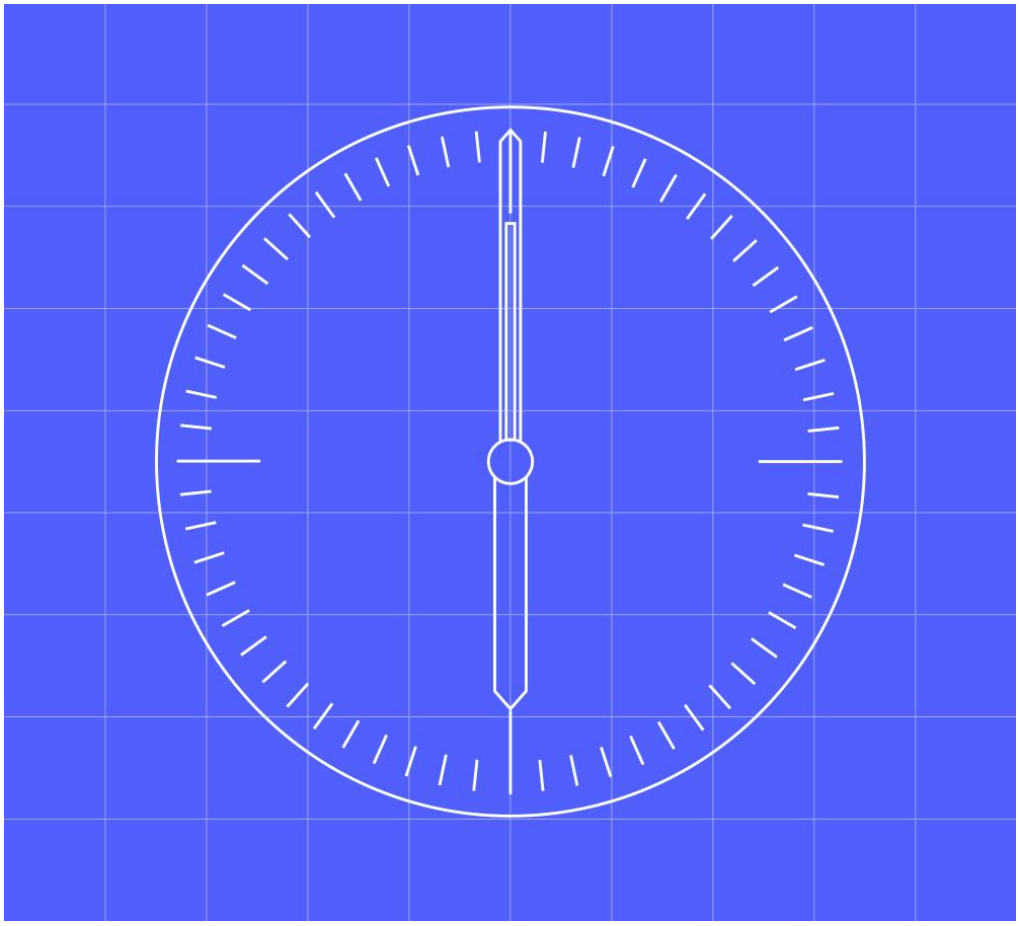
MHHS TOM – End-to-End Solution Architecture



Document owner	Document number	Version	Status:	Date
MHHS DAG	**MHHS-E2E001**	**Version 3.1.12.4**	**Design Updates~~Review~~**	**6320th ~~March~~February 2023**

Formatted: Header, Indent: Left: -0.2 cm

Formatted: Header, Centered

Formatted: Header, Right, Right: -0.2 cm

Formatted Table

1 Contents

1.1 Change Record

Date	Author	Version	Change Detail
26 May 2022	RG	0.1	First draft
1 August 2022	RG	0.2	Updated after first review
23 September 2022	RG	1.3	Updates after latest round of industry reviews and design clarification. Key updates: • Remove Sender Envelope Reference • Format of Sender Unique Reference is mandated
31 October 2022	RG	2.0	DAG Approval
10 November 2022	RG	2.1	Minor updates
19 December	RDG	2.2	Minor updates
?		2.3	
3 February 2023	RDG	2.4	Minor updates to align with other changes in the interfaces
		3	DAG Sign-off
20 March 2023	RDG	3.1	Minor updates for mandatory null issue & API end points and other clarifications coming through from detailed design
30 May 2023	RT	3.1.1	Minor corrections following review

Formatted: Normal Indent, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.07 cm, Relative to: Paragraph, Horizontal: 0.32 cm, Wrap Around

1.2 Reviewers

Reviewer	Role
TDWG	Review (first draft)
MHHS Program Parties	Review (second draft)
DAG	For information and sign off

1.3 References

Document/Link	Publisher	Published	Additional Information
MHHS Programme Governance Framework - (Strawman)	MHHS	v1.0	
The Target Operating Model for Market-wide Half Hourly Settlement - Design Working Group's Recommendation to Ofgem	Elexon	V1.1	12 February 2019
MHHS AWG Recommendation	Elexon	v1.0	22 April 2021
MHHS Risk and Management Process	MHHS	v0.1	August 2021 – in preparation
MHHS Architecture Principles	MHHS	V1.0	June 2022
MHHS End-to-End Security Architecture	MHHS	V1.0	
MHHS-DIP001-DIP Functional Specification	MHHS	V2.2	August 2022
MHHS-DIP002-Non-Functional Requirements	MHHS	V2.1	April 2022
MHHS-E2E003-Physical Interface Specifications	MHHS	V4.0	MHHSP-DES138-Interface Catalogue~~Interfaces spreadsheet~~
MHHS-E2E002-Requirements	MHHS	V2.2	End-to-End Requirements

1.4 Terminology

Term	Description
ARP	Advanced Retrieval and Processing Service
AWG	Architectural Working Group
BSCCo	Balancing and Settlement Code Company (Elexon Limited)
COS	Change of Supplier
DEV	Development
DIP	Data Integration Platform
DIPSP	DIP Service Provider
DCC	Data Communications Company
DNO	Distribution Network Operator
DPIA	Data Protection Impact Assessment
DTN	Data Transfer Network
DTS	Data Transfer Service
DWG	Design Working Group
ECOES	Electricity Central Online Enquiry Service
ESO	Enduring Service Owner, i.e. the party with overall responsibility for the DIP
EDA	Event-Driven Architecture
HHR	Half Hour
HHS	Half-Hourly Settlement
HTTP	Hypertext Transfer Protocol
iDNO	Independent Distribution Network Operator
ISD	Industry Standing Data
JSON	JavaScript Object Notation
LDSO	Licensed Distribution System Operator
LSS	Load Shaping Service
MDR	Meter Data Retrieval Service
MDS	Market-wide Data Service
MHHS	Market-Wide Half-Hourly Settlement
MPAN	Metering Point Administration Number
MRS	Meter Reading Service
MSA	Metering Service (Advanced)
MSS	Metering Service (Smart)
mTLS	Mutual Transport Layer Security
Obo	On Behalf Of
PRE-PROD	Pre Production
PROD	Production
PSS	Processing Service (Smart)
RECCo	Retail Energy Code Company (Retail Energy Code Limited)
RFP	Request for Proposal
SD	Settlement Day
SDS	Smart Data Services
SIT	System Integration Test
REGS	Registration Service
SP	Settlement Period
SUP	Supplier
SUR	Sender Unique Reference
SVA	Supplier Volume Allocation
SWIKI	Switching Public Key infrastructure
TLS	Transport Layer Security
TOM	Target Operating Model
UIT	User Integration Test
UMDS	Unmetered Supplies Data Service
UMSO	Unmetered Supplies Operator Service
VAS	Volume Allocation Service

2 Introduction

2.1 Background

Since 2018 Ofgem has hoped that Half-Hourly Settlement (HHS) on a market-wide basis would be introduced into the UK electricity market. A cross-industry Design Working Group (DWG) was established to understand the feasibility of HHS and how it could be delivered. The DWG produced a Target Operating Model (TOM) that outlines the new ways of working which could deliver HHS into the market (*Reference Target Operating Model for Market-wide Half Hourly Settlement).*

In conjunction with the DWG, and Architectural Working Group (AWG) was established to propose an IT architecture that could support the business process outlined in the TOM. The AWG recommendation was that an Event-Driven Architecture (EDA) be implemented (*Reference MHHS AWG Recommendation*). Hence, a new message orientated/event-driven middleware component is required – the Data Integration Platform (DIP) - to support the flow of events/messages between industry participants proposed by the EDA.

On the back of this work the MHHS Programme was established to create a durable, faster, more accurate settlement process for all market participants, enabling broad change across the electricity industry.

The MHHS programme has refined the original TOM. The final TOM defines a set of services required to deliver Settlement Period (SP) data from a Meter to a central Settlement body to enable the calculation of the amount of energy that the electricity Supplier's customers have consumed (or exported) in each Settlement Period for each Settlement Day (SD). This calculation is then used in the Imbalance Settlement process, which compares the Supplier's contracted purchases of energy to the amounts deemed to have been consumed (sales) by each of the Supplier's customers (and recognises any amounts of energy contracted by National Grid under the Balancing Mechanism). Settlement Data is also provided for network charging.

In addition to these core services, a number of supporting services need to be established to ensure the smooth running of the electricity market with the move to market-wide half-hour metering.

2.2 Objective

This document describes the end-to-end solution architecture that will support the proposed MHHS TOM and provides a description of how the different actors within the MHHS landscape interact.

2.3 Assumptions

The end-to-end design has the following assumptions:

1. The choice of the underlying platform to support the DIP message/event processing will not have a consequential effect on the end-to-end design
2. The DIP is not responsible for orchestrating business process, e.g. ensuring correlated message exchanges occur.
3. The End-to-End Architecture Security requirements are covered in *MHHSP-DIP005-E2E Security Requirements.*

2.4 Risks

The document is prepared with the following prevailing risks:

1. ~~The design is progressing to a detailed stage without an appointed DIP Service Provider and chosen architecture.~~ Currently none

2.5 Document Scope

The scope of the document includes

- An overview of the different MHHS Actors and Roles;

- An overview of the MHHS Workflows/business processes:
- A description of the message architecture including:
    - Message formats
    - Message orchestration
- A view of the requirements to deliver the end-to-end solution:
    - Participant responsibilities
    - DIP responsibilities

The document provides a narrative for the detailed requirements for the e2e solution architecture that can be found in the accompanying spreadsheet (*Reference MHHSP-DIP009-E2E Requirements, April 2022).*

Participants need to use this document as well as the following in order to understand the full impact that the MHHS TOM will have on their electricity market IT systems:

The document does not cover the following areas:

- Participant on-boarding
- DIP Reporting

3 MHHS TOM - Functional Architecture

3.1 Key Assumptions/Principles

The following are the key design assumptions when establishing the message flows within the MHHS TOM:

1. The DIP will broker new message flows between Market Participants supporting the business process underpinning the MHHS TOM, i.e. there are no direct point-to-point interfaces between participants.
2. Some of the reworked existing business processes falling under the programme's scope have existing data flows that use the DTN, and these will be retained where there is no change to that interface. When a MHHS initiated change to a DTN flow is required then each change is considered on an individual case-by-case basis as to whether the DTN flow is reworked or re-implemented in the DIP.

3.2 MHHS TOM - Scope

The scope of the MMHS work covers the “Meter to Bank” process for all Supplier Volume Allocation (SVA) Settlement Meters – i.e., all Settlement Meters connected to distribution networks. The business architecture is described as a set of services, with each service defined by a set of requirements and processes required to deliver one function of MHHS.

A service is agnostic of current organisation roles (such as Supplier, Meter Operator, Data Collector, Data Aggregator).

The list of services includes:

- Registration Service - the recording the ownership of supply points and other details information pertinent to Settlement Metering Systems;
- Metering Service - fitting and maintaining Settlement Meters;
- Data Retrieval - getting information from Settlement Meters;
- Data Processing (Data Services) – validating and estimating Settlement Meter data;
- Smart Metering Data Service Provider, i.e. the DCC
- Load Shaping Service – settlement service use to produce Load Shaping data
- Data Aggregation - summing Settlement Meter data to required granularity; and
- Volume Allocation – allocating Meter volumes to Trading Parties' signatories to the BSC.
- Supplier – energy supplier in the retail market
- Messaging Service – passes messages between other services, i.e. the DIP, DTS.
- Electricity Enquiry Service.

3.3 MHHS TOM - Actors/Services

A number of different roles and Users (Market Participants) have been identified. Those actors that are active within the DIP are marked *:

Role ID	DIP Active	Service Name	Market Segment/Role	Market Participant Role
MSA	*	Metering Service (Advanced)	Advanced Market Segment	T
ADS	*	Advanced Data Service		O

Formatted Table

MSS	*	Metering Service (Smart)	Smart and Traditional (non-Smart) Market Segments	S
MRS		Metering Reading Service		6
SDS	*	Smart Data Service		N
MDR		Meter Data Retrieval Service		
UMSO	*	Unmetered Supplies Operator Service	Unmetered Supplies Market Segment	3
UMDS	*	Unmetered Supplies Data Service		O
MDS	*	Market-wide Data Service	BSC Central Settlement (CS)	n/a
LSS	*	Load Shaping Service		n/a
ISD	*	Industry Standing Data		n/a
VAS	*	Volume Allocation Service		n/a
REGS	*	Registration Service	Registration	P
SUP	*	Supplier	Supplier	X
MAP		Meter Asset Provider		8
LDSO	*	Licensed Distribution System Operator (IDNOs & DNOs)	Distribution Network Operator	R
EES	*	Electricity Enquiry Service	RECCo	L
~~**SMSO**~~[1]		~~Smart Meter System Operator~~		
CSS		Central Switching Service	Messaging/Orchestration Service	
DCC		Data Communication Company		
DTS		Data Transfer Service		
DIP		Data Integration Platform		
DCP	*	DIP Connection Provider		

Some of these different roles have the capacity to operate within several different contexts for each the TOM business processes, for example a Supplier within a business process can act as the incumbent supplier, or be a previous or future supplier. Where the separation of the roles into different contexts is required this is described in the corresponding business process diagrams and summarised below.

3.3.1 Supplier (SUP)

The Supplier are responsible for supplying electricity to the end customer identified by the customer's MPAN. The Supplier is responsible for migrating MPANs from the current settlement into the MHHS arrangements.

[1] ~~SMSO, CSS & DTS are not active within the DIP~~

3.3.2 Metering Service (MSA & MSS)

The principal functions of a Metering Service (Smart (MSS) and Advanced (MSA)) are to install, commission, test, maintain, rectify, energise and remove faults in respect of Metering Equipment (including, where applicable, associated Communications Equipment). The MSS and MSA will also maintain and make available Meter asset information and, where required, Meter configuration information.

3.3.3 Data Service (SDS & ADS)

The Data Services are responsible for data retrieival, validation, and submission of consumption data into Settlements. The Smart Data Service (SDS) will support Smart and Traditional meters, with the Advanced Data Service (ADS) supporting Advanced Meters. Both Data Services will remotely recover data from their respective meter populations, making arrangements for manual readings where remote connection is not possible. The Service Request type and schedule are provided by the Processing Service (PSS) for each Metering Point Administration Number (MPAN) which is under the responsibility of the PSS.

3.3.4 Meter Data Retrieval Service (MDR)

The Meter Data Retrieval (MDR) Service is really a sub-component of the Smart Data Service (SDS). However, is called out here as it is a qualified party in its own right. The MDR will take specific responsibility for all interactions with the DCC, in terms of DCC Service Request management for access to Smart Meters. All data recovered or exceptions received will be passed back to the SDS for investigation and resolution.

3.3.5 Unmetered Supplies Operator Service (UMSO)

The Unmetered Supplies Operator (UMSO) is responsible for validating the detailed unmetered supplies inventory data for equipment attached to its distribution network and providing information to other industry stakeholders. It interfaces with customers who own/operate the unmetered equipment (referred to as the Unmetered Supplies customer).

3.3.6 Unmetered Supplies Data Service (UMSDS)

The Unmetered Supplies Data Service (UDS aka UMSDS) is responsible for calculating Settlement Period (SP) level consumption data for unmetered equipment, for example, streetlights and traffic signals.

