Interface Specification of NCMC Ecosystem

Version 1.2 (Part IV to Part VII)

Submitted to

Ministry of Housing and Urban Affairs

Prepared by

Centre for Development of Advanced Computing, Noida

A Scientific Society of the Ministry of Electronics and Information Technology (MeitY), Govt. of India

Anusandhan Bhawan, C-56/1, Institutional Area, Sector 62, Noida-201307

Ph: 91-120-3063311-14 Website: http:\\www.cdac.in

February 28, 2020

Contents

Acknowledgement

Centre for Development of Advanced Computing (CDAC) wishes to extend its deepest thanks to Ministry of Housing and Urban Affairs (MoHUA) for sanctioning of the Research project entitled as “Research study for developing ecosystem for Metro Gate Validation Terminal”. CDAC also appreciates and extends the gratitude for the timely support and coordination at each and every level of communication with MoHUA.

CDAC has been very much thankful for the Expert group which consists of IIT Bhilai, Delhi Metro Rail Corporation (DMRC), National Payments Corporation of India (NPCI), Bureau of Indian Standards (BIS), Reserve Bank of India (RBI), Department of Financial Services (DFS), Delhi Integrated Multi-Modal Transit System (DIMTS), State Bank of India (SBI), AXIS BANK, and STQC for their valuable review comments and timely support to bring out this Interface Specification Document.

CDAC also extends special thanks to National Payments Corporation of India (NPCI) and Delhi Metro Rail Corporation (DMRC) for their guidance and knowledge sharing. CDAC also expresses the heartiest wishes to the all the team members of the project who put their effort to bring the Interface Specification Document in a commendable manner.

CDAC extends thanks to BEL and all those who have, directly or indirectly, contributed to the Preparation and Finalization of Interface Specification Document, in particular to the financial support provided by MoHUA and all of its members.

And last but not the least, CDAC would like to extend its gratitude to all the public transport operators who has given their feedback on this Interface Specification document and subsequently provided their invaluable inputs and suggestions during the discussions held over two workshop sessions at the India Habitat Centre, New Delhi on 27th December, 2019 and at the CDAC, Noida premises on 13th

January,2020.

Introduction

The Ministry of Housing and Urban Affairs (MoHUA) have envisaged the development of a cashless fare payment mechanism which will work across all the public transport systems in the country such as metros, buses etc. leading to the establishment of an Interoperable Fare Management System (IFMS) bringing extreme convenience and ease of payment for passengers.

A committee was constituted by MoHUA under the chairmanship of Additional Secretary MoHUA with representatives of National Informatics Centre (NIC), Centre for Development of Advanced Computing (C-DAC), Bureau of Indian Standards (BIS), National Payment Corporation of India (NPCI) and the Ministry of Finance to indigenously develop an interoperable system which is vendor agnostic, operating system agnostic and includes card and device specifications & standards for transit which would also work across existing retail outlets in addition to all transport operators systems regardless of the mode and location of the service provider.

The Recommendation Report of committee, among other items, includes:

i) Adopting of EMV based Transit Specification for NCMC
ii) A list of items for which Standards are to be Developed
iii) Creating of Testing and Certification facilities for various things
iv) Initiate the work on designing the metro gates indigenously to minimize the dependency on international players
v) Develop a support base of vendors for providing certified terminals

In this respect, C-DAC had been awarded the project – "**Research study for developing ecosystem for Metro Gate Validation Terminal**" which is under progress.

The main objective of the project was to develop interface specification of NCMC

ecosystem which can be adopted as Indian Standard. The Interface Specifications of NCMC Ecosystem were drawn by CDAC and NPCI. It consists of seven parts. NPCI has defined the card specification (qSPARC) including Card- Validation Terminal Interface (Part I- III) and CDAC has defined the interface specifications of Automated Fare Collection Systems (Part IV to Part VII) which comprises of (Part IV: Transit Service, Part V: Terminal – AFC Backend Communication Interface, Part VI : AFC Ecosystem – Acquirer Interface and Part VII : Gate – Terminal Interface).These specifications V1.0 were finalized by common consensus of expert group comprising of MoHUA, IIT Bhilai, CDAC, NPCI, DMRC, DFS, RBI, SBI, BIS, STQC, DIMTS and Axis Bank.

To prove the efficacy of the Interface Specifications of NCMC Ecosystem V1.0, CDAC developed NCMC AFC solution and deployed in Delhi Metro Rail Corporation (DMRC) at three stations for field trials. The pilot implementation at DMRC was inaugurated by MoHUA on 31st January, 2019 and has been running successfully since then. Under this pilot, NCMC cards issued by multiple banks (around 25 banks) have been successfully tested. NCMC card deployed in Noida Metro Rail Corporation Limited (NMRCL) have also been successfully tested for interoperability in DMRC.

The international launch of NCMC based AFC System was done by Honourable Prime Minister at Vastral Gam Metro Station in Ahmedabad on 4th March 2019. This is India's First Indigenously Developed Payment Eco-system for transport consisting of NCMC Card, SWEEKAR (Swachalit Kiraya: Automatic Fare Collection System) and SWAGAT (Swachalit Gate).

Based on the feedback received during field trial, CDAC updated the Interface Specifications of NCMC Ecosystem V1.0 and submitted the Specifications V1.1 to MoHUA on 14th November 2019. MoHUA shared the specifications to Chief Secretaries of all States/UTs and Managing Directors of various Metro Rail Corporations for their feedback on 20th November 2019.

Two workshops were organized at the India Habitat Centre, New Delhi on 27th December, 2019 and at the CDAC, Noida premises on 13th January, 2020to discuss the feedback received from State Governments and various Metro Rail Corporations. Based on the discussions held in the workshop, the Specifications has been updated. This document contains interface specification of AFC Ecosystem V1.2 (Part IV-Part VII).

The Interface Specifications are planned to be adopted as National Standard through BIS. Along with Standardization, the suitable testing and certification process is also being made in the country for NCMC Ecosystem. With instruction from MoHUA, CDAC and STQC are working together to bring the testing methods and platforms at the earliest.

Part-IV describes the common service data elements, operator service data elements, transit service risk management and Fare Management. Once, the card-terminal interaction has been established as per the qSPARC specification then, the card is verified and authenticated. The transit risk management is necessary to be performed before allowing the commuter to pass through the transit gate.

Part-V specifies the interface between Server and Terminal and also describes the Inter Server Interface. Since this document is intended to offer the common standard specifications for the entire transit network used through-out the country, therefore the AFC system interface with terminal tends to be standardized and interoperable. The objective of this document is to build interoperable Terminal-Server interface based on the standard specifications mentioned and provide common design criteria for it. All of these are depicted in this document.

Part VI expresses the interaction between the AFC Ecosystem and the Acquirer. The specifications are followed by any entity which communicates with the acquirer. The ISO-8583 and RuPay Global Clearing and Settlement Technical Message Specification_V-1.9 must be referred for detailed information of AFC Ecosystem to Acquirer Interface and the communication data interface.

Part VII illustrates the basic requirement for the AFC gate which may be used in transit systems. The document also explains the interface requirement for the gate and terminal communication.

After each of interfaces is explained in separate parts such as Part IV to Part VII as mentioned above in detail, a common findings and recommendations are explained in last chapter of this document.

Chapter 1

Interface Specification of NCMC Ecosystem

PART IV:

Transit Service

Centre for Development of Advanced Computing (CDAC), Noida
Ministry of Communications & Information Technology (MeitY)
Government of India

Contents

List of Tables

List of Figures

Revision History

Table 1: Revision History

Date	Version	Author	Comments
May 8, 2018	V 1.0	CDAC, Noida	First Release
Nov 14, 2019	V 1.1	CDAC, Noida	• Changed transit service structure. • Include common service area and operator service area. • Change Terminal ID to Terminal Info • Change wallet balance before transaction to card balance. • Change size of terminal info from 40b to 48b. • Change size of card balance from 16b to 20b. • Change size of transaction amount from 12b to 16b. • Add Annexure – I for data elements of CSA. • Add Annexure – II for data elements of OSA. • Add Annexure – III for Error Codes • Add Annexure – IV for data elements of Metro, Bus and Parking • Add Annexure – V for Language Code
Feb 28, 2020	V 1.2	CDAC, Noida	• Updated Version History • Add Revision History • Updated introduction information regarding version number of qSPARC, OSA discounted fare type product is added.

Date	Version	Author	Comments
			• Version number of CSA defined by NPCI is added in section “Common Service Area” • Updated card balance information in CSA • Updated in section 3.2 “Fare Amount” • Updated in section 3.3 “Transaction Sequence Number” • OSA defined in section 4 is “recommendation” is added. • Updated section 3.3 and 4.2 transaction status i.e. “0000” and “0011” are the same in double tap and single tap respectively. • Modified metro transaction flow i.e. remove “if common service access flag is true” decision box from Exit flow. • Modified bus transaction flow i.e. include decision box for double tap fixed terminal and process block for terminal behave as Entry/Exit w.r.t valid card log as well as operator BR. • Updated section 4.1, OSA may be created at the time of pass creation. • Add Annexure VI: Examples

Version History

Table 2: Version History

Version	Prepared By	Reviewed By	Comments
V 1.0	Ms. Babita Joshi, Mr. Amit	Mr. Chandan Maity	First Release
V 1.1	Mr. Amit	Mr. Rajesh Kr. Kushwaha, Ms. Babita Joshi	Updated Draft Release
V 1.2	Mr. Amit	Mr. Rajesh Kr. Kushwaha, Ms. Babita Joshi	Final Release

References

Table 3: Reference

Reference	Title
RuPay Interface	RuPay Terminal Specification, V 2.0.0
NCMC	Common Service Area, V 3.1.4
NCMC	Part 5: Terminal to AFC Backend Interface
NCMC	Part 6: AFC Ecosystem – Acquirer Interface
NCMC	Part 7:Gate Terminal Interface

1. Introduction

This document describes the interface between terminal and transit service areas (CSA & OSA) present in NCMC card. NCMC Card has Common Service Area (CSA) and Operator Service Area (OSA) which shall be interacting with the terminal application to achieve the interoperability. This specification describes:

- CSA Data structure and its usage for interoperability.
- Transaction flow corresponding to metro, bus and parking using CSA.
- OSA data structure and its usage to facilitate the different types of product e.g. pass for metro, pass for bus and pass for parking as per their business rule.

1.1. Terms and Abbreviations

Table 1 shows the terms and abbreviations used in this document. All the “Type A” related definitions are used and described in the ISO/IEC 14443 document.

Table 4: Abbreviations

Abbreviation	
ACQ	Acquirer
b	Bit
B	Byte
BCD	Binary Coded Decimal
CSA	Common Service Area
IEC	International Electro-technical Commission
ISO	International Organization for Standardization
Km	Kilometer
n	Numeric
NCMC	National Common Mobility Card
OSA	Operator Service Area
PT	Penalty Time
qSPARC	Quick Specification for Payment Application of RuPay Chip
Ref	Reference
RFU	Reserved for Future Use
tp	1 unit of 10 paisa
TSN	Terminal Serial Number

1.2. Scope

NCMC card may contain or be linked with many services where common service is one of them. This document describes the common service area data elements, operator service area data elements, Metro, Bus and Parking processes in the eco-system. The scope of this document is limited to transit and its related application. The document is classified into two sections:

Common Service Area: It defines the data elements and its aspects associated with the logical and operational activity of transit services by any operator on global wallet linked with CSA.

Operator Service Area: It describes the products e.g. Pass in metro, bus or parking and products with discounted fare (operations required to be performed on global wallet).

Card-Terminal Interface is out of scope of this document.
Before creation of OSA, CSA should be present otherwise it needs to be created first.
Service (CSA/OSA) creation and its update as per RuPay Terminal Specification V2.0.0 defined by NPCI are out of scope of this document.

1.3. Data Element Types

Table 5: Data Element Types

Element Type	Meaning
Persistent	The values of such data elements can be set during personalization. It may be provided by the card application to the terminal and holds meaning even outside transaction processing. E.g. Version Number, Language info
Transient	The values of such data elements are changed by card application during transaction processing. E.g. Validation data elements, History
Mandatory	The data element is a required field and its value must be present in a transaction.
Optional	The data element is not a required field and its value can be left blank in a transaction.

NOTE: All data format will be present in binary form except where it is explicitly defined otherwise.

2. Transit Service Management

Introduction to Common Service & Operator Service Area

For Common Service Area guidelines refer “Common Service Area for transit data” defined by NPCI. Refer Annexure– I: CSA Data Elements and Annexure– II: OSA Data Elements for details about the data structure of CSA and OSA respectively.

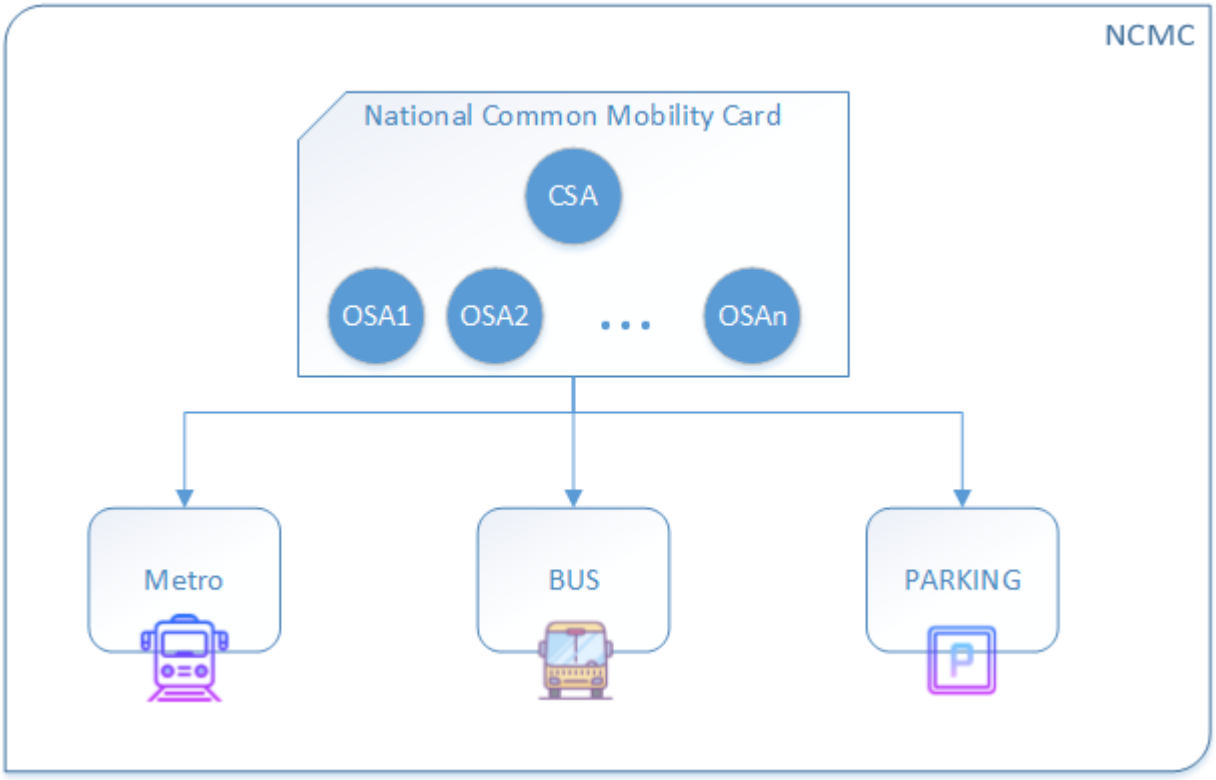


Figure 1: CSA & OSA

CSA- Common Service Area

OSA- Operator Service Area

In NCMC card, one common service (Mandatory) shall be present and there may be multiple Operator Service (Optional).

CSA

Common service is linked with the global wallet during the time of service creation and the data elements mentioned (Refer Annexure– I: CSA Data Elements) should be updated by operators during transaction. Common service shall be used by any operator i.e. operator will debit the amount from global wallet linked with the common

service and at the same time terminal shall require to write the log inside the common service area data element. The data elements defined in CSA is applicable for Metro, Bus and Parking etc.

OSA

The Operator service is operator specific and would be utilized for products defined by operator itself. If the data elements suggested (Refer Annexure– II: OSA Data Elements) are used, that shall be updated by operator during transaction.

3. Common Service Area Data Elements

Common service shall be accessed by all operators. Whenever there is a debit transaction from global wallet, then history and validation data elements shall always be updated. Service id for the CSA shall be defined by NPCI. For Common Service Area guidelines refer “Common Service Area for transit data (V 3.1.4)” defined by NPCI. CSA data elements are described in Annexure-I.

3.1 General Data

General data is generated & stored inside the card during the time of CSA creation. General data are persistent in nature i.e. it will not be changed during any transaction. It will consist of 2 bytes which are segmented by version no. (CSA version is defined by NPCI), Language and RFU. The detailed general data segment is shown in Fig 2.

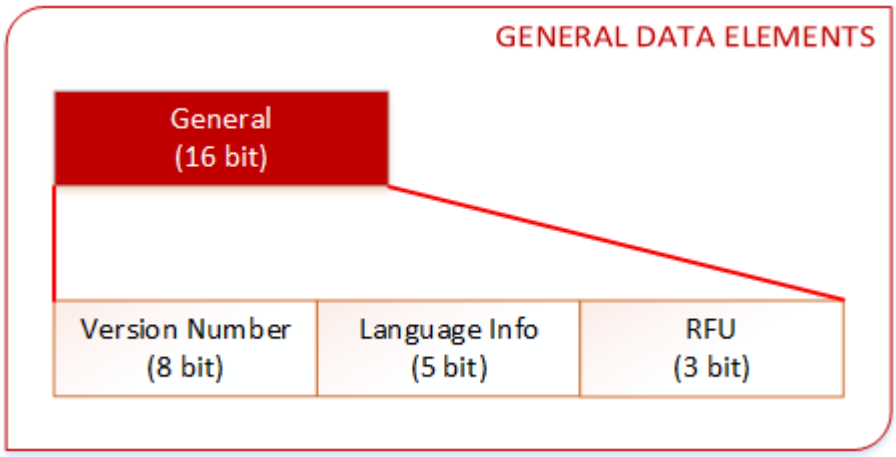


Figure 2: CSA General Data Elements

As per the above structure, general data parameters are persistent parameters and doesn’t required to be change during the transaction.

Table 6: General Data Parameters

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Version Number	M	Terminal	Persistent	8
Language Info	M	Terminal	Persistent	5
RFU	-	-	-	3

Version Number

It represents the version of the data specific to transit service. And it will remain common in all future specifications so that terminal application may be designed to handle multiple card layout revisions based on version number present in general data elements.

It is further divided into major and minor numbers (Refer Annexure– I: CSA Data Elements).

Language

Each value is specific to particular language which would be utilized by the terminal during the transaction for user to display information on the screen. (Refer Annexure– V: Language Code)

3.2 Validation Data

Validation data is generated and stored inside the card during the time of transaction at Entry/Exit/One tap terminal in Metro/Bus/Parking. Validation data are transient in nature i.e. it will be changed during the transaction at the validation terminal. It will consist of 19 bytes which are segmented by Error code, Product Type, etc. The detailed validation data segment is shown in Fig 3 below. (Refer Metro Transaction Flow described in section “Transit Service Management”)

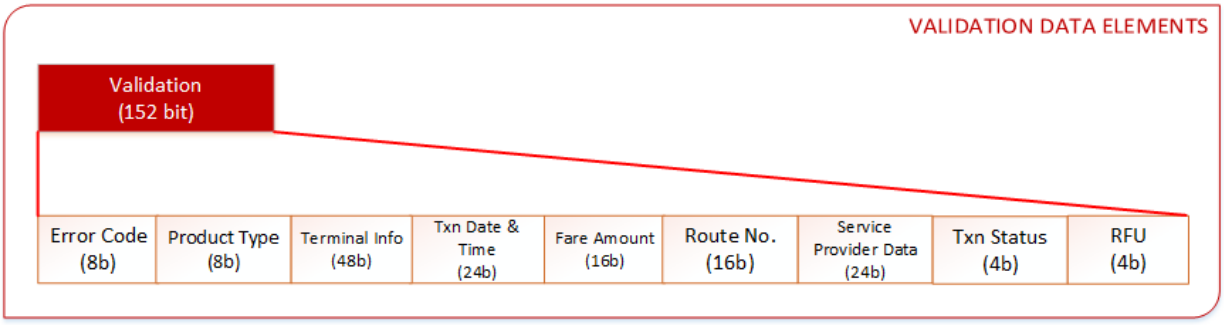


Figure 3: CSA Validation Data Elements

As per the above structure, validation data parameters are transient parameters and it would be changed during the transaction.

Table 7: Validation Data Parameters

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Error Code	M	Terminal	Transient	8
Product Type	M	Terminal	Transient	8
Terminal Info	M	Terminal	Transient	48
Transaction Date & Time	M	Terminal	Transient	24
Fare Amount	M	Terminal	Transient	16
Route Number	M	Terminal	Transient	16
Service Provider Data	M	Terminal	Transient	24
Transaction Status	M	Terminal	Transient	4
RFU	-	-	-	4

Error Code

It is a 1-byte code in which if any error occurs specific to transaction, it would be written into the card (e.g. Entry not found, Exit not found, etc.). Refer Annexure– III: Error Codes.

Product Type

It specifies the type of product i.e. single journey, discounted fare and pass.

“0x00 to 0x1F”&“0xFF” values are reserved by NCMC and other values (“0x20” to “0xFE”) would be used by operator.

Terminal Info

Terminal Info represents Acquirer, operator and terminal Id (operator specific). It is a composite parameter that means its combination of following parameters:

- Acquirer ID – 1 byte
  It shall be globally unique.
- Operator ID – 2 bytes.
  It shall be globally unique.
- Terminal ID – 3 bytes

It is operator specific.

Transaction Date & Time

It represents the date & time of the transaction. This field value will be stored with respect to card effective date[1].

Fare Amount

The maximum amount that shall be deducted by the terminal from the card which also includes the penalty charges if any. In double tap scenario, the operator will debit the amount on behalf of other operator in case the commuter did not finish their last journey (e.g. tailgating). In the case of single tap scenario, it will have the actual fare amount paid by the commuter. The value would be defined by the operator business rules.

Route Number

This code specific to routes and would be written by the terminal on the card as per the format defined by operator.

Service Provider Data

This data would be used by Operator as per their requirement and not required to change if not used.

Transaction Status

This will give the information about the transaction status as given in the following table.

The transaction status “0000” in double tap system is the same as “0011” in single tap system i.e. both show the previous transaction completed.

[1]Card Effective Date is the date on which the card is activated and customer can start using it.

Table 8: Transaction Status

Data	Value (4b)
Previous Transaction Completed/Exit Done	0000
Entry	0001
One Tap/Ticket	0011
RFU	0100 to 1111

3.3 History Data

History data is generated & stored inside the card during the time of non-zero transactions performed on a global wallet. History is transient in nature i.e. it will be changed during non-zero transactions. It will consist of 17bytes which are segmented by Terminal Id, Transaction Date & Time, etc. The detailed history data segment is shown in Fig 4.

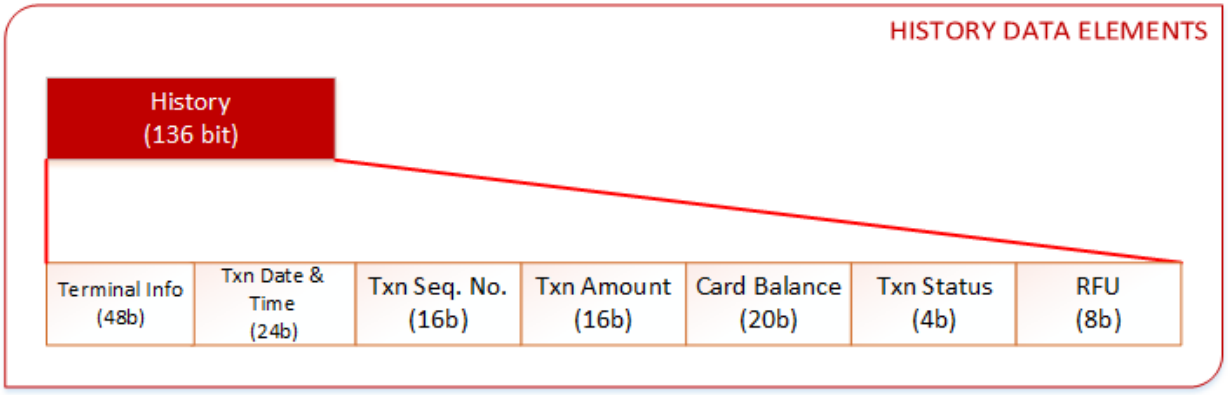


Figure 4: CSA History Data Elements

In CSA, four histories would be maintained while performing any operation on global wallet.

Table 9: History Data Elements

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Terminal Info	M	Terminal	Transient	48
Transaction Date & Time	M	Terminal	Transient	24
Transaction Sequence Number	M	Terminal	Transient	16

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Transaction Amount	M	Terminal	Transient	16
Card Balance	M	Terminal	Transient	20
Transaction Status	M	Terminal	Transient	4
RFU	-	-	-	8

Terminal Info

Terminal Info represents Acquirer, operator and terminal ID (operator specific). It is a composite parameter that means its combination of following parameters;

- Acquirer ID – 1 byte
  It shall be globally unique.
- Operator ID – 2 bytes
  It shall be globally unique.
- Terminal ID – 3 bytes
  It is operator specific.

Transaction Date & Time

It represents the date & time of the transaction. This field value will be stored with respect to card effective date.

Transaction Sequence Number

It is a terminal transaction unique number and will be updated if any non-zero debit operation is performed.

Card Balance

It represents the value corresponding to the amount available inside the card before a debit transaction takes place on a terminal. It is represented by “tp” (Refer Table **4**: Abbreviations).

Transaction Amount

This represents the fare which has been deducted by the terminal from the card balance. It is represented by “tp”.

Transaction Status

This will give the information about the transaction status as given in the following table. The transaction status “0000” in double tap system is the same as “0011” in single tap system i.e. both signify completed transactions.

Table 10: Transaction Status

Data	Value (4b)
Previous Transaction Completed/Exit Done	0000
Penalty Applied	0010
One Tap/Ticket	0011
RFU	0100 to 1111

Common Service Area Structure

Complete common service structure is shown below:

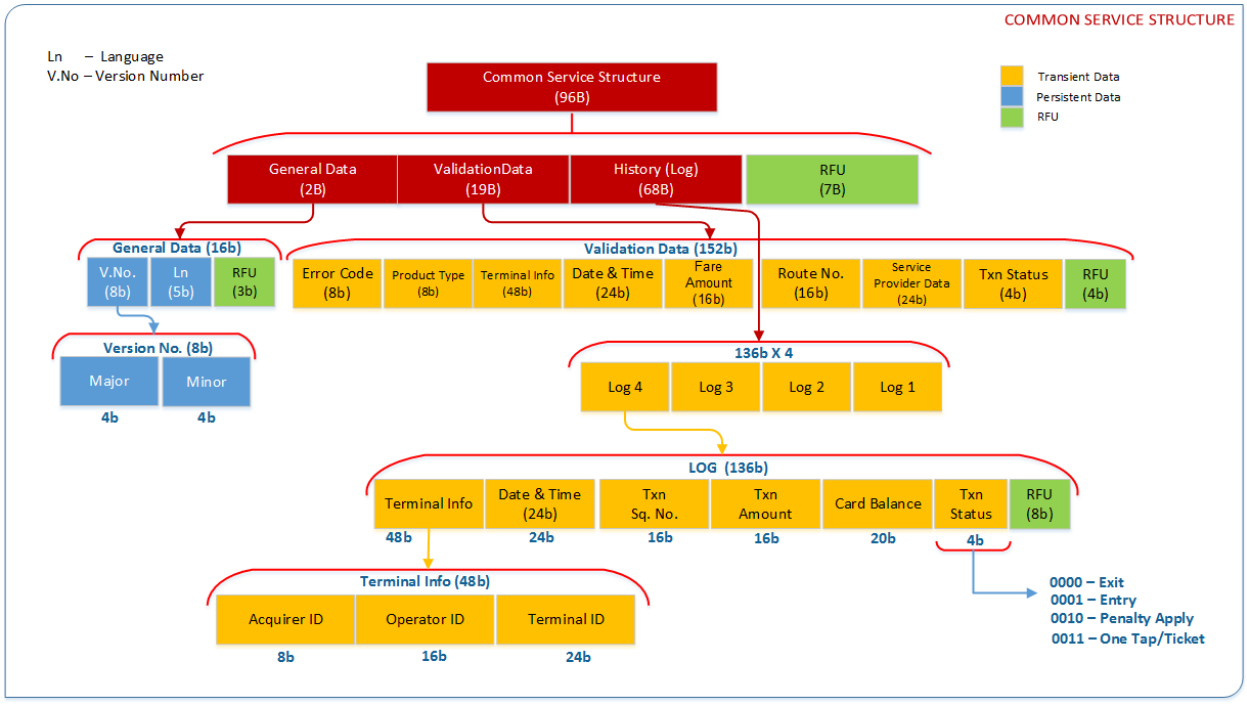


Figure 5: Common Service Area Structure

4. Operator Service Area Data Elements

Operator Service Area is operator specific service area and all data elements defined in this specification related to OSA are recommendation only which would be used by any operator as per their business rule. The area will be accessed by the operator who has created the service. This area has a space to store the last 2 history logs and 3 product types e.g. Metro pass, Bus pass and Parking pass.

If any pass has been accessed then,

At Entry, validation data should be updated,

At Exit or for one tap transaction, history & validation should be updated.

In case the product is a discounted fare type like an operator pass in OSA, where in the global wallet needs to be accessed to complete the transaction; the transaction should be done in the CSA. Validation data – validation as well as history– must be updated at Entry and Exit.

4.1 General Data

General data would be generated and stored inside the card during pass creation in OSA. General data is persistent in nature i.e. it will not change during the transactions at terminals. It shall consist of 2 bytes which are segmented by Version no., Language and RFU. The structure of the general data segment is shown below:

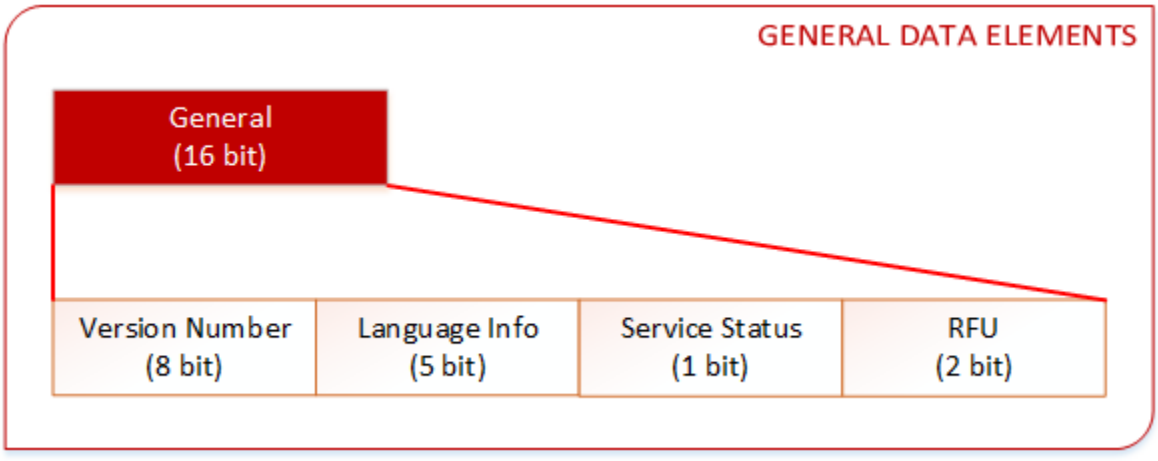


Figure 6: OSA General Data Elements

Table 11: General Data Parameters

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Version Number	M	Terminal	Persistent	8
Language Info	M	Terminal	Persistent	5
Service Status	M	Terminal	Persistent	1
RFU	-	-		2

Version Number

It represents the version of the data specific to transit service. It will remain common in all future specifications so that terminal application can be designed to handle multiple card layout revisions based on the version number present in general data.

It is further divided into major (b8-b5) and minor (b4-b1) number.

Language

Each value is specific to a particular language which would be utilized by the terminal during the transaction to display information on the screen to the user. Language codes are defined in Annexure–V: Language Code.

Service status

It will give information about the validity of a pass or passes (in case more than one pass is present). This 1-bitvalue will give information only when it is set to '1'.

1 – It means, if passes in OSA are present but none of them are valid, it will be set to 1 by customer care or terminal (as per operator rules).

0 – it doesn't give any information whether passes are valid or not.

4.2 Validation Data

Validation data is generated & stored inside the card during the time of transaction at Entry/Exit/One tap terminal. Validation data are transient in nature i.e. it will be changed during the transaction at terminal. It shall consist of 8bytes which are segmented by Error code, Product Type, etc. The detailed validation data segment is shown below:

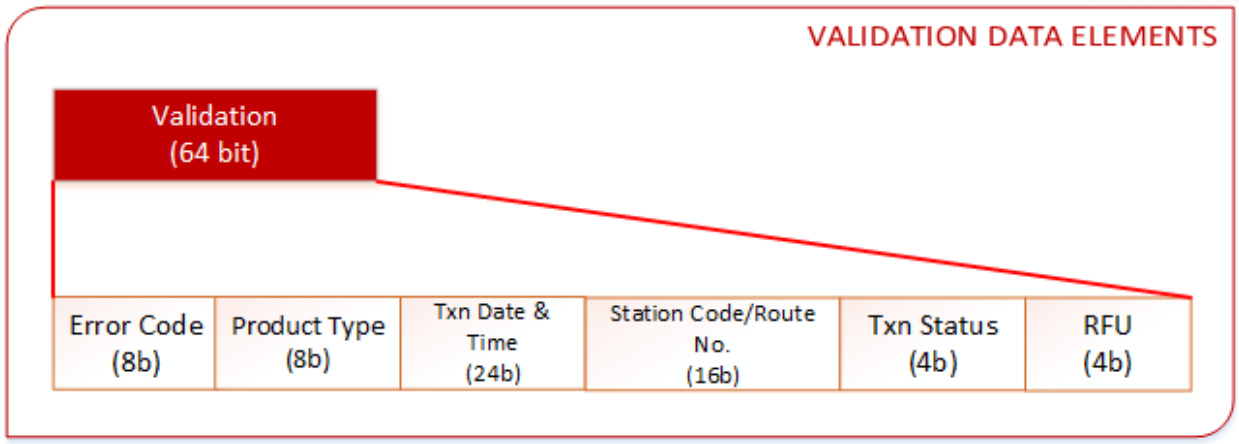


Figure 7: OSA Validation Data Elements

As per the above structure, validation data parameters are transient parameters and it would be changed during the transaction.

Table 12: Validation Data Parameters

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Error Code	M	Terminal	Transient	8
Product Type	M	Terminal	Transient	8
Transaction Date & Time	M	Terminal	Transient	24
Station Code/Route No.	M	Terminal	Transient	16
Transaction Status	M	Terminal	Transient	4
RFU	-	-	-	4

Error Code

It is a 1-bytefield in which, if any transaction specific error occurs, the error code would be written into the card e.g. Entry not found, Exit not found, etc.

Product Type

It specifies the type of product i.e. single journey, discounted fare and pass.

“0x00 to 0x1F” and “0xFF” values are reserved for NCMC and other values (“0x20 to 0xFE”) may be used by operator.

Transaction Date & Time

It represents the date & time of the transaction. The field value will be stored with respect to card effective date.

Route Number

This is the code specific to routes and would be written by the terminal on to the card as per the format defined by operator.

Transaction Status

This will give the information about the transaction status as given in the following table.

The transaction status “0000” in double tap system is the same as “0011” in single tap system i.e. both show the previous transaction completed.

Table 13: Transaction Status

Data	Value (4b)
Previous Transaction Completed/Exit Done	0000
Entry	0001
One Tap/Ticket	0011
RFU	0100 to 1111

4.3 History Data

History data is generated & stored inside the card during the time of transaction at a validation terminal. History is transient in nature i.e. it changes during a transaction performed at the validation terminal. It will consist of 18 bytes which are segmented by Terminal Info, Transaction Date & Time, etc. The detailed history data segment is shown below:

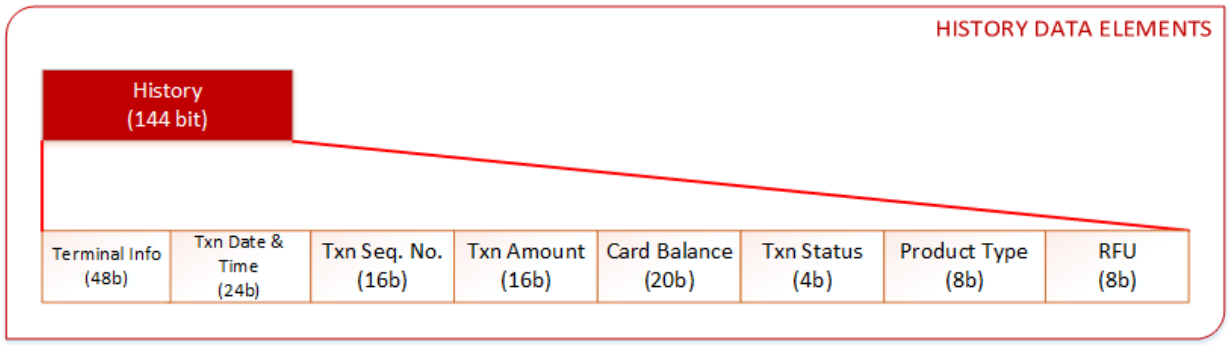


Figure 8: OSA History Data Elements

In OSA, two histories would be maintained while performing any operation on a pass during exit.

Table 14: History Data Parameters

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Terminal Info	M	Terminal	Transient	48
Transaction Date & Time	M	Terminal	Transient	24
Transaction Sequence Number	M	Terminal	Transient	16
Transaction Amount	M	Terminal	Transient	16
Card Balance	M	Terminal	Transient	20
Transaction Status	M	Terminal	Transient	4
Product Type	M	Terminal	Transient	8
RFU	-	-	-	8

Terminal Info

Terminal Info represents Acquirer, operator and terminal Id (operator specific). It is a composite parameter that means its combination of following parameters;

- Acquirer ID – 1 byte
- Operator ID – 2 bytes
- Terminal ID – 3 bytes

Transaction Date & Time

It represents the date & time of the transaction. This field value will be stored with respect to card effective date.

Transaction Sequence Number

It is a terminal transaction unique number.

Card Balance

It represents the value corresponding to the amount available inside a card before a transaction takes place on a validation terminal. It is represented by “tp”. It may also indicate trips or reward points left on the card depending on the product type.

Transaction Amount

It represents the fare which has been deducted by the terminal from the card balance. It is represented by “tp”. It may also indicate trips or reward points deducted from the card.

Transaction Status

This will give the information about the transaction status as given in the following table.

The transaction status “0000” in double tap system is the same as “0011” in single tap system i.e. both show the previous transaction completed.

Table 15: Transaction Status

Data	Value (4b)
Previous Transaction Completed/Exit Done	0000
Entry	0001
Penalty Applicable	0010
One Tap/Ticket	0011
RFU	0100 to 1111

4.4 Pass Info

Operator may define their own data element for pass info as per their requirement. However, it is recommended to use the Pass Info structure defined in Annexure–IV: Data Elements for Pass. A maximum of 3 passes can be stored in the operator specific area.

4.5 Pass Creation/Renewal

Operator may define pass in OSA as per their business rule. The purchase amount for a pass or its renewal may be taken by operator from global wallet, cash or other modes as per operator requirement.

Operator Service Area Structure

Complete operator service structure is shown below:

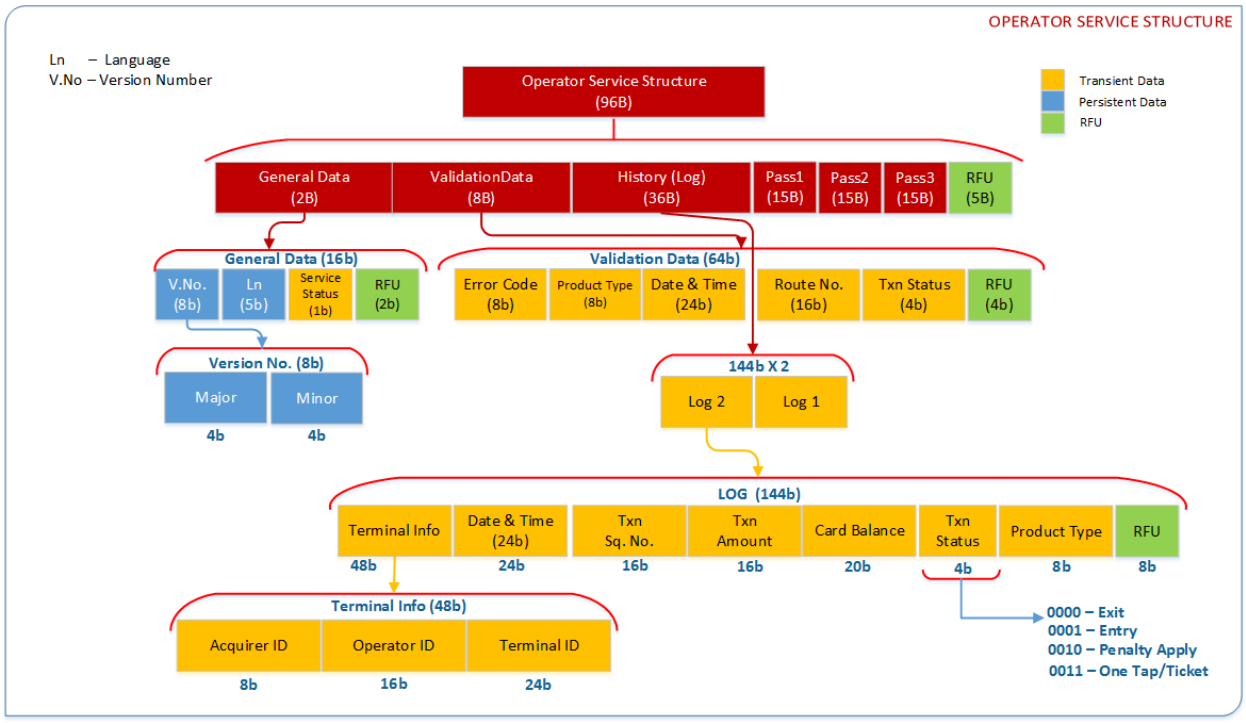


Figure 9: Operator Service Area Structure

5. Transit Service Management

It defines the parameters for calculating the transit fare. In transit service management, transit fare can be calculated by using parameters shown in CSA & OSA. The transit fare rule and policy are outside the scope of this document. Utilizing the parameters in different architecture i.e. Metro, Bus & Parking are described below.

5.1. Metro Transaction Flow Architecture

Transaction flow described will be applicable to services like Metro where both Entry and Exit terminals are present.

Entry/Exit transaction flow during transit in Metro is shown in Figure 10: Metro Transaction Flow.

The flow describes the usage of common service as well as operator services present inside card at the time of transaction performed by terminal.

Note:

1. Common service would be accessed by any operator and perform debit operation using global wallet but operator service will be specific to individual operator which cannot be accessed by other operator and this service will handle the transaction by incrementing or decrementing the limit (created by operator itself e.g. pass limit) present in operator service data element i.e. operator service will not be doing any transaction using Global Wallet. In case, any operator required to doa transaction using the global wallet then, it has to be done using CSA only.
2. If the terminal is accessing the Global Wallet then, it shall update the history data element present in CSA. Updates are done in FIFO manner irrespective of whether the terminal is ENTRY or EXIT.

At Entry terminal,

After placing the card, terminal checks whether operator service is present inside the card or not.

If yes,

1. Terminal will proceed with the operator service id by selecting the service Id allocated to it then, validate each product e.g. tourist pass created for Metro.

At a time, in operator service, only 3products can be created by operator. Multiple

pass would be possible as described below:

i. Metro pass
ii. Feeder Bus pass
iii. Parking pass

2. Validate product (by default, the first product will have the highest priority when more than one product is present)

During validation of product, 3 scenarios are possible

Scenario – 1: Valid product present

Scenario – 2: Invalid product present

Scenario – 3: Valid product present but will require amount to be debited from global wallet (e.g. student pass with discounted fare)

Scenario – 1:

If product is validated successfully, then the transaction will be processed as per the business rule specific to operator.

Scenario – 2:

In case of Metro, if all products fail validation, then terminal will not perform any transaction and information would be displayed to commuter like “Go to Customer Care”[2]. Displayed information would be as per the language set inside the card.

Scenario – 3:

If a discounted fare type product is present, then the terminal is required to select CSA using the Service Id allocated to it. Then it should check the validation data element present inside the CSA whether the previous transaction performed by any operator was successful or not. This is determined using the error code and subsequently transaction status.

If the error code and transaction status is not “00”, then terminal will prompt commuter with message “Go to Customer Care”.

If Error code and transaction status is “00” then,

[2]This message is for illustration purposes only. Operators may choose to display this message in their own form.

Terminal will check whether the required minimum balance present in global wallet or not.

If either error code or transaction status is not equal to "00" or minimum balance is not present then,

Prompt commuter to "Go to Customer Care"

If OSA not present,

Update validation data elements present in CSA (if money is not debited at entry terminal); or update the history data elements in a FIFO manner; also update validation data element if money is debited at the time of ENTRY (this depends on the Business rule of operator)

At Exit terminal,

It depends on the operator whether they will give priority to CSA or OSA at the EXIT terminal. This will make the transaction process faster if any operator knows that all commuters travel through them have to have operator pass inside card then, terminal will have a priority to OSA rather than CSA and terminal will select OSA directly i.e. transaction will be finished in single cycle (it means, from reading card till updating of data elements in OSA).

Two scenarios would be possible:

1. Terminal first select common service area (CSA)
2. Terminal first select operator service area (OSA)

Scenario – 1: Terminal first select CSA

1. Terminal first check the validation data elements present in CSA i.e. two data elements required to check
   a. **Error Code**

If error code data element is "00", then check "transaction status".

Else, prompt the commuter to "Go to Customer Care" as per the language set in card.

   b. **Transaction Status**

If transaction status is "ENTRY"

If transaction status data element is "ENTRY" type, then the terminal proceeds to check if ENTRY has been done before through same operator or different operator. (This check is performed using Operator Id present with the Terminal ID field. This situation only arises when commuter exits the 1st operator without showing card at EXIT terminal (tailgating) and then enters into 2nd operator premises, tailgates his way through the ENTRY terminal and is now showing his card at the EXIT terminal. The 1st and 2nd operator may be the same).

If same operator ID then,

Terminal will check, whether commuter is traveling within time limit (defined by operator)

If exceeding time limit then,

Terminal writes the error code in the validation type data element and prompts commuter to "Go to Customer Care"

Else (within time limit) then,

Terminal will check whether fare calculated is less than the Global Wallet amount or not.

If fare is less than the global wallet available balance then,

Clear the transaction type and error code (i.e. "00") and update the history data element in FIFO manner while debiting the amount.

If fare is more than the balance available in global wallet then,

Write the error code in CSA validation data element and prompts commuter to "Go to Customer Care"

Else (if different operator Id) then,

Write the error code in CSA validation data element and prompts commuter to "Go to Customer Care"

Else (transaction status is not ENTRY) then,

There will be one possibility:

1. Commuter has entered through different product e.g. pass present in operator service

   If operator service id is not present in card or pass is not available then,

   Prompts commuter to "Go to Customer Care".

Note:

In case terminal is not saving the information whether the operator service present or not at the time of application selection, then terminal needs to perform 2nd transaction by attempting to select the operator service.

Else (operator service id present in card) then,

Terminal will select the operator service i.e. 2nd transaction and do the checks for data elements value present in OSA

a. Error Code
b. Transaction Status
c. Within Time Limit

a. Error code

If error code not present i.e. "00" then,

check transaction status in OSA validation data element

Else,

Prompts the commuter to "Go to Customer Care".

b. Transaction Status

If transaction status is "Entry" then,

Check commuter travelled within time limit (as per operator Business Rule)

Else,

Write the error code for "Last Entry Skipped" in OSA validation data element and prompts the commuter to "Go to Customer Care".

Scenario – 2: Terminal first select OSA

1. Terminal first checks whether operator service is present in card or not.
   In case not present, transaction will process as per the process described in Scenario-1 (above, except select OSA again).
   If present then, proceed to point 2 described below.
2. Terminal checks the validation data element present in OSA i.e. two data elements required to check

a. Error Code

If error code data element is “00”, then check “transaction status”.

Else, prompts the commuter to “Go to Customer Care” as per the language bit set in card if terminal support multiple language.

b. Transaction Status

If transaction status is “Entry”

If transaction status data element is “ENTRY” type then, terminal proceed for checking whether commuter has travelled within time limit (as per business rule defined by operator).

If exceeding time limit then,

Terminal writes the error code in validation type data element and prompts commuter to “Go to Customer Care”

Else (within time limit) then,

Terminal will process the pass as per the business rule defined by operator

Else (transaction status is not Entry)

In case not present, transaction will be processed as per the process described in Scenario-1 (above, except select OSA again).

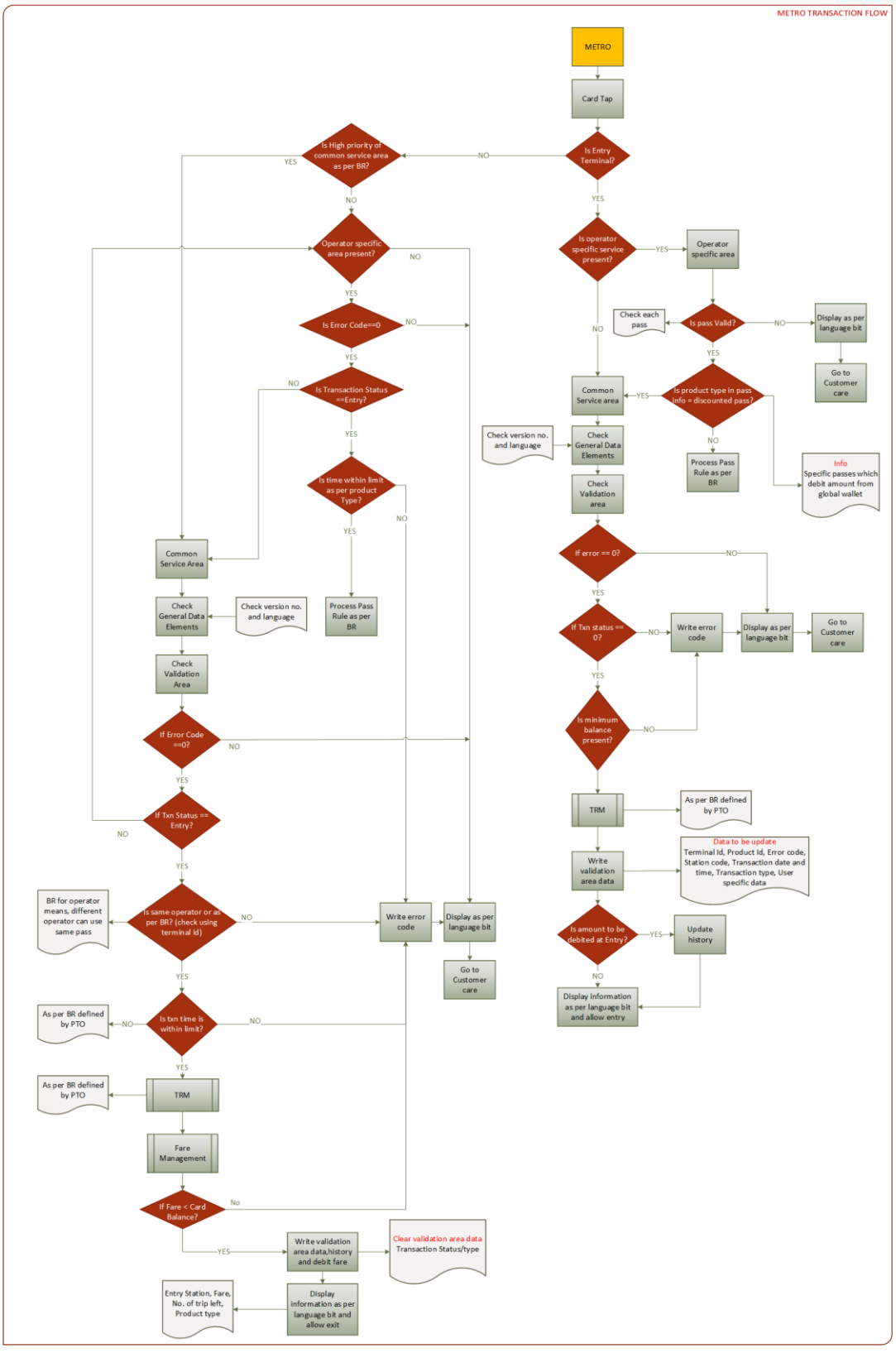


Figure 10: Metro Transaction Flow

5.2. Bus Transaction Flow Architecture

The service defined as common service and operator service will be applicable for bus transit.

In Bus transit, there will be 3 possibilities:

1. **Single Tap (Unattended Terminal)**

In single tap, commuter is required to show their card at the time of ENTRY and select the destination bus stand, according to which terminal will debit amount from CSA or decrement the pass limit present in OSA. However, option to select source and destination/Fixed Fare may also be given based on operator requirement.

2. **Double Tap (Unattended Terminal)**

In double tap, transaction will be same as METRO transaction flow. Please refer METRO transaction flow describe before.

3. **Double Tap Single Fixed Validator (Unattended Terminal)**

In double tap, transaction will be same as METRO transaction flow once the terminal is configured as Entry/Exit according to valid card log as well as operator business rule. Please refer METRO transaction flow describe before.

4. **Single Tap (Attended Terminal)**

In this case, commuter will show their card the time of ENTRY to BUS conductor who will debit the amount as per the start and destination stand either from CSA or OSA

Scenario – 1: Single Tap (Unattended Terminal)

Terminal first checks whether card has operator service using the service Id defined by NPCI.

If card is having operator service (OSA) then,

1. Terminal first checks, error code and transaction status are “0”. In case other than “0”, terminal prompts to commuter to “Go to Customer Care”.

   Else (error code & transactions status is “0”) then,

   Terminal will check service status (it will give information whether any pass

available or not) is “0”, then go for validation of pass.

Else (no pass is available) then,

Terminal will select CSA after getting confirmation input from commuter.

2. If pass available then, 3 possibilities will be applicable:

2.1. Pass available and valid

2.2. Pass available but not valid

2.3. Pass available and valid but discounted fare type (debit amount from global wallet)

2.1. Pass available and valid

In this case, terminal will process pass as per operator business rule.

2.2. Pass available but not valid

In this case, terminal first validate the pass (or passes if more than one pass available) as per the operator business rule and if none of the available pass is valid then terminal will prompt commuter to “Go to customer care[3]”. Customer care or terminal (as per operator) would set the service status bit to ‘1’ in general data element so that next time any other terminal will not require validating the pass.

Terminal may select CSA after getting confirmation input from commuter.

2.3. Pass available but discounted fare type (debit amount from global wallet)

In this case, terminal will select CSA

In CSA,

Terminal will check whether error code & transaction status is both “0”.

If Yes,

Fare calculation has been done as per the inputs (Entry Bus stand), Destination Bus stand (by commuter), Route, Class etc. as per business rule defined by operator.

If fare is more than available balance in card global wallet then,

Terminal will write the error code without debit required fare in CSA validation area and prompts commuter to “Go to Customer Care”.

[3]Depending on the language value set in General data section of OSA and if terminal supports multiple languages.

Else (fare less than available balance) then,

Terminal requires to update history data element in cyclic manner and make error code as well as transaction status to “0”

If No,

Terminal prompts commuter to “Go to Customer Care”

Scenario – 2: Double Tap (Unattended Terminal)

This will be same as Metro transaction flow.

Scenario – 3: Single Tap (Attended Terminal)

Working will be same as single tap (unattended terminal), instead of commuter required to operate the terminal, conductor will do the operation who will be much more familiar with the terminal and its operation.

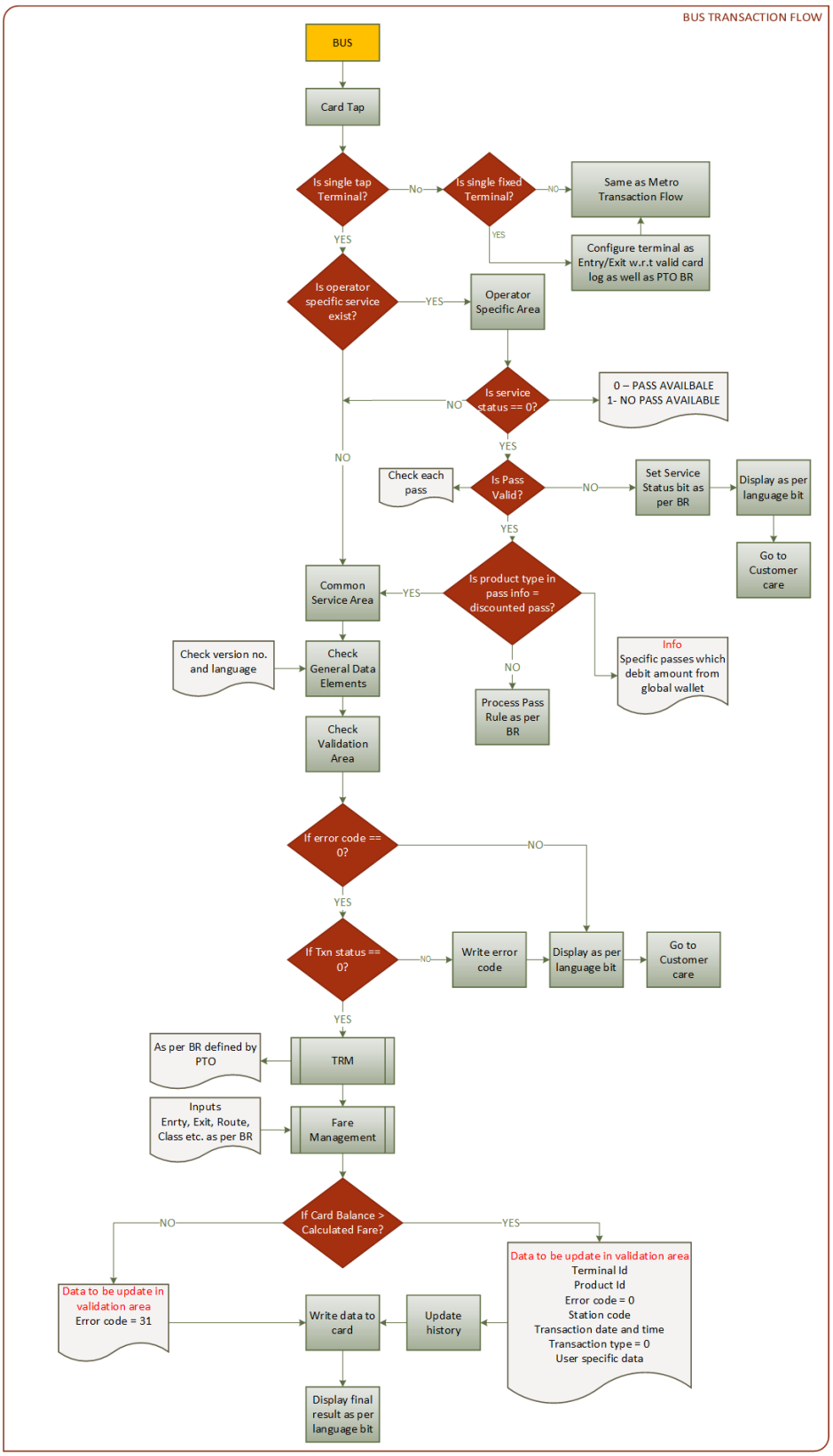


Figure 11: Bus Transaction Flow

5.3. Parking Transaction Flow Architecture

The service defined as common service and operator service will be applicable to parking as well.

