UNIVERSAL DATA INTEGRATION LAYER (UDIL) - MULTIVENDOR COMPLIANT ADVANCED METERING INFRASTRUCTURE (AMI) IN PAKISTAN FEBRUARY 19, 2019

SUSTAINABLE ENERGY FOR PAKISTAN (SEP) PROJECT

Submission Date: February 19, 2019

Contract No.: AID-OAA-I-13-00028
Task Order: AID-391-TO-16-00005
Activity Start Date and End Date: August 3, 2017 to April 26, 2021

Submitted by: Tetra Tech ES, Inc.
1320 North Courthouse Road, Suite 600
Arlington, VA 22201
Tel. +1-703-387-2100 | Fax +1-703-243-0953

www.tetratech.com

This report was produced for review by the United States Agency for International Development. It was prepared by Tetra Tech ES, Inc. for the Sustainable Energy for Pakistan (SEP) Project.

UNIVERSAL DATA INTEGRATION LAYER (UDIL) – MULTIVENDOR COMPLIANT ADVANCED METERING INFRASTRUCTURE IN PAKISTAN

FEBRUARY 19, 2019

SUSTAINABLE ENERGY FOR PAKISTAN (SEP) PROJECT

CONTENTS

1. EXECUTIVE SUMMARY 7

1.1 PROJECT OVERVIEW 7

1.1.1 BACKGROUND 7

1.1.2 SUSTAINABLE ENERGY FOR PAKISTAN PROJECT (SEP) 7

1.1.3 SEP SCOPE OF WORK 7

1.2 AMI GAP ANALYSIS 8

2. OPEN ARCHITECTURE-BASED AMI LANDSCAPE 10

2.1 UNIVERSAL DATA INTEGRATION LAYER (UDIL) 10

2.1.1 SYSTEM DESIGN ARCHITECTURE 10

2.1.2 GENERAL SCOPE 11

2.1.3 BUSINESS PROCESS SUPPORT 12

2.1.4 FUNCTIONAL REQUIREMENTS 12

3. UDIL DATA STRUCTURE 13

3.1 READ REQUESTS – TABULAR 13

3.1.1 INSTANTANEOUS DATA 13

3.1.2 BILLING DATA 14

3.1.3 MONTHLY_BILLING_DATA 15

3.1.4 LOAD PROFILE DATA 17

3.1.5 EVENTS 18

3.1.6 METER VISUALS 19

3.2 READ REQUESTS – AUTHORIZATION PROTOCOL 20

3.2.1 AUTHORIZATION SERVICE 20

3.3 READ REQUESTS – API BASED 20

3.4 WRITE REQUESTS – COMMANDS 21

3.4.1 AUX RELAY OPERATIONS 21

3.4.2 TIME SYNCHRONIZATION 22

3.4.3 SANCTIONED LOAD CONTROL 22

3.4.4 LOAD SHEDDING SCHEDULING 23

3.4.5 TIME OF USE CHANGE 24

3.4.6 DEVICE CREATION 24

3.4.7 UPDATE IP PORT 26

3.4.8 METER DATA SAMPLING 26

3.4.9 ACTIVATE METER OPTICAL PORT 27

3.4.10 WAKE UP SIM NUMBER UPDATE 28

3.4.11 METER STATUS UPDATE 29

3.5 ON-DEMAND REQUESTS – API BASED 29

3.5.1 TRANSACTION TYPES 30

3.5.2 TRANSACTION STATUS 30

3.5.3 TRANSACTION CANCEL 32

4. IMPLEMENTATION EXAMPLES 33
4.1 AUTHORIZATION SERVICE 33
4.2 AUX_RELAY_OPERATIONS 34
4.3 TIME_SYNCHRONIZATION 35
4.4 SANCTIONED_LOAD_CONTROL 37
4.5 LOAD_SHEDDING_SCHEDULING 38
4.6 TIME_OF_USE_CHANGE 40
4.7 DEVICE_CREATION 41
4.8 UPDATE_IP_PORT 43
4.9 METER_DATA_SAMPLING 44
4.10 ACTIVATE_METER_OPTICAL_PORT 45
4.11 WAKE_UP_SIM_NUMBER_UPDATE 47
4.12 METER_STATUS_UPDATE 48
4.13 ON-DEMAND REQUESTS 49
4.14 TRANSACTION_STATUS_REQUEST 50

ABBREVIATIONS & ACRONYMS

AEB	Area Electricity Board (former name for a DISCO)
AEDB	Alternative Energy Development Board
AMI	Advanced Metering Infrastructure
AMR	Automated Meter Reading
AT&C	Aggregate Technical and Commercial
B2B	Business-to-Business
BI	Business Intelligence
BW	Business Warehouse
CE (Ops)	Chief Engineer Operations
CE (P&E)	Chief Engineer Planning and Engineering
CIS	Customer Information System
CIM	Common Interface Model
COP	Chief of Party
CP	Commercial Procedures
CSD	Customer Service Director
CT	Current Transformer
DBMS	Data Base Management System
DISCO	State-owned Electricity Distribution Company
D&S	Design & Standards
ETL	Extract, Transform and Load
FESCO	Faisalabad Electric Supply Company Limited
FY	Financial year
GEPCO	Gujranwala Electric Power Company Limited
GIS	Geographic Information System
GoP	Government of Pakistan
HE	Head End System
HESCO	Hyderabad Electric Supply Company Limited
HT	High Tension, or High Voltage Level
IESCO	Islamabad Electric Supply Company Limited
KWh	Kilo Watt Hour
KPIs	Key Performance Indicators
LDI	USAID's Load Data Improvement Project
LESCO	Lahore Electric Supply Company Limited
LT	Low Tension, or Low Voltage Level
MDC	Meter Data Collector
MDM	Meter Data Management
MEPCO	Multan Electric Power Company Limited
M&E	Monitoring and Evaluation
MoE	Ministry of Energy
MW	Megawatt
NEPRA	National Electric Power Regulatory Authority

NPCC	National Power Control Center
NTDC	National Transmission and Despatch Company
PEPCO	Pakistan Electric Power Company Limited
PESCO	Peshawar Electric Supply Company Limited
PDC	Power Distribution Control Center
PITC	Pakistan Information Technology Company
QESCO	Quetta Electric Supply Company Limited
RO	Revenue Officer
SDO	Sub Divisional Officer
SE	Superintending Engineer
SEP	USAID's Sustainable Energy for Pakistan Project
SEPCO	Sukkur Electric Power Company Limited
S&S	Standards & Specification Department under PEPCO (formerly D&S department)
SQL	Structured Query Language
TESCO	Trible Area Electric Supply Company Limited
TC	SEP Technical Component
UDIL	Universal Data Integration Layer
USAID	United States Agency for International Development
WAPDA	Water and Power Development Authority
XEN	Executive Engineer

1. EXECUTIVE SUMMARY

1.1 PROJECT OVERVIEW

1.1.1 BACKGROUND

As part of the USAID Power Distribution Program (PDP), smart meters were installed on feeders at the grid stations across all the DISCOs for recording load profiles and real-time monitoring of power dispatch. The project was further extended to provide support to three DISCOs (MEPCO, PESCO and HESCO) with smart meters installation at the customer level. Customers were selected based on certain parameters, such as high load profiles and agricultural context (tube-wells). The project had mainly three components; provision of smart meters, installation of headend applications, and development of a web-based portal for monitoring of the installed smart meters. Initially, the solution was designed to support integration of all types and makes of smart meters, but unfortunately due to lack of understanding and technical capacity, the Advanced Metering Infrastructure (AMI) project remained confined to a single vendor specific solution and ended up in a "vendor lock" situation. In turn, it prevented DISCOs from further expanding AMI solution both at the grid and consumer levels. The installed system does not provide support to the smart meters from multiple vendors and therefore there is an immediate need to find solution to provide a level playing field to all meter manufacturers enabling them to integrate their devices with the existing system.

As a part of the solution. Sustainable Energy for Pakistan Project (SEP) intended to standardize Universal Data Integration Layer (UDIL) structure and introduce a Meter Data Management (MDM) solution along with Business Intelligence (BI) tool to expose multiple vendor Head Ends (HEs) in common AMI landscape and provide structured information through analytics over dashboards enabling DISCOs to access near real-time information for prudent decision making. Such a vendor independent plug-and-play arrangement will help DISCOs to procure smart meters through competitive bidding leading to the most cost-effective AMI deployment.