3.3.7 Central Settlement (CS)

Central Settlement is the parent role undertaken by BSCSCo (Elexon). It encompasses the child roles of MDS, LSS, ISD and VAS.

3.3.8 Market-wide Data Service (MDS)

The Market-wide Data Service (MDS) is responsible for processing Settlement Period level data from the Smart Data Service (SDS) for smart and non-smart Meters; and Advanced Data Service (ADS) for Advanced Meters; and Unmetered Supplies Data Services (UMDS) for unmetered equipment. The MDS will provide data aggregations for Imbalance Settlement and other purposes (such as network charges and flexibility offerings (if required).

3.3.9 Load Shaping Service (LSS)

The Load Shaping Service (LSS) is responsible for calculating energy consumption (import and export) Load Shapes for a number of defined categories of Metering Systems. The LSS uses validated actual Settlement Period (SP) level data accessed from the Data Services. The Data Services will then use the Load Shape data to convert Register Readings (RRs) or daily consumption values into SP level data. The Load Shape data will also be used to estimate invalid SP level data for Meters and default where data is missing or unavailable.

3.3.10 Volume Allocation Service (VAS)

The Volume Allocation Service (VAS) is responsible for accessing aggregated Settlement Period (SP) level data from the Market-wide Data Service (MDS); and SP level data (Grid Supply Point Group Takes) from the Central Data Collection Agent (CDCA). The VAS calculates SP level energy volumes for Balancing Mechanism Units (BMUs) using these two datasets. The data is processed for each Settlement Day in a scheduled run called a Volume Allocation Run (VAR). The processed BMU data is used in the Imbalance Settlement calculations. The VAS will also allocate or aggregate data for other purposes and provide a wide range of data reporting.

3.3.11 Industry Standing Data (ISD)

Industry Standing Data process is responsible for publishing Industry Standing Data. The process is owned and managed by the BSCCo. ISD can be downloaded from the Elexon portal on an ad-hoc basis, and updates will be notified by the DIP to Market Participants to an agreed timetable (monthly). It will replace the current D269 and D270 data flows.

3.3.12 Data Integration Platform (DIP)

The Data Integration Platform (DIP) is responsible for brokering messages between the different MHHS actors and services. The DIP performs the addressing and routing of messages between participants.

3.3.13 Registration Service (REGS)

The Registration Service is the LDSO service that holds Meter point standing data information about each MPAN within its distribution Region. Data includes the BRP the processing and metering services appointed to the MPAN. It also includes information on the type of customer, the Measurement Class, Energisation Status and Line Loss Factor Class. In the MHHS TOM the Registration Service will act as the definitive source of all settlement impacting data items. The Registration Service, leveraging is existing connectivity with the CSS (Central Switching Service) shall also provide the MHHS view of Supply Point ownership.

3.3.14 Data Transfer Service (DTS)

The DTS is the existing mechanism that is used to exchange information between participants in the UK utility markets. The DTS provides a managed file transfer service that allows market participants to share data efficiently and securely to perform their roles in the market. The DTS service operates over the Data Transfer Network (DTN). Often the two terms are used synonymously.

Many of the workflows described below will use the data exchanges that the DTN orchestrates, and the DTN is considered similarly to the DIP.

3.3.15 Central Switching Service (CSS)

The CSS provides the capability for domestic consumers to changes energy suppliers. Consumers are able to transfer data, payment details and account information within five working days, and will later be able to switch within 24 hours.

3.3.1 Electricity Enquiry Service (EES)

The Electricity Enquiry Service (EES) allows users to access market data where they are entitled to do so. Data is sourced from either the Supplier Meter Registration Service, or the Central Switching Service (CSS) and moving forward the DIP. The service consists of an on-line portal where the data for all electricity Registrable Measurement Points (RMPs) is available and API service which allows Enquiry Service User to gather information from the service in a specified manner.

3.3.2 DIP Connection Provider (DCP)

A DIP Connection Provider is a third-party who provides connectivity services to the DIP on behalf of DIP Participants. Some DIP Participants will not want to have a direct connection to the DIP and instead want to have this service undertaken by a DIP Connection Provider.

3.4 MHHS TOM - Workflows

The following section describes at a high level the initial list of workflows that the programme is aiming to deliver.

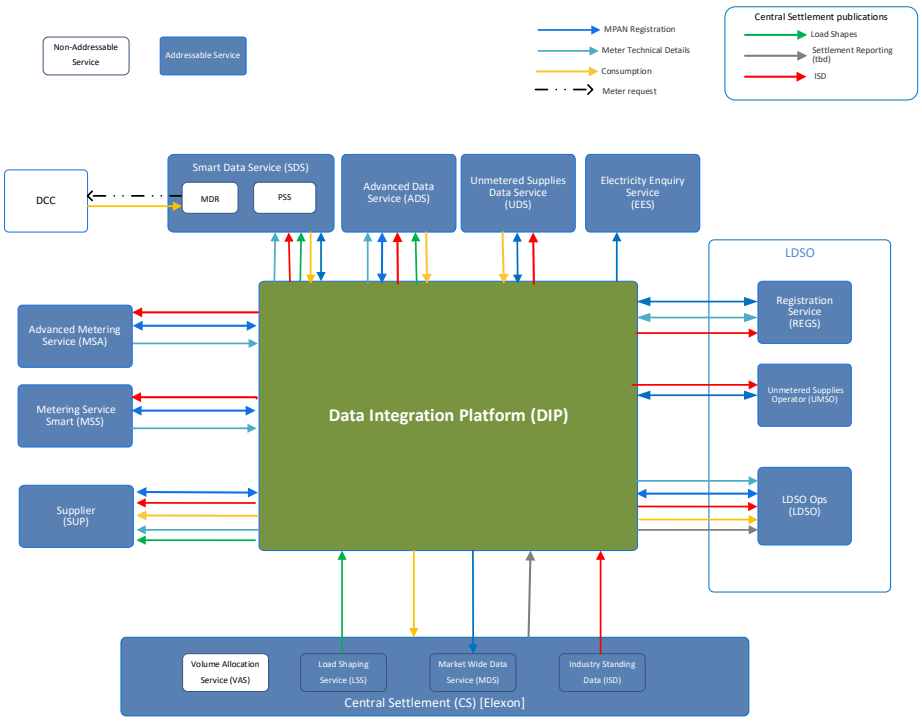


Figure 1 MHHS TOM Workflows

MHHS TOM landscape will be a distributed network of services and roles requiring constant data communication for operational purposes. There are approximately 27 Registration Services that will need to maintain operational data integrity and consistency across approximately 35 Data Services, 85 Metering Services, 17 DNO's and 60 Suppliers. In addition, the Data Services must provide an approximate total of 32 million daily (15 billion annual) consumption events to Central Settlement.

The MHHS TOM is divided in 6 different functional areas:

1. MPAN Registration
2. Consumption
3. Meter Technical Details
4. Load Shapes
5. Industry Standing Data (ISD)
6. Settlement Reporting

A summary of the business processes (workflows) orchestrated within the MHHS TOM, including the dataflows (interfaces) that need to be undertaken by the DIP, are provided below:

3.4.1 BP-001 Change of Supplier

BP-001	Change of Supplier
TOM Sector	Registration
Principal Actors	Supplier (Outgoing/Incoming) Registration Service (REGS) Central Settlements (CS)
Additional Actors	CSS, LDSO, Metering Services, EES
Correlation Id	
Interfaces	IF-001/PUB-001 - Notification of Change of Supplier IF-002/PUB-002 - Change of Supplier – Registration Update to Supplier
Other interfaces (DTN, CSS)	CSS1000 CSS2800 CSS2370 CSS2860
Linked BP	BP002 BP003 BP004 BP009 BP008 BP003C
Linked design artefact	BPD001

3.4.2 BP-002 Change of Service – Metering Service

BP-002	Change of Service – Metering Service
TOM Sector	Registration
Principal Actors	Supplier (SUP) (Prospective/Current) Registration Service (REGS) New Metering Service (MSS, MSA, UMSO) Old Metering Service (MSS, MSA, UMSO) New/Existing Data Service (SDS, ADS, UMSDS) LDSO
Additional Actors	Electricity Enquiry Service (EES)
Correlation Id	
Interfaces	IF-031/PUB-031 - Supplier Service Appointment Request IF-032/PUB-032 - Reg. Serv. Response to Supplier Service App Request IF-033/PUB-033 - Registration Service Request for Service Appointment IF-034/PUB-034 - Service Provider Response to Appointment Request IF-035/PUB-035 - Registration Service Appointment Status Notification IF-036/PUB-036 - Reg. Serv. Notification of Service App. & Supporting Info IF-037/PUB-037 - Registration Service Notification of Service De-Appointment IF-038/PUB-038 - Customer Direct Contract Advisory IF-039/PUB-039 - Customer Direct Contract Advisory Response
Other interfaces (DTN, CSS)	D149 / D150 – Traditional Meter MTDs D268/D383/D384 – Advanced Meter MTDs
Linked BP	BP001-Change of Supply BP007-Disconnection BP011-Change of Market Segment BP004-Data Collection

	BP003 – Change of Service – Data Service BP003A – CSS/DCC Update
Linked design artefact	BPD002

3.4.3 BP-003 Change of Service – Data Service

BP-003	Change of Service – Data Service
TOM Sector	Registration
Principal Actors	Supplier (SUP) (Prospective/Incumbent) Registration Service (REGS) New Data Service (SDS, ADS,UMSDS) Metering Service (MSS, MSA, UMSO) Existing Data Service (SDS, ADS, UMSDS) LDSO Electricity Enquiry Service (EES)
Correlation Id	
Interfaces	IF-031/PUB-031 - Supplier Service Appointment Request IF-032/PUB-032 - Reg. Serv. Response to Supplier Service App Request IF-033/PUB-033 - Registration Service Request for Service Appointment IF-034/PUB-034 - Service Provider Response to Appointment Request IF-035/PUB-035 - Registration Service Appointment Status Notification IF-036/PUB-036 - Reg. Serv. Notification of Service App. & Supporting Info IF-037/PUB-037 - Registration Service Notification of Service De-Appointment IF-038/PUB-038 - Customer Direct Contract Advisory IF-039/PUB-039 - Customer Direct Contract Advisory Response
Other interfaces (DTN, CSS)	D149 / D150 – Traditional Meter MTDs D268 – Advanced Meter MTDs
Linked BP	BP001-Change of Supply BP007-Disconnection BP011-Change of Market Segment BP003A – CSS/DCC Update BP003C – Transfer of Reads BP004B – UMS Data Collection
Linked design artefact	BPD003

3.4.4 BP-003A CSS and DCC Update

BP-003A	CSS and DCC Update
TOM Sector	Registration
Principal Actors	Registration Service CSS (Central Switching Service) DCC (Data & Communications Co.) Data Service (Incoming) Metering Service (Incoming)
Correlation Id	

Interfaces	IF-035/PUB-035 - Registration Service Appointment Status Notification
Other interfaces (DTN, CSS)	Existing CSS to DCC Interface Existing DCC to CSS Interface CSS0200
Linked BP	BP003-Change of Service – Data Service BP003B – Change of Existing Service Appointment
Linked design artefact	BPD03A

3.4.5 BP003B Change of Existing Service Appointment

BP-003B	Change of Existing Service Appointment
TOM Sector	Registration
Principal Actors	Registration Service Supplier Existing Metering Service (MSA, ~~UMSO~~MSS, UMSO) Existing Data Service (SDS, ~~UMSDS~~ADS, UMSDS)
Additional Actors	EES
Correlation Id	
Interfaces	IF-031/PUB-031 - Supplier Service Appointment Request IF-032/PUB-032 - Reg. Serv. Response to Supplier Service App Request IF-033/PUB-033 - Registration Service Request for Service Appointment IF-034/PUB-034 - Service Provider Response to Appointment Request IF-035/PUB-035 - Registration Service Appointment Status Notification
Other interfaces (DTN, CSS)	
Linked BP	BP-003A - CSS/DCC Update BP-004 - Data Processing
Linked design artefact	BPD03B

3.4.6 BP-003C Transfer of Reads - Change of Data Service/Agreed Reads Process

BP-003C/D	Transfer of Reads – Change of Data Service
TOM Sector	Metering & Data Services
Principal Actors	New/Outgoing Supplier New/Outgoing Data Service (SDS, ADS)
Additional Actors	
Correlation Id	
Interfaces	IF-015/PUB-015 - Request Consumption History IF-041/PUB-041 - Cumulative Reading D0300 – Disputed or missing Readings on Change of Supplier
Other interfaces (DTN, CSS)	D0010 – Register Reading(s)

Linked BP	BP-001 Change of Supplier BP-003 Change of Data Service BP-004 – Data Collection BP-005 - Data Processing
Linked design artefact	BPD03C

3.4.7 BP-004 Data Collection

BP-004	Data Collection
TOM Sector	Metering & Data Services
Principal Actors	Supplier Data Service (SDS, ADS, UMSDS) Metering Service (MSA, UMSO)
Additional Actors	DCC MDR
Correlation Id	
Interfaces	IF-024/PUB-024 - Supplier Advisory Notifications IF-041/PUB-041 - Cumulative Reading IF-047/PUB-047 - Notification of the Publication of a Downloadable Asset (ISD)
Other interfaces (DTN, CSS)	D0010 – Register Reading(s) D0004 – Notification of Failure to Obtain Reading D0001 – Request Metering System Investigation DCC- SRV5.1 DCC – SRV4.x.x D0388 – UMS Inventory D0389 – UMS Response
Linked BP	BP001 - Change of Supply BP003 - Change of Agent – Data Services BP008 – Change of Energisation BP005 – Data Processing
Linked design artefact	BPD004

3.4.8 BP-005 Data Processing

BP-005	Data Processing
TOM Sector	Metering & Data Services
Principal Actors	Data Service (SDS, ADS, UMSDS) LDSO Supplier Central Settlements (CS, LSS, MDS, VAS) Registrations Service
Additional Actors	EES
Correlation Id	
Interfaces	IF-021/PUB-021 – Settlement Period Consumption Data IF-022/PUB-022 - LSS Period Data IF-023/PUB-023 – LSS Totals Data IF-040/PUB-040 – Annual Consumption

	IF-041/PUB-041 – Cumulative Reading
Other interfaces (DTN, CSS)	D0010 – Register Reading(s)
Linked BP	BP004 – Data Collection BP018 – Load Shaping Service BP019 – Market-wide Data Service
Linked design artefact	BPD005

3.4.9 BP-007 Disconnection

BP-007	Disconnection
TOM Sector	Registration
Principal Actors	Supplier (SUP) LDSO Registration Service (REGS) Data Service (SDS, ADS, UMSDS) Metering Service (MSS, MSA, UMSO) Central Settlements (CS)
Other Actors	Electricity Enquiry Service (EES) CSS
Correlation Id	
Interfaces	IF-009/PUB-009 - Registration Notification of Disconnection IF-037/PUB-037 - Notification of De-Appointment
Other interfaces (DTN, CSS, other)	CSS1900 CSS02370 CSS0300 CSS02860 D0262 – Rejection of Disconnection D0132 – Details of Disconnection of Supply DB03
Linked BP	BP008-Change of Energisation BP009-Change of Meter
Linked design artefact	BPD007

3.4.10 BP-008 Change of Energisation

BP-008	Change of Energisation
TOM Sector	Registration
Principal Actors	Supplier (SUP) LDSO Registration Service (REGS) Data Service (SDS, ADS, UMSDS)

	Metering Service (MSS, MSA, UMSO) Central Settlements (LSS, MDS)
Other Actors	EES
Correlation Id	
Interfaces	IF-007/PUB-007 - Update of Change of Energisation Status Outcome to Registration IF-008/PUB-008 - Registration Service Notification of Change of Energisation Status IF-041/PUB-041 - Cumulative Meter Reading
Other interfaces (DTN, CSS)	D010 – Register Reading(s) D0134 – Request to Change Energisation Status D0139 – Confirmation or Rejection of Energisation Status Change D0179 – Confirmation of Energisation/De-Energisation of Prepayment Meter
Linked BP	BP005 – Data Processing BP004 – Data Collection BP009 – Change of Metering
Linked design artefact	BPD008