In Parking, there are 2 possibilities:

1. **Double Tap (Attended Terminal i.e. same terminal will be used at ENTRY & EXIT)**

In this, same terminal will be used at the time of ENTRY/EXIT and operator will select the option (configure the application) as per the situation whether commuter going inward/outward respectively.

2. **Double Tap (Unattended Terminal i.e. different terminal will be used at ENTRY & EXIT)**

In this, two different terminals will be present at different points and commuter required to input the information (as per Business Rule of operator) and present the card at terminal for writing the ENTRY information inside the card.

Same card will be used at the time of EXIT and commuter only allowed once verified the ENTRY information or other details (as per the Business Rule).

Scenario – 1: Double Tap (Attended Terminal i.e. same terminal will be used at ENTRY & EXIT)

At ENTRY,

Terminal will check whether operator service is present in the card or not.

If No,

Terminal will proceed further as per the business rule define by operator (i.e. whether terminal will create OSA using online terminal)

If Yes,

Terminal will check whether any pass present in the card corresponding to Parking or not.

If pass is not present then,

Terminal will proceed further as per the business rule define by operator (i.e. whether terminal will create pass or not). If creation of pass is not successful then, proceed as per business rule.

If pass is present then,

Terminal will validate the pass (as per business rule defined by operator i.e. check data elements to validate a pass)

3 case will be possible i.e. pass invalid, pass valid and pass valid but discounted fare type.

Validation of pass would be done by Pass information present in OSA as per defined by operator. In case, pass is not valid then prompts commuter to “Go to Customer Care”.

Terminal will also send the same information to AFC. AFC data will be useful in case rider card is misplaced.

AT EXIT,

Terminal will check whether operator service is present in the card or not.

If No,

Terminal will proceed further as per the business rule define by operator (i.e. whether terminal will select CSA or not)

If Yes,

Terminal will check whether any pass present in the card corresponding to Parking or not.

If pass is not present then,

Terminal will proceed further as per the business rule define by operator (i.e. whether terminal will select CSA or not)

If pass is present then,

Terminal will validate the pass (as per business rule defined by operator i.e. what to check to validate a pass)

3 case are possible i.e. pass invalid, pass valid and pass valid but discounted fare type.

If pass is not valid then,

Terminal will proceed further as per the business rule define by operator. (i.e. whether terminal will select CSA or not)

If pass is valid then,

Terminal will process the pass as per the business rule defined by operator.

If pass is valid but discounted fare type then,

Terminal will calculate the fare (as per business rule) and select the CSA for debit the calculated fare from Global wallet and at the same time terminal will update the history data element present in CSA in FIFO manner.

Terminal will also send the same information to AFC.

Scenario – 2: Double Tap (Unattended Terminal i.e. different terminal will be used at ENTRY & EXIT)

In this case, one terminal configured as ENTRY terminal and other will be as EXIT.

Process of transaction will be same as defined in Scenario – 1.

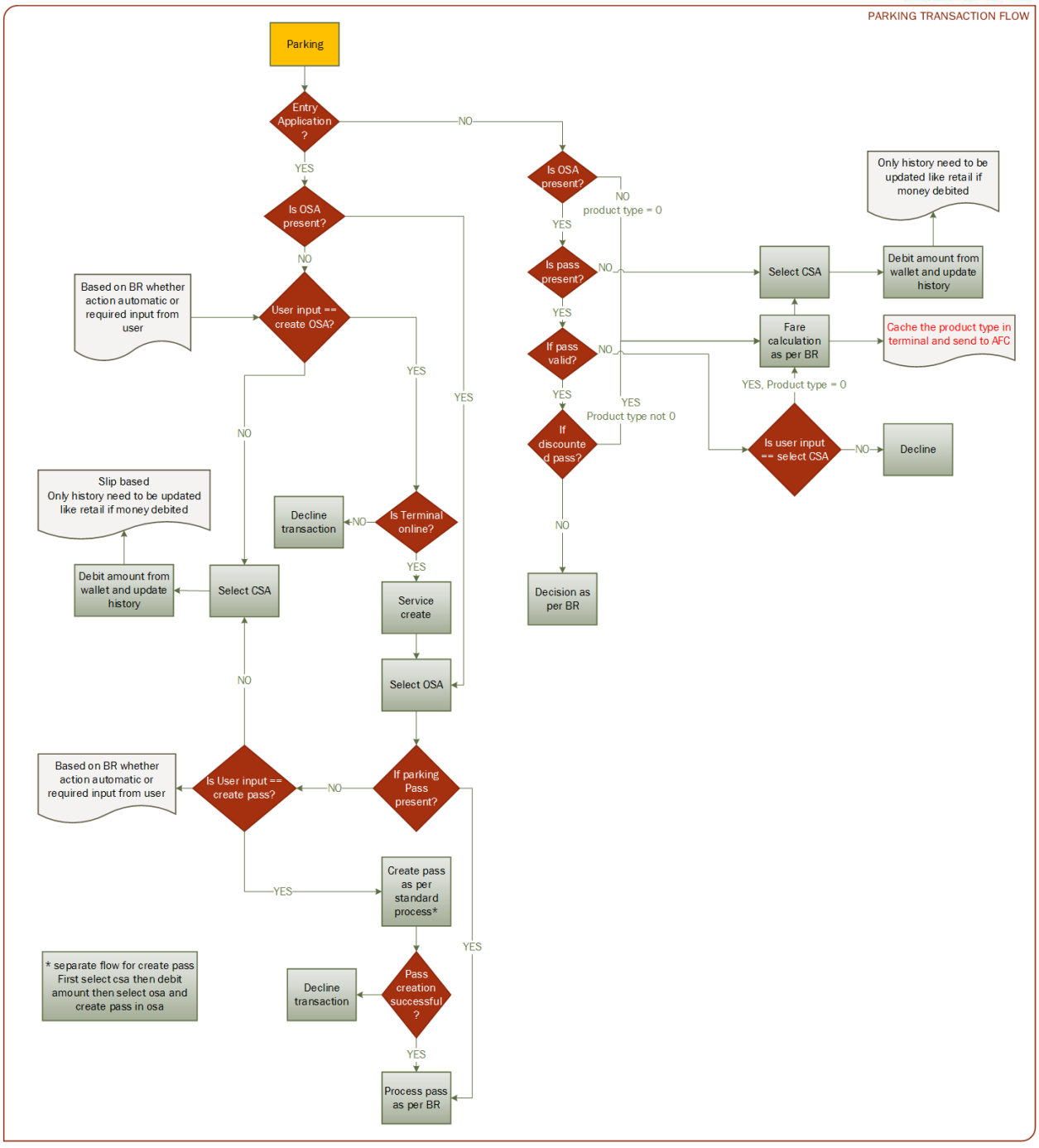


Figure 12: Parking Transaction Flow

Annexure– I: CSA Data Elements

Table 16: Data Element in Common Service Area

Sr. No	Field	Bits	Remarks
General Info			
1.	Version Number	8	It represents the version of the data specific to transit service. This is a one-byte numeric data of which the MSB nibble (b8-b5) describes the major and the LSB nibble (b4-b1) tells the minor version number
2.	Language Info	5	Bit value represents language code, which would be utilized by the terminal to display information on the screen. “00000” - English (Default language) Language list includes all eighth schedule languages.
3.	RFU	3	
Length of General Info Data		16	2 Bytes
Validation			
4.	Error Code	8	Error code would be update on card during transaction. 00 - No error recorded refer error code define in global error code list
5.	Product Type	8	It specifies the type of product. “0x00 to 0x1F” and “0xFF” is reserved; other value (“0x20 to 0xFE”) will be used by operator.

Sr. No	Field	Bits	Remarks

6.	Terminal Info	48	Validation terminal ID, Combination of Acquirer ID - 1 bytes Operator ID - 2 bytes Terminal ID - 3 bytes
7.	Transaction Date and Time	24	The date & time of the transaction. EPOCH[4] Format, Base date will be effective date
8.	Fare Amount	16	Basis on operator rule one of the following will be updated: - Maximum Fare - Penalty for metro operator - Fare amount for bus
9.	Route No	16	This is the code specific to route would be written by the terminal on the card as per the format defined by operator
10.	Service Provider Data	24	Data update by Service provider
11.	Transaction Status	4	Transaction Type 0000 - Exit/Transaction complete 0001 - Entry 0010 - Penalty Apply 0011 - One Tap/Ticket Rest combinations - RFU
12.	RFU	4	
Total Length of Validation Data		152	19 Bytes

Sr. No	Field	Bits	Remarks

[4] Transaction date & time is calculated by considering card effective date as reference date instead of 01/01/1970. The data stored in minutes only i.e. not consider seconds.

History			
13.	Terminal Info	48	Unique Id of a Validation terminal, Combination of Acquirer ID - 1 byte Operator ID - 2 bytes Terminal ID - 3 bytes
14.	Transaction Date and Time	24	The date & time of the transaction. EPOCH[5] Format, Base date will be effective date
15.	Transaction Sequence Number	16	It represents terminal transaction unique number.
16.	Transaction Amount	16	Fare Amount, which has been deducted by global wallet
17.	Card Balance	20	Amount available in global wallet before transaction
18.	Transaction Status	4	Transaction: 0000 - Exit/Transaction complete 0010 - Penalty Apply 0011 - One Tap/Ticket Rest combinations - RFU
19.	RFU	8	
Total Length of History Data		136	17 Bytes (17 * n = 68 Bytes) Where n is no. of logs in history. i.e. 4
20.	RFU	56	7 Bytes

[5]Transaction date & time is calculated by considering card effective date as reference date instead of 01/01/1970. The data stored in minutes only i.e. not consider seconds.

Annexure– II: OSA Data Elements

Table 17: Data Element in Operator Service Area

Sr. No	Field	Bits	Remarks
General Info			
1.	Version Number	8	It represents the version of the data specific to transit service. This is a one-byte numeric data of which the MSB nibble (b8-b5) describes the major and the LSB nibble (b4-b1) tells the minor version number
2.	Language Info	5	The Value represents language code, which would be utilized by the terminal to display information on the screen. “00000” - English (Default language) Language list includes all eighth schedule languages.
3.	Service Status	1	1 – It means, if passes in OSA are present but none of them are valid, it will be set to 1 by customer care or terminal (as per operator). 0 – It doesn’t give any information about passes are valid or not.
4.	RFU	2	
Length of General Info Data		16	2 Bytes
Validation			
5.	Error Code	8	Error code would be update on card during transaction. 00 - No error recorded refer error code define in global error code list

Sr. No	Field	Bits	Remarks
6.	Product Type	8	It specifies the type of product. “0x00 to 0x1F” and “0xFF” is reserved; other value (“0x20 to 0xFE”) will be used by operator.
7.	Transaction Date and Time	24	The date & time of the transaction. EPOCH[6] Format, Base date will be effective date
8.	Route No	16	This is the code specific to route would be written by the terminal on the card as per the format defined by operator.
9.	Transaction Status	4	Transaction Type 0000 - Exit/Transaction complete 0001 - Entry 0010 - Penalty Apply 0011 - One Tap/Ticket 0100 to 1111 - RFU
10.	RFU	4	
Total Length of Validation Data		64	8 Bytes

[6] Transaction date & time is calculated by considering card effective date as reference date instead of 01/01/1970. The data is stored in minutes only i.e. seconds are not accounted for.

Sr. No	Field	Bits	Remarks
History			
11.	Terminal Info	48	Unique ID of a Validation terminal, Combination of Acquirer ID - 1 byte Operator ID - 2 bytes Terminal ID - 3 bytes
12.	Transaction Date and Time	24	The date & time of the transaction. EPOCH[6] Format, Base date will be effective date
13.	Transaction Sequence Number	16	It represents terminal transaction unique number.
14.	Transaction Amount/Limit	16	Fare Amount, which has been deducted by global wallet
15.	Card Balance/Limit	20	Amount available in global wallet before transaction
16.	Transaction Status	4	Transaction: 0000 - Exit/Transaction complete 0010 - Penalty Apply 0011 - One Tap/Ticket 0100 to 1111 - RFU
17.	Product Type	8	It specifies the type of product. “0x00 to 0x1F” and “0xFF” are reserved, other value (“0x20 to 0xFE”) will be used by operator.
18.	RFU	8	
Total Length of History Data		144	18 Bytes (18 * n = 36 Bytes) Where n is no. of logs in history i.e. 2
Pass			
19.	Pass	120	15 Bytes
Total Length of Pass Info		360	45 Bytes (15 * n = 45 Bytes)

			Where n is no. of pass i.e. 3
20.	RFU	40	5 Bytes

Annexure– III: Error Codes

Table 18: Error Code

S.No.	Error Name	Error Code
1.	Amount not sufficient for Entry/Exit	1
2.	Torn Transaction	2
3.	Entry not found in validation area in CSA	3
4.	Exit not found in validation area in CSA	4
5.	service area present but all pass invalid	5
6.	Time Exceed	6
7.	Card Expired	7
8.	RFU	8- 100 and 255
9.	Operator Specific	101 - 254

Note: - The value 0 is not used here. Error code 0 signifies no error and nothing is written to the card.

Annexure–IV: Data Elements for Pass

Table 19: Data Element for Metro Pass

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Product Type	M	Terminal	Persistent	8
Pass Limit/No. of Trip	M	Terminal	Transient	8
Activation/Start Date and Time	M	Terminal	Transient	24
Expiry Date	M	Terminal	Persistent	16
Valid Route No./Zone	M	Terminal	Persistent	10
Valid Entry Station ID	M	Terminal	Persistent	10
Valid Exit Station ID	M	Terminal	Persistent	10
Bonus Amount/Trip	M	Terminal	Persistent	10
Class/Privileges	M	Terminal	Persistent	2
Daily Limit	M	Terminal	Persistent	4
RFU	-	-	-	18

Table 20: Data Elements for Bus Pass

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Product Type	M	Terminal	Persistent	8
Pass Limit/No. of Trip	M	Terminal	Transient	8
Activation/Start Data and Time	M	Terminal	Transient	24
Expiry Date	M	Terminal	Persistent	16
Valid Route No./Zone	M	Terminal	Persistent	10
Valid Entry Station ID	M	Terminal	Persistent	10
Valid Exit Station ID	M	Terminal	Persistent	10

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Bonus Amount/Trip	M	Terminal	Persistent	10
Class/Privileges	M	Terminal	Persistent	2
Daily Limit	M	Terminal	Persistent	4
RFU	-	-	-	18

Table 21: Data Elements for Parking Pass

Data Element	Mandatory (M)/ Optional (O)	Source	Persistent/ Transient	Size (b)
Product Type	M	Terminal	Persistent	8
Pass Limit/No. of Trip	M	Terminal	Transient	8
Activation/Start Data and Time	M	Terminal	Transient	24
Expiry Date	M	Terminal	Persistent	16
Valid Parking Zone/Station	M	Terminal	Persistent	10
Class/Privileges	M	Terminal	Persistent	2
Daily Limit	M	Terminal	Persistent	4
Vehicle Number	M	Terminal	Persistent	40
RFU	-	-	-	8

Annexure–V: Language Code

Table 22: Language Code

S. No.	Language	Code (5b)
1.	English	00000
2.	Hindi	00001
3.	Bengali	00010
4.	Marathi	00011
5.	Telugu	00100
6.	Tamil	00101
7.	Gujarati	00110
8.	Urdu	00111
9.	Kannada	01000
10.	Odia	01001
11.	Malayalam	01010
12.	Punjabi	01011
13.	Sanskrit	01100
14.	Assamese	01101
15.	Maithili	01110
16.	Santali	01111
17.	Kashmiri	10000
18.	Nepali	10001
19.	Sindhi	10010
20.	Dogri	10011
21.	Konkani	10100
22.	Manipuri	10101
23.	Bodo	10110
24.	RFU	10111 to 11111

Annexure– VI: Examples

Scenario 1

Two tap i.e. first tap at Entry terminal and second tap at Exit terminal (fare debit at Exit say ₹10) using CSA for Single operator 1 in metro with an assumption that card is new which is having Balance (say ₹500)and commuter is going from Station 1 to Station 2.

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date - 01/01/19.
2. No error code present in card validation data elements present in CSA.
3. Txn Status is previous transaction completed i.e. “0000b” in card validation data elements present in CSA.
4. New card, first transaction i.e. all CSA validation and history data elements are 0x00.

Assumptions for terminal data elements

1. Product Type- 0 (E.g. Normal Card)
2. Acquirer ID- 1 (E.g. ACQ 1)
3. Operator ID- 1 (E.g. Operator 1)
4. Entry Station
   a) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1
   b) Txn Seq No. of above Terminal- 0x00, 0x00
5. Exit Station
   a) Terminal ID
      i. Station ID (12b) -2 (E.g. Station 2)

ii. Device Category (6b) - 2 (E.g. Automatic Gate)

iii. Device Number (6b) -2

b) Txn Seq No. of above Terminal- 0x00, 0x00

Steps

1. Card general data element present in CSA before commuter has shown their card to terminal is shown below:

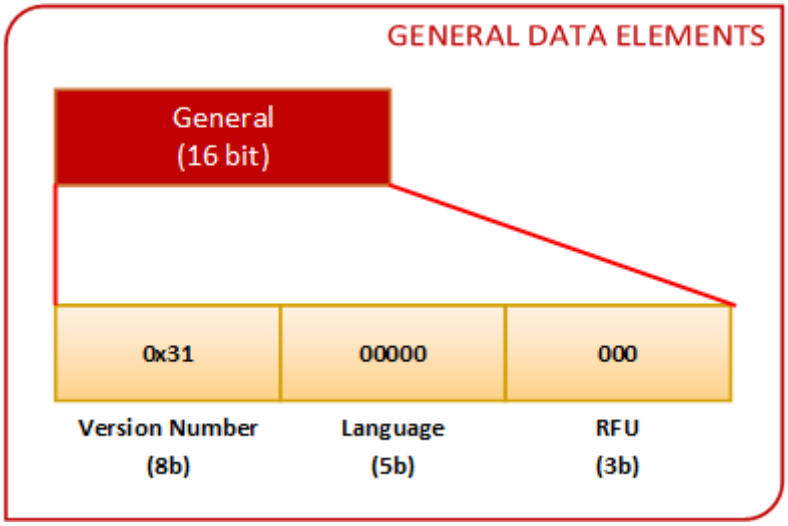


Figure 13: Scenario 1 - CSA General Data Elements

2. Card validation data present in CSA before commuter has shown their card to terminal is shown below:

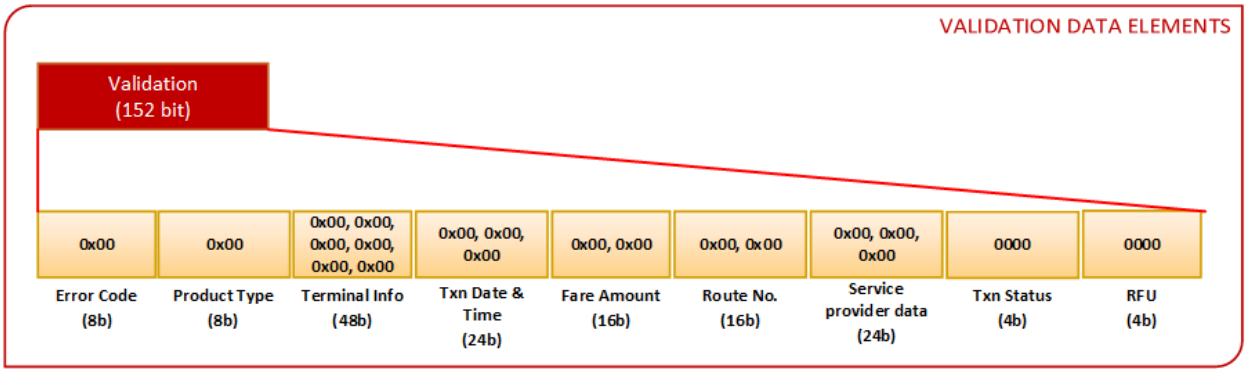


Figure 14: Scenario 1 - CSA Validation Data Elements

3. Card history data present in CSA before commuter has shown their card to terminal is shown below:

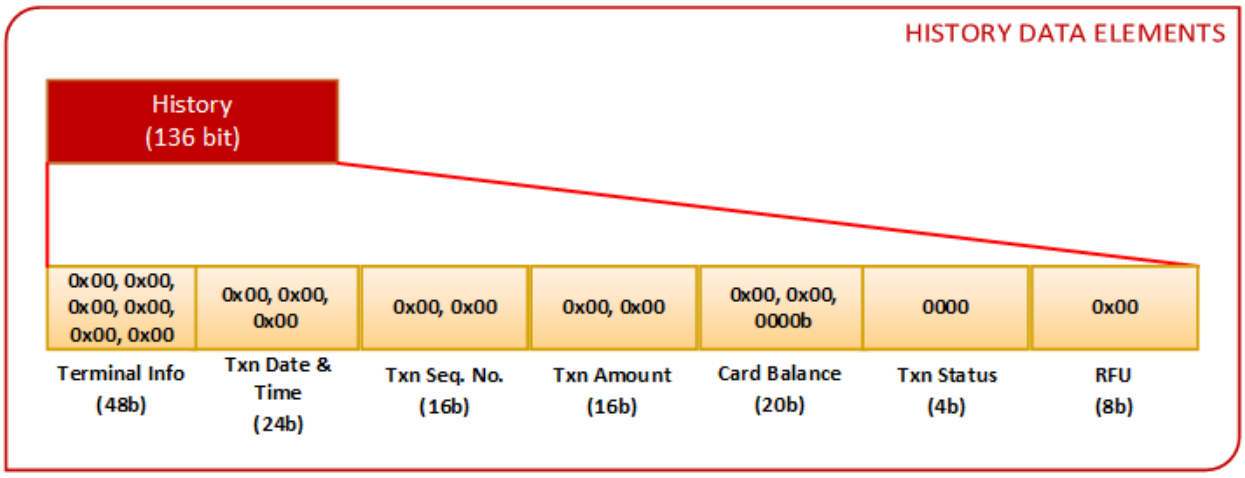


Figure 15: Scenario 1 - History Data Elements before Entry

4. Commuter will place the card on the terminal.

5. At Entry,

a) Terminal will check CSA general data elements:
   i. Version Number (8b) - 00110001b (3.1)
   ii. Language Info (5b) - 00000b (English)

b) Terminal will check CSA validation data elements:
   i. Error Code (8b) -00000000b (No Error)
   ii. Txn Status (4b) - 0000b (Exit Done)

c) Terminal will check the minimum balance present in global wallet as per operator business rule.

d) Terminal will write in CSA validation data elements:
   i. Product Type (8b)- 00000000b (Normal card).
   ii. Terminal Info (48b)-Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b).Terminal ID consists of Station ID (12b), Device Category (6b), and Device Number (6b).

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00010000b,10000001b

Terminal ID - 0x00, 0x10, 0x81

iii. Txn Date & Time (24b) -0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Fare Amount (16b) - As per operator, it will specify the maximum penalty amount in case commuter has escaped the premises of operator without punching at EXIT gate (e.g. Tailgating). Amount will be in units of “tp”.

= 0x01, 0x90 (i.e. 400tp)

v. Route No. (16b) - 0x00, 0x00 (operator specific, assume not used)

vi. Service Provider Data (24b) - 0x00, 0x00, 0x00 (operator specific, assume not used)

vii. Txn Status (4b) - 0001b (Entry)

viii. RFU (4b) - 0000b (Not used)

= 0x10 (Combined Txn status and RFU)

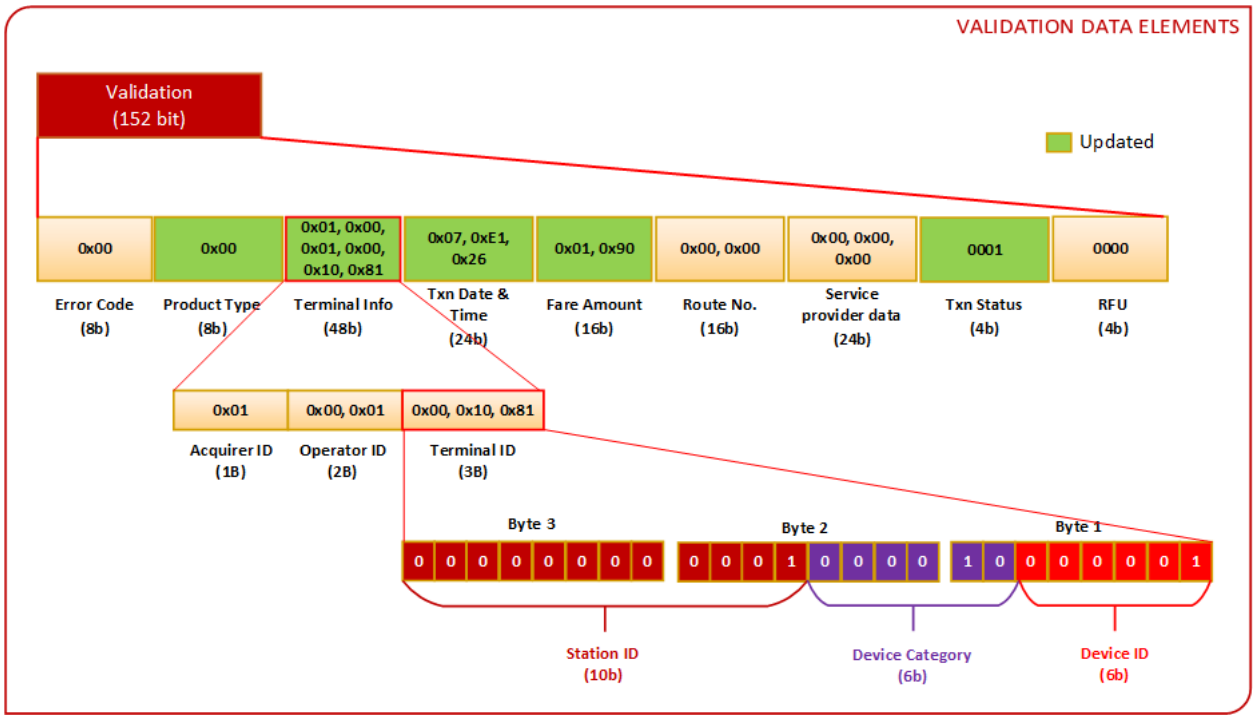


Figure 16: Scenario 1 - CSA Validation Data Elements after Entry

e) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

f) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

g) Terminal will give the command to open the gate as per NCMC specification part 7.

6. At Exit,

a) Terminal will check CSA general data elements:

i. Version Number (8b) - 00110001b (3.1)

ii. Language Info (5b) - 00000b (English)

b) Terminal will check CSA validation data elements in card:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0001b (Entry)

iii. Terminal Info (48b)

1) Acquirer ID (8b) - 0x01

2) Operator ID(16b) - 0x00, 0x01

3) Terminal ID(24b) - 0x00, 0x10, 0x81

c) For Same operator (using operator ID present in card) then, terminal will check:

i. Product Type (8b) - 00000000b (Normal card)

ii. Txn Date & Time (24b) - 0x07,0xE1,0x26 (Assume entry log is of 25/12/19 14:30)

d) **Assume current time at exit terminal is 25/12/19 14:35**

e) Terminal will check the time limit allowed between entry and exit as per operator business rule. (Assuming exit happened within time limit)

f) Terminal will calculate fare (say ₹10) as per operator business rules. Fare is less than the available card balance.

g) Terminal will update the validation data elements.

Txn Status (4b) - 0000b (Exit Done)

h) Terminal will update the history data elements

i. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Station ID (12b), Device Category (6b) and Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b,00100000b, 10000010b

Terminal ID - 0x00, 0x20, 0x82

ii. Txn Date & Time (24b) - 25/12/19 14:35 (0x7E12B)

iii. Txn Seq. No. (16b) - 0x00, 0x01

iv. Txn Amount (16b) -0x00, 0x64 (i.e. 100tp)

v. Card Balance (20b) -0x01, 0x38, 1000b (i.e. 5000tp)

vi. Txn Status (4b) -0000b

i) CSA validation data elements after Exit is shown below:

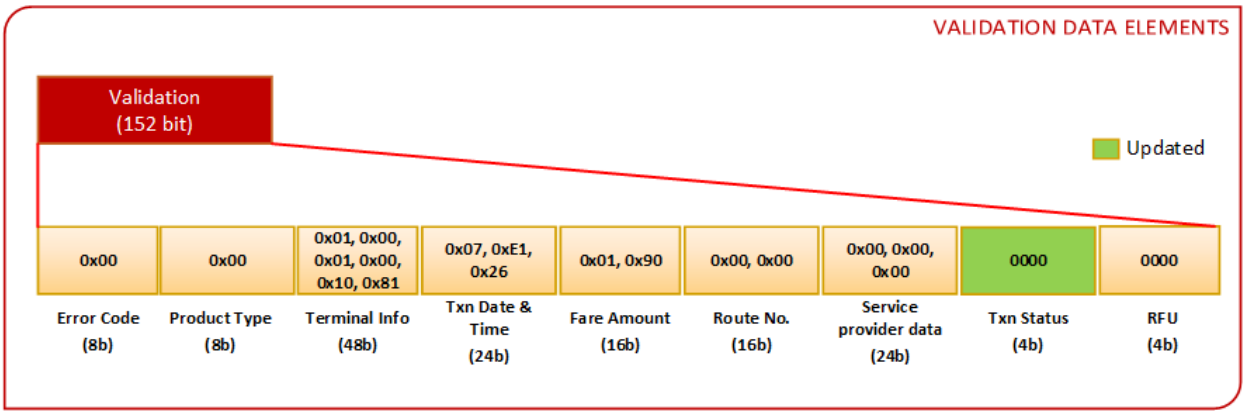


Figure 17: Scenario 1 - CSA Validation Data Elements after Exit

j) CSA History data elements after Exit is shown below:

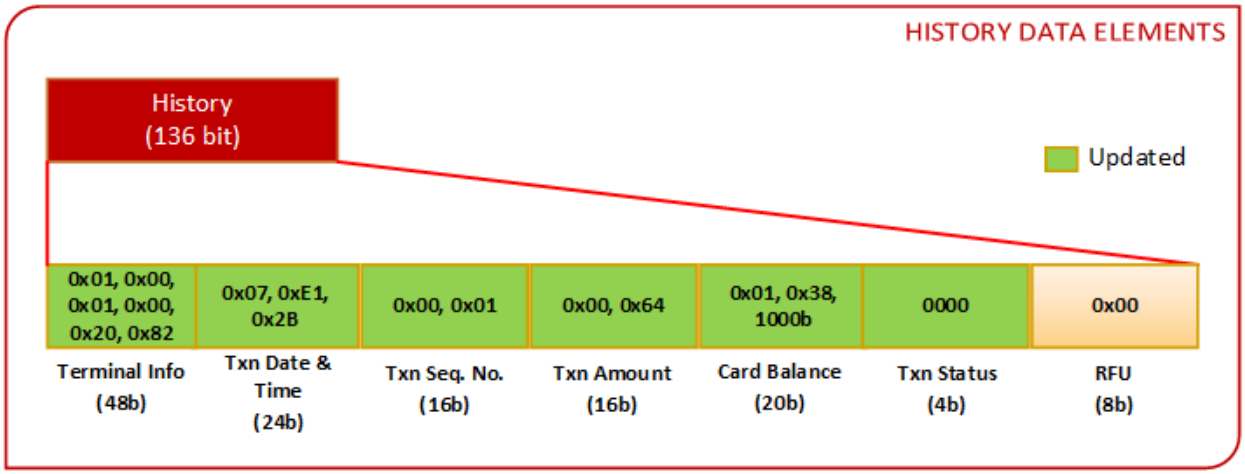


Figure 18: Scenario 1 - CSA History Data Elements after Exit

k) Terminal will generate the transit file and send it to AFC system as per NCMC Specification part 5.

l) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

m) Terminal will give the command to open the gate as per NCMC specification part 7.

Scenario 2

Two Tap occurs when a commuter places his card at some operator's Entry terminal e.g. 1 but escapes the premises without showing his card at any Exit terminal (i.e. tailgating). Then next time he enters the same operator's premises and shows his card again at an Entry terminal e.g. 2. We assume that card has sufficient balance (say ₹500) and a penalty of ₹40 will be debited by customer care.

The two taps are – first tap at Entry terminal 1(operator 1) and second tap at Entry terminal 2 (operator 1) using CSA. Terminal and operator numbers are only for ease of illustration.

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date - 01/01/19.
2. No error code present in card validation data elements present in CSA.
3. Txn status is showing Entry i.e. "0001b" in CSA card validation data elements set by operator 1 Entry terminal 1.
   i. Txn Date & Time (24b) - 0x07,0xE1,0x26 (Assume entry log is of 25/12/19 14:30)

Assumptions for terminal data elements

1. Product Type -0 (E.g. Normal Card)
2. Acquirer ID - 1 (E.g. ACQ 1)
3. Operator ID - 1 (E.g. operator 1)
4. Entry Station
   a) Terminal ID (Entry Terminal 1)
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)

iii. Device Number (6b) - 1

b) Txn Seq No. - 0x00, 0x00

5. Entry Station
   a) Terminal ID(Entry Terminal 2)
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) -2 (E.g. Automatic Gate)
      iii. Device Number (6b) -2
   b) Txn Seq No. - 0x00, 0x00
6. Customer Care Terminal at operator 1
   a) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) -3 (E.g. Customer Care)
      iii. Device Number (6b) - 1
   b) Txn Seq No. - 0x00, 0x00

Steps at Entry terminal 2 (After Tailgating)

1. Card general data element present in CSA before commuter has shown his card to terminal (operator 1) is shown below:

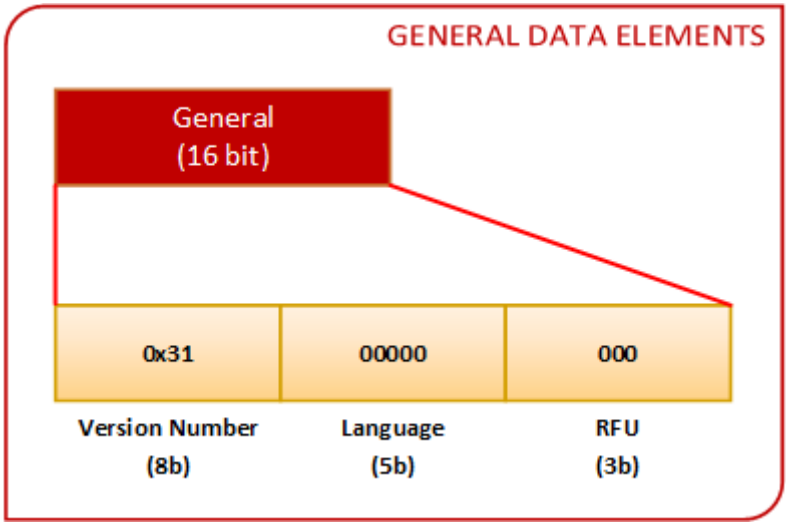


Figure 19: Scenario 2 - CSA General Data Elements

2. Card Validation data elements in CSA before commuter has shown their card to terminal (operator1) is shown below:

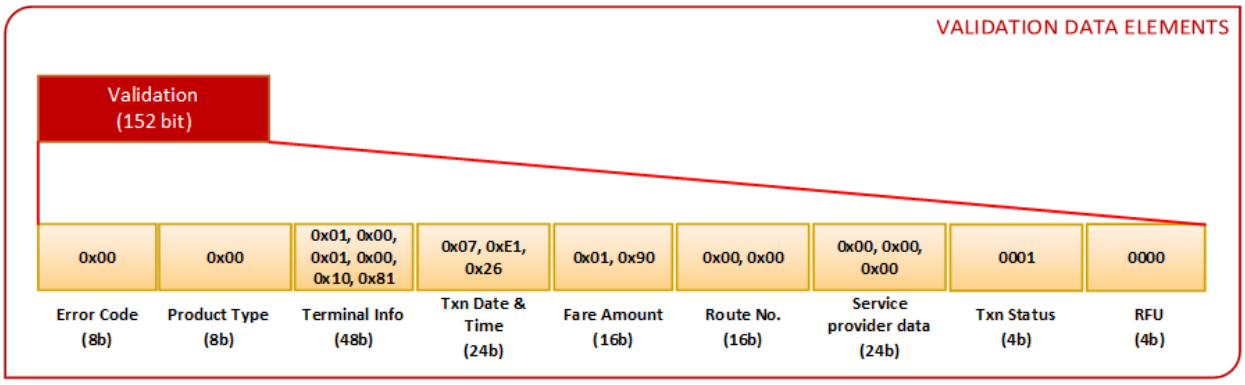


Figure 20: Scenario 2 - CSA Validation Data Elements before Entry

3. Card history data elements in CSA before commuter has shown their card to terminal (operator 1) is shown below:

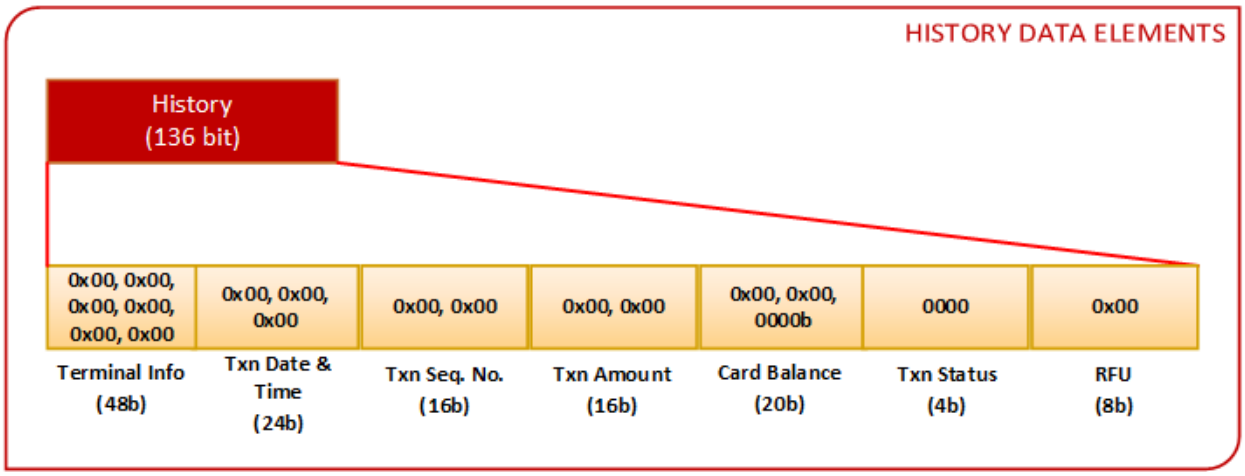


Figure 21: Scenario 2 - History Data Elements before Entry

4. Commuter will place their card on the Entry terminal2 at operator 1.
5. **At Entry,**
   a) Terminal will check CSA general data elements:
      i. Version Number (8b) - 00110001b (3.1)
      ii. Language Info (5b) - 00000b (English)
   b) Terminal will check CSA validation data elements:
      i. Error code (8b) -00000000b (No Error)
      ii. Txn Status (4b) - 0001b (Entry)

c) Since commuter has not done proper Exit during previous journey and hence not paid the fare to operator 1, next time when commuter approaches again then, Entry terminal at operator 1 will write:

i. Error Code (8b) - 00000100b i.e. 4 (Journey not completed i.e. Exit not found in CSA validation area)

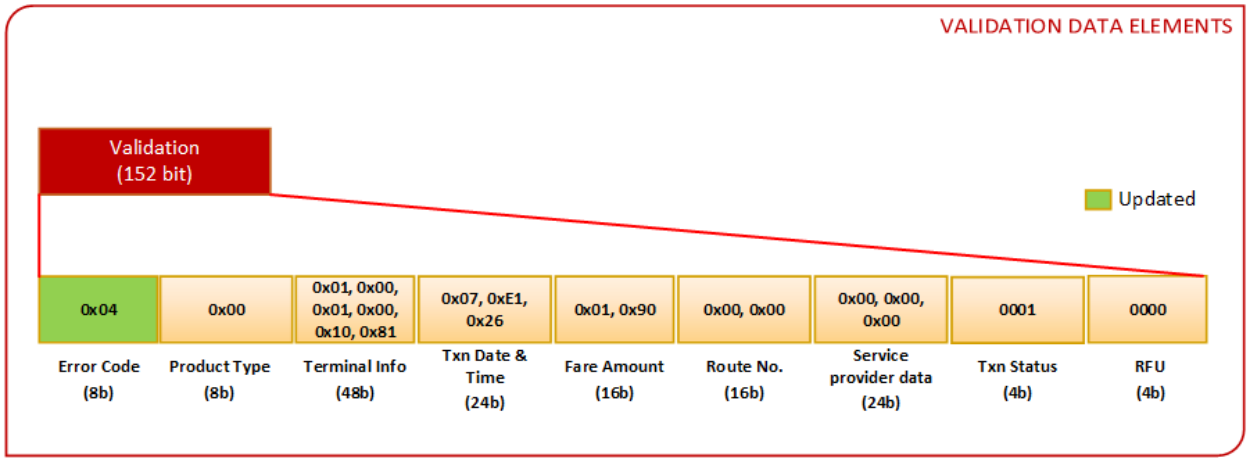


Figure 22: Scenario 2 - CSA Validation Data Elements at Entry

d) Prompts the commuter to approach customer care.

6. Commuter will go to customer care and then customer care will clear the “Error Code” after debiting the amount (say ₹40) as per business rule of operator 1.

7. CSA validation data element after removing the error code by customer care at operator 1 is shown below:

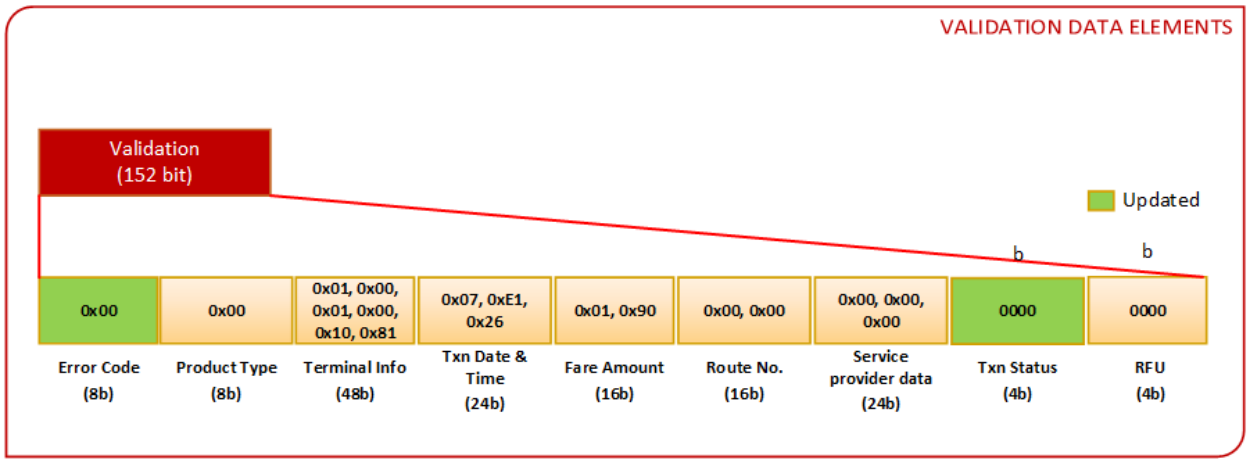


Figure 23: Scenario 2 - CSA Validation Data Elements at Customer Care

8. CSA validation date element after Customer care debit the penalty is shown below:
   i. Txn Date & Time (24b) - 0x07,0xF9,0xE0 (Assume entry log is of 30/12/19 14:00)

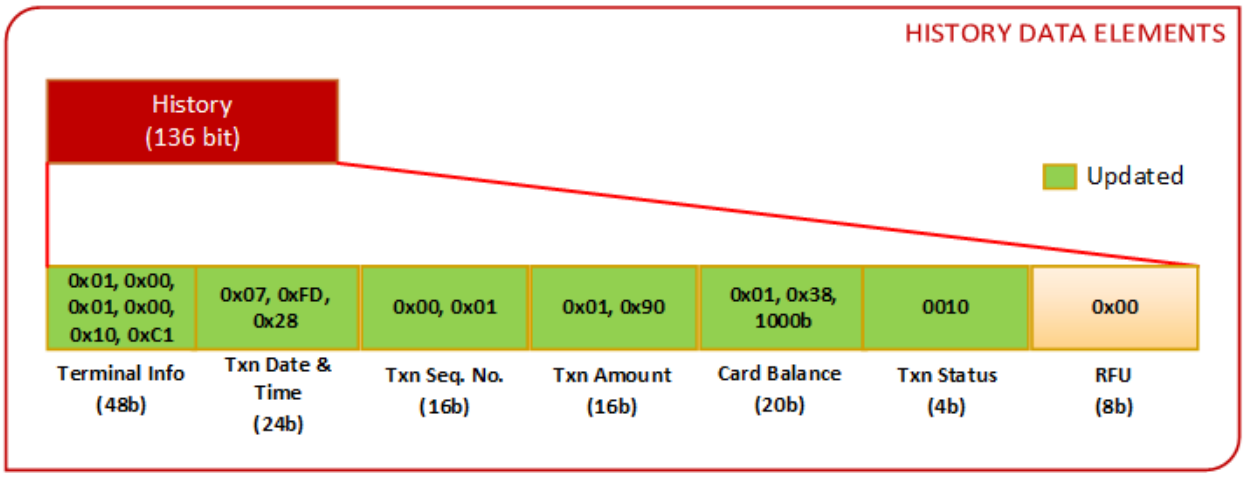


Figure 24: Scenario 2 - CSA History Data Elements at Customer Care

9. Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.
10. Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

Scenario 3

Two Tap i.e. Commuter has placed the card at operator 1 Entry terminal and escaped the premises of operator 1 without showing their card at operator 1 Exit terminal (i.e. tailgating) and approaches to another operator e.g. operator 2 with an assumption that card has Balance (say ₹500) and penalty (say ₹40) will be debited by customer care. i.e. first tap at Entry terminal (operator 1) and second tap at Entry terminal (operator 2) using CSA.

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date - 01/01/19.
2. No error code present in card validation data elements present in CSA.
3. Txn status is showing Entry i.e. “0001b” in CSA card validation data elements set by operator 1 Entry terminal.
   i. Txn Date & Time (24b) - 0x07,0xE1,0x26 (Assume entry log is of 25/12/19 14:30)

Assumptions for terminal data elements

1. Product Type -0 (E.g. Normal Card)
2. Acquirer ID - 1 (E.g. ACQ 1)
3. Entry Terminal at operator 1
   a) Operator ID - 1 (E.g. operator 1)
   b) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1
   c) Txn Seq No. - 0x00, 0x00

4. Entry Terminal at operator 2
   a) Operator ID - 2 (E.g. operator 2)
   b) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 2)
      ii. Device Category (6b) -1 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1
   c) Txn Seq No. - 0x00, 0x00
5. Customer Care Terminal at operator 2
   a) Operator ID - 2 (E.g. operator 2)
   b) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 2)
      ii. Device Category (6b) -3 (E.g. Customer Care)
      iii. Device Number (6b) - 1
   c) Txn Seq No. - 0x00, 0x00

Steps at Entry terminal operator 2 (After Tailgating from operator 1)

1. Card general data element present in CSA before commuter has shown their card to terminal (operator 2) is shown below:

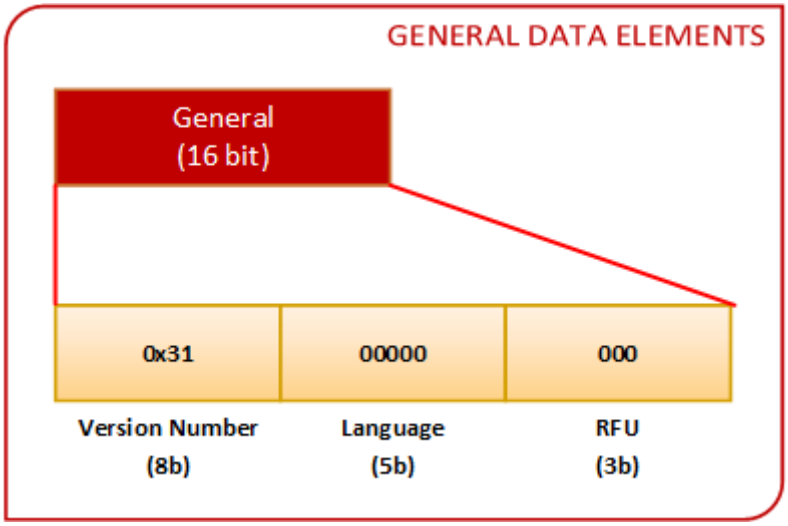


Figure 25: Scenario 3 - CSA General Data Elements

2. Card validation data elements in CSA before commuter has shown their card to terminal (operator 2) is shown below:

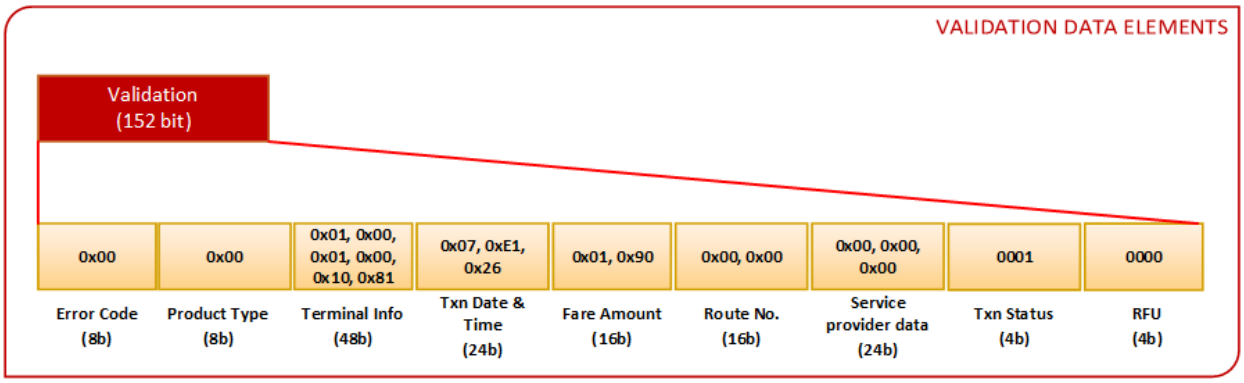


Figure 26: Scenario 3 - CSA Validation Data Elements before Entry

3. Card history data elements in CSA before entry at operator 2 as shown below:

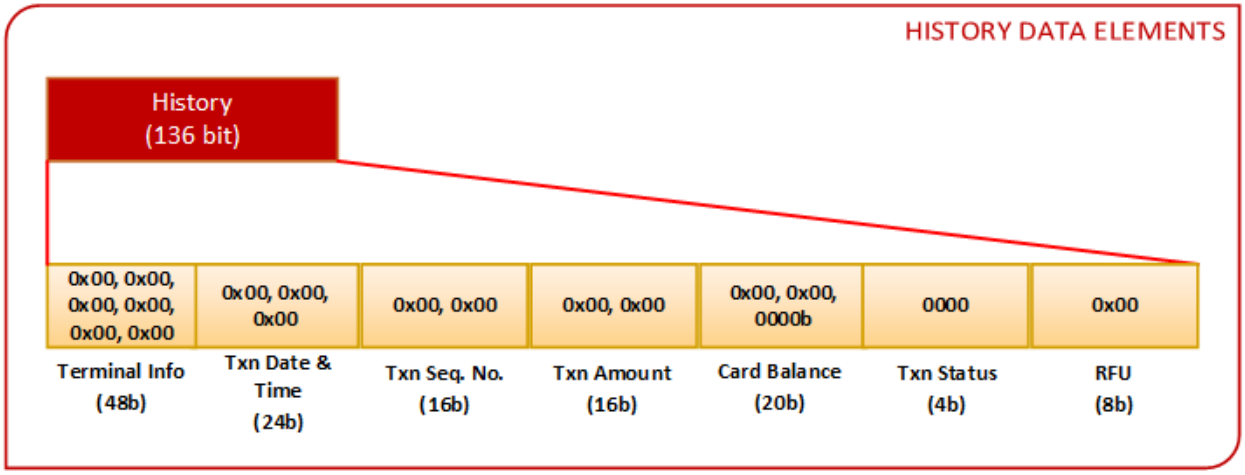


Figure 27: Scenario 3 - CSA History Data Elements before Entry

4. Commuter will place their card on the Entry terminal at operator 2.
5. **At Entry (operator 2),**
   a) Terminal will check CSA general data elements:
      i. Version Number (8b) - 00110001b (3.1)
      ii. Language Info (5b) - 00000b (English)
   b) Terminal will check CSA validation data elements:
      i. ErrorCode (8b) -00000000b (No Error)
      ii. Txn Status (4b) - 0001b (Entry)
   c) Since commuter did not do a proper Exit during previous journey(and hence did not pay the fare to operator 1), Terminal at operator 2 will write the error code as follows:

i. Error Code (8b) - 00000100b i.e. 4 (Journey not completed i.e. Exit not found in CSA validation area)

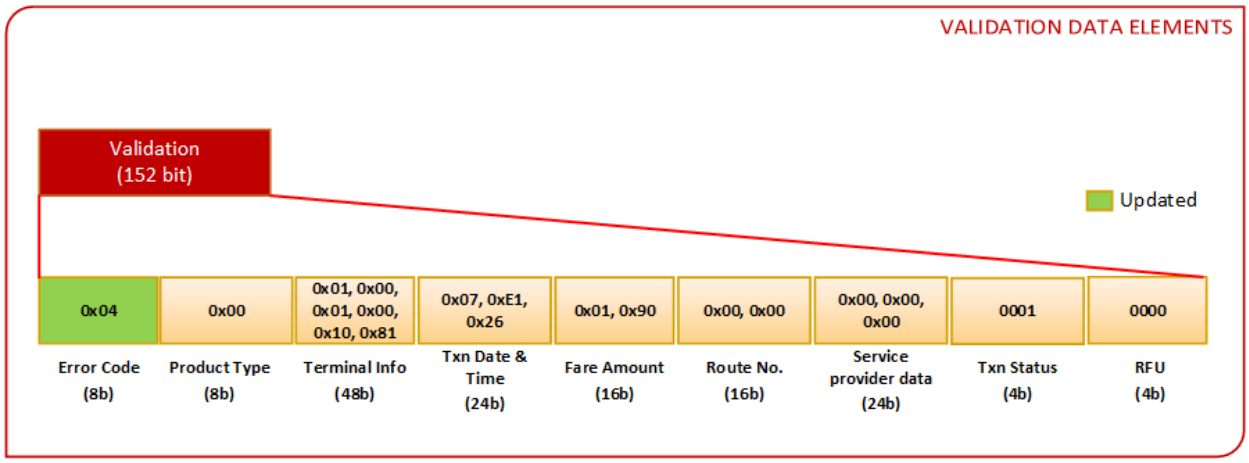


Figure 28: Scenario 3 - CSA Validation Data Elements at Entry

d) Prompts the commuter to approach customer care.

6. Commuter will go to customer care and then customer care will clear the “Error Code” after debiting the amount mentioned in “Fare Amount” field present in CSA validation data elements.
7. CSA validation data elements after removing error code by customer care at operator 2 is shown below:

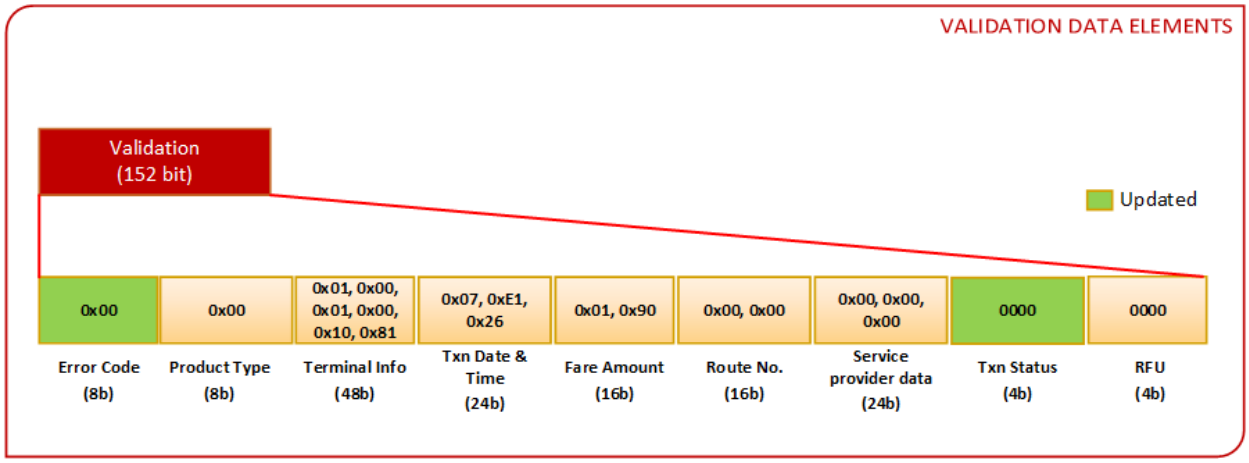


Figure 29: Scenario 3 - CSA Validation Data Elements at Customer Care

8. CSA history data elements after debit penalty amount (say ₹40) is shown below:
   i. Txn Date & Time (24b) - 0x07,0xF9,0xE0 (Assume entry log is of 30/12/19 14:00)

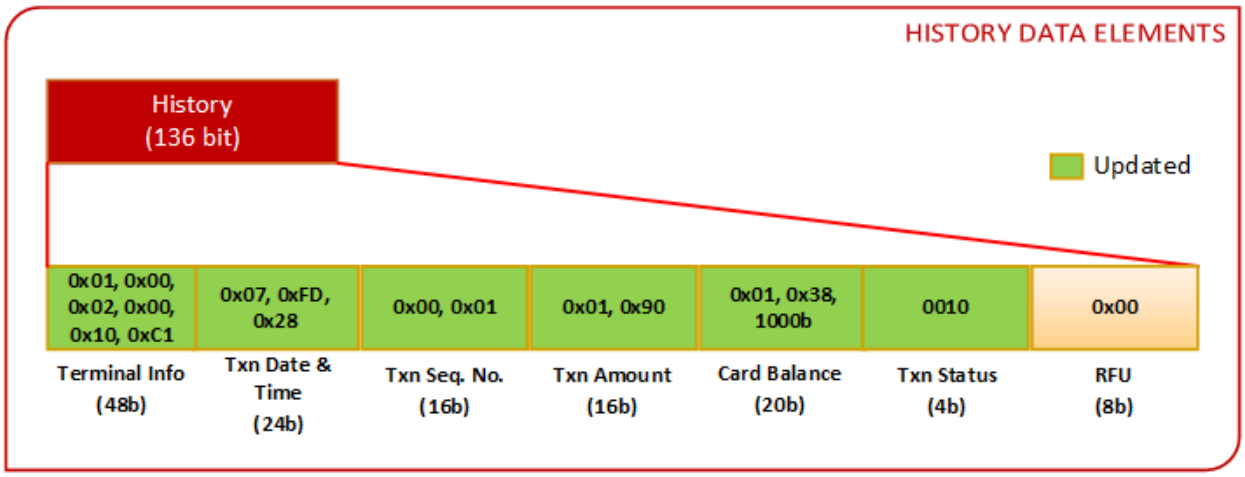


Figure 30: Scenario 3 - CSA History Data Elements at Customer Care

9. Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.
10. Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

Scenario 4

Multiple Tap in single operator i.e. there will be multiple Entry Exit gates and commuter will get the discount (if any) as per operator business rule if commuter is continuing their journey and placed their card at another Entry gate (after completion of previous journey) within certain time limit in the single operator. Fare will be debited at every Exit gate. i.e. First tap at Entry terminal (operator 1), second tap at Exit terminal (operator 1), third tap at Entry terminal (operator 1) and last tap at Exit terminal (operator 1) using CSA.

