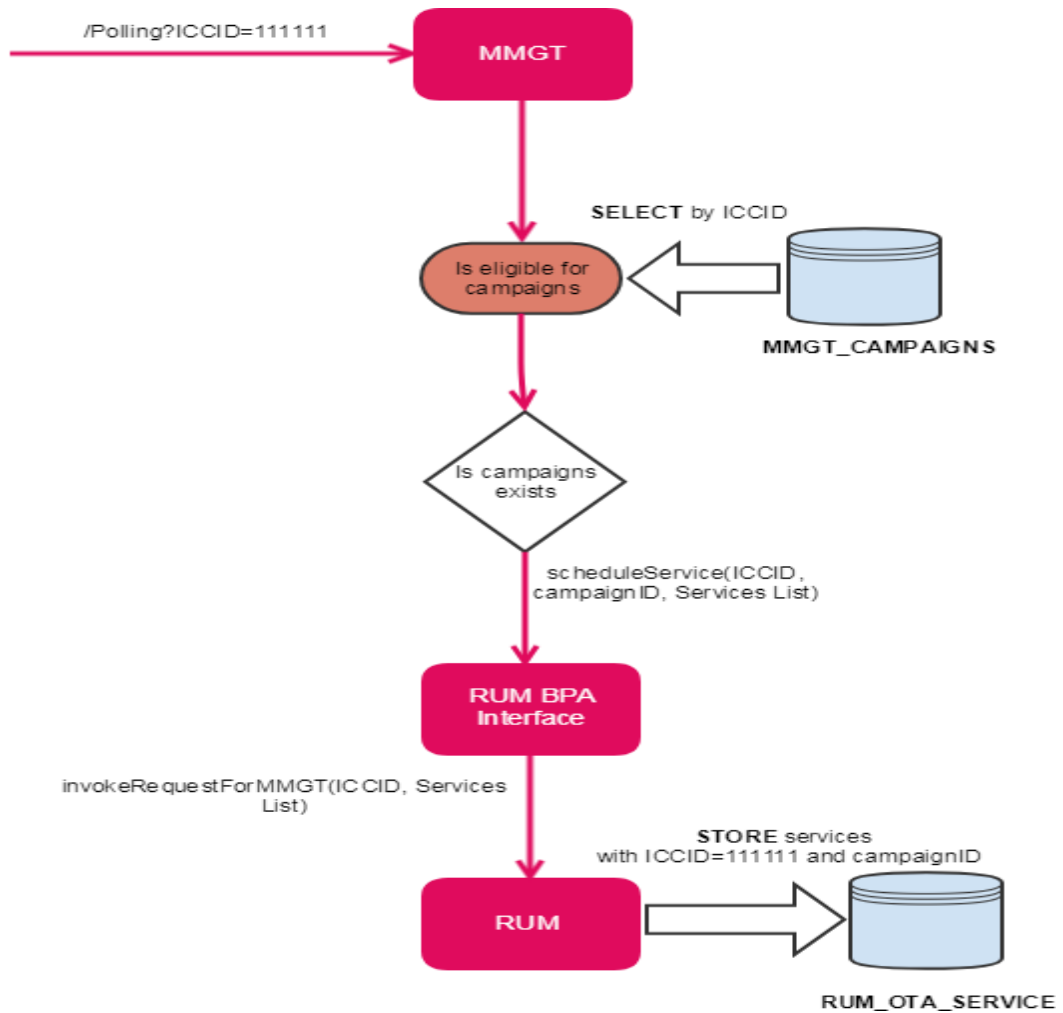


Fig 3.4.6(b)

### Step 3 - OTAIP Triggers RUM

After the step above is finished, MMGT will return to OTAIP. Then OTAIP will trigger RUM.

