UK NATIONAL TOBACCO TRACK & TRACE GATEWAY: TECHNICAL SPECIFICATIONS AND DATA DICTIONARY

V1.1

dentsu
TRACKING

Summary of changes

Date	Version	Done by	Comment
21/03/2022	1.0	Dentsu Tracking	Initial version
17/05/2022	1.1	Dentsu Tracking	Updated AUTH URL

Publication

Date	Version	Submitted to
17/05/2022	1.1	Uploaded to Document Center

Table of Contents

1

1 Introduction

This document sets out the technical specifications and data dictionary referred to in Commission Implementing Regulation (EU) 2018/574, as applied and amended by The Tobacco Products (Traceability System and Security Features) (Amendments) (EU Exit) Regulations 2020.

This document is intended for all Economic Operators required to connect and report to the UK tobacco track and trace system, as required by the UK Regulations.

2 System Overview

This document defines the technical specifications for the Gateway of the UK Tobacco Track & Trace System established and operated by Dentsu Tracking. The information provided in this document includes information about data types, messages, authentication, security, detailed message validation description, error messages, and an overall system overview.

2.1 Definitions and Abbreviations

The definitions set out in Commission Implementing Regulation (EU) 2018/574 and its annexes apply.

Expression or Acronym	Definition
CIR 2018/574	Commission Implementing Regulation (EU) 2018/574, as amended by the Tobacco Products (Traceability System and Security Features) (Amendments) (EU Exit) Regulations 2020
EO	Economic Operator
EOID	Economic Operator Identifier code
FID	Facility Identifier Code
MID	Machine Identifier Code
FCTC	World Health Organization Framework Convention on Tobacco Control
HMRC	Her Majesty's Revenue and Customs 'the Authority' (the Contracting Authority)
KPI	Key Performance Indicator
RoW	Rest of World. Other countries outside the UK.
SLA	Service Level Agreement
UI	Unique Identifier. The alphanumeric code enabling the identification of a unit packet or an aggregated packaging of tobacco products.
GB	Great Britain
NI	Northern Ireland

NI Protocol	Northern Ireland Protocol (to the UK/EU Withdrawal Agreement)
DLR	De La Rue
Service Provider	Third party acting on behalf of one or multiple Economic Operator, submitting messages to the UK Gateway
ATD	Anti-Tampering Devices.

2.2 Key design principles

The UK Track & Trace system consists of the following sub-systems: UK ID Issuer, UK Gateway and UK Data Repository.

- The **UK ID issuer** generates and issues unique identifier codes for tobacco products destined for and travelling through the UK Territory (Great Britain and Northern Ireland), as well as identifier codes that enable the registration of Economic Operator, Facilities and Machines in the system. For detailed information on the ID Issuer, please refer to the separate document 'UK ID Issuer Technical Specifications'.
- Economic operators must record all product movements and transactional data for tobacco products, from the manufacturer to last economic operator before the first retail outlet. All recorded data must be transmitted to the **UK Gateway** which will perform the applicable business and technical validations on the submitted data and provide a positive or negative acknowledgment message to the sender accordingly.
- The received data is subsequently transferred to the **UK Data Repository** where it is accessible to authorised personnel of HMRC and any other nominated authority

2.3 UK Track and Trace System

The following diagram represents the overall system design of the UK Track & Trace System.
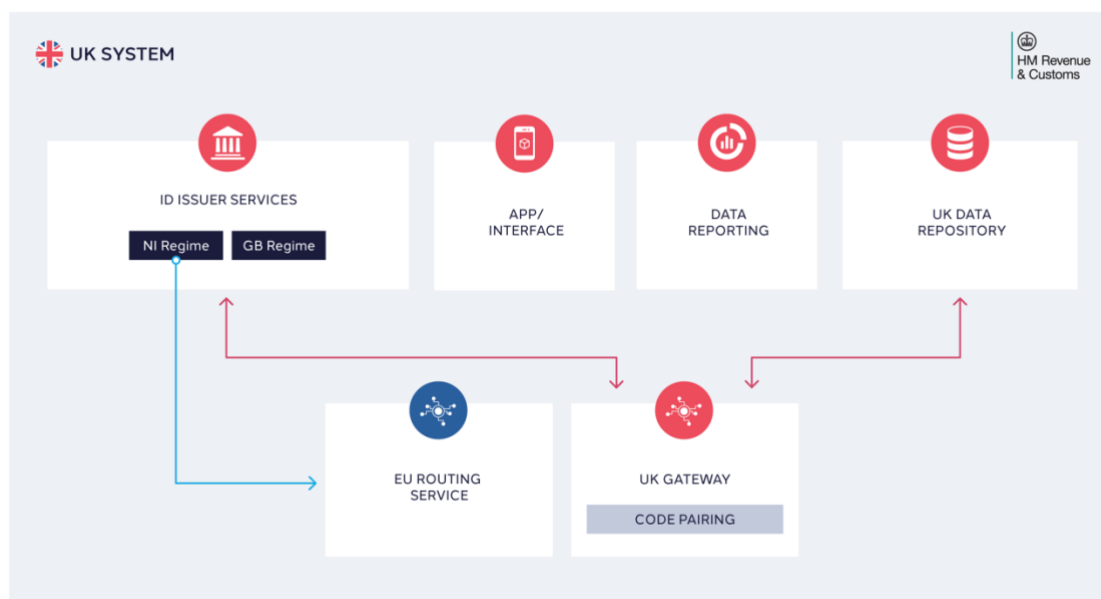
Figure 1 Overall system interconnection

2.3.1 UK ID Issuer

Tobacco manufacturers and importers, along with all other businesses in the tobacco supply chain, must be registered with the Dentsu UK ID Issuer to obtain an economic operator ID (EOID). Once registered, manufacturers must also register for Machine IDs (MID) for machines integral to the manufacturing process. All businesses in the tobacco supply chain must register for Facility IDs (FID) for all premises where they manufacture, store, handle or sell their products. IDs previously registered in the system operated by DLR are migrated to the new UK system operated by Dentsu. Newly registered IDs (either EOIDs, MIDs or FIDs) are provided free of charge by Dentsu.

The process to request additional identifier codes (EOID, FID, MID) is described in the UK ID Issuer Technical Specifications document.

Each packet of cigarette and hand rolling tobacco products manufactured in, or imported into the UK, must have a UK UI applied to and/or associated with the individual product. This requirement applied to other tobacco products from May 2024.

Where cigarette and hand rolling tobacco products imported into the UK are already marked with a UI issued by another country operating a track and trace system, such as in the EU, the UK UI is not printed on the packet. Instead, it is digitally associated with the UI already marked on the packet. This is a mandatory requirement in UK legislation.

UIs come in the form of an alphanumeric code. They can be either unit packet UIs or aggregate UIs. An aggregate UI is used for anything that holds more than one unit packet e.g., cartons, master cases, pallets, trailers and shipping containers. It is then linked to the unit packets that it contains. Manufacturers and importers can request unit packet UIs from Dentsu for a fee. Aggregate UIs can be either purchased from Dentsu or generated by registered economic operators themselves. Economic operators are responsible for encoding the digital UK UI and ensuring its correct application (see UK ID Issuer Technical Specifications).

When Economic Operators decide to self-generate the aggregated level identifier codes, these must comply with the ISOs described in the CIR (EU) 2018/574.
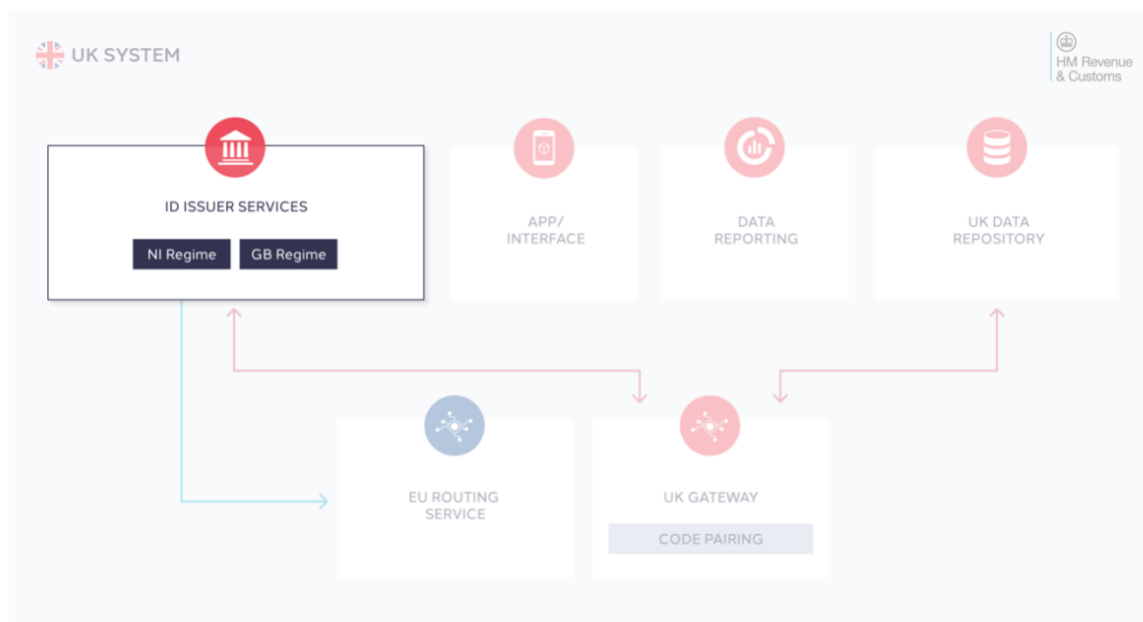
Figure 2 - UK System - ID Issuer services

Dentsu operates the UK ID Issuer under two territorial regimes: NI Regime and GB Regime.

Under the NI Regime, the UK ID Issuer is connected to both the UK System and the EU System. IDs (EOID, FID, MID) and UIs issued for the NI territory are to be stored in both the UK System and the EU System.

The UK ID Issuer component for NI complies with the requirements outlined CIR 2018/574, enabling the UK and economic operators to comply with their obligation under the NI Protocol. Existing IDs previously registered in the DLR system for the NI Regime are transferred from DLR to Dentsu via a data migration.

Under the GB Regime, the UK ID Issuer is connected only to the UK System. Economic Operators can register IDs and request UIs. Additionally, for products destined for the GB territory the UK System (UK Gateway and UK Data Repository) will facilitate the transmission of code-pairing requests, allowing Manufacturers and Importers to digitally associate a third country UI already marked on the packet, with a UK UI.

2.3.2 UK Gateway

All data regarding cigarettes and hand rolling tobacco products manufactured in and imported into the UK (and from May 2024, other tobacco products) are routed to the UK data repository.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

7

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

Figure 3 - UK System - UK Gateway

The UK Gateway is a central component of the UK track and trace system. The UK Gateway receives events from the UK ID Issuer and messages reported by Economic Operators and forwards them to the UK Repository. It therefore serves as a central reporting point for all Economic Operators. The Gateway also ensures a high level of data reporting quality and the overall integrity of all reported UIs due to sophisticated, high performing validation mechanisms (business and technical validations) that are applied to all received messages.

The following diagram shows all parties and components involved in this process:
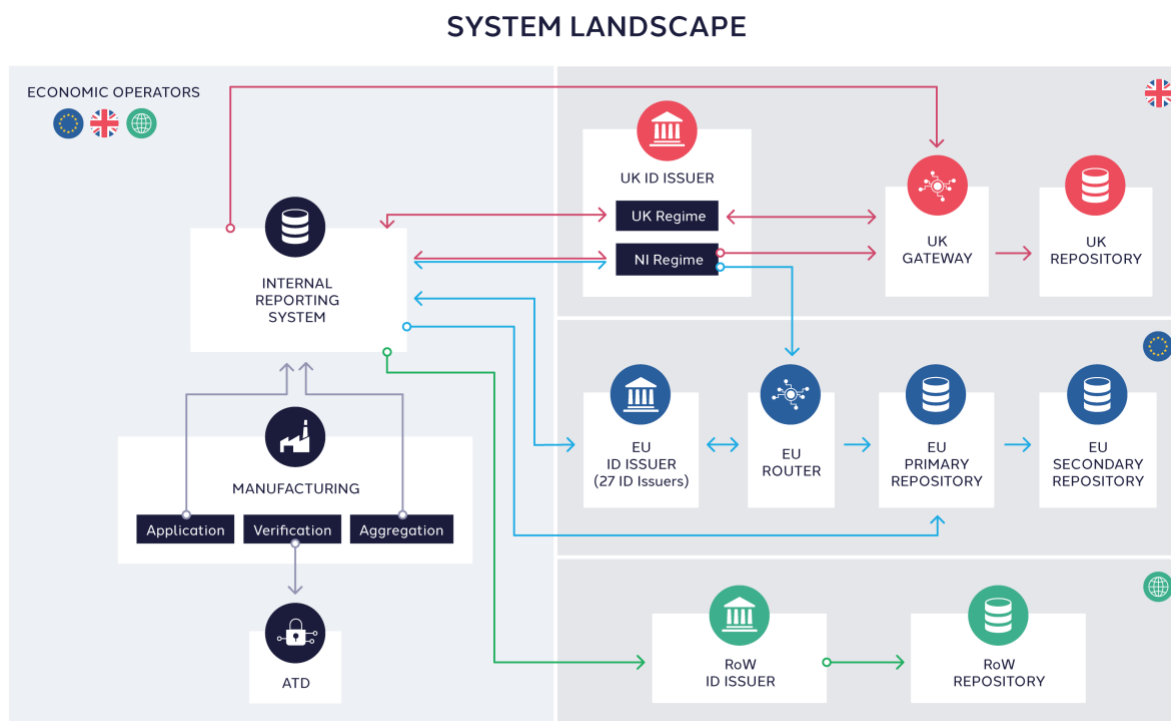
Figure 4 - UK Track & Trace involved parties

Economic Operators will be responsible for transmitting messages to the UK Gateway on the reporting of product movement and transactional events.

Economic Operators will be responsible for transmitting messages to the UK Gateway and the EU Router on the reporting of product movement and transactional events that occur on the NI territory.

The UK Gateway only transmits data to the UK Repository and is not interfacing with any other Track & Trace system from third country regimes, such as the EU tobacco traceability system. Economic Operators are responsible for ensuring transmission of relevant data to any other third country systems, such as the EU system, where applicable.

2.3.3 UK Data Repository

All information registered in and transferred to the UK track and trace system are routed by the UK Gateway to the UK Data Repository, where that information will be accessible to the UK Authority. All data are stored in a secure environment that is only accessible to authorised government personnel.
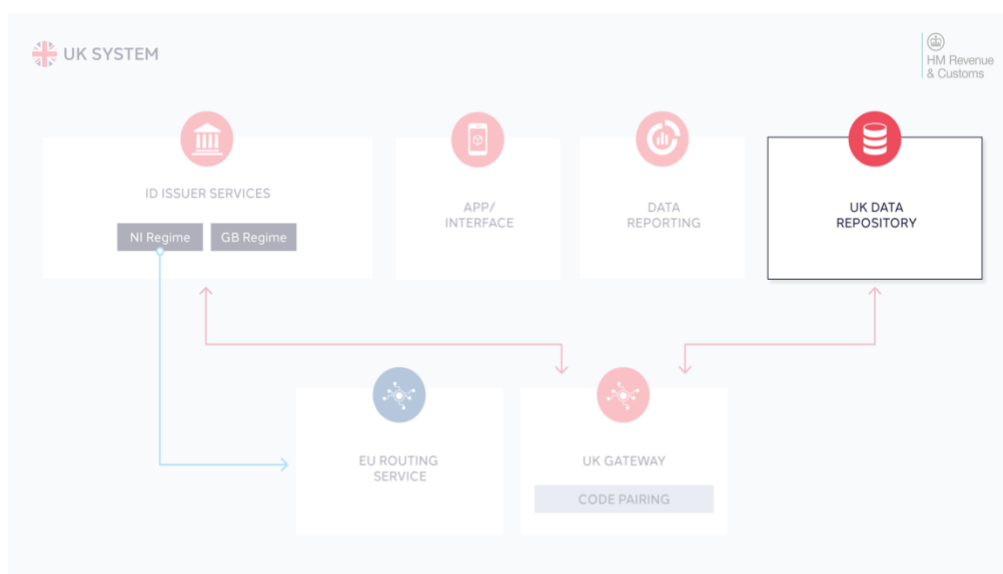
Figure 5 - UK Data Repository

3 Interfaces

3.1 Overview

This section describes the interfaces through which Economic Operators and Service Providers interact with the UK Gateway. The UK Gateway will be accessible via API.