Flow: - Entry 1 (14:30) -> Exit 1 (14:35) -> Entry 2 (14:40) -> Exit 2 (14:45)

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date -01/01/19.
2. No error code present in card validation data elements present in CSA.
3. Txn status is previous transaction completed i.e. “0000b” in card validation data elements present in CSA.
4. New card, first transaction i.e. all CSA validation and history data elements are 0x00.

Assumptions for terminal data elements

1. Product Type -0 (E.g. Normal card)
2. Acquirer ID - 1 (E.g. ACQ 1)
3. Operator ID - 1 (E.g. operator 1)
4. Entry Station 1
   a) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1

b) Txn Seq No. of above Terminal- 0x00, 0x00

5. Exit Station 1
   a) Terminal ID
      i. Station ID (12b) -2 (E.g. Station 2)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) -2
   b) Txn Seq No. of above Terminal - 0x00, 0x00
6. Entry Station 2
   a) Terminal ID
      i. Station ID (12b) -2 (E.g. Station 2)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) -3
   b) Txn Seq No. of above Terminal - 0x00, 0x00
7. Exit Station 2
   a) Terminal ID
      i. Station ID (12b) -3 (E.g. Station 3)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) -4
   b) Txn Seq No. of above Terminal - 0x00, 0x00

Steps

1. Card general data element present in CSA before commuter has shown his card to terminal is shown below:

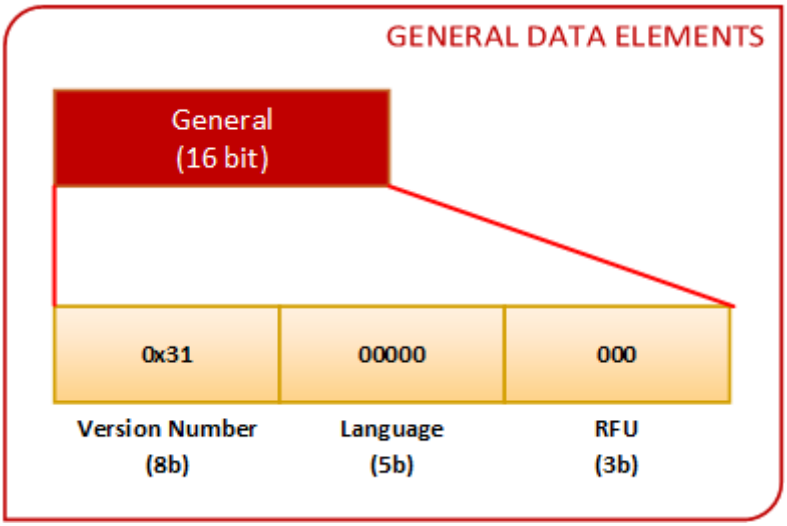


Figure 31: Scenario 4 - CSA General Data Elements

2. Validation date elements in CSA before commuter has shown his card on terminal is shown below:

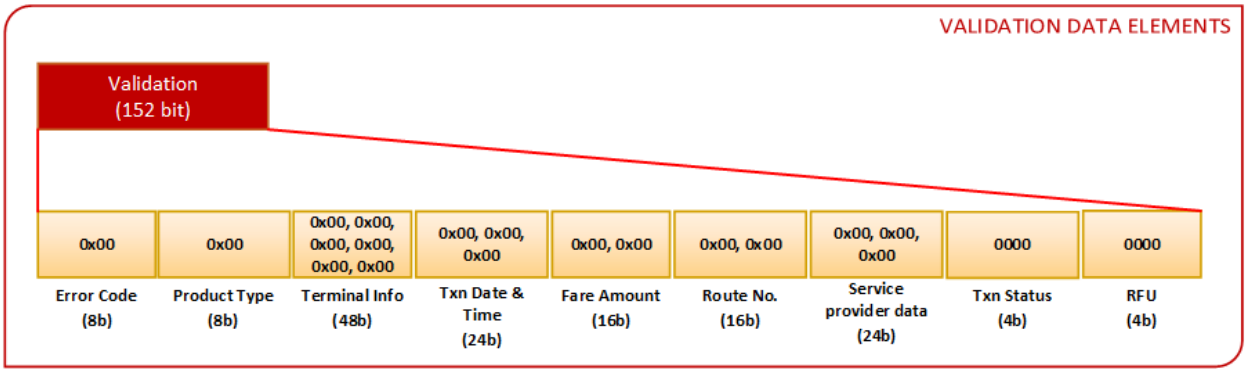


Figure 32: Scenario 4 - CSA Validation Data Elements before Entry

3. Commuter will place the card on the Entry terminal.
4. **At Entry 1 (Automatic Gate),**

a) Terminal will check CSA general data elements:

   i. Version Number (8b) - 00110001b (3.1)

   ii. Language Info (5b)- 00000b (English)

b) Terminal will check CSA validation data elements:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0000b (Exit Done)or 0011b (One Tap)

c) Terminal will write in CSA validation data elements:

i. Product Type (8b) - 00000000b (Normal Card)

ii. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Station ID (12b), Device Category (6b) and Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00010000b,10000001b

Terminal ID - 0x00, 0x10, 0x81

iii. Txn Date & Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Fare Amount (16b) - As per operator, it will specify the maximum amount in case commuter has escaped the premises of operator without punching at EXIT gate (e.g. Tailgating)

= 0x01, 0x90 (i.e. 400tp)

v. Route No. (16b) - 0x00, 0x00 (operator specific, assume not used)

vi. Service provider data (24b) - 0x00, 0x00, 0x00 (operator specific)

vii. Txn Status (4b) - 0001b (Entry)

viii. RFU (4b) - 0000b (Not used)

= 0x10 (combined Txn status and RFU)

d) CSA validation data elements after Entry is shown below:

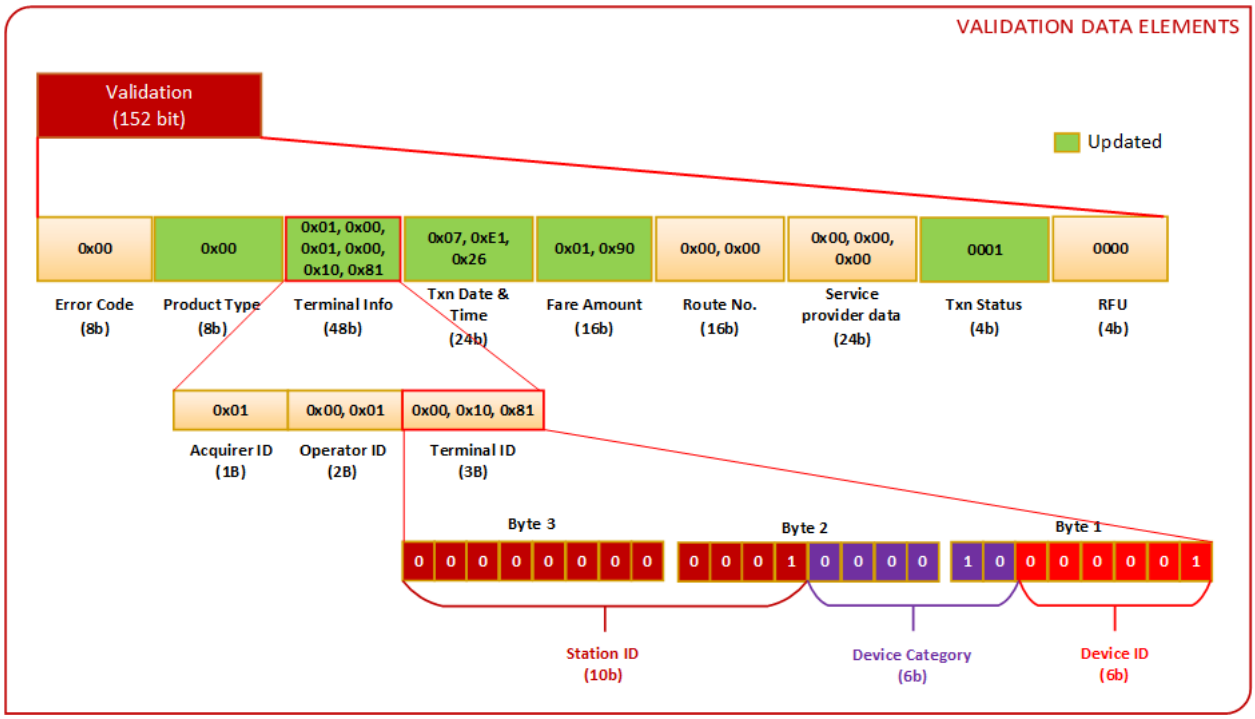


Figure 33: Scenario 4 - CSA Validation Data Elements after Entry

e) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

f) Terminal will generate the financial file and send and send it to acquirer as per NCMC specification part 6.

g) Terminal will give the command to open the gate as per NCMC specification part 7.

5. At Interchange Gate (Exit 1),

Steps

1. Validation data elements in CSA before commuter has shown their card on Exit 1 gate is shown below:

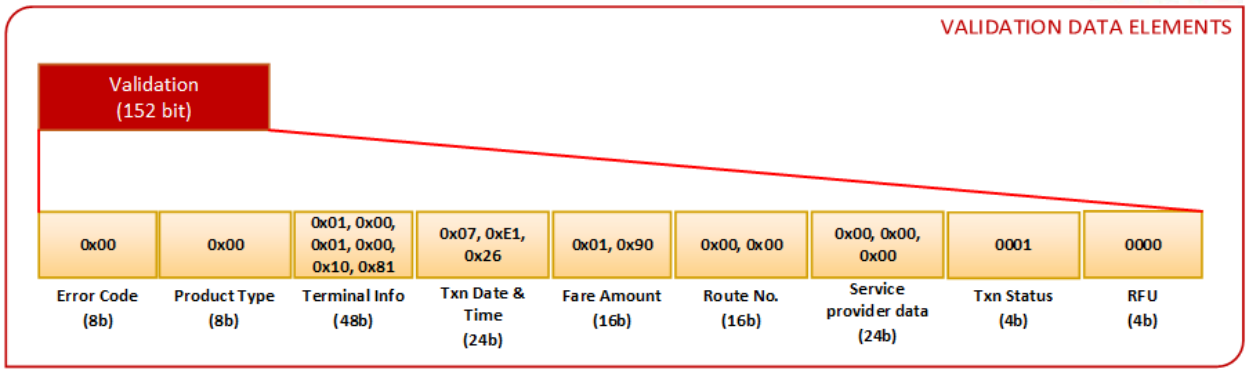


Figure 34: Scenario 4 - CSA Validation Data Elements at Exit 1

2. **Commuter will place the card on the Exit 1 gate**

   a) Terminal will check CSA general data elements:

      i. Version Number (8b) - 00110001b (3.1)

      ii. Language Info (5b) - 00000b (English)

   b) Terminal will check CSA validation data elements:

      i. Error code (8b) - 00000000b (No Error)

      ii. Txn Status (4b) - 0001b (Entry)

   c) Terminal will write in CSA validation data elements:

      i. Service Provider Data (24b) - It consist of Interchange gate ID. (15b), Time in minutes (9b).

      Interchange Gate ID - 1 (Exit 1)

      Time at Exit 1 - 14:35

      Store difference between Exit 1 date & time and Entry 1 date & time in minutes as per BR. i.e. 516395 – 516390 = 5

2 Byte	1 Byte
00000000, 00000010	00000101

      Service Provider Data - 0x00, 0x02, 0x05

   d) Card CSA validation data elements after commuter has crossed the Exit 1 is shown below:

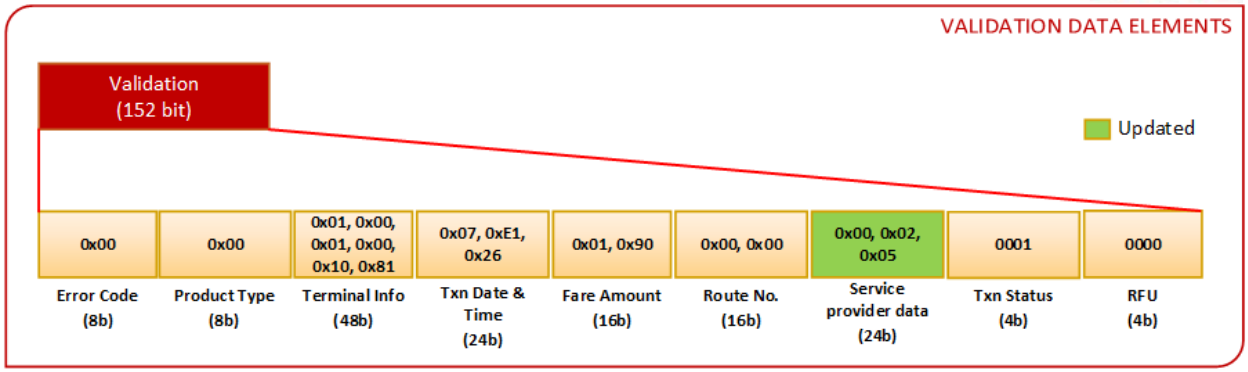


Figure 35: Scenario 4 - CSA Validation Data Elements after Exit 1

e) Terminal will update the history data elements

i. Terminal info (48b) -Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID (24b) consists of Station ID (12b), Device Category (6b), Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00100000b, 10000010b

Terminal ID - 0x00, 0x20, 0x82

ii. Txn Date & Time (24b) - 25/12/19 14:35 (0x7E12B)

iii. Txn Seq. No. (16b) - 0x00, 0x01

iv. Txn Amount (16b) - 0x00, 0x64 (i.e. 100tp)

v. Card Balance (20b) - 0x01, 0x38, 1000b (i.e. 5000tp)

vi. Txn Status (4b) - 0000b

f) CSA history data elements after Exit is shown below:

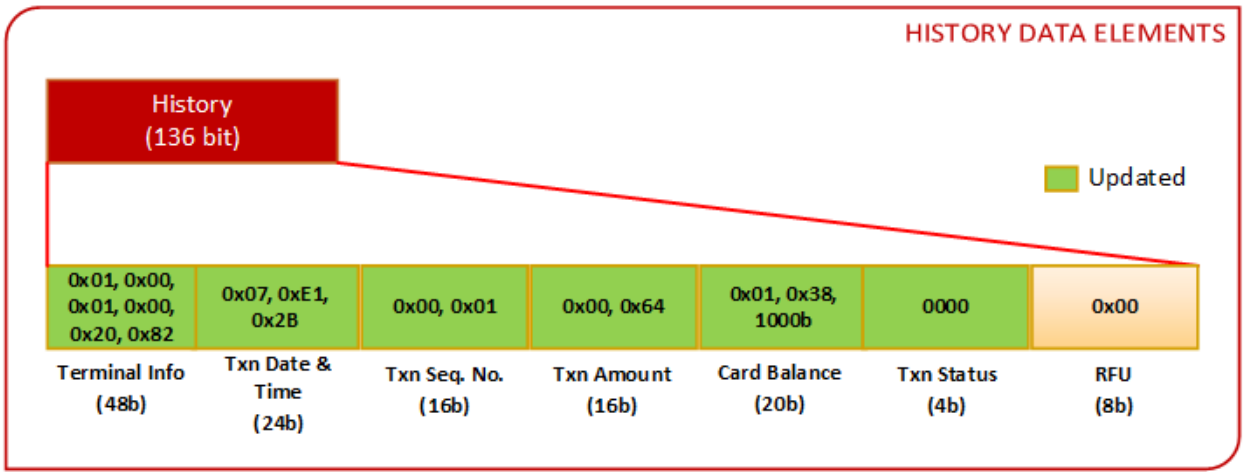


Figure 36: Scenario 4 - CSA Validation Data Elements after Exit

g) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

h) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

i) Terminal will give the command to open the gate as per NCMC specification part 7.

6. At Interchange Gate (Entry 2),

Steps

a) Terminal will check CSA general data elements:

i. Version Number (8b) - 00110001b (3.1)

ii. Language Info (5b) - 00000b (English)

b) Terminal will check CSA validation data elements:

i. Error code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0000b (Exit Done)

c) Terminal will write in CSA validation data elements without overwriting the existing information regarding Interchange gate no (Exit 1):

i. Service Provider Data (24b) - It consist of Interchange gate no. (15b), Time in minutes (9b)

Interchange Gate ID - 2 (Entry 2)

Time at Entry 2 - 14:40

Store difference between Entry 2 date & time and Entry 1 date & time in minutes as

per BR. i.e. 516400 – 516390 = 10

Note: Don't change the IG (Exit 1) gate ID which is set by IG (Exit 1).

2 Byte	1Byte
00000000, 00000110	00001010

Service Provider Data - 0x00, 0x06, 0x0A

d) Card CSA validation data elements after commuter has crossed the Entry 2 is shown below:

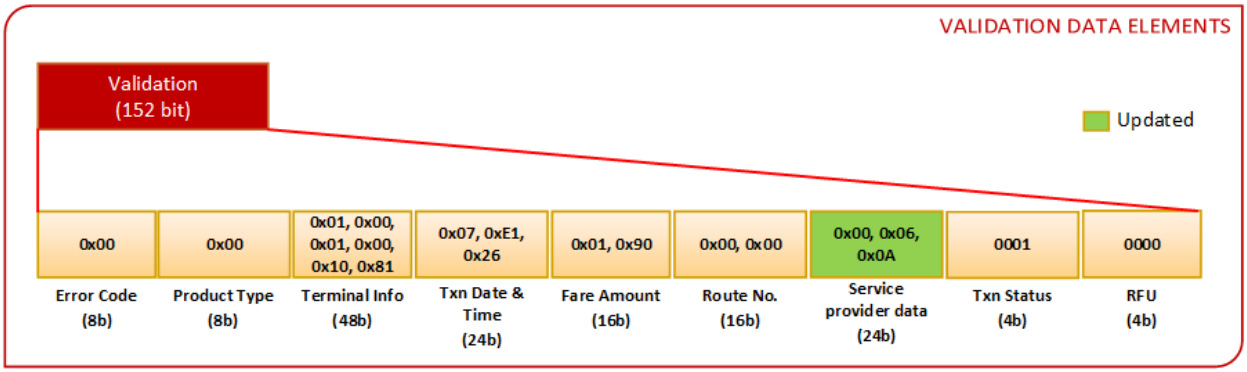


Figure 37: Scenario 4 - CSA Validation Data Elements after Entry 2

7. At Exit 2 (Automatic Gate),

a) Terminal will check CSA general data elements:

i. Version Number (8b) - 00110001b (3.1)

ii. Language Info (5b) - 00000b (English)

b) Terminal will check CSA validation data elements in card:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0001b (Entry)

iii. Terminal Info (48b) -

1) Acquirer ID - 0x01

2) Operator ID- 0x00, 0x01

3) Terminal ID - 0x00, 0x10, 0x81

c) For Same operator (using operator ID present in card) then, terminal will check:

i. Product Type (8b) - 00000000b (Normal card)

ii. Txn Date & Time (24b) - 0x07,0xE1,0x26 (Assume entry log is of 25/12/19 14:30)

d) **Assume current time at exit terminal is 25/12/19 14:45**

e) Terminal will check the time limit allowed between entry and exit as per operator business rule.

f) Terminal will calculate fare as per operator business rule (i.e. it may calculate on basis route no., service provider data i.e. Exit 1&Entry 2)

g) Terminal will update the validation data elements

Txn Status (4b) - 0000b

h) CSA validation data after Exit is shown below:

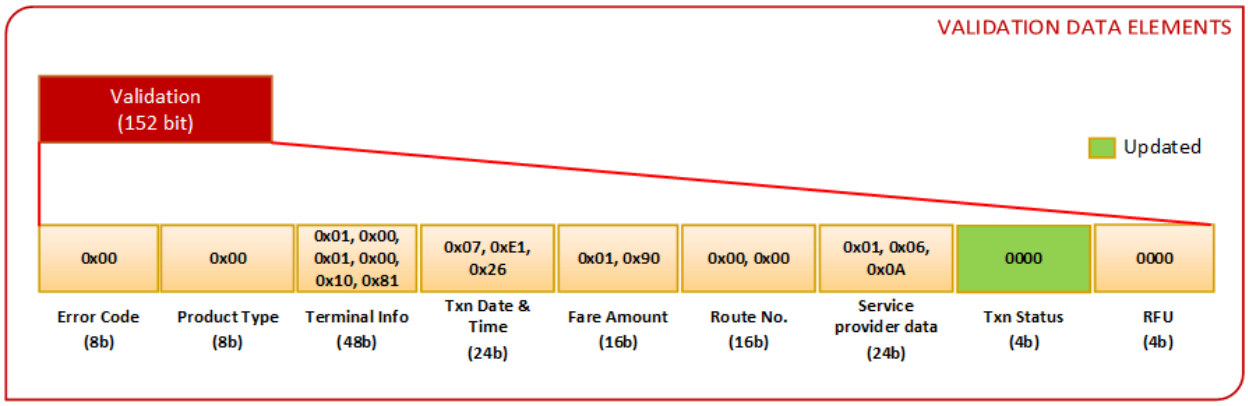


Figure 38: Scenario 4 - CSA Validation Data Elements after Exit

i) Terminal will update the history data elements

vii. Terminal info (48b) -Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID (24b) consists of Station ID (12b), Device Category (6b), Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00110000b, 10000100b

Terminal ID - 0x00, 0x30, 0x84

viii. Txn Date & Time (24b) - 25/12/19 14:45 (0x7E135)

ix. Txn Seq. No. (16b) - 0x00, 0x01

x. Txn Amount (16b) - 0x00, 0x64 (i.e. 100tp)

xi. Card Balance (20b) -0x01, 0x38, 1000b (i.e. 5000tp)

xii. Txn Status (4b) -0000b

j) CSA history data elements after Exit is shown below:

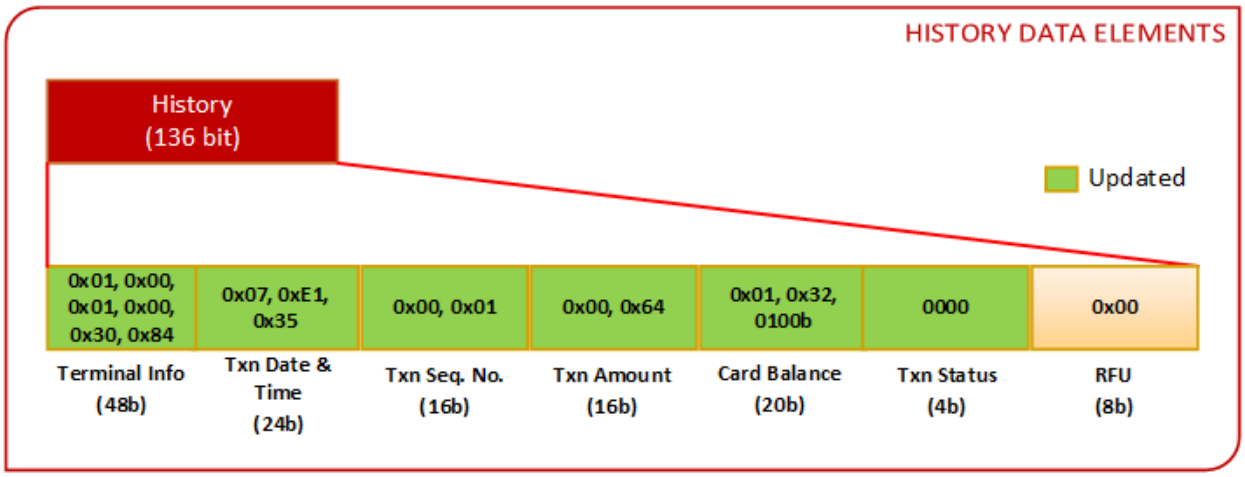


Figure 39: Scenario 4 - CSA Validation Data Elements after Exit

k) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

l) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

m) Terminal will give the command to open the gate as per NCMC specification part 7.

Scenario 5

In bus, single tap where money (fixed fare) will be debited at the time of Entry using CSA with an assumption that card is new and having card balance (say₹200), bus is going from Source Bus Stop 1 to Destination Bus Stop 10 and commuter is going from Bus Stop 2 to Bus Stop 9 of fixed fare (say ₹10).

The terminal is of Mobile Validator (Single Tap / Fixed Fare) Type. Here the commuter taps the card once upon entry and a fixed fare is deducted irrespective of source and destination stop.

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date -15/02/19.
2. No error code present in card validation data elements present in CSA.
3. Txn status is previous transaction completed i.e. “0000” in card validation data elements present in CSA.
4. New card, first transaction i.e. all CSA validation and history data elements are “00”.

Assumptions for operator specific data elements

1. Route is indicated by a 4-digit number; route number chosen for this application is ‘1’.
2. Bus stops are referenced in card by a route specific bus stop index. The index is fixed with the ‘up’ direction of the route and doesn’t change for the return trip.
3. In the illustration considered below, it is assumed that the route ‘1’ has a total of 10 stops and the commuter boards the bus at stop indexed 2 and alights at stop indexed 9.

Assumptions for terminal data elements

1. Product type - 0 (E.g. Normal Card)
2. Acquirer ID- 1 (E.g. ACQ 1)
3. Operator ID- 2 (E.g. operator 2)
4. In Bus, the following representative nomenclature is used:-
   a) Terminal ID
      i. Device registration bus depot ID (5b) - 1 (00001b) (E.g. Bus Depot1).
      ii. Device Category (3b) - 4 (100b) (Mobile Terminal Fixed Fare, refer PART V)
      iii. Device Unique Number (16b) - 1 (0x0001)
   b) Txn Seq No. of above Terminal- 0x00, 0x00

Steps

1. Card general data present in CSA before commuter has shown their card on terminal is shown below.

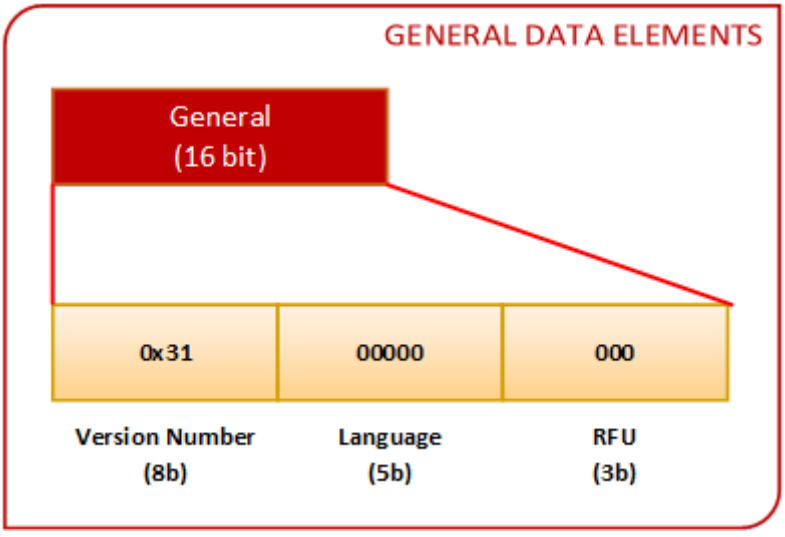


Figure 40: Scenario 5 - CSA General Data Elements before Entry

2. Card validation data present in CSA before commuter has shown their card on terminal is shown below:

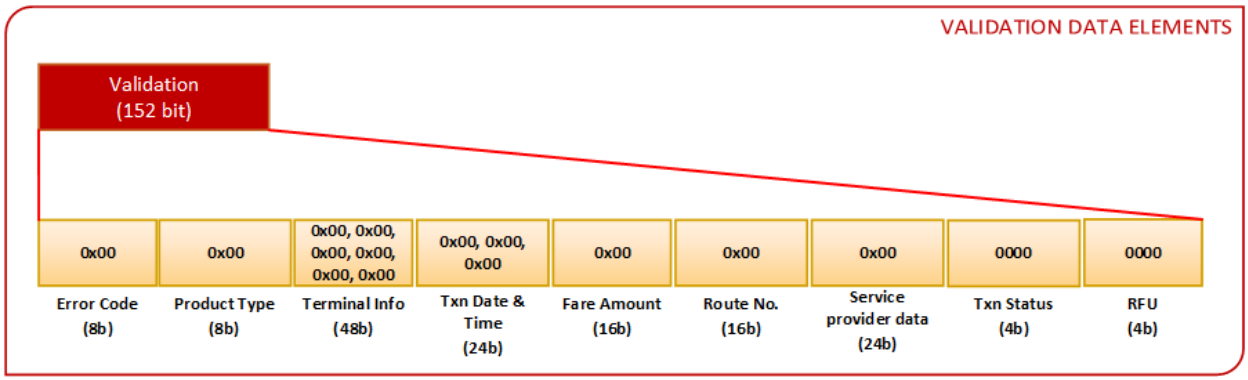


Figure 41: Scenario 5 - CSA Validation Data Elements before Entry

3. Card history data element in CSA before commuter has shown their card on terminal is shown below:

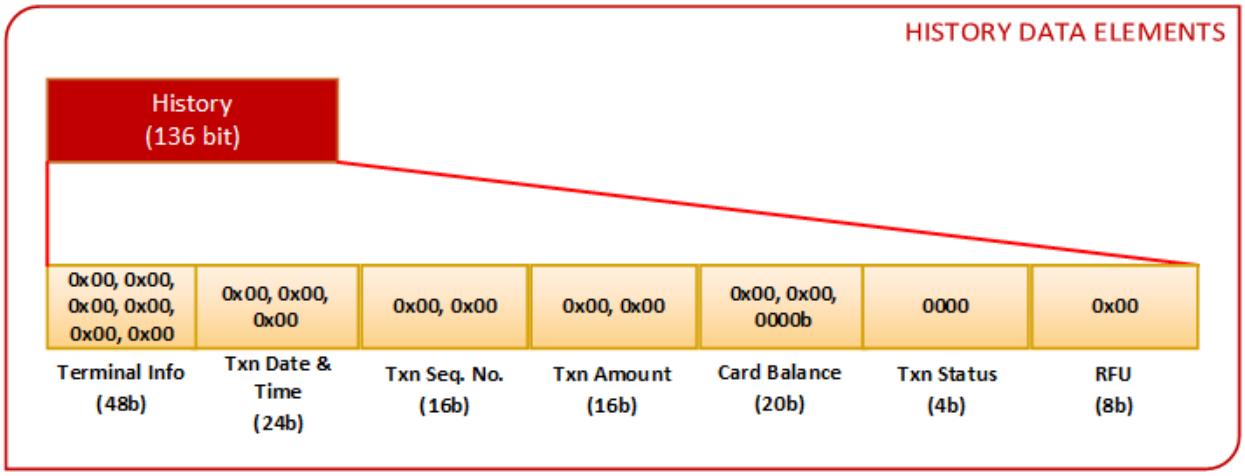


Figure 42: Scenario 5 - History Data Elements before Entry

4. Commuter will place the card on the terminal.
5. Following operations will be performed by terminal
   a) Terminal will check CSA general data elements:
      i. Version Number (8b) - 00110001b (3.1)
      ii. Language Info (5b) - 00000b (English)
   b) Terminal will check CSA validation data elements:
      i. Error code (8b) - 00000000b (No Error)
      ii. Txn Status (4b) - 0000b (Exit Done) or 0011b (One Tap i.e. may be present when card is not new)
   c) Terminal will write in CSA validation data elements:

i. Product Type (8b) - 00000000b (Normal Card)

ii. Txn Status (4b) - 0011b (One Tap/Ticket)

iii. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Depot ID (5b), Device Category(3b), Device Unique Number (16b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000010b	000001100b,00000000b, 00000001b

Terminal ID - 0x0C, 0x00, 0x01

iv. Txn Date & Time (24b) -0x08, 0x5C, 0xE6 (01/03/20 14:30).

v. Fare Amount (16b) - As per operator, it will specify the amount commuter has paid in Bus = 0x64 (i.e. 100tp)

vi. Route No. (16b) - 0x00, 0x01 (operator specific, assume Route No. 1 is Source Bus Stop 1 ->Destination Bus Stop 10)

vii. Service provider data (24b) –

i. Boarding bus stand indicated as route specific stop index (1 byte).

ii. Alighting bus stand indicated as route specific stop index (1byte).

iii. Not used (1 byte); since this is a fixed fare the boarding and alighting stops are not of interest. The starting and ending stop indices are used for operator reference.

Boarding Stop (1B)	Alighting Stop (1B)	Not Used (1B)
00000010b	00001001b	00000000b

Service provider data = 0x01, 0x0A, 0x00

viii. Txn Status (4b) - 0011b (One tap/Ticket)

ix. RFU (4b) = 0000b (not used)

= 0x00 (combined Txn Status and RFU)

d) CSA validation data elements after Entry is shown below:

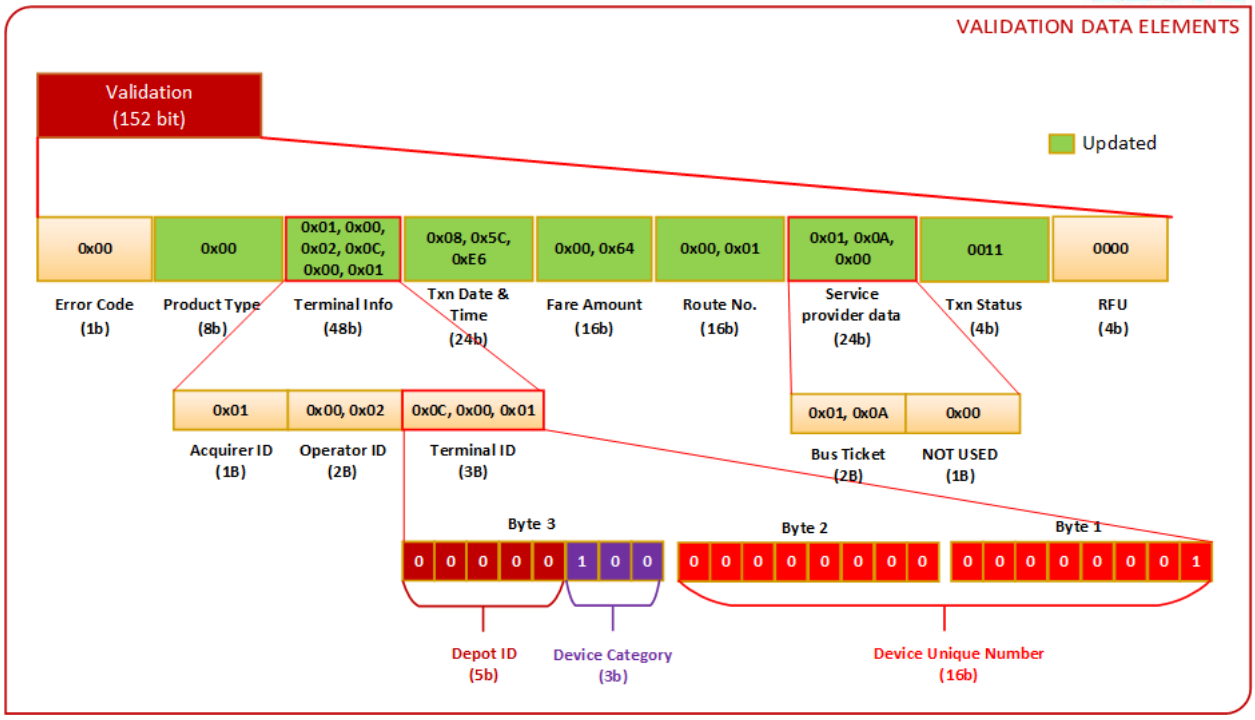


Figure 43: Scenario 5 - CSA Validation Data Elements after Entry

e) CSA history data elements after Entry is shown below:

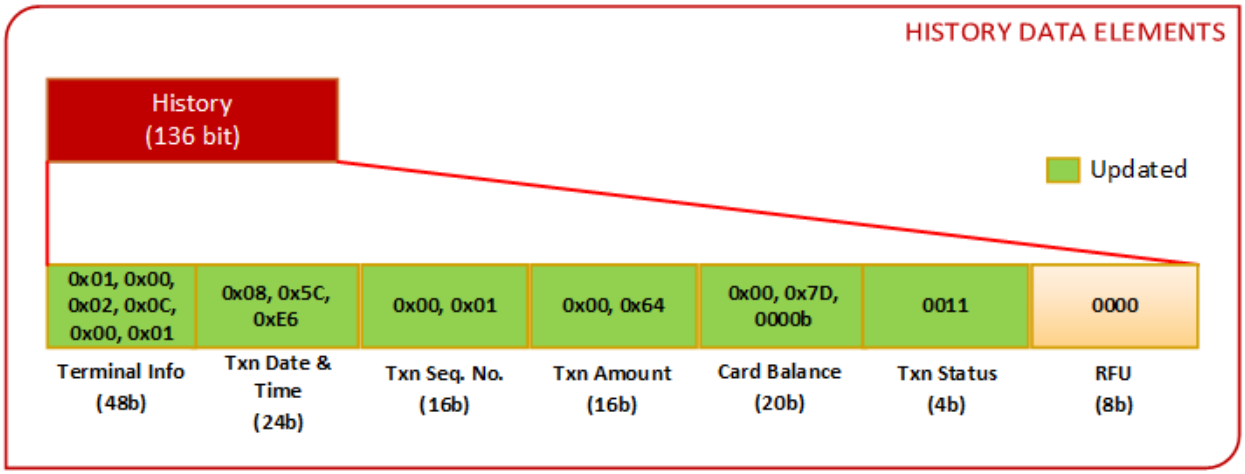


Figure 44: Scenario 5 - CSA History Data Elements after Entry

f) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

g) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

Scenario 6:

In bus, consider the scenario wherein a conductor is issuing ticket via an electronic ticket issuance machine (ETIM). The conductor selects a source (stop 1) and destination stop (stop 2) on the ETIM as requested by the commuter and then takes payment from the commuter's NCMC card with an assumption that card has balance (Say ₹190). A single 'ticket' transaction is made on the card in which a distance based fare (say₹12) is deducted.

The terminal is of Mobile Terminal - ETIM Type

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.

Assumptions for card data elements & CSA data elements

1. Card effective date - 15/02/19.
2. No error code present in card validation data elements present in CSA.
3. Assume that the card is a not new card that has completed scenario 4 before being used here (i.e. CSA validation data elements will not be all 0x00).
4. Txn status is previous transaction completed - Ticket i.e. "0011" in card validation data elements present in CSA.

Assumptions for operator specific data elements

1. Route is indicated by a 4-digit number; route number chosen for this application is '1'.
2. Bus stops are referenced in card by a route specific stop index. The index is fixed with the 'up' direction of the route and doesn't change for the return trip.
3. In the illustration considered below, it is assumed that the route '1' has a total of 10 stops and the commuter boards the bus at stop indexed 9 and alights at stop indexed 2.

Assumptions for Terminal data elements

1. Product Type - 0 (E.g. Normal Card)
2. Acquirer ID - 1 (E.g. ACQ1)
3. Operator ID - 2 (E.g. operator 2)
4. In Bus, the following representative nomenclature is used:-
   a) Terminal ID
      i. Device registration bus depot ID (5b) - 1 (00001b) (E.g. Bus Depot1)
      ii. Device Category (3b) - 5 (101b) (Mobile Terminal ETIM, refer PART V)
      iii. Device Unique Number (16b) - 2 (0x00, 0x02)
   b) Txn Seq No. of above Terminal - 0x01, 0x00

Steps

1. Card general data present in CSA before commuter has shown their card to terminal is shown below:

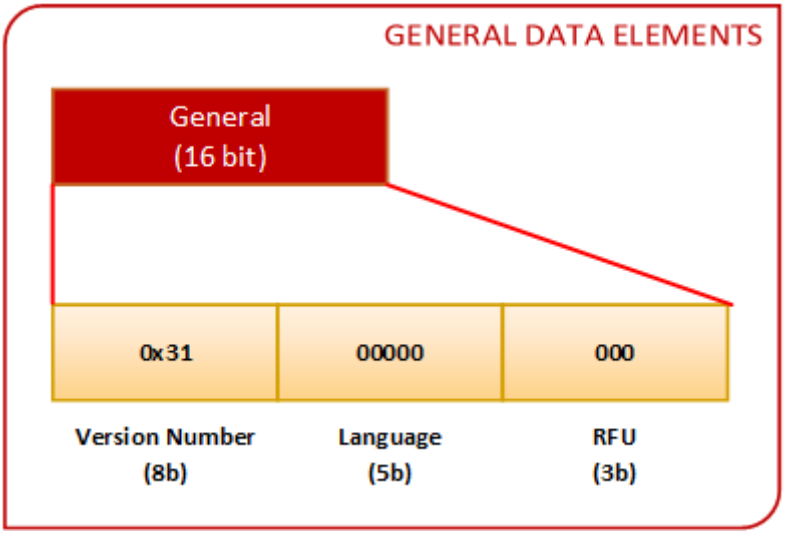


Figure 45: Scenario 6 - General Data Elements before Entry

2. Card validation data present in CSA before commuter has shown their card to terminal is shown below:

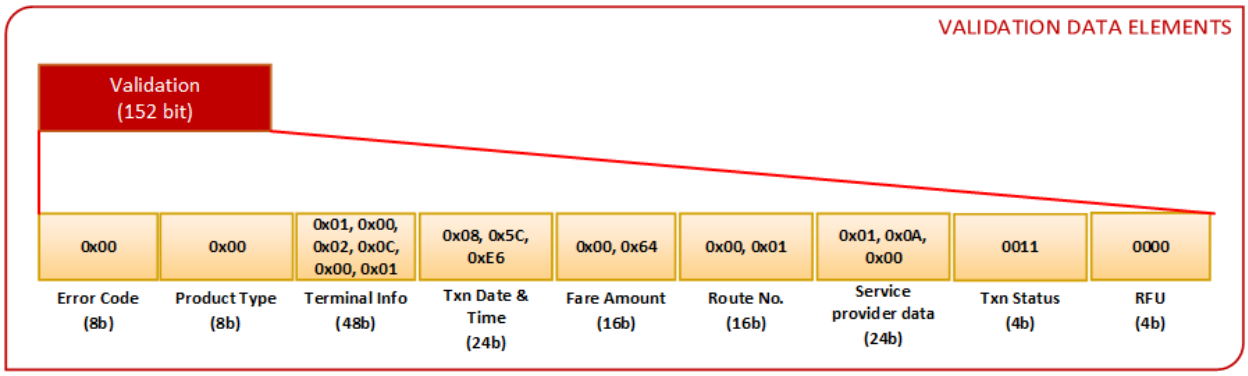


Figure 46: Scenario 6 - Validation Data Elements before Entry

3. Card history data present in CSA before commuter has shown their card to terminal is shown below:

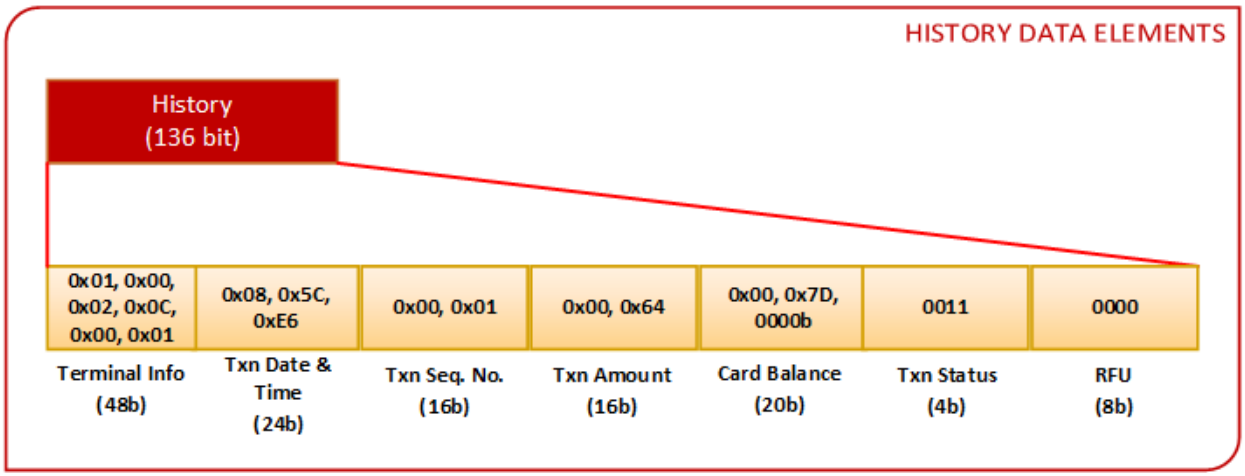


Figure 47: Scenario 6 - History Data Elements before Entry

4. Commuter will place the card on the terminal.
5. Following operations will be performed by terminal
   a) Terminal will check CSA general data elements:
      i. Version Number (8b) - 00110001b (3.1)
   b) Terminal will check CSA validation data elements:
      i. Error code (8b) - 00000000b (No Error)
      ii. Txn Status (4b) - 0011b (One Tap/Ticket)

c) For Same operator (using operator ID present in card) then, terminal will check:

i. Route No. (8b) - 00000001b

ii. Txn Date & Time (24b) - 0x08, 0x5C, 0xE6

Conclusion: If the Route No. present in card is same as Route No. present in terminal & Txn Date & Time is matching with current date within terminal current trip parameters then, it means commuter has already purchase the ticket.

Otherwise terminal will update the validation data.

d) Terminal will write in CSA validation data elements:

i. Product Type (8b) - 00000000b (Normal Card)

ii. Txn Status (4b) - 0011b (One Tap/Ticket)

iii. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Depot ID (5b), Device Category(3b), Device Unique number (16b)

Acquirer ID (1B)	**Operator ID (2B)**	**Terminal ID (3B)**
00000001b	00000000b, 00000010b	000001101b,00000000b, 00000010b

Terminal ID = 0x0D, 0x00, 0x02

iv. Txn Date & Time (24b) - 0x08, 0x5F, 0x7A (02/03/2020, 01:30)

v. Fare Amount (16b) - As per operator, it will specify the amount commuter has paid for the current 'ticket' = 0x78 (i.e. 120tp) (As per PART IV money is indicated as an integer multiple of 10 paisa)

vi. Route No. (16b) - 0x00, 0x01 (operator specific, assume Route No. 1 is Source Bus Stop -> Destination Bus Stop)

vii. Service provider data (24b) –

i. Boarding bus stand indicated as route specific stop index (1 byte).

ii. Alighting bus stand indicated as route specific stop index (1byte).

iii. Not used (1 byte); since this is a fixed fare the boarding and alighting stops are not of interest. The starting and ending stop indices are used for operator reference.

Boarding Stop (1B)	Alighting Stop (1B)	Not Used (1B)
00001001b	00000010b	00000000b

Service Provider Data = 0x09, 0x02, 0x00

viii. Txn Status (4b) = 0011b (One tap/Ticket)

ix. RFU (4b) = 0000b (not used)

= 0x00 (combined Txn status and RFU)

e) CSA validation data elements after Entry is shown below:

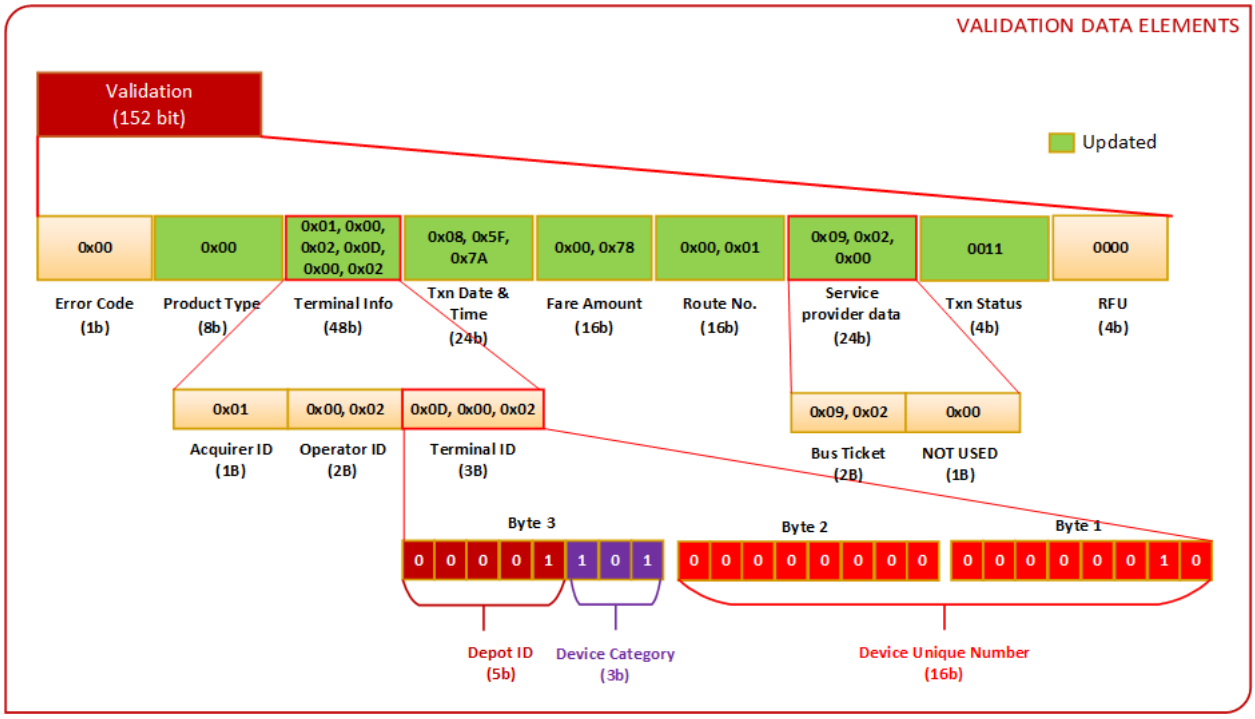


Figure 48: Scenario 6 - Validation Data Elements after Entry

f) CSA history data elements after Entry is shown below:

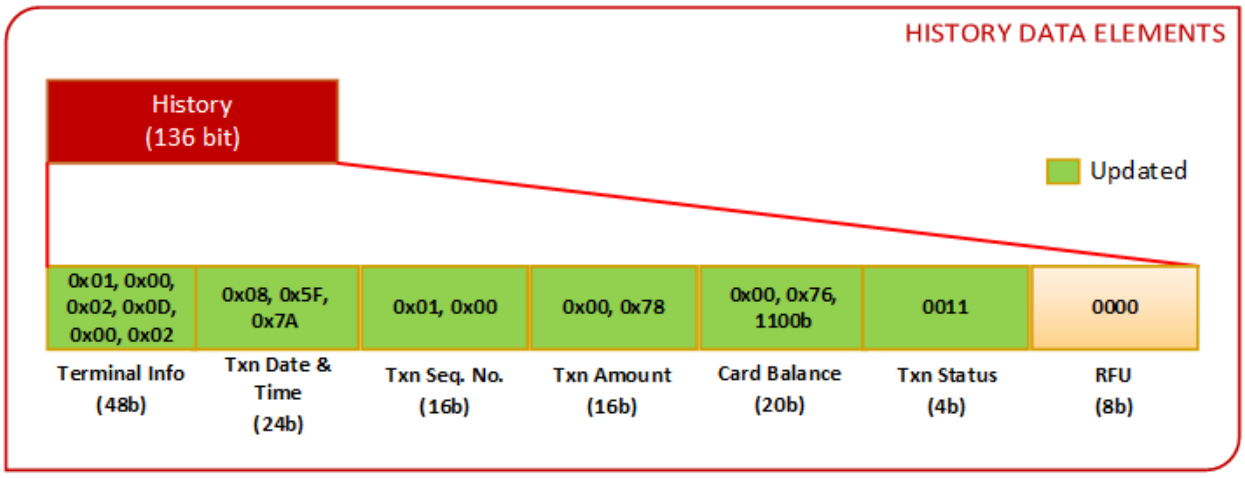


Figure 49: Scenario 6 - History Data Elements after Entry

g) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

h) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

Scenario 7

Two tap using pass present in OSA i.e. first tap at Entry terminal and second tap at Exit terminal. The pass limit will be decremented by 1 at Exit with an assumption that card is new and having a Pass Limit/No. of Trip present is 10 and commuter is going from Station 1 to Station 2 which is in Zone 1.

(Note: All steps and data elements are recommendation only for OSA)

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.
3. Operator Service Area (OSA) should be present.
4. Metro pass should be present in OSA.

Assumptions for metro pass data elements in OSA

i. Product Type - 0x20 (E.g. Metro Pass)
ii. Pass Limit/No. of Trip - 0x0A
iii. Activation/Start Date and Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)
iv. Expiry Date (16b) - 0x01, 0x6C (31/12/19)
v. Valid Route No./Zone (10b) - 0x00, 01b (E.g. Zone 1)
vi. Valid Entry Station ID (10b) - 1 (E.g. Station 1)
vii. Valid Exit Station ID (10b) - 2 (E.g. Station 2)
viii. Bonus Amount/Trip (10b) - Not used
ix. Class/Privileges (2b) - Not used
x. Daily Limit (4b) - Not used

Assumptions for card data elements &OSA data elements

1. Card effective date - 01/01/19.
2. No error code present in card validation data elements present in OSA.
3. Txn status is previous transaction completed i.e. “0000b” in card validation data elements present in OSA.
4. New card, first transaction i.e. all OSA validation and history data elements are 0x00.

Assumptions for terminal data elements

1. Product Type - 0x10 (E.g. Metro Pass)
2. Acquirer ID - 1 (E.g. ACQ 1)
3. Operator ID - 1 (E.g. operator 1)
4. Entry Station
   a) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1
   b) Txn Seq No. of above Terminal- 0x00, 0x00
5. Exit Station
   a) Terminal ID
      i. Station ID (12b) -2 (E.g. Station 2)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) -2
   b) Txn Seq No. of above Terminal - 0x00, 0x00

Steps

1. Card general data element present in OSA before commuter has shown their card to terminal is shown below:

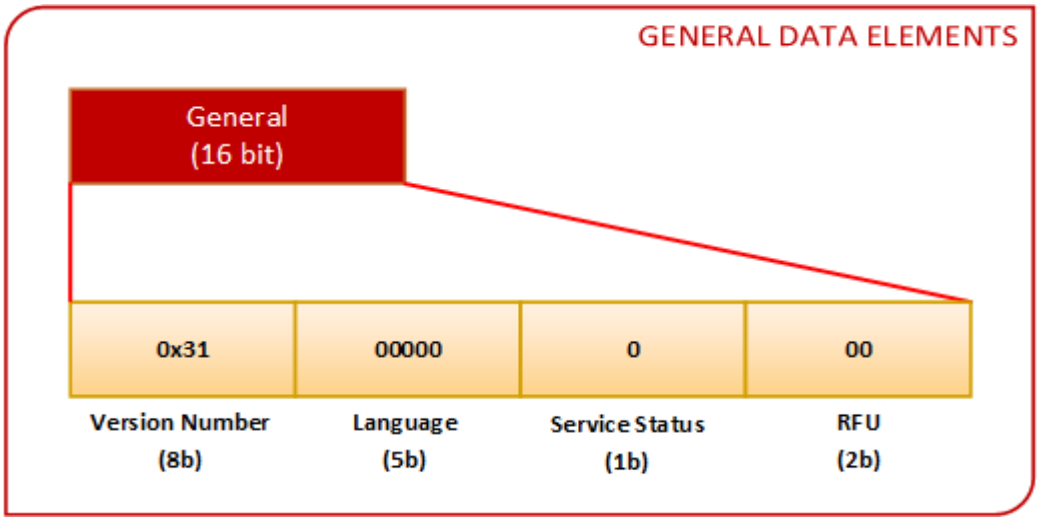


Figure 50: Scenario 7 - OSA General Data Elements

2. Card validation data present in OSA before commuter has shown their card to terminal is shown below:

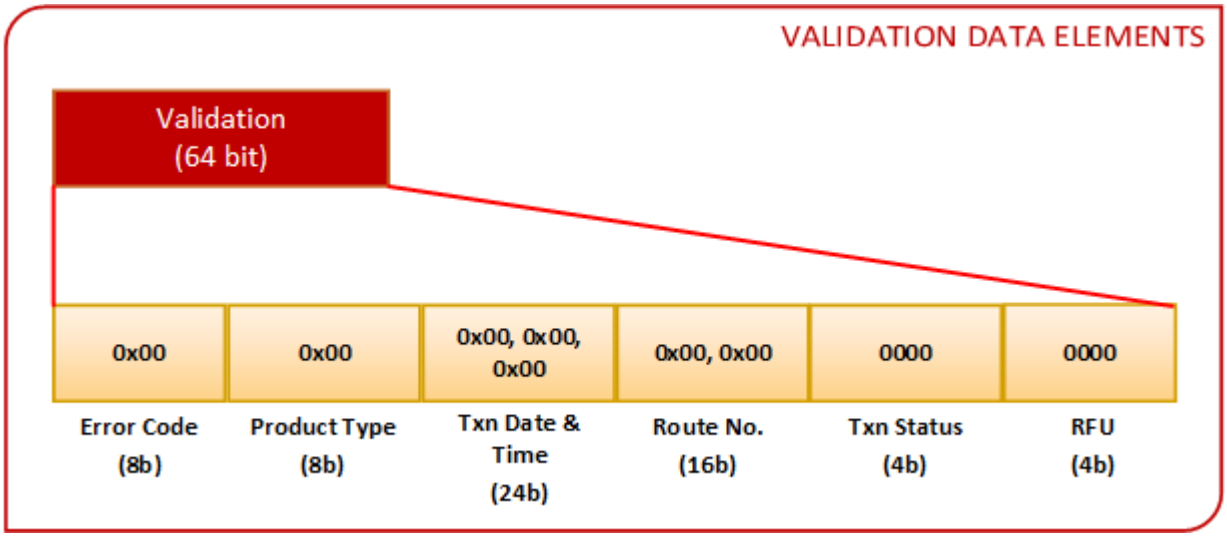


Figure 51: Scenario 7 - Validation Data Elements before Entry

3. Card history data elements data present in OSA before commuter has shown their card to terminal is shown below:

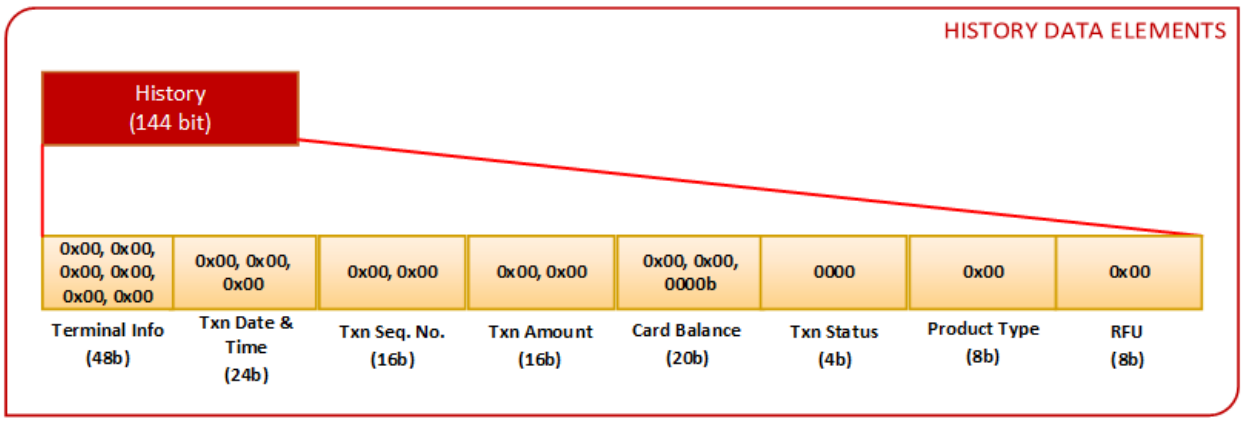


Figure 52: Scenario 7 - History Data Elements before Entry

4. Commuter will place the card on the terminal.
5. **At Entry,**
   a) Terminal will check OSA general data elements:
      i. Version Number (8b) -00110001b (3.1)
      ii. Language Info (5b) - 00000b (English)
      iii. Service Status (1b) - 0b (Pass may be present)
   b) Terminal will check OSA validation data elements:
      i. Error Code (8b) -00000000b (No Error)

ii. Txn Status (4b) - 0000b (Exit Done)

c) Terminal will check the validity of metro pass:

i. Product Type - 0x20 (Metro Pass)

ii. Pass Limit/No. of Trip - 0x0A

iii. Activation/Start Date and Time (24b)- 0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Expiry Date (16b) - 0x01, 0x6C(31/12/19)

v. Valid Route No./Zone (10b) - 0x00, 01b (E.g. Zone 1)

vi. Valid Entry Station ID (10b)- 1 (E.g. Station 1)

vii. Valid Exit Station ID (10b) - 2 (E.g. Station 2, but not used at Entry)

viii. Bonus Amount/Trip (10b) - Not used

ix. Class/Privileges (2b) - Not used

x. Daily Limit (4b) - Not used

d) Terminal will write in OSA validation data elements:

i. Product Type (8b) - 00100000b (Metro Pass).

ii. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Station ID (12b), Device Category (6b), and Device Number (6b).