1.1.2 SUSTAINABLE ENERGY FOR PAKISTAN PROJECT (SEP)

With significant increase in generation during the previous government, the gap between supply and demand has narrowed down significantly and sufficient installed base of generation is now available to meet the demand. Despite projected surplus generation, many risks and constraints to the continued development and sustainable supply of electricity remains. The most significant of all these is accumulation of circular debt at a rapid pace in view of increased supply of electricity in the distribution system with excessively high technical as well as commercial losses. Unless the DISCOs are disciplined to manage their businesses and improve financial performance, such effort is not self-sustaining. Cash starved DISCOs would not be able to invest in infrastructure which is necessary to support the transmission and distribution of increased supply of power, which in turn is necessary to maintain reliability of the grid.

High loss DISCOs in general, consume significant chunk of the country's total generation which puts excessive financial burden on the entire distribution sector and is an impediment to the economic growth of the country. DISCOs' business processes are heavily reliant on manual processing, supplemented by information technology components allowing for errors and potential manipulation of results. The limited capacity of its staff to embrace the latest technology and the inability to meet with the growing challenges, greatly compromise overall DISCOs' performance. In this case, outdated policies and procedures, poor governance and inefficient work practices are all detrimental to the GOP efforts to enhance value of all public-sector DISCOs for future privatization.

1.1.3 SEP SCOPE OF WORK

The objective of USAID's Sustainable Energy for Pakistan (SEP) Project is to help Pakistan attract private sector engagement in developing sustainable, clean energy future for the country through its

various activities across all segments of the clean energy domain; generation, transmission, distribution, and governance.

The SEP Project activities are categorized into following four (4) Technical Components:

Technical Component 1 (TC 1) – Create a credit-worthy business environment that attracts private sector investors in a competitive, transparent energy market accessible to all stakeholders

Technical Component 2 (TC 2) – Transform the transmission system operator – the National Transmission and Despatch Company (NTDC) into an entity capable of managing and expanding the national grid while ensuring reliable, efficient, and stable transmission and dispatch services, and support the market operator and regulator (Central Power Purchasing Agency Limited (CPPA-G) and National Electric Power Regulatory Authority (NEPRA)) in transitioning to an open marketplace.

Technical Component 3 (TC 3) – Employ a systematic methodology for utility turnaround, developed and implemented for electricity distribution programs, delivering dramatic reduction in Aggregate Technical and Commercial (AT&C) losses and effectively enhancing the value of State-owned electricity distribution company (DISCOs) for privatization.

Technical Component 4 (TC 4) – Overcome specific barriers to clean energy (CE) investments through suitable policy, regulatory, or legislative amendments and transaction support services.

Based on SEP approved work plan for TC3, the Ministry of Energy (MOE) proposed to USAID to undertake list of tasks under the SEP project mainly aimed at reduction of AT&C losses and improving operational efficiency especially for the poor performing DISCOs.

Given the limited scope and resources for the commodity support under the USAID SEP project, it was envisaged to remain focused on selected priority tasks for providing technical assistance leading to improved financial and operational efficiency of selected DISCOs. SEP TC3 team together with USAID reviewed the list of tasks provided by the MOE to select the ones which are aligned with the scope and TOR of the USAID's SEP project. Successive consultation meetings were held with Pakistan Electric Power Company (PEPCO) and PESCO to develop a suitable technical assistance strategy.

Keeping in view the earlier selection of PESCO and MEPCO as turnaround DISCOs under the previous USAID Power Distribution Program (PDP), where considerable technical and commodity assistance was provided, detailed discussions were held with the senior management of both the DISCOs leading to the selection of the following short-listed tasks for immediate technical assistance:

- Preparation of plans for further upgrades and enhanced utilization of the Load Data Improvement Project (LDIP).
- Gap analysis and selected upgrading of installed AMR/AMI projects.
- Installation of Aerial Bundled Conductor (ABC) in congested areas.
- Specific gap analysis of ERP implementation.

1.2 AMI GAP ANALYSIS

Given an inefficient power distribution network infrastructure, lack of efficient system planning, high technical as well as commercial losses, financial mismanagement, and gaps in standards and specifications, DISCOs face an uphill task of meeting continuous regulator-mandated technical, operational and financial performance goals. In the absence of adequate technology updating, DISCOs are facing major issues in effectively operating and maintaining their respective distribution networks. Mainly the commercial operations of meter-to-cash are flawed by using outdated procedures and processes where they mainly rely on legacy metering technologies which result in the loss of cash collection owing to deficient billing systems.

One of the most significant interventions that was ever implemented in Pakistan was the commissioning of Advanced Metering Infrastructure (AMI). USAID under its Power Distribution Program (PDP) implemented the AMI System for PESCO, MEPCO and HESCO employing smart AMR

meters, hardware / software for Meter Data Collection (MDC), end-user interface and commissioning of an integrated Advanced Metering Infrastructure (AMI). The objective of this initiative was to introduce a fully automated metering system which would help the distribution sector in reducing losses, enhance load management controls, provide access to real-time customer load profiles, improve billing quality and revenue resulting in reduced billing complaints, increase in operational efficiency, reduced operating costs and optimize the DISCOS metering and billing operations. The AMI head-end systems were installed at PITC Data Centre / NOC at WAPDA House Lahore and DISCOs field users were provided with remote access through secure private physical and logical VPN links. They were provided with access to the meter data repository through web-based applications to use the meter readings for billing and analyses purpose whereas sub-divisions, divisions and revenue offices were connected over the IP cloud to perform day-to-day operational and other commercial activities. DISCOs acknowledged that the AMI rollout by USAID's PDP has resulted in tremendous savings on account of energy sale and improvement in the revenue collection in addition to other benefits such as technical energy loss reduction and improved system reliability. It also helped them with improved customer service, considerable reduction in billing complaints, increased operational efficiency, demand side load management, reduced operating costs and fully automated electricity metering and billing operation.

The biggest disadvantage with the existing AMI system is that it does not support integration of smart meters from multiple vendors and has created monopoly in favor of a single vendor. Given the benefits accrued over the years, DISCOs desired to expand AMR meter installations but could not accomplish it owing to difficulties in solicitations due to the vendor lock situation. Another major obstacle in expanding AMI activities is the absence of standards and specifications to promote standardized multivendor compliant AMI platform. National Transmission and Despatch Company (NTDC)'s design department who is custodian of the technical specifications, PITC which is responsible for extending IT services to DISCOs, AMR meter manufacturers and AMI integrators all should have thought through a workable solution to enable DISCOs expand their smart meter deployment but unfortunately, no concrete efforts were made to address the problem.

2. OPEN ARCHITECTURE-BASED AMI LANDSCAPE

Automating the metering process is a proven most efficient and cost-effective way to streamline distribution operations which combines interval data measurements with continuously available remote communication. It ensures timely billing, increased billing accuracy, flexible billing cycles, easier energy theft detection and creates customer energy profiles for targeting energy efficiency/demand response programs. With the rising demand, operational and environmental constraints, and aging infrastructures, nowadays utilities around the world are looking for more than just bill creation using the data. They use this information to improvise better and efficient customer services, improved utility asset management, timely outage management as well as effective system planning.

To mitigate the difficulty in integrating smart meters of various manufacturers on the existing landscape, SEP in collaboration with DISCOs, PITC, NTDC's S&S and Smart Meter Manufacturers initiated the design and development of an open architecture solution for establishing an integrated AMI platform offering equal opportunity to all vendors including international manufacturers. This document contains a detailed design of an MDM-based Multivendor Compliant AMI platform employing integrated middleware layer and standardized Universal Data Integration Layer (UDIL) to aggregate the data received from various brands of smart meters through individual MDCs/HEs.

2.1 UNIVERSAL DATA INTEGRATION LAYER (UDIL)

Presently the MDC systems are provided by a specific meter manufacturers and support recording of brand specific meter profiles only. To perform billing operations, DISCOs are required to switch to AMR manufacturer specific interface and fetch the billing determinants, transform the metering data to acceptable format and compile it before submitting it to the billing engine which in turn increases the chances of data corruption and loss. In other words, MDCs of each of the vendors work in silos and the data does not converge on a single platform which from the operations view point is highly inefficient. This warrants a standard Universal Data Integration Layer (UDIL) content and data structure that each of the vendor can expose at the MDC/HEs for data aggregation.

The primary purpose of developing a standardized UDIL is to facilitate DISCOs to integrate data from multi-vendor AMR meters with a common data warehouse in a standard format to:

- Enable integration of the multiple brand MDCs in the smart metering enterprise landscape.
- Consolidate metering data received from multiple MDCs using a common platform.
- Provide one portal to end-users for administering the smart meters irrespective of brands and makes.
- Integrate the MDM system with the billing system through universal data integration layer, for providing billing determinants and to retrieve payment information.
- Facilitate data analytics over a uniform e-system.
- Enable end-users to slice/dice data and generate charts/reports as per their requirements.
- End current single vendor-lock situation and open up the AMI landscape for all AMR meter manufacturers / suppliers.
- Enable DISCOs to deploy smart meters in a cost-effective manner with increased market access.