3.4.11 BP-009 Change of Metering

BP-009	Change of Meter
TOM Sector	Registration
Principal Actors	Supplier (SUP) LDSO Registration Service (REGS) Data Service (SDS, ADS) Metering Service (MSS, MSA)
Other actors	EES
Correlation Id	
Interfaces	IF-005/PUB-005 - Metering Service MTD Updates to Registration IF-006/PUB-006 - Registration Service Notification of MTD Update(s) IF-041/PUB-041 - Cumulative Meter Reading
Other interfaces (DTN, CSS)	D0149/D0150 – Traditional MTDs D268/D383/D384 – Advanced MTDs D0010 – Register Readings(s) D0142 – Request for Installation or Change to a Metering System D0002 – Fault Resolution Report or Request Decision for Further Action D0221 – Notification of Failure to Install Metering System
Linked BP	BP008-Change of Energisation BP001 – Change of Supply BP007-Disconnection BP011-Change of Market Segment BP004 – Data Collection
Linked design artefact	BPD009

3.4.12 BP-010 Change of Registration Data

BP-010(A/B/C/D)	Change of Registration Data
TOM Sector	Registration
Principal Actors	Supplier (SUP) CSS LDSO

	Registration Service (REGS) Data Service (SDS, ADS, UMSDS) Metering Service (MSS, MSA, UMSO) Central Settlements (LSS/MDS) DCC
Additional Actors	EES
Correlation Id	
Interfaces	IF-018/PUB-018 - Notification of Registration Data Item Changes IF-019/PUB-019 - Maintain MPAN Relationship IF-020/PUB-020 - Maintain MPAN Relationship Response IF-025/PUB-025 - Supplier Updates to Registration IF-026/PUB-026 - Registration Service Notification of Supplier Data Changes
Other interfaces (DTN, CSS)	CSS-2000 CS00300 DB02 D0386 – Manage Metering Point Relationships DB05 Existing DCC to Registration Interface
Linked BP	BP004 – Data Collection BP009 – Change of Meter BP011 – Change of ConType and/or Mkt Segment
Linked design artefact	BPD010

3.4.13 BP-011 Change of Connection Type and/or Market Segment

BP-011(11B/11C)	Change of Connection Type and/or Market Segment
TOM Sector	Registration
Principal Actors	Supplier Registration Service Data Service Existing & Prospective (ADS, SDS) Metering Service Existing & Prospective (MSS,MSA) LDSO Central Settlement
Additional Actors	EES
Correlation Id	
MHHS Interfaces	IF-031/PUB-037 – Supplier Service Appointment Request IF-032/PUB-032 – Registration Response to Service Appointment Request IF-033/PUB-033 – Registration Request for Service Appointment IF-034/PUB-034 – Service Provider Response to Appointment Request IF-035/PUB-035 – Registration Service Appointment Status Notification IF-036/PUB-036 – Registration Service Notification of Service Appointment & Supporting Info IF-037/PUB-037 – Notification of De-Appointment IF-005/PUB-005 - Metering Service MTD Updates to Registration IF-006/PUB-006 - Registration Service Notification of MTD Updates IF-043/PUB-043 - Notification of Change of Connection Type IF-044/PUB-44 – Notification of Change in Market Segment IF-045/PUB-45 – Notification of Invalid or No SP's Appointed
Other interfaces (DTN, CSS)	D0142 – Request for Installation or Change to a Metering System D0002 – Fault Resolution Report or Request Decision for Further Action D0221 – Notification of Failure to Install Metering System
Linked BP	BP002 - Change of Agent-Metering Services BP003 - Change of Agent- Data Service

	BP009 - Change of Meter BP010 – Change of Registration Data
Linked design artefact	BPD011

3.4.14 BP-013 Demand Disconnection

BP-013	Demand Disconnection Events
TOM Sector	Elexon Central Systems
Principal Actors	Electricity System Operator (ESO) Licensed Distribution System Operator (LDSO) Balancing Settlement Code Company (BSCCo) Market-wide Data Service (MDS) Industry Standing Data (ISD) Volume Allocation Run (VAS) Other BSC Systems
Correlation Id	
MHHS Interfaces	PUB-001 - Notification of Change of Supplier PUB-008 - Registration Service Notification of Change of Energisation Status PUB-009 - Notification of LDSO Disconnection / CSS De-Registration IF-013/PUB-013 -MDS Defaults Applied PUB-018 - Notification of Registration Data Item Changes IF-014/PUB-014 -Rejected Consumption Data Submission PUB-021- UTC Settlement Period Consumption Data PUB-026 - Registration Service Notification of Supplier Data Chg PUB-036 - Registration Service Notification of Service Appointment & Supporting Info PUB-043 - Registration Service Notification of Change of Connection Type PUB-044 - Registration Service Notification of Change of Segment PUB-045 - Registration Service Notification of Invalid Segment or No Agents Appointed
Other interfaces (DTN, CSS)	Reports MHHS-REP-001 MHHS-REP-002 MHHS-REP-005 MHHS-REP-006 MHHS-REP-D0369 MHHS-REP-D0370 MHHS-REP-D0373 MHHS-REP-D0374
Linked BP	BP005-Data Processing BP010-Change of Registration Data
Linked design artefact	BPD013

3.4.15 BP-016 Consumption Amendment

BP-016	'Override Read' Submission & Consumption Amendment Processing

TOM Sector	Elexon Central Systems
Principal Actors	Supplier Data Service Existing/Previous (SDS, ADS, UMSDS) Central Settlements BSC Dispute Process Consumption Amendment 'Audit Function' TBD LDSO
Correlation Id	
MHHS Interfaces	IF-027/PUB-027 - Consumption Amendment Request IF-028/PUB-028 - Consumption Amendment Outcome IF-021/PUB-021- UTC Settlement Period Consumption Data IF-014/PUB-014 ECS Rejection of Settlement Period Cons. Data IF-041/PUB-041 Cumulative Reading
Other interfaces (DTN, CSS)	
Linked BP	BP005 – Data Collection BP019 – Market-wide Data Service
Linked design artefact	BPD016

3.4.16 BP-018 Load Shaping Service

BP-018	Load Shaping Service
TOM Sector	Elexon Central Systems
Principal Actors	Load Shaping Service (LSS) Industry Standing Data (ISD) Market-wide Data Service (MDS)
Other Actors	EES
Correlation Id	No
MHHS Interfaces	IF-001/PUB-001- Notification of Change of Supplier IF-008/PUB008 - Registration Service Notification of Change of Energisation Status IF-009/ PUB-009 - Notification of LDSO Disconnection / CSS De-Registration IF-018/ PUB-018 - Notification of Registration Data Item Changes IF-021/PUB-021 - UTC Settlement Period Consumption Data IF-022/PUB-022 - LSS Period Data IF-023/PUB-023 – LSS Totals Data IF-026/PUB-026 -Registration Service Notification of Supplier Data Chg IF-036/PUB-036 - Registration Service Notification of Service Appointment & Supporting Info IF-043/PUB-043 - Registration Service Notification of Change of Connection Type IF-044/PUB-044 - Registration Service Notification of Change of Segment
Other interfaces (DTN, CSS, Elexon)	EL-022 EL-023
Linked BP	BP005 – Data Processing BP010 – Change of Registration Data BP019 – Market-wide Data Service BP001 – Change of Supply BP003 – Change of Data Service BP008 – Change of Energisation

	BP007 – Disconnection BP011 – Change of ConType and/or Mkt Segment
Linked design artefact	BPD018

3.4.17 BP-019 Market Wide Data Service

BP-019	Market Wide Data Service
TOM Sector	Elexon Central Systems
Principal Actors	Load Shaping Service (LSS) Volume Allocation Service (VAS) Industry Standing Data (ISD) Market-wide Data Service (MDS)
Other Actors	Other BSC Systems Electricity Market Reform (EMRS)
Correlation Id	No
MHHS Interfaces	IF-001/PUB-001 - Notification of Change of Supplier IF-008/PUB008 - Registration Service Notification of Change of Energisation Status IF-009/PUB-009 - Notification of LDSO Disconnection / CSS De-Registration IF-013/PUB-013 -MDS Defaults Applied IF-014/PUB-014 -Rejected Consumption Data Submission IF-018/PUB-018 - Notification of Registration Data Item Changes IF-021/PUB-021 - UTC Settlement Period Consumption Data IF-022/PUB-022 - LSS Period Data IF-026/PUB-026 -Registration Service Notification of Supplier Data Chg IF-036/PUB-036 - Registration Service Notification of Service Appointment & Supporting IF-040/PUB-040 - Notification of ECS Annual Consumption IF-043/PUB-043 - Registration Service Notification of Change of Connection Type IF-044/PUB-044 - Registration Service Notification of Change of Segment IF-045/PUB-045 Registration Service Notification of Invalid Segment or No Agents Appointed REPORTS MHHS-REP-002 MHHS-REP-002A MHHS-REP-090 MHHS-REP-006 MHHS-REP-020 MHHS-REP-030 MHHS-REP-060 ELEX-REP-001 MHHS-REP-009 MHHS-REP-D0354
Other interfaces (DTN, CSS, Elexon)	ELEX-REP-001
Linked BP	BP005 – Data Processing BP010 – Change of Registration Data BP018 – LSS BP012 – ISD BP020 – Volume Allocation Service BP001 – Change of Supply BP003 – Change of Service Provider – Data Service BP008 – Change of Energisation

	BP007 – Disconnection BP011 – Change of ConType and/or Mkt Segment
Linked design artefact	BPD019

3.4.18 BP-020 Volume Allocation Service

BP-020	Volume Allocation Service
TOM Sector	Elexon Central Systems
Principal Actors	Volume Allocation Service (VAS) Industry Standing Data (ISD) Market-wide Data Service (MDS)
Other actors	Other BSC Systems
Correlation Id	No
MHHS Interfaces	REP-003 REP-004 REP-007 REP-008 REP-090 ELEX-REP-040 ELEX-REP-050 ELEX-REP-060 ELEX-REP-080 ELEX-REP-P0048 ELEX-REP-P0236 ELEX-REP-P0237 ELEX-REP-D0081 ELEX-REP-D0296 ELEX-REP-D0266 ELEX-REP-D0369 ELEX REP-D0370 ELEX-REP-D0374 ELEX-REP-D0373
Other interfaces (DTN, CSS)	P0181 P0182 P0183 P0012 P0034 P0236 P0191
Linked BP	BP005 – Data Processing BP019- Market Wide Data Service BP021- Industry Standing Data
Linked design artefact	BPD020

3.4.19 BP-021 Industry Standing Data

BP-021	Industry Standing Data
TOM Sector	Elexon Central Systems

Principal Actors	BSCCo Industry Standing Data (ISD)
Other actors	Other BSC Systems (LSS, MDS, VAS) All TOM actors for receipt
Correlation Id	No
MHHS Interfaces	IF-047/PUB047 – ISD Data Notification
Other interfaces (DTN, CSS)	
Linked BP	BP018 – Load Shaping Service BP019 – Market-wide Data Service BP020 – Volume Allocation Service
Linked design artefact	BPD021

3.5 MHHS Message/Event Channel Instances

The MHHS interfaces have the non-functional requirements placed on each of the messages channels. The following metrics are defined:

- Sender – DIP will check participant has the correct role for the channel
- Recipient:
  - Always
  - Primary
  - Secondary
- Workflow Correlation Id – used to track business process – section 4.2.3.3 for description.
- Security category

For an up to date list see *MHHSP-DES138 Interfaces Catalogue* – IF_LIST tab

Formatted: Font: Italic

3.6 Actor Workflows Summary

The table below summarises the data flows that each service has some participation within; all the outbound (interfaces) and the inbound (publications) messages/events. (n.b. inbound/outbound with respect to the service).

Message Event Channels			
Actor	Context	Out bound	In bound
MSA	Sending, Current, Previous, Incoming, Prospective	IF-005, IF-007, IF034, IF-038, IF-041	PUB-006, PUB-008, PUB-018, PUB-026, PUB-033, PUB-035, PUB-036, PUB-037, PUB-039, PUB-043, PUB-045, PUB-047, PUB-020
MSS	Sending, Current, Previous, Incoming, Prospective	IF-005, IF-007, IF034, IF-038, IF-041	PUB-004, PUB-006. PUB-008, , PUB-018, PUB-026, PUB-033, PUB-035, PUB-036, PUB-037, PUB-039, PUB-043, PUB-045, PUB-047, PUB-020

UMSO	Sending, Current, Previous, Incoming, Prospective	IF-007, IF-034	PUB-008, PUB-033, PUB-035, PUB-036, PUB-037, PUB-043,
ADS	Sending, Current, Previous, Incoming, Prospective	IF-021, IF-015, IF034, IF-038, IF-041, IF-028	PUB-006, PUB-008, PUB-009, PUB-016, PUB-013, PUB-014, PUB-018, PUB-022, PUB-023, PUB-024, PUB-026, PUB-027 PUB-033, PUB-035, PUB-036, PUB-037, PUB-039, PUB-040, PUB-041, PUB-043, PUB-045, PUB-047, PUB-020
SDS	Sending, Current, Previous, Incoming, Prospective	IF-021, IF034, IF-038, IF-041, IF-028	PUB-006, PUB-008, PUB-009, PUB-013, PUB-014, PUB-018, PUB-022, PUB-023, PUB-024, PUB-026, PUB-027, PUB-033, , PUB-035, PUB-036, PUB-037, PUB-039, PUB-040, PUB-041, PUB-043, PUB-045, PUB-047, PUB-020
UMSDS	Sending, Current, Previous, Incoming, Prospective	IF-021, IF-034, IF-028	PUB-008, PUB-013, PUB-014, PUB-018, PUB-022, PUB-023, PUB-024, PUB-027, PUB-033, PUB-035, PUB-036, PUB-037, PUB-040, PUB-043
MDS		IF-013, IF-014, IF-040	PUB-001, PUB-008, PUB-009, PUB-018, PUB-021, PUB-022, PUB-026, PUB-036, PUB-043, PUB-044, PUB-045, PUB-020
CS		IF-013, IF-014, IF-022, IF-023, IF-047, IF-040	PUB-008, PUB-009, PUB-018, PUB-021, PUB-026, PUB-036, PUB-043, PUB-044
LSS		IF-014, IF-022, IF-023	PUB-001, PUB-008, PUB-009, PUB-018, PUB-021, PUB-026, PUB-036, PUB-043, PUB-044
REGS		IF-001, IF-002, IF-006, IF-008, IF-009, IF-018, IF-026, IF-032, IF-033, IF-035, IF-036, IF-037, IF-039, IF-043, IF-044, IF-045, IF-020	PUB-005, PUB-007, PUB-051 (non-DIP), PUB-031, PUB-025, PUB-034, PUB-038, PUB-047, PUB-040, PUB-019
SUP	Current, Previous, Incoming	IF-024, IF-025, IF-027, IF-031, IF-041, IF-019	PUB-001, PUB-002, PUB-004, PUB-006, PUB-007, PUB-008, PUB-009, PUB-018, PUB-021, PUB-022, PUB-023, PUB-026, PUB-028, PUB-032, , PUB-035, PUB-036, PUB-037, PUB-039, PUB-041, PUB-043, PUB-044, PUB-047, PUB-020
LDSO			PUB-001, PUB-006, PUB-008, PUB-009, PUB-013, PUB-014, PUB-018, PUB-019, PUB-020, PUB-021, ,PUB-026, PUB-036, PUB-037, PUB-040, PUB-041,

			PUB-043, PUB-044, PUB-045, PUB-047
EES			PUB-001, PUB-006, PUB-008, PUB-009, PUB-018, PUB-020, PUB-026, PUB-036, PUB-037, PUB-039, PUB-040, PUB-043, PUB-044, PUB-050
Any			
Public			PUB-022, PUB-023

4 Messaging Architecture

4.1 Logical View

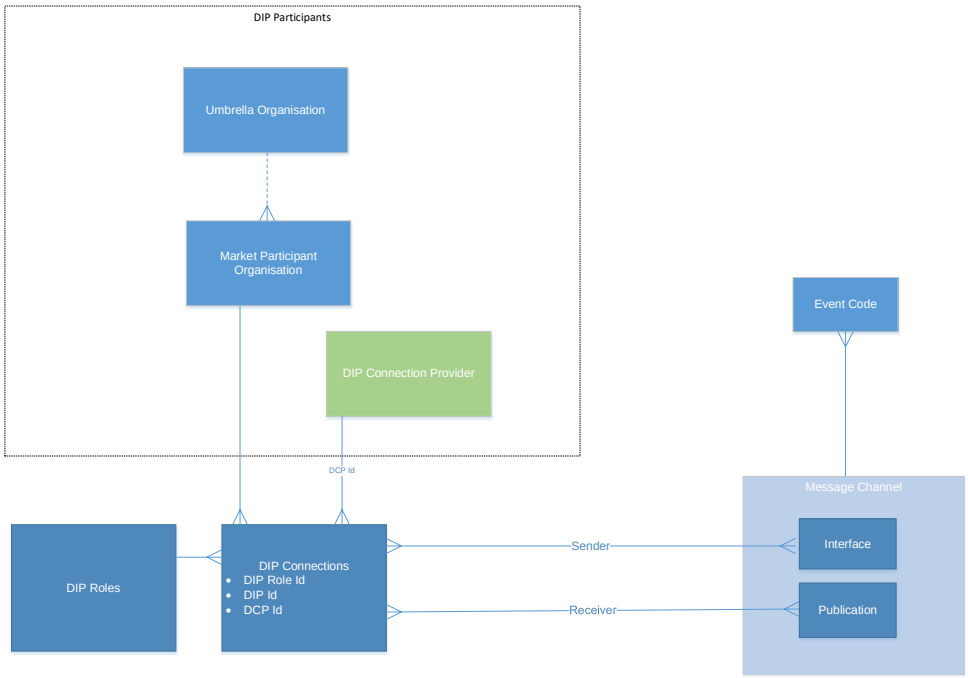


Figure 2 - DIP Workflows logical view

The ***MHHS TOM workflows*** will be orchestrated by ***Message/Event channels*** where messages are exchanged between ***Market Participants*** via the DIP. A ***Message/Event channel*** represent a message exchange between two ***Market Participants***, an inbound flow called an ***Interface*** and a corresponding outbound flow called a ***Publication***.