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00010000b,10000001b

Terminal ID - 0x00, 0x10, 0x81

iii. Txn Date & Time (24b) -0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Route No. (16b) - 0x00, 0x00 (operator specific, assume not used)

v. Txn Status (4b) - 0001b (Entry)

vi. RFU (4b) - 0000b (Not used)

= 0x10 (Combined Txn status and RFU)

e) OSA validation data elements after commuter has entered is shown below:

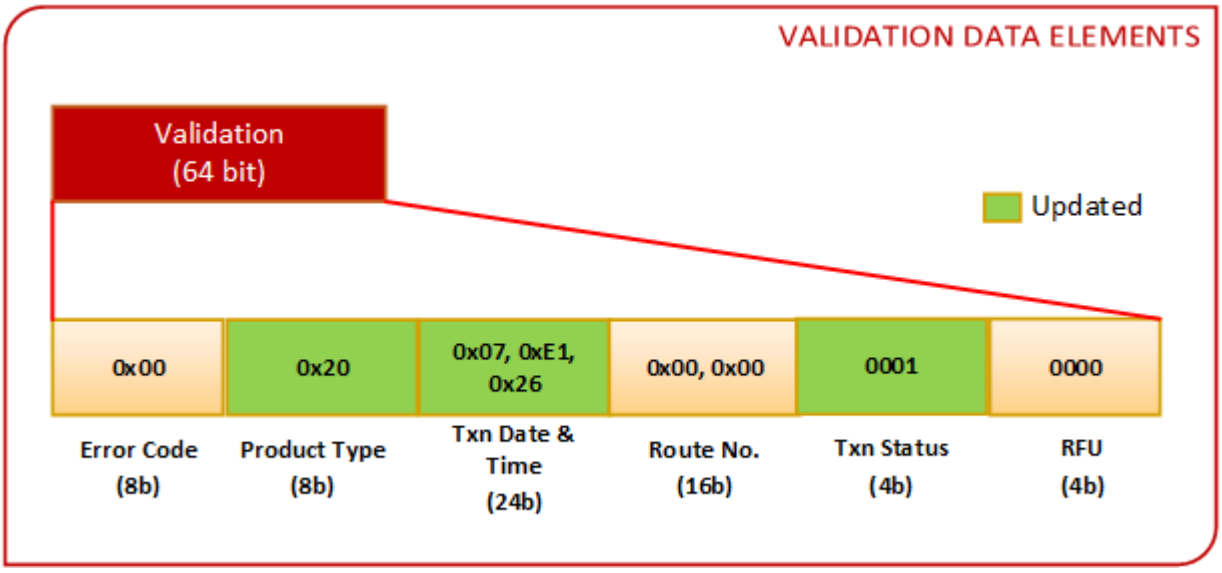


Figure 53: Scenario 7 - OSA Validation Data Elements after Entry

6. At Exit,

a) Terminal will check OSA validation data elements in card:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0001b (Entry)

iii. Product Type (8b) -00100000b (Metro Pass)

iv. Txn Date & Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)

b) Terminal will check the validity of metro pass:

i. Product Type - 0x20 (E.g. Metro Pass)

ii. Pass Limit/No. of Trip - 0x0A

iii. Activation/Start Date and Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Expiry Date (16b) – 0x01, 0x6C (31/12/19)

v. Valid Route No./Zone (10b) - 0x00, 01b (E.g. Zone 1)

vi. Valid Entry Station ID (10b) - 1 (E.g. Station 1)

vii. Valid Exit Station ID (10b) - 2 (E.g. Station 2, but not used at Entry)

viii. Bonus Amount/Trip (10b) - Not used

ix. Class/Privileges (2b) - Not used

x. Daily Limit (4b) - Not used

c) **Assume current time at exit terminal is 25/12/19 14:35**

d) Terminal will check the time limit allowed between entry and exit as per operator business rule. (Assuming exit happened within time limit)

e) Terminal will decrement pass limit by 1as per operator business rules.

f) Terminal will update the validation data elements.

Txn Status (4b) - 0000b(Exit Done)

g) OSA validation date elements after Exit is shown below:

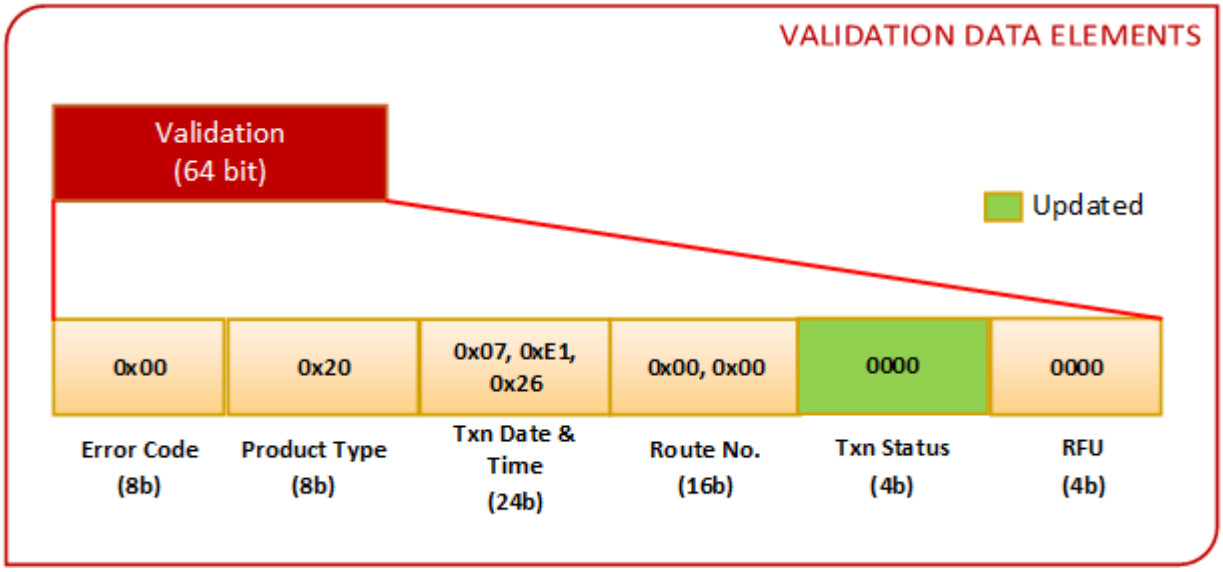


Figure 54: Scenario 7 - OSA Validation Data Elements after Exit

h) Terminal will update the history data elements

i. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Station ID (12b), Device Category (6b), Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b,00100000b, 10000010b

Terminal ID - 0x00, 0x20, 0x82

ii. Txn Date & Time (24b) - 25/12/19 14:35 (0x7E12B)

iii. Txn Seq. No. (16b)- 0x00, 0x01

iv. Txn Amount (16b) -0x00, 0x00 (Assume not used)

v. Card Balance (20b) -0x00, 0x00, 0000b (Assume not used)

vi. Txn Status (4b) -0000b

i) OSA history data elements after Exit is shown below:

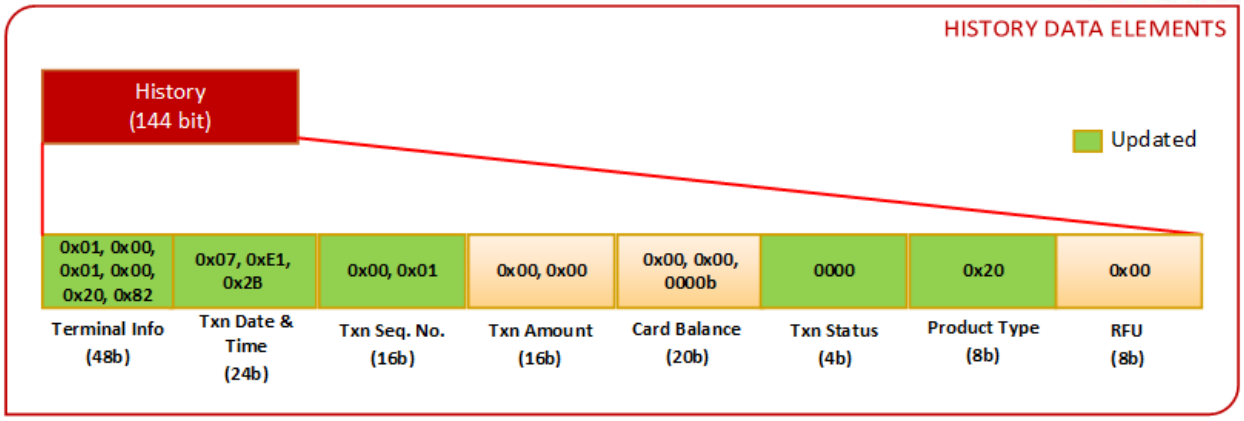


Figure 55: Scenario 7 - OSA History Data Elements after Exit

j) Terminal will update the pass limit/No. of Trip present in pass info and rest of the fields will remain the same

Pass Limit/No. of Trip = 0x09

k) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

l) Terminal will give the command to open the gate as per NCMC specification part 7.

Scenario 8

Two tap using pass present in OSA i.e. first tap at Entry terminal and second tap at Exit terminal. The pass is of discounted fare type and fare will be debited at Exit with an assumption that card is new and having a balance (say ₹500) and commuter is going from Station 1 to Station 2.

(Note: All steps and data elements are recommendation only for OSA)

Dependency on card

1. Card effective date should be present.
2. Common Service Area (CSA) should be present.
3. Operator Service Area (OSA) should be present.
4. Metro pass should be present in OSA.

Assumptions for metro pass data elements in OSA

i. Product Type - 0x21 (E.g. Metro Pass with discounted fare)
ii. Pass Limit/No. of Trip - Not used
iii. Activation/Start Date and Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)
iv. Expiry Date (16b) - 0x01, 0x6C (31/12/19)
v. Valid Route No./Zone (10b) - 0x00, 01b (E.g. Zone 1)
vi. Valid Entry Station ID (10b) - 1 (E.g. Station 1)
vii. Valid Exit Station ID (10b) - 2 (E.g. Station 2)
viii. Bonus Amount/Trip (10b) - Not used
ix. Class/Privileges (2b) - Not used
x. Daily Limit (4b) - Not used

Assumptions for card data elements & OSA data elements

1. Card effective date - 01/01/19.
2. No error code present in card validation data elements present in OSA.
3. Txn status is previous transaction completed i.e. “0000b” in card validation data elements present in OSA.
4. New card, first transaction i.e. all OSA validation and history data elements are 0x00.

Assumptions for terminal data elements

1. Product Type - 0x21 (E.g. Metro Pass with discounted fare)
2. Acquirer ID - 1 (E.g. ACQ 1)
3. Operator ID - 1 (E.g. operator 1)
4. Entry Station
   a) Terminal ID
      i. Station ID (12b) - 1 (E.g. Station 1)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) - 1
   b) Txn Seq No. of above Terminal- 0x00, 0x00
5. Exit Station
   a) Terminal ID
      i. Station ID (12b) -2 (E.g. Station 2)
      ii. Device Category (6b) - 2 (E.g. Automatic Gate)
      iii. Device Number (6b) -2
   b) Txn Seq No. of above Terminal - 0x00, 0x00

Steps

1. Card general data element present in OSA before commuter has shown their card to terminal is shown below:

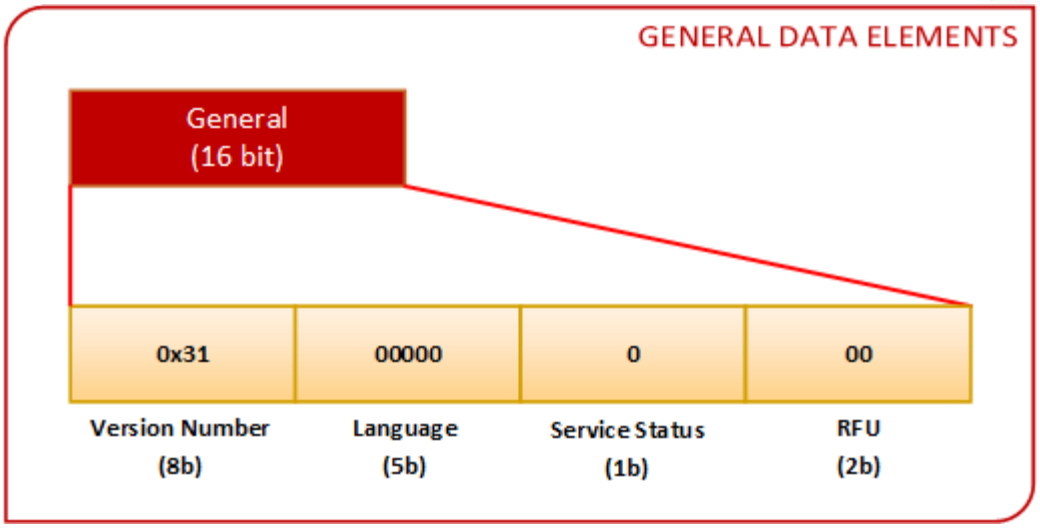


Figure 56: Scenario 8 - OSA General Data Elements

2. Card validation data present in OSA before commuter has shown their card to terminal is shown below:

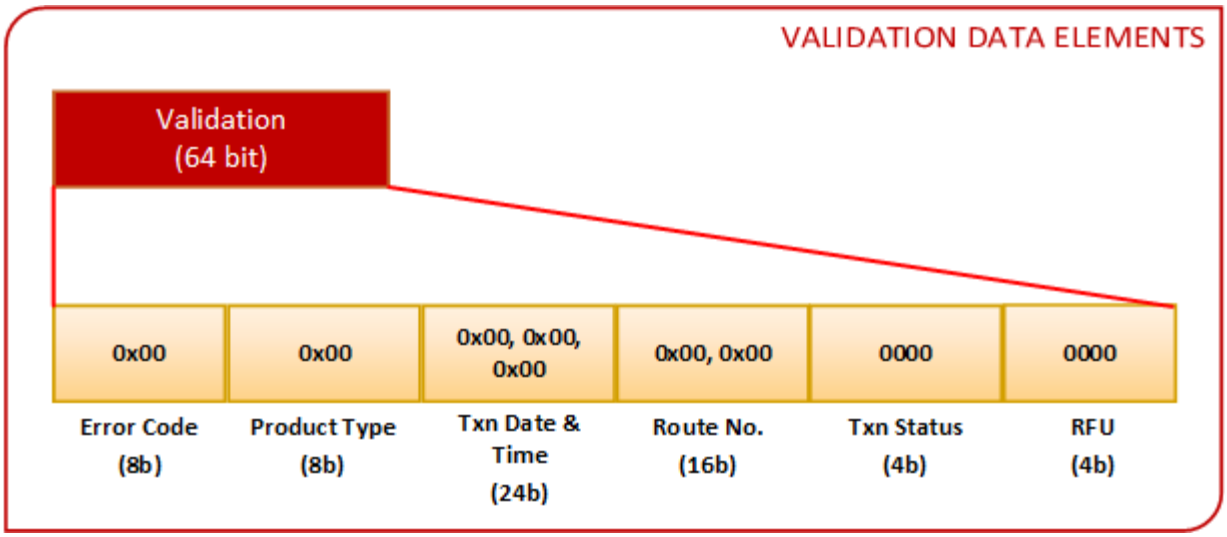


Figure 57: Scenario 8 - Validation Data Elements before Entry

3. Card history data elements data present in OSA before commuter has shown their card to terminal is shown below:

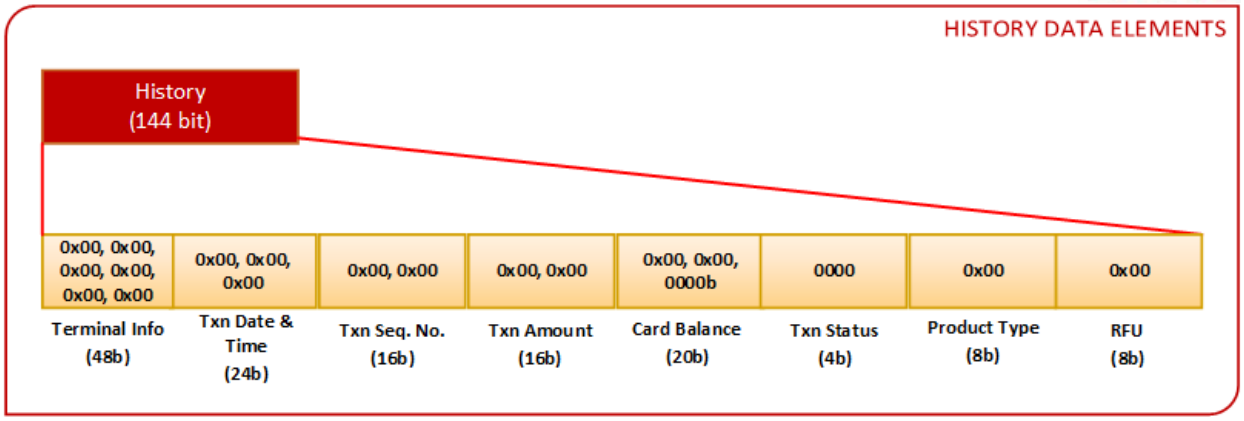


Figure 58: Scenario 8 - History Data Elements before Entry

4. Commuter will place the card on the terminal.

5. At Entry,

a) Terminal will check OSA general data elements:

i. Version Number (8b) - 00110001b (3.1)

ii. Language Info (5b) - 00000b (English)

iii. Service Status (1b) - 0b (Pass may be present)

b) Terminal will check OSA validation data elements:

i. Error Code (8b) -00000000b (No Error)

ii. Txn Status (4b) - 0000b (Exit Done)

c) Terminal will check the validity of metro pass:

i. Product Type - 0x21 (Metro Pass with discounted fare)

ii. Pass Limit/No. of Trip - Not used

iii. Activation/Start Date and Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Expiry Date (16b) - 0x01, 0x6C (31/12/19)

v. Valid Route No./Zone (10b) - Not used

vi. Valid Entry Station ID (10b) - Not used

vii. Valid Exit Station ID (10b) - Not used

viii. Bonus Amount/Trip (10b) - Not used

ix. Class/Privileges (2b) - Not used

x. Daily Limit (4b) - Not used

d) Terminal will write in CSA validation data elements:

i. Product Type (8b) -00100001b (Metro Pass with discounted fare).

ii. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b).Terminal ID consists of Station ID (12b), Device Category (6b), and Device Number (6b).

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b, 00010000b,10000001b

Terminal ID - 0x00, 0x10, 0x81

iii. Txn Date & Time (24b) -0x07, 0xE1, 0x26 (25/12/19, 14:30)

iv. Route No. (16b) - 0x00, 0x00 (operator specific, assume not used)

v. Txn Status (4b) - 0001b (Entry)

vi. RFU (4b) - 0000b (Not used)

= 0x10 (Combined Txn status and RFU)

e) CSA validation data elements after commuter has entered is shown below:

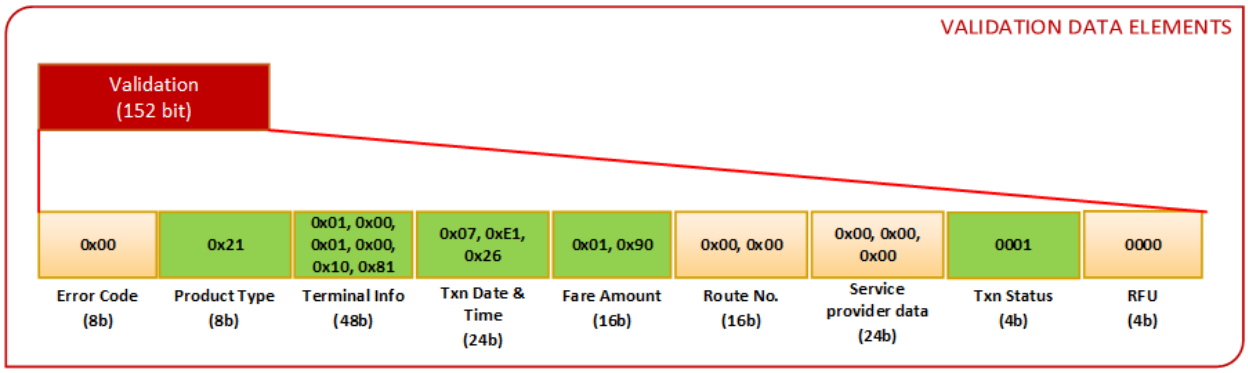


Figure 59: Scenario 8 - CSA Validation Data Elements after Entry

6. At Exit,

a) Terminal will check OSA validation data elements in card:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0000b (Exit)

iii. Product Type (8b) - 00100000b (E.g. Metro Pass)

iv. Txn Date & Time (24b) - 0x07, 0xE1, 0x26 (25/12/19, 14:30)

b) Terminal will check CSA general data elements:

i. Version Number (8b)- 00110001b (3.1)

ii. Language Info (5b) - 00000b (English)

c) Terminal will check CSA validation data elements in card:

i. Error Code (8b) - 00000000b (No Error)

ii. Txn Status (4b) - 0001b (Entry)

iii. Terminal Info (48b)

1) Acquirer ID (8b) - 0x01

2) Operator ID (16b)- 0x00, 0x01

3) Terminal ID (24b) - 0x00, 0x10, 0x81

d) For Same operator (using operator ID present in card) then, terminal will check:

i. Product Type (8b) - 00100001b (Metro Pass with discounted fare)

ii. Txn Date & Time (24b) - 0x07,0xE1,0x26 (Assume entry log is of 25/12/19 14:30)

e) **Assume current time at exit terminal is 25/12/19 14:35**

f) Terminal will check the time limit allowed between entry and exit as per operator business rule. (Assuming exit happened within time limit)

g) Terminal will debit amount as per operator business rules.

h) Terminal will update the validation data elements.

Txn Status (4b) -0000b (Exit Done)

i) CSA validation date elements after Exit is shown below:

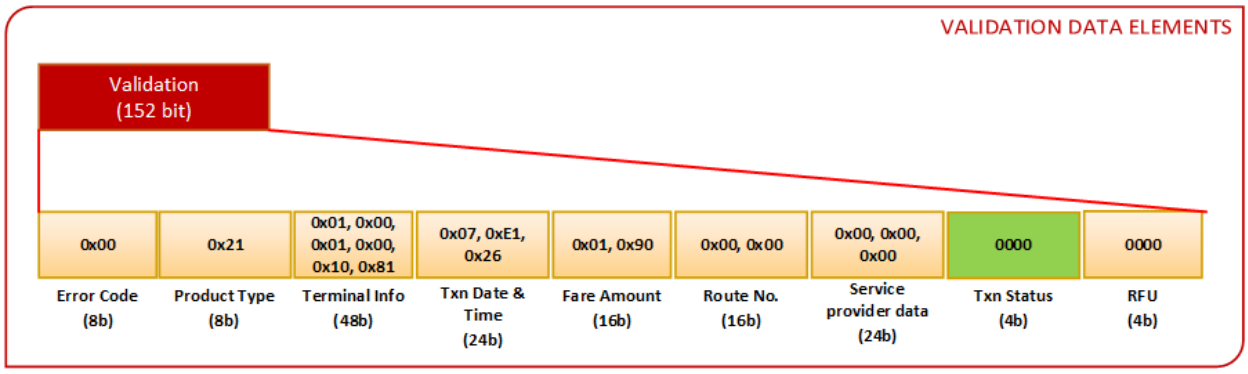


Figure 60: Scenario 8 - CSA Validation Data Elements after Exit

j) Terminal will update the history data elements

i. Terminal Info (48b) - Acquirer ID (1B), Operator ID (2B) and Terminal ID (24b). Terminal ID consists of Station ID (12b), Device Category (6b), Device Number (6b)

Acquirer ID (1B)	Operator ID (2B)	Terminal ID (3B)
00000001b	00000000b, 00000001b	00000000b,00100000b, 10000010b

Terminal ID - 0x00, 0x20, 0x82

ii. Txn Date & Time (24b) - 25/12/19 14:35 (0x7E12B)

iii. Txn Seq. No. (16b)- 0x00, 0x01

iv. Txn Amount (16b)-0x00, 0x64 (100tp)

v. Card Balance (20b)-0x01, 0x38, 1000b (5000tp)

vi. Txn Status (4b)-0000b

k) CSA history data elements after Exit is shown below:

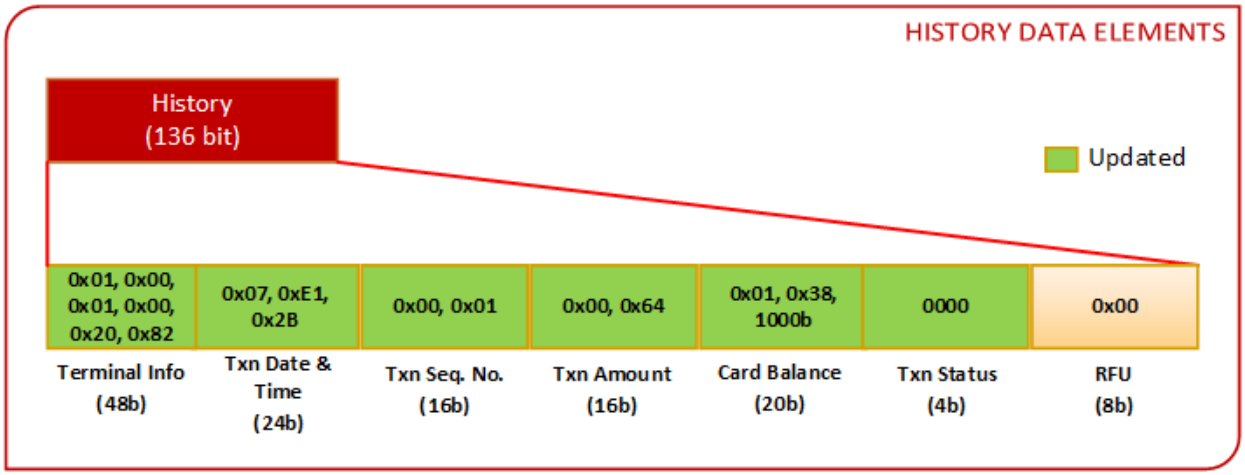


Figure 61: Scenario 8 - CSA History Data Elements after Exit

l) Terminal will generate the transit file and send it to AFC system as per NCMC specification part 5.

m) Terminal will generate the financial file and send it to acquirer as per NCMC specification part 6.

n) Terminal will give the command to open the gate as per NCMC specification part 7.

**********End of Chapter 1 **********

Chapter 2

Interface Specification of NCMC Ecosystem

PART V:
Terminal - AFC Backend Communication Interface

Centre for Development of Advanced Computing (CDAC), Noida
Ministry of Electronics and Information Technology (MeitY)
Government of India

Contents

List of Tables

List of Figures

Version History

Date	Version	Author	Comments
May 9, 2018	V1.0	CDAC, Noida	First Release
November 11, 2019	V1.1	CDAC, Noida	Updated Table 2: References, Section 2.1, 2.3, Section 3.1.1, 3.1.1.1, 3.2.2, Section 4 Inserted Section 5, Section 6, Section 7, Section 8, Section 9,9.1 Updated Annexure A and Inserted Annexure B
Feb 28, 2020	V1.2	CDAC, Noida	Updated Section 4. Added Configuration, Alarm, General Message and Transaction Examples. Modification in Common Message Object structure, Configuration, Alarm and Transaction data Structure.

1. Introduction

This document specifies the interface about the Terminal-AFC Backend and Inter Server Interface. Since the document specifies the common standard specifications for the entire transit network used throughout the country, therefore the AFC Backend (Server) interface with terminal needs to be standardized and interoperable. The objective of the document is to elucidate the building blocks of an interoperable Terminal-Backend interface based on industrial standard specifications encompassing features like robust design, scalability, security and easy adaptability.

1.1. Abbreviations

Table 23: Abbreviations

Abbreviations	
ACK	Acknowledgement
AFC	Automatic Fare Collection
AG	Automatic Gate
AN	Alpha Numeric
ANS	Alpha Numeric with special characters
AVM	Add Value Machine
CS	Central Server
EMV	Euro pay, MasterCard and Visa
ETIM	Electronic Ticket Issuing machine
GPRS	General Packet Radio Service
HID	Human interface Device
ITMS	Intelligent Transport Management System
JSON	JavaScript Object Notation
LAN	Local Area Network
NCMC	National Common Mobility Card
NTP	Network Time Protocol
PAN	Primary Account Number
PCI DSS	Payment Card Industry Data Security Standard
P2PE	Point to Point Encryption
RRN	Request Retrieval Number
SSL	Secure Socket Layer
TLS	Transport Layer Security
TOM	Ticket Office Machine
USB	Universal Serial Bus
XML	Extensible Markup Language

1.2. Scope

This document specifies a design framework about Terminal interface with AFC Backend and other interfaces viz., Inter Server Interface. All the transit related data will be transferred from the Terminal to the AFC Backend i.e. AFC system. The financial or sensitive data shall be sent from the Terminal to the Acquirer either in an Indirect Mode (via the AFC System) or in a Direct Mode (directly from the Terminal). Design of the Terminal-AFC Backend interface for financial & transit data, in case of indirect mode; and Terminal-AFC Backend interface for transit data, in case of direct mode are discussed in detail in subsequent sections of this document. The AFC-Acquirer interface is not in scope of this document. For that interface details, kindly refer Part VI: AFC Ecosystem - Acquirer Interface.

The document defines the following requirements:

1. Physical and logical communication interface between Terminal and AFC Backend, and Inter-Server Interface.
2. Terminal-AFC Backend Communication Capability.
3. AFC System architecture and the communication mechanism.
4. Security framework for data transmission in AFC ecosystem between the Terminal and AFC Backend.
5. Structure of the Transaction Data.
6. Parameter configuration required for communication between the Terminal and AFC Backend.

1.3. References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Table 24: References

Reference	Title
ISO	ISO_8583-1_2003: Financial transaction card originated messages — Interchange message specifications — Part 1: Messages, data elements and code values.
RuPay	RuPay Global Clearing and Settlement Technical Message Specification Version 1.9
IETF	JSON standard reference - RFC 8259, ECMA-404
W3C	W3C XML Version 1.0
NCMC	Part VI: AFC Ecosystem - Acquirer Interface

1.4. Definitions

Table 25: Definitions

Terminology	Meaning
Terminal / AFC Equipment	Terminals are equipments like AG, AVM and TOM. On one end, they are customer-facing in that they provide user interaction capability via card reading media and on the other hand, they then generate transactions based on those interactions which are then exchanged with the Frontend or Backend Servers.
Frontend Server / Frontend	The server which directly interacts with the Terminal. It may be an optional entity. It may be Station or Intermediate Server System.
Backend Server /Backend	The server which may directly interact with the Terminal (if Frontend Server does not exist) and Acquirer/Issuer. If Frontend Server exists, Backend Server shall interact with terminal vi Frontend Server. It may be Central Server System.
AFC System	It consists of Frontend/Backend server which is facing the terminal and also facing the acquirer. (AFC System does not consist of Terminal and gate)
AFC Ecosystem	The AFC Ecosystem consists of Terminal, Gate and AFC System).
Acquirer	An acquirer may be bank or financial institution that processes credit or debit card payments on behalf of a merchant. The acquirer allows merchants to accept payments from the card-issuing banks within an association. It may also be Operator.
Issuer	An issuer may be a bank that offers card association branded payment cards directly to user. The name is derived from its practice of issuing payment to the acquiring bank on behalf of its user.

Terminology	Meaning
Financial Data	Financial data or Financial transaction data are the sets of data elements that carry the required field for financial debit, credit or settlement required by issuer bank and as defined by “Rupay Global Clearing and Settlement Technical Message Specification_V-1.9.” specification
Transit Data	Transit data is a set of data elements that are required by the Operator to maintain the transit service required driven by various business rules. This data is transferred from terminal to AFC Backend.
Transaction Data	Transaction data is a combination of Financial and transit data.
Direct Mode	Direct mode is a type of interface through which validation terminal directly communicates with acquirer/issuer interface. In this mode, transaction data splits into two different structures at the validation terminal: i.e. • Financial Data • Transit data. Financial data is directly transferred to Acquirer/Issuer bank from terminal. Transit data is transferred to AFC server and terminates there. In this case, Terminal must comply with PCIDSS or equivalent security standard. The terms “Direct Mode” and “Direct Transmission” have been used interchangeably in this document.

Terminology	Meaning
Indirect Mode	Indirect mode is a type of interface through which validation terminal communicates with acquirer/issuer interface via AFC System. In this mode, financial transaction data is transferred to Acquirer via AFC Backend. In this case, the complete AFC Ecosystem must comply with PCI-DSS or equivalent security standard. The terms “Indirect Mode” and “Indirect Transmission” have been used interchangeably in this document.
Standard Communication Process	As per the standard communication process, the LSB shall be transferred first in the network. The rightmost byte in the data structure is the LSB.
Mobile Terminals	The terminals which are installed on the bus, ferry or any moving vehicle. It may be carried by the conductor or fixed on the moving vehicle for ticket issuance, ticket access and validation etc. For e.g. ETIM, ticket validators, POS device etc.
Fixed Terminals	The terminals which are installed on fixed location for e.g. Validation Terminal installed on the Gate, Ticket Vending Machine, Ticket Office Machine etc.
Granularity	It means the configuration can be changed by the interval mentioned in the respective section.
RRN	For information about it, Kindly refer RuPay Global Clearing and Settlement Technical Message Specification Version 1.9

2. AFC Ecosystem and its Sub-systems

This section describes a broad overview of entire AFC ecosystem and its important subsystems.

2.1. System Architecture

This section gives a detailed description about the Terminal to AFC Backend Communication architecture where the financial data needs to propagate from Terminal to acquirer either directly (direct mode) or through AFC system (Indirect mode) and transit data from terminal to AFC Backend. The document does not imply any strict mandate to follow the exact architecture.

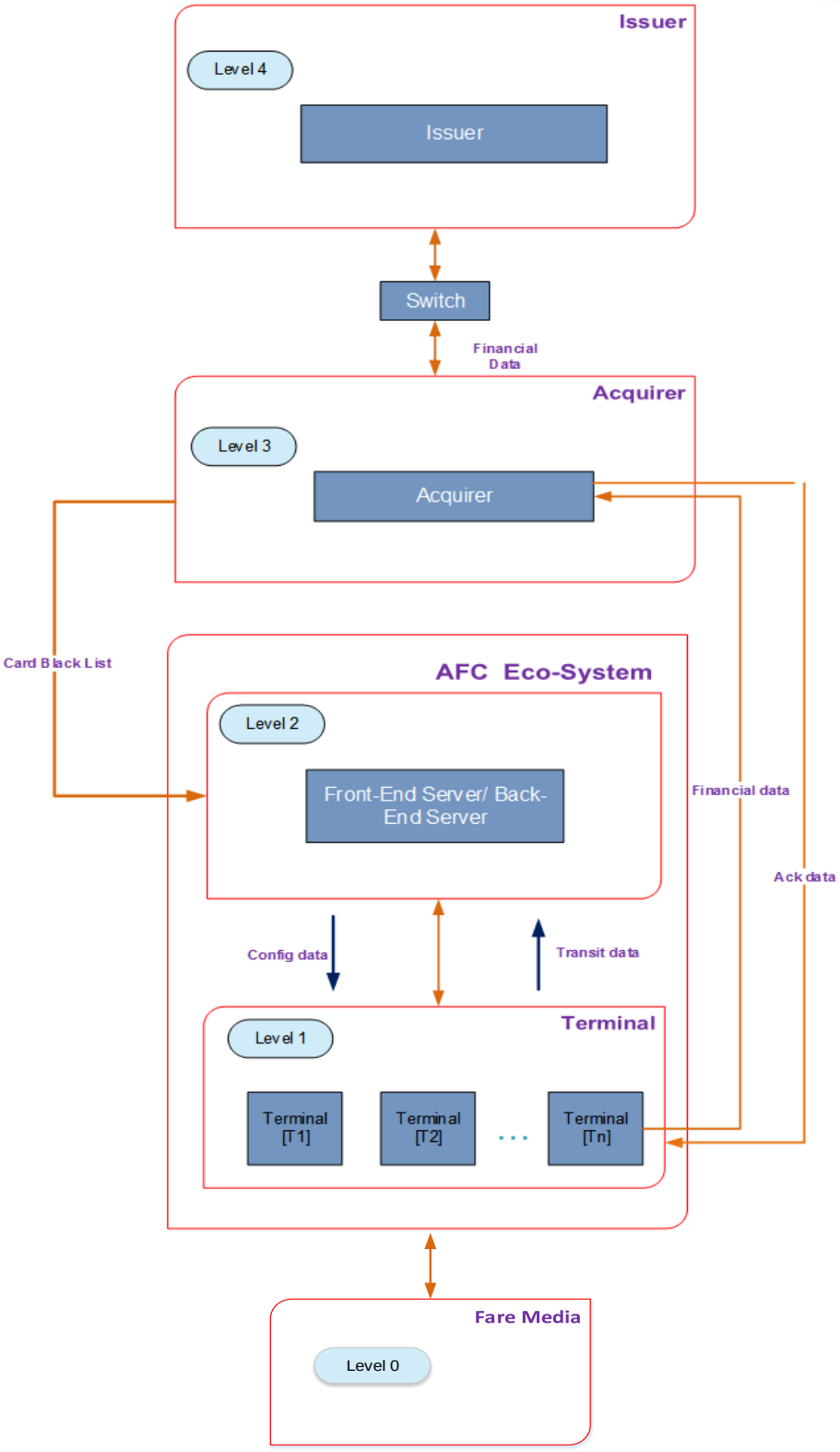


Figure 62: Direct Mode System Architecture

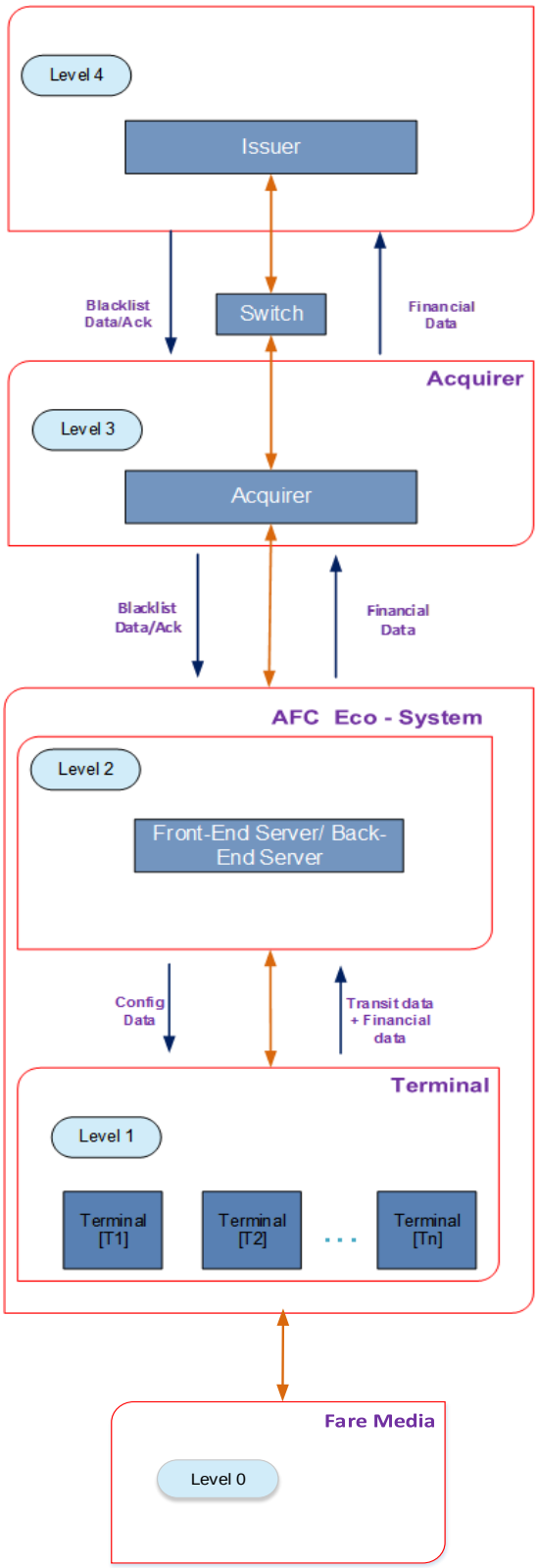


Figure 63: Indirect Mode System Architecture

i. Level 0: Fare Media

Level 0 contains the fare media using which the passenger travels and makes payments to the transport service provider. It enables passengers to make journeys using the chosen mode of transport. The ticket media shall be Account based ticketing, open loop, Quick Response (QR) code, Near Field Communication (NFC), and Mobile ticketing, Cash etc.

ii. Level 1: Terminal

AFC Terminal is the equipment which is used to read the media carried by the commuter and allows/disallows based on “Transit service management” defined in part IV- Transit Service. It also provides one or more customer services such as add value, issuing tickets, collecting fare, validation of tickets etc based on operator requirements. AFC terminals includes Validation Terminals/Validators/ETIM devices etc based on the Operator requirements.

The AFC terminals can be fixed or mobile depending upon the usage. The terminal shall be connected with the frontend Server/Backend Server of the AFC System depending upon the Operator/AFC requirement. In case of centralized AFC architecture, frontend server will not exist, validation terminal is directly connected to backend server. Against each successful transaction, the Terminal shall generate one or both the below mentioned transaction data:

1. **Financial Data**
2. **Transit Data**

Table below describes the most common cases for which transit and financial data would be generated or not.

Table 26: Transit and Financial Data

Type of Transaction	Transit Data	Financial Data
Zero Value Transaction	✓	Optional*
Non-Zero Value Transaction	✓	✓

Add Value Transaction	✓	✓
Torn Transaction**	✗	✗

* Optional means financial data may or may not be generated depending upon Operator-Acquirer agreement.

** In case of Torn transaction, no files are shall be generated. If the user taps the card again within 'n' seconds after the torn, then both Transit and Financial files shall be generated. n shall be decided by the Terminal kernel specifications.

Transaction data from terminal shall be transferred to AFC Backend (Transit) and the Acquirer (Financial) through secure communication channels. At AFC System (Level 2 in Figure 62 & Figure 63), in case of Indirect mode, the financial data may be saved for a configured period of time before it is forwarded to the acquirer.

iii. Level 2: AFC System

Frontend Server

Frontend Server shall exist depending on operator or AFC architecture and the type of transport network (In proposed use case, Station Server shall be treated as Frontend server). In case, frontend servers are present then the transaction data shall be transmitted from terminal to the Backend Server via Frontend Server. The Frontend Server shall receive configuration parameter data from the Backend Server, which shall be further routed to the end nodes or to the AFC terminals.

In addition, functions of the Frontend Server include:

- Supervision and Monitoring of AFC terminals.
- Remote configuration of AFC terminals.
- Report generation of revenue data and passenger flow data etc.

Backend Server

The Backend Server also known as Central Server (CS) is installed at the centralized location. It also has an interface with frontend server (if it exists) and external networks such as Banks/Acquirer or may be with third party services such

as ITMS etc.

The major function of the Backend Server includes:

- Filtering Financial and Transit Data (in case of indirect mode) and send the Financial Data to Acquirer for processing, authorization and settlement.
- Management and distribution of various data items and parameters required for the operation of AFC terminals through Secure Channel.
- Supervision and monitoring of subordinate AFC equipments and Frontend Server.
- Interfacing with Acquirer for execution of necessary settlement processing including transfer of Black-listed card details.
- Report Generation on revenue data and passenger flow data.

iv. Level 3: Acquirer

The acquirer shall group the financial transaction data received either from terminal or AFC server and send it to the issuer for clearing and settlement within the configurable time interval. For details kindly refer Part VI:AFC Ecosystem - Acquirer Interface.

v. Level 4: Issuer

The Issuer is involved in clearing and settlement of financial transactions. The issuer receives the transactions via switches for clearing and settlement. Issuer shall take accountability of financial management process related with the card wallet such as recharge or top-up of Global-wallet balance.

2.2. Operating System

The specification document does not specify any specific operating system requirement for the AFC Ecosystem. However, all the standard specification related with the communication protocol stack and configuration parameters must be available with the operating system present in the terminal, Frontend Server and Backend Server of the AFC system.

In case of the terminal, Frontend or Backend Server, the AFC system is not equipped with any standard operating system; the related driver for communication channel must be available with full featured functionality and stack as indicated in physical requirements in Section 3.1.

Following parameters shall be considered to select any operating system inside the AFC ecosystem:

OS Availability: For proprietary OS, at least 10 years of future support / up-gradation shall be provided by concerned entity.

For open source OS, the previous availability shall be at least 10 years with active updates. It must be available in public repository.

2.3. Data Backup

Data shall be stored in terminal or AFC System as per Operator requirements. The data shall be securely stored for some pre-configurable period of time at all levels of AFC Ecosystem even after transferring the data to other location. Once the configurable time expires the data shall be purged from that level in FIFO basis. However, in case of a communication failure, the data will not be purged till the data is sent to the other level even after expiry of the pre-configurable time. In that case it shall stop receiving new transactions. The data from the system shall also be taken out securely following the security guidelines as defined by Operator.

Following configurable parameters must be managed in Terminal and AFC system:

The duration of data storage will be either in transaction count or in time duration or both depending upon Operator requirements.

- In case of **time duration,** following configuration is needed:

  The time period of both Financial and Transit data preservations on the terminal shall be configurable for at least minimum 7 days with granularity of 6 hours. The option must provide individual time configuration for both Financial and Transit data. The time period for financial (in case of indirect mode) and transit data storage on AFC System shall be configurable for at least 3 months with granularity of 1 week.

- In case of **transaction count**, following configurations are needed:

  Numbers of stored transaction in terminal: It shall be configurable with at least 5 lakh transactions with a granularity of 10,000 records.

  Numbers of stored transaction in AFC System: It shall be configurable with at least 2500 lakh transactions with a granularity of 100 lakh records.

Table 27: Data Backup

Type	Time Duration/Granularity	Transaction Count / Granularity (in lakhs)
Terminal	7 days/ 6 hours	5/0.1
AFC System	3 months/1 week	2500/100

As per requirement of latest stable version of PCI DSS, data storage/back-up shall be configurable as per required for legal, regulatory, and/or business requirements.

3. Terminal to Frontend/Backend Server Interface

This section describes the components, systems and functions necessary to implement the Terminal to AFC System and Acquirer Communication Interface. The specification definitions are limited to the external interfaces for compatibility of Terminal and Frontend/Backend server connectivity. The definitions and explanations related to internal communication mechanisms within server/s if multiple servers exist, mentioned in this document are for explanatory purpose and are out of the scope of the specification.

3.1. Communication Capability

This section describes the physical communication capability of the terminal and the Frontend/Backend server. The communication link specifies the communication media over which the data is to be transmitted. This section defines the minimum specification of the hardware media which shall meet the requirement for the Terminal-Frontend/Backend Server interface.

3.1.1. Terminal Communication Capability

The Terminal must have the capability to connect to the server/servers installed in the

system. It must be capable to connect to the Frontend server/Backend by various means of physical interfaces with necessary security mechanisms defined by this specification document.

The following communication modes are acceptable with Terminal:

3.1.1.1. Communication Port

The terminal will be communicating with AFC system and the external world such as Acquirer through suitable communication port. The physical interfaces are governed by various kind of transit ecosystem and this specification document defines a set of physical communication mode where the selection of the mode is decided by the Operator and the compatibility of the terminal and AFC system shall meet the requirement. The terminal must have Ethernet communication interface, and also shall contain other multiple selective communication port which can be paired with suitable part of AFC system. In case of fixed terminals installed on the metro gate, Serial port (RS232) is mandatorily required (for communication with the Gate) in addition to Ethernet Interface. There shall also be other ports depending upon the Operator requirements. In case of mobile terminals, in addition to Ethernet Port, cellular communication is mandatorily required.

For any kind of communication physical interface, the detailed documentation clearly indicating the default setting and the suitable method of change/update of physical channel configuration must be available. In case, the AFC network is established with split data transfer (direct transmission) capability (i.e. financial data is transferred from terminal to Acquirer and transit data only will flow from terminal to AFC system) the terminal may have more than one physical port to reduce the scope of security certification dependency as defined in PCI-DSS as requirement of banking service. But the physical separation of the interface shall be required as per the requirement of Operator and as defined by Acquiring bank. However, the logical interfaces also shall be provided to split the data with a single physical port. Therefore, in case of direct transmission, logical separation of the interface shall be done. The implementation of either type of architecture is Operator specific.

Following are the sets of communication channels any one must be present in Terminal**:**

i. Serial Port (RS232 Port)

The terminal equipped with RS232 compliant serial port, must support a set of de-facto standard data-rates above and including 19200, with all possible configuration of start, stop, parity and data-bits with asynchronous mode of communication.

Ref: TIA/EIA-232-F, Revision F, October 1, 1997.

The DE9 (DB9) connector is a valid interface for this purpose.

ii. USB communication

The terminal equipped with the USB port must meet or exceed USB 2.0 specification and must support the Communication Device Class. Any other class of communication like HID is not applicable for the physical interface.

Ref: Universal Serial Bus Specification, Revision 2.0, April 27, 2000 with ECN for Mini-B Connector Revision D; On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0 version 1.1a, July 27, 2012; Universal Serial Bus Micro-USB Cables and Connectors Specification, Revision 1.01, April 4, 2007.

iii. Ethernet

The Ethernet port of the terminal must be capable of at least 100Mbit/s data rate over twisted pair cables (100BASE-T). Must meet or exceed IEEE Standard 802.3-1998.

Ref: 802.3-1998 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

iv. Wireless LAN (Wi-Fi)

The Wi-Fi communication mode of the terminal (if present) must be compliant with IEEE Std. 802.11 a/b/g/n modes of communication. The terminal must always run in “Station Mode” and never as an Access Point. If for some reason a terminal needs to run in Access Point mode, then the SSID must always be hidden from public and must be visible only to necessary participating entities. The location of the main WiFi adapter/modem must also be strategically chosen so that all the participating entities always receive at least ‘Reliable’ quality signal strength (<67dBm).

Ref: 802.11-2007 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

v. Cellular Communication

The terminal shall implement a cellular data communication link using validated and widely used communication technologies belonging to the second, third, fourth or transitional generations. Care must be taken that the existing cellular infrastructure supports the wireless technology used. All associated standards of the used wireless technology and associated Indian standards must be complied with.

3.1.2. Frontend/Backend Server Communication Capability

The AFC system can communicate with one or multiple number of terminals simultaneously in a very scalable way without interfering the communication of other link of Terminal-Server. The server must be capable to connect to any terminal by various means of physical interfaces with necessary security mechanisms as defined by this specification document.

Ethernet is the primary mode of communication to achieve higher bandwidth, high level of security and robustness in the AFC system. However, Operator and AFC provider shall add further data security layer such as Tokenization, P2P encryption which are not mandated as per current specification.

E.g. When the communication link is established between Frontend to Backend server or inter-server, high speed communication link must be established. The legacy communication port and short-distance communication media must be avoided. RS232-serial ports, USB are such communication mode and those must be avoided if the communication link is not in between terminal to frontend server (or Terminal-AFC system) pair.

The following communication modes must be available with the AFC system:
In case, the frontend/backend server is not having a direct option to facilitate the specific communication media to communicate with the terminal, then a suitable media must be available to convert the communication mode and required protocol standard without compromise, and with security requirements and compatibility of the Terminal.

E.g. The Frontend/backend server (that links with terminal) in the AFC System may come with Ethernet interface only, so in that case:

- Suitable add-on modules in the form of software or hardware system or sub-system must be available with the AFC system.

If the network is designed with split data transfer or direct mode mechanism, there shall be two logical or physical communication channels from terminal, one for AFC system and other for acquirer interface. However, based on specific operation of the terminal, the terminal capability may be configured with limited features, for specific example:
In case a terminal is used only for top-up or recharge purpose to Global-wallet, it shall have direct physical and logical interface with the acquirer but it's interface with the AFC system may only be for the purposes of network management (e.g. if the terminal is malfunctioning, then the AFC remains informed).
In contrast with at least one or multiple communication port requirement at the Terminal, the AFC system must provide all communication either by default package or as the modular fashion by software or hardware sub-modules.

3.1.2.1. Communication Port

The Frontend/Backend server will communicate with the validation Terminal and the external world such as the Banking system, through suitable communication port. The physical interfaces are governed by various kinds of transit ecosystem components. This specification document defines a set of physical communication mode where the selection of the physical interface will be decided by the Operator. The AFC system must have all the physical communication port in direct or indirect modular fashion and that must be ready to be paired with suitable type of Validation Terminal.

Following are the sets of physical communication modes which shall present in the Server or AFC System:

i. Ethernet:

Ethernet communication port must be available as primary communication mode between backend and frontend unit of the AFC system if multiple servers exists. This shall be capable to communicate with a “network compliant Ethernet bus” of at least 100Mbps with IEEE 802 standard. The primary protocol to communicate with the validation terminal as well as internal hierarchical servers must have TCP stack. However, the AFC system provider or the Operator, shall require other type of protocol stack as extra requirement.

ii. Other Interfaces:

Any other mode of communication shall exist as alternate communication port for communication with the terminal. Such interfaces shall exist in the Server for ex- Serial Port, USB Communication, Wireless LAN etc. The specifications for the interfaces are the same as defined in Section 3.1.1.1 of this document.

iii. Interface Conversion Module:

A suitable conversion module must be available in form of system/sub-system made with hardware/software to make suitable bridge between the Terminal and AFC system. Such conversion module shall exist if the terminal has any of the hardware

module such as RS232/USB/3G/4G/GPRS etc.

3.2. Security Requirements

3.2.1. Communication Security

This section describes the security mechanism related to the channel and the data.

3.2.1.1. Channel Security

AFC system has to ensure the security of the channel through which the Information will be transmitted. Ethernet may be the primary medium which communicates with terminal and with entire internal network of the AFC Ecosystem. The communication channel between all the units (i.e. the Terminal, Frontend /Backend Server) must be secured.

The specification mandates both the entity verification (i.e. Mutual Authentication). The authentication check must be performed by both entity (i.e. the client and the server) at first time of communication establishment. If communication is broken for any physical or logical reason and new session is started with the new TCP link establishment, the authentication verification process must be performed before transferring any type of data at starting of each session.

The Communication Channel Security ensures transfer of sensitive data without overhearing and tampering. To transmit the transaction data, the secure connection initiation (TLS Handshake Protocol) and data transmission (TLS record protocol) shall be used.

If the communication channel does not implement Ethernet mode and mentioned SSL/TLS security, suitable alternate secure communication mechanism must be implemented (such as P2PE, Tokenization). In either case, the whole system must comply with latest stable version of PCI-DSS Implementation.

Ref: TLS ver.1.2 and SSL ver. 3.0 Protocol Standard or latest stable version.

3.2.1.2. Data Security

Encryption of data ensures confidentiality and data security of the sensitive data. It can be done with any type of standard cryptographic algorithms or by Tokenization. It must

be ensured that the data is in encrypted form before transferring through network.

In direct mode communication, this document does not mandate any security standard for AFC system but terminal must be PCI-DSS standard complied. For direct communication mode, if Card Holder Sensitive data is not transferred to AFC system, the data security is optional for Terminal-AFC system link.

In indirect communication, the complete AFC system along with the terminal must be PCI-DSS security standard complied.

It is to be noted that the scoping of PCI-DSS standard shall be defined by Acquiring bank as per the requirements of Operator.

Ref: PCI Data Security Standard, Version 3.2 or latest stable version, PCI Payment Application Data Security Standard, Version 3.2 or latest stable version.