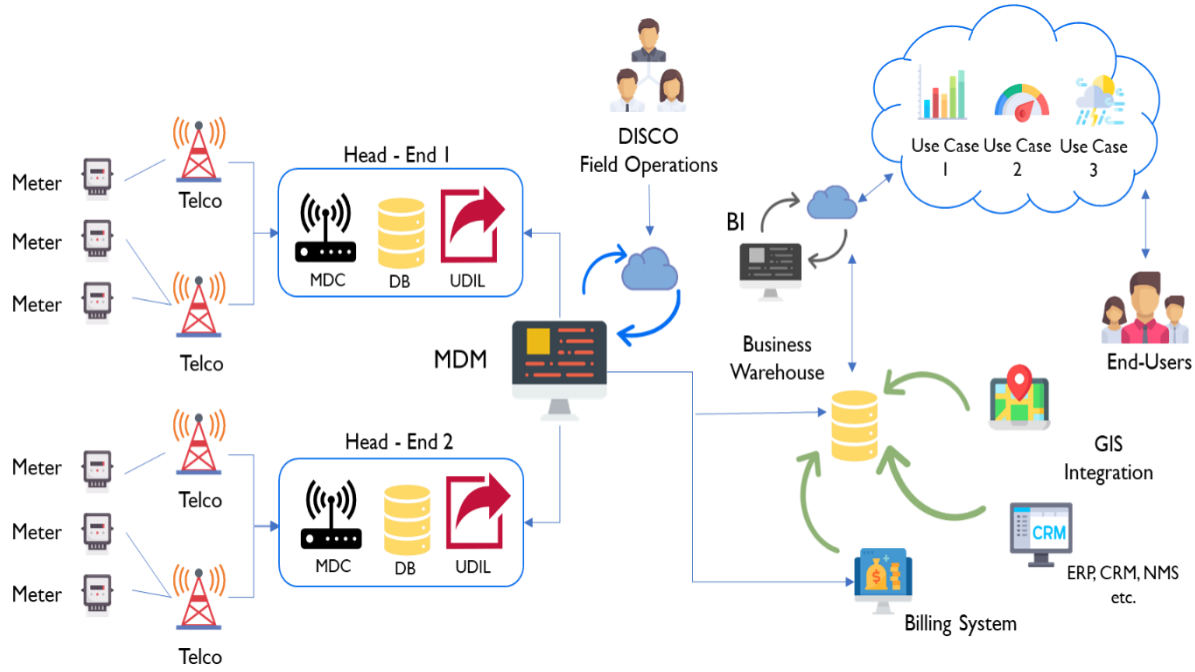
2.1.1 SYSTEM DESIGN ARCHITECTURE

To achieve multi-vendor compliance, a standardized universal data integration layer is required to aggregate the data streams from various MDCs on a common platform. The MDCs will collect

metering data directly from their respective AMR meters and push this data into the MDM systems by utilizing the Universal Data Integration layer (UDIL). Consequently, any MDM can easily bind all of the available MDCs metering data through UDIL and convert it into actionable information. From MDM, the billing determinants will flow to the billing system for consumer invoicing whereas the load data will aid the DISCOs to carry out prescriptive analysis and subsequent actions.

The UDIL will have the capacity to perform various rules on data packets received from multiple MDCs and facilitate its integration with the common data warehouse and billing systems. The central data warehouse will serve as the reporting server to MDM without any burden in collecting packets from multiple MDCs. The typical system architecture will be as follows:

AMI Enterprise Landscape



2.1.2 GENERAL SCOPE

SEP in consultation with all the stakeholders (PEPCO, NEPRA, PITC, DISCOs, AMR meter manufacturers, AMI vendors and NTDC's Standards & Specification Department) has designed and developed standardized specifications for Universal Data Integration Layer (UDIL) database structure, views, commands and controls. An MDM system will be installed in the enterprise landscape to consolidate meter data, process commands and services enabling the reading of all headend exposed views, as per standard UDIL specifications thus creating an integrated AMI landscape employing multi-vendor headend systems.

The salient system functionalities will be to:

- Communicate with the multiple MDCs on defined intervals/schedule for exchange of data packet including meter profiles, grid profiles, alerts, commands and to store the acquired data in the intermediate database.
- Provide one single interface to end-users for sending commands to meters via MDC and recording of the activity and response in the intermediate database.
- Provide dashboard to various stakeholders for access to instant information on key parameters and generate technical reports.

- Facilitate integration with the billing system, flat file-based or automatically feeding billing determinants for customer invoicing.
- Fetch transactional data specific to smart meter from billing system and store it in the intermediate database for energy accounting and vigilance.
- Introduce web-based customizable dashboard for users to convert data into structured information and develop appropriate analytics.

2.1.3 BUSINESS PROCESS SUPPORT

The MDM-based system will support the business and operational processes in line with the prevalent WAPDA Commercial Procedures (CP) such as:

- CP2: New Connection
- CP3: Temporary Disconnection
- CP4: Permanent Disconnection
- CP5: Reconnection
- CP6: Meter Reading
- CP7: Meter Change
- CP8: New Connection (Revenue Office)
- CP9: Meter Change (Billing Control Section)
- CP10: Billing
- CP13: Temporary Disconnection (Revenue Office)
- CP14: Permanent Disconnection (Revenue Office)
- CP15: Reconnection (Revenue Office)

2.1.4 FUNCTIONAL REQUIREMENTS

The functional requirements of the system are primarily aimed at providing an interface to DISCOs for managing the meter installation and operations through a universal data integration layer. This will connect with MDM system to perform operations through relevant meters MDCs. Therefore, the user will not be concerned on how the back-end communication takes place for multiple brand of meters and will be exposed to a single interface to execute routine functions.

The MDM system will generate billing determinants for export to the billing systems for invoicing functions. Transactional data from the billing systems will be mirrored into the MDM system so that users could compare technical information with the commercial codes.

Smart meter manufacturers / vendors will be required to expose key information from their respective MDCs on standard data structures as defined in this document by fetching data packets in a unified manner. The MDM will employ the functionality of reading multiple MDCs regardless of communication mediums such as web services, data tables, data views, flat file etc.

3. UDIL DATA STRUCTURE

To establish communication with MDM systems, MDCs of each of the meter manufacturer have to follow the standard data structures. This includes field name, field type and field description in transact SQL. The structures are categorized in this section below. paragraphs Meter vendors have liberty to make it available via services or by database tables/views. In case of web services, the field names will be classified as service parameters. As a thumb rule, length of varchar/string is 50 chars and decimal refer to 3 digits precision. Composite Key are represented with light blue background color and symbol.🔑

It is pertinent to mention here that no restriction has been imposed on the smart meter manufacturers to follow the exclusive data transfer options described in this document. Conventional methods such as CIM and Multi-speak are also encouraged but require packet structure as per specification laid down in this document. Further, all the write requests from MDM systems, through universal data integration layer, needs to be generated in asynchronous mode i.e. jobs executed in the backgrounds. The db_datetime field is used for indexing purpose (preferred binary indexing).

3.1 READ REQUESTS – TABULAR

The tabular requests are based on fetching information directly from database tables/views exposed by MDC for the MDM system via universal data integration layer.

3.1.1 INSTANTANEOUS DATA

These are electrical parameters recorded by MDC. The request is made by end-user on ad hoc basis and includes basic grid and meter profiles. The required list is as follow:

TABLE ID: ***INSTANTANEOUS_DATA***

Table Field	Data Type	Description
current_tariff_register	Int	Current Tariff Register
signal_strength	Decimal	Signal Strength
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
current_phase_a	Decimal	Current Phase A
current_phase_b	Decimal	Current Phase B
current_phase_c	Decimal	Current Phase C
voltage_phase_a	Decimal	Voltage Phase A
voltage_phase_b	Decimal	Voltage Phase B
voltage_phase_c	Decimal	Voltage Phase C
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
average_pf	Decimal	Average Power Factor
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

3.1.2 BILLING DATA

These registers are based on tariff types programmed in meter for recording of billing determinants at defined intervals. The packet of information includes energy, maximum demand for each of the register defined as per the tariff requirements.