The process to request credentials for the API will be shared as part of the onboarding process in the UK Track & Trace System. Economic Operators will be able to request credentials on demand.

The planned period to start to be able to request programmatic access to the UK Gateway is scheduled to start on the 1 May 2022 for the QA system and on 1 June 2022 for the production system.

3.1.1 Accepted protocols

3.1.1.1 JSON

The API supports HTTPS-based REST API in JSON format.

HTTP Method POST is used for all requests.

3.1.2 Encoding

All messages are encoded in UTF-8.

3.1.3 Secured Communication

Communication between the interacting participants and the UK Gateway is secured by TLS 1.2 encryption AES256 cypher. Less secure Cypher suites are not supported. If the TLS version or cypher used proves to be corroded or vulnerable, Dentsu reserves the right to replace the affected item with a state-of-the-art item after prior announcement.

The Repository System uses OAuth 2.0 to authorize access to the web service methods. OAuth 2.0 is the industry-standard protocol for authorisation. OAuth 2.0 supersedes the work done on the original OAuth protocol created in 2006. OAuth 2.0 focuses on client developer simplicity while providing specific authorisation flows for web applications, desktop applications and server to server communication. The system uses the OAuth client credential flow. The client credentials flow is used as authorisation grant because the authorisation scope is limited to the protected resources previously arranged with the authorisation server (UK Repository).

Access tokens are issued in the form of credentials used to access protected resources. An access token is a string representing an authorisation issued to the client. The string is opaque to the client and passed in the authentication header. Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorisation server. Tokens expire after 3600 seconds (1 hour).

3.2 UK Gateway interface

3.2.1 System Reception Timestamp

In some cases, economic operator systems can generate bursts of messages. Several messages can be produced during the same second and therefore will have the same EventTime and the same MessageTimeLong.

In order to implement efficiently the required sequence validation controls, the System Reception_Time is defined at the level of milliseconds.

The Reception_Time will be recorded and added by the UK Gateway upon reception of a given message.

3.2.2 Message identification and RecallCode

3.2.2.1 Overview

The Gateway assigns a unique chain of characters that uniquely identifies each message, called RecallCode.

3.2.2.2 Message Recall

Economic operators have the possibility to recall messages after those were transmitted to the UK system.

The reasons for recalling the original message may be:

1. Reported event did not materialise (only for messages related to dispatch events and trans-loading);
2. Message contained erroneous information;
3. Other.

3.2.2.3 RecallCode structure

ReallCode structure follows version 5 of the UUIs standards set out in ISO/IEC 11578:1996.

3.2.2.4 Messages

The following table describes the messages that may be recalled by economic operators.

	Annex II Reference	
PAR	(3.A1)	Code Pairing
EUA	(3.1)	Application of unit level UIs on unit packets
EPA	(3.2)	Application of aggregated level UIs on aggregated packaging
EDP	(3.3)	Dispatch of tobacco products from a facility
ERP	(3.4)	Arrival of tobacco products at a facility
ETL	(3.5)	Trans-loading
EUD	(3.6)	Disaggregation of aggregated level UIs
EVR	(3.7)	Report of delivery carried out with a vending van to retail outlet
EIV	(4.1)	Issuing of the invoice
EPO	(4.2)	Issuing of the order number
EPR	(4.3)	Receipt of the payment

3.2.2.5 Recall Process

The recall must include the respective message's RecallCode which was provided to the message sender in the acknowledgement of the original message. The recall must also contain the following information:

- Reason for recalling the original message;
- Description of the reason for recalling the original message;
- Any additional explanations on the reason for recalling the original message.

Recall of a given message results in flagging that message as cancelled but it does not delete the original message's database record.

3.2.2.6 RecallCode Field

The RecallCode is obtained from the original message's "code" property:

```
JSON Example response:
{
    "Code": "6854f9a6-a2b2-4c08-8000-0173f3c35567",
    "Message_Type": "EPA",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

Whereas "Code" is the RecallCode.

1.1.5 Message response

A message transmission corresponds to a message request performed by the sender system and a message response provided by the destination system. The message response contains the http status and the body of the message response.

3.2.2.7 Successful response or event acknowledgment

In line with CIR 2018/574, messages / events are considered reported upon the receipt of the acknowledgement message (successful) transmitted from the UK Gateway to the sender system. The http status for the message "positive response without any warning" is 200 and 202.

Note: A warning message (http status 299) is also considered as a successful response (message is acknowledged).

3.2.2.8 Negative response

The UK Gateway sends a negative response if the reported event does not meet the technical specifications. Negative response http status is in the range of 400-499 and 500-599.

3.2.2.9 Timeout

If the UK Gateway did not produce a response within the time that the sender system was prepared to wait, the sender system may repeat the request without modifications at any later time. The absence of a response (or the http timeout response) indicates that the message is not acknowledged.

3.2.2.10 Timeout handling

In case of a timeout for a certain request, the sender system should retransmit the original message (identical payload). If the sender system changes the original message (e.g., by updating the Message Time Long), the UK Gateway will consider the message as a different message.

If the UK Gateway was able to process the original message, but the sender never got a reply and retransmits the message with an identical payload, then the UK Gateway will return the Error Code "PAYLOAD_NOT_UNIQUE" along with the original RecallCode assigned to the original request. In this case, the sender will consider the original message to be positively acknowledged and does not need to resubmit it a third time.

3.2.2.11 JSON Successful response sample

```
HTTP Status 202
{
    "Code": " 6854f9a6-a2b2-4c08-8000-0173f3c35567",
    "Message_Type": "EPA",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

3.2.3 Error response detailed

The UK Gateway provides sufficient details that allow external system administrators to identify the issue and act accordingly.

The response message can contain multiple errors, see below:

```
JSON
    "Errors": [
        { << Error >>},
        { << Error >>},
        { << Error >>},
    ],
```

Each error contains the following information:
- **Error_InternalID** is the unique identification of the message processing and validation activity.
- **Error_Code** is the identifier of the type within the system.
- **Error_Descr** is the description in human readable format containing specific error information.
- **Error_Data** is the data to which the error relates. This can be: EOIDs, FIDs, MIDs and UIs.
  - Note: the UK Gateway uses the character **#** as separator for the UIs in case a list of UIs is provided in the error data field.

Example List of Errors:

13

```JSON
{
    "Error_InternalID": "yndkFz7TBEO706frD38hzA",
    "Error_Code": "INVALID_REQUEST_FORMAT",
    "Error_Descr": "The EconomicOperatorIdentifier field is unknown."
    "Error_Data": "123456789123456#123456789123455#123456789123444"
}
```

Note that "Error_InternalID" is only returned for system errors (HTTP 500).

3.2.3.1 Security errors

Security errors are provided **before** the sender can be authenticated in the UK system. Therefore, these error message types do not return the ordinary error code structure of a negative acknowledgement as foreseen under CIR 2018/574. Instead, these are transmitted in standardised version of HTTP error responses, such as:

HTTP status	Error Code	
401		Invalid security token
401		Expired security token

3.2.3.2 Processing Errors

Processing Errors are generated **after** the sender has been authenticated in the UK system but are the result of a malformed message. In this case, the Error Code is provided as follows:

HTTP status	Error Code	
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.
504	Gateway Timeout	The request timeout

Error body sample:

```
JSON
{
    "Code": null,
    "Message_Type": null,
    "Error": true,
    "Errors": [
        {
            "Error_InternalID": "yndkFz7TBEO706frD38hzA",
            "Error_Code": "INVALID_REQUEST_FORMAT",
            "Error_Descr": "The EconomicOperatorIdentifier field is required.",
            "Error_Data": "54G7J"
        }
    ],
    "Checksum": "G6HF5H"
}
```

Note that "Error_InternalID" is only populated for system errors (HTTP 500).

3.2.4 Message integrity and hash

The UK Gateway always verifies the message checksum to ensure that data was not tampered with during transit. Messages containing an invalid hash are not accepted.

The client will add a MD5 hash to the X-OriginalHash HTTP header. This structure is then added to the message.

Message Header – JSON

X-OriginalHash	1234567890abcdefghijklmnopqrstuvwxyz
Content-Type	application/json
Authorisation	<Token>

If the MD5 hash validation fails, the UK Gateway will return the following error code:

HTTP status	Error Code	
401	INVALID_SIGNATURE	"The message signature does not match"

3.2.5 Message size

The maximum message size is 6MB. The limit on the HTTP header size is 10'240 bytes.

3.2.5.1 Maximum number of UI

Online sequence validation controls require a limitation on the number of UIs (sum of the unit level UI and the aggregated level UI) per message for the following events:

Message Type	Annex II Reference	Message description	Number of UI (upUI + aUI)
IDA	(2.3)	Request for deactivation of UIs	10 000
EUA	(3.1)	Application of unit level UIs on unit packets	10 000
PAR	(3.A1)	Paring event of unit level UIs	10 000
EPA	(3.2)	Application of aggregated level UIs on aggregated packaging	10 000
EDP	(3.3)	Dispatch Event	10 000
ERP	(3.4)	Reception event	10 000
ETL	(3.5)	Trans-loading event	10 000
EUD	(3.6)	Message to report an UI disaggregation	10 000
EVR	(3.7)	Report the delivery carried out with a vending van to retail outlet	10 000
EIV	(4.1)	Message to report an invoice	10 000
EPO	(4.2)	Purchase order	10 000
EPR	(4.3)	Payment record	10 000

3.2.6 Number of simultaneous connections

There is no limit for simultaneous connections between the sending systems and the UK Gateway.

3.2.7 Message Sequence

Message sequence must comply with the requirements of CIR 2018/574. The reporting of messages to the UK Gateway is completed upon receipt of an acknowledgement message by the UK Gateway.

3.2.8 Buffering and Burst transmissions

Messages should be transmitted continuously by the different systems without buffering.

In case of buffering caused by technical maintenance activities, the transmitting system should implement mechanisms to ensure the correct sequencing of the events

3.2.9 Message Retransmission limitation

A message that was positively acknowledged must not be retransmitted a second time (unless the sender received a timeout and is not aware of the positive acknowledgement).

In case that a positive acknowledge message is retransmitted to the UK Gateway, the system will return a "PAYLOAD_NOT_UNIQUE" error by computing an MD5 hash of the body of the message to identify the duplicate retransmission.

4 Data description

4.1 Data types

Data Type	Description	Type	Example or regular expression
ARC	Administrative Reference Code (ARC) or any successive code adopted under the Excise Movement and Control System (EMCS)	Text(30)	'15GB0123456789ABCDEF0' Validation RegEx: ^[a-zA-Z0-9]*$
aUI	Aggregated level unique identifier coded with: either The invariant set of ISO646:1991 and composed of four blocks: (a) ID issuer's prefix in accordance with ISO15459-2:2015, (b) serialization element in the format established by the ID issuer, (c) tobacco facility identifier code following the Data Type: FID and (d) timestamp following the Data Type: Time(s) or The invariant set of ISO646:1991 forming a code structured in accordance with ISO15459-1:2014 or ISO15459-4:2014 (or their latest equivalent))	Text(100)	
Boolean	Boolean value	Boolean	• 0 (false/disabled) • 1 (true/enabled)
Component	A data type defined in the data dictionary		Aggregation
Country	Country name coded with ISO-3166-1:2013 alpha-2 (or its latest equivalent)	Text(2)	'GB'
Currency	Currency name coded with ISO 4217:2015 (or its latest equivalent)	Text(3)	'GBP'

Date	A UTC data in text corresponding to the following format: YYYY-MM-DD	Text(10)	E.g. '2017-03-31'
Decimal	Number values, decimal allowed	Decimal	E.g. '1' or '22.2' or '333.33'
Email	Maximum 80 characters	Text(80)	E.g. 'info@test.com' Validation RegEx: `^((([a-zA-Z]|\d|[!#\$%&'\*\+\-\/=\?\^_`{|}~]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+(\.([a-zA-Z]|\d|[!#\$%&'*\+\-\/=\?\^_`{|}~]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])+)*)|((\x22(((\x20|\x09)*(\x0d\x0a))?(\x20|\x09)+)?(([\x01-\x08\x0b\x0c\x0e-\x1f\x7f]|\x21|[\x23-\x5b]|[\x5d-\x7e]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFEF])|(`**Error! Hyperlink reference not valid.**
EOID	Economic operator identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO8859-15:1999 EOID starts with the alphanumeric characters that constitute the ID issuer identification code, followed by alphanumeric sequence which is unique within the code pool of the ID issuer.	Text(50)	

FID	Tobacco facility identifier code corresponding to the format established by ID issuer coded with the invariant set of ISO8859-15:1999	Text(50)	
Integer	Rounded number values, no decimal numbers	Integer	E.g. '1' or '22' or '333'
IIID	ID Issuer code in line with the issuing agency codes of ISO/IEC 15459	Text(35)	E.g. 'QCUKT'
ITU	Individual transport unit code (e.g. SSCC) generated in accordance with ISO15459-1:2014 (or its latest equivalent)		'00791234560000000018'
List	Must be only one of the values present in the 'Values' column		
MID	Machine identifier code corresponding to the format established by ID Issuer coded with the invariant set of ISO8859-15:1999.	Text(50)	
MRN	Movement Reference Number (MRN) is a unique customs registration number. It contains 18 digits and is composed of the following elements: (a) last two digits of the year of formal acceptance of export movement (YY), (b) country name coded with ISO3166-1:2013 alpha-2 (or its latest equivalent) of the country to which the declaration was sent, (c) unique identifier for entry/import per year and country, and (d) check digit.	Text(18)	'19IT9876AB88901235' Validation RegEx: `^[0-9]{2}[A-Z]{2}[a-zA-Z0-9]+[0-9]{1}$`
TP_PN	Product number – numeric identifier used in the CEG system to identify product	Text(30)	'00012345600012'

	presentations (e.g. GTIN (Global Trade Identification Number) of the product).		
SEED	Excise number composed of: (a) country name coded with ISO-3166-1:2013 alpha-2 (or its latest equivalent) (e.g. 'LU') and (b) eleven alphanumeric characters, if needed, padded to the left with zeroes (e.g. '00000987ABC').	Text(13)	'LU00000987ABC' Validation RegEx: ^[A-Z]{2}[a-zA-Z0-9]{11}$
Serial	Number corresponding with the invariant set of ISO646:1991 used for serialisation.		
SSCC	SSCC-18 container code generated in line with ISO6346:1995 (or its latest equivalent)	Text(20)	00791234560000000018
Text (X)	Alphanumeric values coded with ISO8859-15:1999 limited to X characters		E.g. 'Abcd' or '123455588845'
Time(s)	UTC (Coordinated Universal Time) time in the following format: YYMMDDhh	Text(8)	'19071619'
Time(L)	UTC (Coordinated Universal Time) time in the following format: YYYY-MM-DDThh:mm:ssZ	Text(34)	E.g. '2020-03-31T23:16:45Z'
Time(ms)	Time(ms) format format : yyyy-MM-ddTHH:mm:ss.fffZ		E.g '2020-08-13T16:01:34.477Z'
TPID	Tobacco Product Identifier (TP-ID) – numeric identifier used in the CEG system in the format: NNNNN-NN-NNNNN	Text(14)	'02565-16-00230' Validation RegEx: ^[0-9]{5}-[0-9]{2}-[0-9]{5}$
upUI(L)	Unit packet level unique identifier coded with the		

	invariant set of ISO646:1991 and composed of three blocks: (a) ID Issuer's prefix in line with ISO154592:2015, (b) middle block in the format established by ID Issuer and (c) timestamp following the Data Type: Time(s)		
upUI(s)	Unit packet level unique identifier coded with the invariant set of ISO646:1991 and composed of two blocks: (a) ID Issuer's prefix in line with ISO154592:2015 and (b) serialisation element in the format established by ID issuer (i.e. UI made visible in the human readable format on the unit packets)		
upUI(M)	Unit packet level unique identifier coded with the invariant set of ISO646:1991 and composed of 2 blocks: ID Issuer's prefix in accordance with ISO154592:2015, middle block in the format established by the ID issuer.		"LECNIzzzz000010000SLE"

4.2 Priority types

Type	Explanation
Mandatory (M)	The variable must be completed.
Optional (O)	The variable is for optional fields which could be filled depending on the record status or type.