3.3. Hardware Security

The terminal must implement permanently active tamper detection mechanisms that monitor for intrusion by means of, (but not limited to) drills, lasers, chemical solvents, opening covers, splitting the casing/seams, and using ventilation openings) and respond to such events with the automatic and immediate erasure of sensitive information (secret or private keys used to protect PIN or card data) within the device and rendering the device inoperative. For transit NCMC specification does not recommend including PIN pads in the terminal and does not allow exposing PIN pads for user accessibility for contactless card based transit operation. The Terminal shall also maintain a log of tamper events including the time of attack and the type of attack.

Ref: PCI Data Security Standard, Version 3.2 or latest stable version.

3.4. Network Availability

Network availability is ensured by both the entity (i.e. sender and the receiver) by sending periodic signals signifying that the network is up and running. Terminal and Frontend/Backend server communicates with each other on the physical network in case of fixed terminals. The data can only be transferred in case of network availability. The receiver will send acknowledgement to the sender in case of receipt of data by the receiver. If any type of interrupt takes place at the network end during data

transmission, then the data shall be transferred from the point of interrupt once the network is resumed. When a response is not received from the other side within a certain period of time due to network failure or otherwise, the sender will retry to transmit data automatically. The timer required to detect timeout and the retransmission count shall be configurable.

Following configurable parameters must be managed at the Terminal and the AFC system**:**

- **Sync Signal:** Sync Signal used to ensure the network connectivity. It must be Configurable from 5 minutes to 60 minutes with 5 minute granularity.
- **Connection Retry Option:** Configurable from 0 to 100 with granularity of 1.
- **Alarm Option:** Configurable Local/Remote alarm as enable/disable for connectivity failure.

3.5. Time Synchronization

All the entity logically and physically linked with the AFC system and sharing or exchanging of data has to be synchronized with standard NTP server. The specification does not indicate any fixed host or public IP, but it is recommended to establish a local NTP server, which shall be linked with at least two public NTP servers as redundant source of time synchronization. The selection of NTP server shall be carried out with common agreement of issuer, Operator and acquirer.

The NTP client must be an integral part of all devices exchanging data. If NTP client is polling the data, the synchronization polling request interval must not be more than 12 hours. The maximum time drift allowed in each NTP client device shall not be more than 2 secs in reference with local or global referred NTP host server.

4. Common Message Structure

This module defines the Common Message Structure for sending/receiving message between the sender and receiver i.e. Terminal-server and Inter-Server. This structure can hold any type of message like Alarm Message, Configuration Message, Audit Message, Transaction Message etc.

TOKEN (H64)	DATE TIME (N12)	KEY (Upto AN15)	DATA (Variable Size ANS String)

Figure 64: Common Message Structure

Given below is the description of the message structure: -

Table 28: Description of Common Message Structure

S. No.	Common Message Parameter	Data Size	Data Format	Descriptions
1	**Token**	H64	Hex String	This field stores the 64-digithex character string value of the data that is being transmitted using this common message structure. In Request messages, Hash value is 32-byte hash of the data and rest 32 bytes are 0's. In Response messages, Hash Value is initial 32 byte of the Token of Request message and rest 32 bytes are hash value of the Data.
2	**Date and Time**	N12	Numeric String	Date Time of the message (YYMMDDHHMMSS)
3	**Key**	Upto AN15	Alpha Numeric String	Refer Table 29
4	**Data**	Depends upon message object(Variable)	ANS	It holds the actual data that is shared between sender and receiver (between different levels of AFC equipment). Refer Table 29

Table 29: Common Message Structure with modules

S. No.	Module Name	KEY in Common Message Structure (String)	Structure of Data(Object)
1	Configuration	CONFIG_REQUEST	ASSOCIATION_DETAILS
		CONFIG_RESPONSE	ASSOCIATION_DETAILS
			CONFIG_CONTENT
2	General message	GENMSG_REQUEST	KEY(CUSTOM STRING)
			VALUE(CUSTOM STRING)
		GENMSG_RESPONSE	KEY(CUSTOM STRING)
			VALUE(CUSTOM STRING)
3	Alarm	ALARM_REQUEST	TERMINAL_ID
			DATE_TIME
			ERROR_CODE
			SUB_ERROR_CODE
		ALARM_RESPONSE	TERMINAL_ID
			DATE_TIME
			ERROR_CODE
			RESPONSE_CODE
4	Transit Transaction	TXN_REQUEST	ASSOCIATION_DETAILS
			TXN_DATA
		TXN_RESPONSE	ASSOCIATION_DETAILS
5	Financial Transaction	TXN_REQUEST	ASSOCIATION_DETAILS
			TXN_DATA
		TXN_RESPONSE	ASSOCIATION_DETAILS
			ACK_DATA

The above table shows the common Message structure. This structure is used by all the services for communication from one level of AFC to another level. For transmission of configuration data, alarm data, general message etc common message

structure is used.

Here note down that in packet form, when common message is being transmitted, suitable separator (such as '#') shall be used for separation of data elements.

4.1. Configuration

Configuration is an operating feature of the AFC ecosystem that shall be fully parameterized to provide flexibility for modification of operational parameters / conditions. These changes shall be done at the AFC Backend server and downloaded by the equipment directly or if intermediate server is existing, then the data is downloaded via the intermediate server. There shall be provision to transfer EOD Parameter to one station or all the stations from AFC Backend server. An acknowledgement from the equipment shall be sent to AFC Backend server automatically for the acceptance and validation of new EOD parameters.

Global parameters are maintained at the AFC Backend server and then be distributed to all the validation terminals via AFC Backend servers.

For all the services at the terminal, AFC Backend Server level, the configuration file defines the configuration parameters which are required for the service to execute itself. The structure of configuration is common in the entire system.

AFC Parameters are managed both at the remote and local configuration level. In case of remote configuration, the parameters are governed from the Backend Server level and sent to the Frontend Server or Terminal. In case of local configuration, the parameters are managed at the local system level. The parameters are being generated as per the defined structure in Fig 8 and then the Backend Server sends these parameters to the Frontend Server or to Terminals. The configuration data shall be maintained in any file format i.e. JSON or XML depending upon the Operator implementation. Here we have taken example of Configuration data in JSON format.

4.1.1. Configuration Request

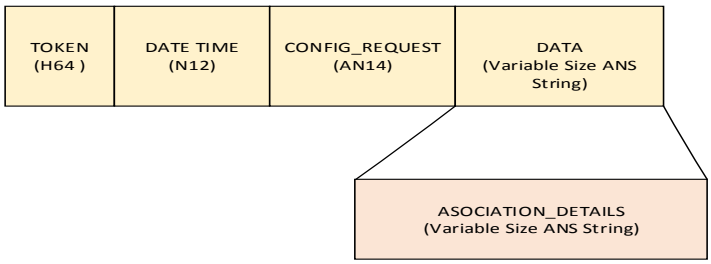


Figure 65: Configuration Request Structure

The detailed description of the Configuration Request structure which is contained in the DATA of the Common Message Structure is as follows:

Table 30: Configuration Request Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	Indicates the association details of requested configuration data. Association details shall be a directory path, file name or DB information.

4.1.2. Configuration Response

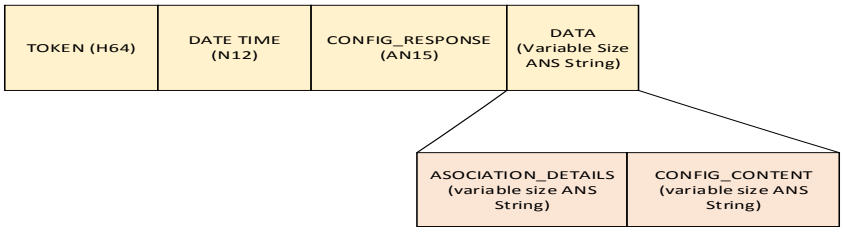


Figure 66: Configuration Response Structure

The detailed description of the Configuration Response structure is as follows:

Table 31: Configuration Response Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	It indicates the association details of requested configuration data. Association details shall be a directory path, file name or DB information.
2	CONFIG_CONTENT	Depending upon the data content(Variable)	ANS	It holds the requested configuration data (payload).

Examples of Configuration Module

Example 1: Configuration request message using CMO

Assumption:

- 2 Global Files (GF) are located on backend server.
- Terminal requires both files to execute some function.
- GF1 - TRM_CONFIG.json (Configuration file required while performing Transit Risk Management and consist of parameters like minimum balance required in card, time limit after crossing of which penalty is imposed, penalty decision etc.

Sample TRM_CONFIG.json

```
{
  "Editable": "true",
  "MINIMUM_BALANCE": "10",
  "TIME_LIMIT": "2",
  "PENALTY_DECISION": "0"
}
```

Figure 67: Sample file GF1 Configuration Request

GF2 - GATE_CONFIG.json (Configuration file required for setting different gate modes)
Sample TRM_CONFIG.json

```
{
        "Editable": "true",
        "MODE": "NORMAL_MODE",
         "EMERGENCY_MODE": "CLOSE",
         "DIRECTION": "ABA",
         "AISLE_MODE": "DISABLE",
         "ENTRY_EXIT_OVRIDE": "DISABLE",
      "TIME_MODE_OVRIDE": "DISABLE"
     "HIGH_SECURITY_MODE": "DISABLE"
}
```


Figure 68: Sample file GF2 Configuration Request

- CMO Parameters Separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Configuration request message generation time at terminal - January 25th' 2020 at 21:15:30

Table 32: Request Message CMO

Parameter	Description	Value as per assumption
DATA	It shall contain association details only since it is configuration request. It contains the file name/s (separated by suitable separator in case requested file count > 1)	TRM_CONFIG.json# GATE_CONFIG.json
KEY	It shall contain the unique key being used for identifying configuration message request.	CONFIG_REQUEST
DATE TIME	It shall contain configuration request message generation date and time	200125211530
TOKEN (MD5[DATA] + 32 0s)	It shall contain 32 bytes hash of DATA field followed by 32 bytes containing 0s since it is request message.	60e5fd7dd9bb1a705ce7f9ca2fcddb400000000000000000000000000000000

Procedure:

- Terminal maintains a local file which consists a list of global files names (GF1 and GF2) to be requested from backend.
- Whenever terminal turns on, it checks for the list of global files names in its local file and sends a request message constructed using CMO.

CMO packet construction(refer Table 32**):**

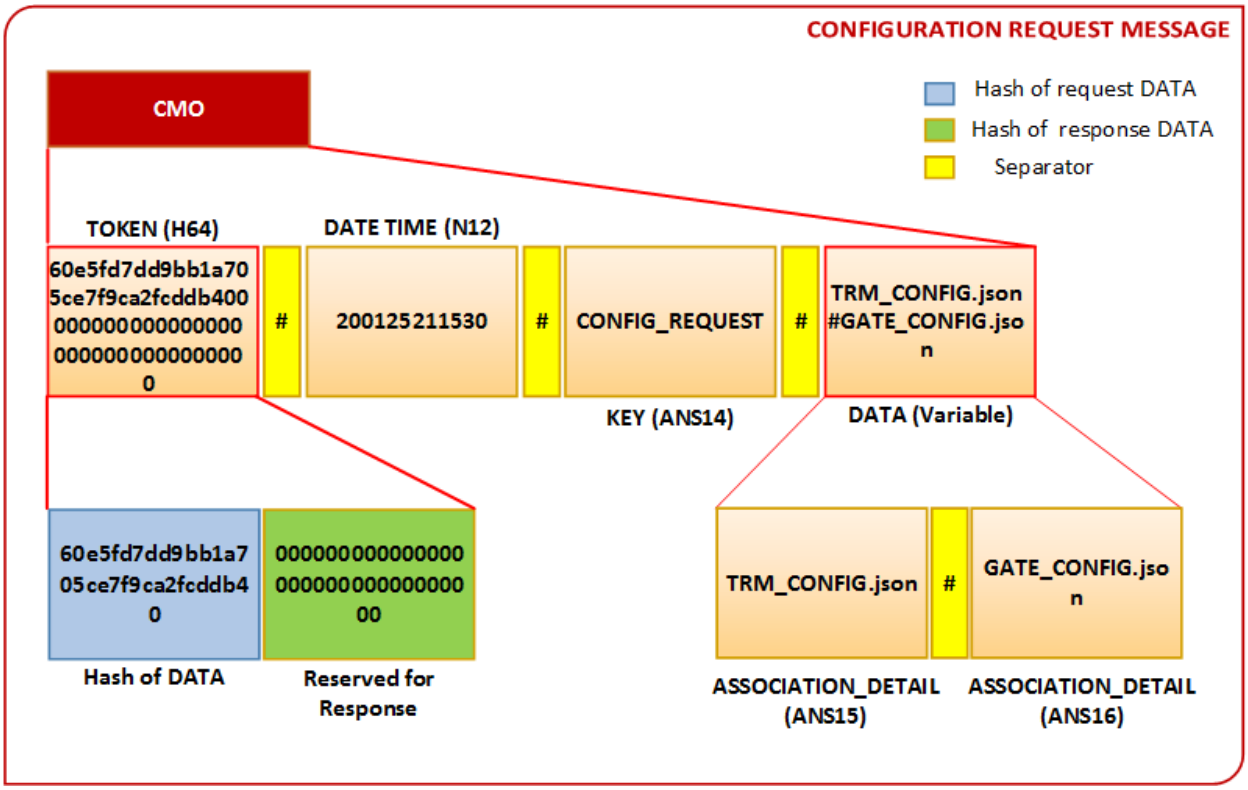


Figure 69: Configuration Request CMO

Example 2: Configuration response message using CMO

Assumption:

- 2 Global Files (GF) arelocated on backend server.
- GF1 - TRM_CONFIG.json (Configuration file required while performing Transit Risk Management and consist of parameters like minimum balance required in card, time limit after crossing of which penalty is imposed, penalty decision etc.
-

Sample TRM_CONFIG.json

```
{
        "Editable": "true",
        "MINIMUM_BALANCE": "10",
        "TIME_LIMIT": "2",
        "PENALTY_DECISION": "0"
}
```


Figure 70: Sample file GF1 – Configuration Response

- GF2 - GATE_CONFIG.json (Configuration file required for setting different gate modes)
-

Sample TRM_CONFIG.json


```
{
        "Editable": "true",
        "MODE": "NORMAL_MODE",
         "EMERGENCY_MODE": "CLOSE",
         "DIRECTION": "ABA",
         "AISLE_MODE": "DISABLE",
         "ENTRY_EXIT_OVRIDE": "DISABLE",
       "TIME_MODE_OVRIDE": "DISABLE"
      "HIGH_SECURITY_MODE": "DISABLE"
}
```


Figure 71: Sample file GF2 – Configuration Response

- CMO Parameters separator - "**#**"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Configuration response message generation time at terminal - January 25th' 2020 at 21:15:32

Table 33: Response Message CMO

Parameter	Description	Value as per assumption
DATA	It shall contain the association detail of requested configuration file followed by configuration file. If requested file count > 1, association detail of requested configuration file followed by configuration file shall be appended	TRM_CONFIG.json# < TRM_CONFIG.json file content># GATE_CONFIG.json# < GATE_CONFIG.json file content>
KEY	It shall contain the unique key being used for identifying configuration	CONFIG_RESPONSE

	message response.	
DATE TIME	It shall contain configuration response message generation date and time	200125211530
TOKEN (Token in request message + MD5[DATA])	It shall contain 32 bytes token present in request message followed by 32 bytes hash of response DATA.	60e5fd7dd9bb1a705ce7f9ca2fcddb408e15995d4befcf45427f01e6acb664f

Procedure:

- Once request is received at backend server from a terminal, it maintains a list of files requested by each terminal. It constructs the response using CMO and transfers the response to terminal in both cases,
  i. When terminal requests for the first time after it turns on,
  ii. Whenever requested file parameters are updated at backend.

CMO packet construction (refer Table 33):

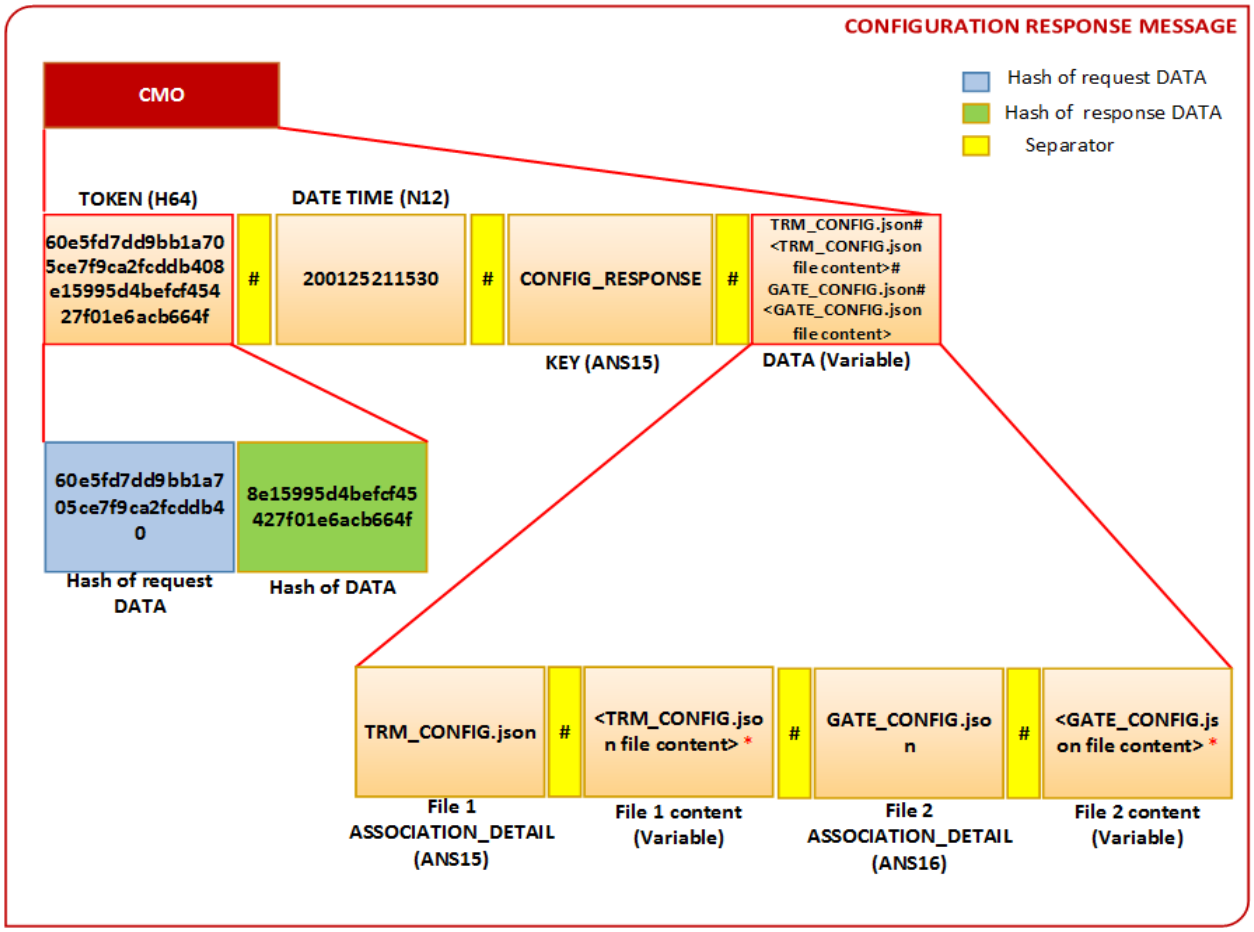


Figure 72: Configuration Response CMO

*Refer sample files Figure 70, Figure 71

4.2. General Message

The General Message Structure holds the information about the general messaging (i.e all messages except Alarm, Transaction and Configuration).The message shall be Audit data, penalty data, etc. It sends message to server after filling General message data structure in Common Message structure. The data in the General Message shall be maintained in any of the mentioned file format i.e. JSON or XML format. Here we have taken example of Audit Data which is in JSON format.

4.2.1. General Message request

The structure of general message request is as follows:

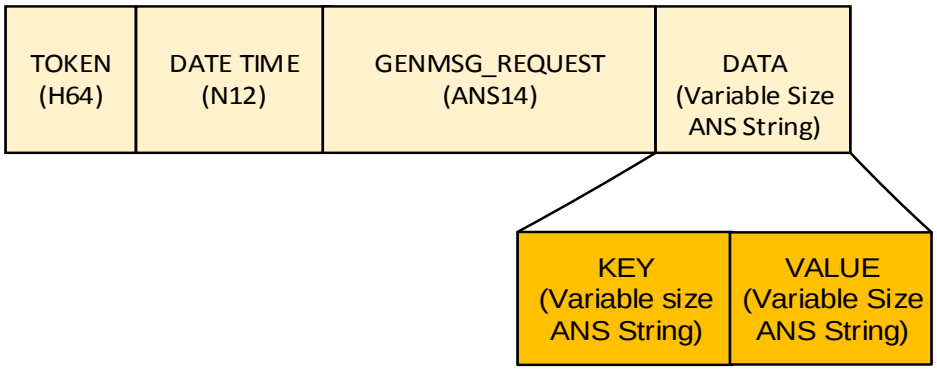


Figure 73: General Message Request Structure

4.2.2. General Message Response

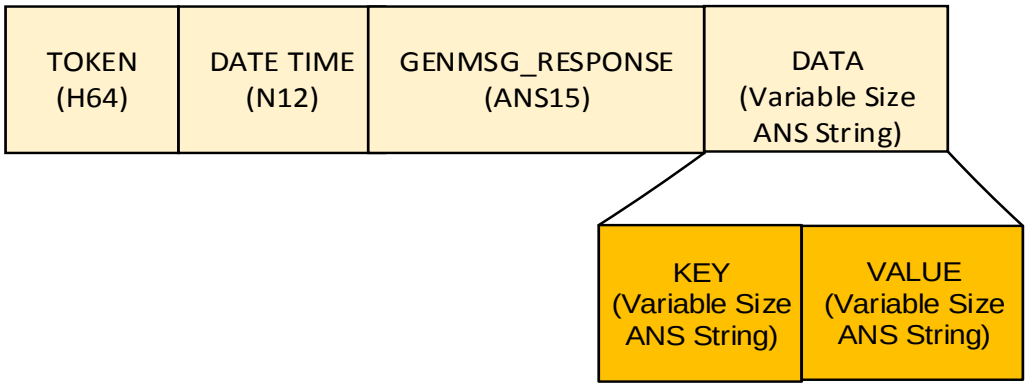


Figure 74: General Message Response Structure

The detailed description of General Message structure is as follows: -

Table 34: General message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	KEY	Variable size, depending upon the key used	ANS	The key describes the module name and the request or response. For ex- To send audit request data, Key can be “AR_REQUEST” (i.e when Audit data sent from terminal to server) and “AUDIT_FILE_RECEIVED”is sent back to terminal.

2	VALUE	Variable size	ANS	It holds actual data. It shall be pass data or audit data etc. Here the Value can be nested with further Key Value data. Therefore this structure can be used to transmit any number of data in the packet.

Examples of General Message

Example 3: General Message request using CMO

Assumption:

- Audit data is being generated at terminal with terminal info “010102001081” and being transferred to Backend server.
- General message is being used to send Audit data.
- Key used to identify that audit data is being transferred - AR_REQUEST

Sample Audit File

```
{
        "TERMINAL_INFO":"010102001081",
        "TIME":"191225143208",
        "CC_terminal":
        {
        "Time_Penalty":6,
        "Pass_Created":1,"Double_Entry":1,"Double_Exit":"3"
        }
}
```

Figure 75 : Sample file Audit Request

- Key used to identify that general message is being used to transfer audit data - GENMSG_REQUEST
- CMO Parameters Separator - “#”
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Audit data generation time at terminal - December 25th’ 2019 at 14:32:08

Table 35: Audit Request Message CMO

Parameter	Description	Value as per assumption
DATA	General Message DATA parameter holds KEY followed by suitable separator and VALUE. KEY defines the message type being sent through general message	AR_REQUEST#<Audit file content>

	VALUE contains the actual data	
KEY	It shall contain the unique key being used for identifying general message request.	GENMSG_REQUEST
DATE TIME	It shall contain general message request generation date and time	191225143208
TOKEN (MD5[DATA] + 32 0s)	It shall contain 32 bytes hash of DATA field followed by 32 bytes containing 0s since it is request message.	5999bccb051db3c5bd9ebdb6fcd2b4240 00000000000000000000000000000000

Procedure:

- Whenever terminal generates audit file at configurable time, it sends the audit constructed using CMO.

CMO packet construction (refer Table 35):

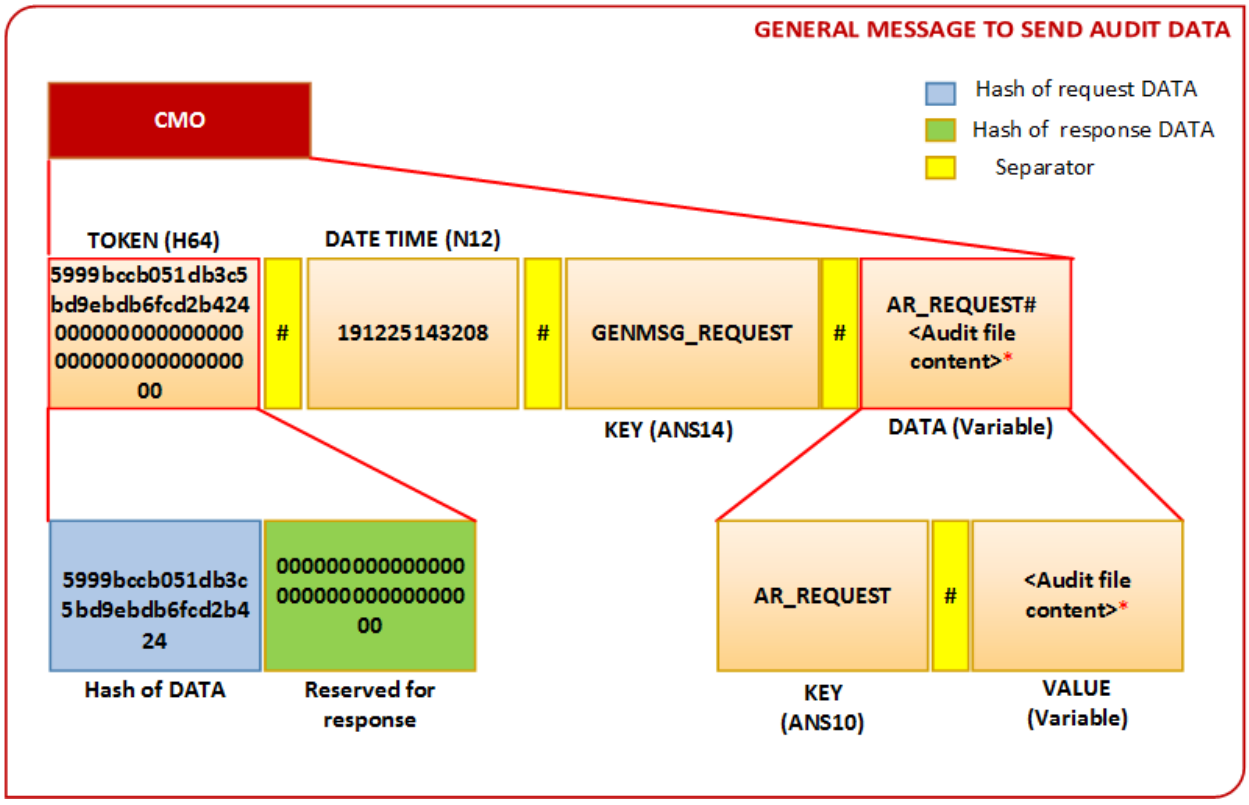


Figure 76: Audit Request CMO

*Refer sample file in Figure 75

Example 4: General Message response using CMO

Assumption:

- Audit data sent by terminal is received at Backend server.
- General message is being used to send Audit data response.
- Key used to identify audit data response - AR_RESPONSE
- Value in case audit data is received successfully - AUDIT_FILE_RECEIVED
- Key used to identify that general message is being used to transfer audit data response - GENMSG_RESPONSE
- CMO Parameters Separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Audit data response generation time at terminal - December 25th' 2019 at 14:32:09

Table 36: Audit Response Message CMO

Parameter	Description	Value as per assumption
DATA	General Message DATA parameter holds KEY followed by suitable separator and VALUE. KEY defines the message type being sent through general message VALUE contains the acknowledgement of request message	AR_RESPONSE# AUDIT_FILE_RECEIVED
KEY	It shall contain the unique key being used for identifying general message response.	GENMSG_RESPONSE
DATE TIME	It shall contain general message response generation date and time	191225143209
TOKEN (Token in request message + MD5[DATA])	It shall contain 32 bytes token present in request message followed by 32 bytes hash of response DATA.	5999bccb051db3c5bd9ebdb6fcd2b424

Procedure:

- Whenever backend server receives general message request, it sends the acknowledgement in general message response constructed using CMO back to terminal.

CMO packet construction (refer Table 36):

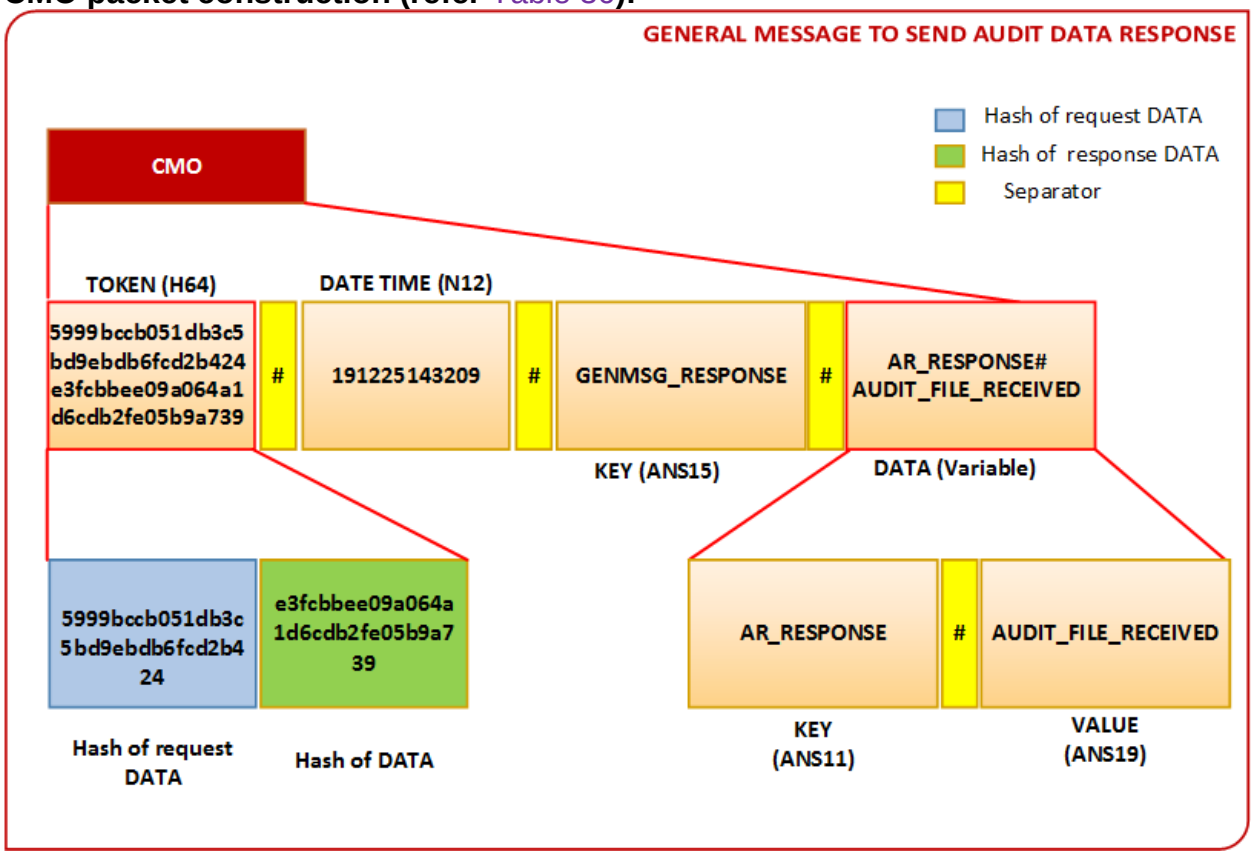


Figure 77: Audit Response CMO

4.3. Alarm/Event

Alarms and Event are managed by the AFC system in case of any faults or failure occurs at the Terminal or in the AFC Backend or the intermediate server if it exists. It defines the responsibility of terminal/AFC Backend/Frontend server and generates Event/Alarm message. If any fault/failure occurs, then a message is being sent to the AFC Backend from Terminal directly. If front end server exists, then message is send to backend server via front end server for resolving the error to maintain smooth transit operation of AFC system. After receiving alarm message by Frontend Server/Backend Server, it generates the alarm response message and sends it back to the alarm source. Alarm source receive response message and take appropriate action to resolve the issue. The Alarm data shall

be in json or xml format. Here we have taken example of alarm in json format.

There are many reasons due to which Event/Alarm will be generated, each reason must have a unique ID (i.e. error code).

Few of the possible reasons of Alarm /Event are listed below:

1. Validation Terminal's hardware not working properly
2. Invalid Entry/Exit station (i.e. pass card)
3. Card Read/Write failure etc.
4. Tailgating
5. Wrong Entry/Exit
6. Flap Failure
7. Forceful cabinet Open/Close
8. Network connection Loss
9. Data Storage Alarm
10. Maintenance Door Open/Close.

4.3.1. Alarm/Event Request

Data Structure of Alarm/Event Request which is contained in the DATA of the Common Message Structure for remote communication are shown below:

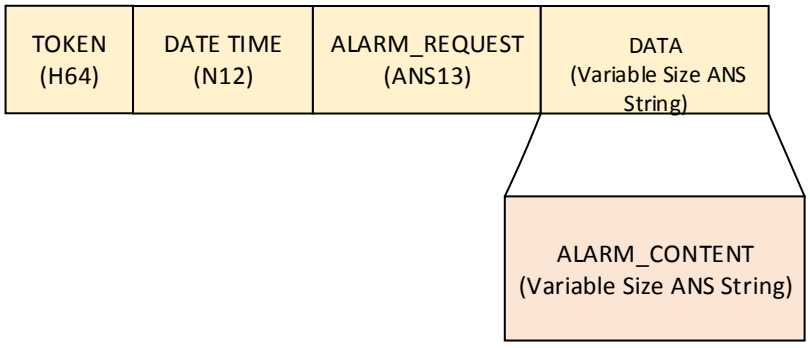


Figure 78: Alarm/Event Request Structure

The detailed description of the Alarm Request structure is mentioned below: -

Table 37: Alarm Event Request Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ALARM_CONTENT	Variable	ANS	It holds the alarm data in the form of a file which shall be either json or xml.

Table 38: Structure of ALARM_CONTENT

S. No.	Alarm/ Event Parameter	Data Size	Data Format	Descriptions
1	Operator Specific Terminal info	Upto H12	Hexadecimal String	The information about the terminal which generates the alarm. For detailed Structure Kindly refer Table 43.
2	DATE TIME	N12	Numeric String	Date Time of Alarm/Event Generation (YYMMDDHHMMSS)
3	ERROR CODE	N2	Numeric String	Error Code pertaining to the alarm which denotes the alarm error.
4	SUB ERROR CODE	N2	Numeric String	Sub Error Code of Error which denotes the alarm error generated at the terminal.

4.3.2. Alarm/Event Response

Data Structure of Alarm/Event Response which is contained in the DATA of the Common Message Structure for remote communication are shown below.

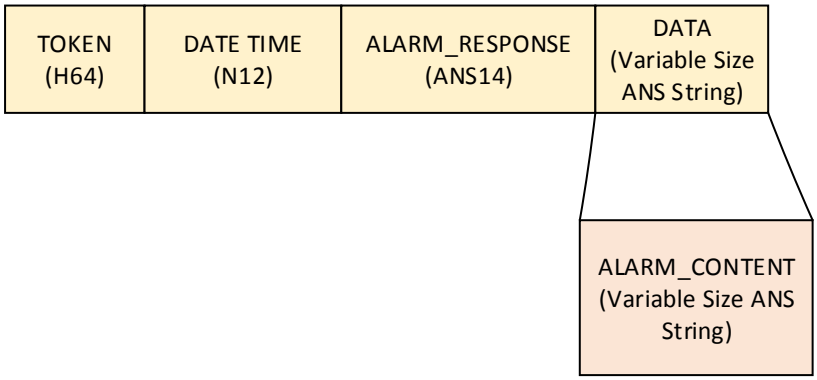


Figure 79: Alarm/Event Response Structure

The detailed description of the Alarm Response structure is mentioned below:-

Table 39: Alarm Event Response Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ALARM_CONTENT	Variable	ANS	It holds the alarm data in the form of a file which shall be either json or xml.

Table 40: Structure of ALARM_CONTENT

S. No.	Alarm/ Event Parameter	Data Size	Data Format	Descriptions
1	Operator Specific Terminal info	Upto H12	Hexadecimal String	The information about the terminal which generates the alarm. For detailed Structure Kindly refer Table 43.
2	DATE TIME	N12	Numeric String	Date Time of Alarm/Event Response
3	ERROR CODE	N2	Numeric String	Error Code pertaining to the alarm which denotes the alarm error.
4	RESPONSE CODE	N2	Numeric String	Response code of error which denotes the action to be taken by the terminal.

All Alarms shall be categorized and configurable into local alarm or remote alarm or local and remote alarm. Local alarm means an alarm which is not required to be sent to the server. The Response code is generated at the terminal itself.
However, the remote alarm will be sent to the server from where the response or corrective action will be generated and sent back to the alarm generating device (terminal).
Alarms shall also be local and remote both. That means the alarm is generated at the terminal and then will be sent to the server. The Response code will be generated either at the server end or the terminal (at the alarm generating device).

Examples of Alarms

Example 5: Alarm request using CMO

Maintenance Door of gate is suddenly opened

In case when someone forcefully opens the maintenance panel of the gate in that case terminal's alarm application received a unique signal (i.e. input that indicate gate panel is open) from the gate via GCU (Gate Control Unit) and terminal's alarm application generates "maintenance door open" alarm message and send it to AFC Backend.
When maintenance gate panel is open, terminal must generate the alarm message (as given below) and send it to the AFC Backend after filling common message structure.
In the below case if Error code is 08 and Sub Error Code is 03 then it denotes that the alarm generated is Maintenance door open.
Assumption:

- Alarm is generated at terminal with Terminal Info "010102001081" and transferred to Backend server.

Sample File


```
{
 "TERMINAL_INFO":"010102001081",
 "TIME":"191225143206",
 "Error_Code":"08 ",
 "Sub_Error_Code":"03"
 }
```


Figure 80: Sample file Alarm Request

- Key used to identify that alarm data is being transferred - ALARM_REQUEST
- CMO Parameters Separator - "**#**"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Alarm data generation time at terminal - December 25th' 2019 at 14:32:06

Table 41: Alarm Request Message CMO

Parameter	Description	Value as per assumption
DATA	It holds the alarm data	<Alarm request file content>
KEY	It shall contain the unique key being used for identifying alarm request.	ALARM_REQUEST
DATE TIME	It shall contain alarm request generation date and time	191225143206
TOKEN (MD5[DATA] + 32 0s)	It shall contain 32 bytes hash of DATA field followed by 32 bytes containing 0s since it is request message.	a0e52b9a21821db2b4e5db10def2e5c300000000000000000000000000000000

Procedure:

- Whenever alarm is generated, it constructs the request message using CMO and sends to backend server.

CMO packet construction (refer Table 41):

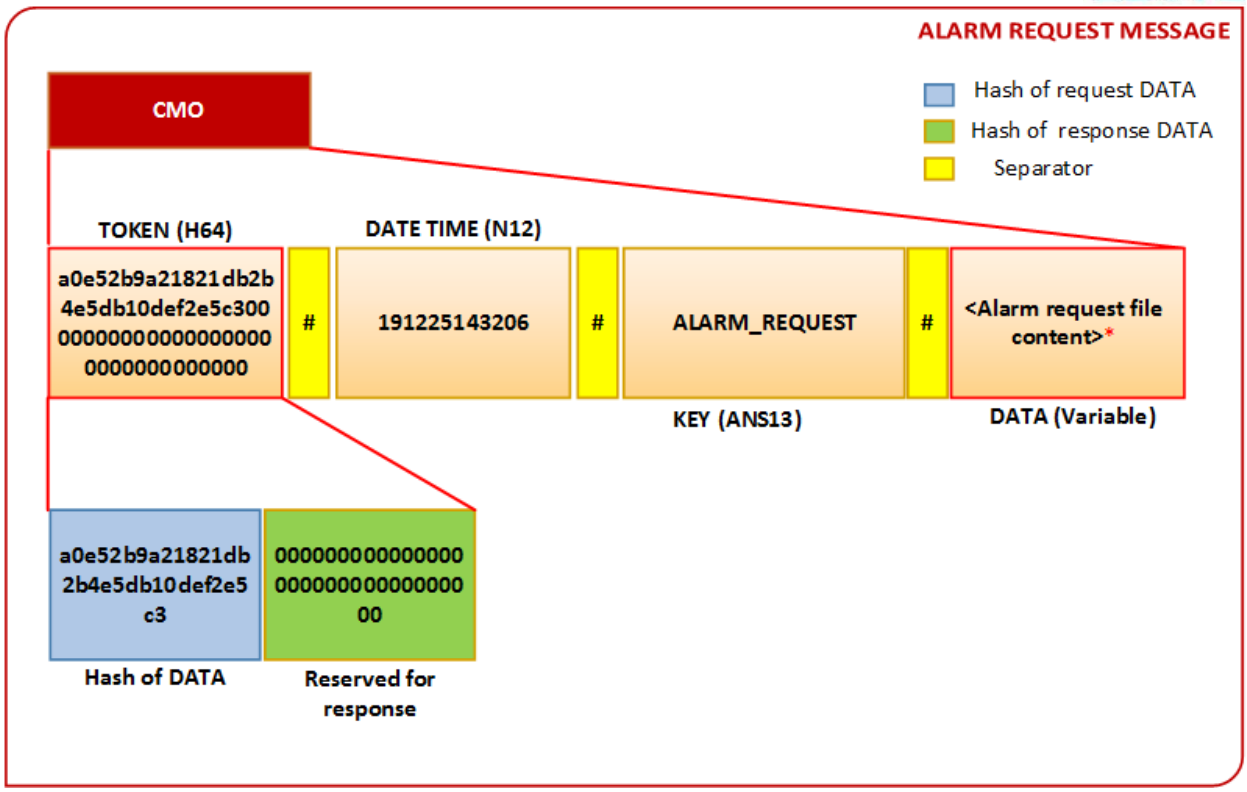


Figure 81: Alarm Request CMO Structure

*Refer sample file in Figure 80

Example 6: Alarm response using CMO

When AFC Backend receive the alarm message that comes from the terminal, AFC Backend checks the Error code and Sub Error Code, on the basis of the value it generates appropriate Alarm response message.

In Above alarm message, value of error code is “08” and value of sub Error code is “03”. AFC Backend read the meaning of code from the configuration file (stored at AFC Backend side).

For e.g. 08 means Gate access monitoring (written in configuration file)
03 means Maintenance door open (Written in Configuration file)

On the basis of above alarm message, AFC Backend generate the response message (given below) and sends it to terminal: -

Assumption:

- Alarm data sent by terminal is received at Backend server.

Sample File

```
{
```

```
"TERMINAL_INFO":"010102001081",
"TIME":"191225143208",
"Error_Code":"08 ",
"Response_Code":"13"
}
```


Figure 82: Sample file Alarm Response

- Key used to identify audit data response - ALARM_RESPONSE
- CMO Parameters Separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Alarm data response generation time at terminal - December 25th' 2019 at 14:32:08

Table 42: Alarm Response Message CMO

Parameter	Description	Value as per assumption
DATA	It holds the alarm data response	<Alarm response file content>
KEY	It shall contain the unique key being used for identifying alarm response.	ALARM_RESPONSE
DATE TIME	It shall contain alarm response generation date and time	191225143208
TOKEN (Token in request message + MD5[DATA])	It shall contain 32 bytes token present in request message followed by 32 bytes hash of response DATA.	a0e52b9a21821db2b4e5db10def2e5c35bbc5ab87b84d6e2071d0241f3f7899d

Procedure:

- Whenever backend server receives alarm request, it sends the acknowledgement in alarm response constructed using CMO back to terminal.

CMO packet construction (refer Table 42):

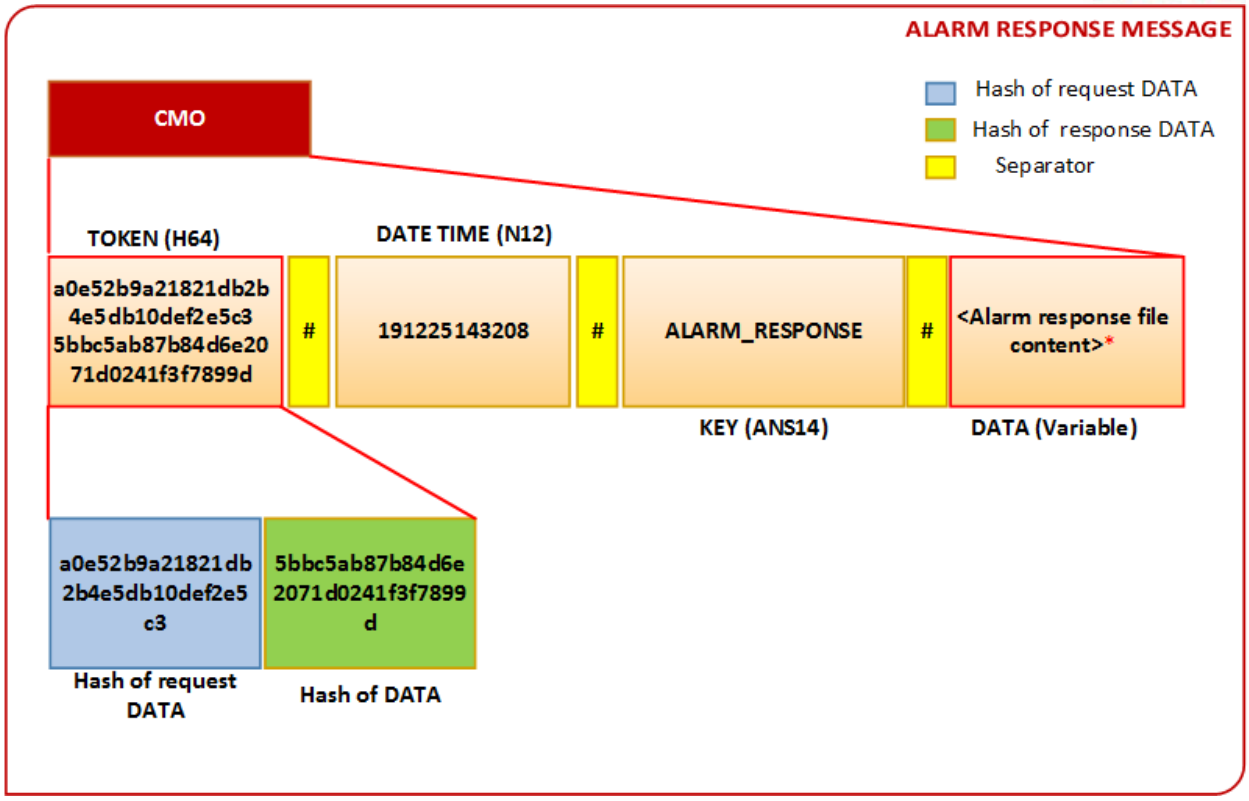


Figure 83: Alarm Response CMO Structure

*Refer sample file in Figure 82

When AFC Backend sends above response message to terminal, terminal read the value of error code and sub error code from configuration file (i.e. store inside the terminal) and performed action accordingly.

If terminal's configuration file contains: -

08 means Gate access monitoring (written in configuration file)
13 means Go to Maintenance Mode (written in configuration file)

After reading the actual meaning of code from configuration file, terminal sends the Maintenance mode command to gate via GCU.

4.4. Transaction

When a successful transaction happened at the terminal, the transaction data is generated. There are two types of transaction data generated at the terminal, Transit and Financial. The transit transaction data will be transmitted from the terminal to AFC System. The Financial transaction data will be transmitted from terminal to Acquirer (in Direct mode) and terminal to AFC System (in indirect mode). The transit transaction data shall be in any of the mentioned file formats i.e. JSON or XML format. The

Financial transaction data is mentioned in Part VI: AFC Ecosystem - Acquirer Interface.

Here in this document we have taken example of transit transaction data in xml format.

4.4.1. Transaction Data

The transaction data (TXN_DATA) contains two categories of data elements-

a) Common data elements (Common for all media types)
b) Variable data elements (Vary from one media to another)

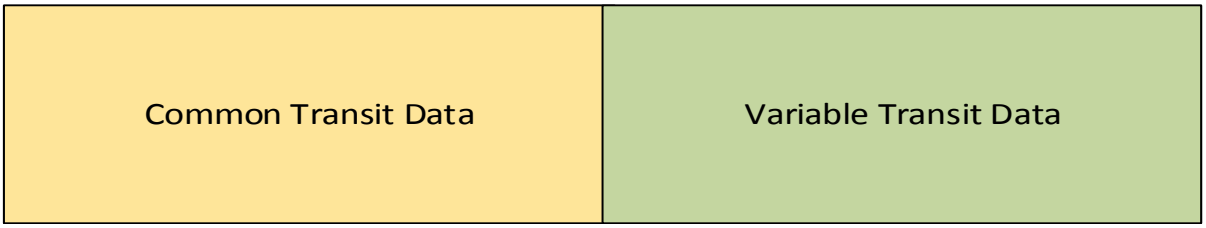


Figure 84: Transaction Data Structure

4.4.2. Common data elements (Common for all media types)

These data elements which are common for all media types are called common data elements.

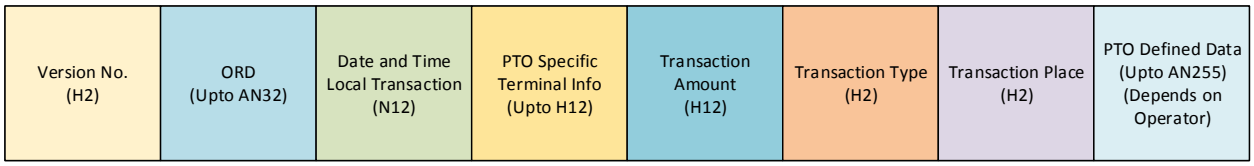


Figure 85: Common Transit Data Elements

The description of the common transit data elements is mentioned below:-

Table 43: Common Transit Data Elements for all Media Types

S. No.	Field	Data Format	Size	Xml Tags	Remarks
1	Version no.	Hexadecimal string	H2	<nVERSION>	It represents the version of the common data layout. Version is 1.0 and represented as 10.

2	ORD	Alpha Numeric	Upto AN32	<nORD>	Operator Reference Data. This element is a unique reference number generated for all transactions.(ORD to consist of Acquirer ID(H2) + Operator ID(H4) + Terminal ID(H6) + Media Type (H1) + Julian Date(N6 YDDDHH) + STAN(N6) + LRC(H2)
3	Date and Time Local Transaction	Numeric	N12	<nDtTmLcTxn>	The date & time of the transaction in YYMMDDHHmmss
4	Operator Specific Terminal info	Hexadecimal string	Upto H12	<nTrmIdOp>	Unique code identifying a terminal managed by Acquirer and Operator.(Acquirer ID [H2] + Operator ID [H4] + Terminal ID [H6])
5	Transaction Amount	Hexadecimal String	H12	<nAmtTxn>	It carries the amount in the transaction currency.
6	Transaction Type	Hexadecimal String	H2	<nTxnTypQ>	Indicates the type of transaction takes place. Required for Transit Operation
7	Transaction Place	Hexadecimal String	H2	< nTxnPlace>	Indicate the place where the transaction takes place at the card accepter location.
8	Operator Defined	Depends on Operator	Up-to AN255	<nOpData>	Data specific to Operator. Vary from one Operator

	Data				to another.

4.4.3. Variable data elements (Varies from one media type to another)

These data elements vary from one media to another. Different media type can be NCMC, QR, Token, NFC etc.

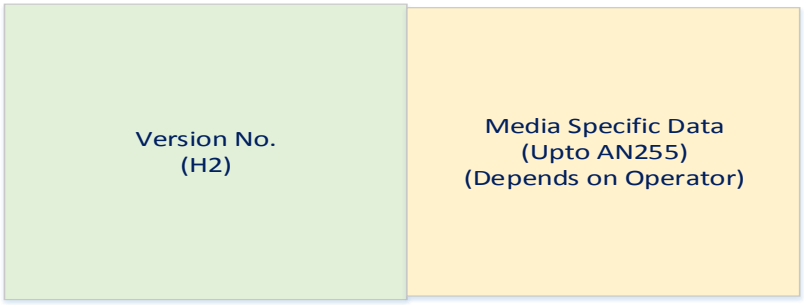


Figure 86: Variable Transit Data Elements

The description of the variable transit data elements is mentioned in the table below.

Table 44: Variable Transit Elements

S. No	Field	Data Format	Size	XML Tags	Remarks
1	Version no	Hexadecimal string	H2	<nMVERSION>	It represents the version of the data specific to the service. Version is 1.0 and represented as 10.
2	Media Specific Data	Depends on Operator / media type	Variable		Data specific to Operator and media type which may be of varying length.

4.4.3.1. Media specific data elements in case of NCMC

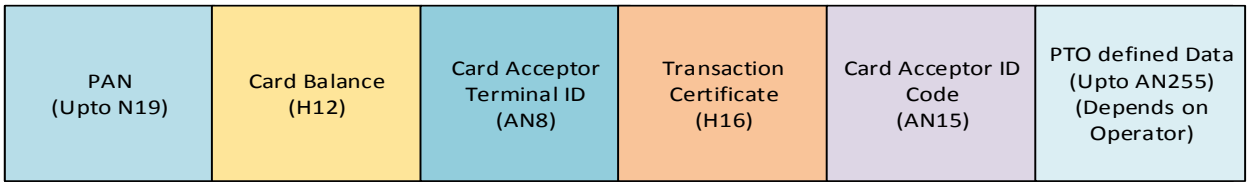


Figure 87: Variable Transit Data Elements (NCMC Specific)

Table 45: Media specific Transit Data Elements for NCMC

S. No.	Field	Data Format	Size	XML Tags	Remarks
1	PAN	Numeric	Upto N19	<nPAN>	Identifies the Cardholder' PAN. The PAN in the transit data must be in tokenized or masked form.
2	Card Balance (Global Wallet)	Hexa decimal String	H12	<nCardBAL>	It represents value corresponding to amount available inside card before transaction Happened on terminal.
3	Card Acceptor Terminal ID	Alpha Numeric	AN8	<nCrdAcptTrmID>	This element depicts the unique code assigned to a terminal at the card acceptor location (TID) by the Acquirer
4	Transaction Certificate	Hexadecimal String	H16	<nTC>	Cryptogram that is generated by signing data elements when the card approves the payment for clearing and settlement is known as

					the Transaction Certificate (TC).
5	Card Acceptor ID Code	Alpha numeric	AN15	<nCrdAcpIDCd>	This element uniquely identifies the merchant id in an acquiring system (MID).
6	Operator Defined Data	Depends on Operator	Up-to 255 char	<nPTOInfo>	Data specific to Operator.

4.4.3.2. Media Specific Data Elements in case of QR

The specification for QR ticketing shall be followed.

4.4.3.3. Media Specific Data Elements in case of Token

The specification for Token based ticket shall be followed.

4.4.4. Transaction File

Transaction file shall contain the common and variable data elements specific to all media types. Transaction file shall be generated in two ways:-

I. **Single file single transaction:** The transaction file contains a single transaction of single media type. The transaction file will contain either card based transactions or QR based transactions or any other media type.

II. **Single file multiple transactions:** The transaction file contains multiple transactions in a single file. For e.g. a file containing NCMC based, QR based and other media based transactions.

The transit file structure is as follows: -

Table 46: Transit File Structure

File format	File Structure			
XML	Header(<hdr>)	Sequence no. for file identifier		
		Version number of structure		
		Generation time		
	Transaction Block	Transaction 1	Common Transit Elements	Structure of Common Transit Element defined in Table 43
			Variable Transit Elements	Structure of Variable Transit Element defined in Table 44
		Transaction 2	Common Transit Elements	Structure of Common Transit Element defined in Table 43
			Variable Transit Elements	Structure of Variable Transit Element defined in Table 44

		Transaction n	Common Transit Elements	Structure of Common Transit Element defined in Table 43
			Variable Transit Elements	Structure of Variable Transit Element defined in Table 44
	Trailer	No of Records		
		Total Amount		

Description of Header part is as follows: -

Table 47: Transit Header Data Elements

S. No.	Field	Data Format	Size	XML Tags	Remarks
1	Sequence no. for file identifier	Numeric	N6	<nFlSeqno>	Identifies sequence number of the file
2	Version no of the structure	Hexadecimal string	H2	<nFlVERSION>	It represents the version of the structure of the file. Version is 1.0 and represented as 10.
3	Generation time	Numeric	N12	<nFlDtTmLcTxn>	The date & time of the generation of the file in YYMMDDHHmmss

Details of the Trailer part is as follows: -

Table 48: Transit Trailer Data Elements

S. No.	Field	Data Format	Size	XML Tags	Remarks

1	No of Records	Numeric	N3	<nNoRecords>	Identifies no of records in the file
2	Total Amount	Hexadecimal string	H12	<nTotalAmt>	It is the total amount claimed in the transaction file

Detailed xml Transaction file Structure for single transaction in a single file: -

Refer to file in Figure 90

Detailed xml Transaction file Structure for multiple transaction in a single file:

```
<?xml version="1.0" encoding="UTF-8"?>
<File>
<hdr>
<nFlSeqno></nFlSeqno>
<nFlVERSION></nFlVERSION>
<nFlDtTmLcTxn></nFlDtTmLcTxn>
</hdr>
<txnblock>
<txn RecNum=”1”>
<!--Subblock for Common Data Element-->
<nVERSION></nVERSION>
<nORD></nORD>
<nTrmIdOp></nTrmIdOp>
<nAmtTxn></nAmtTxn>
<nDtTmLcTxn></nDtTmLcTxn>
<nTxnTypQ></nTxnTypQ>
<nTxnPlace></nTxnPlace>
<nOpData></nOpData>
<nCard>
<!--Subblock for NCMC card specific data-->
<nMVERSION></nMVERSION>
```

```
<nTC></nTC>
<nCrdAcptTrmId></nCrdAcptTrmId>
<nCrdAcpIDCd></nCrdAcpIDCd>
<nPAN></nPAN>
<nCardBAL></nCardBAL>
<nPTOInfo></nPTOInfo>
</nCard>
</txn>

<txn RecNum=”2”>
<nVERSION></nVERSION>
<nTrmIdOp></nTrmIdOp>
<nAmtTxn></nAmtTxn>
<nDtTmLcTxn></nDtTmLcTxn>
<nTxnTypQ></nTxnTypQ>
<nTxnPlace></nTxnPlace>
<nORD></nORD>
<nOpData></nOpData>
<nCard>
<!--Subblock for NCMC card specific data-->
<nMVERSION></nMVERSION>
<nTC></nTC>
<nCrdAcptTrmId></nCrdAcptTrmId>
<nCrdAcpIDCd></nCrdAcpIDCd>
<nPAN></nPAN>
<nCardBAL></nCardBAL>
<nPTOInfo></nPTOInfo>
</nCard>
</txn>
   .
   .
<txn RecNum=”n”>
```

```
<nVERSION></nVERSION>
<nTrmIdOp></nTrmIdOp>
<nAmtTxn></nAmtTxn>
<nDtTmLcTxn></nDtTmLcTxn>
<nTxnTypQ></nTxnTypQ>
<nTxnPlace></nTxnPlace>
<nORD></nORD>
<nOpData></nOpData>
<nCard>
<!--Subblock for NCMC card specific data-->
<nMVERSION></nMVERSION>
<nTC></nTC>
<nCrdAcptTrmId></nCrdAcptTrmId>
<nCrdAcpIDCd></nCrdAcpIDCd>
<nPAN></nPAN>
<nCardBAL></nCardBAL>
<nPTOInfo></nPTOInfo>
</nCard>
</txn>
</txnblock>
<trl>
<nNoRecords></nNoRecords>
<nTotalAmt></nTotalAmt>
</trl>
</File>
```

4.4.5. Transmission of Transaction File using CMO

Transaction files can be transmitted to the AFC System either individually or in batches. Terminal sends transaction file called Transaction Request contained in the DATA element of the CMO as shown in the figure below.