TABLE ID: ***BILLING_DATA***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
active_energy_pos_t1	Decimal	T1 Active kWh (Import)
active_energy_pos_t2	Decimal	T2 Active kWh (Import)
active_energy_pos_t3	Decimal	T3 Active kWh (Import)
active_energy_pos_t4	Decimal	T4 Active kWh (Import)
active_energy_pos_tl	Decimal	Total Active kWh (Import)
active_energy_neg_t1	Decimal	T1 Active kWh (Export)
active_energy_neg_t2	Decimal	T2 Active kWh (Export)
active_energy_neg_t3	Decimal	T3 Active kWh (Export)
active_energy_neg_t4	Decimal	T4 Active kWh (Export)
active_energy_neg_tl	Decimal	Total Active kWh (Export)
reactive_energy_pos_t1	Decimal	T1 Reactive kVARh (Import)
reactive_energy_pos_t2	Decimal	T2 Reactive kVARh (Import)
reactive_energy_pos_t3	Decimal	T3 Reactive kVARh (Import)
reactive_energy_pos_t4	Decimal	T4 Reactive kVARh (Import)
reactive_energy_pos_tl	Decimal	Total Reactive kVARh (Import)
reactive_energy_neg_t1	Decimal	T1 Reactive kVARh (Export)
reactive_energy_neg_t2	Decimal	T2 Reactive kVARh (Export)
reactive_energy_neg_t3	Decimal	T3 Reactive kVARh (Export)
reactive_energy_neg_t4	Decimal	T4 Reactive kVARh (Export)
reactive_energy_neg_tl	Decimal	Total Reactive kVARh (Export)
active_mdi_pos_t1	Decimal	T1 Active MDI kW (Import)
active_mdi_pos_t2	Decimal	T2 Active MDI kW (Import)
active_mdi_pos_t3	Decimal	T3 Active MDI kW (Import)
active_mdi_pos_t4	Decimal	T4 Active MDI kW (Import)
active_mdi_pos_tl	Decimal	Total Active MDI kW (Import)
active_mdi_neg_t1	Decimal	T1 Active MDI kW (Export)

Table Field	Data Type	Description
active_mdi_neg_t2	Decimal	T2 Active MDI kW (Export)
active_mdi_neg_t3	Decimal	T3 Active MDI kW (Export)
active_mdi_neg_t4	Decimal	T4 Active MDI kW (Export)
active_mdi_neg_tl	Decimal	Total Active MDI kW (Export)
cumulative_mdi_pos_t1	Decimal	T1 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t2	Decimal	T2 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t3	Decimal	T3 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t4	Decimal	T4 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_tl	Decimal	Total Cumulative Active MDI kW (Import)
cumulative_mdi_neg_t1	Decimal	T1 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t2	Decimal	T2 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t3	Decimal	T3 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t4	Decimal	T4 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_tl	Decimal	Total Cumulative Active MDI kW (Export)
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

3.1.3 MONTHLY_BILLING_DATA

The information maintained in this table is a summarized view of cumulative information as maintained in previous table. An additional counter maintains as to how many times MDI register have been reset for audit purpose.

TABLE ID: ***MONTHLY_BILLING_DATA***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
active_energy_pos_t1	Decimal	T1 Active kWh (Import)
active_energy_pos_t2	Decimal	T2 Active kWh (Import)
active_energy_pos_t3	Decimal	T3 Active kWh (Import)
active_energy_pos_t4	Decimal	T4 Active kWh (Import)
active_energy_pos_tl	Decimal	Total Active kWh (Import)
active_energy_neg_t1	Decimal	T1 Active kWh (Export)
active_energy_neg_t2	Decimal	T2 Active kWh (Export)
active_energy_neg_t3	Decimal	T3 Active kWh (Export)
active_energy_neg_t4	Decimal	T4 Active kWh (Export)

Table Field	Data Type	Description
active_energy_neg_tl	Decimal	Total Active kWh (Export)
reactive_energy_pos_t1	Decimal	T1 Reactive kVARh (Import)
reactive_energy_pos_t2	Decimal	T2 Reactive kVARh (Import)
reactive_energy_pos_t3	Decimal	T3 Reactive kVARh (Import)
reactive_energy_pos_t4	Decimal	T4 Reactive kVARh (Import)
reactive_energy_pos_tl	Decimal	Total Reactive kVARh (Import)
reactive_energy_neg_t1	Decimal	T1 Reactive kVARh (Export)
reactive_energy_neg_t2	Decimal	T2 Reactive kVARh (Export)
reactive_energy_neg_t3	Decimal	T3 Reactive kVARh (Export)
reactive_energy_neg_t4	Decimal	T4 Reactive kVARh (Export)
reactive_energy_neg_tl	Decimal	Total Reactive kVARh (Export)
active_mdi_pos_t1	Decimal	T1 Active MDI kW (Import)
active_mdi_pos_t2	Decimal	T2 Active MDI kW (Import)
active_mdi_pos_t3	Decimal	T3 Active MDI kW (Import)
active_mdi_pos_t4	Decimal	T4 Active MDI kW (Import)
active_mdi_pos_tl	Decimal	Total Active MDI kW (Import)
active_mdi_neg_t1	Decimal	T1 Active MDI kW (Export)
active_mdi_neg_t2	Decimal	T2 Active MDI kW (Export)
active_mdi_neg_t3	Decimal	T3 Active MDI kW (Export)
active_mdi_neg_t4	Decimal	T4 Active MDI kW (Export)
active_mdi_neg_tl	Decimal	Total Active MDI kW (Export)
cumulative_mdi_pos_t1	Decimal	T1 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t2	Decimal	T2 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t3	Decimal	T3 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t4	Decimal	T4 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_tl	Decimal	Total Cumulative Active MDI kW (Import)
cumulative_mdi_neg_t1	Decimal	T1 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t2	Decimal	T2 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t3	Decimal	T3 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t4	Decimal	T4 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_tl	Decimal	Total Cumulative Active MDI kW (Export)
mdi_reset_datetime	Datetime	MDI Reset Date & Time
reset_count	Int	MDI Reset Count Number
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

3.1.4 LOAD PROFILE DATA

The load data records the complete meter and grid profiles at pre-defined intervals.

TABLE ID: ***LOAD_PROFILE_DATA***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
channel_id	Int	Load Profile Channel ID where applicable
interval	Int	Interval of Profile
active_energy_t1	Decimal	T1 Active kWh
active_energy_t2	Decimal	T2 Active kWh
active_energy_t3	Decimal	T3 Active kWh
active_energy_t4	Decimal	T4 Active kWh
active_energy_tl	Decimal	Total Active kWh
reactive_energy_t1	Decimal	T1 Reactive kVARh
reactive_energy_t2	Decimal	T2 Reactive kVARh
reactive_energy_t3	Decimal	T3 Reactive kVARh
reactive_energy_t4	Decimal	T4 Reactive kVARh
reactive_energy_tl	Decimal	Total Reactive kVARh
active_mdi_t1	Decimal	T1 Active MDI
active_mdi_t2	Decimal	T2 Active MDI
active_mdi_t3	Decimal	T3 Active MDI
active_mdi_t4	Decimal	T4 Active MDI
active_mdi_tl	Decimal	Total Active MDI
cumulative_mdi_t1	Decimal	T1 Cumulative Active MDI
cumulative_mdi_t2	Decimal	T2 Cumulative Active MDI
cumulative_mdi_t3	Decimal	T3 Cumulative Active MDI
cumulative_mdi_t4	Decimal	T4 Cumulative Active MDI
cumulative_mdi_tl	Decimal	Total Cumulative Active MDI
current_phase_a	Decimal	Current Phase A
current_phase_b	Decimal	Current Phase B
current_phase_c	Decimal	Current Phase C
voltage_phase_a	Decimal	Voltage Phase A
voltage_phase_b	Decimal	Voltage Phase B
voltage_phase_c	Decimal	Voltage Phase C
active_pwr_pos_phase_a	Decimal	Active Power Import Phase A
active_pwr_pos_phase_b	Decimal	Active Power Import Phase B
active_pwr_pos_phase_c	Decimal	Active Power Import Phase C

Table Field	Data Type	Description
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
active_pwr_neg_phase_a	Decimal	Active Power Export Phase A
active_pwr_neg_phase_b	Decimal	Active Power Export Phase B
active_pwr_neg_phase_c	Decimal	Active Power Export Phase C
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
reactive_pwr_pos_phase_a	Decimal	Reactive Power Import Phase A
reactive_pwr_pos_phase_b	Decimal	Reactive Power Import Phase B
reactive_pwr_pos_phase_c	Decimal	Reactive Power Import Phase C
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
reactive_pwr_neg_phase_a	Decimal	Reactive Power Export Phase A
reactive_pwr_neg_phase_b	Decimal	Reactive Power Export Phase B
reactive_pwr_neg_phase_c	Decimal	Reactive Power Export Phase C
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
average_pf	Decimal	Average Power Factor
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

3.1.5 EVENTS

Events data structure is for recording of the alerts and alarms generated by the meter.