4.3 Cardinality types

Type	Explanation
Simple (S)	Single value
Multiple (M)	Multiple values

4.4 Master Data Types

4.4.1 Country Codes

Code	Value
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AI	Anguilla
AL	Albania
AM	Armenia
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
AX	Åland Islands
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi

BJ	Benin
BL	Saint Barthélemy
BM	Bermuda
BN	Brunei Darussalam
BO	Bolivia (Plurinational State of)
BQ	Bonaire, Sint Eustatius and Saba
BR	Brazil
BS	Bahamas
BT	Bhutan
BV	Bouvet Island
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CC	Cocos (Keeling) Islands
CD	Congo, Democratic Republic of the
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CU	Cuba
CV	Cabo Verde

CW	Curaçao
CX	Christmas Island
CY	Cyprus
CZ	Czechia
DE	Germany
DJ	Djibouti
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
EH	Western Sahara
ER	Eritrea
ES	Spain
ET	Ethiopia
FI	Finland
FJ	Fiji
FK	Falkland Islands (Malvinas)
FM	Micronesia (Federated States of)
FO	Faroe Islands
FR	France
GA	Gabon
GB	United Kingdom of Great Britain
GD	Grenada
GE	Georgia
GF	French Guiana

GG	Guernsey
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GP	Guadeloupe
GQ	Equatorial Guinea
GR	Greece
GS	South Georgia and the South Sandwich Islands
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HK	Hong Kong
HM	Heard Island and McDonald Islands
HN	Honduras
HR	Croatia
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IM	Isle of Man
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran (Islamic Republic of)

IS	Iceland
IT	Italy
JE	Jersey
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KH	Cambodia
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis
KP	Korea (Democratic People's Republic of)
KR	Korea, Republic of
KW	Kuwait
KY	Cayman Islands
KZ	Kazakhstan
LA	Lao People's Democratic Republic
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya

MA	Morocco
MC	Monaco
MD	Moldova, Republic of
ME	Montenegro
MF	Saint Martin (French part)
MG	Madagascar
MH	Marshall Islands
MK	Macedonia, the former Yugoslav Republic of
ML	Mali
MM	Myanmar
MN	Mongolia
MO	Macao
MP	Northern Mariana Islands
MQ	Martinique
MR	Mauritania
MS	Montserrat
MT	Malta
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NC	New Caledonia
NE	Niger
NF	Norfolk Island
NG	Nigeria

NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NR	Nauru
NU	Niue
NZ	New Zealand
OM	Oman
PA	Panama
PE	Peru
PF	French Polynesia
PG	Papua New Guinea
PH	Philippines
PK	Pakistan
PL	Poland
PM	Saint Pierre and Miquelon
PN	Pitcairn
PR	Puerto Rico
PS	Palestine, State of
PT	Portugal
PW	Palau
PY	Paraguay
QA	Qatar
RE	Réunion
RO	Romania
RS	Serbia
RU	Russian Federation
RW	Rwanda

SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore
SH	Saint Helena, Ascension and Tristan da Cunha
SI	Slovenia
SJ	Svalbard and Jan Mayen
SK	Slovakia
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
SS	South Sudan
ST	Sao Tome and Principe
SV	El Salvador
SX	Sint Maarten (Dutch part)
SY	Syrian Arab Republic
SZ	Eswatini
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau

TL	Timor-Leste
TM	Turkmenistan
TN	Tunisia
TO	Tonga
TR	Turkey
TT	Trinidad and Tobago
TV	Tuvalu
TW	Taiwan, Province of China
TZ	Tanzania, United Republic of
UA	Ukraine
UG	Uganda
UM	United States Minor Outlying Islands
US	United States of America
UY	Uruguay
UZ	Uzbekistan
VA	Holy See
VC	Saint Vincent and the Grenadines
VE	Venezuela (Bolivarian Republic of)
VG	Virgin Islands (British)
VI	Virgin Islands (U.S.)
VN	Viet Nam
VU	Vanuatu
WF	Wallis and Futuna
WS	Samoa
YE	Yemen
XI	Northern Ireland (in accordance with the Northern Ireland Protocol)
YT	Mayotte

ZA	South Africa
ZM	Zambia
ZW	Zimbabwe

4.4.2 DeactivationReasonType

Value	Name
1	Product destroyed
2	Product stolen
3	UI destroyed
4	UI stolen
5	UI unused
6	Other

4.4.3 EventType

Value	Name
IDA	Request for deactivation of UIs
PAR	Pairing between marked codes and virtual paired codes
EUA	Application of unit level UIs on unit packets
EPA	Application of aggregated level UIs on aggregated packaging
EDP	Dispatch Event
ERP	Arrival event
ETL	Trans-loading event
EUD	Message to report an UI disaggregation
EVR	Report the delivery carried out with a vending van to retail outlet
EIV	Message to report an invoice
EPO	Purchase order
EPR	Payment record

RCL	Recall messages

4.4.4 InvoiceType

Value	Name
1	Original
2	Correction
3	Other

4.4.5 NotificationType

Value	Name	Description
1	Informative	The notification only includes descriptive information, but not related to any error or abnormal situation.
2	Warning	The notification includes information about some alert or warning to be considered.
3	Alarm	The notification includes information about some alarm triggered by the System.
4	InternalError	The notification includes information about some error that has occurred within the System.
5	Other	The notification includes information about any other types of situation, not listed above, that occurred within the UK track and trace system.

4.4.6 PaymentType

Value	Name
1	Bank transfer
2	Bank card
3	Cash
4	Other

4.4.7 RecallReasonType

Value	Name
1	Reported event did not materialise

2	Message contained erroneous information
3	Other

4.4.8 RegisterStatus

Value	Name
1	Registered
2	De-registered

4.4.9 TobaccoProductType

Value	Name
1	Cigarette
2	Cigar
3	Cigarillo
4	Roll your own tobacco
5	Pipe tobacco
6	Waterpipe tobacco
7	Oral tobacco
8	Nasal tobacco
9	Chewing tobacco
10	Novel tobacco product
11	Other

4.4.10 TransportMode

Value	Name
0	Other
1	Sea Transport
2	Rail transport
3	Road transport
4	Air transport

5	Postal consignment
6	Fixed transport installations
7	Inland waterway transport

4.4.11 UniqueIdentifierState

Value	Name	Description
1	Generated	The UK ID Issuer reports the issuing of codes and the UK Data Repository creates a UI record with the initial state (i.e. Generated).
2	Activated	The UI, after being verified by the manufacturer/importer, matches one UI stored in the UK Data Repository under the status "Generated" and/or "Paired". The information contained in the date element of information matches the valid activation date for that UI.
3	Deactivated	An Economic Operator reports the deactivation of a given UI.
4	Expired	For UIs that have been issued, but their activation was reported within a certain period of time (i.e. expiration time, 6 months from issue date as per the regulation), the UK Data Repository renders those UIs as "Expired".
5	Paired	A UK issued UI is digitally associated (paired) with a third-country system "printed_code", where the pairing takes place prior to the activation and application of the foreign code. The foreign code (printed_code) is also created with status "paired" in the UK Data Repository

4.4.12 UniqueIdentifierType

Value	Name	Description
1	UnitPacket	Unique identifier at unit packet level
2	AggregatedPackaging	Unique identifier at aggregated packaging level

5 Messages

5.1 Message types to be exchanged

The message types set out in this section are laid down in Annex II to CIR 2018/574 Which is the official source and reference for all message fields and corresponding entries.

There are five categories of message, relating to:

- Identifier codes for economic operators, facilities and machines *(further described in the UK ID Issuer Technical Specifications)*
- Unique identifiers for unit level UIs or to be paired with associated unit level UIs, and for aggregated level UIs
- Recording and transmission of information on product movements
- Transactional events
- Recalls

The following table summarizes the JSON formatted messages.

Message Type	Annex II Reference	Message description
IDA	(2.3)	Request for deactivation of UIs
EUA	(3.1)	Application of unit level UIs on unit packets
PAR	(3.A1)	Code pairing of unit level UIs
EPA	(3.2)	Application of aggregated level UIs on aggregated packaging
EDP	(3.3)	Dispatch Event
ERP	(3.4)	Reception event
ETL	(3.5)	Trans-loading event
EUD	(3.6)	Message to report an UI disaggregation
EVR	(3.7)	Report the delivery carried out with a vending van to retail outlet
EIV	(4.1)	Message to report an invoice
EPO	(4.2)	Purchase order
EPR	(4.3)	Payment record
RCL	(5.0)	Recall messages

5.2 JSON Messages

5.2.1 Basic information block concerning the request

The following field is common to all messages.

Basic information block concerning the request - schema					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Message_Type	The identifier of the type of message	Text(Limited to set of known message_types)	S	M	See above types of messages list

5.2.2 Basic information block concerning the response

Basic information block concerning the response - schema					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Message_Type	The identifier of the type of message that the response refers to	Text	S	M	See above types of messages list
Error	Indicates the failure of the message reception	Boolean	S	M	0 – No 1- Yes
Errors	Array containing Error_Code, Error_Data (string), Error_Descr, Error_InternalID	Array of objects	S	M if Error = 1	
Code	Unique identifier of the message. Used for recall too.	Text(50)	S	M	
Checksum	The calculated checksum of the data received	Text(5000)	S	M	

5.2.2.1 Basic Error block description

Data Type	Description	Type	Example or regular expression
Error_Code	Error code describing the error.	Text(30)	
Error_Data	Text field containing error related data such as values of attributes, list of UIs For all lists, use the # character as separator.	Text(5000)	

37

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

Error_Descr	Description of the error code, that must contain the related controls, related RecallCode and fields when applicable.	Text(5000)	
Error_Internal ID	**Optional** internal ID of the error. This internal ID can be used for maintenance or audit purpose.	Text(50)	

5.2.3 Empty array and null values

5.2.3.1 Empty array

An array structure is represented as square brackets surrounding zero or more values (or elements)

JSON
```
{
   …
     "Errors": [],
   …
 }
```

5.2.3.2 Null value

A JSON null value MUST be a literal named null.

JSON
```
{
   …
     "Code": null,
   …
 }
```

5.2.4 Decimal points

According to JSON Standard RFC7159, a JSON decimal separator value MUST be a period ".".

JSON
```
{
   …
     "DecimalValue": 35.21
   …
 }
```

5.2.5 Common Error codes

HTTP status	Error Code	Error Description
401	SECURITY_INVALID_OR_EXPIRED_TOKEN	Invalid or Expired security token Please note that in this case the code or internal ID is not returned, as the message has not reached the processing service yet.
400	INVALID_SIGNATURE	Invalid signature
400	REQUIRED_FIELD_FAILED_VALIDATION MAX_LENGTH_FAILED_VALIDATION MIN_LENGTH_FAILED_VALIDATION ENTRY_LENGTH_FAILED_VALIDATION	When one or multiple fields do not contain a valid format.
400	PAYLOAD_NOT_UNIQUE	When the message has already been processed successfully.
400	INVALID_REQUEST_FORMAT	This error is returned when at least one of the mandatory fields are missing.
400	INVALID_MESSAGE_TYPE	When the field "Message_Type" is out of the defined list.
400	INVALID_INPUT_FORMAT	When the body of the message doesn't contain a valid JSON.
500	SYSTEM_ERROR	Internal system error.

6 Message Validation

6.1 Overview

Validation is the process to accept or reject an incoming message.

The UK Track & Trace system carries out real-time validations on two core aspects:

- **Technical validations** that focus on the structure of the information. Ensuring that the message is readable, and that the information contains all elements described in CIR 2018/574 and the technical specifications.

- **Business validations** that focus on the sequence of the events:

  a) **Application check**: for the application of unit level UIs, verify that the application of a UI takes place at the location associated with that UI at the time of its generation/issuing.
  b) **Code-pairing checks**: for each message, ensure the correct association of UK UIs with foreign UIs (e.g., EU or other RoW T&T regimes).
  c) **Aggregation checks**: for the aggregation of UIs, verify that an aggregated-level UI and all UIs at lower packaging levels relate to the same facility.
  d) **Activation and duplication check**: for each transmitted reporting event, verify that the UI is activated, not duplicated, and that the UI is not marked as stolen, lost, or destroyed.
  e) **Location check**: for the dispatch of tobacco products, verify that a product is dispatched from the same location at which it was previously stored; for the arrival of tobacco products, verify that a product is marked as dispatched and in transit.
  f) **Sequence check**: Considering the history of the UIs in an event, avoid the declaration of double dispatch of products or double arrival of products.

General functioning of validations:

Upon the receipt of reported events relating to one or multiple UI(s), a certain number of calculated information called metadata are computed and managed internally on the UK repository and Gateway. The UI metadata consists of

- The **UI state information** (indicating if the UI is in stock in a location, in transit and other)
- The **UI Location FID** (current location or last known location)
- The **UI Aggregation context** (if the UI is part of an aggregation)
- The **UI Event history**

This metadata is internally used by the UK Repository and the UK Gateway to perform each validation in real time, ensuring compliance with the required response times for acknowledgment messages as set out in the contractual KPIs.

Once the event is validated and accepted by the Gateway or the UK repository, the metadata for the respective UIs are updated.

6.2 Validation response

The Https status of the response provide information on the way the receiving system is handling the data.

- Http status **200 – 202**: the message is processed successfully

- Http status **299**: the message contains one or more warnings (validation failure), the message is processed even with the warnings (some processing might be limited). The response contains the list of failed validations.
- Http status **400**: the message contains one or more validation errors. The message will not be processed. An audit of the message and its corresponding response will be stored in the UK Data Repository. The response contains the list of failed validations.

Note that the HTTP status 299 is used in multiple situations. This warning status is used to informing the sending system of a UI failing the validations during the "Transition Period" (described later in the document). This status is also used in case a message is reported outside the 24hour (defined on the validation VAL_EVT_24H). The response message contains the list of validations failures that caused the warning.

6.3 Technical and business validation

The validation process is composed of a technical validation element and a business validation element. The technical validation ensures that the message follows the general format and message structure, including field availability and predefined values. This validation is performed on all components. The business validation is performed after the technical validation took place and it focuses on the business data as such. Both technical and business validation are executed in accordance with the rules of CIR 2018/574.

6.4 Timestamps

Control	Event Time	Message Time Long	Reception Time	Record Time
Source	Provided by EO system	Provided by EO system	Set by the UK Gateway	Established by the UK repository upon message consolidation
Precision	hour	second	milliseconds	milliseconds
		Used for sequence validation	Used for sequence validation as a replacement of the Message Time Long	Used for audit and reporting purposes
Note		Note: the Message time can be the Event Time on the second precision		

6.5 Type of validation

6.5.1 Security validation

The security validation is the first part of the technical validation ensuring that the connecting systems are authenticated and authorised to transmit data to the UK system.

Control	Description	Scope
VAL_SEC_HASH	Integrity check of the checksum	All messages
VAL_SEC_TOKEN	Oauth Security Token validation	All messages
VAL_SEC_CLAIM	Ensure that only Manufactures and Importers are able to submit the messages 3.1 (EUA) and 3.A1 (PAR)	EUA - PAR

6.5.2 Message Structure validation

The technical validation ensures that all incoming messages follow the technical guidelines. It also allows the UK system to access the message data accurately.

Control	Description	Scope
VAL_MSG_JSON	JSON structure validation	All JSON messages
VAL_MSG_TYPE	Message type validation	All messages
VAL_FIE_MAN	Mandatory Field validation (per message type)	All messages
VAL_FIE_FORMAT	Field format validation	All messages
VAL_FIE_REF	Existence of correctly reference enumerations. (As defined in data dictionary)	All messages

6.5.3 Message Transmission validation

The transmission controls are established to prevent technical duplication and processing of messages.

Control	Description	Scope
VAL_MSG_DUPLICATE	Message payload already processed successfully by the Gateway should be rejected.	*IDA – PAR - EUA – EPA – EDP – ERP- ETL- EUD- EVR – EIV – EPO – EPR*

6.5.4 Unique Identifiers validation

The following validations are performed on each UI that is present in a message.

6.5.4.1 Message level validation

The UI present in a message should be present only once. If the UI is present multiple times, the message will be rejected as non-compliant.