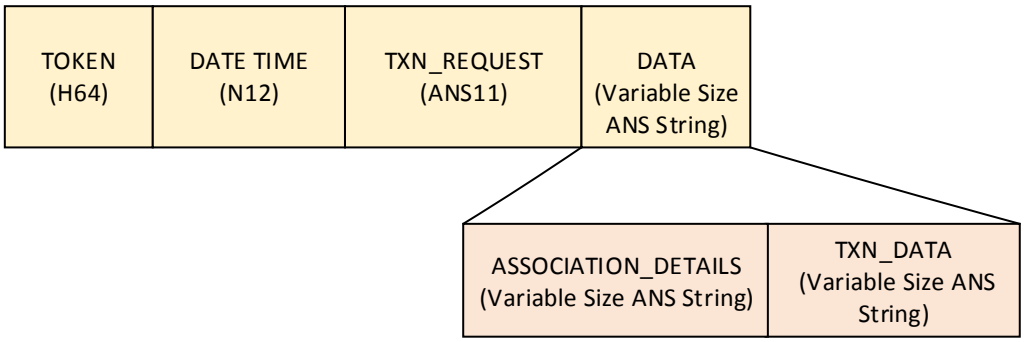


Figure 88: Transaction Request Structure

Transaction request message structure is described in the table below.

Table 49: Transaction Request Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	It indicates the association details of the transaction file. Association details shall be the File Name.
2	TXN_DATA	Depending upon the data content(Variable Size)	ANS	It holds the Content of the Transaction file which shall be in either in xml or json format.

AFC System sends the acknowledgement of received transaction request in a Transaction Response. The response is contained in the DATA element of the CMO as shown in the figure below.

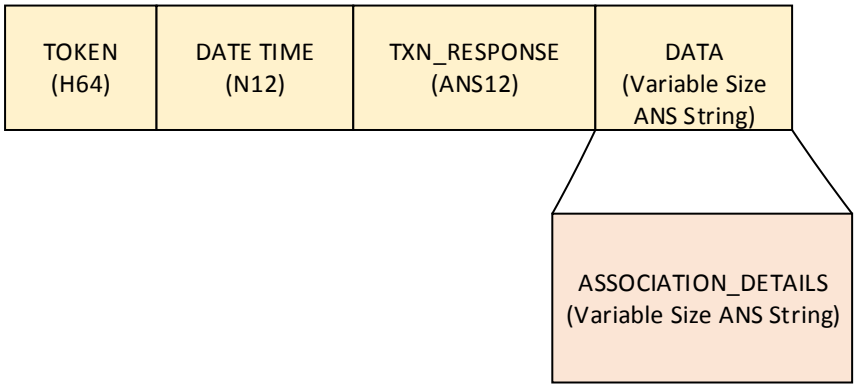


Figure 89: Transaction Response Structure

Transaction response message structure is described in the table below.

Table 50: Transaction Response Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	It indicates the association details of the transferred file. Association details shall be the File Name.

Examples of Transaction File transmission

Example 7: Transaction request message using CMO

Assumption:

- Single transit file is sent each time.
- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Transit file request message generation time at terminal - December 25th' 2019 at 14:29:06
- Key used to identify transaction request message - TXN_REQUEST
- Association details -tA1123120190207170314773456903890381700000062.xml (Name of transit file to be sent in request message)

Sample file

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<File>
<hdr>
<nFlSeqno>000001</nFlSeqno>
<nFlVERSION>10</nFlVERSION>
<nFlDtTmLcTxn>191225142906</nFlDtTmLcTxn>
</hdr>
<txnblock>
<txn RecNum="1">
<!--Subblock for Common Data Element-->
<nVERSION>10</nVERSION>
<nTrmIdOp>010102001081</nTrmIdOp>
<nAmtTxn>000000000100</nAmtTxn>
<nDtTmLcTxn>191225143032</nDtTmLcTxn>
<nTxnTypQ>01</nTxnTypQ>
<nTxnPlace>01</nTxnPlace>
<nORD>01010200108119359140019911 2</nORD>
<nOpData>00001100000000000000000012</nOpData>
<nCard>
<!--Subblock for NCMC card specific data-->
<nMVERSION>10</nMVERSION>
<nTC>9f12d456b23c43d4</nTC>
<nCrdAcptTrmId>A1123110</nCrdAcptTrmId>
<nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>
<nPAN>608326xxxxxx0015</nPAN>
<nCardBAL>000000001000</nCardBAL>
<nPTOInfo>00001100000000000000000023</nPTOInfo>
</nCard>
</txn>
</txnblock>
<trl>
<nNoRecords>001</nNoRecords>
<nTotalAmt>000000000100</nTotalAmt>
</trl>
</File>
```

Figure 90: Sample file Transaction Request

Table 51: Transaction Request Message Structure

Parameter	Description	Value as per assumption
DATA	It shall contain association detail followed by transit file data	tA1123120190207170314773456903 8 9038170000062.xml

Parameter	Description	Value as per assumption
KEY	It shall contain the unique key being used for identifying transaction request message.	TXN_REQUESTs
DATE TIME	It shall contain transaction request message generation date and time	191225142906
TOKEN (MD5[DATA] + 32 0s)	It shall contain 32 bytes hash of DATA field followed by 32 bytes containing 0s since it is request message.	446f4ce6347e6347b49cfecf897e7bae0000000000000000000000000000000000

Procedure:

- Terminal generates the transit file as and when transaction happens, construct transit file request message using CMO (refer figure below) and transmits the message to AFC System.

CMO packet construction (refer Table 51):

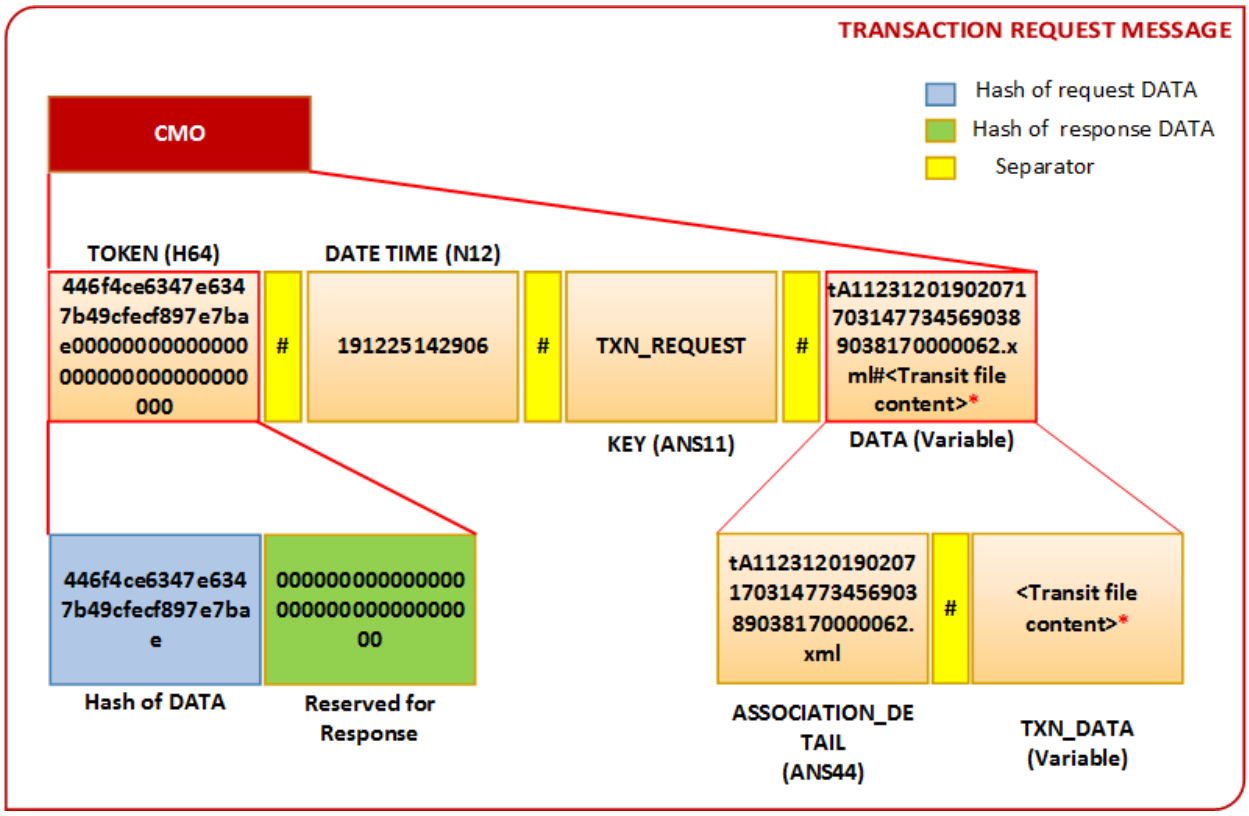


Figure 91: Transaction Request CMO

Example 8: Transaction response message using CMO

After receiving the transaction, AFC System generates the Transaction response/Acknowledgment and sends it back to client.

Assumption:

- Response for single transit file is sent.
- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN – MD5
- Transit file response message generation time at backend - December 25th' 2019 at 14:31:20
- Key used to identify transaction response message - TXN_RESPONSE
- Association details -tA11231201902071703147734569038 9038170000062.xml (Name of transit file sent by terminal in request message)

Table 52: Transaction Response Message Structure

Parameter	Description	Value as per assumption
DATA	It shall contain the association details i.e. the transit file name received in request message.	tA11231201902071703147734569038 9038170000062.xml

Parameter	Description	Value as per assumption
KEY	It shall contain the unique key being used for identifying transaction message response.	TXN_RESPONSE
DATE TIME	It shall contain transaction response message generation date and time	191225143120
TOKEN (Token in request message + MD5[DATA])	It shall contain 32 bytes token present in request message followed by 32 bytes hash of response DATA.	446f4ce6347e6347b49cfecf897e7baefb6a6c61857f89c1590259829fb9fd10

Procedure:

- Whenever AFC System receives transaction request message, it sends acknowledgement in transaction response message constructed using CMO back to terminal.

CMO packet construction (refer Table 52):

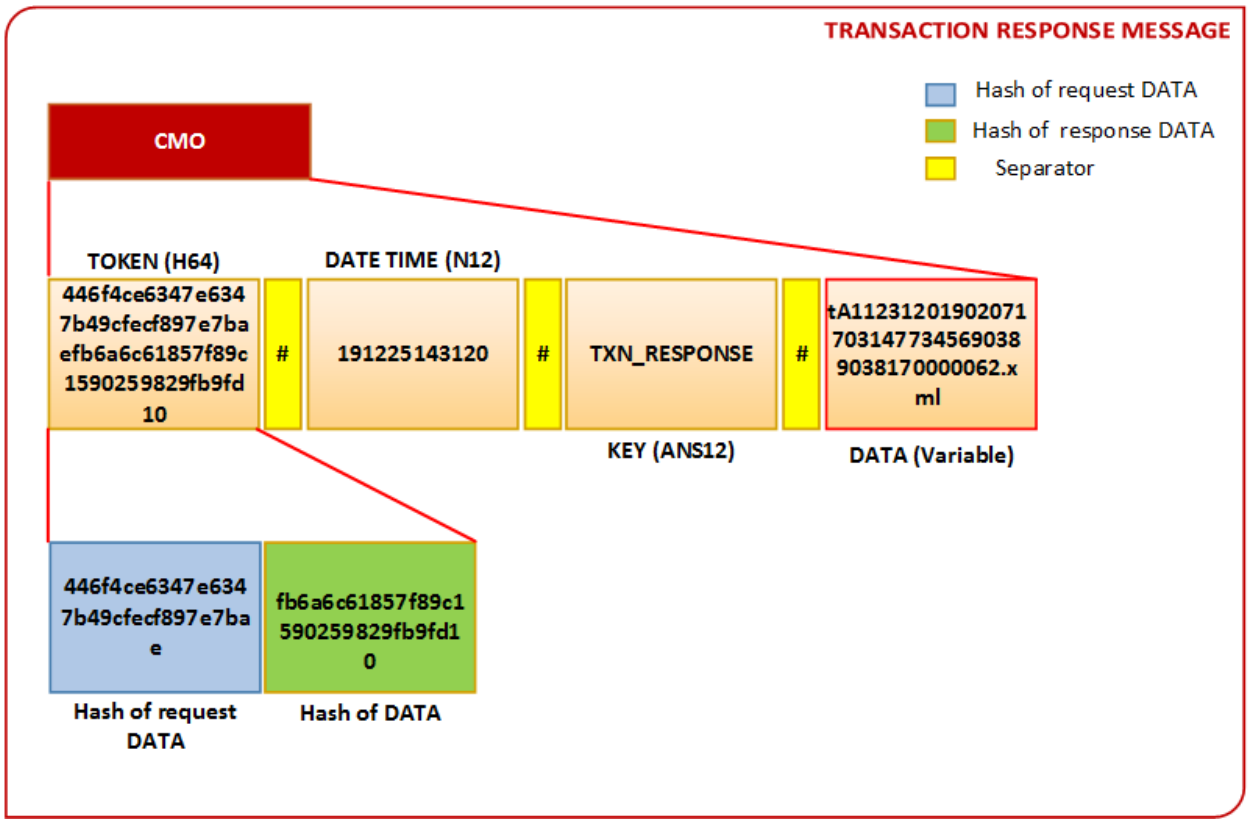


Figure 92: Transaction Response CMO

4.4.6. How Transaction is linked with RRN

The financial files shall be generated in two ways: -

a) File containing single transaction
b) File containing multiple transactions.

In any of the above mentioned ways, a financial file can be sent to the Acquirer. The examples of both the cases are mentioned in Part VI: AFC Ecosystem - Acquirer Interface. Based on the financial file sent to the Acquirer, AFC System shall receive single acknowledgement or batch acknowledgement. Both the Acknowledgement shall be in xml format. For the response details and format, kindly refer Part VI AFC – Acquirer Interface. Once the acknowledgement is received by the AFC System the AFC System saves the acknowledgement data. AFC System makes a Consolidated RRN file containing all the acknowledgement which it received in pre-configurable time and saves it in its repository. The file generated in pre-configurable time shall be in .xml format. The Structure of Consolidated RRN format is as per below format:-

Table 53: Consolidated RRN Message Structure

File format	File Structure	RRN File Structure	
XML	Header	Sequence no. for file identifier	Refer Table 47
		Version number of structure	
		Generation time	
	RRN Block	CRRN 1	ORD1*
			RRN1
			TC1
			Response Code1
		CRRN 2	ORD2
			RRN2
			TC2
			Response Code2
		.	
		.	
		CRRN n	ORDn
			RRNn
			TCn
			Response Code
	Trailer	No of Records	Refer Table 48

- **For detailed Structure of ORD, Kindly refer** Table 43**.**
- **For detailed Structure of RRN, TC and Response Code, refer Part VI: AFC Ecosystem to Acquirer Interface.**

The Consolidated RRN file is also transmitted to AFC Backend using the Common Message object format as shown in the below figure: -

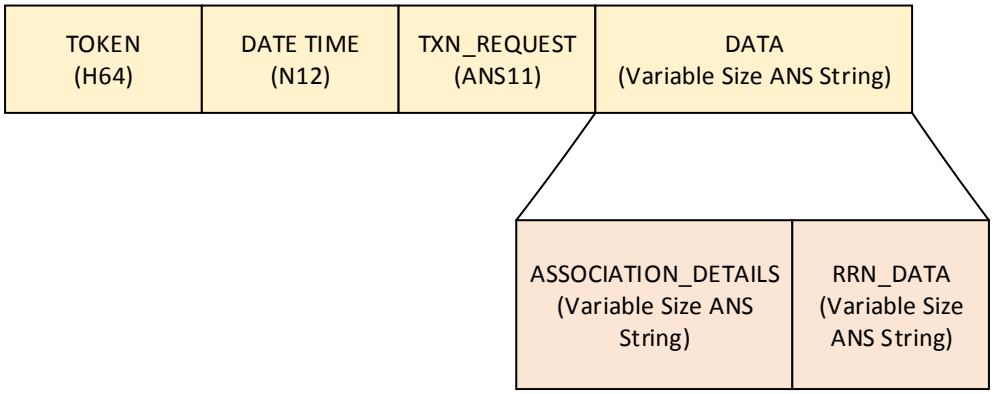


Figure 93: Consolidated RRN Request Structure

The message structure of RRN request is shown in the table below.

Table 54: Consolidated RRN Request Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	It indicates the association details of the Consolidated RRN file. Association details shall be the Consolidated RRN File Name.
2	RRN_DATA	Depending upon the data content(Variable Size)	ANS	It holds the Content of the Consolidated RRN File which shall be in xml format.

For every Consolidated RRN file transmitted to AFC Backend, AFC Backend sends back the Acknowledgement back to client in the CMO Data Format.

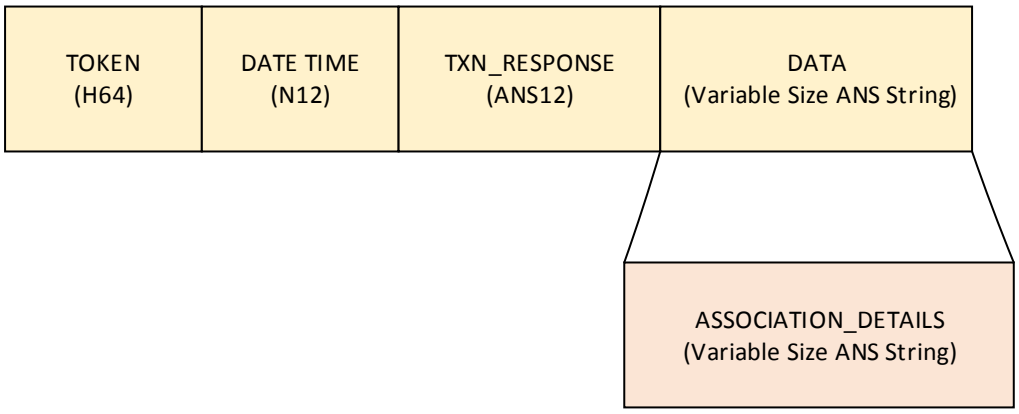


Figure 94: Consolidated RRN Response Structure

The message structure of RRN response is shown in the table below.

Table 55: Consolidated RRN Response Structure

S. No.	Configuration Parameter	Data Size	Data Format	Descriptions
1	ASSOCIATION_DETAILS	Variable	ANS	It indicates the association details of the Consolidated RRN file. Association details shall be the Consolidated RRN File Name.

The TXN_DATA of the transit file also contains the ORD and TC. Thus RRN is linked with every transaction based on ORD and TC.

Example 9: Consolidated RRN request message using CMO

Assumption:

- Transaction service is being used to send consolidated RRN file.
- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Consolidated RRN file request message generation time at terminal - December 25th' 2019 at 14:32:52
- Key used to identify transaction request message - TXN_REQUEST
- Association details -ConsRRN_000001_191225143252.xml (Name of Consolidated RRN file to be sent in request message)

Sample file

```
<?xml version="1.0" encoding="UTF-8"?>
<File>
<hdr>
<nFlSeqno>000001</nFlSeqno>
<nFlVERSION>10</nFlVERSION>
<nFlDtTmLcTxn>191225143252</nFlDtTmLcTxn>
</hdr>
<RRNblock>
<CRRN RecNum= "1">
<!--Subblock for RRN-->
<nORD>01010200108119359140019911 2</nORD>
<nRRN>012894365736</nRRN>
<n9F26></n9F26>
<nRC>00</nRC>
</CRRN>
</RRNblock >
<trl>
<nNoRecords>001</nNoRecords>
</trl>
</File>
```

Figure 95: Sample file Consolidated RRN Request

Table 56: Consolidated RRN Request Message Structure

Parameter	Description	Value as per assumption
DATA	It shall contain association detail followed by consolidated RRN file data	ConsRRN_000001_191225143252.xml #<Consolidated RRN file content>
KEY	It shall contain the unique key being used for identifying transaction request message.	TXN_REQUEST
DATE TIME	It shall contain transaction request message generation date and time	191225143252
TOKEN (MD5[DATA] + 32 0s)	It shall contain 32 bytes hash of DATA field followed by 32 bytes containing 0s since it is request message.	993defafb2b0e5f996751cf529dfb81400 00000000000000000000000000000000

Procedure:

- Terminal prepares a Consolidated RRN file containing all the acknowledgement which it received from acquirer in pre-configurable time.
- Terminal transmits the file to backend server constructed using CMO.

CMO packet construction (refer Table 56):

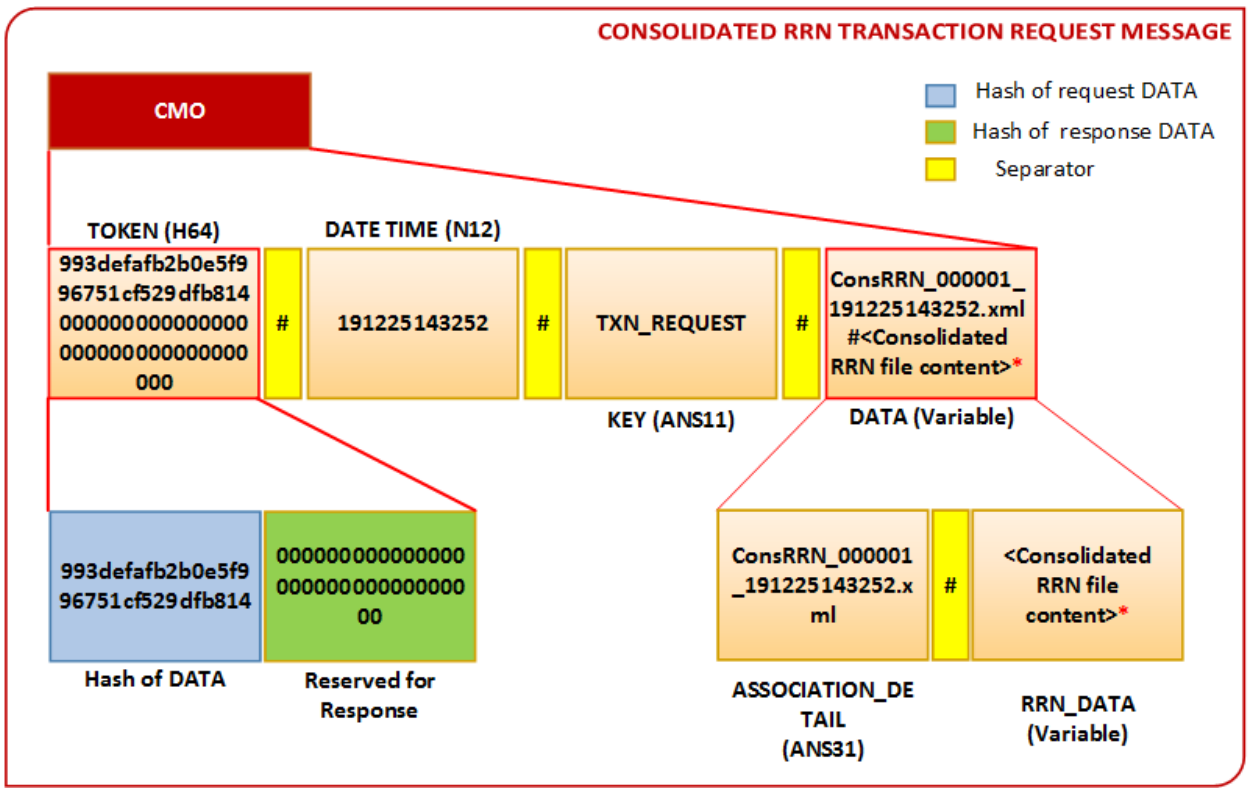


Figure 96: Consolidated RRN Request CMO

***Refer sample file in Figure 95**

Example 10: Consolidated RRN response message using CMO

Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN – MD5
- Consolidated RRN file response message generation time at backend - December 25th' 2019 at 14:32:54
- Key used to identify transaction response message - TXN_RESPONSE

- Association details -ConsRRN_000001_191225143252.xml (Name of Consolidated RRN file sent by terminal in request message of Example 9)

Table 57: Consolidated RRN Response Message Structure

Parameter	Description	Value as per assumption
DATA	It shall contain the file name of the file received from terminal in transaction request message	ConsRRN_000001_191225143252.xml
KEY	It shall contain the unique key being used for identifying transaction message response.	TXN_RESPONSE
DATE TIME	It shall contain transaction response message generation date and time	191225143254
TOKEN (Token in request message + MD5[DATA])	It shall contain 32 bytes token present in request message followed by 32 bytes hash of response DATA.	993defafb2b0e5f996751cf529dfb8140217e5ff38b84d39dec779727f7b84b5

Procedure:

- Whenever backend server receives transaction request message, it sends acknowledgement in transaction response message constructed using CMO back to terminal.

CMO packet construction (refer Table 57):

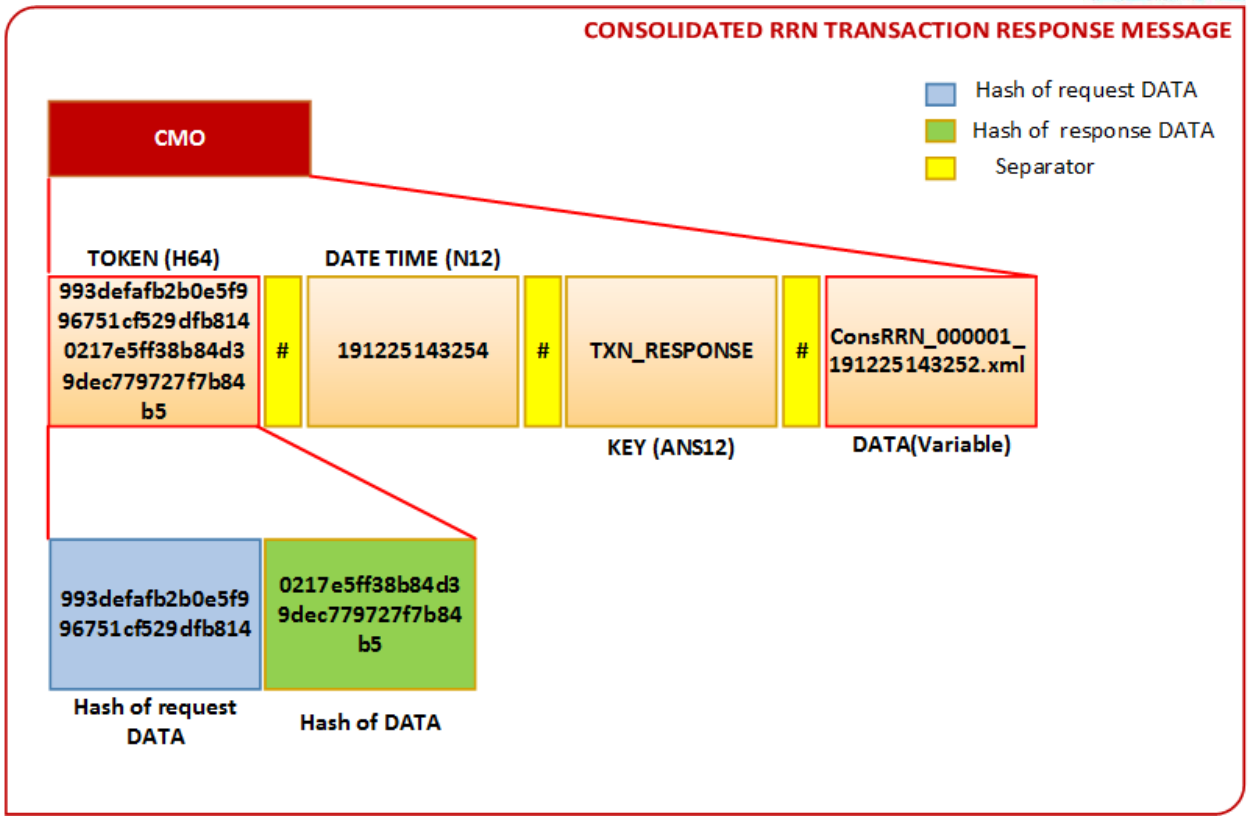


Figure 97: Consolidated RRN Response CMO

Annexure A: Transaction Data Elements

A1 Transaction Type:

Transaction Type data size is 1 byte. The structure of 1-byte transaction type is as follows:

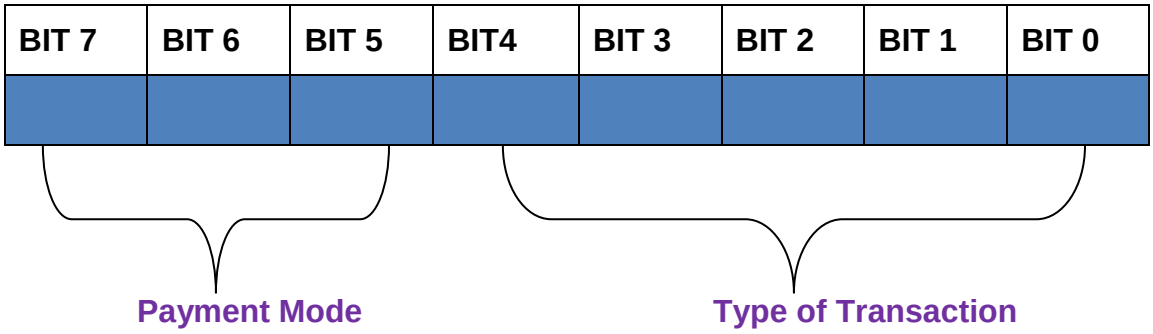


Figure 98: Transaction Type Bitwise Structure

The first three bits denotes the payment mode and the remaining 5 bits specify the transaction type.

Table 58: Bitwise mapping of payment modes

Payment Mode	Bit 7	Bit 6	Bit 5
NCMC	0	0	0
CASH	0	0	1
QR	0	1	0
TOKEN	0	1	1
RFU	1	0	X
USER DEFINED	1	1	X

Table 59: Bitwise mapping of transaction type for NCMC

Payment Mode	Type of Transaction	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NCMC	No Transaction	0	0	0	0	1
	Fare Debit Transaction	0	0	0	1	0
	Auto Topup	0	0	0	1	1
	Cash Topup	0	0	1	0	0
	Penalty Txn	0	0	1	0	1
	Torn Transaction	0	0	1	1	0
	Payment Reversal	0	0	1	1	1
	Online Recharge	0	1	0	0	0
	Pass Creation Transaction	0	1	0	0	1
	Pass Transit transaction	0	1	0	1	0
	RFU	0	1	0	1	1
	RFU	0	1	1	X	X
	User defined	1	X	X	X	X

Table 60: Bitwise mapping of transaction type for Cash based

Payment Mode	Type of Transaction	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CASH	Transaction	0	0	0	0	1
	RFU	0	X	X	1	X
	User defined	1	X	X	X	X

Table 61: Bitwise mapping of transaction type for QR

Payment Mode	Type of Transaction	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QR	RFU	X	X	X	X	X

Table 62: Bitwise mapping of transaction type for Token

Payment Mode	Type of Transaction	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TOKEN	RFU	X	X	X	X	X

A2 Transaction Place

Transaction Place data size is 1 byte. The structure of 1-byte transaction Place is as follows:

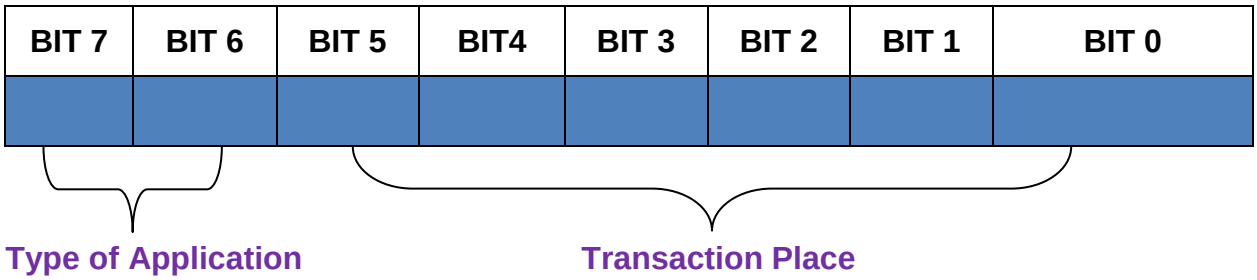


Figure 99: Transaction Place Bitwise Structure

The first two bits denote the type of application and remaining 6 bits specifies the transaction place.

Table 63: Bitwise mapping of Type of Application

Terminal Type	Bit 7	Bit 6
FIXED	0	0
MOBILE	0	1
OTHERS	1	X

Table 64: Bitwise mapping of transaction place for Fixed Terminal

TERMINAL TYPE	Type of Transaction place	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIXED	Entry	0	0	0	0	0	1
	Exit	0	0	0	0	1	0
	Customer Care	0	0	0	0	1	1
	Kiosk	0	0	0	1	0	0
	RFU	0	X	X	1	0	1
	RFU	0	X	X	1	1	X

	User defined	1	X	X	X	X	X

Table 65: Bitwise mapping of transaction place for Mobile Terminal

TERMINAL TYPE	Type of Transaction place	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MOBILE	Entry	0	0	0	0	0	1
	Exit	0	0	0	0	1	0
	Multi Role	0	0	0	0	1	1
	Fixed Fare	0	0	0	1	0	0
	ETIM	0	0	0	1	0	1
	Inspector	0	0	0	1	1	0
	RFU	0	X	X	1	1	1
	RFU	0	X	1	X	X	X
	User defined	1	X	X	X	X	X

Table 66: Bitwise mapping of transaction place for Third party application

Type of Application	Type of Transaction place	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OTHERS	Transaction	0	0	0	0	0	1
	RFU	0	X	X	X	1	X
	User defined	1	X	X	X	X	X

Annexure B Recommended data elements specific to Bus

Table 67: Bus related Transit Data Elements

S. No.	Field	Data Format	Size	Remarks
1	Bus No	AN	Upto 32 chars	Bus no in which the commuter is travelling
2	Route No	Numeric	4 chars	This is the code specific to route where the commuter is travelling
3	Conductor ID	Numeric	6 chars	Unique ID of the conductor who issues the ticket
4	Source Station	Numeric	5chars	Entry station configured for the bus
5	Destination Station	Numeric	5chars	Exit station configured for the bus
6	Concession Type	Numeric	2 chars	Concession Type in case of special category i.e. student etc.

Recommended Data elements in case of Pass usage

Table 68: Pass Related Transit Data Elements

S. No.	Field	Data Format	Size	Remarks
1	Pass Issuer ID	Numeric	4 Char	Unique ID of Operator.
2	Pass Limit	Hexadecimal String	12 Char	Trip count value
3	Pass Type	Hexadecimal String	2 char	It defines the type of the pass like staff pass, travel pass, tourist pass etc

S. No.	Field	Data Format	Size	Remarks
4	Entry Station	Alphanumeric	2 char	Entry station configured for the pass
5	Exit Station	Alphanumeric	2 char	Exit station configured for the pass
6	Start Date	Hexadecimal String	4 char	Start Date of the pass
7	End Date	Hexadecimal String	4 char	End date of the pass
8	Route	Hexadecimal String	2 char	Route code Ex-violet line.

**********End of Chapter 2 **********

Chapter 3

Interface Specification of NCMC Ecosystem

PART VI:
AFC Ecosystem - Acquirer Interface

Centre for Development of Advanced Computing (CDAC), Noida
Ministry of Electronics & Information Technology (MeitY)
Government of India

Contents

List of Tables

List of Figures

Revision History

Date	Version	Author	Comments
9th May, 2018	V 1.0	CDAC Noida	First Release
14th November,2019	V 1.1	CDAC Noida	i. Table 4: Add tags <nICCData> & <nATD>, remove tags <nServCd>, <nTC> & <nPSN>. ii. Table 7: Add tags <nICCData> & <Fee> in Table -7. iii. Annexure A: New Transaction file. iv. Add section: 2.5 Reconciliation. v. Remove 2.4.
28th February,2020	V1.2	CDAC Noida	i) Update Sec 2.3 Replaced ARD with ORD. ii) Added Section 2.4,2.5. iii) Updated Section 2.6. iv) Updated Section 2.7 (Elaborated it in detail), ARD is replaced by ORD globally. v) Updated all examples for request, response and reconciliation vi) Updated diagrams

References

Table 69: References

Reference	Title
RuPay Interface	RuPay Global Clearing and Settlement Technical Message Specification Version 1.9
	RuPay - Online Switching Interface Specification Version 1.8
ISO	ISO_8583-1_2003: Financial transaction card originated messages — Interchange message specifications — Part 1: Messages, data elements and code values.
NCMC	Part V:Terminal - AFC Backend Communication Interface
W3C	W3C XML Version 1.0

1. Introduction

This document describes the interaction between the Automated Fare Collection (AFC) Ecosystem and the Acquirer. The specification described in this document needs to be followed by both the operator and the acquirer. The documents mentioned in References must be referred for detailed technical information of AFC Ecosystem to Acquirer Interface and Communication interface.

1.1. Abbreviations

This section briefly describes various terms and abbreviations used in the document.

Table 70: Abbreviations

Abbreviation	
AFC	Automated Fare Collection
AN	Alpha-Numeric
ANS	Alpha-Numeric-Symbol
b	Binary
C	Conditional
CMO	Common Message Object
EOD	End of Day
H	Hexadecimal String
LSB	Least Significant Byte
M	Mandatory
MTI	Message Type Identifier
N	Numeric
O	Optional
ORD	Operator Reference Data
PAN	Primary Account Number
PCI/DSS	Payment Card Industry Data Security Standard
POS	Point Of Sale
PSN	PAN Sequence Number
RRN	Retrieval Reference Number
T Day	Transaction Business Day
TC	Transaction Cryptogram
XML	Extensible Markup Language

1.2. Terms and Definitions

Table 71: Definitions

Term	Description
Direct Transmission	The transmission mode when Terminal sends Financial data directly to the Acquirer without the intervention of the AFC System. For details, refer to Part V: Terminal - AFC Backend Communication Interface specification.
Indirect Transmission	The transmission mode when Terminal sends Financial data to Acquirer through AFC Backend. In this case both Financial and Transit data propagates together. Ref: Part V: Terminal - AFC Backend Communication Interface for detailed description.
Offline Presentment	The offline presentment message transmitted by the acquirer for transactions completed offline, i.e. the transactions which were not authorized by the issuer via the NPCI network.
Frontend Server	The server system which directly interacts with the Validation Terminal e.g. Station Server System.
Backend Server	The server system which shall directly interacts with the validation terminal (based on operator) and Acquirer/Issuer e.g. Central Server System.
Intermediate Server	All server systems in the AFC System other than Frontend and Backend Server.
AFC System	It consists of Frontend/Backend server facing the terminal and the acquirer. AFC System does not consist of the Terminal and Gate.
AFC Ecosystem	The AFC Ecosystem consists of Terminal, Gate and AFC System (Consisting of Frontend server, other intermediate servers (if they exist) and Backend server).
Normal Data Flow	This mode applies when all mandatory and optional

	financial data flows from the AFC Ecosystem to acquirer or vice versa. Normal Data Flow can take place in both Direct and Indirect Transmission modes. The details of mandatory and optional data elements are defined in the respective sections.
Efficient Data Flow	This mode applies when only mandatory financial data flows from the terminal to Acquirer and optional financial data is added by the acquirer. Efficient Data Flow can take place in both Direct and Indirect Transmission modes. The details of mandatory and optional data elements are defined in the respective sections.
Mandatory data elements	These data elements of offline presentment are mandatory requirement and produced by the terminal in both efficient and normal mode of communication.
Optional Data Elements	These data elements of offline presentment are optional requirement and produced by the terminal in normal mode and added by the Acquirer in efficient mode. Most of the optional data elements are static in nature. In case, any optional data element is dynamic, it must be generated at terminal and shall be sent to acquirer either directly or through AFC system.
Static Elements	Most of the Optional data elements are static in nature. These elements values will not change with every transaction.
Dynamic Elements	These elements values will change with every transaction.
Standard Communication Process	As per the standard communication process, the LSB shall be transferred first in the network. The rightmost byte in the data structure is the LSB.
Operator	“Operator” means any commercial establishment that accepts monetary transactions made through NCMC or other media type.
Tokenized PAN	It means Primary Account Number (PAN), the primary piece

	of cardholder data is replaced with a surrogate value, called tokens.
Service ID	Service area in cards like NCMC card is referred by a unique “Service ID” which is managed and provided by NPCI to any operator. For details kindly refer to NPCI RuPay EMV Dual Interface Card Application Specification V 2.0.

1.3. Scope

The document specifies the communication interface between the AFC Ecosystem and Acquirer. It covers the communication interface between (a) Terminal and Acquirer and (b) AFC System and Acquirer.

The document also describes the data structures of all the data elements that comprise the messages between the AFC Ecosystem and the acquirer. Description of any other kind of data such as protocol-specific data or other operator-specific transactional data is not within the scope of this document.

The specifications pertaining to the underlying physical and logical layers of the communication channels of participating entities are also not in scope of this document. For details about those specifications refer to Part V: Terminal - AFC Backend Communication Interface**.**

It is possible that the acquirer may send financial transaction information to the issuing bank for clearance or settlement. Specifications and details of such communication is not within the scope of this document. Dispute specific details are also not within the scope of this document. The current document does not specify any dispute-specific details like data format or interface.

2. AFC Ecosystem - Acquirer Interface

During any general transit operation (Entry/Exit), financial transaction data generated by the AFC Ecosystem is sent to the acquirer for settlement and reconciliation of funds with the operator. Because of the significant time complexity involved in processing card data, all such transactional data cannot be handled through online authorization and settlement. Hence, the NCMC mandates the use of two modes of transfer –Direct and Indirect Transmission modes. This section defines the requirements, standards and data structures of message elements of all such exchanges.

2.1. Requirements and standards

Messages exchanged between the AFC Ecosystem and Acquirer must only be sent through secure communication channels over and above meeting the necessary compliance with the security standards prescribed in PCI-DSS Standards and requirements of this interface (physical or logical) are described in the sections below.

2.1.1. Physical Interface Requirement

The standards and requirements for the physical communication interface between the AFC Ecosystem and Acquirer is similar as defined in Part V: Terminal - AFC Backend Communication Interface Specification.

2.1.2. Security Requirement

Systems exchanging financial information are mandated to follow the information security standard of PCI-DSS to protect sensitive data of customers. In the AFC Ecosystem, participating entities must also thereby first comply with the latest stable version of PCI-DSS security standard before information exchange takes place. The table below describes the necessity of security requirement on the AFC Ecosystem w.r.t compliance with PCI-DSS.

Table 72: PCI-DSS Security Requirement of AFC Systems

AFC System / Modes	AFC Backend	Terminal
Direct Transmission	✗	✓
Indirect Transmission	✓	✓

2.1.3. Communication Standard

Communication standard for message exchanges between AFC Frontend/Backend Server &Acquirer or Terminal & Acquirer for clearance and settlement must be implemented by following either the Efficient Data Flow or the Normal Data Flow.

During transit using NCMC, the fare for the journey gets deducted from the Global wallet of the customer. In such cases, the financial data of the transaction must be sent to the acquirer and subsequently to issuer. These transactions – direct or indirect – are sent in a predefined manner either sequentially or in buffered batches at specified time intervals. The time period or interval is configurable and must be pre-agreed upon through mutual consent by all participating entities viz. the AFC Eco-System, acquirer, switch and issuer so as to optimize data transfer – taking into consideration network performance aspects like busy hours, system load and network traffic. Provisions must however be made to cater to ad-hoc or emergency transfers when the need arises.

2.2. Information Exchange Structure

This section describes the structure and format of the financial transaction data. The format of the financial file shall be XML. File transfers shall take place using secure application protocols like SFTP, HTTPS or TCP/IP with SSL.

The data structure of all financial transactions are as per the offline presentment message structure. However, the content of the actual transaction varies as per the type of data flow as described below.

a) **Efficient Data Flow: -** In this mode, only the mandatory fields of the offline presentment structure will flow from the terminal to the acquirer. All the optional data elements are added by the Acquirer itself.

b) Normal Data Flow: - In this mode, all the elements of the offline presentment structure – mandatory and optional – are sent to the acquirer.

Details of transaction records are described in Table 73.

2.3. Offline Presentment Data Elements

This section defines elements of the "Offline Presentment" message structure. Each element has a specific meaning and format as described in "RuPay Global Clearing and Settlement Technical Message Specification V 1.9". Each message is wrapped inside a header and a trailer record. The data structure of messages remains the same irrespective of the mode of transmission – Direct or Indirect.

Figure 100 shows the message format of the Offline Presentment data structure in the form of XML data. The elements of transaction record are described in the table below.

Table 73: Data Elements for Transaction Record

S. No.	Field Name	XML Tag	To Acquirer	Length
1.	MTI	<nMTI>	O	N4
2.	Function Code	<nFunCd>	O	N3
3.	Record Number	<nRecNum>	M	N8
4.	Date and Time, Local Transaction	<nDtTmLcTxn>	M	N12
5.	Primary Account Number #1 Tokenized PAN	<nPAN>	M	N12-19
6.	Operator Reference Data	<nORD>	M	AN32
7.	Acquirer Institution ID code	<nAcqInstCd>	O	N11
8.	Card Acceptor Terminal ID	<nCrdAcptTrmId>	M	ANS8

S. No.	Field Name	XML Tag	To Acquirer	Length
9.	Amount, Transaction	<nAmtTxn>	M	N12
10.	Currency Code, Transaction	<nCcyCdTxn>	O	N3
11.	Transaction Originator Institution ID code	<nTxnOrgInstCd>	O	AN11
12.	Processing Code – Txn type	<nProcCd>	O	N6
13.	POS Entry Mode	<nPosEntMode>	O	N3
14.	POS Condition Code	<nPosCondCd>	O	N2
15.	POS Data Code	<nPosDataCd>	O	ANS41
16.	Card Acceptor Business Code	<nCrdAcpBussCd>	O	N4
17.	Action Code	<nActnCd>	O	N2
18.	Card Acceptor ID Code	<nCrdAcpIDCd>	M	ANS15
19.	Card Acceptor Name	<nCrdAcpNm>	O	ANS23
20.	Card Acceptor Location/ address	<nCrdAcpLoc>	O	ANS20
21.	Card Acceptor City	<nCrdAcpCity>	O	ANS13
22.	Card Acceptor State Name	<nCrdAcpStNm>	O	A2
23.	Card Acceptor Country Code	<nCrdAcpCtryCd>	O	A2
24.	Additional Data#2	<nAddData>	O	ANS.256
25.	ICC system related data*	<nICCData>	M	b….255
26.	Additional Transaction Data	<nATD>	M	ANS16

27.	Service Code	<nServCd>	O	AN3

#1Tokenized PAN: Tokenized PAN shall be used when financial data sent from Terminal or AFC System to acquirer.

#2 Additional Data: This element carries the Additional Data.

Note: Service ID shall be transferred inside the Additional Data Parameter with tag ID "**ServId**".

* <nICCData> containing TC in the <n9F26> tag.

Header Record

Header Record of the final data structure shall be populated along with each sub-part of individual transaction following offline presentment structure. It contains pre-defined standard fields which are used to validate the offline presentment data elements. The acquirer may perform suitable validations on each field of the header before sending it further to switch or acquirer bank.

Table 74: Header Record for Offline Presentment

S. No.	Field Name	XML Tag	Length
1.	MTI	<nMTI>	N4
2.	Function Code	<nFunCd>	N3
3.	Record Number	<nRecNum>	N8
4.	Date and Time, file generated	<nDtTmFlGen>	N12 (YYMMDDhhmmss)
5.	Date, Settlement	<nDtSet>	N6 (YYMMDD)
6.	Member Institution ID Code	<nMemInstCd>	AN11
7.	Unique File Name	<nUnFlNm>	AN21
8.	Product Code	<nProdCd>	AN5 (A3+N2)

9.	Settlement BIN	<nSetBIN>	AN6
10.	File Category	<nFlCatg>	A1
11.	Version Number	<nVerNum>	ANS5(Version is 01.00)

Trailer Record

Trailer Record of the final data structure shall be populated along with each sub-part of individual transaction following offline presentment structure. It contains pre-defined standard fields which are used to validate the offline presentment data elements. The acquirer may perform suitable validations on each field of the header before sending it further to switch or acquirer bank.

Table 75: Trailer Record for Offline Presentment

S. No.	Trailer Message	XML Tags	Length
1.	MTI	<nMTI>	N4
2.	Function Code	<nFunCd>	N3
3.	Record Number	<nRecNum>	N8
4.	Unique File Name	<nUnFlNm>	AN21
5.	Transactions Count	<nTxnCnt>	N8
6.	Run Total Amount	<nRnTtlAmt>	N15

Ref: RuPay Global Clearing and Settlement Technical Message Specification_V1.9.

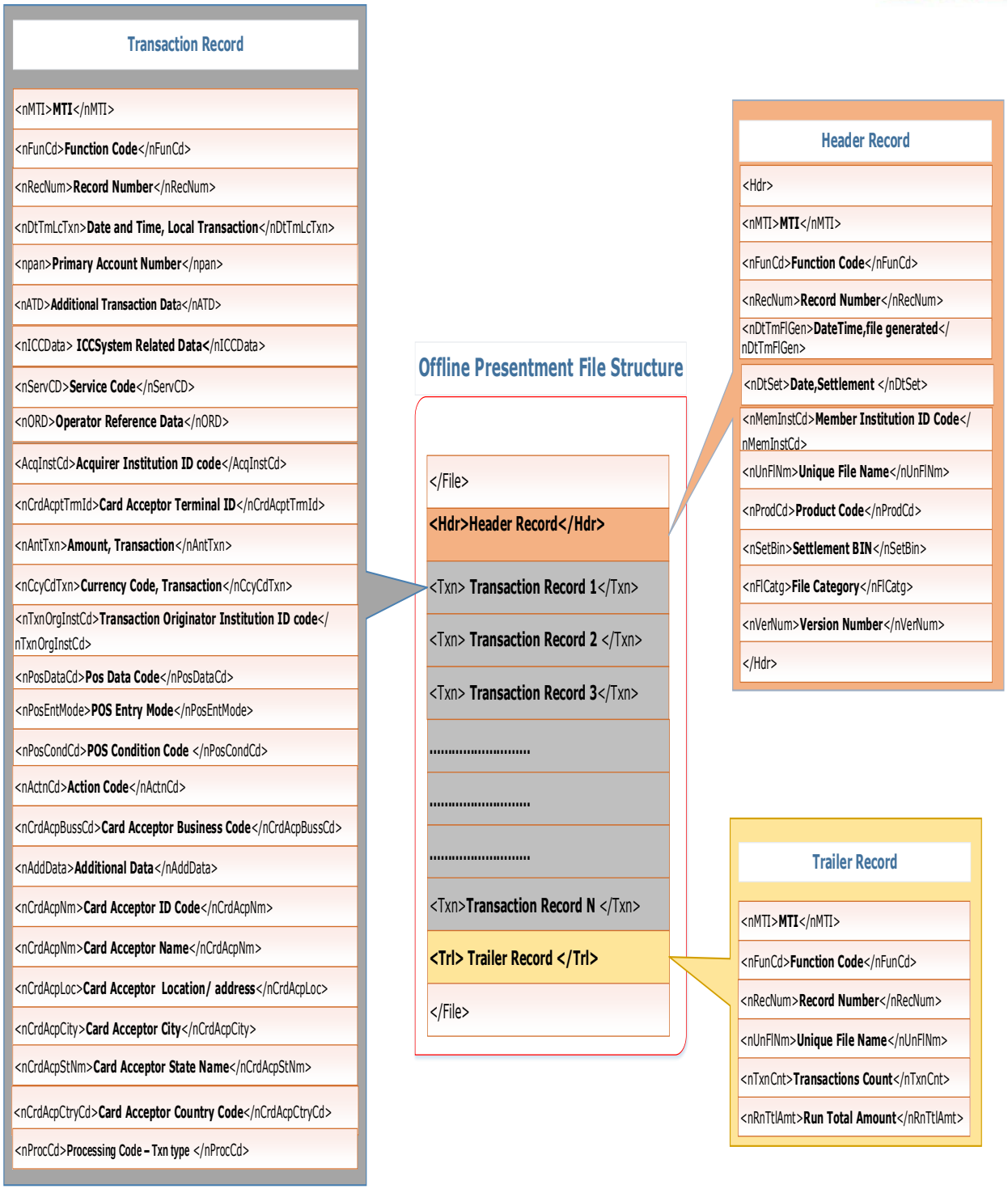


Figure 100: Data Structure of Offline Presentment

2.4. Data Transmission Modes

To transfer Financial files to Acquirer, AFC Ecosystem uses either of following two modes: -

a) Direct Transmission
b) Indirect Transmission

2.4.1. Direct Transmission Mode

In this mode, all financial transaction data will be transferred to acquirer directly from the Terminal for settlement. Anyone of the both data structures (Normal or Efficient) shall be followed for transmitting the data. In case of Normal transmission mechanism, all the data elements (Mandatory and Optional) are transmitted from the terminal while in case of Efficient transmission mechanism, only mandatory data elements shall be transmitted from the terminal while all the optional data elements shall be attached to the data by the Acquirer. The data format of the transaction is defined in Section2.5. The logical interface from Terminal to Acquirer shall be any secured channel i.e. SFTP, HTTPS, TCP/IP with SSL, other secure channels etc.

2.4.2. Indirect Transmission Mode

In Indirect mode, financial transaction data shall be transferred from the terminal via AFC Backend. In case of normal transmission mechanism, all mandatory and optional data elements shall be transmitted from the terminal to Acquirer via AFC Backend. In case of efficient transmission, only mandatory data shall be transmitted from the terminal, the optional data elements shall be attached with the data by Acquirer. The data format of transaction shall be same as in direct mode i.e. xml format. Thus terminal transmits financial file to AFC Backend in xml format and AFC Backend transmits the file to the Acquirer. The logical interface between Terminal to Server is as defined in Part V: Terminal - AFC Backend Communication Interface. The logical interface between AFC Backend and Acquirer shall be any secure communication protocol i.e. SFTP, HTTPS, TCP with SSL etc.

2.5. Financial Transaction Data

In both Direct and Indirect mode, the Terminal transmit all the financial files to Acquirer using the Common Message object. The format of the Common Message Structure is

defined in Section 4 of Part V: Terminal - AFC Backend Communication Interface. The financial file sent to Acquirer is shown below.

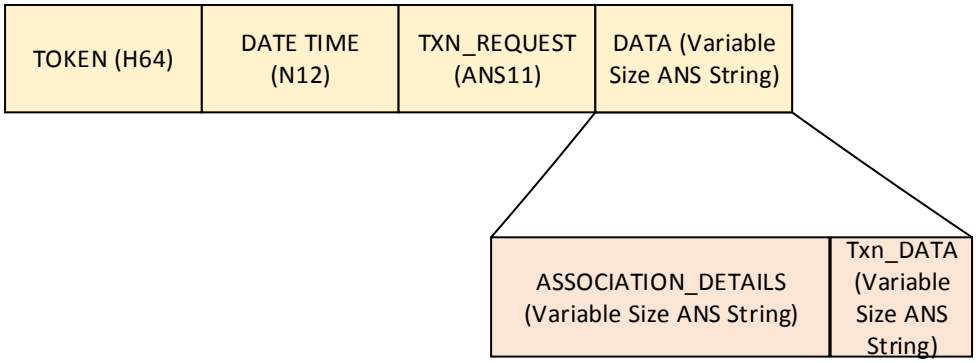


Figure 101: Financial Transaction Message Structure

Table 76: Financial Transaction Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description
1.	**ASSOCIATION_DETAILS**	Variable	ANS	It indicates the association details of the transaction file. Association details shall be the File Name. It shall be in xml format.
2.	**Txn_DATA**	Depending upon the data content(Variable Size)	ANS	It holds the content of the Transaction file which shall be in xml format.

Table 77:Txn_DATA Structure (Financial)

File Format	Data Structure	Data Elements	XML Tags

XML	**Header(<hdr>)**		Refer Table 74	Refer Table 74
	Transaction block(<txnblock>)	Txn 1 **<txnRecNum="1">**	Refer Table 73	Refer Table 73
		Txn 2 **<txnRecNum="2">**	Refer Table 73	Refer Table 73
		.		
		.		
		Txn n **<txnRecNum="n">**	Refer Table 73	Refer Table 73
	Trailer(<trl>)		Refer Table 75	Refer Table 75

Example 11: Financial transaction request message using CMO

Assumption:

- Financial data is transmitted using direct mode.
- Single financial file is sent each time.
- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN – MD5
- Financial file request message generation time at terminal - December 25th' 2019 at 14:30:50
- Association details -fA11231201902071703147734569038903817000062.xml (Name of financial file to be sent in request message)

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<File>
  <Hdr>
        <nMTI>1644</nMTI>
        <nFunCd>670</nFunCd>
        <nRecNum>00000001</nRecNum>
        <nMemInstCd>CDAC0010001</nMemInstCd>
        <nProdCd>POS01</nProdCd>
        <nSetBIN>CDAC01</nSetBIN>
        <nFlCatg>P</nFlCatg>
        <nVerNum>01.00</nVerNum>
        <nDtTmFlGen>191225143050</nDtTmFlGen>
  <nUnFlNm>fA1123120190207170314773456903890381700000062</nUnFlNm>
  </Hdr>
  <txnblock>
        <txn>
              <nORD>0101020010811935914001991l2</nORD>
              <nPAN>5499750b87c0233f</nPAN>
              <nAmtTxn>000000000100</nAmtTxn>
              <nDtTmLcTxn>191225143032</nDtTmLcTxn>
              <nCrdAcptTrmId> A1123110</nCrdAcptTrmId>
              <nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>
              <nMTI>0220</nMTI>
              <nFunCd>260</nFunCd>
```

```
                <nCcyCdTxn>356</nCcyCdTxn>

                <nTxnOrgInstCd>CDAC2013070</nTxnOrgInstCd>

                <nProcCd>00</nProcCd>

                <nPosEntMode>072</nPosEntMode>

                <nPosCondCd>02</nPosCondCd>


<nPosDataCd>71121241022000000000110053BARAKHAMBAMETROS</nPosDataCd
>

                <nCrdAcpBussCd>4111</nCrdAcpBussCd>

                <nActnCd>00</nActnCd>

                <nAcqInstCd>722203</nAcqInstCd>

            <nCrdAcpNm>CDACDMRCNCMCBARAKHAAMBA</nCrdAcpNm>

                <nCrdAcpLoc>0BARAKHAMBARDMETROSN</nCrdAcpLoc>

                <nCrdAcpCity>00000NEWDELHI</nCrdAcpCity>

                <nCrdAcpStNm>DL</nCrdAcpStNm>

                <nCrdAcpCtryCd>IN</nCrdAcpCtryCd>

                <nATD>9922200000</nATD>

                <nRecNum>00000002</nRecNum>

                <nICCData>

                        <n9F02>000000000500</n9F02>

                        <n9F26>9f12d456b23c43d4</n9F26>

                        <n82>1900</n82>

                        <n9F36>00e1</n9F36>

                        <n9F27>40</n9F27>


<n9F10>0fb501940000002014000000000047300000000047800000111195000000
```

```
00</n9F10>
                        <n9F33>000868</n9F33>
                        <n9F1A>0356</n9F1A>
                        <n95>0000000000</n95>
                        <n5F2A>0356</n5F2A>
                        <n9A>190207</n9A>
                        <n9C>00</n9C>
                        <n9F37>8aa90022</n9F37>
                    </nICCData>
            </txn>
    </txnblock>
    <Tlr>
            <nMTI>1644</nMTI>
            <nFunCd>671</nFunCd>
            <nRecNum>00000003</nRecNum>
            <nTxnCnt>00000001</nTxnCnt>
    <nUnFlNm>fA11231201902071703147734569038903817000062</nUnFlNm>
            <nRnTtlAmt>000000000100</nRnTtlAmt>
    </Tlr>
</File>
```

Figure 102: Sample XML (Financial Transaction Request)

Procedure:

- Terminal generates the financial file as and when transaction happens, construct financial file request message using CMO (refer figure below) and transmits the message to acquirer.