TABLE ID: ***EVENTS***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑event_datetime	Datetime	Meter Date & Time when event occurred
🔑event_code	Int	Event Code
event_description	Varchar	Event Name/Description
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

Basic event codes, as currently utilized, are as follow:

101 : MDI reset
102 : Parameterization
111 : Power fail start
112 : Power fail end
113 : Phase failure
114 : Over Voltage
115 : Under Voltage
116 : Demand over load

117 : Reverse Energy (active energy)
118 : Reverse Polarity
121 : CT bypass

3.1.6 METER VISUALS

The visual record presents the latest information about the meter. This information gets replaced with the next latest update received from the meter.

TABLE ID: ***METER_VISUALS***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
last_command	Varchar	Last Command Sent to Meter
last_command_datetime	Datetime	Date & Time of Last Command
last_command_resp	Varchar	Response of Last Command
last_command_resp_datetime	Datetime	Response Date & Time of Last Command
last_active_energy_pos_tl	Decimal	Last Value of Positive Absolute kWh
last_active_energy_pos_tl_datetime	Datetime	Latest Reporting Date & Time for kWh
last_active_energy_neg_tl	Decimal	Last Value of Negative Absolute kWh
last_active_energy_neg_tl_datetime	Datetime	Latest Reporting Date & Time for kWh
last_reactive_energy_pos_tl	Decimal	Last Value of Positive Absolute kVARh
last_reactive_energy_pos_tl_datetime	Datetime	Latest Reporting Date & Time for kVARh
last_reactive_energy_neg_tl	Decimal	Last Value of Negative Absolute kVARh
last_reactive_energy_neg_tl_datetime	Datetime	Latest Reporting Date & Time for kVARh
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
aggregate_active_pwr_pos_datetime	Datetime	Latest Reporting Date & Time for positive kW
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
aggregate_active_pwr_neg_datetime	Datetime	Latest Reporting Date & Time for negative kW
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
aggregate_reactive_pwr_pos_datetime	Datetime	Latest Reporting Date & Time for positive kVAR
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
aggregate_reactive_pwr_neg_datetime	Datetime	Latest Reporting Date & Time for negative kVAR
last_contactor_on_datetime	Datetime	Last Contactor 'On Date & Time'
last_contactor_off_datetime	Datetime	Last Contactor 'Off Date & Time'
last_communication_datetime	Datetime	Last Communication Date & Time
last_signal_strength	Decimal	Latest Signal Strength

Table Field	Data Type	Description
meter_load_limit_set	Decimal	Latest Sanctioned Load Limit set in the meter
meter_ls_schedule_startdatetime	Datetime	Start Date time set for Load Shedding
meter_ls_schedule_enddatetime	Datetime	End Date time set for Load Shedding
meter_ls_schedule_slabs	Varchar	Array containing action time & relay operate
meter_last_comm_datetime	Datetime	Meter Date & Time of last communication
meter_tariff_identification	Varchar	Array containing tariff identification & time slabs
meter_activation_status	Int	1 for 'Active', 0 for 'Inactive'

3.2 READ REQUESTS – AUTHORIZATION PROTOCOL

The Application Programming Interface (API) requests, as part of the universal data integration layer are based on fetching information directly from the database via exposed services of MDC for the MDM system.

For secure communication between the MDCs and the MDM system, through universal data integration layer, there is a requirement of passing a unique key. The key shall be time bound for 30 minutes minimum and the following function will be passed on to the MDC for getting the unique identification key.

3.2.1 AUTHORIZATION SERVICE

SERVICE NAME: ***AUTHORIZATION_SERVICE***

Input Parameters

Function	Argument	Description
Param	username	Username for Web Service
Param	password	Password for Web Service
Param	code	Special Code for Web Service

Return Values

Param	status	0 for Failed, 1 for Success
Param	privatekey	Encrypted Value for using APIs
Param	message	Optional Message

3.3 READ REQUESTS – API BASED

Should the meter vendor opt for API-based information exchange process, the below functions and its parameters shall be passed on to the MDC by the MDM system through universal data integration layer. The name of the service is mentioned in the section and the output of the functions will follow same variables and data types as fields specified in the section (see READ REQUEST – TABULAR BASED Section 3.1)

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	start_datetime	Data Start Time
Param	end_datetime	Data End Time
Param	privatekey	Encrypted Key

Return Values

Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing respective data items as per request

3.4 WRITE REQUESTS – COMMANDS

There are several commands that may be required to pass on to meters for performing certain functions based on need basis. MDM being the central authority should pass on these commands to various MDC on standard format through universal data integration layer. These commands and their parameters are defined below. MDCs are required to interpret these commands correctly and subsequently perform and acknowledge back to the MDM system through universal data integration layer.

3.4.1 AUX RELAY OPERATIONS

The objective is to remotely disconnect/reconnect a meter.

SERVICE NAME: ***AUX_RELAY_OPERATIONS***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	relay_operate	0 for Disconnect, 1 for Connect
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing command output against each meter

3.4.2 TIME SYNCHRONIZATION

The objective is to synchronize meter time with the MDC time.

SERVICE NAME: ***TIME_SYNCHRONIZATION***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.3 SANCTIONED LOAD CONTROL

The load limiting service is to be used to disconnect the supply (with warnings) after exceeding the sanctioned load range.

SERVICE NAME: ***SANCTIONED_LOAD_CONTROL***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	load_limit	Threshold Limit for kW
Param	maximum_retries	Maximum Retries in number
Param	retry_interval,	Time Interval for retry in seconds
Param	threshold_duration	Duration to accept threshold crossing limit in seconds
Param	retry_clear_interval	Time after retries count is cleared in seconds
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.4 LOAD SHEDDING SCHEDULING

The load management service is to be used to auto disconnect/reconnect the supply on defined intervals as set by the DISCO.

SERVICE NAME: ***LOAD_SHEDDING_SCHEDULING***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	start_datetime	Date Time for starting schedule
Param	end_datetime	Date Time till schedule lasts
Param(set)	load_shedding_slabs	Array Containing action time & relay operate values for multiple time slabs with relay status. Relay Operate value will be 0 for Disconnect & 1 for Connect
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.5 TIME OF USE CHANGE

Service to be used for remotely conducting possible changes of time slots for peak and off-peak readings.

SERVICE NAME: ***TIME_OF_USE_CHANGE***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	season	Season month range
Param	days	Number of days for tariff
Param	time_slab	Starting and Ending Time Slabs Range
Param	tariff_identification	Tariff Number
Param	activation_datetime	Date Time for the activation of condition
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.6 DEVICE CREATION

Service to be used for first time creation of a device in the MDC system.

SERVICE NAME: ***DEVICE_CREATION***

Input Parameters

Function	Argument	Description
Param(set)	device_identity	Associative array containing "dsn" as Device Serial Number(s) / Meter Serial Number(s) along with "global_device_id" as unique identifier of device

Function	Argument	Description
Param(set)	device_type	Device Type 1 for meter 2 for DCU 3 for other
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	communication_mode	1 for GPRS/3G/4G 2 for RF 3 for PLC 4 for Ethernet 5 for other
Param	communication_type	'0' for always on '1' for periodic-on '2' for on-demand
Param	communication_interval	Communication Interval in Minutes if Communication Type value is SET for '2'
Param	sim_number	SIM Number
Param	sim_id	SIM ID
Param	mdi_reset_date	Date of Every Month Ranges from 1 to 28
Param	mdi_reset_time	Time at which MDI Reset Occurs
Param	phase	1 for Single 3 for Three-phase
Param	meter_type	1 for Normal 2 for Whole Current 3 for CTO 4 for CTPT
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.7 UPDATE IP PORT

Service to be used for changing the IP Address and port of meter.

SERVICE NAME: ***UPDATE_IP_PORT***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	primary_ip_address	New Primary IP Address
Param	secondary_ip_address	New Secondary IP Address
Param	primary_port	New Primary Port Address
Param	secondary_port	New Secondary Port Address
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.8 METER DATA SAMPLING

Service to be used for configuring sampling interval of data stored in meter.

SERVICE NAME: ***METER_DATA_SAMPLING***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	activation_datetime	Date Time for the activation of condition

Function	Argument	Description
Param	data_type	Valid data type values will be 'INST' for Instantaneous Data 'BILL' for Billing Profiles 'LPRO' for Load Profile
Param	sampling_interval	Sampling Interval in Minutes ranges from 1 – 59
Param	sampling_initial_time	Start Time Slot ranges from 0 to 59 e.g. If Sampling interval is set to 15 and Sampling Initial Time is 0 then data will be stored in meter at following time interval 00:00, 15:00, 30:00, 45:00 of every hour on daily basis
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.9 ACTIVATE METER OPTICAL PORT

This service is used to activate the Optical Port of the meter for the authorized user for a defined time limit. MDC will send activate/deactivate command to the meter to let the user access the optical port.