Control	Description	Scope
VAL_UI_MULT_MSG	Multiple duplicate UI present in the messages. For EPA (message 3.2), the validation on the parent UI will also be performed with the list of children in order to avoid first level cyclical reference.	*IDA – EUA – PAR - EPA – EDP – ERP- ETL- EUD- EVR – EIV – EPO - EPR*

6.5.4.2 Application Validation

Validate that the upUI(s) is only applied once to a upUI(L)

Control	Description	Scope
VAL_UI_EXIST_APP	The upUI(s) is not found in the UK Repository	*EUA – PAR - IDA*
VAL_UI_DUPLICATE_APP	Check if the upUI(s) has already been applied to a upUI(L)	*EUA - PAR*
VAL_UI_FID_APP	Validate that the FID inserted when requesting upUI(s) codes from the UK ID Issuer is matching the FID in the activation message.	*EUA*

6.5.4.3 Existence

The following control ensures that the UIs (upUI and aUI) comply exist and are valid and therefore can be used to report on product movement or transaction events.

Control	Description	Scope
VAL_UI_EXIST_UPUI	UI existence upUI exists and has been successfully applied.	*EPA (Children) – EDP – ERP- ETL- EVR - EIV – EPO - EPR*
VAL_UI_EXIST_AUI	aUI existence aUI has been aggregated (part of an EPA).	*IDA – EPA (Children) – EDP – ERP - ETL-*

Control	Description	Scope
		EVR - EIV – EPO - EPR
VAL_UI_EXIST_UPUI_SEQ	UI validity • upUI exists and has been successfully applied; • upUI has not been part of any deactivation message.	EPA (Children) – EDP – ERP – ETL – EVR - IDA
VAL_UI_EXIST_AUI_SEQ	aUI validity • aUI has been aggregated (part of an EPA); • and has not been disaggregated (including implicit disaggregation); • nor deactivated.	IDA – EPA (Children) – EDP – ERP - ETL-EVR

6.5.4.4 upUI Expiration

upUI(s) and aUIs that are issued by the UK ID Issuer and delivered to the Economic Operator are subject to a limited validity period.

Control	Description	Scope
VAL_UI_EXPIRY	Validate that the application or the aggregation date does not exceed the 6 months period from the time the UI was issued and transmitted to the Economic Operator.	EUA, EPA, PAR

6.5.4.5 UI level Message sequence validation

6.5.4.5.1 Sequence Validation overview

The following tables provide a summary overview of permitted UI events, based on the last event received for that specific UI.

Note that transactional events (4.x) are not described as there is no sequence validation implemented for those events.

Legend:

• **Yes (Green):** the message is accepted for that specific UI, no location validation is applied.
• **Yes (blue)**: the message is accepted, and a location validation control is applied.
• **No**: the message should be rejected

Control	Description	Scope

VAL_UI_ORD_SEQUENCE	The general sequence validation.	IDA – PAR - EUA – EPA – EDP – ERP – ETL – EUD - EVR

Message Received	PREVIOUS MESSAGE on the UI present in the received message								
	upUI GENERATED	aUI GENERATED	IDA 2.3	PAR (3.A1)	EUA 3.1	EUA 3.1 Import	EPA 3.2 parent UI	EPA 3.2 parent UI Import	EPA 3.2 Child
IDA 2.3	No	No	No	No	Yes	Yes	Yes	Yes	Yes
PAR 3.A1	Yes	No	No	No	No	No	No	No	No
EUA 3.1	Yes	No	No	Yes	No	No	No	No	No
EUA 3.1 Import	Yes	No	No	Yes	No	No	No	No	No
EPA 3.2 parent UI	No	Yes	No	No	No	No	No	No	No
EPA 3.2 child UI (upUI)	No	No	No	No	Yes	Yes	No	No	Yes
EPA 3.2 child UI (aUI)	No	No	No	No	No	No	Yes	Yes	Yes
EDP 3.3 Export (type 1)	No	No	No	No	Yes	No	Yes	No	Yes
EDP 3.3 (type 2)	No	No	No	No	Yes	No	Yes	No	Yes
EDP 3.3 VM (type 3)	No	No	No	No	Yes	No	Yes	No	Yes
EDP 3.3 VV (type 4)	No	No	No	No	Yes	No	Yes	No	Yes
ERP 3.4	No	No	No	No	No	Yes	No	Yes	No
ERP 3.4 (Return)	No	No	No	No	No	No	No	No	No
ETL 3.5	No	No	No	No	No	No	No	No	No
ETL 3.5 (Export)	No	No	No	No	No	No	No	No	No
EUD 3.6	No	No	No	No	No	No	Yes	Yes	Yes
EVR 3.7	No	No	No	No	No	No	No	No	No
EIV 4.1	No	No	Yes		Yes	Yes	Yes	Yes	Yes
EPO 4.2	No	No	Yes		Yes	Yes	Yes	Yes	Yes
EPR 4.3	No	No	Yes		Yes	Yes	Yes	Yes	Yes

Message Received	PREVIOUS MESSAGE on the UI present in the received message										
	EDP 3.3 (type 1) Export	EDP 3.3 (type 2)	EDP 3.3 (types 3) VM	EDP 3.3 (type 4) VV	ERP 3.4	ERP 3.4 (Return)	ETL 3.5	ETL 3.5 Export	EUD 3.6	EUD 3.6 (aUI implicitly disaggregated) - reuse of aUI	EVR 3.7
IDA 2.3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PAR 3.A1	No	No	No	No	No	No	No	No	No	No	No
EUA 3.1	No	No	No	No	No	No	No	No	No	No	No
EUA 3.1 Import	No	No	No	No	No	No	No	No	No	No	No
EPA 3.2 parent UI	No	No	No	No	No	No	No	No	Yes	No	No
EPA 3.2 child UI (upUI)	No	No	No	No	Yes	Yes	No	No	No	No	No
EPA 3.2 child UI (aUI)	No	No	No	No	Yes	Yes	No	No	No	No	No
EDP 3.3 Export (type 1)	No	No	No	No	Yes	Yes	No	No	No	No	No
EDP 3.3 (type 2)	No	No	No	No	Yes	Yes	No	No	No	No	No
EDP 3.3 VM (type 3)	No	No	No	No	Yes	Yes	No	No	No	No	No
EDP 3.3 VV (type 4)	No	No	No	No	Yes	Yes	No	No	No	No	No
ERP 3.4	No	Yes	No	No	No	No	Yes	No	No	No	No
ERP 3.4 (Return)	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes
ETL 3.5	No	Yes	No	No	No	No	Yes	No	No	No	No
ETL 3.5 (Export)	Yes	No	No	No	No	No	No	Yes	No	No	No
EUD 3.6	No	No	No	No	Yes	Yes	No	No	No	Yes	No
EVR 3.7	No	No	No	Yes	No	No	No	No	No	No	No
EIV 4.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
EPO 4.2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

EPR 4.3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

For dispatch event 3.3 EDP, type refers to the Destination_ID1 field:

- Type 1 – Non-UK dest.
- Type 2 – UK destination other than VM – fixed quantity delivery
- Type 3 – UK VM(s)
- Type 4 – UK destination other than VM – delivery with VV

Type 'return' in ERP (3.4) is based on the Product_Return field:

- 0 – No
- 1– The arrival is of the type product return

The field Destination_ID1 in the ETL (3.5) event indicates if the ETL concerns an export event or is destined for a UK location:

- 0 – No
- 1 – Yes

6.5.4.5.2 Application and deactivation sequence validation

Control	Description	Scope
VAL_UI_ORD_REACTIVATION	upUI(s) that has been deactivated should not allow any application event (EUA).	*EUA*
VAL_UI_ORD_DEACTIVATED	UI – presence of UI in a message after being deactivated.	*EPA – PAR - EDP – ERP- ETL- EUD- EVR - IDA*

6.5.4.5.3 Aggregation and Disaggregation Principles

Principle 1: All aggregation events are full aggregations. Once the aUI is aggregated, a subsequent aggregation event on the same parent aUI should be rejected.

Principle 2: All disaggregation events are full disaggregations. Once the aUI is disaggregated, no movement should be reported on that aUI (until the next disaggregation and aggregation event are reported.).

Principle 3: Implicit disaggregation. Disaggregation event reporting is mandatory only in advance of when an aUI is reused in a subsequent aggregation event (as a parent aUI). The implicit disaggregation is detected when at least one child UI is reported in an aggregation or product movement event. The parent aUI of this child UI will be considered as disaggregated. In the case that the child UI is part of an aggregation hierarchy, all parent aUIs will be disaggregated.

Consequently, once an implicit disaggregation is detected, no movement should be reported on that parent aUI (until the next aggregation event is reported).

Principle 4: All disaggregation must be performed at a registered location. Disaggregation during transport is not permitted.

Note that implicit disaggregation might be triggered during transit by

- a deactivation event IDA (2.3),
- a delivery carried out by vending van EVR (3.7),
- or an arrival ERP (3.4) of type Product Return.

6.5.4.5.4 Implicit disaggregation

Disaggregation events are only required when an economic operator intends to reuse the parent aUI. Therefore, it is likely that certain product movement messages will be reported at lower-level hierarchy, triggering an implicit disaggregation event.

6.5.4.5.4.1 Triggers

The implicit disaggregation will be detected / triggered when a given UI, belonging to a lower-level hierarchy in the overall aggregation tree, is identified in one of the following messages: IDA (2.3), EPA (3.2), EDP (3.3), ERP (3.4) of type Return, EUD (3.6) and EVR (3.7).

Note that transactional events and code pairing events do not trigger any implicit disaggregation.

Example of implicit disaggregation triggered by EDP (3.3):
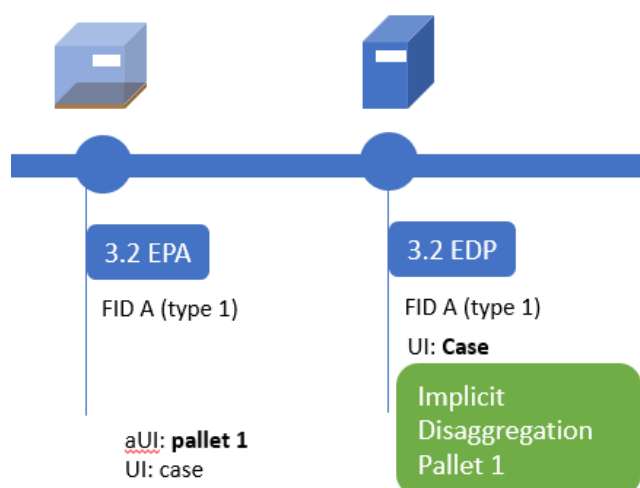
Figure 6 Implicit disaggregation trggered by EDP (3.3)

Example of implicit disaggregation triggered by EVR (3.7):
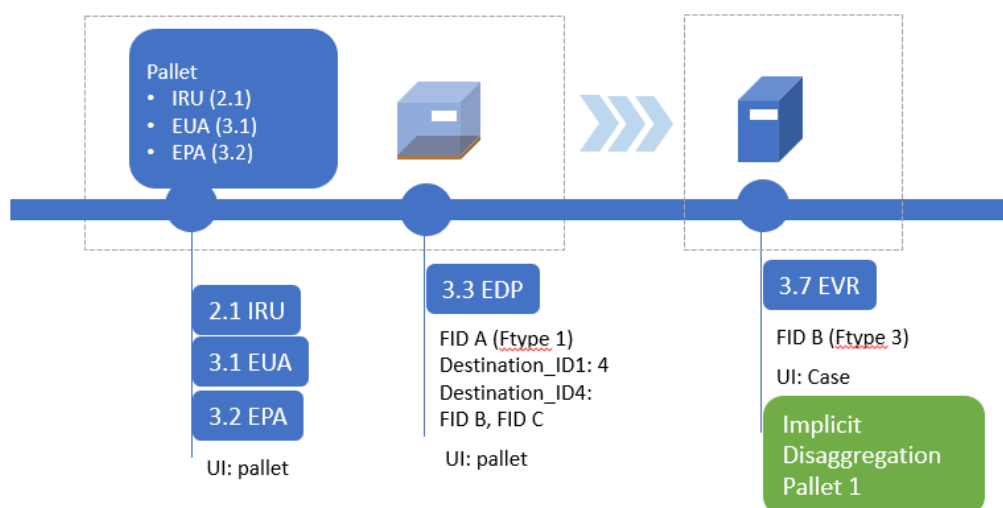
Figure 7 Implicit disaggregation triggered by EVR (3.7)

6.5.4.5.4.2 Disaggregated state

Once the UI is implicitly disaggregated, the UI should be considered disaggregated and should not be part of any subsequent product movement events (3.x). The definition of implicit disaggregation is that the affected parent aUI is considered completely unlinked from all children UIs that it was associated with at the time of the implicit disaggregation trigger. If a reported UI has more than one linked parent at the time of reporting, all parents will be implicitly disaggregated, including grand-parents and recursively upwards linked UIs until the upmost level.

6.5.4.5.4.3 Recall

It is important to note that implicit disaggregation will be rolled back in case the event that triggered it is recalled.

6.5.4.5.4.4 Explicit disaggregation after implicit disaggregation

It is possible to disaggregate an aUI with message EUD (3.6) after it was part of an implicit disaggregation.

6.5.4.5.5 Aggregation and Disaggregation Validation

Control	Description	Scope
VAL_UI_ORD_AGG_MULT	Validation that a parent UI has not been part of any prior aggregation event (as parent) without being part of a disaggregation event. This control prevents the reuse of an aUI without prior explicit disaggregation.	*EPA*
VAL_UI_ORD_DISAGG	Validation that an aUI that has been disaggregated cannot be part on any product movement	*EDP - ERP – ETL – EVR – EPA (Children)*

	prior of being aggregated as a parent.	
VAL_UI_ORD_IMPLDISAGG	Validation that an aUI that has been implicitly disaggregated cannot be part of any product movement prior of being part of an EUD message and then aggregated as parent with an EPA.	*EDP – ERP – ETL – EVR- EPA (Children)*

6.5.4.5.6 Location based Validation

6.5.4.5.6.1 UI Location update

The sequence validation also considers the implementation of location-based controls.

FID of the UI is updated upon processing of the following events.

- ERP (3.4)
- EPA (3.2) update of the newly created parent id
- EVR (3.7) In case of implicit disaggregation, only the UIs present in the event will have the location metadata updated to the retail outlet.
- RCL (5.0) rolls back to the previous state

6.5.4.5.6.2 Location based controls

Control	**Description**	**Scope**
VAL_UI_ORD_AGG_FID	Validation that the aggregation and the disaggregation events must happen at the same facility (FID) where the products have been either created or arrived. **Exception**: The validation is not applicable for products before being imported into the UK.	*EPA - EUD*

6.5.4.5.6.3 Location validation exception for Imports

Due to the fact that product movement outside of the UK are not subject to being reported into the UK Gateway (i.e. Dispatch/Arrivals), in the event that an Economic Operator requires to modify the hierarchy of the goods before the import into UK, there will be Aggregation (3.2 – EPA) and Disaggregation (3.6 – EUD) messages on different locations (Facility IDs) without a Dispatch/Arrival movement between them.

For these cases, the Location validation (VAL_UI_ORG_AGG_FID) will not be applicable (so all products BEFORE the first Arrival 3.4 into the UK Territory may be modified in terms of their logistic hierarchy).

6.5.4.5.7 Dispatch and arrival Validation

Principle 5: Reporting on the arrival event should be associated with the same UI that was previously reported during the Dispatch/Transloading process. This is a consequence

of principle 4. This means that an Arrival event that contains child UIs of an UI reported during the Dispatch/Transloading Event will be rejected. The same UI must be reported.

_Exception VAL_UI_ORD_ARRIVAL_RETURN:_ arrival of type return skips the principle 5 but still enforces the sequence validation (there must be a prior Dispatch or Transloading event for the UIs)

Control	Description	Scope
VAL_UI_ORD_ARRIVAL	Validation that a UI is part of a prior reported dispatch or transloading event (EDP 3.3, ETL 3.5). This validation concerns the sequence of events. _Exception:_ Imported products	_ERP_ _(Product_Return = 0)_
VAL_UI_ORD_ARRIVAL_RETURN	Validation that a UI is part of a prior reported dispatch or transloading event (EDP 3.3, ETL 3.5, EVR 3.7). This validation concerns the sequence of events. In this validation principle 5 doesn't apply.	_ERP_ _(Product_Return = 1)_

6.5.4.5.7.1 Imported goods exception

When the message request 2.1 (request upUIs) is sent to the UK ID Issuer, the message contains the import flag information. This flag will be used to assess if the exception should be implemented.

a) Unit packs that contain the import flag information and that have been applied with an EUA (message 3.1) and eventually aggregated with an EPA (message 3.2) in a facility whose country is **outside** the UK must be reported in an arrival message with a facility whose country is inside the UK (the physical importation of the goods) before being part of any aggregation or logistic movement inside the UK.

b) Unit packs that contain the import flag information and that have been applied with an EUA (message 3.1) and eventually aggregated with an EPA (message 3.2) in a facility whose country is **inside** the UK, must not report an arrival message. In this case, it is understood that the physical import should have taken place between the generation of the codes (message 2.1) and the application/aggregation of the UIs with EUA (message 3.1) and EPA (message 3.2) events.