When on-boarded a ***Market Participant Organisation (MPO)*** will be assigned one or many~~a~~ ***DIP (Participant) IDs*** and will have one or many ***market roles or services***, depending on the roles they have registered for and the roles they are permitted to have.

For organisations that control multiple Market Participant Organisations the concept of an Umbrella Organisation exists; the Umbrella organisation provides a mechanism to logically link their different Market Participant Organisations, e.g. UKPN operates as three distinct MPOs: Eastern Power Networks PLC, London Power Networks PLC and South Eastern Power Networks PLC .

Access to individual ***Message/Event channels*** will be determined by a set of ***market roles (services)***, or possibly only a single pair of roles, that can send or receive the messages on that channel. A role will either be allowed to send or receive messages on a specific channel.

Market Participants will be responsible for managing the message/event channels determined by their ***market roles***; this will be the default configuration.

Market Participants will have different options to manage how to manage the physical delivery/sending of messages by extending the default configuration. They will be able to assign a ~~They will have the ability to define proxies for the physical sending/receiving of messages. There will be two options:~~

~~1.~~ *DIP Connection Provider* – A ***Market Participant*** may wish for an alternate party/agent, a ***DIP Connection Provider***, to proxy a specific role or all their roles ~~within the DIP~~ on their behalf. In this scenario, the ***Market Participant*** will need to assign responsibility of their access to DIP to the ***DIP Connection Provider*** by allocating the chosen role(s) to their provider. A ***DIP Connection Provider*** is a separate organisation that must be on-boarded separately as a ***Market Participant*** with a DIP Connection Provider role.

~~2. *Alternate Id* – A ***Market Participant*** may wish to separate the management of their ***market roles*** as it suits the structure of their company or their IT systems. In this scenario a Market Participant will be able to create an Alternate Id to which they can allocate specific roles. This alternate Id does not need to go through the on-boarding process, it just provides the Market Participant a method for logically separating their message channels.~~

This ability to provide different proxies for managing different channels does not alter the addressing of messages; messages will still be addressed to the primary ***DIP Participant ID*** rather than any of the proxy recipients, i.e. the addressee recipient or sender will always be the logical recipient/sender. The proxy senders/recipients, i.e. the physical senders/recipients, will require their own set of certificates in order to interact with the DIP. The DIP will be aware of the proxy recipients and will digitally sign messages appropriately, i.e. the physical sender or receiver of the message.

4.1.1 DIP & Market Participant IDs/Roles Codes

Within the MHHS TOM there are a number of business processes that are orchestrated by both DIP and DTN data flows. In order to support these business process Market Participants need a common mechanism for identification across both systems. This will be achieved by having a mapping between DIP IDs/role codes and Market Participant IDs/roles codes. Three options were considered:

1. Reuse of Market Participant Id/ Market Role Codes in DIP Participant Ids/Role Codes
2. Map Market Participant Ids to DIP Ids and map Market Participant Role Codes to DIP Role Codes
3. Map Market Participant Ids/Roles Code pairs to DIP Ids/Role Codes

The first two options were considered to be too restrictive and tied up with legacy issues around the data, and the third option provided the greatest flexibility for Market Participants and hence was the option chosen.

In practice this will work in the following way:

- DIP Participant ID and DIP Role will be used to Identify participants *within* the DIP
- DIP Addressing/Routing will utilise DIP Participant ID & DIP Role code
- ISD (entity M16) will enable a 1-2-1 mapping between each DIP Participant ID /Role combination and DTN Participant ID/Role combination
- Participants will need to utilise ISD to use DIP Participant ID/Role to identify the 'DTN delivery point' when sending D-Flows
- Conversely, ISD will need to be used to understand where D-Flows are received via DTN, what DIP Participant/Role of that party was
- The rules for mapping between the two sets of codes is still under discussion.

4.2 Message/Event Channel

Each message channel will use a standard RESTful architecture for both the inbound Interface and the outgoing publication: a Send Message~~n~~ API (https:/api.{environmment}.energydataintegrationplatform.co.uk/{version}/dip-channel/{IF-xxx}~~sendEvents~~) for incoming messages and a Receive Messages webhook ~~(receiveEvents)~~ for outgoing messages. Each message channel has both synchronous and asynchronous methods for reporting status/error messages back to the Sender.

A standard event/message channel has the following pattern. For context, the points at which the different level of validation are also shown:

Formatted: No bullets or numbering

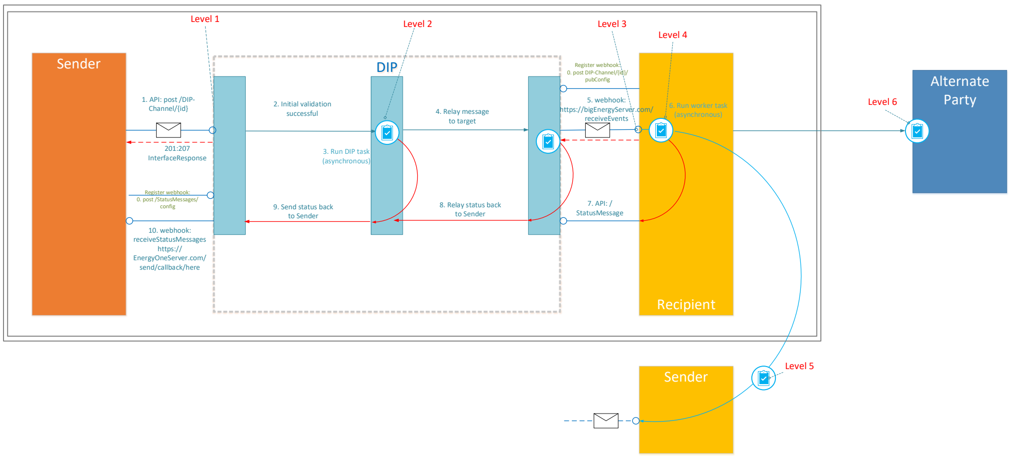


Figure 3 - Message Channel Pattern and validation levels

In a message exchange there will be 4 levels of message acceptance/validation within the message channel itself:

- Level 1 - initial synchronous validation by the DIP
- Level 2 - secondary asynchronous validation by the DIP
- Level 3 - initial synchronous validation by the Participant
- Level 4 - secondary asynchronous validation by the Participant

Then, outside of the message channel other validation may occur, levels 5 &6, which relates to the workflow being undertaken:

- Level 5 – Response through a new message channel where the original recipient becomes the new sender, e.g. IF-006 provides a response to IF-005 (see section 5.4)
- Level 6 – where the response is via a third-party, e.g. Service Appointment by the Supplier (IF-031) via the Registration Service - the outcome and if appropriate error reason code will be contained with the body of a subsequent message (IF-035).

The processing within a message channel follows the following pattern:

0. Recipient registers webhook for receipt of publication; sender registers webhook for the receipt of status messages.

~~0.~~1. Sender sends message/events to the DIP via the post /DIP-Channel/{IF-xxx} ~~sendEvents~~ API, see section 4.4~~4.4~~.

~~1.~~2. Initial receipt of events/messages by the DIP is accompanied with an auditable acknowledgement. This will include the Level 1 validation http response, ~~-~~see section 4.4.3.

~~2.~~3. The message traverses through the DIP where it undertakes internal processing where Level 2 validation also occurs, see section 4.3.

~~1.~~4. Message is relayed to the intended recipients

~~3.~~5. Onward delivery of message/events from the DIP to the recipient occurs via registered~~the receiveEvents~~ webhook with an auditable acknowledgement and Level 3 validation, see section 4.5.

~~2.~~6. The recipient will undertake further validation checks (Level 4).

~~4.~~7. If an error condition is found these are reported back to DIP via the ~~send~~StatusMessage API, see section 4.6

~~3.~~8. Both the Level 3 synchronous and Level 4 asynchronous response reported back to Sender

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 0 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 0 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 0 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 0 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

4.9. DIP relays all Level 2/3/4 responses to the Sender

5.10. Sender receives all Level 2/3/4 responses via the ~~receive~~StatusMessages webhook, see section 4.7

It should be noted:

- the DIP handles the message in two phases, a synchronous check at the API that are communicated directly back to the Sender, and then further asynchronous checks and addressing that are carried out once the message has been initially received. The rationale behind the split, is that the initial API is intended to be as lightweight as possible thereby increasing message submission speed.
-

- Where no error condition is encountered no event/message acknowledgement other than the initial API response is sent back to the Sender.
- Some business process acknowledgements, either positive or negative, are recognised as distinct message/event flows (identified in table *MHHS Interfaces Catalogue*).
- Security between the Market Participants and the DIP is ensured by securing a MTLS connection via TLS handshake (*see MHHS Code of Connection document*)
- The guideline for message processing is that on receipt of message a Market Participant should endeavour to validate the message contents as much as feasibly possible and relay all validation errors back to the Sender, i.e. not stop at the first error.
- Detail around the validation checks that are undertaken are described in the *MHHS Interfaces Catalogue.*

4.2.1 Message Structure

Each message channel is defined by a specific ***interface*** definition, which defines the incoming messages, and also a ***publication***, which defines the corresponding outgoing message. There is a direct 1:1 mapping between ***interfaces*** and **publications**.

Figure 4 –Message Channel definition – data

Each message will have two distinct components a common block and a custom block. The contents of the common block in the interface incoming message will differ from that of the outgoing message in the publication whilst the custom block passes through the DIP unchanged.

In a standard message exchange the following blocks are sent and received:

		Sender	DIP	Archive	Receiver
Interface		~~sendEvents~~ API /dip-channel/{IF-xxx}			~~receiveEvents~~ MP webhook
Header/SO	Sender	✕		◇	◇
Header/S1	Sender	✕		◇	◇
Header/AO	Address	⊠	⊠	◇	◇
Header/RO	Response	⊠	⊠	◇	◇
Header/DO	DIP		✕	◇	◇
Header/MO	MPAN	✕		◇	◇
Message Body		✕[o]		◇	◇

When a message is sent the Sender sends a message with the blocks indicated above. The blocks with a ✕ are the sole responsibility of the Sender or the DIP, whilst those ⊠ some of the fields are populated by both the Sender and DIP. ✕[o] – denotes the potential for some data fields to be obfuscated by the DIP. The message blocks written to archive and sent to the Recipient are marked ◇

The descriptions of each of the common blocks is given in the API definitions below.

The descriptions of custom block will depend on the specific interface/publication and is defined in the *MHHS Interface catalogue*.

4.2.2 Optional/Nullable Fields

Messages adopt the following convention ~~conventional JSON notation~~ for representing optional and required data ~~items~~objects ~~by default,~~ all data ~~items~~object properties are marked as required, ~~optional and~~ where data ~~items~~properties are optional then they will be marked as nullable. ~~compulsory then they will be marked as required. Also, where messages have compulsory fields but have no data then the field will be marked with a null.~~The underlying datatype for each data item will reflect the nullable property, i.e. nullable – true/false.

4.2.3 Message Traceability & Repudiation

Messages exchanged over the DIP will leave a discernible audit path as they progress from initial receipt to final delivery via the API transaction logs and the message archive. Each individual message will be uniquely identified with a Sender Unique Reference by the Sender and the DIP will provide a corresponding Transaction Id for each message that will be returned to the Sender in the API HTTP response body of the Send Messages ~~sendEvents~~ API call.

4.2.3.1 Sender Unique Reference

The Sender Unique Reference identifier for the event/message provided by the message Sender. The algorithm for setting this is provided below:

Formatted: Left, Tab stops: 1.32 cm, Left + 1.4 cm, Centered

Formatted: Superscript

S - Interface Id - Participant Id - Role Id - Date - Sequence (hex)/Participant Ref

Eg.

Interface Id	'IF-005'
Participant Id	'~~03458900823'~~0123456789'
Role Id	'SUP~~P~~'
Date	'2022-03-13'
Sequence/Ref	'12345687a1234567a ~~12345687a~~'
Hence	'S-005-012345678910090123450345890082 3-SU~~P~~P-20222313-12345687a1234567a'

It is imperative that the first part of the Sender Reference obeys the format described above so that a message can be instantly recognised and traced, i.e. Interface Id, Participant Id, Role Id and date. The final sequence/ref has only the following restrictions:

- must use standard alphanumeric characters
- needs to guarantee uniqueness for the Sender Unique Reference
- not exceed the length of the Sender Unique Reference (200 - which includes the prefix components)

The DIP will validate the Interface Id, Participant Id and Role Id only on receipt of the message, i.e. no individual check of date or sequence, however, the whole Sender Unique Reference will be checked for duplicates.

4.2.3.2 Transaction Id

Unique identifier for the event/message provided by the DIP and for the Sender it provides a technical acknowledgment for the message. The Transaction Id is added to the message (D0 block) and can be used to uniquely identify a message as it traverses through the system. Transaction Id generated by the DIP will take the form:

T - Interface Id - Participant Id - Role Id - Date - Sequence (hex)

Eg.

Interface Id	'IF-006'
Participant Id	'03458900823'
Role Id	'SUP'
Date	'20220401'
Sequence	'1234cc091234560987654abc' (hex)

'T-006-0345890082312345678 90-SUP-20220401-1234cc091234560987654abc1234c~~c~~'

The sequence component will be sequential within each message channel, however, this cannot be relied upon categorically for sequential message processing. The processing behind the APIs will be event driven serving multiple endpoints simultaneously and maintaining a genuine ordered sequence is not a requirement. The sequence will only guarantee uniqueness.

4.2.3.3 (Workflow) Correlation Id

For some MHHS business processes (defined in the *MHHS Interfaces Catalogue*) there is a requirement for each specific workflow instance to be uniquely identified and then for all messages in that workflow to reference the unique ID. This will be the Workflow Correlation ID. (A similar requirement exists in Faster Switching). This will operate as follows:

1. The party initiating the workflow will send the first message in the sequence to the DIP.
2. The DIP will recognise events/messages arriving on certain channels that need a unique Workflow Correlation ID to be generated.