SERVICE NAME: ***ACTIVATE_METER_OPTICAL_PORT***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key
Param	optical_port_on_datetime	Date & Time to enable Optical Port
Param	optical_port_off_datetime	Date & Time to disable Optical Port

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.10 WAKE UP SIM NUMBER UPDATE

Service to be used for changing SIM numbers from which wakeup SMS or voice call is allowed.

SERVICE NAME: ***WAKE_UP_SIM_NUMBER_UPDATE***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	wakeup_number_1	SIM NUMBER1 for wakeup
Param	wakeup_number_2	SIM NUMBER2 for wakeup
Param	wakeup_number_3	SIM NUMBER3 for wakeup
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.4.11 METER STATUS UPDATE

This service is used to change the status of meter to active or inactive. MDC will only entertain/read meter having meter activation status equal to 1. All meters with meter activation status equal to 0 will be considered stopped due to any work flow operation.

SERVICE NAME: ***METER_STATUS_UPDATE***

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key
Param	meter_activation_status	1 for Active, 0 for Inactive Meters

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global device id, msn & indv_status for representing individual status of command against each meter

3.5 ON-DEMAND REQUESTS – API BASED

The read requests, as defined above are based on fetching information from the MDC database at defined intervals. However, there can be a need to request the MDC to bring latest information from the meter on ad hoc basis. Note that the meters are not essentially kept on 'Always Alive' mode. Hence to achieve the requirement, MDM must be able to pass the information to MDC through the universal data integration layer to fetch scenario-specific information directly from the meter as defined in section 3.1. For all such data retrieval requests made to the meter via the MDC, the input parameter is defined below. The parameter is supplemented with a unique key named 'transactionid' generated by the universal data integration layer for reference purpose i.e. all the requests sent to multiple MDCs will be having a unique key assigned for differentiation. Apart from unique key, there will be a unique variable 'Type' (length 4 characters) passed through for identification of the service request. The unique key will be additionally supplemented with defined set of parameters designed to meet end-user criteria. These 'Types' and their parameters are defined below.

3.5.1 TRANSACTION TYPES

Following list of values are classified as valid values:

- Type 'INST' – Instantaneous Data
- Type 'BILL' – Billing Profiles
- Type 'MBIL' – Monthly Billing Profile
- Type 'LPRO' – Load Profile
- Type 'EVNT' – Events
- Type 'VISL' – Visuals

Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Unique key for each request
Param	type	Request Type
Param	start_datetime	Start Date & Time of Data to be fetched from meter
Param	end_datetime	End Date & Time of Data to be fetched from meter

Note:

For Type 'INST' & 'VISL' start_datetime & end_datetime will be blank. Response/Returned parameters will be same as in case of Read Requests Section 3.1

3.5.2 TRANSACTION STATUS

Each command is required to be maintained in the database (either view/table) of MDC with the latest stage indicator 'msn' wise. This will help the MDM to get to know the status of command it has sent to the meter through the universal data integration layer and for all msn separated in rows. For standard, the command status numbers (status_level) are as follow:

- 0 – refers to waiting for processing.
- 1 – refers to commencing command processing.
- 2 – refers to communication request sent to meter.
- 3 – refers to communication established with meter.
- 4 – refers to command sent to meter.
- 5 – refers to command executed by meter.

MDC is required to make this command stored in a table for monitoring purposes and making MDM to contact back through the universal data integration layer for response. The table format is defined below and additionally serves as the storage node for output received from meter. It is pertinent to mention that this table/view should reflect each msn separately in rows i.e. in case of a transaction comprising an update for a set of meters, the rows are to be populated separately for each msn having the same transactionid repeated. This will give the visibility for getting to know the individual status of each of the msn.

TRANSACTION NAME: ***TRANSACTION_STATUS***

Table Field	Data Type	Description
transactionid	Varchar(Max)	Transaction ID of command from MDM
msn	Text	Meter Serial Number
global_device_id	Varchar(Max)	Unique ID for each device by MDM
type	Varchar(Max)	Type of Request
type_parameters	Varchar(Max)	Type Parameters sent specific to Request
command_receiving_datetime	DateTime	Date & Time of command received by MDC
status_level	Int	Status level range from 0 – 5. Latest value of status will be stored here. MDC will continuously update this column according to status.
status_1_datetime	DateTime	Date & Time when command processing started by MDC
status_2_datetime	DateTime	Date & Time when request was forwarded to meter from MDC
status_3_datetime	DateTime	Date & Time when meter gets connected with MDC (Hand Shake)
status_4_datetime	DateTime	Date & Time when request forwarded to meter from MDC
status_5_datetime	DateTime	Date & Time when meter successfully executed command
indv_status	Int	'0' or otherwise '1'
request_cancelled	Int	'0' or otherwise '1' if MDC cancels the request
request_cancel_reason	Varchar	Reason for cancelation of request
request_cancel_datetime	DateTime	Date & time of cancelation of request

In addition, there can be alternative generic command status read service required to enquire the output of on demand and write request. The service will have following input parameters but returned values will be same as in the above defined table in JSON/XML format.

Input Parameters

Function	Argument	Description
Param(set)	transactionid	Transaction ID(s) from MDM
Param	privatekey	Encrypted Key

3.5.3 TRANSACTION CANCEL

The purpose of this service is to cancel the command against specific devices. MDC will cancel commands against provided devices (identified by global_device_id) if and only if the command status numbers (status_level) of these devices is 0, 1 or 2 (Section 3.5.2).

TRANSACTION NAME*:* ***TRANSACTION_CANCEL***

Not Input Parameters

Function	Argument	Description
Param(set)	global_device_id	Unique ID for each device by MDM
Param	transactionid	Transaction ID from MDM
Param	request_datetime	Date & Time at which request is made
Param	privatekey	Encrypted Key

Return Values

Param	transactionid	Transaction ID from MDM
Param	status	0 for Failed, 1 for Success
Param	message	Optional Message
Param(set)	data	JSON/XML array Containing list of global_device_id, msn & indv_status for representing individual status of command against each meter

Note:

- All API/services will follow HTTPS protocol for secure and encrypted sharing of information.
- All API/services will follow standard HTTP codes for each response e.g.:
  - In case of valid request and response, service will send status =1 along with HTTP Status Code of 200.
  - Similarly, in case of Un-Authorize call, service will send status = 0 along with HTTP Status Code of 401 Unauthorized.
  - Similarly, in case of In-Valid request format, service will send Status = 0 along with HTTP Status Code of 400 Bad Request.
- All Input and Output Parameters are of string type or array.
- All Datetime fields in Tables/Views/APIs should follow "yyyy-MM-dd HH:mm:ss" format.

4. IMPLEMENTATION EXAMPLES

Below are the implementation examples for all services.

4.1 AUTHORIZATION SERVICE

Request:

Method: POST

URL: https://host:port/authorization_service

Headers:

Header Name		Header Value
username	|	"mdm_user"
password	|	"WSnt!@SS^12*"
code	|	"223"

Response

JSON:

```
{
    "status" : "1" ,
    "privatekey": "7h6g5d4chyhdxg875576v5f5gv8b7bv",
    "message": "You are authenticated successfully. Private key will be valid for 30 Minutes"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
<privatekey>7h6g5d4chyhdxg875576v5f5gv8b7bv</privatekey>
<message>You are authenticated successfully. Private key will be valid for 30 Minute</message>
</root>
```

4.2 AUX_RELAY_OPERATIONS

Request:

Method : POST

URL : https://host:port/aux_relay_operations

Headers:

Header Name	Header Value
transactionid	9Xcvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m96541254","m97524158","m98562412"]
request_datetime	"2018-10-15 13:00:00"
relay_operate	"0"

Response:

JSON:

```
{
    "status": "1" ,
    "transactionid": "9Xcvfg675d5hg85fhjjfved47f65d3s3g",
    "data": [
        {"global_device_id":"m96541254" ,msn="4096541254", "indv_status":"1"},
        {"global_device_id":"m97524158" , msn="4097524158", "indv_status":"0"},
        {"global_device_id":"m98562412" , msn="4098562412", "indv_status":"1"}],
    "message": "Command for global_device_id:40m97524158 Cannot be entertained as corresponding meter has no relay connected to it. MDC will turn off all meters with status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <status>1</status>
  <transactionid>9Xcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
```