6.5.4.5.7.2 Arrival of type return

The reporting of the arrival of type return can be performed on child UIs. This operation is allowed and will trigger an implicit disaggregation as long as the UIs have a prior Dispatch or Transloading event.

6.5.4.5.7.3 Arrival and Return

Flows from the retail outlets, even in the context of message 3.7, are considered to constitute "product returns".

The flows constitute "product returns" even if the product is returned to the same facility as it was originally dispatched from or to another one.

6.5.4.5.7.4 Dispatch validation

Control	Description	Scope
VAL_UI_ORD_DISPATCH	Validation that a UI's last location (FID) matches the source location (FID) of the dispatch event. The UI must have been: - applied or aggregated at that specific location (FID) - arrived at that location.	*EDP*

6.5.4.6 Message Event Time Validation

The following messages validation compares the event time (Event Time) to the actual reception time of the event by the first point of entry.

Control	Description	Scope
VAL_EVT_24H	Validation that the Events are reported within 24 hours from the occurrence of the event. This validation is performed on the Event Time compared to the Reception Time of the UK Gateway. NOTE: this validation will be reduced to 3 hours after 20 May 2028 **Failing this validation will not trigger a negative acknowledgement but rather generate a warning to the sender system**	*EUA – EPA – EVR – EIV – EPO – EPR*

VAL_EVT_TIME	"Within 24 hours prior to the occurrence of the event" rule for dispatch and trans-loading event messages is a rule and the system shall reject non-compliant messages. Control is based on the "Reception Time of the UK Gateway – Event Time of the event" time difference. **Failing this validation will not trigger a negative acknowledgement but rather generate a warning to the sender system.**	*EDP – ETL*

6.5.4.7 Identifier Code validation

Identifier codes (IDs) are used in several messages. Validating the existence and active status of IDs is part of the business validation, as described in the table below.

Control	Description	Scope
VAL_ENT_EXIST_EOID	Check if the EOID exists in the UK registry	*IDA – PAR - EUA – EPA – EDP – ERP – ETL – EUD - EVR*
VAL_ENT_EXIST_FID	Check if FID, exists in the UK registry	*EUA – EPA - EDP – ERP – ETL – EUD - EVR*
VAL_ENT_ACTIVE_EOID	Check if EOID is marked as active in the repository	*IDA – PAR - EUA – EPA – EDP – ERP – ETL – EUD- EVR*
VAL_ENT_ACTIVE_FID	Check if FID is marked as active in the repository	*EUA – EPA*

6.5.4.8 Recall Validation

6.5.4.8.1 General recall rules

The sequence validation on the product movement introduces additional controls on the recall process. To maintain the consistency of the history of the UI, only the recall of the last event for each UI will be authorised.

If a message to be recalled contains a UI (any in the reported list of the event that refers to) that is also associated with subsequent event(s), the subsequent event(s) must be recalled first.

For the sake of clarity, the following scenarios describe the process of recall:

An EO reports two product movements on an UI (Event 1 and Event 2). If the EO wishes to recall Event 1, the EO has first to recall Event 2 and only afterwards the recall of

Event 1 will be possible. Moreover, Event2 must be the last event that occurred on all UIs contained in Event2, otherwise Event2 cannot be recalled.

6.5.4.8.2 Transaction events

Transaction events (4.x) are not subject to this rule as they are not impacted by the sequence validation control and therefore, transaction events (4.x) can be recalled at any time

6.5.4.8.3 Recall Validation Controls

Control	Description	Scope
VAL_RECALL_EXIST	Check if RecallCode exists	*RCL*
VAL_RECALL_LAST	Check if for all UIs related to the event identified by RecallCode is the very last unrecalled event occurred on all such UI including related implicitly disaggregated parents.	*RCL*

6.5.4.9 Code-Pairing Validation

The code pairing flow from the UK Gateway point of view is expected to be:

1) The UK ID Issuer generates UK UIDs (upUIs)
2) The UK Gateway receives the code-pairing message (PAR) with the UK UIDs and the foreign printed codes
3) The UK Gateway receives the application message (EUA) for the foreign printed codes

The following validations are implemented:

Control	Description	Scope
VAL_PRINTED_CODES_ALREADY_USED	Check that the Printed Codes are not already paired with UK issued UIDs.	*PAR*
VAL_PAIRED_CODES_ALREADY_USED	Check that the UK issued UIDs are not already paired with other Printed Codes.	*PAR*

6.5.5 Validation Scope

	IDA (2.3)	EUA (3.1)	PAR (3A.1)	EPA (3.2)	EDP (3.3)	ERP (3.4)	ETL (3.5)	EUD (3.6)	EVR (3.7)	EIV (4.1)	EPO (4.2)	EPR (4.3)	RCL (5)
Technical validation													
VAL_SEC_HASH	X	X	X	X	X	X	X	X	X	X	X	X	X

	1	2	3	4	5	6	7	8	9	10	11	12	13
VAL_SEC_TOKEN	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_SEC_CLAIM		X	X										
VAL_MSG_JSON	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_MSG_TYPE	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_FIE_MAN	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_FIE_FORMAT	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_FIE_REF	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_MSG_DUPLICATE	X	X	X	X	X	X	X	X	X	X	X	X	X
VAL_UI_MULT_MSG	X	X	X	X	X	X	X	X	X	X	X	X	
Business rule validation													
UI creation													
VAL_UI_EXIST_APP	X	X	X										
VAL_UI_DUPLICATE_APP		X	X										
VAL_UI_EXIST_UPUI				X	X	X	X		X	X	X	X	
VAL_UI_EXIST_AUI				X	X	X	X	X	X	X	X	X	
VAL_UI_EXIST_UPUI_SEQ	X			X	X	X	X		X				
VAL_UI_EXIST_AUI_SEQ	X			X	X	X	X	X	X				
VAL_UI_EXPIRY		X	X	X									
Entity Validation													
VAL_ENT_EXIST_EOID	X	X	X	X	X	X	X	X	X	X	X	X	
VAL_ENT_EXIST_FID		X		X	X	X	X	X	X				
VAL_ENT_ACTIVE_EOID	X	X	X	X	X	X	X	X	X				
VAL_ENT_ACTIVE_FID	X	X		X	X	X	X	X	X				
Pairing Validation													
VAL_PRINTED_CODES_ALREADY_USED			X										
VAL_PAIRED_CODES_ALREADY_USED			X										
Sequence Validation													
VAL_UI_FID_APP		X											
VAL_UI_ORD_REACTIVATION		X											
VAL_UI_ORD_DEACTIVATED	X			X	X	X	X	X	X				
VAL_UI_ORD_AGG_MULT			X										

VAL_UI_ORD_DISAGG					X	X	X		X				
VAL_UI_ORD_IMPLDISAGG				X	X	X	X		X				
VAL_UI_ORD_AGG_FID				X				X					
VAL_UI_ORD_ARRIVAL					X								
VAL_UI_ORD_ARRIVAL_RETURN					X								
VAL_UI_ORD_DISPATCH				X									
VAL_UI_ORD_SEQUENCE	X	X	X	X	X	X	X	X	X				
Message Timing													
VAL_EVT_24H		X	X	X					X	X	X	X	
VAL_EVT_TIME				X			X						
Recall													
VAL_RECALL_EXIST													X
VAL_RECALL_LAST													X

6.5.6 Validation and Error Code

	Error Code	http Status
Technical validation		
VAL_SEC_HASH	INVALID_SIGNATURE	400
VAL_SEC_TOKEN	INVALID_OR_EXPIRED_TOKEN	401
VAL_MSG_JSON	INVALID_INPUT_FORMAT REQUIRED_FIELD_FAILED_VALIDATION MAX_LENGTH_FAILED_VALIDATION MIN_LENGTH_FAILED_VALIDATION ENTRY_LENGTH_FAILED_VALIDATION EXCISE_NUMBER_NOT_VALID NON_COMPATIBLE_UIS NOT_THE_SAME_NUMBER_OF_ITEMS	400
VAL_MSG_TYPE	FAILED_VALIDATION	400
VAL_FIE_MAN	FAILED_VALIDATION	400
VAL_FIE_FORMAT	INVALID_INPUT_FORMAT	400
VAL_FIE_REF	FAILED_VALIDATION	400

VAL_MSG_DUPLICATE	PAYLOAD_NOT_UNIQUE	400
VAL_UI_MULT_MSG	MULTIPLE_UI UI_NOT_VALID UIS_NOT_VALID	400
Business rule validation		
UI creation		
VAL_UI_EXIST_APP	UIS_APPLICATION_ERROR	400
VAL_UI_DUPLICATE_APP	UIS_APPLICATION_ERROR	400
VAL_UI_EXIST_UPUI	UI_NOT_EXIST UI_NOT_VALID	400
VAL_UI_EXIST_AUI	UI_NOT_EXIST	400
VAL_UI_EXIST_UPUI_SEQ	UI_NOT_VALID	400
VAL_UI_EXIST_AUI_SEQ	UI_NOT_EXIST	400
VAL_UI_EXPIRY	UI_EXPIRED	400
Entity Validation		
VAL_ENT_EXIST_EOID	EOID_NOT_EXIST_OR_ACTIVE	400
VAL_ENT_EXIST_FID	FID_NOT_EXIST_OR_ACTIVE	400
VAL_ENT_ACTIVE_EOID	EOID_NOT_EXIST_OR_ACTIVE	400
VAL_ENT_ACTIVE_FID	FID_NOT_EXIST_OR_ACTIVE	400
Pairing Validation		
VAL_PRINTED_CODES_ALREADY_USED	PRINTED_CODES_ALREADY_USED	400
VAL_PAIRED_CODES_ALREADY_USED	PAIRED_CODES_ALREADY_USED	400
Sequence Validation		
VAL_UI_FID_APP	FID_MISMATCH	400
VAL_UI_ORD_REACTIVATION	UI_DEACTIVATED	400
VAL_UI_ORD_DEACTIVATED	UI_DEACTIVATED	400
VAL_UI_ORD_AGG_MULT	MULTIPLE_AGGREGATION	400
VAL_UI_ORD_DISAGG	UI_ALREADY_DISAGGREGATED	400

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

57

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

VAL_UI_ORD_IMPLDISAGG	UI_ALREADY_DISAGGREGATED	400
VAL_UI_ORD_AGG_FID	LOCATION_MISMATCH	400
VAL_UI_ORD_ARRIVAL	ARRIVAL_NOTALLOWED	400
VAL_UI_ORD_ARRIVAL_RETURN	ARRIVAL_NOTALLOWED	400
VAL_UI_ORD_DISPATCH	LOCATION_MISMATCH	400
VAL_UI_ORD_SEQUENCE	UI_SEQUENCE_ERROR	400
Message Timing		
VAL_EVT_24H	OPERATION_WITHIN_24_HOURS	299
VAL_EVT_TIME	SHIPMENT_WITHIN_24_HOURS	299
Recall		
VAL_RECALL_EXIST	CODE_NOT_EXIST CODE_NOT_UNIQUE	400
VAL_RECALL_LAST	RECALL_NOT_LAST_EVENT	400

6.6 Period of Grace for validation rules

Validation rules set out in this document are implemented from 1 July 2022. However, to facilitate the transition between De La Rue (current Supplier) and Dentsu (new Supplier), a **Period of Grace** will be implemented during which certain validations are 'relaxed'. These will therefore not result in negative acknowledgements that would block the concerned products from moving forward in the supply chain. Instead, these 'relaxed validations' will return WARNING messages explaining to the sending Economic Operators the identified validation issue.

Validation rules for upUIs/aUIs and traceability messages are applied and enforced according to the following general principle:

For UIDs issued/created before 1 July 2022	For UIDs issued/created between 1 July 2022 and 31 Dec 2022	For UIDs issued/created as of 1 Jan 2023
Most technical validations result in warning messages and only a few technical validations are enforced with error messages.	Most technical validations are enforced with error messages and only a few technical validation result in warning messages.	All technical validations are enforced.

All business validations result in warning messages.	Most business validations result in warning messages and only a few business validations are enforced with error messages.	All business validations are enforced.

Note: there is no period of grace for identifier code (EOID, FID, MID) data validation.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

59

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

The tables below outline the applicability of the different validation rules prior to, during, and after the Period of Grace.

Control	Description	Scope	Pre-Grace Period	Grace Period	Post-Grace Period
VAL_SEC_HASH	Integrity check of the checksum	All messages	ERROR	ERROR	ERROR
VAL_SEC_TOKEN	Oauth Security Token validation	All messages	ERROR	ERROR	ERROR
VAL_MSG_JSON	JSON structure validation	All messages	ERROR	ERROR	ERROR
VAL_MSG_TYPE	Message type validation	All messages	ERROR	ERROR	ERROR
VAL_FIE_MAN	Mandatory Field validation (per message type)	All messages	ERROR	ERROR	ERROR
VAL_FIE_FORMAT	Field format validation	All messages	ERROR	ERROR	ERROR
VAL_FIE_REF	Existence of correctly reference enumerations. (As defined in data dictionary)	All messages	ERROR	ERROR	ERROR
VAL_UI_MULT_MSG	Multiple duplicate UI present in the messages. For EPA (message 3.2), the validation on the parent UI will also be performed with the list of children in order to avoid first level cyclical reference.	All messages	ERROR	ERROR	ERROR
VAL_MSG_DUPLICATE	Message payload already processed successfully by the Gateway should be rejected.	All messages	ERROR	ERROR	ERROR
VAL_UI_EXIST_APP	The upUI(s) is not found in the UK Repository	EUA (3.1) – PAR (3.A1) – IDA (2.3)	WARNING	ERROR	ERROR
VAL_UI_ORD_AGG_MULT	Validation that a parent UI has not been part of any prior aggregation event (as parent) without being part of a disaggregation event.	EPA (3.2)	WARNING	WARNING	ERROR

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

60

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

	This control prevents the reuse of an aUI without prior explicit disaggregation.				
VAL_UI_FID_APP	Validate that the FID inserted when requesting upUI(s) codes from the UK ID Issuer is matching the FID in the activation message.	EUA (3.1)	**WARNING**	**ERROR**	**ERROR**
VAL_UI_EXIST_UPUI	upUI existence upUI exists and has been successfully applied.	EPA (3.2) (Only Children) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7) – EIV (4.1) – EPO (4.2) – EPR (4.3)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_EXIST_AUI	aUI existence aUI has been aggregated (part of an EPA).	IDA (2.3) – EPA (3.2) (Children) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7) – EIV (4.1) – EPO (4.2) - EPR (4.3)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_EXIST_UPUI_SEQ	upUI exists and has been successfully applied; upUI has not been part of any deactivation message.	EPA (3.2) (Children) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7) – IDA (2.3)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_EXIST_AUI_SEQ	aUI has been aggregated (part of an EPA); and has not been disaggregated (including implicit disaggregation); nor deactivated	IDA (2.3) – EPA (3.2) (Children) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7)	**WARNING**	**WARNING**	**ERROR**
VAL_ENT_EXIST_EOID	Check if the EOID exists in the UK registry	IDA (2.3) – PAR (3.A1) – EUA (3.1) – EPA (3.2) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EUD (3.6) – EVR (3.7)	**ERROR**	**ERROR**	**ERROR**
VAL_ENT_EXIST_FID	Check if FID, exists in the UK registry	IDA (2.3) – PAR (3.A1) – EUA (3.1) – EPA (3.2) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EUD (3.6) – EVR (3.7)	**ERROR**	**ERROR**	**ERROR**
VAL_ENT_ACTIVE_EOID	Check if EOID is marked as active in the repository	IDA (2.3) – PAR (3.A1) – EUA (3.1) – EPA (3.2) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EUD	**ERROR**	**ERROR**	**ERROR**