The DIP will generate the Workflow Correlation ID, written to the message for onward processing and returned to the message sender as part of the HTTP response body.

The Workflow Correlation Id generated by the DIP will take the form:

CI - Business Process - Date - Sequence

Eg.

Business Process	'BP09'
Date	'20220401'
Sequence	'1234567890abc1234cc0912345698754abc 1234567890abc' (hex)

'CI- BP09-20220401-1234567890abc1234cc0912345698754abc'

4.2.4 Message Security

Some of the data flowing through the DIP will include sharing personal data and fall within the remit of the UK GDPR or confidential or sensitive information. Hence, there will be a requirement to ensure the security of the data, which should be achieved by the use of an mTLS connection between DIP and Participant. The subject of message security is covered in greater detail in the *MHHS End-to-End Security Architecture* document.

There is also a requirement for message repudiation, i.e., ensuring a message is sent from only the expected party, hence all messages sent to the DIP will be digitally signed by the Sender.

To help simplify the message privacy classification, initial analysis has identified four different security categories:

Category	Description	Message Signing
1	Public Data	Digitally signed
2	MPAN	Digitally Signed
3	MPAN + PII	Digitally Signed
4	MPAN + Consumption data	Digitally Signed

Each of the MHHS interfaces will be assigned one of these categories.

The requirement is for all messages to be signed (details to be provided in the MHHS Code of Connection document). Also, for information, the requirement for end-to-end encryption on message channels was dropped earlier in Spring 2022.

4.2.5 Service Desk Integration

When a message is received in the DIP, as part of its internal processing, the DIP will add a Service Desk URL to a message. This is a pre-emptive and will provide a URL with pre-populated parameters, where either the receiver of the original message or the response message, will be able to create a new or access an existing service desk ticket in the event on an issue with the message. It is envisaged that Market Participants will have application front-ends displaying logs of DIP messages from their own IT systems, and activating this URL will access the DIP Service Desk. Obviously, this link could equally well be activated from a DIP console showing DIP message logs. Under everyday operations this

capability is not required as the majority of messages are processed successfully, however, for the situations where an issue arises, parties will have an easy path into the Service Desk. (To be agreed during next phase of DIP design).

4.2.6 Audit Trail Integration

The proposal is for the DIP to add a URL to the message that will locate the message in the corresponding audit screen on the DIP (to be agreed during next phase of DIP design).

4.3 DIP Processing

At the physical interface the DIP will establish an encrypted mTLS connection in which both parties use X.509 digital certificates to authenticate with each other. The details of the connection requirements *see MHHS Code of Connection document*.

The DIP will also validate the digital signature of the message using the Sender's public key.

After completing the security checks the DIP will undertake the following steps:

- Message validation
- Message obfuscation
- Writing message to archive
- Addressing & Routing – forward message to each of the intended recipients - see 4.3.1 below

A full description of the DIP's internal processing is provided in the *DIP Functional Specification*.

4.3.1 Message Addressing & Routing

The DIP is responsible for sending message to the correct recipients. Within the MHHS TOM there are three recognised types of message addressing:

Name	Action	Responsibility
Targeted or Primary Routing	DIP uses the Primary recipient List in the A0 block in the message header. The Primary recipient is only known by the Sender.	Sender
MPAN Based Lookup or Secondary Routing	The MPAN core in the M0 block and optionally the Event Code and/or other fields (to determine the targeted roles) provides the lookup to the MPAN lookup address function.	DIP
Always	For a specific message channel a message is either always sent to a named participant or all participants assigned to a designated role (role is assigned to the message channel)	DIP

The DIP can apply a single or all types of addressing to a single message channel. Messages are sent to DIP Participant Id/DIP Role Codes, hence a DIP Participant can receive a message more than once if they undertake multiple roles that are the recipients for the message. However, each message will only be sent once to a ~~sinlge~~single DIP Participant Id/DIP Role Code pairing.

The Interface Id and event Code/other fields determine which type of addressing is appropriate (see routing rules table in the Interfaces Catalogue (ref *MHHS-E2E003-Physical Interface Specifications*)), if message is inappropriately addressed by the Sender, i.e. primaryRecipient is undefined when it is required, then an error message is returned.

The DIP will add the correct addresses to the Addressing block (A0) to the message before the message is archived.

4.3.2 Message/Event Obfuscation

There is the requirement ~~is~~ for the DIP to potentially obfuscate the contents of a message based on the recipient's role within a particular message channel. This requirement is implemented so that the message sender does not have to send multiple messages to different recipients containing a subset of data; instead, a single message is sent to multiple recipients. As all data items within a data block are mandatory, the ~~The~~ DIP will ~~remove~~ replace the data with the flowing values:

String values - multiple 'x' for the maximum length of the field

Date values – the date '1900-01-01 00:00' is used

Numeric values – no requirement at present

Boolean values - true

Obviously it is the responsibility of the recipient to know that they are receiving obfuscated data on a specific message channel and must not solely rely on message contents~~the required fields to those recipients where obfuscation is required; the default position will not to obfuscate, i.e. apply a filter~~.

4.4 Send ~~Events~~ Messages API

The send event API (https:/api.{environmment}.energydataintegrationplatform.co.uk/{version}/dip-channel/{IF-xxx}) is used to submit messages to the DIP. The full Open API definition can be found here: *https://app.swaggerhub.com/organizations/MHHSPROGRAMME* [2]

The message sender will need to send messages to channel specific endpoints, i.e. all messages need to be of the same IF. ~~sending Participant can send an array of messages (50,000 has been initially proposed).~~ The messages within a single transaction (API call) can have mixed MPANs.

The return code and response body provides the sending Participant with an auditable trace on the success of the receipt of the messages by the DIP.

4.4.1 Sender Responsibilities

On sending a set of events/messages the Sender will undertake the following:

Message Construction

On sending a message the Sender is responsible for construction the message header and payload as described in the section below. The use of Sender Unique Reference is described below:

Message Addressing

When a message is sent, a Participant will need to be aware of the type of addressing that needs to be applied and address the message accordingly. The type of addressing for each message channel is defined in MHHS Interfaces Catalogue.

Message Signing

Once the message has been constructed it needs a digital signature (*see MHHS Code of Connection document*) and written to the message header

API Call

The sender has the ability to send multiple event/messages in a single API. Maximum number of records sent in a single call (initial proposal is 50,000). There are no restrictions regarding sending messages on different messages channels within the same API call. This is left to the discretion of the message sender.

API Response

[2] Whilst the APIs are being designed the API server and base URL defined in the API swagger definitions are only interim values. Once the final hosting arrangements for the DIP are known then these will change to the final correct values.

The message sender will need to process the API response. The HTTP return code will provide the return for the whole transaction and needs to be considered with the response body as it will provide the details for the acceptance or otherwise for each message sent and will provide Transaction Ids for each message referenced by the Sender Unique Reference, details below.

Back Off and Retry

Participants are expected to adopt a retry with exponential back off (up to a configurable maximum wait time) if there is a failure in connecting to the DIP.

Multiple Connections

Participants will be allowed to have multiple connections to the API endpoint. Participants may wish to logically separate their own interfaces and have a single connection per interface and the DIP will support this pattern. In the detailed design phase detailed connection patterns will be described.

4.4.2 Message Structure

4.4.2.1 Common Block

The Sender will need to populate the following blocks on sending a message. The tables detail the validation the DIP will undertake on each of the message fields.

The Sender blocks are split between S0 & S1. Both are fixed formats, however, the S0 block has different enumerated values for each of the respective fields, whilst the S1 blocks does not have this separation. Consequentially, each interface definitions has a bespoke S0 block. Please see the API definitions for the next level of detail.

4.4.2.2 Sender (S0) block (mandatory)

The Sender (S0) block describes:

Field	Description	DIP Validation
Message Interface ID	The interface Id	Check valid interface Id
Event Code	Used to route messages to specific parties	Check Interface Id/Event Code pair is valid
Schema Version	Version of the schema used in the message	Check version is valid for the interface in question.

4.4.2.3 Sender (S1) block (mandatory)

The Sender (S1) block describes:

Field	Description	DIP Validation
Environment	The environment indicator, i.e. PROD, PREPROD, SIT, UIT, DEV	Check environment is in allowed list of values. Check environment is correct for the current instance.
Sub Text	Optional field used in testing to uniquely identify a set of messages	n/a
Sender Unique Reference	Unique identifier for the event/message provided by the message Sender. See note below that provides guidance on setting this field	Check sender unique reference obeys format specified below
Sent Timestamp	Date/time (UTC) message was sent	Check valid/date, i.e. complies with RFC-3339
Sender ID	The Market Participant who sent the message.	Check valid participant
Sender Participant Role	The capacity in which the message was sent	Check Sender ID/ Sender Participant role

Field	Description	DIP Validation
		is entitled to send message
DIP Connection Provider ID	If message sent by a nominated third-party	Check DIP CP ID/ Sender Participant Role is authorised to send messages on behalf of Sender ID
Sender Correlation Id	Populated when the IF is used in the response to the primary message in the workflow.	Check obeys format

The S1 is mandatory for all messages.

4.4.2.4 *Response (R0) block (optional)*

Field	Description	DIP Validation
Response Code	Response Code	n/a
Response Message	Response Message	n/a

The response block is only used when the message is a "response" to an initial message, e.g. IF-006 is a response to an IF-005 message (see *MHHSP-DES138 -Interface Catalogue* for the interfaces that require a R0 block), and the Response Correlation Id is that sent in the original message – see section 5.4.

4.4.2.5 *Address (A0) block (optional)*

Field	Description	DIP Validation
Primary Recipient Id(s)	Used to route messages to targeted parties (Primary Addressing)	Check participant has the corresponding role (permission) to receive message

The A0 block is only populated on an interface-by-interface basis, i.e. not all Interfaces require primary addressing. The interface catalogue (*MHHSP-DES138-Interface Catalogue*) has the corresponding details.

4.4.2.6 *Message Body MPAN (M0) block (optional)*

Field	Description	DIP Validation
MPAN Core	Used to route messages to specific parties (Secondary Addressing)	
Distributor Id	The market-wide unique reference for the distributor who is responsible for the distribution network that a metering point is connected to.	
GSP Group Id	Identifies the distinct grid supply point group (physical region of the country) where the metering point is located.	Check valid GSP Id (enumerated values)

The M0 is an optional block

4.4.2.7 *Custom Block*

The message body will be dependent which interface is being submitted. These definitions are not repeated in this document, instead they can be found in the swagger definitions and *MHHS-E2E003-Physical Interface Specification*.

4.4.3 DIP Processing (Level 1 Validation)

The purpose of the initial API checks is provide basic syntax checking of the message header to ascertain whether the message is syntactically correct and can be processed further. A 'handshake' return message provides a reference between the message sender's unique reference and the DIP's transaction Id.

The DIP will perform the following for each separate message received:

- Validation of API key and establish Sender
-
- Check message structure~~, i.e. mandatory fields are present for the corresponding interface~~. If schema validation fails on a single message within a transaction then all messages within the transaction are rejected with a '400' response
- Check message structure, i.e. mandatory blocks are present for the corresponding interface and event code
- Check the message validation checks for the common blocks (see message validation above in 4.4.2 for details):
    - Check the environment is set correctly
    - Check Interface Id is set correctly
    - Check Interface Id/Event Code pair is an allowable combination
    - Check the message Schema version aligns (API enumeration defines allowable values)
    - Check the Sender Id ("logical" sender) is entitled to send message for the Event Channel for the given role
    - Check the DIP Connection Provider is entitled to send message on behalf of the "logical" sender for the given role/channel
    - Check Sender Unique Reference is formatted correctly.
- Generate a Transaction Id (see below)
- Generate a Transaction timestamp, i.e. record the current date/time (UTC)
- If the message channel is configured to generate a Workflow Correlation Id (see below)
- Generate service desk/audit links
- Write the Transaction Id, Transaction timestamp, Workflow correlation Id, status code, status message and audit/service desk links to the response body

The DIP will not be enforcing any capitalisation checks on message ingress for path names and message structure. However data items within messages are case sensitive and the DIP will not convert between cases: data will pass through unaltered. Enumeration checks undertaken by the DIP are case sensitive, e.g. for the CommonBlock/S1/environment the term 'PROD' is accepted, whilst 'prod' or 'Prod' are rejected. A significant amount of legacy data items where case is important pass through the DIP and hence the reason why these checks are applied).

Generate return code based on the results of the above checks for the all the messages received in the single transaction.

4.4.4 Return Code and Response Body

4.4.4.1 Return Code

Each connection will result in a HTTP return code that will indicate the success or otherwise of the complete transaction.

Return code	Meaning

202 – Accepted	All messages accepted
207 – Multi status	Some messages accepted; see response body for details
4xx, 5xx	Other error responses due to bad request – no messages sent

The ~~types of response that will generate a 4xx will be picked up in the detailed design phase~~full list of response codes is available in the swagger definition.

4.4.4.2 Response Body

The response uses the common Standard Response body:

Field	Description
Transaction Id	Unique DIP transaction Id
~~Transaction~~ Sent timestamp	DIP Receipt timestamp
Sender Unique Reference	The original Sender Unique Reference
Sender ID	DIP identified
Correlation Id	Correlation ID relayed back (optional)
Recipient Id	Recipient of the message
DCP Id	DCP (optional)
Status Code	Message status code
~~Status~~ Message	Information on the message status
Help	Extra help information
Service Ticket URL	~~To be picked up during next design phase~~URL to SNOW ticket related to the message
~~Audit Trail URL~~	~~To be picked up during next design phase~~

The response body of the HTTP call will deliver the Sender a transaction ID and optionally a correlation ID against the Sender's Unique Reference for each message.

In the example below, two messages are sent. The multi-status response code 207 is return as the first rejected and the second accepted:

```
HTTP/1.1 207 Multi-status
Content-Type: application/sendEvent.api+json
{ "dipchannel~~sendEventapi~~": { "version": "1.0" },
{
 "messageArray": [
  {
   "transactionId": "T-006-1234567890123-SUP-20220401-1234CC0123456789",
   "senderUniqueReference": "S-005-03458900823-SUP-20222313-12345687A",
   "correlationId": "CI-20220401-1234567890123abce123092",
   "sentTimestamp": "2022-03-21T19:05:00+00:00",
   "senderId": "10000000",
```

Formatted: Font: (Default) Calibri Light

```
    "recipientId": "1009012345",
    "DIPConnectionProviderId": null,
    "message": "MSG0101 Application 101 error message",
    "help": "Sender Unique reference must take the format S - Interface Id - Participant Id - Role Id - Date - Sequence (hex)/Participant Ref, e.g. 'S-005-03458900823-SUP-20222313-12345687a'",
    "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id=-1%26sysparm_query=<field=value>"
  },
{
    "transactionId": "T-006-1234567890123-SUP-20220401-1234CC0123456123",
    "senderUniqueReference": "S-005-03458900823-SUP-20222313-12345687b",
    "correlationId": NULL,
    "sentTimestamp": "2022-03-21T19:05:00+00:01",
    "senderId": "10000000",
    "recipientId": "1009012345",
    "DIPConnectionProviderId": null,
    "message": "MSG0000 oK",
    "help": "Sender Unique reference must take the format S - Interface Id - Participant Id - Role Id - Date - Sequence (hex)/Participant Ref, e.g. 'S-005-03458900823-SUP-20222313-12345687a'",
    "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id=-1%26sysparm_query=<field=value>"
  }
 ],
 "timestamp": "2022-03-21T19:05:00+00:00"
}"events": [
{
  "transactionId": "T-006-1234567890123-SUPP-20220401-1234CC",
  "sentTimestamp": "2022-03-21T19:05:00+00:00",
  "senderId": "0123456789",
  "recipientId": "1009012346",
  "DIPConnectionProviderId": "1009012345",
  "message": "DIP0000 ok ",
  "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id= 1%26sysparm_query=<field=value>"
},
{
  "transactionId": "T-006-1234567890123-SUPP-20220401-1234DE",
  "sentTimestamp": "2022-03-21T19:05:01+00:00",
  "senderId": "0123456789",
  "recipientId": "1009012346",
  "DIPConnectionProviderId": "1009012345",
  "message": " DIP1003 - Sender Role not permitted to send messages on this channel"
  "help": " DIP1003 - Sender Role not permitted to send messages on this channel",
  "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id= 1%26sysparm_query=<field=value>"

}
```

}

}

As all API activity is logged audit reporting facilities will be made available to allow participants to check on the progression of individual message exchanges using the references described below:

4.5 Receive MessagesEvents Webhook

The Rreceive MessageseEvents webhook is used to relay publication events/messages to the DIP. The latest version of the full Open API definition can be found here https://app.swaggerhub.com/apis/MHHSPROGRAMME/SubmitEvents/1.00.2.