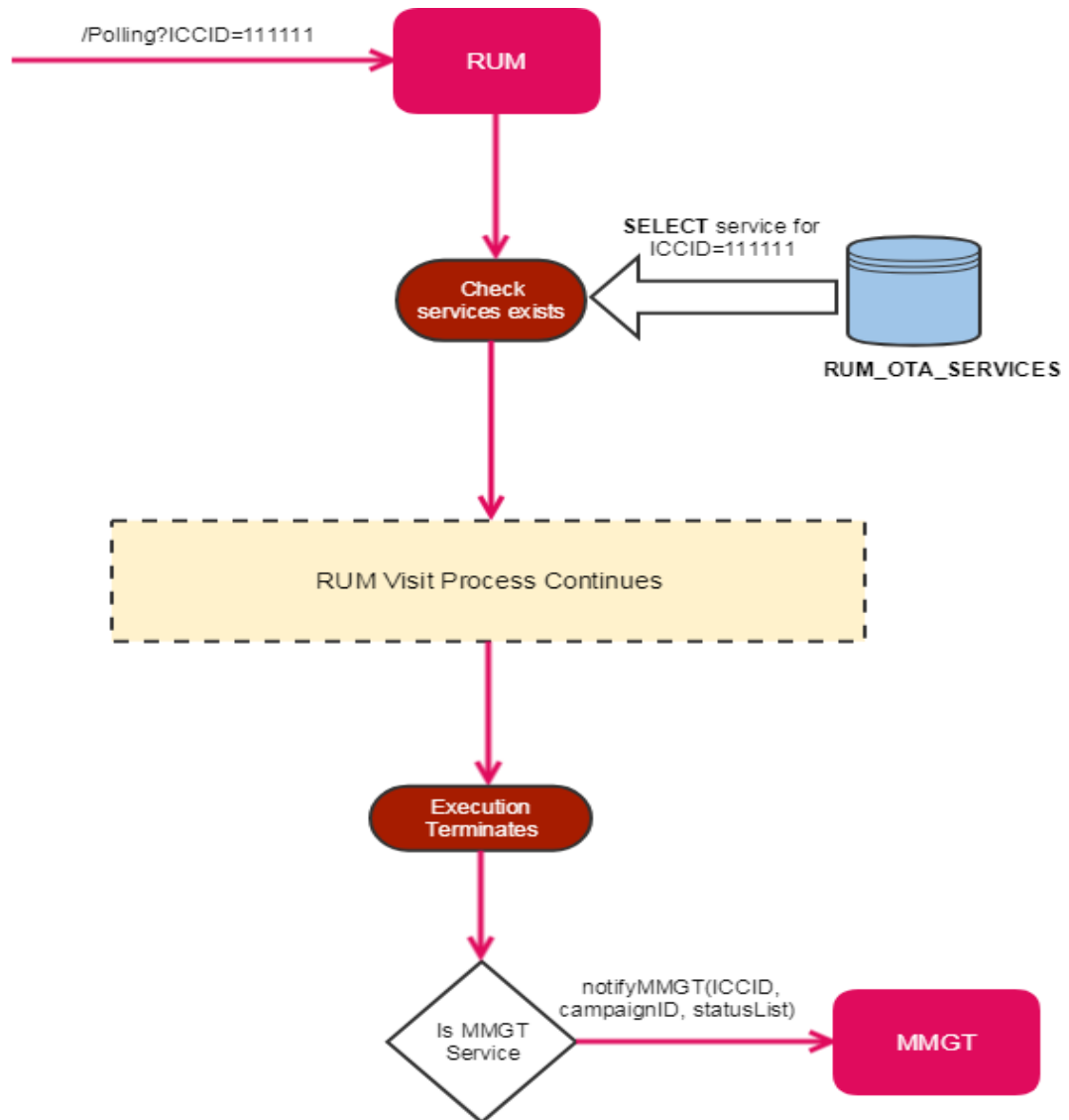


Fig 3.4.6(c)

After the services are getting selected from RUM\_OTA\_SERVICES, RUM considers this a ordinary service. RUM will continue the service execution with the regular flow, but at the end of the service execution RUM notifies the execution result to MMGT.

### **3.4.2 Implementation Issue**

The various issue faced in implementation were related to the RUM services and LPM service connectors that were needed to be added:

- Logging was to be enabled
- Proper entries needed to done for the entry inside jndi tree.
- The apdu sent needed to be corrected.
- The test cases which was developed to check the restriction,need to use try catch to show the exception
- DB entries should be handled via scripts , not depending on the GUI provided by the sqldeveloper
- Proper care should be taken while making session beans and not two mappings should be made to two classes.
- Xsd were not correctly updated while creating new service