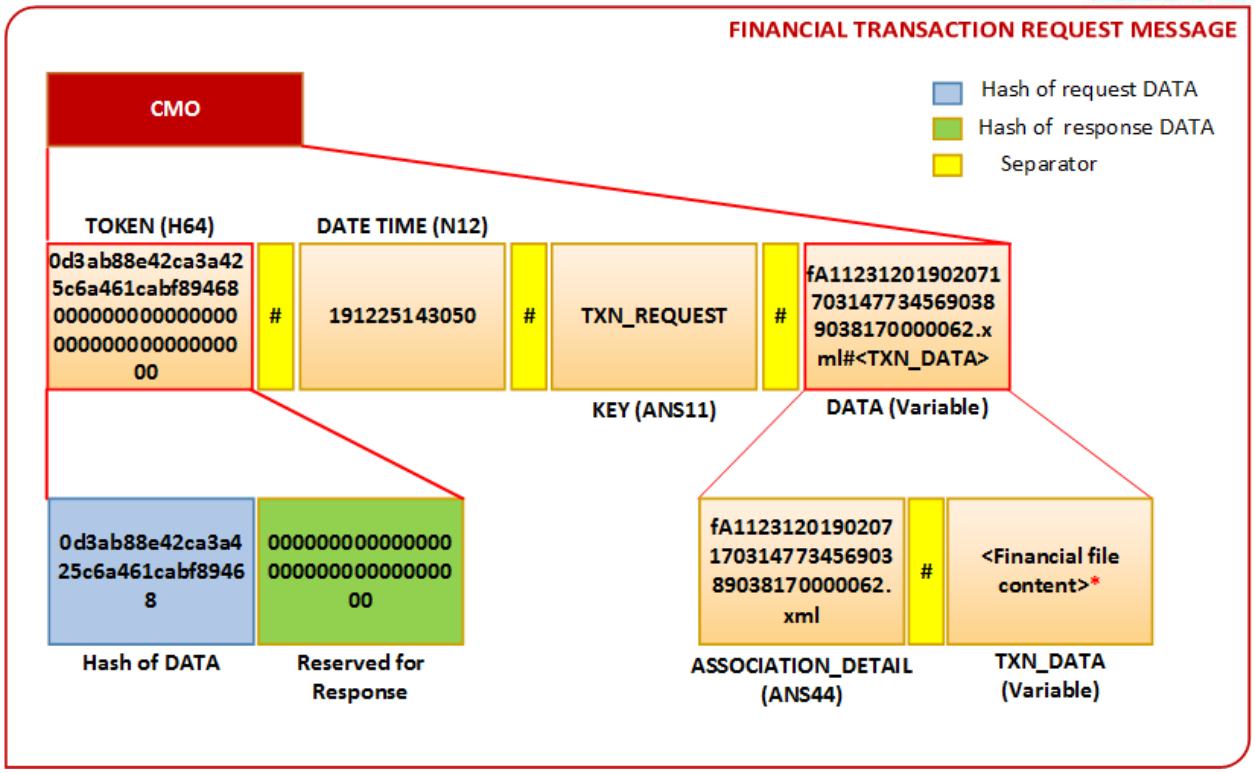


Figure 103: Financial Transaction Request CMO

*Refer to the sample file in

.

2.6. Acknowledgment from Acquirer

Acquirer shall generate an Acknowledgment in response of every financial transaction. The Acknowledgement shall be received in individual or batched mode which depends on how the terminal (in direct mode) or AFC system (in indirect mode) is transferring the transaction data to the acquirer. If the terminal or AFC system sends individual transaction to the acquirer, then individual acknowledgement shall be transmitted by the Acquirer and if the terminal or the AFC system sends the batched transactions then batched acknowledgement shall be transmitted by the Acquirer. Both the individual and batched acknowledgement transactions shall be transmitted in xml format.

The same Common Message Structure is used to transmit both the individual or batched Acknowledgement as described in Section 4 of Part V: Terminal - AFC Backend Communication Interface.

The data structure of the Acknowledgement is shown below: -

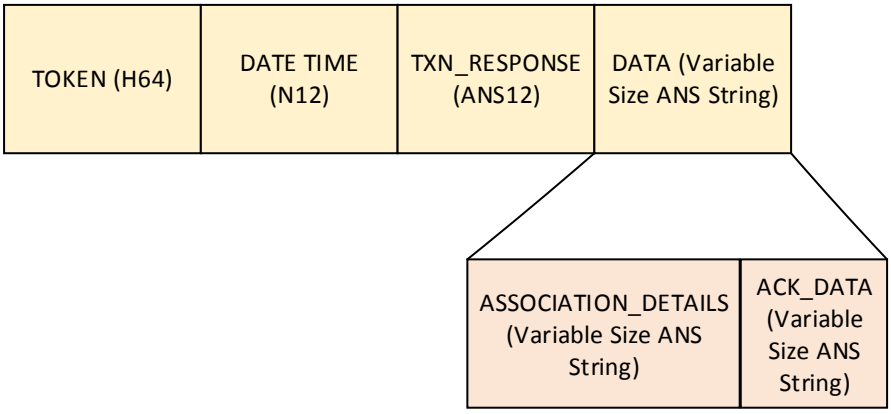


Figure 104: Financial Acknowledgement Message Structure

Table 78: Financial Acknowledgement Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description
1.	**ASSOCIATION_DETAILS**	Variable	ANS	It indicates the association details of the acknowledgement file. Association details shall be the Acknowledgement File Name. It shall be in xml format.
2.	**ACK_DATA**	Depending upon the data content(Variable Size)	ANS	It holds the content of the Acknowledgement file which shall be in xml format.

Table 79: Acknowledgement Data Structure

File Format	Data Structure(ACK_DATA)		
XML	**Header***	Version number of Structure	
		Generation time	
	Acknowledgement block	Record 1	ORD
			RRN
			TC
			Response code
		Record 2	ORD
			RRN
			TC
			Response code
		.	ORD
			RRN
			TC

			Response code
		.	ORD
			RRN
			TC
			Response code
		Record n	ORD
			RRN
			TC
			Response code
	Trailer*	No of Records	

*Detailed Header and Trailer structure are described below in Table 80 & Table 81.

Table 80: Header Structure of RRN file

S. No.	Field	Data Format	Size	XML Tags	Remarks
1.	Version no of the structure	Hexadecimal string	H2	<nFlVERSION>	It represents the version of the structure of the file. The version is 1.0 and represented as 10.
2.	Generation time	Numeric	N12	<nDtTmFlGen>	The date & time of the generation of the file in YYMMDDHHmmss format.

Table 81: Trailer Structure of RRN file

S. No.	Field	Data Format	Size	XML Tags	Remarks
1.	No of Records	Numeric	N3	<nNoRecords>	Identifies no of records in the file

The table below shows the structure of the Acknowledgement which the Acquirer sends in response to a successfully received financial transaction.

Table 82: Data Element of Acknowledgement Response

S. No.	Data Field	Data Type	Description
1.	ORD	AN32	Operator Reference Data. Unique transaction number which recognizes transaction in the operator system.
2.	RRN	AN12	Retrieval Reference Number unique for each transaction generated by Acquirer. Refer: "RuPay Global Clearing and Settlement Technical Message Specification_V1.9" and "RuPay - Online Switching Interface Specification Version 1.8" and above
3.	TC	B8	Cryptogram that is generated by signing data elements when the card approves the payment for clearing and settlement is known as the Transaction Certificate (TC).
4.	Response Code	N2	Response Code defines the response to a request /advice for a transaction. Refer:"RuPay - Online Switching Interface Specification Version 1.8" and above

Example 12: Financial transaction response message using CMO

Assumption:

- Financial data is transmitted using direct mode.
- Response for single financial file is sent.
- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN – MD5
- Financial file request message generation time at acquirer - December 25th' 2019 at 14:32:52
- Association details -fA11231101901311156087734569031903111000066 9.xml (Name of financial file sent by terminal in request message)

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>

<File>

  <Hdr>

        <nFlVERSION>10</nFlVERSION>>

        <nDtTmFlGen>191225143252</nDtTmFlGen>

  </Hdr>

  <Ack>

        <Rec RecNum="1">

              <nORD>01010200108119359140019911 2</nORD>

              <nRC>00</nRC>

                    <nRRN>012894365736</nRRN>

                    <n9F26>9f12d456b23c43d4</n9F26>

        </Rec>

  </Ack>

  <Tlr>

        <nNoRecords>001</nNoRecords>
```

```
  </Tlr>

</File>
```

Figure 105: Sample XML (Financial Transaction Response)

Procedure:

- Once acquirer receives the request message, it constructs the financial file response message using CMO and transmits back to terminal.

In case if the response is not received by the AFC System, then the AFC System will retransmit it to Acquirer again and again until the Acknowledgement is received. The time interval between retransmission and the number of retries is configurable.

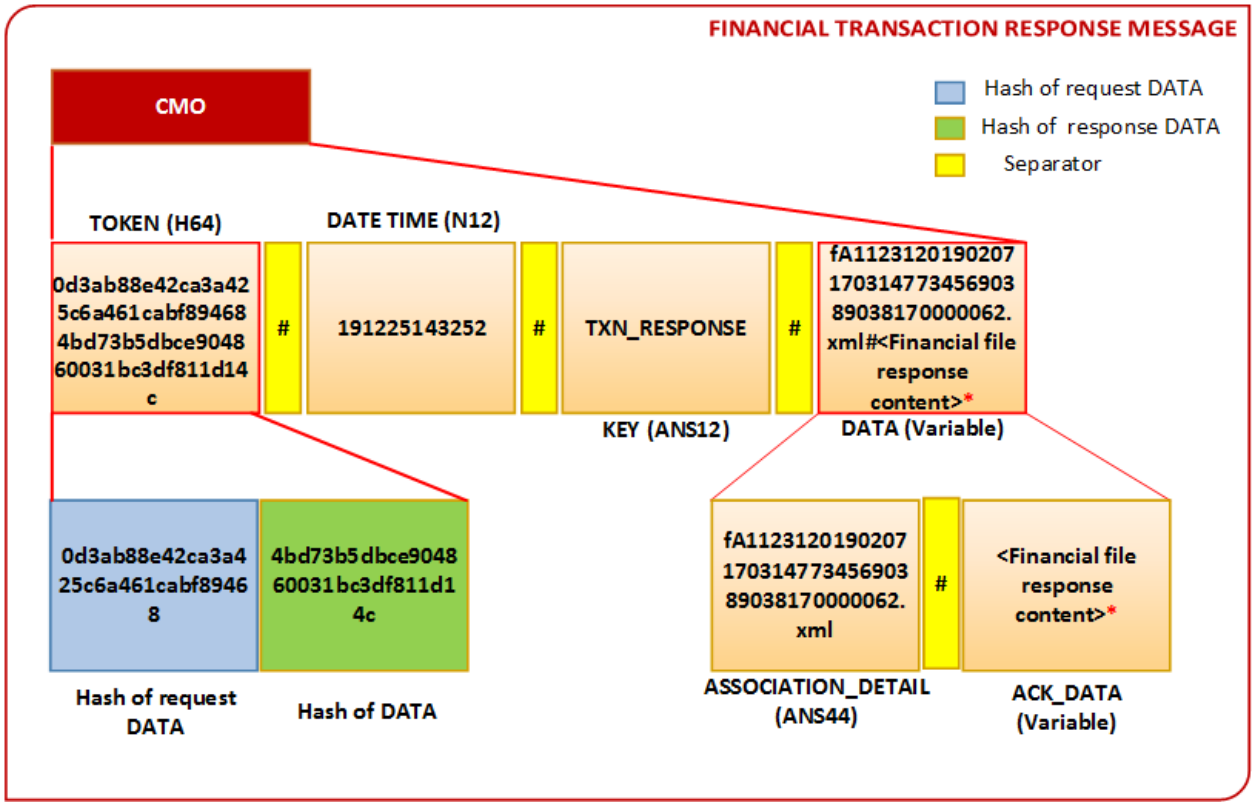


Figure 106: Financial Transaction Response CMO

*Refer to the sample file in Figure 105.

2.7. Reconciliation (Recon)

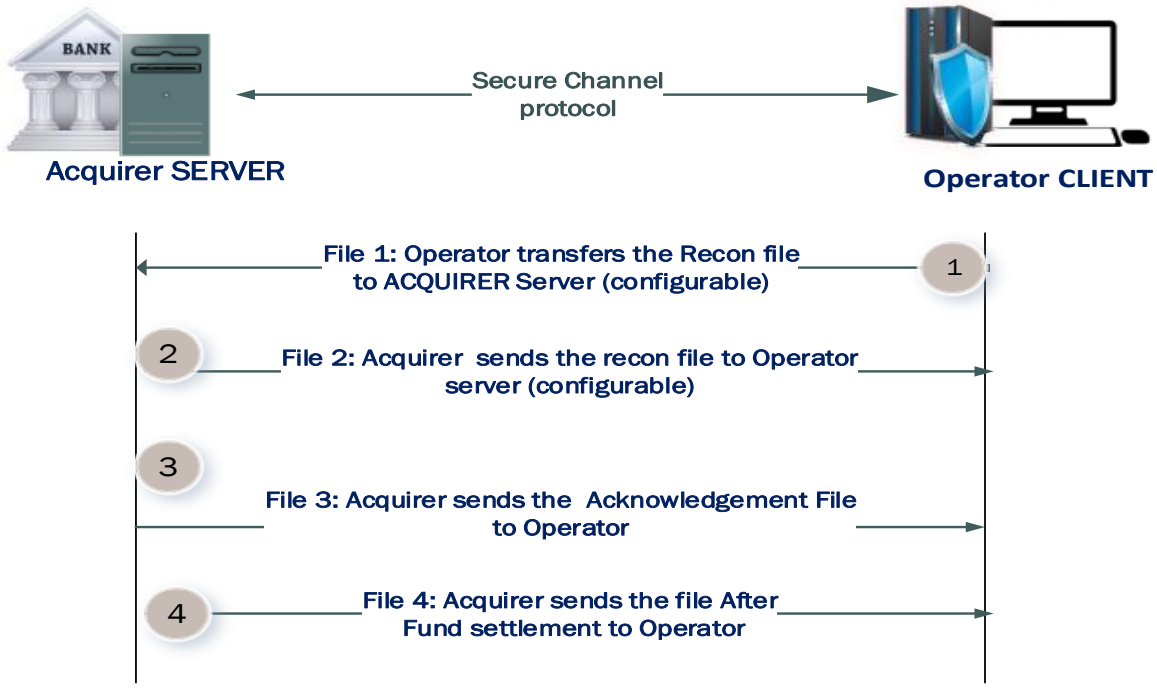
To settle the transaction file, operator shall generate a Reconciliation file. The interval of Transmission of reconciliation file shall be as per Operator-Acquirer Agreement. The transactions generated at the operator end by the terminals will be reconciled i.e. grouped together (batched) and sent to the operator's acquiring bank. Acquirer shall also share its reconciliation file with the operator at a predefined time interval based on the Operator-Acquirer Agreement after the files – File 3 and File 4 – are shared by Acquirer with the Operator. Against all the financial transactions, the Issuer bank will credit the amount as presented by the acquirer into the operator's account. That is the final step in settling the transaction.
All files shall be communicated or transferred through below communication channel with following mechanisms:

i. SFTP
ii. HTTPS
iii. TCP with SSL

As described below, four different types of files are shared between acquirer and operator based on business logic maintained by acquirer for the purpose reconciliation/settlement. All files shall be in XML format. As shown in the figure below,

- File 1 (reconciliation file of the operator) is shared with the Acquirer initially.
- File 2 (reconciliation file of the Acquirer) is shared by the Acquirer with the Operator.
- File 3 (acknowledgement for File 1)is shared with the Operator. File 3 contains those transactions which are successfully staged by the Acquirer to the clearing Switch.
- File 4 (fund settlement file)is shared by Acquirer with the Operator after the amount of the staged transactions are credited to the operator's account.

Figure 107: Reconciliation Process Flow



It should be noted that in all the file – File 1 to File 4, Mandatory/Optional data elements means those data elements which should be mandatorily or optionally shared with each other. On the operator's side, ORD will be the reference for the data index in all the files, meaning the operator will track the status of the transaction based on ORD. All the files are shared between operator and Acquirer in Common Message Object structure as per Section 4 of Part V: Terminal - AFC Backend Communication Interface. The next section describes the data structures of recon files shared between operator and Acquirer.

2.7.1. File 1 – Operator Reconciliation File

The table below describes the structure of File 1.

Table 83: Data Elements of File 1(Operator Reconciliation)

S.No.	Data Field	Data Type	M/O	XML Tags	Description
1.	DATE/TIME	N12	M	<nDtTmLcTxn>	Transaction generation date and time (YYMMDDHHMMSS format)
2.	Card Acceptor ID Code	ANS15	M	<nCrdAcpIDCd>	Unique ID assigned to operator by acquirer
3.	Card Acceptor Terminal ID	ANS8	M	<nCrdAcptTrmId>	This element depicts the unique code assigned to a terminal at the card acceptor location (TID) by the Acquirer
4.	PAN (masked)	UPTO N12-16	M	<nPAN>	Cardholder's PAN.
5.	RRN7	AN12	M	<nRRN>	It is used to identify and track all messages related to a given cardholder transaction. If RRN is not present in File 1, then in that case, RRN value in xml will be kept blank.
6.	AMOUNT	N12	M	<nAmtTxn>	Amount of transaction
7.	ORD7	AN32	M	<nORD>	Operator Reference Data.

					Unique transaction number which recognizes transaction in the operator system. For detailed structure of ORD kindly refer Part V: Terminal - AFC Backend Communication Interface.
8.	TC7	B8	M	<n9F26>	A cryptogram generated by the card in response to a GENERATE AC command.
9.	RESPONSE CODE	AN2	O	<nRC>	Response code received as a part of acknowledgement of financial file from acquirer

At the configured time interval, AFC system generates the Reconciliation file (refer Table 84) to be shared with Acquirer. All transit transactions received at AFC System (with or without the presence of RRN) will be part of File 1. The structure of the message used for sending File 1 to Acquirer is as written below: -

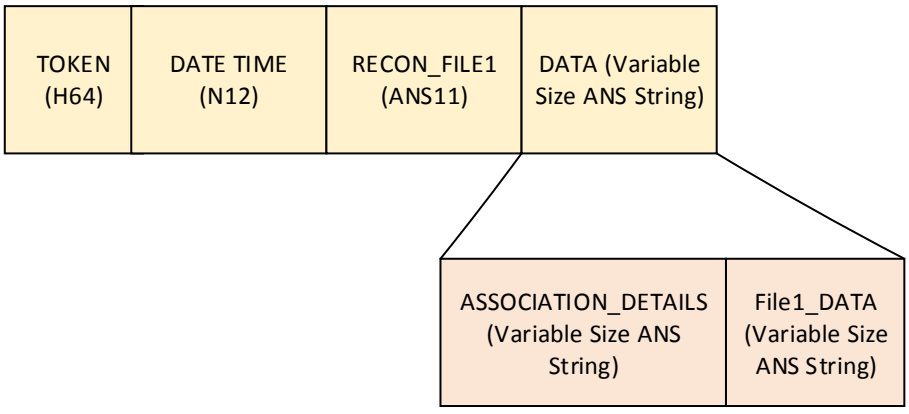


Figure 108: File 1 message structure

Table 84: File 1 Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description

1.	**ASSOCIATION_DETAILS**	Variable	ANS	It indicates the association details of the File 1 file. Association details shall be the File 1 Name. It shall be in xml format.
2.	**File1_DATA**	Depending upon the data content(Variable)	ANS	It holds the content of the File 1 file which shall be in xml format.

Table 85: File1 Data structure

File Format	Data Structure	File1_DATA	Data Elements	XML Tags
XML	**Header(<hdr>)**	Version number of Structure	Refer Table 86	<nFlVERSION>
		Generation time		<nDtTmFlGen>
	File1 block(<File1>)	Record 1 **<Rec RecNum="1" >**	Refer Table **83**	Refer Table **83**
		Record 2 **<Rec RecNum="2" >**	Refer Table **83**	Refer Table **83**
		.		

		.		
		Record n **<Rec RecNum=”n” >**	Refer Table **83**	Refer Table **83**
	Trailer(<trl>)	No of Records	Refer Table 87	<nNoRecords>
		Total Amount		<nTotalAmt>

Every message constructed using CMO consists of a header and trailer record. The header and trailer record structure of the reconciliation files (File 1 thru 4) are the same. They are described here below.

Table 86: Header Structure of File 1, 2, 3 &4

S. No.	Field	Data Format	Size	XML Tags	Remarks
1.	Version no of the structure	Hexadecimal string	H2	<nFlVERSION>	It represents the version of the structure of the file. Versions of structure in File 1, 2, 3 & 4 are 1.0, 1.0, 1.0 & 1.0 and represented as 10, 10, 10 & 10 respectively.
2.	Generation time	Numeric	N12	<nDtTmFlGen>	The date & time of the generation of the file in YYMMDDHHmmss

					format

Table 87: Trailer Structure of File 1, 2, 3 &4

S. No.	Field	Data Format	Size	XML Tags	Remarks
1.	No of Records	Numeric	N3	<nNoRecords>	Identifies no of records in the file
2.	Total Amount	Hexadecimal string	H12	<nTotalAmt>	It is the total amount claimed in the file.

Example 13: Transmission of File 1 - Operator Reconciliation File

Operator reconciliation file request message using CMO

Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN – MD5
- Operator Reconciliation File message generation time at Backend server - December 25th' 2019 at 20:29:06
- CMO Key used for uniquely identifying type of message - "RECON_REQUEST"
- Name of file - File1_ 01010200108119359140019911 2.xml (File1_ORD.xml)

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>

<File>

        <Hdr>

                <nFlVERSION>10</nFlVERSION>>

                <nDtTmFlGen>191225202906</nDtTmFlGen>

        </Hdr>

        <File1>
```

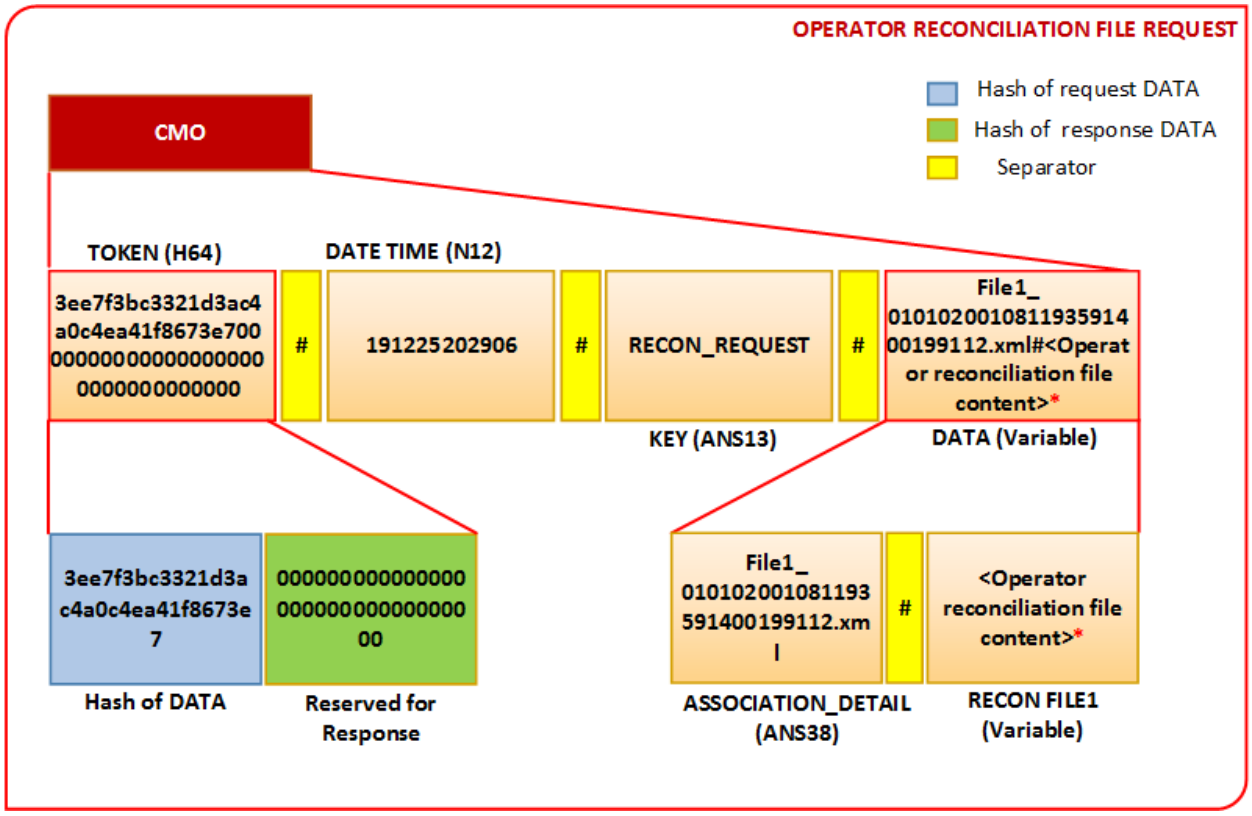
```
            <Rec RecNum="1">
                <nORD>0101020010811935914001991 12</nORD>
                <nPAN>608326xxxxxx0015</nPAN>
                <nAmtTxn>000000000100</nAmtTxn>
                <nDtTmLcTxn>191225143032</nDtTmLcTxn>
                <nCrdAcptTrmId> A1123110</nCrdAcptTrmId>
                <nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>
                    <nRC>00</nRC>
<nRRN>012894365736</nRRN>
                    <n9F26>9f12d456b23c43d4</n9F26>
            </Rec>
        </File1>
        <Tlr>
            <nNoRecords>001</nNoRecords>
            <nTotalAmt>000000000100</nTotalAmt>


        </Tlr>
</File>
```

Figure 109: Sample XML (File 1)

Procedure:

- Once all transit transactions are received at AFC System, the data for all files irrespective of whether RRN corresponding to each transit transaction is received or not, is consolidated into an XML file and shall be sent to acquirer in a CMO packet.

Figure 110: Operator Reconciliation CMO (File 1 Request)



*Refer sample file in Figure 109.

Operator reconciliation file response message using CMO

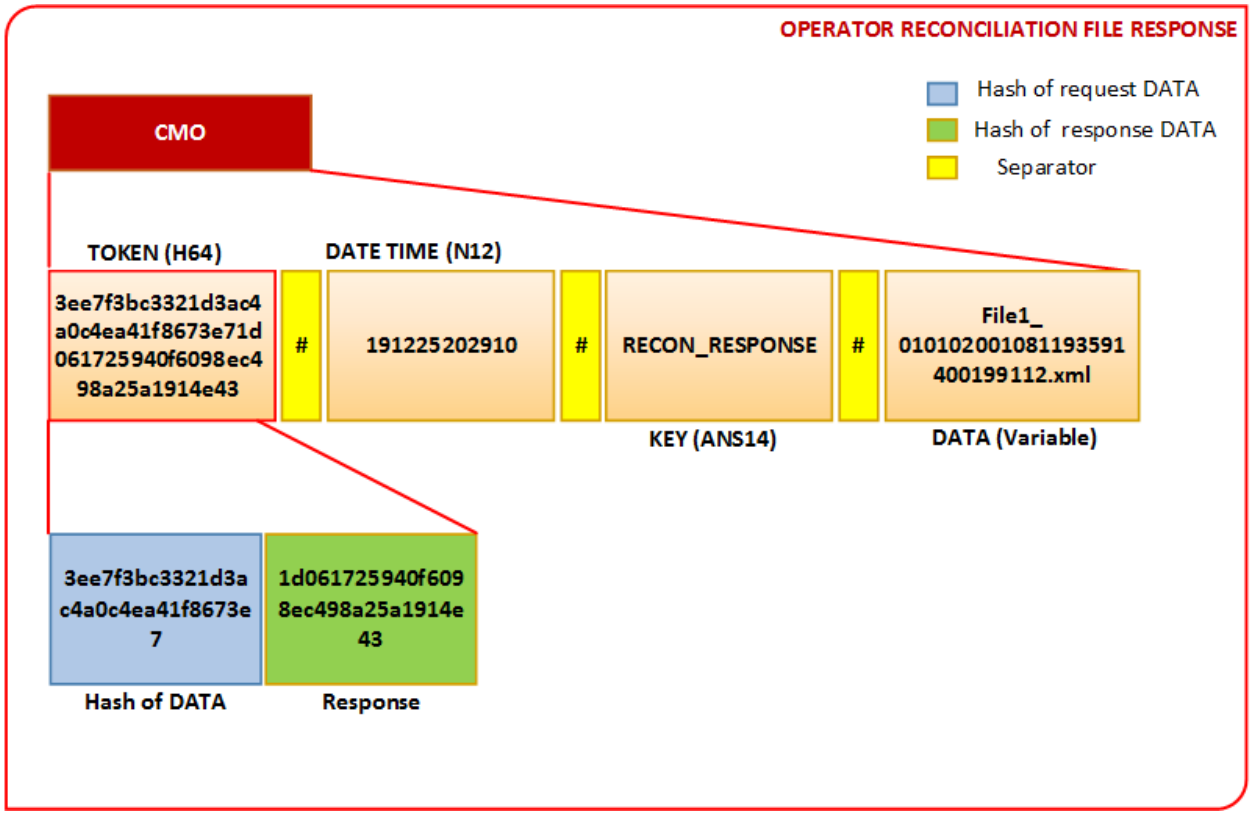
Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Operator Reconciliation File response message generation time - December 25th' 2019 at 20:29:10
- CMO Key used for uniquely identifying type of message - "RECON_RESPONSE"
- Association details -File1_ 01010200108119359140019911 2.xml (Name of Operator reconciliation file sent by AFC backend in request message)

Procedure:

- Once acquirer receives the request message, it constructs the Operator reconciliation file response message using CMO and transmits back to AFC backend.

Figure 111: Operator Reconciliation CMO (File 1 Response)



2.7.2. File 2 – Acquirer Reconciliation File

Acquirer will also prepare the list of all financial transactions received from Operator and share the reconciliation file with operator.
Please note that File 2 is not an acknowledgement of File 1. Files 1 & 2 are shared for reconciliation of the transactions at respective ends. The Acquirer always stage the financial transactions to the issuer on the basis of RRN. However, transactions in which RRN is absent, are staged on the basis of TC.

Table 88: Data Elements of File 2 (Acquirer Reconciliation)

S. No.	Data Field	Data Type	M / O	XML Tags	Description
1.	Card Acceptor ID Code	ANS15	M	<nCrdAcpIDCd>	Unique ID assigned to Operator by acquirer

S. No.	Data Field	Data Type	M / O	XML Tags	Description
2.	Card Acceptor Terminal ID	ANS8	M	<nCrdAcptTrmId>	This element depicts the unique code assigned to a terminal at the card acceptor location (TID) by the Acquirer
3.	PAN (masked)	UPTO N12-16	M	<nPAN>	Cardholder's PAN.
4.	Transaction type	N2	O	<nTxnTyp>	Refer Annexure A1 of PART V: Terminal - AFC System and Acquirer Communication Interface.
5.	Date/Time	N12	M	<nDtTmLcTxn>	Date and time when acquirer received the transaction in YYMMDDHHmmss format
6	AMOUNT	N12	M	<nAmtTxn>	Amount of transaction
7.	RRN[7]	AN12	M	<nRRN>	It is used to identify and track all messages related to a given cardholder transaction

[7]TC, ORD must be same in all the four files i.e. File 1, File 2, File 3, File 4. At the Operator end, ORD shall be the reference to track the status of the transaction. Acquirer shall use any of the ORD, TC or RRN to keep track of a transaction throughout its lifecycle.RRN must be same in the File 1(if present), File 2, File 4. RRN shall be extracted from ARD in File 3 that RRN must also be same as in File 1,2 and 4 for the said transaction.

S. No.	Data Field	Data Type	M / O	XML Tags	Description
8.	RESPONSE CODE	AN2	O	<nRC>	Response code received as a part of acknowledgement of financial file from acquirer
9.	ORD7	AN32	M	<nORD>	Operator Reference Data. Unique transaction number which recognizes transaction in the operator system. For detailed structure of ORD kindly refer Part V: Terminal - AFC Backend Communication Interface.
10.	TC7	B8	M	<n9F26>	A cryptogram generated by the card in response to a GENERATE AC command.

The structure of the message used for sending File 2 to Operator is shown in the figure below.

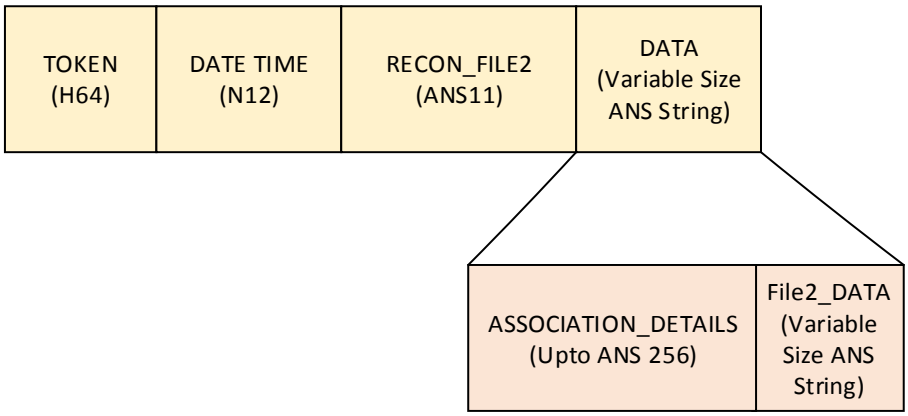


Figure 112: File 2 message structure

Table 89: File 2 Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description
1.	**ASSOCIATION_DETAILS**	Variable	ANS	It indicates the association details of the File 2 file. Association details shall be the File 2 Name. It shall be in xml format.
2.	**File2_DATA**	Depending upon the data content(Variable)	ANS	It holds the content of the File 2 file which shall be in xml format.

Table 90: File2Data structure

File Format	Data Structure	File2_DATA	Data Elements	XML Tags
	Header(<hdr>)	Version number of Structure	Refer Table 86	<nFlVERSION>
		Generation time		<nDtTmFlGen>

XML	**File2 block(<File2>)**	Record 1 **<Rec RecNum=”1”>**	Refer Table 88	Refer Table 88
		Record 2 **<Rec RecNum=”2”>**	Refer Table 88	Refer Table 88
		.		
		.		
		Record n **<Rec RecNum=”n”>**	Refer Table 88	Refer Table 88
	Trailer(<trl>)	No of Records	Refer Table 87	<nNoRecords>
		Total Amount		<nTotalAmt>

*After comparison of File 1 and File 2, these three cases may arise: -

- Case 1:**All the transactions in both the files are same:** In this case, Acquirer will stage all the transactions as listed in File 2 for staging to the issuer.
- Case 2:**If the number of transactions in File 1 is greater than the number of transactions in File 2,** then all the transactions as per File 1 shall be staged by the Acquirer to the Issuer.
- Case 3:**If the number of transactions in File 2is greater than the number of transactions in File 1**,then acquirer shall settle all transactions present in File 2. However, those transactions which are absent in File 1 shall be sent

by Operator to Acquirer within some configurable period of time as per the Operator-Acquirer agreement.

Example 14: Transmission of File 2–Acquirer Reconciliation File

Acquirer reconciliation file request message using CMO

- CMO Parameters separator - “#”
- Algorithm used for generating TOKEN – MD5
- Acquirer Reconciliation File message generation time at acquirer - December 25th’ 2019 at 21:29:06
- CMO Key used for uniquely identifying type of message - “RECON_REQUEST”
- Name of file – File2_ 01010200108119359140019911 2.xml (File2_ORD.xml)

Figure 113: Sample XML (File 2)

```
<? xml version="1.0" encoding=" -8UTF " standalone="no"?>

<File>

        <Hdr>

                <nFlVERSION>10</nFlVERSION>>

                <nDtTmFlGen>191225212906</nDtTmFlGen>

        </Hdr>

        <File2>

                <Rec RecNum="1">

                        <nORD>01010200108119359140019911 2</nORD>

                        <nPAN>608326xxxxxx0015</nPAN>

                        <nAmtTxn>000000000100</nAmtTxn>

                        <nDtTmLcTxn>191225143032</nDtTmLcTxn>

                        <nCrdAcptTrmId> A1123110</nCrdAcptTrmId>

                        <nTxnTyp>02</nTxnTyp>

                        <nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>
```

```
                    <nRRN>012894365736</nRRN>

                              <nRC>00</nRC>

                         <n9F26>9f12d456b23c43d4</n9F26>

            </Rec>

        </File2>

        <Tlr>

            <nNoRecords>001</nNoRecords>

            <nTotalAmt>000000000100</nTotalAmt>

        </Tlr>

</File>
```

Procedure:

Acquirer shall prepare the list of financial transactions as received from PTO and shall transfer the CMO packet containing the Acquirer reconciliation file with PTO.

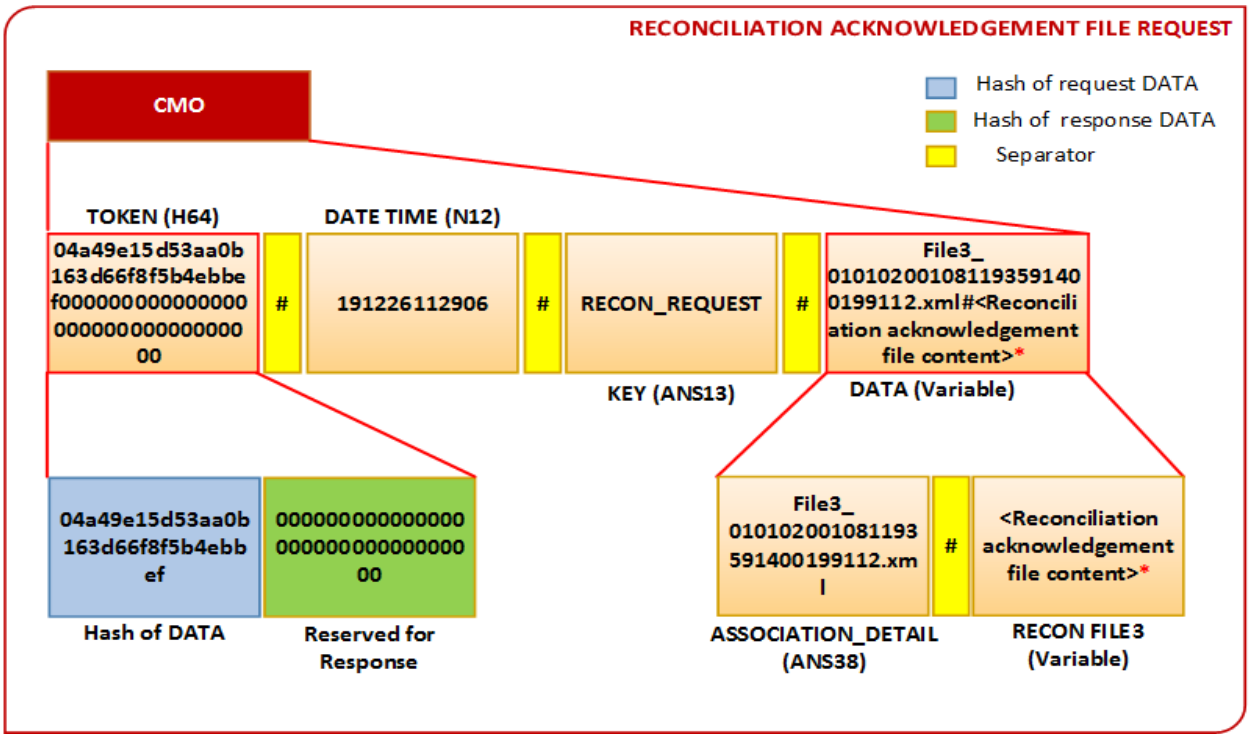


Figure 114: Acquirer Reconciliation CMO (File 2 Request)

*Refer sample file in Figure 113.

Acquirer Reconciliation File response message using CMO

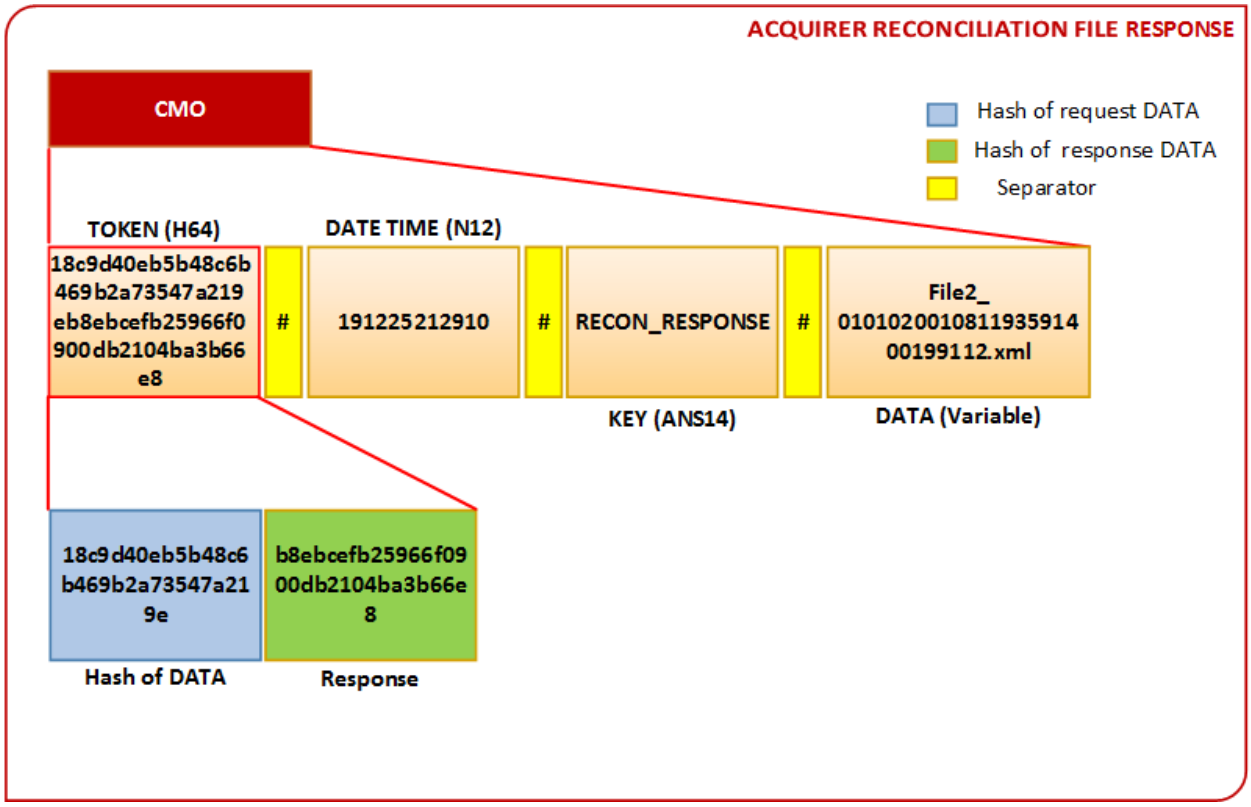
Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Acquirer Reconciliation File response message generation time - December 25th' 2019 at 21:29:10
- CMO Key used for uniquely identifying type of message - "RECON_RESPONSE"
- Association details –File2_ 01010200108119359140019911 2.xml (Name of Acquirer reconciliation file sent by acquirer in request message)

Procedure:

- Once AFC backend receives the request message from acquirer, it constructs the Acquirer reconciliation file response message using CMO and transmits back to Acquirer.

Figure 115: Acquirer Reconciliation CMO (File 2 Response)



2.7.3. File 3 – Reconciliation Acknowledgement File

Acquirer will share the acknowledgement file in xml format which is received from clearing switch to the Operator during file staging. For File format and File naming please refer to "RuPay Global Clearing and Settlement Technical Message Specification_V1.9".

Table 91: Data Elements of File 3 (Recon ACK)

S. No.	Field Name	XML Tag	From Acquirer	Length
1.	MTI	<nMTI>	O	N4
2.	Function Code	<nFunCd>	O	N3
3.	Record Number	<nRecNum>	O	N8
4.	Date and Time, Local Transaction	<nDtTmLcTxn>	M	N12
5.	Primary Account Number(masked)	<nPAN>	M	N12-19
6.	Operator Reference Data7	<nORD>	M	AN32
7.	Acquirer Reference Data*7	<nARD>	M	AN23
8.	Acquirer Institution ID code	<nAcqInstCd>	O	N11
9.	Card Acceptor Terminal ID	<nCrdAcptTrmId>	M	ANS8
10.	Amount, Transaction	<nAmtTxn>	M	N12
11.	Currency Code, Transaction	<nCcyCdTxn>	O	N3
12.	Transaction Originator Institution ID code	<nTxnOrgInstCd>	O	AN11
13.	Processing Code - Txn type	<nProcCd>	O	N6
14.	POS Entry Mode	<nPosEntMode>	O	N3
15.	POS Condition Code	<nPosCondCd>	O	N2
16.	POS Data Code	<nPosDataCd>	O	ANS41

17.	Card Acceptor Business Code	<nCrdAcpBussCd>	O	N4
18.	Action Code	<nActnCd>	O	N2
19.	Transaction Destination Institution ID code	<nTxnDesInstCd>	O	AN11
20.	Unique File Name	<nUnFlNm>	O	AN21
21.	Card Acceptor ID Code	<nCrdAcpIDCd>	M	ANS15
22.	Card Acceptor Name	<nCrdAcpNm>	O	ANS23
23.	Card Acceptor Location/ address	<nCrdAcpLoc>	O	ANS20
24.	Card Acceptor City	<nCrdAcpCity>	O	ANS13
25.	Card Acceptor State Name	<nCrdAcpStNm>	O	A2
26.	Card Acceptor Country Code	<nCrdAcpCtryCd>	O	A2
27.	Date Settlement	<nDtSet>	M	N6
28.	Settlement DR/CR Indicator	<nSetDCInd>	M	A1
29.	Amount, Settlement	<nAmtSet>	M	N12
30.	Currency Code, Settlement	<nCcyCdSet>	O	N3
31.	Conversion Rate, Settlement	<nConvRtSet>	O	N8
32.	Card Acceptor Zip Code	<nCrdAcpZipCd>	O	ANS9
33.	Merchant Telephone Number	<nMerTelNum>	O	AN11
34.	Service Code	<nServCd>	O	AN3
35.	Processing Status	<nProcSts>	M	A1
36.	ICC System Related Data7#	<nICCData>	M	b…255
37.	Fee Type Code 1	<nFeeTpCd>	O	N4

38.	Interchange Category 1	<nIntrchngCtg>	O	N4
39.	Fee DR/CR Indicator 1	<nFeeDCInd>	O	A1
40.	Fee amount 1	<nFeeAmt>	O	N10

* RRN shall be extracted from Acquirer Reference Data.
# Containing TC in the 9F26 tag in ICC System Related Data.

The structure of the message used for sending File 3 to Operator is as follows: -

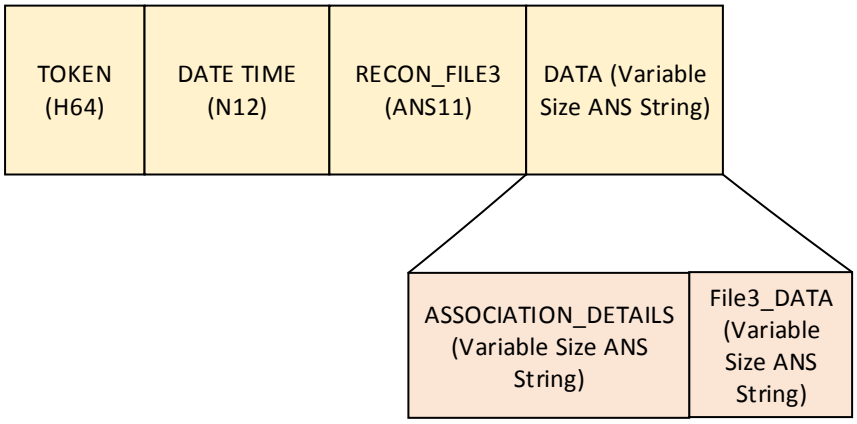


Figure 116: File 3 message structure

Table 92: File 3 Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description
1.	**ASSOCIATION_DETAILS**	Variable	ANS	It indicates the association details of the File 3 file. Association details shall be the File 3 Name. It shall be in xml format.
2.	**File3_DATA**	Depending upon the data content(Variable)	ANS	It holds the content of the File 3 file which shall be in xml format.