<data>
<element>
<global_device_id>m96541254</global_device_id>
<msn>4096541254</msn>
<indv_status>1</indv_status>
</element>
<element>
<global_device_id>m97524158</global_device_id>
<msn>4097524158</msn>
<indv_status>0</indv_status>
</element>
<element>
<global_device_id>m98562412</global_device_id>
<msn>4098562412</msn>
<indv_status>1</indv_status>
</element>
</data>
<message>Command for global_device_id:m97524158 Cannot be entertained as corresponding meter has no relay connected to it. MDC will turn off all meters with indv_status 1</message>
</root>

4.3 TIME_SYNCHRONIZATION

Request:

Method: POST

URL: https://host:port/time_synchronization

Headers:

Header Name	Header Value
transactionid	7Ucvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"

Response

JSON:

```
{
        "status": "1" ,
        "transactionid" : "7Ucvfg675d5hg85fhjjfved47f65d3s3g",
        "data": [
                {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
                {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
        "message": "Time will be updated for requested meters according to system time"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <status>1</status>
  <transactionid>7Ucvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
  <data>
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
 <message>Time will be updated for requested meters according to system time</message>
</root>
```

4.4 SANCTIONED_LOAD_CONTROL

Request:

Method: POST

URL: https://host:port/sanctioned_load_control

Headers:

Header Name	Header Value
transactionid	8Ncvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
load_limit	"20"
maximum_retries	"10"
retry_interval	"60"
threshold_duration	"30"
retry_clear_interval	"3600"

Response:

JSON:

```
{
    "status": "1" ,
    "transactionid": "8Ncvfg675d5hg85fhjjfved47f65d3s3g",
    "data": [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" ,msn="4097524158",  "indv_status":"1"}
],
    "message": "Limits for provided meters will be changed"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

<status>1</status>
<transactionid>8Ncvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
<data>
<element>
<global_device_id>m98562412</global_device_id>
<msn>4098562412</msn>
<indv_status>1</indv_status>
</element>
<element>
<global_device_id>m97524158</global_device_id>
<msn>4097524158</msn>
<indv_status>1</indv_status>
</element>
</data>
<message>Limits for provided meters will be changed</message>
</root>

4.5 LOAD_SHEDDING_SCHEDULING

Request:

Method: POST

URL: https://host:port/load_shedding_scheduling

Headers:

Header Name	Header Value
transactionid	R5cvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
start_datetime	"2018-11-01 00:00:00"
end_datetime	"2018-12-01 00:00:00"
load_shedding_slabs	[{"action_time":"03:00:00", "relay_operate":"0"}, {"action_time":"06:00:00", "relay_operate":"1"}, {"action_time":"09:00:00", "relay_operate":"0"},

{“action_time”:”12:00:00”, “relay_operate”:”1”},
{“action_time”:”16:00:00”, “relay_operate”:”0”}]

Response:

JSON:

```
{
    “status”: “1” ,
    “transactionid”: “R5cvfg675d5hg85fhjjfved47f65d3s3g”,
    “data”: [
        {“global_device_id” : “m98562412” , msn=“4098562412”, “indv_status”:”1”}
    ,
        {“global_device_id” : “m97524158” , msn=“4097524158”, “indv_status”:”1”}
],
    “message”: “Load Shedding Schedule will be programmed in meters with indv_status
1”
}
```

XML:

```
<?xml version=“1.0” encoding=“UTF-8”?>
<root>
  <status>1</status>
  <transactionid>R5cvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
  <data>
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>Load Shedding Schedule will be programmed in meters with indv_status 1</message>
</root>
```

4.6 TIME_OF_USE_CHANGE

Request:

Method: POST

URL: https://host:port/time_of_use_change

Headers:

Header Name	Header Value
transactionid	3Wcvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
activation_datetime	"2018-11-01 00:00:00"
season	"1"
days	"1"
time_slab	["08:00","17:00"]
tariff_identification	"1"

Response:

JSON:

```
{
    "status": "1" ,
    "transactionid": "3Wcvfg675d5hg85fhjjfved47f65d3s3g",
    "data": [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
    "message": "Tariff will be applied to all meters with indv_status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
<transactionid>3Wcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
  <data>
```

```
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>Tariff will be applied to all meters with indv_status 1</message>
</root>
```

4.7 DEVICE_CREATION

Request:

Method: POST

URL: https://host:port/device_creation

Headers:

Header Name	Header Value
transactionid	ATTvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
device_identity	[{"dsn":"4098562412","global_device_id":"m98562412"}, {"dsn":"4097524158","global_device_id":"m97524158"}]
request_datetime	""2018-10-15 13:00:00""
communication_interval	"15"
device_type	"1"
mdi_reset_date	"1"
mdi_reset_time	"00:00:00"
sim_number	"03218090100"
sim_id	"11199oijjh78"
phase	"3"

meter_type | “2”

communication_mode | “1”

Response:

JSON:

```
{
    “status” : 1 ,
    “transactionid” : “ATTvfg675d5hg85fhjjfved47f5d3aRi”,
    “data” : [
        {“global_device_id” : “m98562412” , msn=“4098562412”, “indv_status”:”1”}
,
        {“global_device_id” : “m97524158” , msn=“4097524158”, “indv_status”:”1”}
],
    “message” : “Meters with indv_status 1 are created Successfully”
}
```

XML:

```
<?xml version=“1.0” encoding=“UTF-8”?>
<root>
  <status>1</status>
  <transactionid>ATTvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
   <data>
     <element>
       <global_device_id>m98562412</global_device_id>
         <msn>4098562412</msn>
       <indv_status>1</indv_status>
     </element>
     <element>
       <global_device_id>m97524158</global_device_id>
         <msn>4097524158</msn>
       <indv_status>1</indv_status>
     </element>
   </data>
   <message>Meters with indv_status 1 are created Successfully</message>
</root>
```

4.8 UPDATE_IP_PORT

Request:

Method: POST

URL: https://host:port/update_ip_port

Headers:

Header Name	Header Value
transactionid	0Ycvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
primary_ip_address	"192.168.100.120"
secondary_ip_address	"192.168.100.121"
primary_port	"8080"
secondary_port	"8080"

Response:

JSON:

```
{
    "status": "1" ,
    "transactionid": "0Ycvfg675d5hg85fhjjfved47f65d3s3g",
    "data": [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" ,msn="4097524158",  "indv_status":"1"}
],
    "message": "IP Addresses will be updated in all meters will indv_status 1"
}
```

XML:

```
----------
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <status>1</status>
  <transactionid>0Ycvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
  <data>
```

```
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>IP Addresses will be updated in all meters will indv_status 1</message>
</root>
```

4.9 METER_DATA_SAMPLING

Request:

Method: POST

URL: https://host:port/meter_data_sampling

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	[“m98562412”,”m97524158”]
request_datetime	“2018-10-15 13:00:00”
activation_datetime	“2018-11-01 00:00:00”
data_type	“BILL”
sampling_interval	“30”
sampling_initial_time	“00”

Response:

JSON:

```
{
    “status” : 1 ,
    “transactionid” : “ ARbvfg675d5hg85fhjjfved47f5d3aRi”,
```

```
"data" : [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
"message" : "Sampling interval in meter with indv_status 1 will be changed accordingly"
    }
```


XML:


```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
  <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
   <data>
     <element>
       <global_device_id>m98562412</global_device_id>
         <msn>4098562412</msn>
       <indv_status>1</indv_status>
     </element>
     <element>
       <global_device_id>m97524158</global_device_id>
         <msn>4097524158</msn>
       <indv_status>1</indv_status>
     </element>
   </data>
   <message>Sampling interval in meter with indv_status 1 will be changed accordingly</message>
</root>
```


4.10 ACTIVATE_METER_OPTICAL_PORT

Request:

Method: POST

URL: https://host:port/activate_meter_optical_port

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
optical_port_on_datetime	"2018-10-15 13:00:00"
optical_port_off_datetime	"2018-10-15 13:00:00"

Response:

JSON:

```
{
    "status" : 1 ,
    "transactionid" : " ARbvfg675d5hg85fhjjfved47f5d3aRi",
    "data" : [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
    "message" : "Optical Port has been activated for indv_status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
  <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
   <data>
     <element>
       <global_device_id>m98562412</global_device_id>
         <msn>4098562412</msn>
       <indv_status>1</indv_status>
     </element>
     <element>
       <global_device_id>m97524158</global_device_id>
         <msn>4097524158</msn>
       <indv_status>1</indv_status>
     </element>
   </data>
   <message>Optical Port has been activated for indv_status 1</message>
</root>
```

4.11 WAKE_UP_SIM_NUMBER_UPDATE

Request:

Method: POST

URL: https://host:port/wake_up_sim_number_update

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
wakeup_number_1	"01234567890"
wakeup_number_2	"01234567891"
wakeup_number_3	"01234567892"

Response:

JSON:

```
{
    "status" : 1 ,
    "transactionid" : " ARbvfg675d5hg85fhjjfved47f5d3aRi",
    "data" : [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
    "message" : "Wakeup numbers have been updated for indv_status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
  <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
   <data>
     <element>
       <global_device_id>m98562412</global_device_id>
```

```
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>Wakeup numbers have been updated for indv_status 1</message>
</root>
```

4.12 METER_STATUS_UPDATE

Request:

Method: POST

URL: https://host:port/meter_status_update

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
meter_activation_status	"1"

Response:

JSON:

```
{
    "status" : 1 ,
    "transactionid" : " ARbvfg675d5hg85fhjjfved47f5d3aRi",
    "data" : [
        {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
        {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
    "message" : "Meter Status have been updated for indv_status 1"
```

}

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
 <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
  <data>
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>Meter Status have been updated for indv_status 1</message>
</root>
```

4.13 ON-DEMAND REQUESTS

Request:

Method: POST

URL: https://host:port/on_demand_request

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value
global_device_id	["m98562412","m97524158"]
start_datetime	"2018-10-15 13:00:00"
end_datetime	"2018-10-22 13:00:00"

data_type	|	"BILL"
type	|	"EVNT"

Response:

JSON:

```
{
        "status" : 1 ,
        "transactionid" : " ARbvfg675d5hg85fhjjfved47f5d3aRi",
        "data" : [
                {"global_device_id" : "m98562412" , msn="4098562412", "indv_status":"1"}
,
                {"global_device_id" : "m97524158" , msn="4097524158", "indv_status":"1"}
],
        "message" : "MDM will send all data  of Meter with indv_status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<status>1</status>
 <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
  <data>
    <element>
      <global_device_id>m98562412</global_device_id>
        <msn>4098562412</msn>
      <indv_status>1</indv_status>
    </element>
    <element>
      <global_device_id>m97524158</global_device_id>
        <msn>4097524158</msn>
      <indv_status>1</indv_status>
    </element>
  </data>
  <message>MDM will send all data of Meter with indv_status 1</message>
</root>
```

4.14 TRANSACTION_STATUS_REQUEST

Request:

Method: POST

URL: https://host:port/transaction_status

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters key	Parameters Value

No Body parameters.

Response:

JSON:


```
{
    "status": "1",
    "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3aRi",
    "data": [{
        "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3aRi",
        "global_device_id": "m98562412",
        "msn": "4098562412",
        "type": "On-Demand Request",
        "type_parameters": "",
        "command_receiving_datetime": "2018-10-15 16:45:30",
        "status_level": "4",
        "status_1_datetime": "2018-10-15 16:45:40",
        "status_2_datetime": "2018-10-15 16:45:50",
        "status_3_datetime": "2018-10-15 16:46:00",
        "status_4_datetime": "2018-10-15 16:46:10",
        "indv_status": "1",
        "request_cancelled": "0",
        "request_cancel_reason": "",
        "request_cancel_datetime": "0000-00-00 00:00:00",
        "response_data": [{
            "global_device_id": "m98562412",
            "msn": "4098562412",
            "meter_datetime": "2018-10-11 15:14:00",
            "event_code": "1234",
            "event_counter": "5",
```

```
                    "event_description": "event1",
                    "read_datetime": "2018-10-11 15:13:00"
                },
                {
                    "global_device_id": "m98562412 ",
                    "msn": "4098562412",
                    "meter_datetime": "2018-10-11 15:14:00",
                    "event_code": "2345",
                    "event_counter": "5",
                    "event_description": "event2",
                    "read_datetime": "2018-10-11 15:13:00"
                }
            ]
    },
    {
            "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3aRi",
            "global_device_id": "m97524158",
            "msn": "4097524158",
            "type": "On-Demand Request",
            "type_parameters": "",
            "command_receiving_datetime": "2018-10-15 16:45:30",
            "status_level": "4",
            "status_1_datetime": "2018-10-15 16:45:40",
            "status_2_datetime": "2018-10-15 16:45:50",
            "status_3_datetime": "2018-10-15 16:46:00",`
            "status_4_datetime": "2018-10-15 16:46:10",
            "indv_status": "1",
            "request_cancelled": "0",
            "request_cancel_reason": "",
            "request_cancel_datetime": "0000-00-00 00:00:00",
            "response_data": [{
                    "global_device_id": "m97524158",
                    "msn": "4097524158",
                    "meter_datetime": "2018-10-11 15:14:00",
                    "event_code": "1234",
                    "event_counter": "5",
                    "event_description": "event1",
```

```
                                        "read_datetime": "2018-10-11 15:13:00"
                                    },
                                    {
                                        "global_device_id": "m97524158",
                                        "msn": "4097524158",
                                        "meter_datetime": "2018-10-11 15:14:00",
                                        "event_code": "2345",
                                        "event_counter": "5",
                                        "event_description": "event2",
                                        "read_datetime": "2018-10-11 15:13:00"
                                    }
                                ]
                            }
                        ],
                        "message": "events data successfully sent"
                }
```


XML:


```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <data>
      <element>
         <command_receiving_datetime>2018-10-15 16:45:30</command_receiving_datetime>
         <global_device_id>m98562412</global_device_id>
        <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
         <indv_status>1</indv_status>
         <msn>4098562412</msn>
         <request_cancel_datetime>0000-00-00 00:00:00</request_cancel_datetime>
         <request_cancel_reason> </request_cancel_reason>
         <request_cancelled>0</request_cancelled>
         <response_data>
            <element>
               <event_code>1234</event_code>
               <event_counter>5</event_counter>
               <event_description>event1</event_description>
               <global_device_id>m98562412</global_device_id>
               <meter_datetime>2018-10-11 15:14:00</meter_datetime>
               <msn>4098562412</msn>
```

```
            <read_datetime>2018-10-11 15:13:00</read_datetime>
        </element>
        <element>
            <event_code>2345</event_code>
            <event_counter>5</event_counter>
            <event_description>event2</event_description>
            <global_device_id>m98562412 </global_device_id>
            <meter_datetime>2018-10-11 15:14:00</meter_datetime>
            <msn>4098562412</msn>
            <read_datetime>2018-10-11 15:13:00</read_datetime>
        </element>
    </response_data>
    <status_1_datetime>2018-10-15 16:45:40</status_1_datetime>
    <status_2_datetime>2018-10-15 16:45:50</status_2_datetime>
    <status_3_datetime>2018-10-15 16:46:00</status_3_datetime>
    <status_4_datetime>2018-10-15 16:46:10</status_4_datetime>
    <status_level>4</status_level>
    <type>On-Demand Request</type>
    <type_parameters />
</element>
<element>
    <command_receiving_datetime>2018-10-15 16:45:30</command_receiving_datetime>
    <global_device_id>m97524158</global_device_id>
   <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
    <indv_status>1</indv_status>
    <msn>4097524158</msn>
    <request_cancel_datetime>0000-00-00 00:00:00</request_cancel_datetime>
    <request_cancel_reason></request_cancel_reason>
    <request_cancelled>0</request_cancelled>
    <response_data>
        <element>
            <event_code>1234</event_code>
            <event_counter>5</event_counter>
            <event_description>event1</event_description>
            <global_device_id>m97524158</global_device_id>
            <meter_datetime>2018-10-11 15:14:00</meter_datetime>
            <msn>4097524158</msn>
```

```
            <read_datetime>2018-10-11 15:13:00</read_datetime>
          </element>
          <element>
            <event_code>2345</event_code>
            <event_counter>5</event_counter>
            <event_description>event2</event_description>
            <global_device_id>m97524158</global_device_id>
            <meter_datetime>2018-10-11 15:14:00</meter_datetime>
            <msn>4097524158</msn>
            <read_datetime>2018-10-11 15:13:00</read_datetime>
          </element>
        </response_data>
        <status_1_datetime>2018-10-15 16:45:40</status_1_datetime>
        <status_2_datetime>2018-10-15 16:45:50</status_2_datetime>
        <status_3_datetime>2018-10-15 16:46:00</status_3_datetime>
        <status_4_datetime>2018-10-15 16:46:10</status_4_datetime>
        <status_level>4</status_level>
        <type>On-Demand Request</type>
        <type_parameters />
      </element>
    </data>
    <message>events data successfully sent</message>
    <status>1</status>
    <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
</root>
```