Formatted: Default Paragraph Font

The API call can either register a new webhook, remove or replace an existing webhook.

The receiving Participant has the option of defining multiple callbacks separated by Publication Id. One callback can receive all, many or a single publication(s). It is the receiving Participants responsibility to ensure that all publications they are due to receive are covered by all the callbacks registered.

The messages within a single transaction (API call) can have mixed MPANs. The callback can define the maximum number of messages the recipient can receive in a single transaction.

The return code and response body of the callback API will provide the receiving Participant with an auditable trace on the success of sending the messages by the DIP.

If the webhook request has originated from a DIP Connection Provider the DIP will check that the DIP Connection Provider is authorised to receive messages on behalf of the logical recipient.

4.5.1 DIP Responsibilities

On sending a set of events/messages the DIP will undertake the following:

Message Construction

The DIP will augment the original message with the added information detailed below in section 4.5.2.

API Call

The DIP will call the API declared in the callback. Initially the maximum number of records sent in a single call will be set to 50,000, however, this value will be under review. There are no restrictions regarding sending messages on different messages channels within the same API call. This is left to the discretion of the message sender.

API Response

The DIP will need to process the callback response. The HTTP return code will provide the return for the whole transaction and needs to be considered with the response body as it will provide the details for the acceptance or otherwise for each message (see below for response code).

Back Off and Retry

The DIP will adopt a retry with exponential backoff approach (up to a maximum wait time) and introduce additional “circuit breakers” to ensure efficient handling of broken connections to Market Participants.

4.5.2 Message Structure

4.5.2.1 Common Block

The common block described above in the sendEvents API will be augmented with the D0 block which is added to the message by the DIP.

4.5.2.2 Interface (S0) block (Mandatory)

Contains specific information regarding the message channel of the message.

Field	Description	Validation
Message Interface ID	The interface Id	Check valid interface Id
Schema Version	Version of the schema used in the message	Check version is valid for the environment
Event Code	Used to route messages to specific parties	Check Interface Id/Event Code pair is valid

4.5.2.3 *Sender (S1) block (Mandatory)*

Contains Sender specific message information.

Field	Description	Validation
Environment	The environment indicator, i.e. PROD, PREPROD, SIT.	Check environment is correct
Sub Text	Optional field used in testing to uniquely identify a set of messages	n/a
Sender Unique Reference	Unique identifier for the event/message provided by the message Sender. Setting is this value is described below.	Check format meets specification
Sent Date/time	Date/time (UTC) message was sent	Check valid/date
Sender ID	The Market Participant who sent the message.	Check valid participant
Sender Participant Role	The capacity in which the message was sent	
DIP Connection Provider ID	If message sent by a nominated third-party	

4.5.2.4 *DIP (D0) block (Mandatory)*

The DIP adds to the D0 block to the common block.

Field	Description
Transaction Id	Unique DIP transaction Id -
Transaction timestamp	DIP Receipt timestamp
Publication Id	Publication ID
DIP Correlation Id	Provides unique identification for a specific workflow instance. Only set for those channels were a correlation ID is required – see section 3.4
Replay Indicator	If set, then message has arrived through a replay request rather than a standard message
Service Ticket URL	URL to create/view appropriate service ticket

4.5.2.5 *Response (R0) block (optional)*

Sender populated, dependent on message channel (see *MHHSP Interfaces Catalogue)*

Field	Description	Recipient Validation
Response Code	Response Code	Check response code is valid for the Interface

Field	Description	Recipient Validation
Response Message	Status/Error Message	

4.5.2.6 *MPAN (M0) block (optional)*

Field	Description	Recipient Validation
MPAN Core	Used to route messages to specific parties (Secondary Addressing)	
Distributor Id	The market-wide unique reference for the distributor who is responsible for the distribution network that a metering point is connected to.	
GSP Group Id	Identifies the distinct grid supply point group (physical region of the country) where the metering point is located.	Check valid GSP Id (enumerated values)

Formatted Table

4.5.2.7 *Custom Block*

The message body will be dependent which interface is being submitted. These definitions are not repeated in this document, instead they can be found in the swagger definitions and *MHHSP Interfaces Catalogue*.

4.5.3 Recipient Responsibilities

On receipt of an event/message the Recipient will need to undertake message validation. The validation is split into two: Synchronous (Level 4) and Asynchronous (Level 5) validation. The validation required is described further in section 5.1. Other checks that recipient systems will need to undertake are described in the *MHHSP Interfaces Catalogue*.

If the Recipient is receiving the event/message through a replay request then the subsequent downstream processing and subsequent reply needs to be cognisant that the event/message may have already been processed.

4.5.4 Callback Response

The callback response contains the outcome of the Level 4 validation undertaken by the recipient, the return code the result of the overall transaction and the response body details of the individual messages/events..

~~*4.5.4.1 Return Codes*~~

~~*4.5.4.1 Each connection will result in a HTTP return code that will indicate the success or otherwise of the complete transaction. N.B. this is the return code for the overall transaction, the response to the individual messages are provided in the response body.*~~*Return Code*

Each connection will result in a HTTP return code that will indicate the success or otherwise of the complete transaction.

Return code	Meaning
202 – Accepted	All messages accepted
207 – Multi status	Some messages accepted; see response body for details
4xx, 5xx	Other error responses due to bad request – no messages sent

The full list of response codes is available in the swagger definition.

4.5.4.2 Response Body

The response uses the common Standard Response body:

Field	Description
Transaction Id	Unique DIP transaction Id
Sent timestamp	DIP Receipt timestamp
Sender Unique Reference	The original Sender Unique Reference
Sender ID	DIP identified
Correlation Id	Correlation ID relayed back (optional)
Recipient Id	Recipient of the message
DCP Id	DCP (optional)
Status Code	Message status code
Message	Information on the message status
Help	Extra help information
Service Ticket URL	URL to SNOW ticket related to the message

Where a field is not written/available/optional, then a null value needs to be written.

~~Response~~	~~Meaning~~
~~202 – Accepted~~	~~All messages successfully accepted~~
~~207 – Multi status~~	~~Some messages processed with conditions; see response body for details~~
~~4xx~~	~~Bad request – no messages received~~

~~4.5.4.2~~4.5.4.3 Response Body

~~Field~~	~~Description~~	~~Mandatory/ Optional~~
~~Transaction Id~~	~~Unique DIP transaction Id~~	~~M~~
~~Transaction timestamp~~	~~DIP Receipt timestamp~~	~~M~~
~~Status Message~~	~~Information on the message status~~	~~M~~
~~Service Ticket URL~~	~~To be picked up during next design phase~~	~~O~~
~~Audit Trail URL~~	~~To be picked up during next design phase~~	~~O~~

The response body of the HTTP call will deliver a response back to DIP indicating the success or otherwise of processing each message received.

In the example below, two messages are received, the first accepted and the second rejected:

```
HTTP/1.1 207 Multi-status
Content-Type: application/recieveEventCallback+json
```

```
{ "recieveEventCallback": { "version": "1.0" },
"events": [
{
{
  "transactionId": "T-006-1234567890123-SUPP-20220401-1234CC",
  "sentTimestamp": "2022-03-21T19:05:00+00:00",
  "senderId": "1009012345",
  "recipientId": "1009012346",
  "message": "DIP0000 ok ",
  "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id=-1%26sysparm_query=<field=value>"
 },
{
  "transactionId": "T-006-1234567890123-SUPP-20220401-1234DE",
  "sentTimestamp": "2022-03-21T19:05:01+00:00",
  "senderId": "1009012345",
  "recipientId": "1009012346",
  "message": " REG1020 Meter Install Date [DI-057] cannot take place in the future"
  "help": " Meter Install Date [DI-057] cannot take place in the future ",
  "serviceTicketURL": "https://<baseURL>/nav_to.do?uri=<table name>.do?sys_id=-1%26sysparm_query=<field=value>"

 }
]
}
```


4.6 Send Status Messages API

The Send Status Messages API is used to submit status messages, i.e. the results of level 4 validation, to the DIP. The latest version of the full Open API definition can be found here https://app.swaggerhub.com/apis/MHHSPROGRAMME~~/SubmitEvents/0.4~~

The sending Participant can send an array of status messages (configurable – initial limit set to 50,000).

The return code and response body provides the sending Participant with an auditable trace on the success of the receipt of the messages by the DIP.

4.6.1 Status Message Structure

The format of the incoming message has the structure:

Field	Description	Mandatory/ Optional
Transaction Id	Transaction Id of the original message the DIP	M
Transaction Timestamp	Timestamp the receipt of the original message	M
Sender Id	Logical Sender of the original message	M
Recipient Id	Receiver of the message	M
DIP Connection Provider Id	Physical sender of message (where different to Sender id)	M~~O~~

Field	Description	Mandatory/ Optional
Message	Information regarding subject of message	M
Service Ticket URL	URL to create/view appropriate service ticket	O
Help	Help text	O

4.7 Receive Status Messages Webhook

The Receive Status Messages API is used to receive status messages from the DIP that either been produced by the DIP, or the DIP is passing through from another Participant. The latest version of the full Open API definition can be found here https://app.swaggerhub.com/apis/MHHSPROGRAMME ~~https://app.swaggerhub.com/apis/MHHSPROGRAMME/SubmitEvents/1.10.4~~[3]

Field Code Changed

The Participant webhook will be able to receive an array of status messages (up to 50,000).

The return code and response body provides the sending Participant with an auditable trace on the success of the receipt of the messages by the DIP.

The message structure is the same as defined in section 4.6.1.

4.8 Replay Events API

The Replay Events API is used to by Participants to retrieve messages from archive. The full Open API definition can be found here: tbd.

The furthest a query can retrieve data is dependent upon the retention time set with each message channel (the default retention is 2 years).

The Participant can make a single request to return an array of events (50,000).

4.8.1 Query Parameters

The Replay Events query parameters are:

- Publication ID
- Event Code (optional)
- MPAN (optional)
- Start Message Transaction ID – the message from which to start the replay sequence from, or
- Date/time from – transaction time from which the first message needs to be replayed.
- End Message Transaction ID (optional) – the last message from which to start the replay sequence from, or
- Date/time to (optional) – transaction time message to be replayed

4.8.2 Message Structure

[3] Whilst the APIs are being designed the API server and base URL defined in the API swagger definitions are only interim values. Once the final hosting arrangements for the DIP are known then these will change to the final correct values.

The replayed message will have the same structure as the message received via the webhook, hence in addition to the message body and Sender message blocks (P0, M0) (sender) and message will also have the added D0 block and amended T0 block.

5 Message Choreography

5.1 Simple Message Exchange

The following diagrams describe the 6 different scenarios that can occur on a simple message exchange between a single DIP Participant service and the DIP and include all the different levels of validation.

For diagram clarity, the step showing message being written to the DIP archive are not shown on any of the diagram below.

The diagrams only show a simple scenario where a single message is exchanged between Sender and the DIP, and then relayed by the DIP to recipient. Where multiple messages are exchanged and a mixed response is described in section 5.3

5.1.1 Standard Exchange

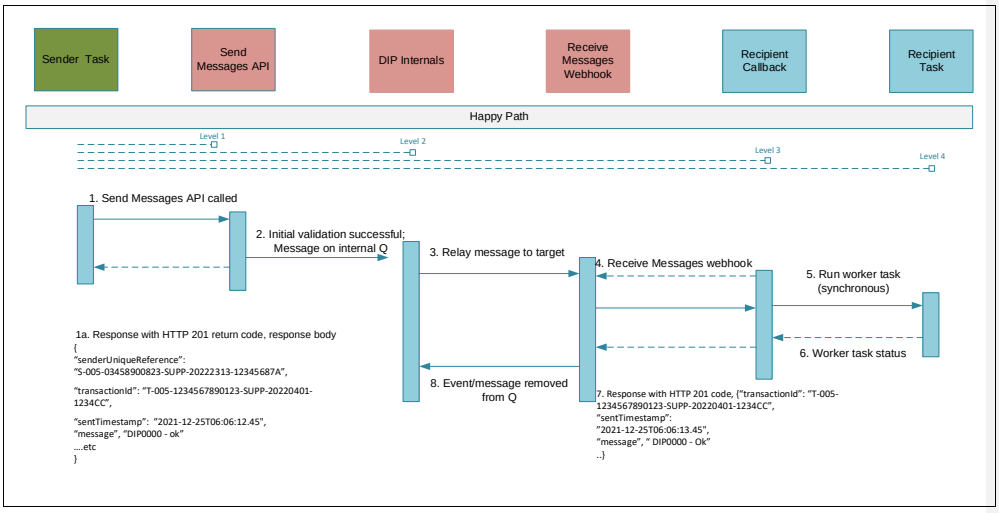


Figure 5 – Standard Exchange

In this exchange the message recipient will only receive the initial synchronous acknowledgement from the DIP (Level 1 validation), there is no acknowledgement from the recipient of the message back to the Sender. However, there is an acknowledgement from the Recipient to the DIP (Level 3 validation) for the message but this is not relayed back to the Sender. This follows the principle that only processing and data errors are returned to the Sender, and successful processing is not routinely reported.

Both synchronous responses, initially by the DIP on the initial API call (steps 1&1a above) and the webhook API call back (steps 4&7) are logged in the DIP to meet the audit requirements.

Message auditing logging is not shown. The audit trail is available to sender via a DIP report.

5.1.2 Level 1 validation - DIP Synchronous rejection

In the scenario where a message is synchronously rejected by the DIP, in the example below the message has been rejected as the Sender does not have the privilege to send messages on Interface IF-006, the response body clearly states the sender is not allowed to send messages on IF-006.

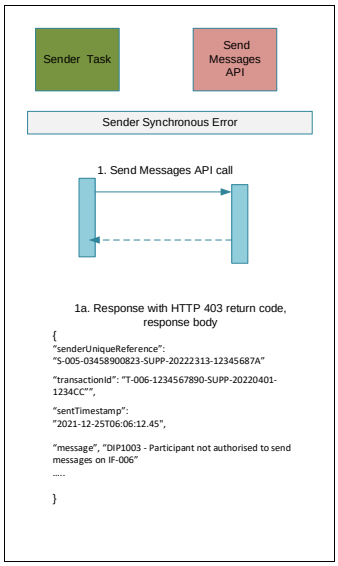


Figure 6 – Level 1 DIP Synchronous Error

5.1.3 Level 2 validation - DIP Asynchronous rejection

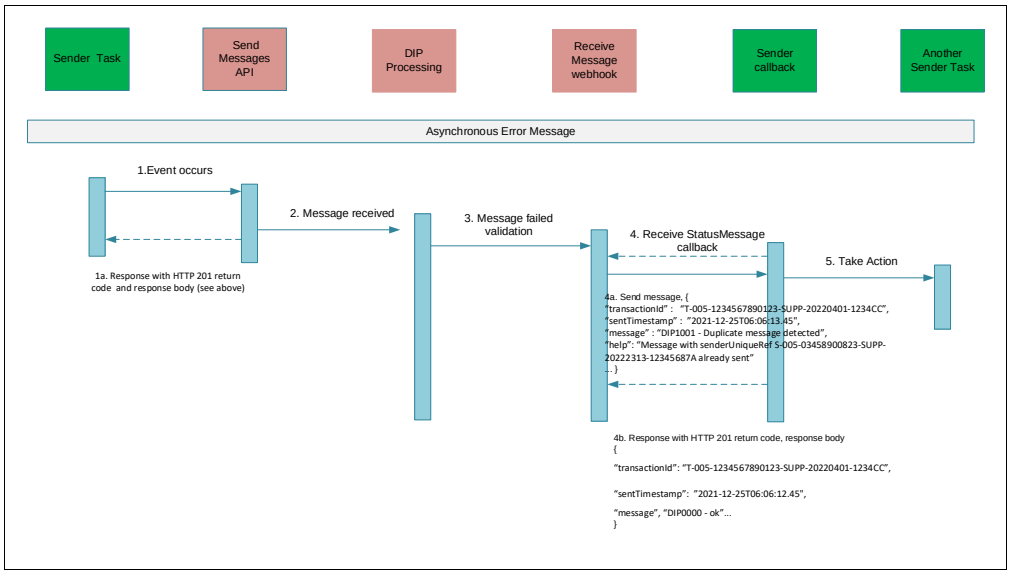


Figure 7 Level 2- DIP Asynchronous Error

After the initial successful receipt of the message the DIP encounters an error with a Sender's message, in this example the DIP finds the signature blocks invalid. The DIP informs the Participant of the problem, a message is sent back via the ~~send~~Status Message webhook, with return message containing the Transaction Id, an error code and a description of problem. The webhook call back responds that the status message has been received and the sender will need to take some action accordingly.