Table 93: File3 Data structure

File Format	Data Structure	File3_DATA	Data Elements	XML Tags
XML	Header(<hdr>)	Version number of Structure	Refer Table 86	<nFlVERSION>
		Generation time		<nDtTmFlGen>
	File3 block(<File3>)	Record 1 **<Rec RecNum="1">**	Refer Table 91	Refer Table 91
		Record 2 **<Rec RecNum="2">**	Refer Table 91	Refer Table 91
		.		
		.		
		Record n **<Rec Rec Num="n">**	Refer Table 91	Refer Table 91
	Trailer(<trl>)	No of Records	Refer Table 87	<nNoRecords>
		Total Amount		<nTotalAmt>

Example 15: Transmission of Reconciliation acknowledgement file

Reconciliation acknowledgement file request message using CMO

Assumption:

- CMO Parameters separator - "**#**"
- Algorithm used for generating TOKEN – MD5
- Reconciliation acknowledgement file message generation time at acquirer - December 26th, 2019 at 11:29:06
- CMO Key used for uniquely identifying type of message - "RECON_REQUEST"
- Name of file – File3_01010200108119359140019911 2.xml (File3_ORD.xml)

Sample file

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<File>
<Hdr>
<nFlVERSION>10</nFlVERSION>>
<nDtTmFlGen>191226112906</nDtTmFlGen>
</Hdr>
<File3>
<Rec RecNum="1">
    <nAcqInstCd>400002</nAcqInstCd>
    <nActnCd>00</nActnCd>
    <nAmtSet>000000000100</nAmtSet>
    <nAmtTxn>000000000100</nAmtTxn>
<nORD>01010200108119359140019911 2</nORD>
<nARD>40000290840012894365736</nARD>
    <nCcyCdSet>356</nCcyCdSet>
    <nCcyCdTxn>356</nCcyCdTxn>
    <nConvRtSet>00000001</nConvRtSet>
    <nCrdAcpBussCd>4111</nCrdAcpBussCd>
    <nCrdAcpCity>CENTRAL DELHI</nCrdAcpCity>
    <nCrdAcpCtryCd>IN</nCrdAcpCtryCd>
    <nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>
    <nCrdAcpLoc>ARAKHAMBHA ROAD MET</nCrdAcpLoc>
```

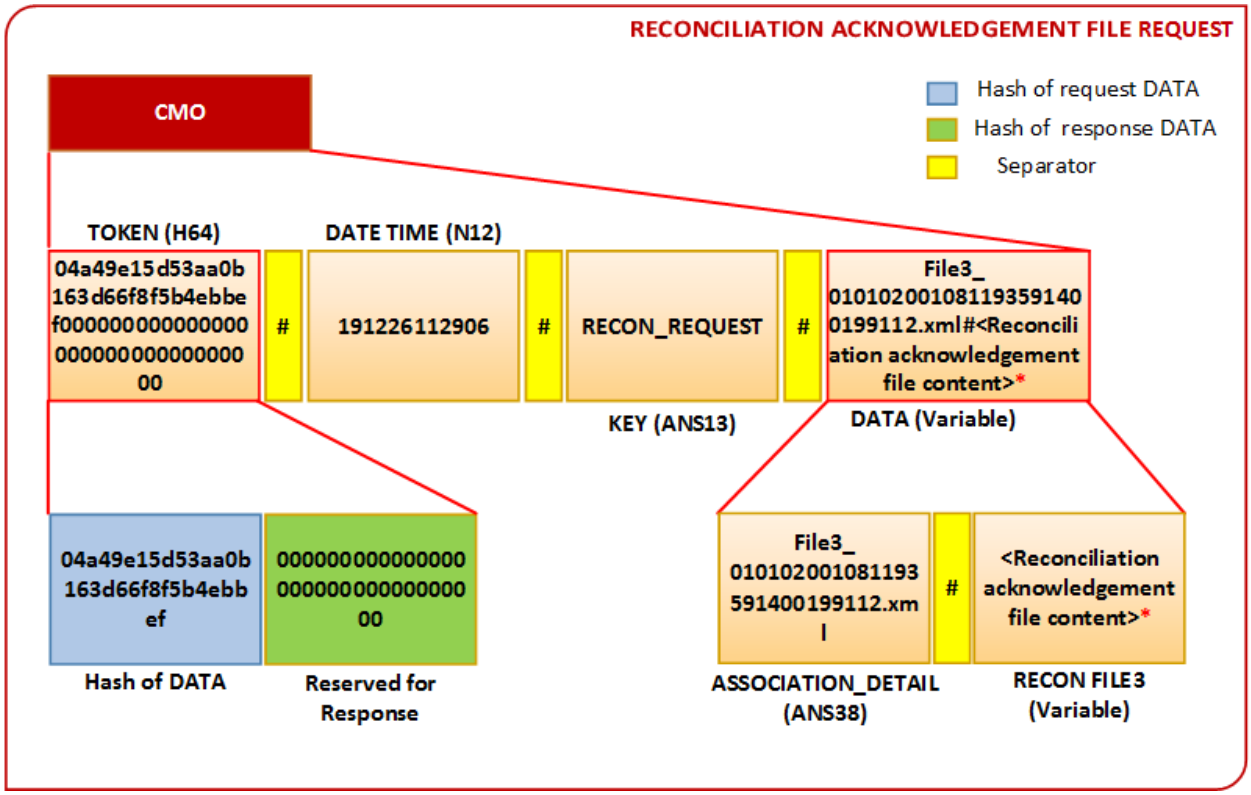
```
<nCrdAcpNm>BARAKHAMBHA ROAD METRO </nCrdAcpNm>
<nCrdAcpStNm>DL</nCrdAcpStNm>
<nCrdAcptTrmId>DL018561</nCrdAcptTrmId>
<nCrdAcpZipCd>000110001</nCrdAcpZipCd>
<nDtSet>190326</nDtSet>
<nDtTmLcTxn>190325133307</nDtTmLcTxn>
<nFunCd>260</nFunCd>
<nICCData>
 <n82>1900</n82>
 <n9F02>000000001000</n9F02>
 <n9F10>0FB5019400000020</n9F10>
 <n9F26>9f12d456b23c43d4</n9F26>
 <n9F27>40</n9F27>
 <n9F36>0056</n9F36>
</nICCData>
<nMerTelNum>ARAKHAMBHA </nMerTelNum>
<nMTI>1240</nMTI>
<nPAN>508538XXXXX1415</nPAN>
<nPosCondCd>00</nPosCondCd>
<nPosDataCd>701212400000000110001ARAKHAMBHA ROAD
MET</nPosDataCd>
<nPosEntMode>070</nPosEntMode>
<nProcCd>00</nProcCd>
<nProcSts>S</nProcSts>
<nRecNum>00000007</nRecNum>
<nServCd>620</nServCd>
<nSetDCInd>C</nSetDCInd>
<nTxnDesInstCd>SBIN0020001</nTxnDesInstCd>
<nTxnOrgInstCd>SBIN0020042</nTxnOrgInstCd>
<nUnFlNm>021SBIN00200421908500</nUnFlNm>
<Fee>
 <nFeeDCInd>D</nFeeDCInd>
```

```
        <nFeeAmt>2</nFeeAmt>
        <nFeeCcy>356</nFeeCcy>
        <nFeeTpCd>0001</nFeeTpCd>
        <nIntrchngCtg>3017</nIntrchngCtg>
      </Fee>
</Rec>
</File3>
<Tlr>
<nNoRecords>001</nNoRecords>
<nTotalAmt>000000000100</nTotalAmt>
</Tlr>
</File>
```

Figure 117: Sample XML (File 3)

Procedure:
Acquirer shall transfer the CMO packet containing consolidated acknowledgement file received from clearing switch to the PTO during file staging.

Figure 118: Reconciliation ACK CMO (File 3 Request)



*Refer sample XML file in Figure 117.

Reconciliation acknowledgement file response message using CMO

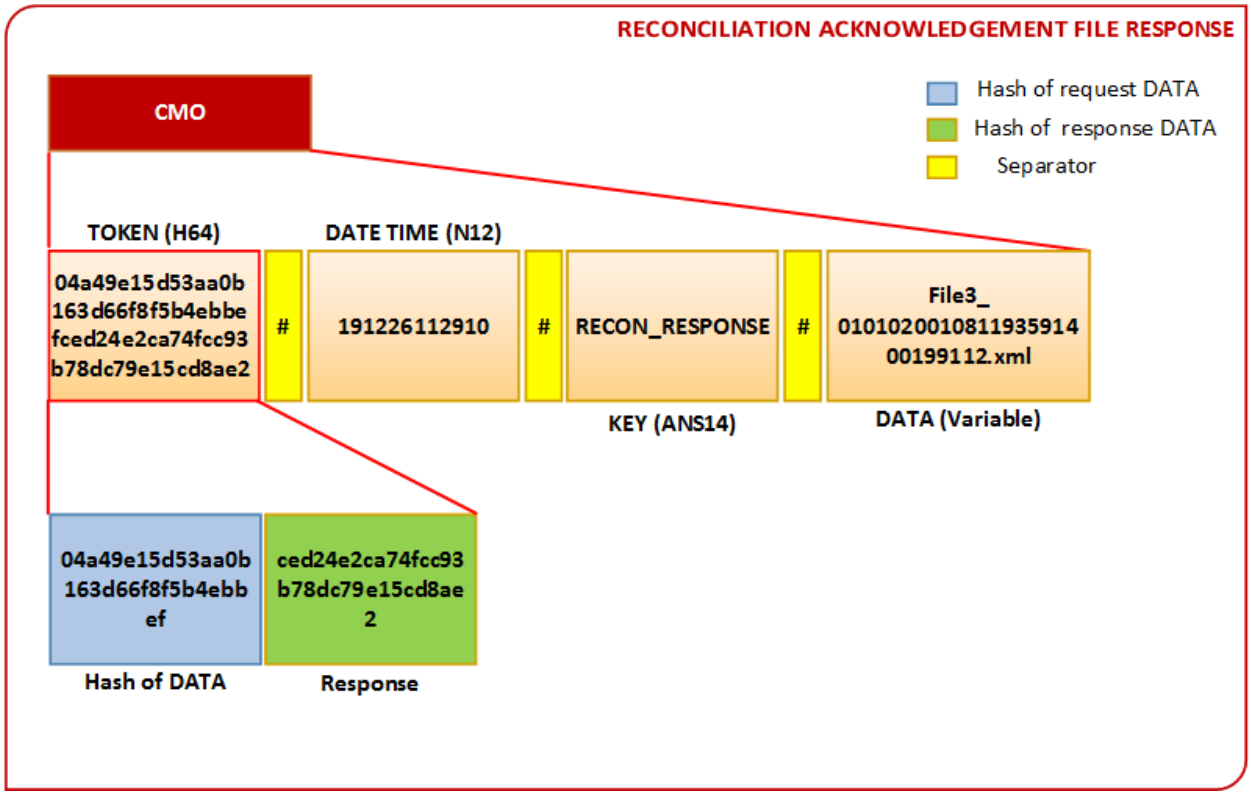
Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Reconciliation acknowledgement file response message generation time - December 26th' 2019 at 11:29:10
- CMO Key used for uniquely identifying type of message - "RECON_RESPONSE"
- Association details –File3_ 01010200108119359140 0199112.xml (Name of Reconciliation acknowledgement file sent by acquirer in request message)

Procedure:

- Once AFC backend receives the request message from acquirer, it constructs the Reconciliation acknowledgement file response message using CMO and transmits back to Acquirer.

Figure 119: Reconciliation ACK CMO (File 3 Response)



2.7.4. File 4: Fund Settlement File

After transferring the fund to operator account, Acquirer shall share the fund settlement report with Operator. The file shall be in xml format. The structure of the file is mentioned in the Table below.

Table 94: Data Elements of File 4 (Fund Settlement)

S. No.	Data Fields	Data Type	XML Tags	Optional/Mandatory
1.	Settled Date(in YYMMDDHHmmss format)	N12	<nSetDt>	M
2.	Settled Account No.	N15	<nSetAcc>	M
3.	Card Acceptor ID Code	ANS15	<nCrdAcpIDCd>	M
4.	Card Acceptor	ANS8	<nCrdAcptTrmId>	M

	Terminal ID			
5.	ORD7	AN32	<nORD>	M
6.	TC7	B8	<n9F26>	M
7.	Amount	N12	<nAmtTxn>	M
8.	RRN7	AN12	<nRRN>	M

The structure of the message used for sending File 4 to Operator is as follows.

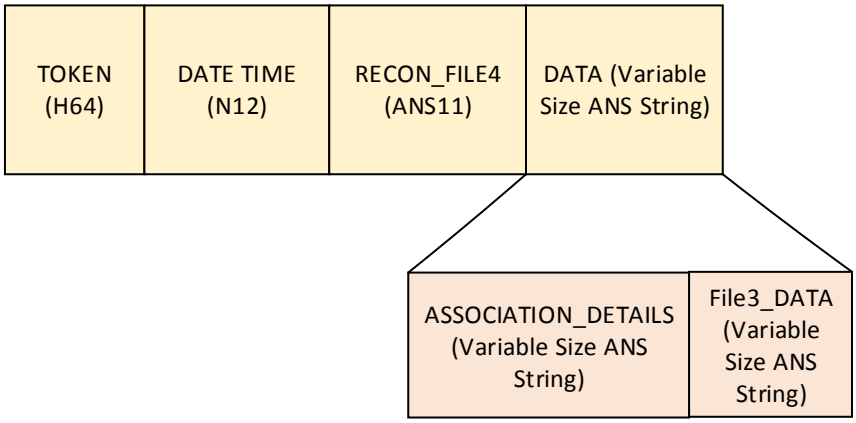


Figure 120: File 4 message structure

Table 95: File 4 Message Structure

S. No.	Configuration Parameter	Data Size	Data Format	Description
1	**ASSOCIATION_ DETAILS**	Variable	ANS	It indicates the association details of the File 4 file. Association details shall be the File 4 Name. It shall be in xml format.
2	**File4_DATA**	Depending upon the data content(Variable)	ANS	It holds the content of the File 4 file which shall be in xml format.

Table 96: File 4 Data structure

File Format	Data Structure	File4_DATA	Data Elements	XML Tags
XML	**Header(<hdr>)**	Version number of Structure	Refer Table 86	<nFlVERSION>
		Generation time		<nDtTmFlGen>
	File4 block(<File4>)	Record 1 **<Rec RecNum="1">**	Refer Table 94	Refer Table 94
		Record 2 **<Rec RecNum="2">**	Refer Table 94	Refer Table 94
		.		
		.		
		Record n **<Rec RecNum="n">**	Refer Table 94	Refer Table 94
	Trailer(<trl>)	No of Records	Refer Table 87	<nNoRecords>
		Total Amount		<nTotalAmt>

Example 16: Transmission of File 4 - Fund Settlement File

Fund Settlement File request message using CMO

Assumption:

- CMO Parameters separator - “#”
- Algorithm used for generating TOKEN – MD5
- Fund settlement file message generation time at acquirer - December 26th’ 2019 at 11:29:06
- CMO Key used for uniquely identifying type of message - “RECON_REQUEST”
- Name of file – File4_ 01010200108119359140019911 2.xml (File4_ORD.xml)

Sample File

Figure 121: Sample XML (File 4)

```
<? xml version="1.0" encoding="UTF-8" standalone="no"?>

<File>

        <Hdr>

                <nFlVERSION>10</nFlVERSION>>

                <nDtTmFlGen>191226112906</nDtTmFlGen>

        </Hdr>

        <File4>

                <Rec RecNum="1">

                <nSetDt>191226102506</nSetDt>

                <nSetAcc>190245611091234</nSetAcc>

                <nORD>01010200108119359140019911 2</nORD>

                <nRRN>012894365736</nRRN>

                <nAmtTxn>000000000100</nAmtTxn>

                <nCrdAcptTrmId> A1123110</nCrdAcptTrmId>

                <nCrdAcpIDCd>000056765432198</nCrdAcpIDCd>

                <n9F26>9f12d456b23c43d4</n9F26>

                </Rec>
```

```
        </File4>
        <Tlr>
                <nNoRecords>001</nNoRecords>
                <nTotalAmt>000000000100</nTotalAmt>
        </Tlr>
</File>
```

Procedure:

After transferring the fund to PTO account, Acquirer shall transfer the CMO packet containing fund settlement report in XML format with PTO.

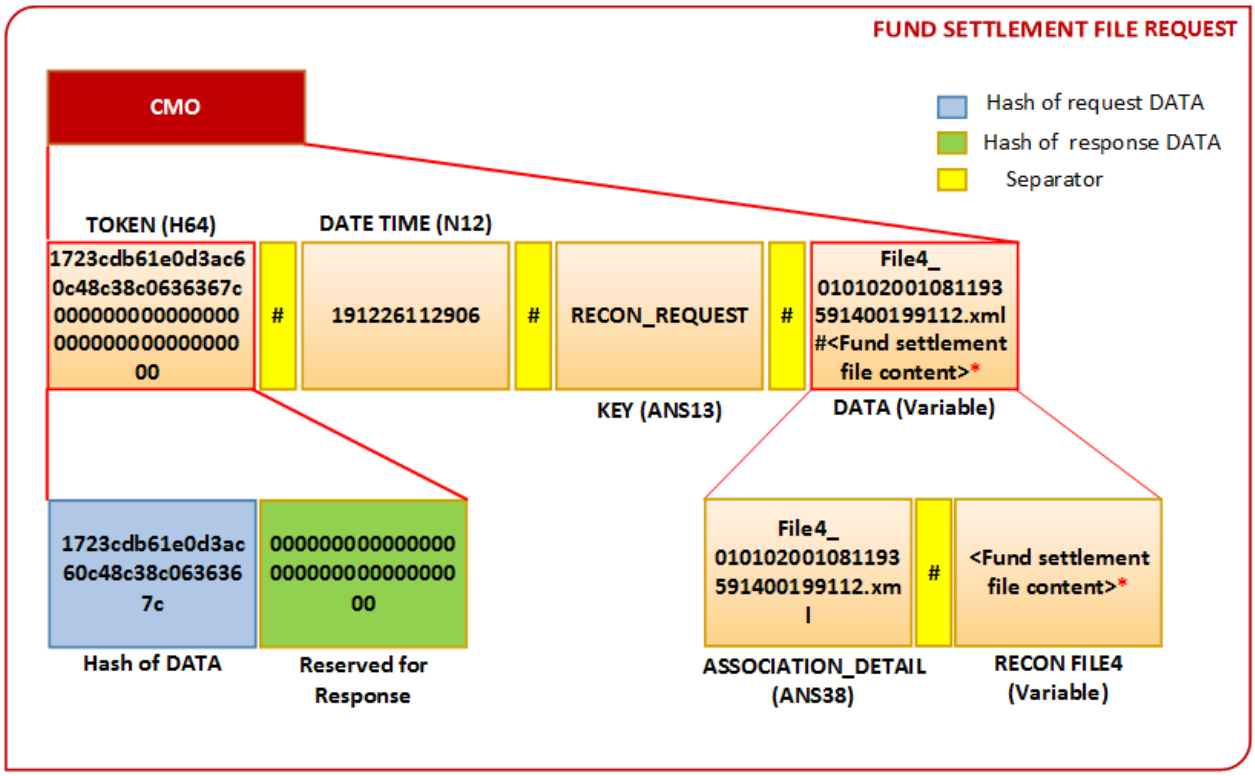


Figure 122: Fund Settlement CMO (File 4 Request)

*Refer to the sample file in Figure 121.

Fund Settlement File response message using CMO

Assumption:

- CMO Parameters separator - "#"
- Algorithm used for generating TOKEN(hash)of DATA – MD5
- Fund Settlement File response message generation time - December 26th' 2019 at 11:29:10
- CMO Key used for uniquely identifying type of message - "RECON_RESPONSE"
- Association details –File4_ 01010200108119359140019911 2.xml (Name of Fund Settlement File sent by acquirer in request message)

Procedure:

- Once AFC backend receives the request message from acquirer, it constructs the Fund Settlement File response message using CMO and transmits back to Acquirer.
-

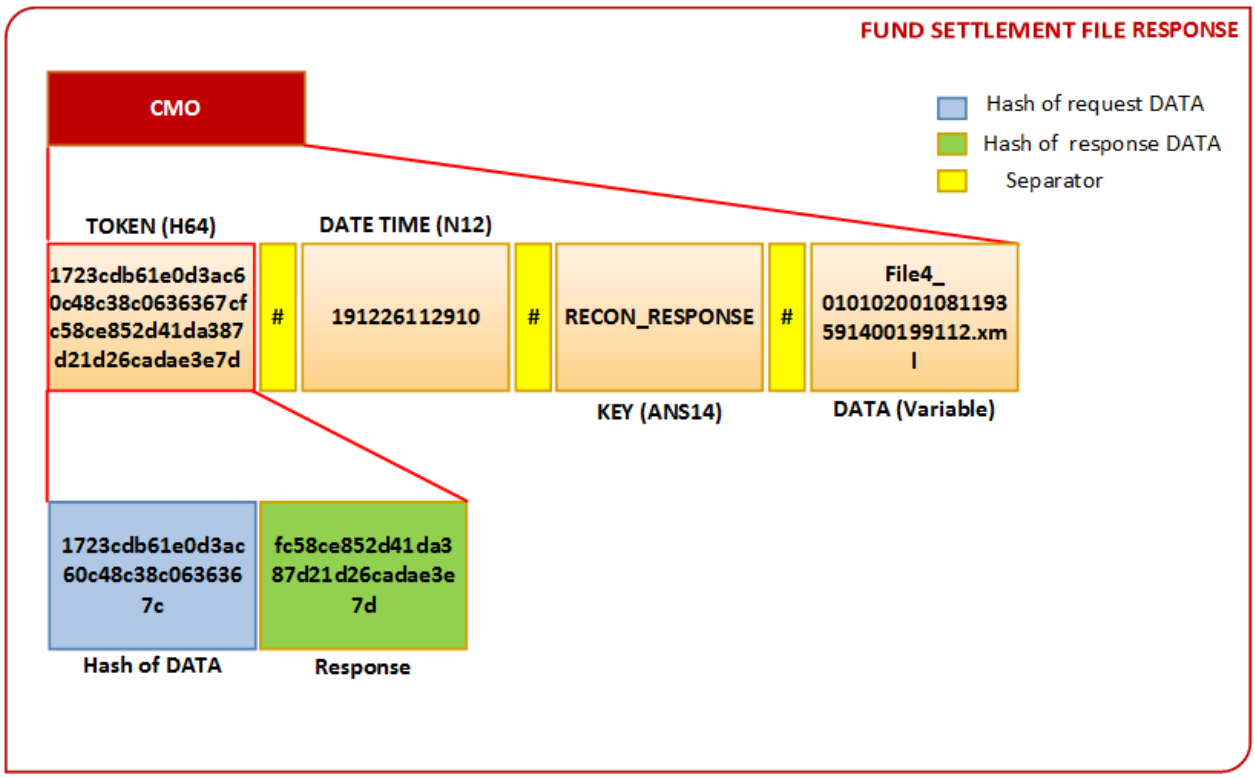


Figure 123: Fund Settlement CMO (File 4 Response)

2.8. Dispute Settlement

For dispute settlement, separate request shall be generated from concerned party to Operator/Acquirer following existing Standard practice. Disputes can happen under

various circumstances and are driven by business rules and financial settlement requirements. For example, a dispute may arise when –

a) No acknowledgement data is received for any financial transaction from the bank
b) Claims made by the operator are not honored by the issuer.
c) Commuter/user raises a claim that is disputed by the Operator.
d) Torn transactions (transactions that are difficult to trace) occur on the terminal.

**********End of Chapter 3 **********

Chapter 4

Interface Specification of NCMC Ecosystem

PART VII:
Gate -Terminal Interface

Centre for Development of Advanced Computing (CDAC), Noida
Ministry of Electronics &Information Technology (MeitY),
Government of India

Contents

List of Tables

List of Figures

Revision History

Date	Version	Author	Comments
May 18, 2017	V 0.1	CDAC, Noida	First draft release
Nov07,2019	V0.2	CDAC,Noida	Second draft release with following updates: • Section 2.2: "Gate–Sensor Interaction Specification" added figure 1 Gate Sensors. • Section 2.2: defined areas of Gate. • Section 2.3": Added Gate Terminal Architecture figure 2. • Section 2.3.1, Added figure 3-6. • Section 1.2 item 7. Added bi-directional movement. • Table 98 item 6, removed the type of material. v • Table 98 item 3, added earthling in electrical specification. • Section 2.2 item ii, extra clearance added in "Gate remain in open state timing" • Table-98 updated ▪ Min Gate width specification changed ▪ Specification for Terminal space added ▪ Display size updated • Additional commands in added Normal mode. • Additional commands and conventions added in Normal mode. • Operator's inputs on Specifications updated.
February 28,2020	V 1.2	CDAC,Noida	Added and updated inputs of various Operator on NCMC Specifications.

Abbreviations

The table below shows the abbreviations used throughout this document.

Table 97: Abbreviations

Abbreviation	
AFC	Automated Fare Collection
EMV	Euro Master Visa
GCU	Gate Control Unit
IR	Infra-Red
ISO	International Organization for Standardization
JSON	Java Script Object Notation
LED	Light Emitting Diode
NCMC	National Common Mobility Card
PTO	Public Transport Operator
RF	Radio Frequency
TLV	Tag-Length-Value
UPS	Uninterrupted Power Supply
VT	Validation Terminal
XML	Extensible Markup Language

1. Introduction

The Validation Terminals (VT) in the AFC System facilitate the entry and exit of users into the operator's premises. This is an automated process that requires the VT to be connected physically to an electro-mechanical machine, which shall be referred to in this document as AFC Gate or simply Gate. This document shall describe how this process of communication between the VT and Gate takes place.

1.1. Scope

This document only describes the interface requirement for communication between AFC Gate and Validation Terminal. The technical specifications of GCU/PLC are not within the scope of this document.

1.2. Terms and Definitions

Some general definitions related to AFC Gate are described here.

1.2.1. Validation Terminal

The Validation Terminal (VT) checks the validity and access rights of the NCMC card presented to it and issues commands to the Gate Control Unit accordingly to operate the Gate as per the requirements of the system. For detailed technical description of Server interface with VT, please refer to the NCMC-Chapter 3 (Part V): Terminal - AFC Backend Interface.

1.2.2. AFC System

The AFC System consists of the Server facing the Terminal and the acquirer. The AFC System does not consist of the Validation Terminal and Gate. For detailed description of Server interface with Validation Terminal, please refer to the NCMC-Chapter 3 (Part V): Terminal - AFC Backend Communication.

1.2.3. Gate Control Unit

The Gate Control Unit handles inputs from various sensors, peripherals and controls the operation of Gate. In addition to Gate control, capability to activate or deactivate the indicators, alarms and sensors are also part of the GCU. All the commands and logical

interfaces may be implemented in the GCU. The commands to the GCU from the Validation Terminal are sent over the RS232 serial port.

1.2.4. Operational Modes

The following are the operational modes of the AFC Gate.

1.2.4.1. Normal Mode

This is the default operating mode of the AFC Gate, wherein the Gate is opened by the VT after user input processing and is automatically closed after the user has exited through the Gate or the Gate Open period has timed out. In this mode, the sensor logic and Gate peripheral operation will be controlled by GCU. The Validation Terminal may only send Gate Open command and GCU will execute the command. But whether that execution becomes successful or not cannot be pre-determined. When the Gate is in Normal mode, the GCU controls all configurations/operations of the associated peripherals including indication configuration, buzzer configuration, Terminal/Display configuration, directional configuration (A→B, B→A or A↔B), etc. Emergency commands must be available in Normal mode of operation. The GCU ensures that emergency commands take effect immediately on priority basis.

1.2.4.2. Manual Mode

In this operating mode, the Gate operations are kept minimal. Manual supervision may be used for access control (but not limited to) for increasing Gate throughput as well as to take control over GCU. Most of the automatic functionality of GCU may be overridden by terminal in this mode. Emergency commands must also be available in manual mode of operation. Gate control based on the sensor inputs are disabled by GCU in this mode. Terminals can issue direct command to control Gate operations.

1.2.4.3. Maintenance Mode

When an AFC Gate enters into the Maintenance mode, the VT as well as GCU will also enter into maintenance mode and VT will not receive any command from the AFC backend. Under this mode, Gate peripherals and its performance can be

tested. The Gate will resume its normal mode of operation on a manual hardware reset. Emergency commands are not applicable in this mode of operation.

1.2.4.4. Default Mode

Normal Mode is the default mode of the AFC Gate. Changing into other modes is done exclusively by the “Mode Change” command. After power-up, the system must stay in Normal Mode.

1.2.5. Busy State

The time period during which the Gate is executing a command shall be referred as the time when it is “busy”. If any new command is received before, Gate has sent the response of previous command it will ignore the new command. However, in case of emergency commands, the GCU must accept the command on a priority basis ignoring all internal processes and execute the Emergency command.

1.2.6. Indeterminate State

It is quite possible that the AFC Gate may sometimes end up in an “indeterminate” state even though the communication between the VT with the GCU appears to be active. In such case GCU does not respond to any commands sent by VT. If in case system goes into an indeterminate state, the Gate should self-recover and normal mode of operations must resume within 10 seconds.

1.2.7. Change Direction

The AFC Gate defines the two ends of a lane, logically as A and B respectively. Hence any AFC Gate will support movement in one of three possible directions: -

i. Unidirectional A→B

Or

ii. Unidirectional B→A

Or

iii. Bi-directional A↔B.

The default direction of the Gate is A to B (A→B). In bi-directional mode, Gate will work on first come first serve basis.

1.3. References

Table 98: Reference

S. No	References
1.	ISO 13406-2: 2001- Ergonomic requirements for work with visual displays based on flat panels
2.	ISO 9241-303:2011- Requirements for electronic visual displays
3.	ISO 11064: Ergonomic design of control centres
4.	ISO 8201:1987 Acoustics - Audible emergency evacuation signal
5.	ISO 13475-1:1999 Acoustics - Stationary audible warning devices used outdoors- Part 1: Field measurements for determination of sound emission quantities
6.	ISO/TS 13475-2:2000 Acoustics - Stationary audible warning devices used outdoors -- Part 2: Precision methods for determination of sound emission quantities
7.	ISO 3864-1:2011 Graphical symbols - Safety colours and safety signs - Part 1: Design principles for safety signs and safety markings
8.	ISO 8995-1:2002 Lighting of work places
9.	ISO 10068:2012 Mechanical vibration and shock - Mechanical impedance of the human hand-arm system at the driving point
10.	ISO 13482:2014 Robots and robotic devices - Safety requirements for personal care robots

2. AFC Gate

2.1. AFC Gate Specification

The table below describes the technical specifications of AFC Gate system:

Table 99: Technical Specification of AFC Gate

S. No.	Name	Item Description
1.	Lane width	Operator specific
2.	Flap/Turnstile/ Gate Opening/Closing time	Less than 600ms (must be tested individually)
3.	Power supply	Single phase, AC-220+/- 10 % V, 50 Hz with provision for earthing of electric conductive parts of the cabinet.
4.	Communication interface	RS232 (Additional RS-485,Ethernet or other interface communication may be provided)
5.	Barrier type	Retractable / Operator Specific
6.	Cabinet	Operator specific
7.	Sensors for movement/tailgating	Suitable number of sensors should be installed in a Gate for detection of movement, tailgating, jumping over, and crawling under.
8.	Gate functions	The Gate should have the capability for configuration, operation, monitoring, control of sensors and Indicators and should have the provision to debug/control through the provided communication port. Gate should be configurable to both the directions.
9.	Direction	Unidirectional (A→Bor B→A)and Bi-directional (A↔B).
10.	Gate base to Display height	Operator Specific/ Recommended 1000 mm +/- 100 mm (optimum view of display and ease of access to the validation terminal)
11.	Gate height	Operator specific/Recommended more than 1100 mm

12.	Gate length	Operator specific
13.	Display Platform size and mounting space	At least 170mm x 110mm (LxW) with maximum 80mm depth of display
14.	Display protective panel	Fully transparent and hardened
15.	Audio Indication / Alarm	Distinguishable and configurable Sound for danger, warning and indication. Operator may follow recommendations in ISO 8201:1987, ISO 13475-1:1999, ISO 13475-2:2000
16.	Visual Indication / Alarm	Configurable light intensity for alarm, indication. Operator may follow recommendations in ISO 3864-1:2011, ISO 8995-1:2002
17.	Light Indicator shape, size	Operator specific parameters.
18.	Retractable Flap closing/opening thrust	Mechanical impact must not be harmful to human body. However, the thrust should not be very light to protect tail-gating and ideally it should configurable with suitable feed-back mechanism to sense the flap activity or tampering or tail-gating. Operator may follow recommendations in ISO 10068:2012, ISO 13482:2014
19.	Antenna Platform position	Right side of entry directions
20.	Size and mounting space for Antenna Platform of Validation Terminal	At least 200% size of ID1 Card in both width and breadth and same ratio as of ID1 card. Antenna space should be customizable to support mounting of antenna platform of variable sizes.
21.	Antenna protective panel	Hardened and relative RF signal permittivity should be near to zero value.

	of Validation Terminal	
22.	Size and mounting space for Validation Terminal	Maximum of 170mm x 170mm x 250 mm (LxWxH). Easily deployable mechanical interface between AFC Gate and Validation Terminal enclosure should be provided.
23.	Provision for Mounting external/third party Hardware	UPS, Ethernet switch, host controller, power junction box, power adaptors etc.
24.	Cable Channels	Separate channels for power cables, data cables and RF cables.
25.	Firmware	The AFC Gate shall have a provision for remote update of its firmware.

2.2. Gate-Sensor Interaction Specification

Sufficient numbers of sensors should be installed in a Gate for movement, tailgating, jumping over, crawling under etc. Positions of the sensors are vendor specific to fulfill the purpose of detection of user or commuter activity inside the lane. The system may be designed as an array of optical sensors (IR sensor) or other sensors like camera, proximity sensors, ultrasonic sensors or panel of single/multiple types of sensors. The GCU logic must not reach an indeterminate state even if the user performs an illegal operation. If in case system goes into an indeterminate state, then normal mode of operations must resume within 10 seconds.

The figure below describes a possible scenario for IR sensor placement mounted on the transit AFC flap Gate. The diagram and its operation are completely exemplary and individual Operator may customize the design as per their requirement.

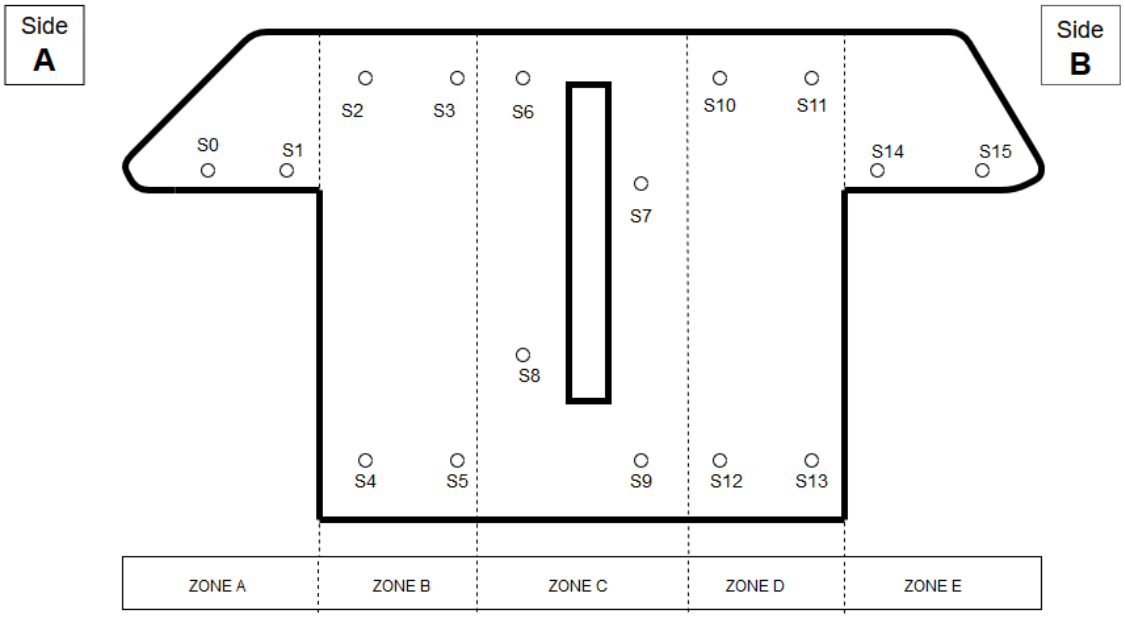


Figure 124: Indicative Gate Scenario

Behavior of flap Gate with respect to IR sensors in normal mode:
According to the position of the sixteen IR sensors (S0 – S15) and direction of entry/exit as shown in the figure above, the entire Gate is divided into 5 zones – Zone A through Zone E.

Scenario for commuter entering at Side A and Exiting at Side B

i. Zone A– The entry side of the Gate begins with this area, having two IR sensors (S0 and S1). This area detects a commuter while showing NCMC card to the Validation Terminal in case of Entry.

ii. Zone B– After crossing Zone A, the commuter enters Zone B, having four IR sensors (S2, S3, S4 and S5). The commuter is supposed to come in this area after the flaps open. The Gate gives alarm/indicators if a commuter enters in this area before the “flap” opens. This area is calibrated in such a manner that after the flaps open only one person is allowed. If the next commuter comes in this area before issuing another Gate open command, then the alarm/buzzer will set off along with the light indicator. The Gate shall remain open until passenger passes the flap area, Zone C. The duration for the flaps to remain open is a configurable parameter. The maximum passenger queue length is configurable based on specific command. Thereafter the Gate will be closed until the next Gate open command is issued.

iii. Zone C– This is the Flap area, which comes after Zone B, having four IR sensors (S6, S7, S8 and S9). The flaps remain open while commuter passes through this area. They close with alarm/buzzer if the commuter stays in this area for a longer time than the allowed configurable time.

iv. Zone D– This area comes after Zone C, having four IR sensors (S10, S11, S12, S13). The flap Gate closes as commuter passes through this area. If a commuter reverses back into this area, then an alarm will start ringing with a light indicator.

v. Zone E– The exit side of flap Gate begins with this area, having two IR sensors (S14 and S15). This area detects a commuter while showing NCMC card to the Validation Terminal in case of Entry.

Scenario for commuter entering at Side B and Exiting at Side A

This scenario is the exact inverse of the above scenario. In this case, the configurations of each zone w.r.t. the previous scenario will be as below:

i. Zone E corresponds to Zone A.

ii. Zone D corresponds to Zone B.

iii. Zone C remains exactly the same.

iv. Zone B corresponds to Zone D.

v. Zone A corresponds to Zone E.

2.3. Gate and Validation Terminal Interface Specification

The block diagram shown in the figure 2 below represents Gate-Terminal architecture. The communication interface of modules inside the Gate are outside the scope of this document.

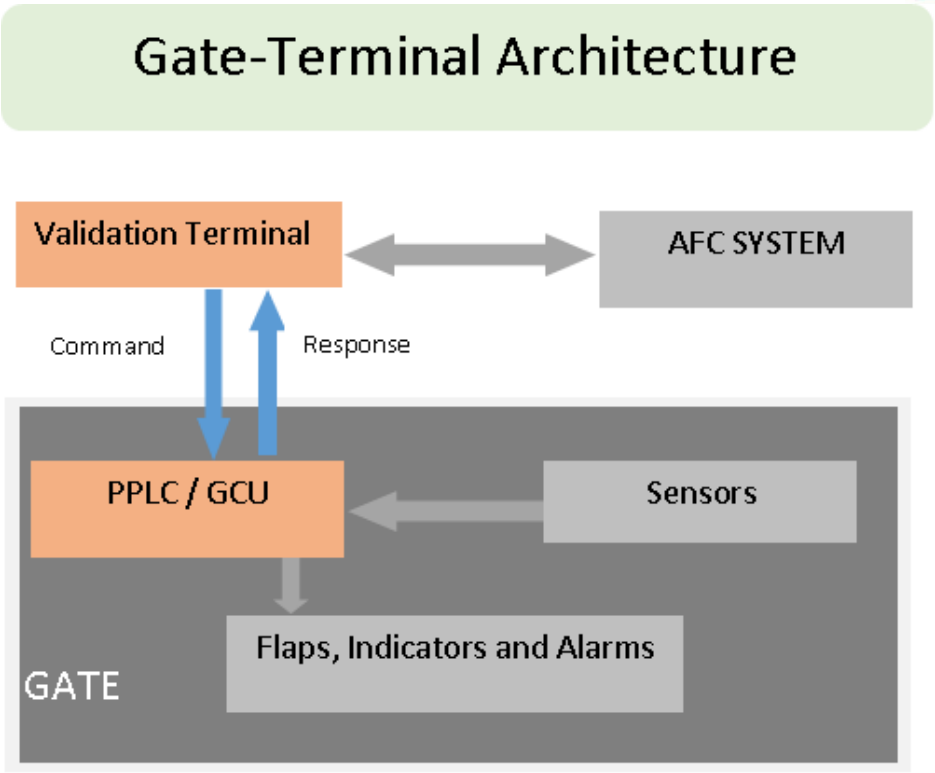


Figure 125: Gate-Terminal Architecture

2.3.1. Gate interface and communication

The communication of command / response will be done using the RS232 interface on the physical layer. Description about the technical details of RS232 protocol is not within the scope of this document. The communication parameters should be configurable between 9600 to 57600 baud with default data setting as 9600 baud, stop Bit-1, no parity, valid data as 8 bits.

The convention for data representation of commands and responses followed in this document are in ASCII and HEX. HEX values shall be represented in the form **"0xXX XX",** where each '**X**' is 1 nibble or 4 bits and one byte is shown as **XX**, while the ASCII characters are represented as such. For example, a 4-byte ASCII value AB23 can also be represented in hexadecimal as "0x41 42 32 33".

Commands are usually of 4 bytes and Responses are usually of 2 bytes. However, in case when the commands and responses contain data, the TLV encoding scheme is followed (Refer 2.3.3 for details).A special 4 byte "Sync" command [FF00 (ascii) or

“0x46 46 30 30” (hex)] has been kept for synchronization of devices at initialization phase. It shall be issued by Terminal after power-up or start of the communication. Table 4 below categorizes various types of errors and its responses from Gate in case of any failure during synchronization with Terminal. If terminal receives an error response, it may again send “Sync” byte to ensure and re-initialize the communication. GCU will send error response as E1 if it receives an unrecognized command from the Validation Terminal. Gate must send response within the timeout period configurable up to 500ms. If Gate does not send response within timeout period, the terminal will resend the same command up to the retries count (configurable max 5). If still there is no response from the Gate, terminal will generate an alarm and send report to AFC network.

If any new command is received at GCU before it has sent the response of previous command GCU will ignore the new command. However, in case of emergency commands, the GCU must accept the command on a priority basis ignoring all internal processes and execute the Emergency command.

The following examples will serve to illustrate the structure of the command and response exchange.

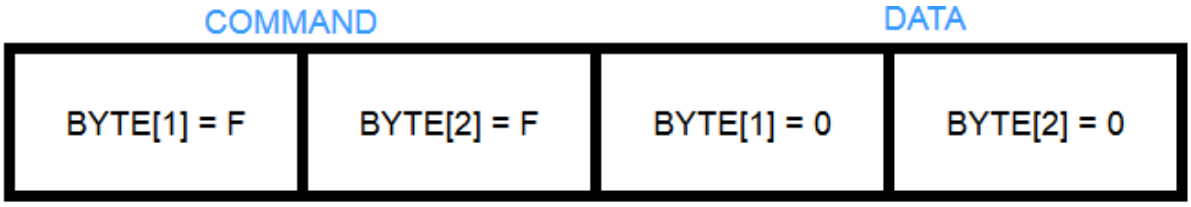


Figure 126: Sync Command

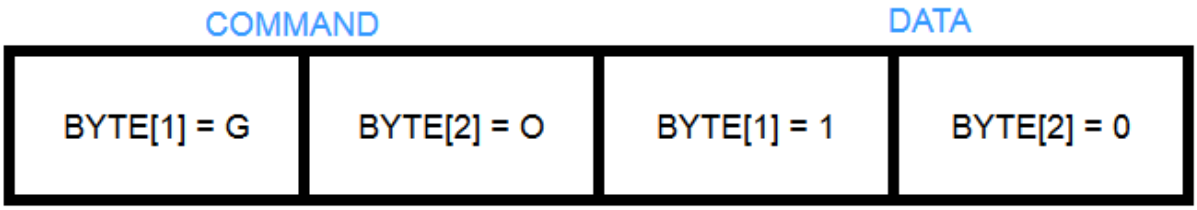


Figure 127: Gate Open Command

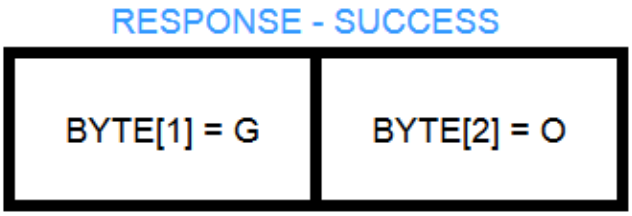


Figure 128: Gate Open Response

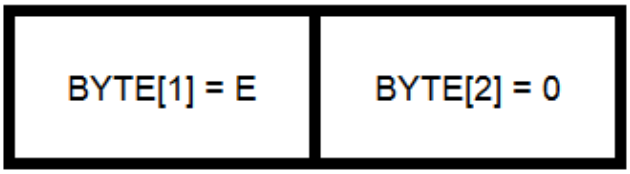


Figure 129: Error Response

Table 100: Error Codes

Code	Source	Sink	Error Description
E0	Gate	Terminal	Gate receives a command outside of its current mode of operation
E1	Gate	Terminal	Gate receives an invalid command/data

2.3.2. Gate – Terminal interface commands and data structure

The GCU may provide extra sets of commands to get enhanced feature of the system as per Operator requirements. The below Tables consists the various command lists to support AFC transit following NCMC specification. The Gate Interface commands are divided into three categories, which are as follows:

1. Normal Mode
2. Manual Mode
3. Maintenance Mode

Switching between modes are shown in the diagram below. After power on and

initialization, the AFC Gate always goes to the Normal Mode. Whenever a hard reboot is performed, the VT as well as the Gate are both rebooted and the AFC Gate follows the same restart process of power on and initialization before going into Normal Mode.

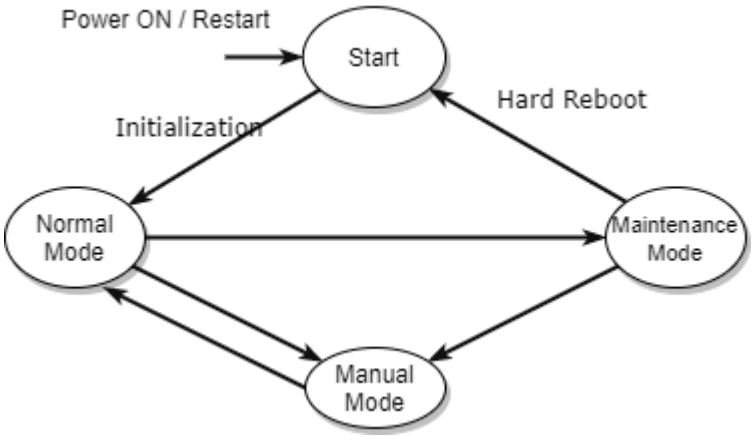


Figure 130: Illustrative Gate Operational Modes

2.3.3. Command and Response between VT and AFC Gate

Commands are usually consisting of 4 bytes and Responses are usually of 2 bytes. However, in case the commands and responses contain data, then the TLV encoding scheme is followed wherein:

- The first two bytes contains the command/response as the Tag.
- The third byte indicates the Length of the value field.
- The Value field indicates the data of interest.

The figure below illustrates the TLV format encoding.

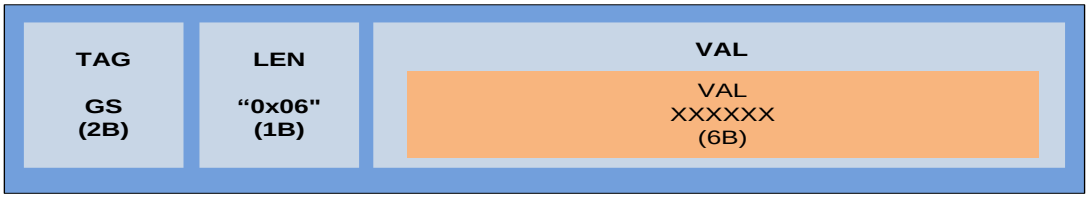


Figure 131: Illustrative TLV Encoding

The specific commands with respect to defined Modes are explained in the tables below.

Table 101: Normal Mode Commands

NORMAL MODE				
Command	**Code**	**Data**	**Response**	**Description**
Alarm On (Buzzer)	GO *e.g. Command: "0x474F"*	00 *e.g. Data: "0x3030"*	GO e.g. *Response: "0x474F"*	For unauthorized activity. Buzzers will ring with 50% Duty cycle(500ms in 1 sec)
Alarm off (buzzer)	GO	01	GO	Buzzers turned off.
Emergency Close	GO	02	GO	Infinite close and disable sensors. After GO04 (emergency release), GCU will take any other command of Normal mode.
Invalid Ticket Alarm	GO	03	GO	For Invalid ticket. Buzzers will ring with 70% Duty cycle(700ms in 1 sec)
Emergency Release	GO	04	GO	Out of Emergency. Resume Normal Mode operation.
Gate Open	GO	10	GO	Gate Open
Emergency Open	GO	12	GO	Infinite open and disable sensors. After GO04 (emergency release), GCU will take any other command of Normal mode.
Mode Change	MO	00 01 02	MO	00 – Switch to Normal Mode 01 – Switch to Manual Mode 02 -- Switch to Maintenance Mode
Out of Service	MO	03	MO	Flaps are closed and X (cross sign) indicators are displayed on both sides.

Command	Code	Data	Response	Description
Station Close	MO	04	MO	Flaps are closed and X (cross sign) indicators are displayed on both sides.
Gate A_B	GD	00	GD	Gate Direction: A to B with arrow at A &X (cross sign) at B
Gate B_A	GD	01	GD	Gate Direction B to A with arrow at B &X (cross sign) at A
Gate Bidirectional	GD	02	GD	Gate bidirectional with arrow at both sides. The entry command is issued on First-cum-First –Serve basis.
Aisle configuration-Normally open	AC	00	AC	Flap is Normally Open
Aisle configuration-Normally closed	AC	01	AC	Flap is Normally Close
Gate Status (Boolean Logic Status)	GK	SS	GK “0x02” “Byte2” “Byte1”	Status of all sensors. Each bit of these 2 bytes represent the status of each sensor as shown in Figure 132.
Firmware ID	FI	D0	FI “0xXX”FXXX	Length (0xXX) is the length of Firmware ID string. (from “0x00” – “0xFF”) ID is prefixed with fixed character F followed by variable ASCII characters as defined.
Device ID	ID	P0	ID “0xXX” DXXX	Length (0xXX) is the length of Device ID string. (from “0x00” – “0xFF”) ID is prefixed with fixed character D followed by variable ASCII characters as defined.

Command	Code	Data	Response	Description
Gate Queue Length	GQ	XX	GQ	Indicates group entry with simultaneous card tapping. Where XX defines the no. of persons count in ASCII.
Sensor Inactivity Time	TM	XX	TM	Feedback time post sensor failure. Wherein XX defined the time period in min. denoted in ASCII.
Flap Hold Time	TF	XX	TF	Time for Flap to remain open in case of commuters overstaying in flap area. Wherein XX defined the time in sec. denoted in ASCII.
Default Flap closing time	TR	XX	TR	Time for Flap to close in case of any inactivity after a valid entry. Wherein XX defined the time in sec. denoted in ASCII.
Buzzer Volume Control	BU	XX	BU	Intensity control for buzzer, Wherein XX may be between 01 to 10 denoted in ASCII.
Light intensity Control	TL	XX	TL	Intensity control for light, Wherein XX may be between 01 to 10 denoted in ASCII.
Health Status info	HS	00	HS“0x04” “Byte4” “Byte3” “Byte2” “Byte1”	For the structure of this response refer to Figure 133.

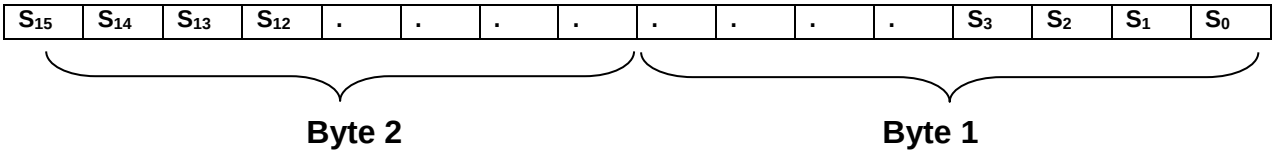


Figure 132: Sensor Status Representation

The 2 bytes represent the sensor status of all the 16 sensors of the Gate. The 0th bit of

Byte 1 represents sensor S_0 through to 8th bit of Byte 2 which represent sensor S_{15}.

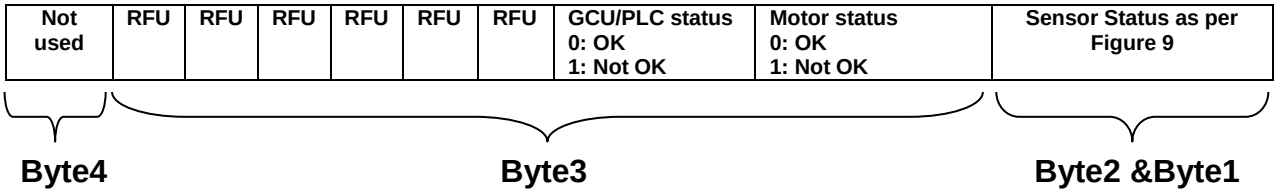


Figure 133: Health Status Info

Health Status Info is represented in 4 bytes. Byte1 and Byte2 provides status of all 16 sensors as defined in

.Byte3 is used to indicate the health status of Motor and GCU. Byte4 is left unused for any operator specific purpose.

Table 102: Alarms and Feedbacks from AFC Gate

Event	Type	Response	Description
Tailgating	Alarm with Feedback	GI“0x03”T“0x01 0x01” *e. g: Response: “0x47 49 03 54 01 01”*	Tailgating event indication *e.g. signifies the HEX data format.*
Wrong Entry	Alarm with Feedback	GI“0x03”W“0x01 01”	Wrong Entry event indication
Panel doors opened	Alarm with Feedback	GI“0x03”M“0x0101”	Panel Doors of Gate opened
Panel doors Closed	Alarm with Feedback	GI“0x03”M“0x0100”	Panel Doors of gate Closed
Returned on Mains Supply	Alarm with Feedback	GI“0x03”U“0x0101”	Gate returned to Mains supply from UPS mode
UPS power	Alarm with Feedback	GI“0x03”U“0x0100”	Gate switched to UPS Mode
Forceful Entry	Alarm with Feedback	GI“0x03”F“0x0101”	Forceful Entry event indication
Gate area obstructed	Alarm with Feedback	GI“0x03”A“0x0101”	Gate area blocked without validation (to detect unauthorized intrusion/occupancy)
Gate area Cleared	Feedback	GI“0x03”U“0x0100”	Gate area cleared for valid entry after unauthorized intrusion/occupancy.
Zone A	Feedback	GF “0x03” P “0x0000”	Feedback generated when person Enters in

			Zone A
Event	**Type**	**Response**	**Description**
Zone B	Feedback	GF “0x03” P “0x0001”	Feedback generated when person Enters in Zone B
Zone C	Feedback	GF “0x03” P “0x0010”	Feedback generated when person Enters in Zone C
Zone D	Feedback	GF “0x03” P “0x0011”	Feedback generated when person Enters in Zone D
Zone E	Feedback	GF “0x03” P “0x0100”	Feedback generated when person Enters in Zone E
Child Zone B	Feedback	GF “0x03” P “0x0101”	Feedback generated for crawling sensors when child Enters in Zone B
Child Zone C	Feedback	GF “0x03” P “0x0110”	Feedback generated for crawling sensors when child Enters in Zone C
Child Zone D	Feedback	GF “0x03” P “0x0111”	Feedback generated for crawling sensors when child Enters in Zone D
A to B side passage Alarm	Feedback	GF “0x03” P “0x1000”	Feedback generated when person passed from A Side to B Side.
B to A side passage Alarm	Feedback	GF “0x03” P “0x1001”	Feedback generated when person passed from B Side to A Side.
Child Pass	Feedback	GF “0x03” P “0x1010”	Feedback generated when child passed from A Side to B Side.
Child Pass	Feedback	GF “0x03” P “0x1011”	Feedback generated when child passed from B Side to A Side.
Crawling Alarm	Alarm with Feedback	GI “0x03” C “0x0101”	Feedback generated while crawling
Shut Down Alarm	Alarm with Feedback	GI “0x03” S “0x0101”	Feedback generated when Gate is under shutdown.
Sensor Failure Status	Feedback	GI ”0x03” X “Byte2” “Byte1”	Provides failure status of all sensors. The status is same as the GK

			command in Figure 132 Sensor inactivity time is configurable by TM command defined earlier.

Following commands are used to fetch values stored in the Audit registers provided within the Gate hardware. These registers are non-resettable and persistent in nature i.e. they retain their data even after power OFF/ON.

Table 103: Audit Registers Commands

Command	Code	Data	Response	Description
No. of times Gate went in Maintenance Mode.	RC *e.g. Command: 0x52 0x43*	01 *e.g. Data: 0x30 0x31*	RC “0x06” NM“0xXX XX XX XX” *e.g. Response: “0x52 43 06 4E 4D 00 00 00 00” upto “0x52 43 06 4E 4D FF FF FF FF”*	Stores count up to 4-billion (4bytes), where 0xXXXX is in HEX. *e.g. signifies the HEX data format*
Tailgating count A to B	RC	02	RC “0x06” TA“0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
Tailgating count B to A	RC	03	RC “0x06” TB “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
No. of times Gate went in UPS mode	RC	04	RC “0x06” UC “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
No. of passengers passed through the Gate in A to B direction.	RC	05	RC “0x06” PA“0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
No. of passengers passed through the Gate in B to A	RC	06	RC “0x06” PB “0xXX XX XX XX”	Stores count up to 4-billion (4bytes),Where 0xXXXX is in HEX.

direction.				
Wrong Entry Count A -B Direction	RC	07	RC “0x06” WA“0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
Wrong Entry Count B - A Direction	RC	08	RC “0x06” WB “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
Passenger count in bidirectional A to B	RC	09	RC “0x06” BA“0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
Passenger count in bidirectional B to A	RC	10	RC “0x06” BB “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
Forceful entry count	RC	11	RC “0x06” FE “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.
No. of time Gate went to out of service mode	RC	12	RC “0x06” OS “0xXX XX XX XX”	Stores count up to 4-billion (4bytes), Where 0xXXXX is in HEX.

Example for packet formation of feedback from AFC Gate

Following is an example of Tailgating explained below.

GI is the “Tag” and the value of “Length” is 3 Bytes where the first Byte in “Value” shall be the prefix in ASCII and next two bytes are the data of interest in Hex.

GI	**“0x03”**	**T “0x01” “0x01”**

TAG LENGTH VALUE

Figure 134: Feedback for Tailgating

Table 104: Manual Mode Commands

MANUAL MODE				
Command	**Code**	**Data**	**Response**	**Description**
Mode Change	MO	00	MO	00 – Switch to

		01 02		Normal Mode 01 – Switch to Manual Mode 02 -- Switch to Maintenance Mode
Out of Service	MO	03	MO	Flaps are closed and X (cross sign) indicators are displayed on both sides.
Station Close	MO	04	MO	Flaps are closed and X (cross sign) indicators are displayed on both sides.
Emergency Close	GO	02	GO	Infinite close and disable sensors. After GO04 (emergency release), GCU will take any other command of Normal mode.
Emergency Release	GO	04	GO	Out of Emergency. Resume Normal Mode operation.
Emergency Open	GO	12	GO	Infinite open and disable sensors. After GO04 (emergency release), GCU will take any other command of Normal mode.
Flap Open	GS	00	GS	Flap Opened
Flap Close	GC	00	GC	Flap Closed
Buzzer On	GS	01	GS	Buzzer ON
Buzzer Off	GC	01	GC	Buzzer OFF
Top Panel Red Light ON	GS	02	GS	Red light ON

Top Panel Red Light OFF	GC	02	GC	Red light OFF
Top Panel Green Light ON	GS	03	GS	Green light ON
Top Panel Green Light OFF	GC	03	GC	Green light OFF
Top Panel Blue Light ON	GS	04	GS	Blue light ON
Top Panel Blue Light OFF	GC	04	GC	Blue light OFF
Top Panel White Light ON	GS	05	GS	White light ON
Top Panel White Light OFF	GC	05	GC	White light OFF
Top Panel Yellow Light ON	GS	06	GS	Yellow light ON
Top Panel Yellow Light OFF	GC	06	GC	Yellow light OFF
Side A Arrow ON	GS	07	GS	Side A Arrow ON
Side A Arrow OFF	GC	07	GC	Side A Arrow OFF
Side A Cross ON	GS	08	GS	Side A Cross ON
Side A Cross OFF	GC	08	GC	Side A Cross OFF
Side B Arrow ON	GS	09	GS	Side B Arrow ON
Side B Arrow OFF	GC	09	GC	Side B Arrow OFF
Side B Cross ON	GS	10	GS	Side B Cross ON
Side B Cross OFF	GC	10	GC	Side B Cross OFF
Gate Status (Boolean Logic Status)	GK	SS	GK “0x02” “Byte2” “Byte1”	For sensor status refer Figure 132.
Sensor status test ON	GT	06	GT *Response: GI ”0x03” Y “Byte2” “Byte1”:* Test pass.	Feedback enabled for sensor status. Status of all sensors can be checked manually. For sensor status refer Figure 132.
Sensor status test OFF	GT	19	GT	Feedback disabled for sensor status

Table 105: Maintenance Mode commands

MAINTENANCE MODE				
Command	**Code**	**Data**	**Response**	**Description**
Flap Buzzer Test	GT	00	GT	Flap Buzzer ON/OFF 5 times.
Intrusion Buzzer Test	GT	01	GT	Intrusion buzzer ON/OFF 5 times.
Wrong Entry Buzzer Test	GT	02	GT	Wrong Entry buzzer ON/OFF 5 times.
Tailgating Buzzer Test	GT	03	GT	Tailgating buzzer ON/OFF 5 times.
Forceful Entry Buzzer Test	GT	04	GT	Forceful Entry buzzer ON/OFF 5 times.
Crawling Buzzer Test	GT	05	GT	Crawling buzzer ON/OFF 5 times.
Sensor status test ON	GT	06	GT *Response: GI "0x03" Y "Byte2" Byte1"* :Test pass.	Feedback enabled for sensor status. Status of all sensors can be checked manually. For sensor status refer Figure 132.
Sensor Status Test Off	GT	19	GT	Feedback disabled for sensor status
Red Light Test	GT	07	GT	ON/OFF for 5 times.
Green Light test	GT	08	GT	ON/OFF for 5 times.
Blue Light test	GT	09	GT	ON/OFF for 5 times.
A Side Arrow On/Off	GT	10	GT	ON/OFF for 5 times.
A Side Cross On/Off	GT	11	GT	ON/OFF for 5 times.
B Side Arrow On/Off	GT	12	GT	ON/OFF for 5 times.
B Side Cross On/Off	GT	13	GT	ON/OFF for 5 times.

Infinite Times Flap Test On	GT	14	GT	Flap open/close for infinite times enabled
Infinite Times Flap Testing Off	GT	15	GT	Flap open/close for infinite times disabled.
Rigorous Flap Test On Command	GT	16	GT	Rigorous Flap open/close test for infinite times enabled.
Rigorous Flap Test Off Command	GT	17	GT	Rigorous Flap open/close for infinite times disabled.
Built-In Self-Test	GT	18	GT “0x03” P “0x0101” :Test pass GT “0x03”F “0x0101” : Test failed	Self-Test Command for Flaps, Top panel lights, front panel lights, buzzer and GCU. The peripherals perform open/ON and close/OFF operation for 3 times.
Set Physical Id	PI	“0x0XX” DXXX	PI	Length (0x0XX) is the length of Physical ID string. (from “0x00” – “0xFF”)

Annexure I: Flow of Configuration data from AFC backend Server to Gate

Please refer **NCMC-Chapter 3 (Part V) Section 4.1** to observe the flow of configuration files from AFC backend to Validation Terminal.

The configuration file received by the VT from AFC server is in JSON or XML format. Once the configuration file for Gate is received by the VT, the commands corresponding to the Keys and Values would be sent to the AFC Gate for further execution.

Let us consider, a configuration file named **GATE_TRIGGER.json,** which issent for execution at AFC Gate: -

```
{
        "MODE"              :    "NORMAL_MODE",
         "DIRECTION"        :     "AB",
        "AISLE_MODE"        :    "DISABLE",
         "BUZZER_VOLUME"    :    "7"
}
```

The commands may be interpreted as -

a) For Key "MODE" the Value is "NORMAL_MODE" and the packet formed to send to the VT is as follows:

   The command "MO" with data "00" therefore "MO00"is sent to the AFC Gate by "Validation Terminal". In response to this command, the response "MO" is received after the execution.

b) For Key "Direction" the Value is "AB" and similarly the packet formed is as:

   The command "GD" with data "00" forms "GD00" it is sent to the AFC Gate by "Validation Terminal". In response to this command, the response "GD" is received and the direction is changed from A→B.

c) For Key "AISLE_MODE" the Value is "DISABLE" the command packet is like: Command is "AC" and data is "01" the packet "AC01" is sent to the AFC Gate by "Validation Terminal" and the flaps attain the normally closed state.

d) For Key "BUZZER_VOLUME" the Value is "7" the command packet is like: Command is "BU" and data is "07" the packet "BU07" is sent to the AFC Gate by "Validation Terminal" and the buzzer volume is set to 7.

   Similarly, any configuration may be sent to AFC Gate as per Operator's requirement.

Note: All commands must be sent one after another only if the previous command's response is successfully received or the time within which the response should be received is passed (500ms which is configurable).

**********End of Chapter 4**********

Conclusion

As directed by the working group committee with the final recommendation on 25th July, 2015, the NCMC specification has been made that will cater all possible major components in transit payment systems such as Card, Card- Terminal, Terminal, Terminal-Mechanical gate, Terminal-AFC, Terminal- Banking system interfaces. The specification document defines the physical interface channels, data structures that the systems must interpret and also the security measures necessary for the system.
Constituting the expert group, the newly drawn specification and system has been considered to scalable, vendor agnostic and has covered all the required major systems which is not overloaded by huge technical infrastructure alteration from present system and also not biased to legacy system.

The complete specification may be classified into two broad segments where the first segment covers NCMC card and Terminal specification which are followed by NPCI proposed qSPARC specification. The other segment comprises of Terminal-Gate, Terminal-AFC, Terminal/AFC-Banking interface which were drawn by CDAC.
The NCMC eco-system specification has been drawn as basic minimal requirement for indigenous and self-producible infrastructure in our country. The evolvement process needs a lot of refinement based on various common requirement of multiple PTO (Public Transport Operator), AFC manufacturers and also banking institutions.
Based on various comments and inputs from multiple stake-holders/ expert- group-members, it is also felt that, further detailed specifications of various internal components and sub-systems of the AFC eco-systems may be necessary for future, rapid development and also to help vendors to make inter-operable common systems across the country.