		(3.6) – EVR (3.7)			
VAL_ENT_ACTIVE_FID	Check if FID is marked as active in the repository	EUA (3.1) – EPA (3.2)	**ERROR**	**ERROR**	**ERROR**
VAL_UI_EXPIRY	Validate that the application or the aggregation date does not exceed the 6 months period from the time the UI was issued and transmitted to the Economic Operator.	EUA (3.1), EPA (3.2), PAR (3.A1)	**ERROR**	**ERROR**	**ERROR**
VAL_UI_ORD_DEACTIVATED	UI – presence of UI in a message after being deactivated.	EPA (3.2) – PAR (3.A1) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EUD (3.6) – EVR (3.7) – IDA (2.3)	**WARNING**	**ERROR**	**ERROR**
VAL_UI_ORD_REACTIVATION	upUI(s) that has been deactivated should not allow any application event (EUA).	EUA (3.1)	**WARNING**	**ERROR**	**ERROR**
VAL_UI_ORD_DISAGG	Validation that an aUI that has been disaggregated cannot be part on any product movement prior of being aggregated as a parent.	EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7) – EPA (3.2) (Children)	**WARNING**	**ERROR**	**ERROR**
VAL_UI_ORD_IMPLDISAGG	Validation that an aUI that has been implicitly disaggregated cannot be part of any product movement prior of being part of an EUD message and then aggregated as parent with an EPA.	EDP (3.3) – ERP (3.4) – ETL (3.5) – EVR (3.7) – EPA (3.2) (Children)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_ORD_AGG_FID	Validation that the aggregation and the disaggregation events must happen at the same facility (FID) where the products have been either created or arrived.	EPA (3.2) – EUD (3.6)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_ORD_DISPATCH	Validation that a UI's last location (FID) matches the source location (FID) of the dispatch event. The UI must have been: - applied or aggregated at that specific location (FID) - arrived at that location.	EDP (3.3)	**WARNING**	**WARNING**	**ERROR**

VAL_UI_ORD_SEQUENCE	The general sequence validation.	IDA (2.3) – PAR (3.A1) – EUA (3.1) – EPA (3.2) – EDP (3.3) – ERP (3.4) – ETL (3.5) – EUD (3.6) – EVR (3.7)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_ORD_ARRIVAL	Validation that a UI is part of a prior reported dispatch or transloading event (EDP 3.3, ETL 3.5). This validation concerns the sequence of events. Exception: Imported products	ERP (3.4) (Product_Return = 0)	**WARNING**	**WARNING**	**ERROR**
VAL_UI_ORD_ARRIVAL_RETURN	Validation that a UI is part of a prior reported dispatch or transloading event (EDP 3.3, ETL 3.5, EVR 3.7). This validation concerns the sequence of events. In this validation principle 5 doesn't apply.	ERP (3.4) (Product_Return = 1)	**WARNING**	**WARNING**	**ERROR**
VAL_EVT_TIME	"Within 24 hours prior to the occurrence of the event" rule for dispatch and trans-loading event messages is a rule and the system shall reject non-compliant messages. Control is based on the "Reception Time of the UK Gateway – Event Time of the event" time difference. Failing this validation will not trigger a negative acknowledgement but rather generate a warning to the sender system.	EDP (3.3) – ETL (3.5)	**WARNING**	**WARNING**	**WARNING**
VAL_EVT_24H	Validation that the Events are reported within 24 hours from the occurrence of the event. This validation is performed on the Event Time compared to the Reception Time of the UK Gateway. NOTE: this validation will be reduced to 3	EUA (3.1) – EPA (3.2) – EVR (3.7) – EIV (4.1) – EPO (4.2) – EPR (4.3)	**WARNING**	**WARNING**	**WARNING**

	hours after 20 May 2028 Failing this validation will not trigger a negative acknowledgement but rather generate a warning to the sender system				
VAL_RECALL_EXIST	Check if RecallCode exists	RCL (5)	**ERROR**	**ERROR**	**ERROR**
VAL_RECALL_LAST	Check if for all UIs related to the event identified by RecallCode is the very last unrecalled event occurred on all such UI including related implicitly disaggregated parents.	RCL (5)	**ERROR**	**ERROR**	**ERROR**
VAL_SEC_CLAIM	Check that caller is allowed to call this method (Only manufacturers or importers can send code activations and code pairings to the UK Gateway)	EUA (3.1) – PAR (3.A1)	**ERROR**	**ERROR**	**ERROR**
VAL_PRINTED_CODES_ALREADY_USED	Check that the Printed Codes are not paired already with UK Issued UIDs.	PAR (3.A1)	**WARNING**	**ERROR**	**ERROR**
VAL_PAIRED_CODES_ALREADY_USED	Check that the UK Issued UIDs are not paired already with other Printed Codes.	PAR (3.A1)	**WARNING**	**ERROR**	**ERROR**
VAL_UI_DUPLICATE_APP	Check if the upUI(s) have already been applied	EUA (3.1) – PAR (3.A1)	**WARNING**	**ERROR**	**ERROR**

6.7 Unique identifiers Messages

6.7.1 IDA – (2.3) Request for deactivation of UIs

6.7.1.1 Description

Changes the status of the UIs listed in the request to "deactivated"

The hierarchy below these UIs will be managed depending on the deactivation reason for the parent UI.

6.7.1.1.1 Product deactivation

If the deactivation reason was Deact_Reason1 = 1 (Product destroyed) or 2 (Product stolen) then the full hierarchy is deactivated.

Example of a pallet product deactivation:
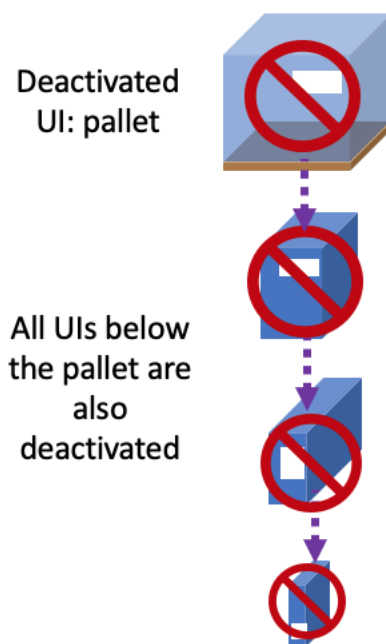
Figure 8 Hierarchy deactivation triggered by an IDA with Deact_Reason1 = 1 or 2

6.7.1.1.2 UI deactivation

If the deactivation reason was Deact_Reason1 = 3 (UI Destroyed), 4 (UI Stolen), 5 (UI Unused) or 6 (Other), then only the explicitly mentioned UIs are deactivated and therefore the related hierarchy UIs would still exist in the UK repository as active. It is the responsibility of the Economic Operator to report on this event once they become aware of the particular event (send a deactivation message for the stolen UIs)

Note that in these cases the Deactivation will also trigger an implicit disaggregation.

Example of UI deactivation for a pallet:
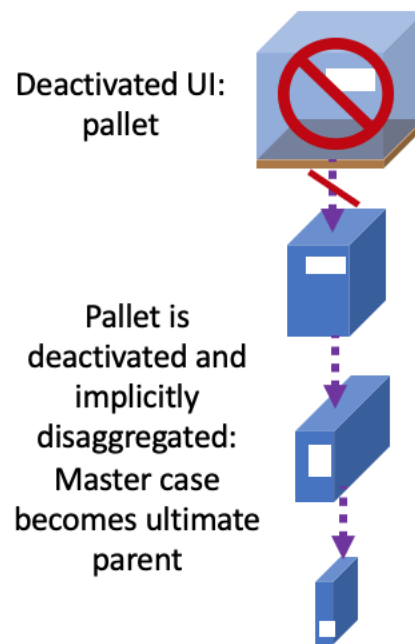
Figure 9 Implicit disaggregation triggered by a IDA Deact_Reason1 = 3,4,5 or 6

6.7.1.2 Description of the fields

request for the deactivation of UIs – request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = IDA
Event_Time	Intended time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Deact_Type	Deactivation of unit packet or aggregated level UIs	Integer	S	M	1 – Unit pack level UIs 2 – Aggregated level UIs
Deact_Reason1	Identification of the reason for deactivation	Integer	S	M	See DeactivationReasonType
Deact_Reason2	Description of other reason	Text(5000)	S	M, if Deact_Reason1	

request for the deactivation of UIs – request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
				= 6 (other reason)	
Deact_Reason3	Additional description of the reason	Text (Limited to the set of known deactivation_types)	S	O	
Deact_upUI	List of unit packet level UIs to be deactivated	upUI(s)	M	M, if Deact_Type = 1	
Deact_aUI	List of aggregated level UIs to be deactivated	aUI	M	M, if Deact_Type = 2	

6.7.1.2.1 Response

request for the deactivation of UIs – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = IDA

6.7.1.1 Business Validation

	IDA (2.3)
Business rule validation	
UI creation	
VAL_UI_EXIST_APP	Deact_upUI
VAL_UI_EXIST_UPUI_SEQ	Deact_upUI
VAL_UI_EXIST_AUI_SEQ	Deact_aUI
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_ACTIVE_EOID	EO_ID
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	Deact_upUI, Deact_aUI

6.7.1.2 Implicit disaggregation trigger

The deactivation event can trigger an implicit disaggregation when a child UI is identified as part of the event.

6.7.1.3 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "Event_Time" : "22032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Deact_Type": 1,
  "Deact_Reason1": 1,
  "Deact_Reason2": "reason one",
  "Deact_Reason3": "reason two",
  "Deact_upUI": [ "DANXXXXXXXXXXXX1PR0123456789" ],
  "Deact_apUI": [],
  "Message_Type": "IDA"
}
```

6.7.1.4 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "IDA",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

6.7.1.5 Error response sample

Processing errors

HTTP status	Error Code	Error Description
<< Common response code >>		
400	UIS_APPLICATION_ERROR	VAL_UI_EXIST_APP
400	UI_NOT_EXIST	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID

400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	UI_SEQUENCE_ERROR	VAL_UI_ORD_SEQUENCE

6.7.2 PAR - (3.A1) Issuing of code pairing message

6.7.2.1 Description

Where a manufacturer or importer intends to leverage a foreign code to be printed onto a unit packet, the following event schedule must be observed:

After the manufacturer or importer received from the UK ID Issuer the UK UIs, and prior to the submission of the application message (3.1 – EUA), the following code-pairing message must be transmitted to the UK System (that is, to the UK Gateway who validates and forwards to the UK repository).

This message will trigger the creation of the link between the UK code and the foreign code (printed code) in the UK Data Repository. Due to this "virtual" link, subsequent traceability messages can be reported based on the printed code.

6.7.2.2 Description of fields

request for the code pairing of UIs – request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = PAR
Event_Time	Intended time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
upUI	List of Paired codes		S	M	
upID	Code pairing detail		M	M	
Printed_Code	Unit packet level UI marked on packs	upUI(L)	S	M	
Paired_Code	Paired UI	upUI(M)	S	M	

Figure 10 - Message 3A.1 Pairing between marked codes and virtual paired codes

The UK System will link the upUI Printed_Code with the upUI Paired_Code applying a 1:1 relationship and then store this information in the UK Repository.

The upUI Printed_Codes must have not been paired beforehand. The UK UI Codes must exist, not be expired, and not have been paired or activated beforehand.

6.7.2.2.1 Response

response for the code pairing of UIs – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = PAR

6.7.2.3 Business validation

	IDA (2.3)
Business rule validation	
UI creation	
VAL_UI_EXIST_APP	upUI Paired_Code
VAL_UI_DUPLICATE_APP	upUI Paired_Code, upUI Printed_Code
VAL_UI_EXPIRY	upUI Paired_Code
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_ACTIVE_EOID	EO_ID
Pairing Validation	
VAL_PRINTED_CODES_ALREADY_USED	upUI Printed_Code
VAL_PAIRED_CODES_ALREADY_USED	upUI Paired_Code
Sequence Validation	
VAL_UI_ORD_SEQUENCE	upUI Paired_Code, upUI Printed_Code

The UK Gateway will apply a sequence validation in which the paired UIs (i.e., printed code + UK code) must be reported first in a message of type 3.A1 (PAR) and then in a 3.1 message (EUA). This is to enforce the applicable regulatory rules requiring that the codes are paired before the application of the foreign code is reported to the system.

When the EUA message is reported on the Printed code after the pairing, the UI will be able to be queried by:

- The upUI(s) version of the UK Code
- The upuI(L) version of the Printed Code
- The Human Readable Version of the Printed Code (as reported in the EUA message)

Note that after a pairing has been confirmed the UK code can only be activated by submitting an activation message using the printed code.

6.7.2.4 JSON Request Sample

```
{
"Message_Type": "PAR",
 "EO_ID": "QCUKT+1AB020054",
 "Event_Time": "22032014", "Message_Time_Long":"2022-03-20T14:16:45Z", "upUI": {

"upID": [ {

"Printed_Code": "PRINTEDXXXXXXXXXXXX1PR012345678919030110",

"Paired_Code": "QCUKTXXXXXXXXXXX10FA000001" },

{
 "Printed_Code": "PRINTEDXXXXXXXXXXXX1PR012345678919030110", "Paired_Code":
"QCUKTXXXXXXXXXXX10FA000001"

} ]

}

}
```

6.7.2.5 JSON Successful response Sample

```
HTTP Status 202

{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "PAR",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

6.7.2.6 Error response sample

Processing errors

HTTP status	Error Code	Error Description
<< Common response code >>		
400	UIS_APPLICATION_ERROR	VAL_UI_EXIST_APP
400	UI_NOT_EXIST	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_EXPIRED	VAL_UI_EXPIRY
400	PRINTED_CODES_ALREADY_USED	VAL_PRINTED_CODES_ALREADY_USED

400	PAIRED_CODES_ALREADY_USED	VAL_PAIRED_CODES_ALREADY_USED
400	UI_SEQUENCE_ERROR	VAL_UI_ORD_SEQUENCE

6.8 Reporting operational events (product movement information)

6.8.1 EUA – (3.1) Application of unit level UIs on unit packets

6.8.1.1 Description

Event notification when the code is applied / printed on unit packets.

6.8.1.2 Description of the fields

upUI application event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EUA
Event_Time	Intended time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
F_ID	Facility identifier code	FID	S	M	
upUI_1	List of unit packet level UIs to be recorded (full length)	upUI(L)	M	M	
upUI_2	List of corresponding unit packet level UIs to be recorded (as visible in human readable format) indicated in the same order as upUI_1	upUI(s)	M	M	
upUI_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.1.2.1 Response

response for the code pairing of UIs – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic	S	M	Message_Type = EUA

response for the code pairing of UIs – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
		Information Response >>			

6.8.1.3 Business validation

	EUA (3.1)
Technical validation	
VAL_UI_MULT_MSG	upUI_1, upUI_2
Business rule validation	
UI creation	
VAL_UI_EXIST_APP	upUI_1
VAL_UI_DUPLICATE_APP	upUI_1
VAL_UI_EXPIRY	upUI_1
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID
VAL_ENT_ACTIVE_EOID	EO_ID
VAL_ENT_ACTIVE_FID	F_ID
Sequence Validation	
VAL_UI_FID_APP	F_ID with information for upUI_1
VAL_UI_ORD_REACTIVATION	upUI_1
Message Timing	
VAL_EVT_24H	Event_Time

6.8.1.4 Sequence validation

The following table represents the authorized transitions for a UI and specifically the previous message for the UI.

Message Received	upUI(s) Codes Generated	PAR (3.A1)
EUA 3.1	Yes	Yes
EUA 3.1 Import	Yes	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

The EUA (3.1) application event must be preceded by the generation of the codes and/or a Code Pairing message.