5.1.4 Level 3 response - Recipient Synchronous Error (with retry)

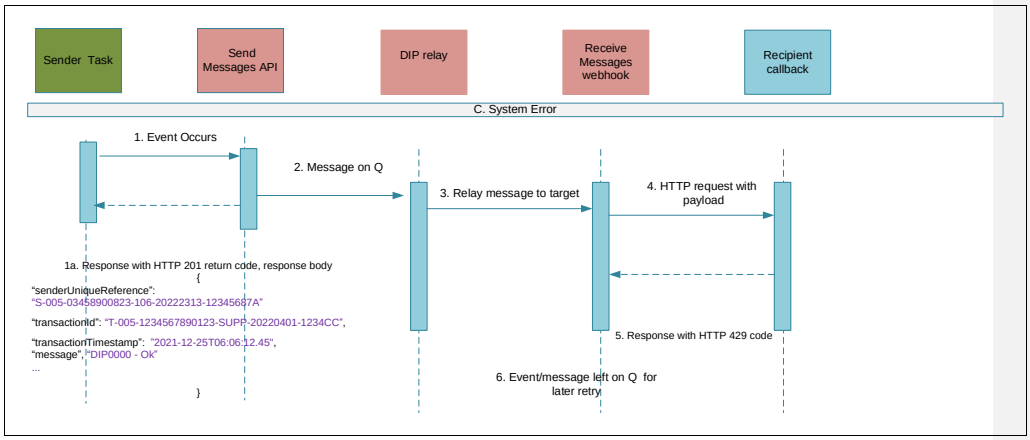


Figure 8 – Level 3 - Recipient Synchronous Error (retry)

Recipient Synchronous Error with retry where the response from the recipient webhook indicates a system error, e.g. with an HTTP 429 Too Many Requests response. The message has not been consumed and is retained in the queue for later processing. The DIP will have the logic to attempt to resend the message after a timeout period.

If the message is still not sent after a maximum retry period has elapsed then the message is moved to the Dead Letter Queue (DLQ) – see section 5.1.8.

5.1.5 Level 3 validation - Recipient Synchronous Response (no-retry)

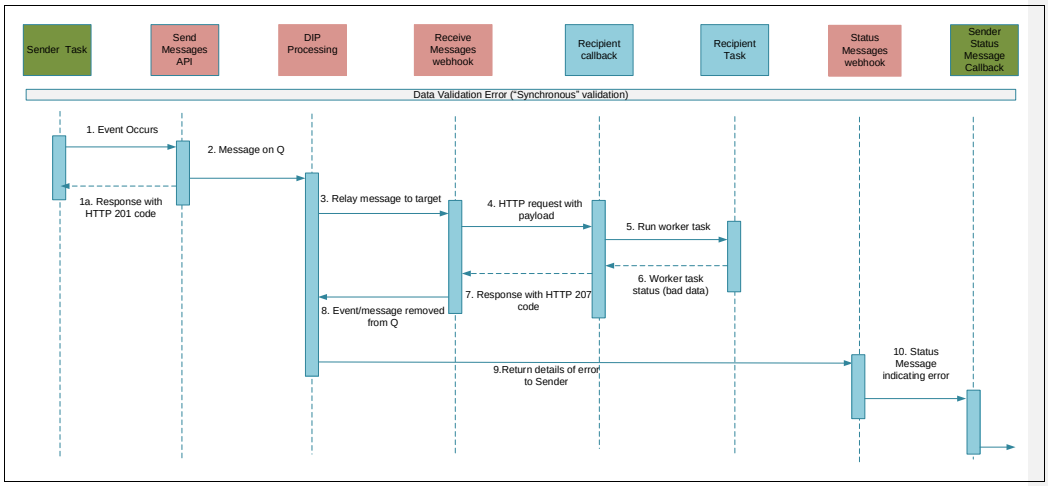


Figure 9 - Recipient synchronous Error (no-retry)

Recipient Synchronous Error with no-retry is the case where the recipient's system rejects the message via the webhook call, i.e. the message has been consumed and rejected. In this scenario, the webhook reports a mixed/error return, and the DIP relays the error received from the R~~r~~eceive Messages~~Events~~ webhook back via the ~~receive~~Status Messages webhook to the originating party and the message is removed from the outbound queue.

5.1.6 Level 4 validation - Recipient Validation Response (Asynchronous)

In this scenario the recipient of the message discovers an error with the message and needs to report the response back to the Sender. i.e. the recipient's system can consume the data, however, there is an inconsistency with the data which is not reported on the initial call back. The initial message exchange is identical to the standard use case described above (see 5.1.1), however there is an additional message sent back via the Send Status Message API.

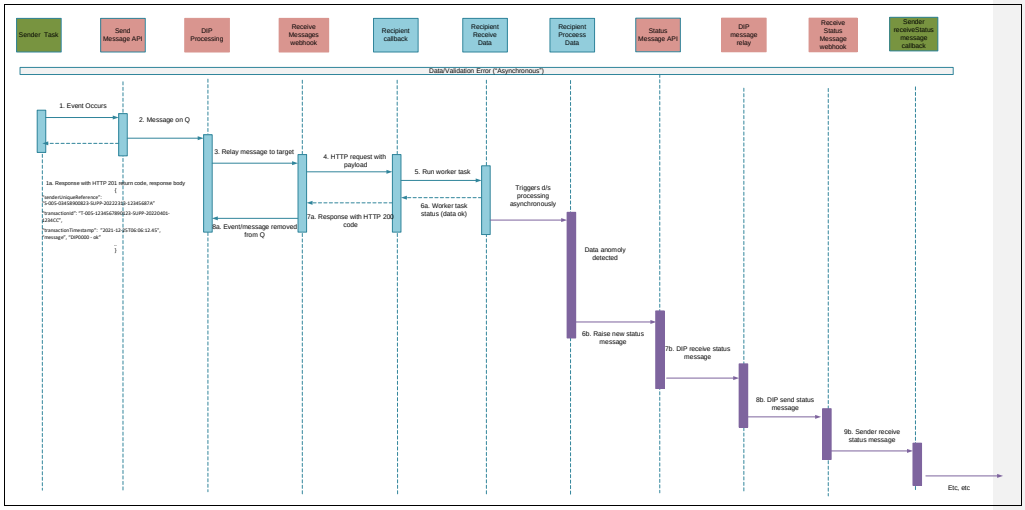


Figure 10 - Recipient Data Error - Asynchronous reporting

5.1.7 Pattern 'B' Message flow

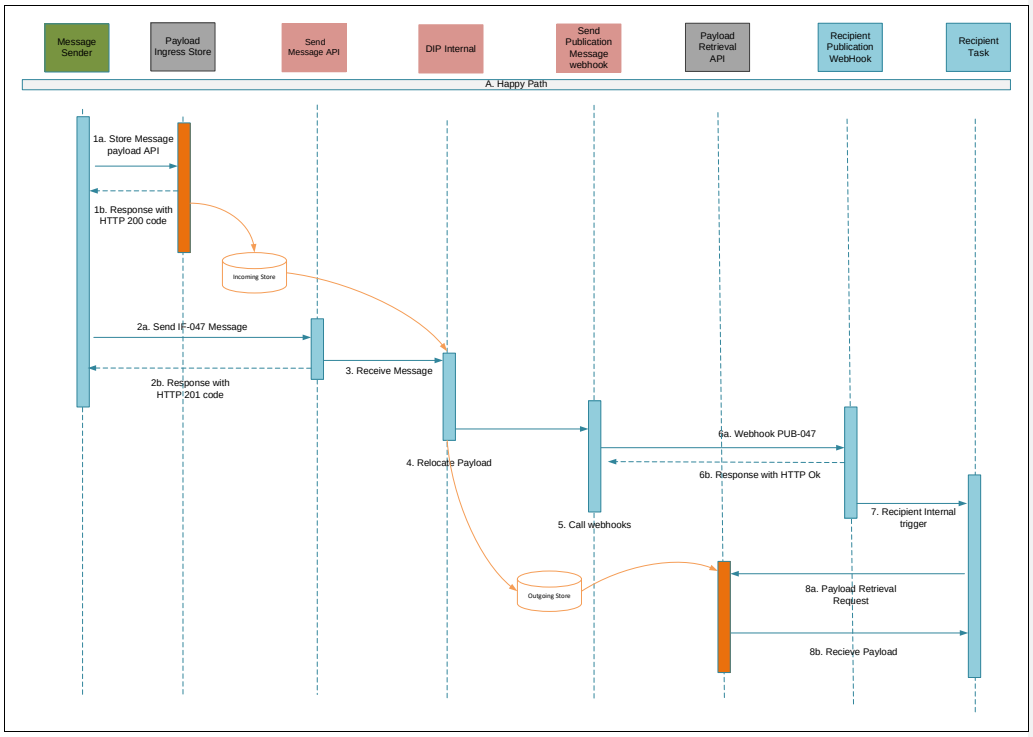


Figure 11 - Message Pattern 'B' Orchestration

In the DIP Functional Specification (*Reference MHHSP-DIP001-Functional Specification v2.0, 6 May 2022)* the concept of message pattern 'A' and Message Pattern 'B' was introduced. ~~This document only discusses a single messaging pattern. It is recognised that for the most part, the proposed pattern 'A' and Pattern 'B' are effectively the same, except that P~~pattern 'B' is a~~n extension of special case of~~ pattern 'A' where the payload is initially written to a URI, and then the intended recipients of the payload are sent a message via the IF-047 interface to communicate the location of the payload to Market Participants. There are two distinct variants for Pattern 'B': DIP hosted payloads (where access controls are required) and public payloads (no access controls required).

The orchestration sequence for Pattern B (DIP hosted) is presented above.

1. Sending MP copies payload to storage account on DIP; response message details the location of the payload
2. Sending MP writes an IF-047 message to the DIP with the details of the uploaded payload.
3. DIP receives message
4. DIP moves payload to destination folders for consumption by receivers
5. DIP calls recipient callback with PUB-047 message
6. Recipient webhook receives PUB-47 message and issues acknowledgment
7. Recipient internal trigger activate payload retrieval API

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

8. Receiving MP retrieves payload from storage account

With the public payload version, the first step is omitted, and it's the responsibility of the Sending MP to host the message payload and provide the required access.

~~contains a URI of a remote resource.~~

~~The payloads in Pattern 'B' are separated from the messages as they are not deleted once the message has been read; instead, they are available for later consumption.~~

Market Participants will also query the data store (identified by a URI within the message contents) to see current and past payloads. This will also enable anonymous access to the data store, and a facility to download current and past payloads is also available.

The ingress and egress files will be held as blobs within Azure storage accounts. Authorisation ~~The orchestration sequence for Pattern B is presented above.~~to the accounts will be achieved via the use of the same client certificates that are used for message exchange. The transfer of files from Market Participant systems to the DIP storage account will be achieved via the use of the azcopy utility.

5.1.8 Recipient Timeout & Dead-Letter Queue Handling

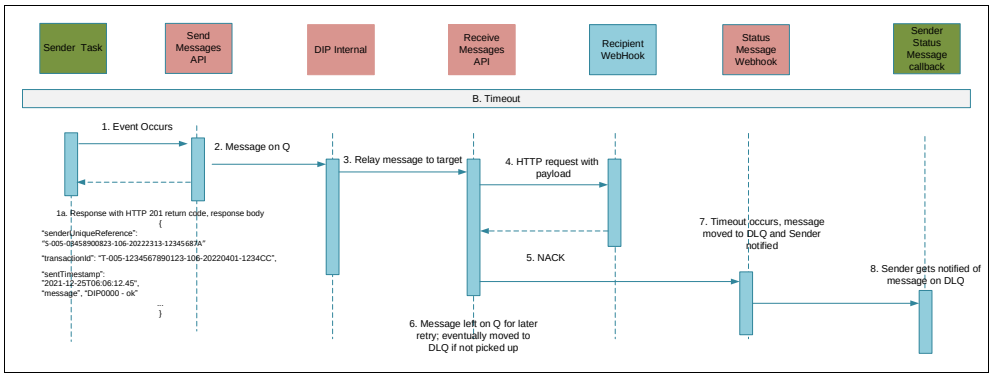


Figure 12 – Recipient Timeout

Timeout – above shows the case where there has been no response from the recipient webhook at the call back times out. The message is retained in the queue for later processing. The DIP will have the logic to attempt to resend the message after a timeout period.

Once a message/event has not be retrieved by a Recipient the DIP will write the message to a dead letter queue (DLQ). Messages on the DLQ will be reported back to the message's sender via the ~~receive~~Status Message webhook (section 4.7.)

5.2 Error Handling & Message Distribution Patterns

This section describes the different message distribution patterns together with some general principles in message validation and how potential discrepancies between recipients are resolved. This is just an initial view, more details on this will be provided in the *DIP Operational Handbook* once the detailed design has progressed.

The majority of the DIP message flows originate from either the Registration system(s) or the ECS systems (MDD, LSS, ISD) and the principle adopted is that these systems are considered to be the "System of Truth" in the message exchanges, and hence discrepancies are raised by DIP Participants that either send or receive messages to/from these services.

Most of message exchanges via the DIP do not involve a simple message being sent from a sender to a single recipient. Depending on the addressing requirements of each message channel (see *MHHS-E2E003-Physical Interface Specifications* for the requirements for each interface) messages are often sent to multiple recipients. This asks the question regarding message validation and what happens in the event of a discrepancy between message recipients.

All the patterns assume that the original message has been consumed by one or more recipients, and the recipient has sent a Status Message back to the Sender via the DIP detailing the issue with the data.

	Pattern	Description	Interfaces	Validation resolution
MD#1	Registration In	A single DIP Participant sends a message solely to the Registration Service	IF-005, IF-007, IF-025, IF-031, IF-034, IF-038	Registration will send validation errors back to the Sender. Sender will aim to correct data and resend; if Sender considers the data to be correct then raise support ticket on Registration.
MD#2	Registration Out	The registration service sends out a broadcast message, typically a confirmed update from the Registration system, to many DIP Participants	IF-001, IF-002, IF-006, IF-008, IF-009, IF-018, IF-026, IF-032, IF-033, IF-035, IF-036, IF-037, IF-039, IF-043, IF-044, IF-045, IF-050	Registration data is considered correct. If participant interface raises an error, they will need to track the error via the DIP. Check if other DIP Participants have raised an error on the message. If sender still considers there is an issue with the data then raise a support ticket on Registration (or update existing ticket)
MD#3	ECS Out (LSS, MDS, ISD, VAS)	One of the ECS services (LSS, MDS, ISD, VAS) sends out a broadcast message to multiple DIP Participants	IF-013, IF-014, IF-022, IF-023, IF-040, IF-047	LSS, MDS, VAS, ISD data are considered correct. If participant raises an error, they will need to track the error via the DIP. Check if other DIP Participants have raised an error on the message. If sender still considers there is an issue with the data then raise a support ticket on LSS/MDS/VAS/ISD (or update existing ticket)
MD#4	IF-021	The Half Hourly consumption data interface is considered separately	IF-021	With IF-021 the default position is that the responsibility will be for the SDS to send correct data. Hence, if an error message is received from MDS then the SDS must take initial responsibility to trace the issue. If IF-021 is rejected by another party than the SDS then the onus is on that party to pursue any action, i.e. raise a support ticket, noting they should establish whether the MDS successfully received the same data via the DIP audit trail.