## Chapter 4

### Performance Analysis

#### 4.1 Agile Methodology

Agile Methodology has its advantage over waterfall model. In case of waterfall model model system is first needed to be completely designed then it is deployed for testing whereas in agile we can make a small chunk of product commit it and side by side test and validate it. It is quite easy to make changes in the code and the product deployment speed is also fast. We, here at Aota, follow agile model approach for building our product

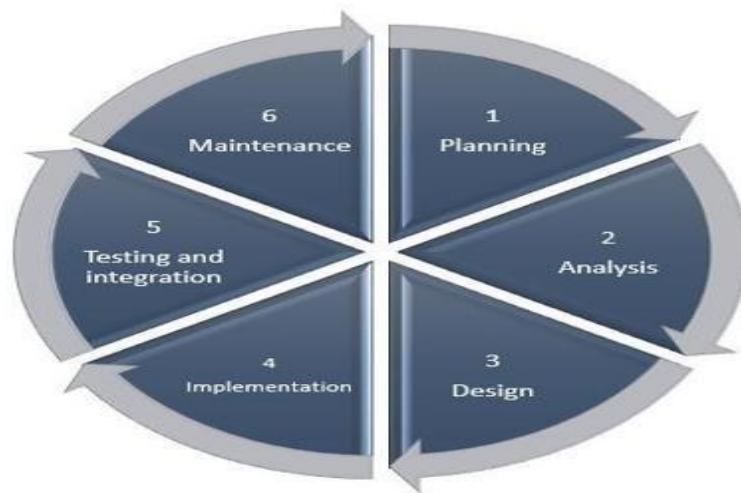


Fig 4.1

## 4.2 Performance Analysis Detail

### Mobile Review

After reproducing the time that was seen on 1.7.1 iOS and comparing it to Duo it was clear that we were seeing a push approval that on average could take 10+ seconds in the best circumstances and 1.5 minutes in some of the worst.

However, this issue largely seems to stem from the usage of an old version of Ezio. In iOS 1.8.0 (in test flight) Ezio has been upgraded to 4.9, and the time it requires to approve a push has dropped to consistently be between 4-6 seconds end to end. See the chart below for a comparison of average time to approve a push between 1.7.1 and 1.8.0 on iOS(Fig 4.2)