6.8.1.5 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "F_ID": "QCUKT<1AB020054000049",
  "Event_Time": "19032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "upUI_1": [
    "DANXXXXXXXXXXXX1PR012345678919030110",
    "DANXXXXXXXXXXXX2PR012345678919030110"
  ],
  "upUI_2": [
    "DANXXXXXXXXXXXX1PR012345678919030110",
    "DANXXXXXXXXXXXX2PR012345678919030110"
  ],
  "upUI_comment": "Comments",
  "Message_Type": "EUA"
}
```

6.8.1.6 JSON Successful response sample

```
HTTP Status 202

{
```

```
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",

    "Message_Type": "EUA",

    "Error": false,

    "Errors": null,

    "Checksum": "G6HF5H"

}
```

6.8.1.7 Error response sample

HTTP status	Error Code	Error Description
<< Common response code >>		
400	MULTIPLE_UI	VAL_UI_MULT_MSG
400	UIS_APPLICATION_ERROR	VAL_UI_EXIST_APP
400	UIS_APPLICATION_ERROR	VAL_UI_DUPLICATE_APP
400	UI_NOT_EXIST	VAL_UI_EXPIRY
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	VAL_ENT_EXIST_FID	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	VAL_ENT_EXIST_FID	VAL_ENT_ACTIVE_FID
400	FID_MISMATCH	VAL_UI_FID_APP
400	UI_DEACTIVATED	VAL_UI_ORD_REACTIVATION
299	OPERATION_WITHIN_24_HOURS	VAL_EVT_24H

6.8.2 EPA – (3.2) Application of aggregated level UIs on aggregated packaging

6.8.2.1 Description

Event notification when the code is applied / printed on an aggregation container. This also records the items that are aggregated into this container.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

75

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

6.8.2.2 Description of the fields

Application of aggregated level UIs on aggregated packaging - request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EPA
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
F_ID	Facility identifier code	FID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
aUI	Aggregated level UI	aUI	S	M	
Aggregation_Type	Identification of aggregation type	Integer	S	M	1 – aggregation of only unit packet level UIs 2 – aggregation of only aggregated level UIs 3 – aggregation of both unit packet and aggregated level UIs
Aggregated_UIs1	List of unit packet level UIs subject to aggregation	upUI(L)	M	M, if Aggregation_Type = 1 or 3	
Aggregated_UIs2	List of aggregated level UIs subject to further aggregation	aUI	M	M, if Aggregation_Type = 2 or 3	
aUI_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.2.2.1 Response

Application of aggregated level UIs on aggregated packaging – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EPA
Basic Information Block	Additional optional acknowledgment Information	Component << Basic	S	O	

Application of aggregated level UIs on aggregated packaging – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
		Information Block >>			

6.8.2.3 Business validation

	EPA (3.2)
Technical validation	
VAL_UI_MULT_MSG	aUI, Aggregated_UIs1
VAL_FIE_REF	aUI, Aggregated_UIs1. Error Descr for Circular Reference issue: The message contains UI values that form a circular reference
Business rule validation	
UI creation	
VAL_UI_EXIST_UPUI	Aggregated_UIs1
VAL_UI_EXIST_AUI	aUI, Aggregated_UIs2
VAL_UI_EXIST_UPUI_SEQ	Aggregated_UIs1
VAL_UI_EXIST_AUI_SEQ	Aggregated_UIs2
VAL_UI_EXPIRY	aUI, Aggregated_UIs1
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID
VAL_ENT_ACTIVE_EOID	EO_ID
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	aUI, Aggregated_UIs2
VAL_UI_ORD_AGG_MULT	aUI, Aggregated_UIs2
VAL_UI_ORD_IMPLDISAGG	aUI, Aggregated_UIs2
VAL_UI_ORD_AGG_FID	F_ID for Aggregated_UIs1 and Aggregated_UIs2

Message Timing	
VAL_EVT_24H	Event_Time

6.8.2.4 Sequence Validation

The following table represents the authorized transitions for a UI and specifically the previous message for the UI.

Message Received	IRA 2.2	EUA 3.1	EUA 3.1 Import	EPA 3.2 parent UI	EPA 3.2 parent UI Import	EPA 3.2 Child	ERP 3.4	ERP 3.4 (Return)	EUD 3.6
EPA 3.2 parent UI	Yes	No	No	No	No	No	No	No	Yes
EPA 3.2 child UI (upUI)	No	Yes	Yes	No	No	Yes	Yes	Yes	No
EPA 3.2 child UI (aUI)	No	No	No	Yes	Yes	Yes	Yes	Yes	No

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

A parent aUI can be

- Self generated parent UI. This event will correspond to the initial commissioning of the aUI.
- ID Issuer generated aUI reported using an IRA (2.2) event.
- As a aUI that is re-used. This aUI must be preceded by a EUD (3.6) (explicit) disaggregation event. Note that in the case of a reuse, the location validation is not performed. In other words, the aUI can be re aggregated in any location.

A Child UI can be reported if it is present in the location of the aggregation.

- A upUI can be applied, following a EUA (3.1) event.
- A upUI can be present in the location following a disaggregation event of a previous aggregation.

- An aUI can be present in the location following a disaggregation event of a previous aggregation where it was reported as child UI.
- A upUI or aUI that are still part of a valid aggregation as child UIs. The reporting of the aggregation event will trigger an implicit disaggregation of the initial aggregation.
- A upUI or aUI that are arrived ERP (3.4) event.

Location validation on the child UI is performed for the aggregation as the UI should be in the location of the aggregation.

6.8.2.5 Implicit disaggregation trigger

This event can trigger an implicit disaggregation when a child UI is identified as part of the event.

Example: Implicit Disaggregation occurring when re-aggregating

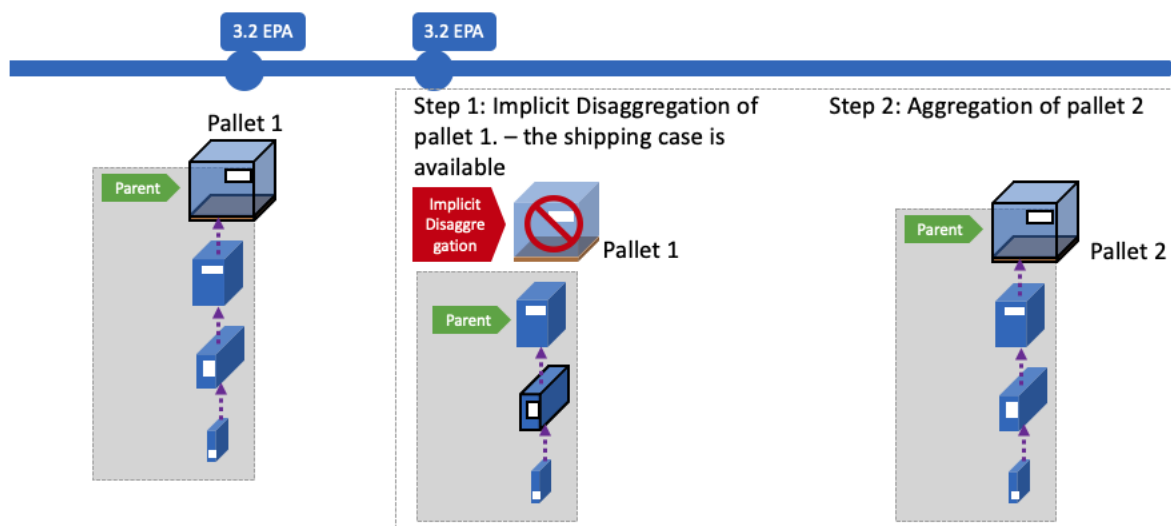*Figure 11 Implicit disaggregation triggered by an EPA (3.2) event*

The second aggregation event (EPA 3.2) result in the creation of the new pallet with the same content as pallet 1 that is implicitly disaggregated.

6.8.2.6 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "F_ID": "QCUKT<1AB020054000049",
  "Event_Time": "19032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Aggregation_Type": "1",
```

```
    "aUI" : "DANXXXXXXXXXXXXX1FA00000119030110",

    "Aggregated_UIs1": ["DANXXXXXXXXXXXXX1PR012345678919030110",
"DANXXXXXXXXXXXXX2PR012345678919030110", "DANXXXXXXXXXXXXX3PR012345678919030110",
"DANXXXXXXXXXXXXX10FA00000119030110"],

    "Aggregated_UIs2": ["DANXXXXXXXXXXXXX10FA00000119030110"],

    "aUI_comment": "Comments",

    "Message_Type": "EPA"
}
```

6.8.2.7 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "EPA",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

6.8.2.8 Error response sample

HTTP status	Error Code	Error Description
<< Common response code >>		
400	UIS_APPLICATION_ERROR	VAL_UI_EXIST_UPUI
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	UI_EXPIRED	VAL_UI_EXPIRY
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_UI_ORD_DEACTIVATED

400	MULTIPLE_AGGREGATION	VAL_UI_ORD_AGG_MULT
400	UI_ALREADY_DISAGGREGATED	VAL_UI_ORD_IMPLDISAGG
400	LOCATION_MISMATCH	VAL_UI_ORD_AGG_FID
299	OPERATION_WITHIN_24_HOURS	VAL_EVT_24H
400	FAILED_VALIDATION	VAL_FIE_REF

6.8.3 EDP – (3.3) Dispatch of tobacco products from a facility

6.8.3.1 Description

Record that the UIs listed in the call have been dispatched by the economic operator.

6.8.3.2 Description of the fields

Dispatch of tobacco products from a facility event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EDP
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Event_Time	Time of event occurrence	Time (s)	S	M	
Message_Time_long	Message sending Time	Time (L)	S	M	
F_ID	Dispatch facility identifier code	FID	S	M	
Destination_ID1	Indication if the destination facility is located on the UK territory and if it is a vending machine (VM)	Integer	S	M	1 – Non UK dest. 2 – UK destination other than VM – fixed quantity delivery 3 – UK VM(s) 4 – UK destination other than VM – delivery with VV

Dispatch of tobacco products from a facility event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Destination_ID2	Destination facility identifier code	FID	S	M, if Destination_ID1 = 2	
Destination_ID3	Destination facility identifier code(s) – possible multiple vending machines	FID	M (limited to 1000 FID)	M, if Destination_ID1 = 3	
Destination_ID4	Destination id facility codes	FID	M (limited to 1000 FID)	M, if Destination_ID1 = 4	
Destination_ID5	Destination facility's full address	Text(5000)	S	M, if Destination_ID 1 = 1	
Destination_ID5_Address_Name	Destination facility's full address - Name part of the Address	Text(5000)	S	O	
Destination_ID5_Address_StreetOne	Destination facility's full address - Street part of the Address	Text(5000)	S	M, if Destination_ID 1 = 1	
Destination_ID5_Address_StreetTwo	Destination facility's full address - Second Element of the Street part of the Address	Text(5000)	S	O	
Destination_ID5_Address_City	Destination facility's full address - City	Text(5000)	S	M, if Destination_ID 1 = 1	
Destination_ID5_Address_PostCode	Destination facility's full address - PostalCode information	Text(5000)	S	O	
Transport_mode	Mode of transport by which the product leaves the facility, see: Commission Regulation (EC) No 684/2009, Annex II, Code List 7	Integer	S	M	See TransportMode in section **Error! Reference source not found.**
Transport_vehicle	Identification of the mode of transport (i.e. number plates, train number, plane/flight number, ship name or other identification)	Text(5000)	S	M	'n/a' is permitted value if Transport_mode = 0 and product movement takes place between adjacent facilities

Dispatch of tobacco products from a facility event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
					and is delivered manually
Transport_cont1	Indication if the transport is containerised and uses an individual transport unit code (e.g. SSCC)	Boolean	S	M	0 – No 1 – Yes
Transport_cont2	Individual transport unit code of the container	ITU	S	M, if Transport_cont1 = 1	
Transport_s1	Indication if the dispatch takes place with the logistic/postal operator who operates its own track and trace system. Only for small quantities of tobacco products (net weight of the products dispatched below 10 kg) destined for exports to other countries	Boolean	S	M	0 – No 1 – Yes
Transport_s2	The logistic operator's tracking number	Text(5000)	S	M, if Transport_s1 = 1	
EMCS	Dispatch under the Excise Movement and Control System (EMCS)	Boolean	S	M	0 – No 1 – Yes
EMCS_ARC	Administrative Reference Code (ARC)	ARC	S	M, if EMCS = 1	
SAAD	Dispatch with a simplified accompanying document, see: Commission Regulation (EEC) No 3649/92	Boolean	S	M	0 – No 1 – Yes
SAAD_number	Reference number of the declaration and/or authorisation which has to be given by the competent authority in the country of destination before the movement starts	T Text(5000)	S	M, if SAAD = 1	
Exp_Declaration	Indication if the Movement Reference Number (MRN) has been issued by the customs office	Boolean	S	M	0 – No 1 – Yes

Dispatch of tobacco products from a facility event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Exp_ DeclarationNumber	Movement Reference Number (MRN)	MRN	S	M, if Exp_Decl aration = 1	
UI_Type	Identification of UI types in the dispatch (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs subject to the dispatch	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs subject to the dispatch	aUI	M	M, if UI_Type = 2 or 3	
Dispatch_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.3.2.1 Response

Response:

Dispatch event – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EDP

6.8.3.3 Business validation

	EDP (3.3)
Technical validation	
VAL_MSG_JSON	EXCISE_NUMBER_NOT_VALID
VAL_UI_MULT_MSG	upUIs, aUIs
Business rule validation	

UI creation	
VAL_UI_EXIST_UPUI	upUIs
VAL_UI_EXIST_AUI	aUIs
VAL_UI_EXIST_UPUI_SEQ	upUIs
VAL_UI_EXIST_AUI_SEQ	aUIs
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID, Destination_ID2, Destination_ID3, Destination_ID4
VAL_ENT_ACTIVE_EOID	EO_ID
VAL_ENT_ACTIVE_FID	F_ID, Destination_ID2, Destination_ID3, Destination_ID4
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	upUIs, aUIs
VAL_UI_ORD_DISAGG	aUIs
VAL_UI_ORD_IMPLDISAGG	aUIs
VAL_UI_ORD_DISPATCH	upUIs, aUIs
Message Timing	
VAL_EVT_TIME	Event_Time

6.8.3.4 Sequence validation

The following table represents the authorized transitions for a UI and specifically the previous message for the UI.

The type of the Dispatch event 3.3 EDP refers to the Destination_ID1 field.

- Type 1 – Non UK dest.
- Type 2 – UK destination other than VM – fixed quantity delivery
- Type 3 – UK VM(s)
- Type 4 – UK destination other than VM – delivery with VV

	EUA 3.1	EUA 3.1 Import	EPA 3.2 parent UI	EPA 3.2 parent UI Import	EPA 3.2 Child	ERP 3.4	ERP 3.4 (Return)

Message Received							
EDP 3.3 Export (type 1)	Yes	No	Yes	No	Yes	Yes	Yes
EDP 3.3 (type 2)	Yes	No	Yes	No	Yes	Yes	Yes
EDP 3.3 VM (type 3)	Yes	No	Yes	No	Yes	Yes	Yes
EDP 3.3 VV (type 4)	Yes	No	Yes	No	Yes	Yes	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

Products can be dispatched from a location only if they were applied (EUA 3.1) or aggregated (EPA 3.2) at that specific location, or if they were previously reported as having been arrived at that location. This means that Dispatch events should follow an Arrival, an Aggregation or an Application message, and the origin of the Dispatch must correspond to the location of previous Arrival, Aggregation or Application event.

6.8.3.4.1 Expected sequences

➢ ERP – 3.4 message > EDP – 3.3 message
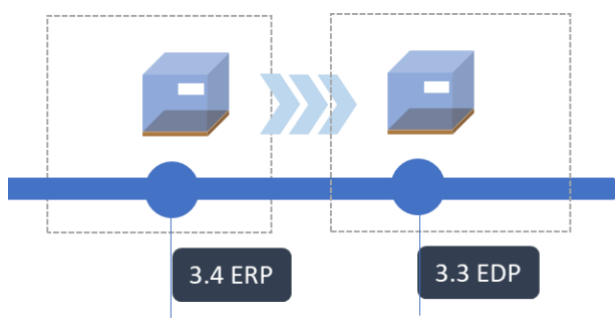
➢ EPA – 3.2 message > EDP – 3.3 message
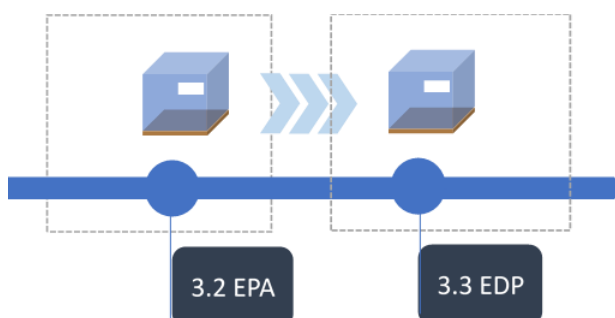
6.8.3.4.2 Example of sequence errors for EDP (3.3):

- UI_SEQUENCE_ERROR:

The error is generated when UIs scanned at Dispatch are not part of a prior Delivery/Arrival, Aggregation or Application message.

The UIs are considered as "in transit" and cannot be dispatched again.

The prior Delivery/Arrival of the UIs in the location might not have been reported or the UIs might have already been dispatched from the location.