MD#5	Point-to-Point	Messages between two single DIP Participants	IF-024, IF-027, IF-028, IF-004	With point-to-point interfaces the onus will again be on the recipient to raise any issues with errors detected in the message exchange.
MD#6	Request	Participant sends a data request to the DIP (normal BAU, not event replay)	IF-015	In the event of an error being returned by the DIP, the onus is on the Sender to raise a support ticket if they see no issues with the original request.
MD#6	Response	Participant receives a data request to the DIP (normal BAU, not event replay)	IF-016	In the event of an error being returned by the DIP, the onus is on the Sender to raise a support ticket if they see no issues with the original request.

5.3 Batch Message Handling

For ease of understanding the examples in section ~~0~~5.1 describe the processing of individual messages rather than a group of messages. In practice participants will send batches of messages rather than individual messages via the Send Messages~~sendEvents~~ API and the callback from receive ~~e~~Events webhook will routinely send multiple messages within a single HTTP transaction. In this scenario the course of individual messages will follow the sequences above, however, as each connection can only result in a single HTTP return code that will indicate the success or otherwise of the complete transaction and this be reflected as follows:

Response	Meaning
202 – Accepted	All messages created
207 – Multi status	Some messages created; see response body for details
4xx/5xx~~03/4xx~~	See swagger definitions~~Bad request – no messages sent~~

Figure 13 - HTTP Response Codes for Batch Messages processing

5.4 Workflow Message Handling

The message flows described above in section 5.1~~0~~ are for a single instance of a message over an interface/publication along with acceptance/rejection paths for that messages. For the workflows, described in section 3.4, an instance of each workflow will compromise of a series of messages with the receipt of one message triggering a processing step that will result in subsequent message. In order to link a single thread of a workflow process, the Workflow correlation Id is used.

The Workflow correlation Id is generated by the DIP on receipt of the first message in a workflow sequence. The DIP will configured to generate Workflow Ids for specific interfaces. Recipients of the initial message, and any further recipients of messages in the workflow sequence, will have the responsibility of transcribing the Workflow correlation Id from the Transaction block from the received message to the transaction block in any subsequent return messages.

Participants will also be able to track specific threads of workflows by tracking Correlation Id and MPAN via the DIP audit reports.

This is best explained with the aid of an example for BP009 – Change of Meter.

BP09 consists of two message flows; IF005 - and IF006. IF005 is the initial message, and IF-006 is the response.

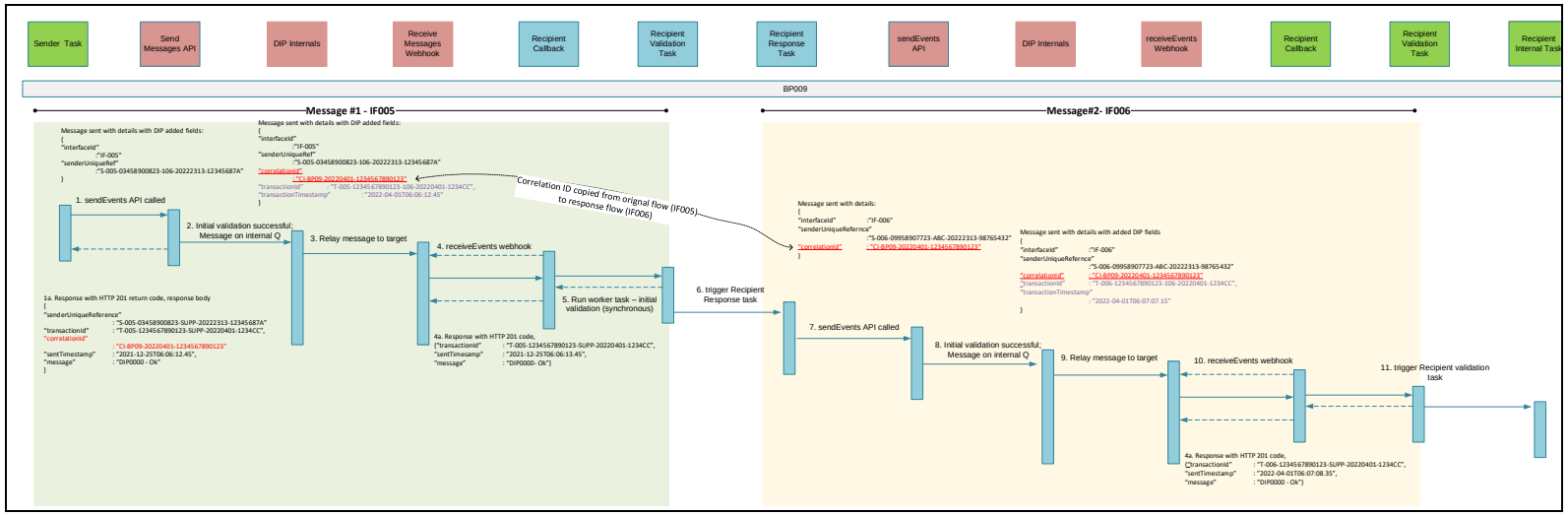


Figure 14 - Correlation Id example with IF-005 & IF-006

Formatted: Header, Indent: Left: -0.2 cm

Formatted: Header, Centered

Formatted: Header, Right, Right: -0.2 cm

Formatted Table

5.5 Message/Event Replay Facility

A basic query facility is required for message/event replay for each message channel for all participants. In the production environment, a request for message/event replay can be used to assist participants' downstream systems is suffering an unrecoverable loss of data. The request for replay will consist of the following:

- Message Channel ID/Event Code
- MPAN (optional)
- Start Message Transaction ID – the message from which to start the replay sequence from, **or**
- Date/time from – transaction time from which the first message needs to be replayed.
- End Message Transaction ID (optional) – the last message from which to start the replay sequence from, or
- Date/time to (optional) – transaction time message to be replayed

If End Message Transaction ID and Date/time to are not specified, the request will send all archived messages up to real-time.

If Start Message Transaction ID and Date/time from are both specified, then the earliest message of the two will be used, and End Message Transaction ID and Date/time to are specified then latest message will be used

There will be two methods by which the Event Replay facility can be initiated:

- Through the UI interface on the DIP
- Through an API Call

As well as being delivered via a separate API, the event replay will deliver the message under a new transaction wrapper with a Transaction ID and Correlation ID (with a reply prefix) so that the event/message are identified as a replayed, and hence downstream processes are not triggered. The initial message transaction wrapper will also be sent.

Only messages the Participant is entitled to view will be sent.

Formatted: Header, Indent: Left: -0.2 cm

Formatted: Header, Centered

Formatted: Header, Right, Right: -0.2 cm

Formatted Table

6 Message Volume and Submission Patterns

One of the design principles of the TOM is, wherever possible, to balance the exchange of messages over the day. The aim is to optimise message response times and to avoid overloading services with a bulk submission of messages that would need to be processed in a short timeframe. The majority of messages flows will not have any exclusive requirements in this area as the volumes are small and are not seen to be restrictive. However, some flows will require monitoring and requirements for managing the message flows may be required, these are described below and in the MHHS-DES001a - Functional Specification -Transaction Volumes - Appendix A v0.2 document.

6.1 Half-Hourly Consumption Data (IF-021)

The Half-hourly consumption data message flow (IF-021) equates to over 90% of the TOMs messages in terms of number and volume. In addition BP005 – Data Processing, the business process that uses this flow, has a gate closure event (00:00 D+4) that could potentially mean services may push to submit their data in a short window just before the gate closure and this needs to be avoided. The proposal is to recommend some "soft" targets so ensure that the Data Services (both Smart and Advanced) submit the messages across the day in order to spread the load. The ideal pattern would seem to be that batches of several thousand meter readings in a single API transaction would be appropriate. The introduction of "soft" limits will hopefully encourage Data Services with larger portfolios to design their submission patterns to spread the load as requested, if the "soft" limits are not adhered to, and extreme submission patterns are exhibited, where for example a Data Service attempts to send all messages with the last 30 minutes before gate closure, then "hard" limits will need to be imposed and the DIP will reject messages once the limit has been reached. It is also recognised that scenarios may arise where the bulk submission of data is required, for example after an upstream issue with the Meter Data Retrieval (MDR) service collecting metering data due to a network outage, in order to meet the settlement deadlines.

The proposed "soft" limits for organisations with large portfolios will be, under normal operating conditions, i.e. where either the DIP or the SDS is not returning from an outage, the maximum number of submissions each data service is encouraged to submit in an hour is 1/6th of their portfolio size. Then to ensure not all messages are submitted within the last six hours up to midnight, no more than ½ of their portfolio size can be submitted in a single 6 hour period. Portfolio size will be measured by the number of distinct MPANs submitted over the IF-021 interface. A suggested definition of large portfolio will be 1 million MPANs.

If Data Services do not want to design in a capability where submission are spread throughout the day, i.e. implement soft limits, then a response to API throttling must be designed in, i.e. hard limits. Hard limits will be enforced by API throttling at the DIP interface.

7 Message Auditing

All API activity will be logged for auditing and made available for audit reporting and message repudiation The DIP FS does present some ideas at a high-level on how the audit reporting requirements can be met, and at this stage there is no further detail than the ideas that were presented. These audit reports will be taken during the DIP design phase.

Auditing reports will enable participants to track messages for both individual messages exchanges and linked workflow messages exchanges using the following criteria:

- Message Channel Id
- Transaction Id
- Workflow Correlation Id
- Date/time range
- MPAN
- Sender
- Receiver

8 Technical Architecture

8.1 Open API Design (Swagger)

The latest ~~D~~dPrototype definitions are hosted here https://app.swaggerhub.com/apis/MHHSPROGRAMME/~~SubmitEvents/~~1.10.24[4]

8.2 Privacy & Security

Please see *End-to-End Security Architecture document*.

8.3 Performance

Performance criteria for all message/event channels are defined in the *E2E Non-Functional Requirements specification*.

8.4 Connection Patterns

Standardised connection patterns will across all services with each messageing/event channel accessed via a pair of API HTTP(s) (incoming) / webhook (outgoing) endpoints over mTLS.

Inbound API

There will be a single API endpoint for eachall message channels; the API will receive a multiple messages comprising a standard header and a mixed payload (encrypted). Each transaction is signed within the HTTP header. Each connection will be able to transmit a number of different messages during each connection (limits will need to be specified during the design phase).

The endpoints are hosted:

`https:/api.{environmment}.energydataintegrationplatform.co.uk/{version}/dip-channel/{IF-xxx}`

`{environment}` will be one of the following : prod; preprod; sit; uit

`{version}` current API version environment specific, at present 1.1

Each connection will result in a HTTP return code that will indicate the success or otherwise of the complete transaction.

The response body of the HTTP call will deliver the Sender a transaction ID, and optionally a correlation ID, against the Sender's Unique Reference for each message.

Each party will be able to make multiple simultaneous connections with the endpoint.

Outbound Webhook

There will also andbe a single endpoint for eachall publicationmessage channels. The expectation is that each receiving Participant will be served from multiplean outgoing message/event queues, and messages will be queued in the order received (however FIFO is not guaranteed). The response body of the HTTP call will record the transaction ID against the success/failure of the call.

If required, pParticipants will be able to register the same have multiple webhook subscriptions foron each publication the endpoint, or alternatively so they canare able to register different subscriptions logically on eachsplit the messages (by Publication Id.) as their services/applications may be hosted in different locations. Hence, participants will be able

[4] Whilst the APIs are being designed the API server and base URL defined in the API swagger definitions are only interim values. Once the final hosting arrangements for the DIP are known then these will change to the final correct values.

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New

~~to define a number of logical queues/endpoints in order to group the message/event channels they choose. The Participant will be able to assign each message/event channel to a specific subscription. These logical queues/endpoints may relate to individual services defined in the TOM.~~

The requirement will be to have standardised connection patterns across all services. All services will be expected to present as minimum API (inbound)/webhook (outbound) HTTP~~S~~ interfaces with signed JSON payload and encrypted in transit with mTLS. This is the minimum requirement for all services and should not rule out the possibility of having other connections on specific services where considered appropriate, for example, the use of proprietary cloud connectors should be considered for the high-volume interfaces, i.e., half-hourly consumption data if the DIP and the source systems are located within the same cloud platform.

8.5 Version Control

Versioning is a crucial part of API design. It gives developers the ability to improve their API without breaking the client's applications when new updates are rolled out. ~~The final details of API versioning will be established once the DIP Service Provider is on-board as they will have their preferred methods to use.~~

Due to the complexity of the data exchange between Market Participants and the DIP being mostly in the format of the messages where minor versioning is required, the proposal is for API versioning through custom headers. However, entity versioning, where there are major changes to the API then versioning through URI path would be more suitable.

As a principle, all API definitions need to be backwardly compatible. In the situation where this does not hold, then a significant transition period must be provided for participants to move from old to new version.

8.5.1 API~~Semantic~~ Versioning

API version control must adopt semantic versioning as the common standard, i.e.:

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

8.5.2 Message Channel Versioning

Each separate message channel has its own version number and is defined by the schemaVersion item in the payload/CommonBlock/S0, for example:

```
[
  {
    "payload": {
      "CommonBlock": {
        "S0": {
          "interfaceId": "IF-001",
          "schemaVersion": "004",
          "eventCode": "[InitialRegistration]"
        },
```

8.5.3 Versioning Examples

- Major version

A new major version of the API will be created when a new piece of functionality is required, or a change is required that will completely break the existing API. Under this scenario all interaces will only have forwards looking data definitions and not support previous versions of data.

Replay facilities would support both old & new data definitions (see below).

- Minor version

Where a new message channel is created, then a new minor version of the API will be created. This will not require a new API to be created, however, all the corresponding interface and publications data definitions will be need to be updated to accommodate the new channel.

Replay facilities would support both old & new data definitions (see below).

- Patch version

Formatted: Heading 3, Space Before: 0 pt, After: 0 pt

Formatted: Heading 3, Space Before: 0 pt, After: 0 pt

Formatted: Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: MHHS Body, Space Before: 0 pt, After: 0 pt, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted: MHHS Body, Space Before: 0 pt, After: 0 pt

Formatted: MHHS Body, Space Before: 0 pt, After: 0 pt, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Where the data definition of a single interface changes, then that will necessitate a patch version of the current version of the interface. Under 'normal' circumstances the incoming interfaces will only support the new version of the data definitions. If there is a requirement to support multiple version of the same interfaces (incoming) then a new minor version is required. In this scenario the publication of the messages will also support just the new current version. All data must have already traversed through the DIP.

Replay facilities would support both old & new data definitions (see below).

8.5.3.1 Replay Facilities

There is a requirement for the DIP message archive to be available to accommodate message replay by Market Participants. Where a message channel has been versioned to accommodate a change the message structure, the message repay will need to support all versions of the message. The schemaVersion in the message header informs. If a participant requests a message reply, it is their responsibility to ensure that all pertinent versions of the message will be accepted.

8.5.4 Multi-version support

There is a requirement for multi-version support of APIs. All new major, minor and patch versions, i.e. x+1, or Y+1, or Z+1, will be available prior to release for industry testing and potential re-qualification for major releases on a non-production environment.

It is difficult to view the type of changes that would be included in a major release, hence any statement on backward compatibility is difficult to make, however, where possible then backward API compatibility should be maintained for a short period after the roll out of a new major version.

For minor releases multi-version support will be dependent on the type of underlying change. It is expected that most will be data changes and new message channels.

Formatted: MHHS Body, Space Before: 0 pt, After: 0 pt

Formatted: Heading 4, Space Before: 0 pt, After: 0 pt

Formatted: Outline numbered + Level: 3 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.5 cm + Indent at: 1.77 cm