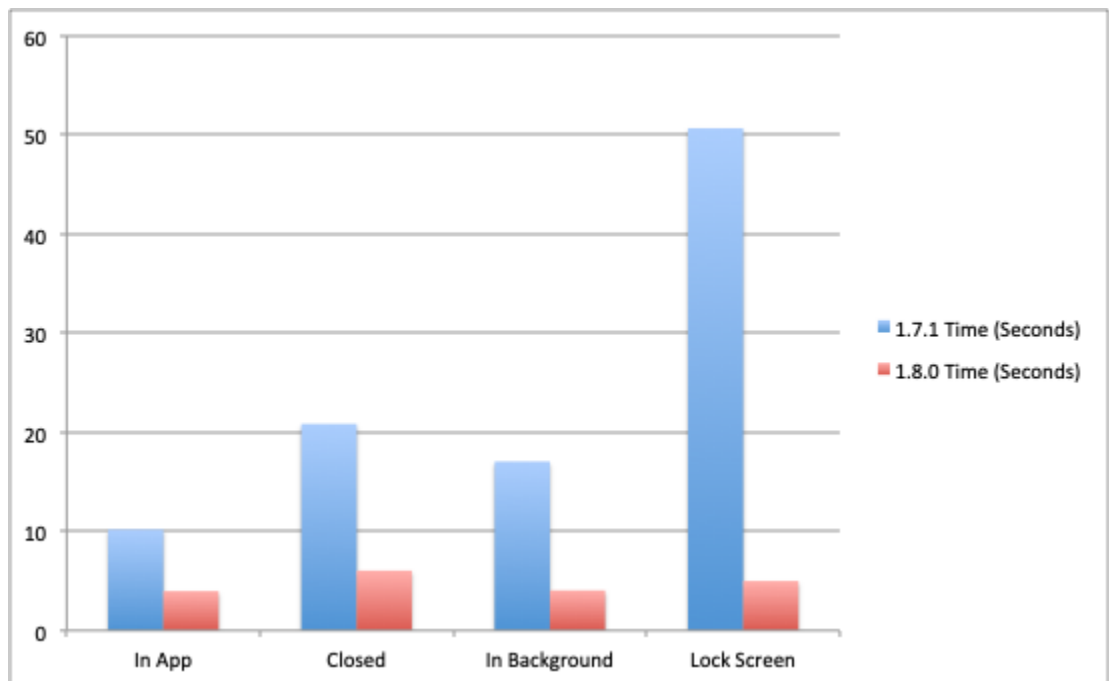


Fig 4.2

## Overview of calculated averages

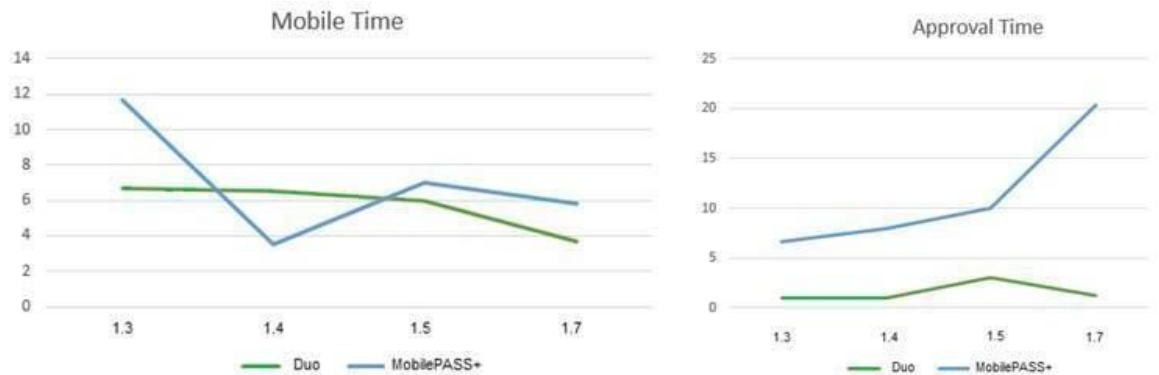


Fig 4.2(b)

On Android we never managed to reproduce these incredibly long push approvals, but Android has been on 4.X of Ezio for at least the past year. Because of this, most of the speed improvements we could hope to achieve are going to be much smaller as the largest issue has already been resolved in the upcoming release of iOS.

For the lengths of time we can still address on mobile, the only step that takes substantial time for the app to handle is the OTP generation time. A locked Moto E which receives a push can take approximately 1.5 seconds to generate an OTP, while a fast device like the Samsung J8 takes a half of that, 0.7 seconds.

This time is reduced if the app is open when approving the push, the Samsung J8 can generate an OTP in approximately 0.1 seconds in the foreground, and the Moto E still takes 0.8 seconds. A break down of all this can be found on the mobile performance page linked above



## Server Review

On the server side there are a variety of steps involves with various services talking to each other, below is the table taken directly from the server report

## Analytics

1. We are lacking tools to help monitor performance of push. The tests we are doing right now are very manual, but we could have tools that monitor exactly how long the server istaking to send and approve push requests. To that end we'd like to create a system that automates that for us, and a ticket is available here: SAS - [Authenticate](#) to see issue details
2. We need better logs and information. It's difficult to determine what exactly is slowing the process down when we are lacking that kind of knowledge. On the server side this is simpler, logs just need to be more fleshed out and the work has been created: SAS - [Authenticate](#) to see issue details . On Mobile we lack any of kindanalytics to begin with so we have to system to build on, adding Analytics would help us identify issues like this when they show up in the field.
3. We should have the same person perform the comparison testing in house again, but using 1.8.0 iOS instead and compare the results.

## 4.3 Improvements

### Server improvements

1. On the server side the MSM response takes a long time and also, we can't answer why it takes so long. We should investigate this and see if this is something we can improve. The work has been created [SAS-33823](#) - [Authenticate](#) to see issue details
2. We are currently using the legacy APNS to deliver messages to iOS devices and we are looking at migrating to use the new APNS HTTPS/2 on the server. When this is complete, the speed involved in delivering the push to the device should improve. This won't affect approval time, but it should make the whole process of triggering a push feel a bit faster for iOS devices at least.

### Mobile Improvements

1. On mobile side there is a substantial difference between generating an OTP in the background and the foreground. This could be a cpu restriction for apps in the background that we can't address, but it could also be something that we can improve within our app. It may be worth investigating, the work is here: [SASMOB-1843](#) - [Authenticate](#) to see issue details
2. On iOS side the notification dismisses as soon as the user hits "Accept" if it is using simplified login. This leaves the user with no feedback that something is still occurring on the device so they are left with seconds of waiting and no feedback. This isn't an issue on

Android as we show user feedback until the push is successfully approved. When the notification dismisses, the page is immediately refreshing to the user. There is no ticket for this work but I would propose we look at the notification on iOS and see if this is something that could be improved.

## **Chapter 5**

### **Conclusion**

After completion of this project we have concluded that this website works as per the need and requirement of the client and is user friendly. All the bugs and errors are thoroughly debugged and thoroughly tested properly. All utility services are successfully integrated at one place in the application. This deploys the kind of application that user want in present time.

#### **5.1 FUTURE SCOPE**

This project has a vast scope in the future as sim cards are need to communicate . Also in future Aota is moving towards Cloud and production of esim will also come in pictures. These 2 areas are the main thing we are targeting in next 2 years.

## References

1. [https://en.wikipedia.org/wiki/SIM\\_card](https://en.wikipedia.org/wiki/SIM_card)
2. [https://en.wikipedia.org/wiki/Oracle\\_WebLogic\\_Server](https://en.wikipedia.org/wiki/Oracle_WebLogic_Server)
3. <https://www.javaworld.com/article/2076777/a-beginner-s-guide-to-enterprise-javabeans.html>
4. [https://en.wikipedia.org/wiki/Smart\\_card\\_application\\_protocol\\_data\\_unit](https://en.wikipedia.org/wiki/Smart_card_application_protocol_data_unit)
5. [https://www.etsi.org/deliver/etsi\\_ts/101200\\_101299/101267/08.18.00\\_60/ts\\_101267v081800p.pdf](https://www.etsi.org/deliver/etsi_ts/101200_101299/101267/08.18.00_60/ts_101267v081800p.pdf)
6. <https://www.globalsign.com/en/blog/glossary-of-cryptographic-algorithms/>
7. [http://thesystem.co.th/prod\\_manager/prev\\_order1\\_system.php?code=A0066574](http://thesystem.co.th/prod_manager/prev_order1_system.php?code=A0066574)
8. <https://www.proofpoint.com/us/glossary/encryption>
9. [https://www.tutorialspoint.com/maven/maven\\_overview.htm](https://www.tutorialspoint.com/maven/maven_overview.htm)

10. [https://www.tutorialspoint.com/maven/maven\\_overview.htm](https://www.tutorialspoint.com/maven/maven_overview.htm)

11. Internal Gemalto Documents

12. [https://www.researchgate.net/publication/315458867\\_Analysis\\_Provisioning\\_Process\\_in\\_SIM\\_Card\\_Activation\\_at\\_Telecommunication\\_Operator\\_using\\_BPM\\_and\\_Balance\\_Scorecard](https://www.researchgate.net/publication/315458867_Analysis_Provisioning_Process_in_SIM_Card_Activation_at_Telecommunication_Operator_using_BPM_and_Balance_Scorecard)

# JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

## PLAGIARISM VERIFICATION REPORT

Date: 18-07-

2020

Type of

PhD Thesis

M.Tech Dissertation/ Report

B.Tech Project Report

Paper

Document (Tick):

Department: CSE

Enrolment No 161312

Name: Abhishek Bhardwaj

Contact No. 8894022073

E-mail. abhishekbhardwajsmi@gmail.com

Name of the Supervisor: Dr. Pradeep Kumar Gupta

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Type text here

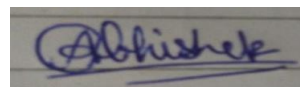
AOTA-Advance Over The Air

### UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

#### Complete Thesis/Report Pages Detail:

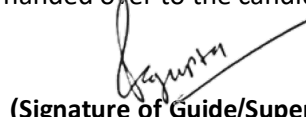
- Total No. of Pages = 70
- Total No. of Preliminary pages = 7
- Total No. of pages accommodate bibliography/references = 62



(Signature of Student)

### FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....<sup>14</sup> (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.



(Signature of Guide/Supervisor)



Signature of HOD

### FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"><li>• All Preliminary Pages</li><li>• Bibliography/Images/Quotes</li><li>• 14 Words String</li></ul>		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)**