Example of EDP (3.3) with prior ERP (3.4) not reported:
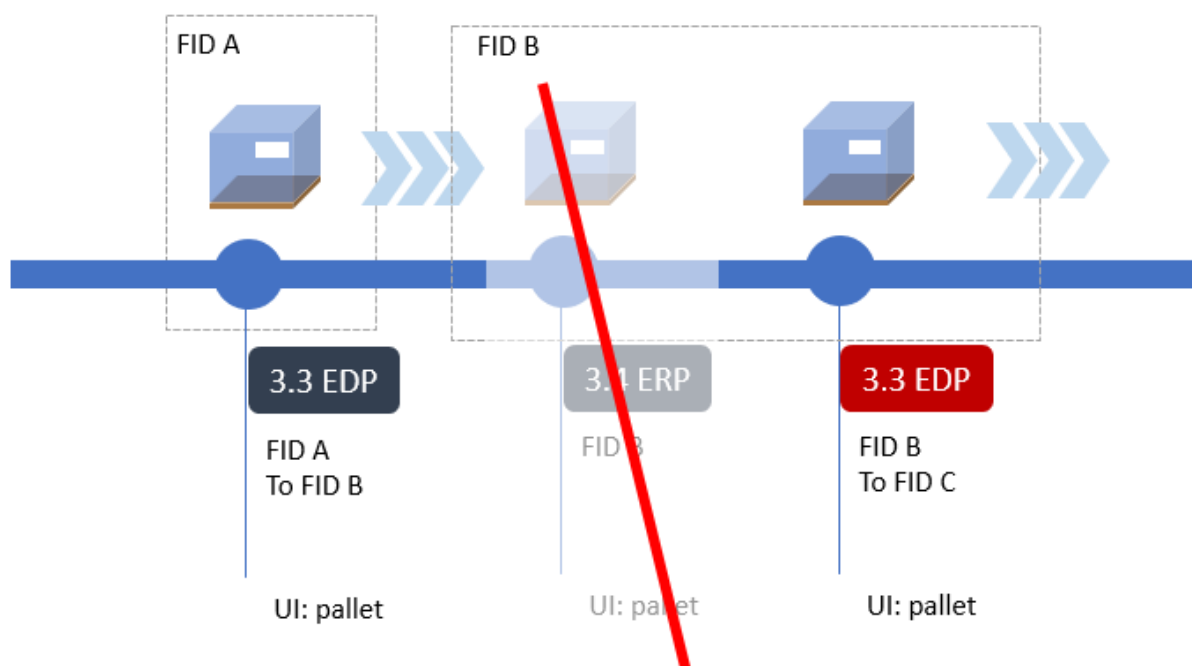
- LOCATION_MISMATCH

The error is generated when the facility of origin of the Dispatch does not match the current location of the UIs reported in the Dispatch message.

There might have been a misreporting in the previous Delivery/Arrival messages leading to a wrong location of the UIs or the current Dispatch does not report the correct facility of origin.

Example of EDP (3.3) with wrong FID:
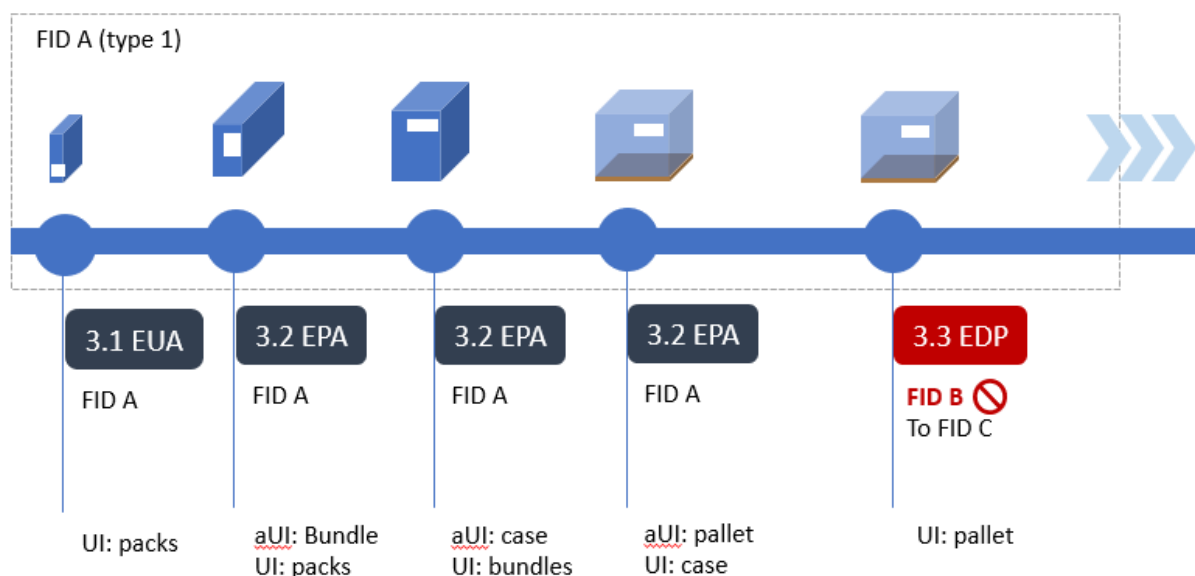
6.8.3.4.3 Import scenario

The imported goods that have been applied and aggregated require to be part of an Arrival (ERP 3.4) event.

6.8.3.5 Implicit disaggregation trigger

This event can trigger an implicit disaggregation when a child UI is identified as part of the event.

Example of an Implicit Disaggregation occurring when Dispatching
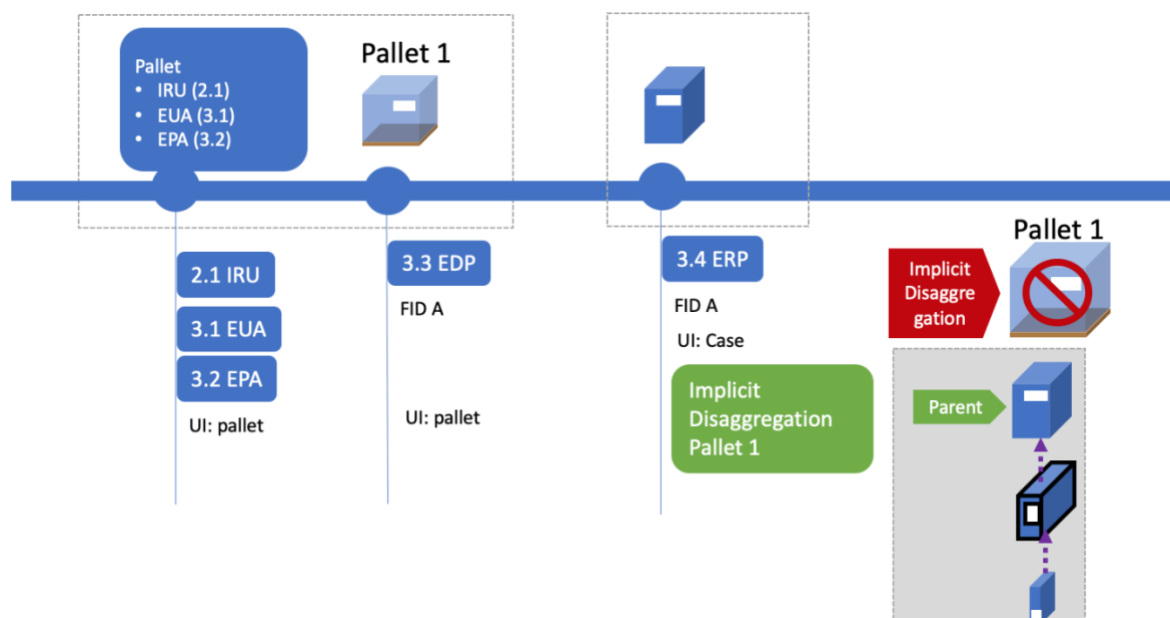
6.8.3.6 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "F_ID": "QCUKT<1AB020054000049",
  "Event_Time": "19032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Destination_ID1": "1",
  "Destination_ID2": "FacilityIdB",
  "Destination_ID3": [ " FacilityIdB ", " FacilityIdB " ],
  "Destination_ID4": [ " FacilityIdB" ],
  "Destination_ID5": "FacilityIdA",
  "Transport_vehicle": "1",
  "Transport_cont1": 1,
  "Transport_cont2": "1",
  "Transport_s1": 1,
  "Transport_s2": "1",
  "EMCS": false,
  "EMCS_ARC": null,
  "SAAD": 1,
  "SAAD_number": 1,
  "Exp_Declaration": 1,
  "Exp_DeclarationNumber": 1,
  "UI_Type": 3,
  "upUIs": [ "DANXXXXXXXXXXXXX1PR012345678919030110",
"DANXXXXXXXXXXXXX2PR012345678919030110" ],
  "aUIs": [ "DANXXXXXXXXXXXXX1FA000000119030110" ],
  "Dispatch_comment": "Comments",
  "Message_Type": "EDP"
}
```

6.8.3.7 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "EDP",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

6.8.3.8 Error response sample

Processing errors

HTTP status		
<< Common response code >>		
400	EXCISE_NUMBER_NOT_VALID	VAL_FIE_FORMAT
400	MULTIPLE_UI	VAL_UI_MULT_MSG
400	UI_NOT_EXIST UI_NOT_VALID	VAL_UI_EXIST_UPUI
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_FID (Gateway only)
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	UI_ALREADY_DISAGGREGATED	VAL_UI_ORD_DISAGG or VAL_UI_ORD_IMPLDISAGG
400	LOCATION_MISMATCH	VAL_UI_ORD_DISPATCH
299	SHIPMENT_WITHIN_24_HOURS	VAL_EVT_TIME
400	UI_SEQUENCE_ERROR	VAL_UI_ORD_SEQUENCE

6.8.4 ERP – (3.4) Arrival of tobacco products at a facility

6.8.4.1 Description

Record that the UIs listed in the message have been received by an economic operator.

6.8.4.2 Description of the fields

		Arrival of tobacco products at a facility			
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = ERP
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
F_ID	Arrival facility identifier code	FID	S	M	
Event_Time	Time of event occurrence	Times(s)	S	M	
Message_Time_long	Message sending Time	Times(L)	S	M	
Product_Return	Indication if the arriving products are a return following complete or partial non-delivery	Boolean	S	M	0 – No 1 – Yes
UI_Type	Identification of UI types received (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs received	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs received	aUI	M	M, if UI_Type = 2 or 3	
Arrival_comment	Comments by the reporting entity	Text	S	O	

6.8.4.2.1 Response

		Arrival of tobacco products at a facility– response			
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = ERP

6.8.4.3 Business Validation

	ERP (3.4)
Technical validation	
VAL_UI_MULT_MSG	upUIs , aUIs
Business rule validation	
UI creation	
VAL_UI_EXIST_UPUI	upUIs
VAL_UI_EXIST_AUI	aUIs
VAL_UI_EXIST_UPUI_SEQ	upUIs
VAL_UI_EXIST_AUI_SEQ	aUIs
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID
VAL_ENT_ACTIVE_EOID	EO_ID
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	upUIs, aUIs
VAL_UI_ORD_DISAGG	aUIs
VAL_UI_ORD_IMPLDISAGG	aUIs
VAL_UI_ORD_ARRIVAL	upUIs, aUIs
VAL_UI_ORD_ARRIVAL_RETURN	upUIs, aUIs

6.8.4.4 Sequence validation

The following table represents the authorized transitions for a UI and specifically the previous message for the UI.

The information contained in these documents is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of dentsu International Switzerland AG.

92

dentsu Tracking | Route des Jeunes 35 | 1227 Geneva | Switzerland
dentsu International Switzerland AG |Giessereistrasse, 18 |8005 Zürich | Switzerland | UI: CHE- 110.279.525

Message Received	EUA 3.1 Import	EPA 3.2 parent UI Import	EDP 3.3 (type 1) Export	EDP 3.3 (type 2)	EDP 3.3 (types 3) VM	EDP 3.3 (type 4) VV	ETL 3.5	ETL 3.5 Export	EVR 3.7
ERP 3.4	Yes	Yes	No	Yes	No	No	Yes	No	No
ERP 3.4 (Return)	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

The type 'return' of the ERP (3.4) is based on the Product_Return field

- 0 – No;
- 1– The arrival is a type return.

Events must be transmitted in sequence: Arrivals are expected to be reported in proper events sequence, following a dispatch, a transloading, or as a (partial) return from Vending Van delivery or from retail outlets.

6.8.4.4.1 Import scenario

In the case of Imported goods, the newly applied (EUA 3.1) or aggregated (EPA 3.2) UIs must be part of an arrival event.

6.8.4.4.2 Arrival after Dispatch

➢ EDP – 3.3 (type 2) message > ERP – 3.4 message
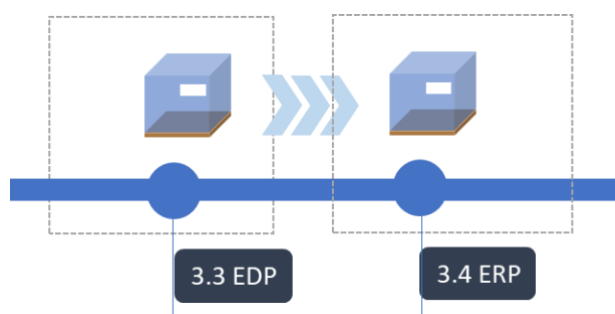
6.8.4.4.3 Arrival after Transloading

> ETL – 3.5 message > ERP – 3.4 message
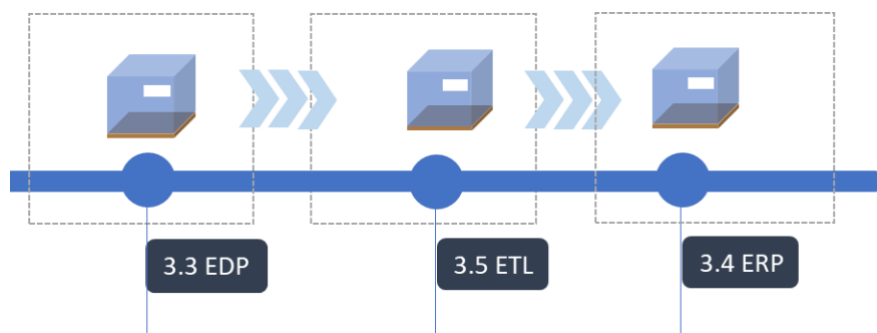
6.8.4.4.4 Arrival after dispatch carried out by vending van

> EVR – 3.7 messages > ERP of type return – 3.4 message
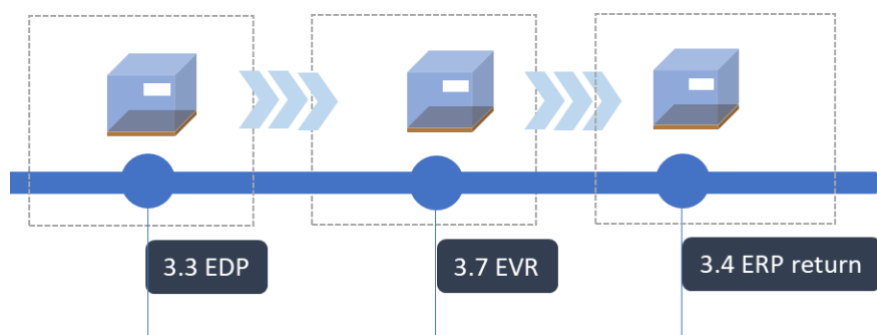
Events must be reported respecting Principle 4: All Disaggregations must be performed at a location. Disaggregation during transport is not permitted.

Events must be reported respecting Principle 5: Reporting on the Arrival should be done on the same UI that was reported during the Dispatch/Transloading process. This is a consequence of Principle 4. This means that an Arrival Event that contains child UIs of an UI reported during the Dispatch/Transloading Event will be rejected. The same UI must be reported.

Exception to principle 5: Arrival of type return can be reported at a different level than the previous dispatch/transloading/delivery with VV.

6.8.4.4.5 Arrival of type return

The Arrival of type return is the proper reporting event for exported goods, goods in transit (dispatched or in transloading).

6.8.4.4.6 Examples of sequence errors for ERP (3.4)

ARRIVAL_NOTALLOWED

The error is generated because UIs in the Arrival message were not part of a previous EDP (3.3) or ETL (3.5) message. The previous Economic Operator should be contacted to verify his Dispatch/Transloading message.

Note: in case of product return, the ERP (3.4) can be reported after an EVR (3.7), this will not trigger ARRIVAL_NOTALLOWED.

Example of ERP (3.4) rejected because the previous EDP (3.3) has not been reported:
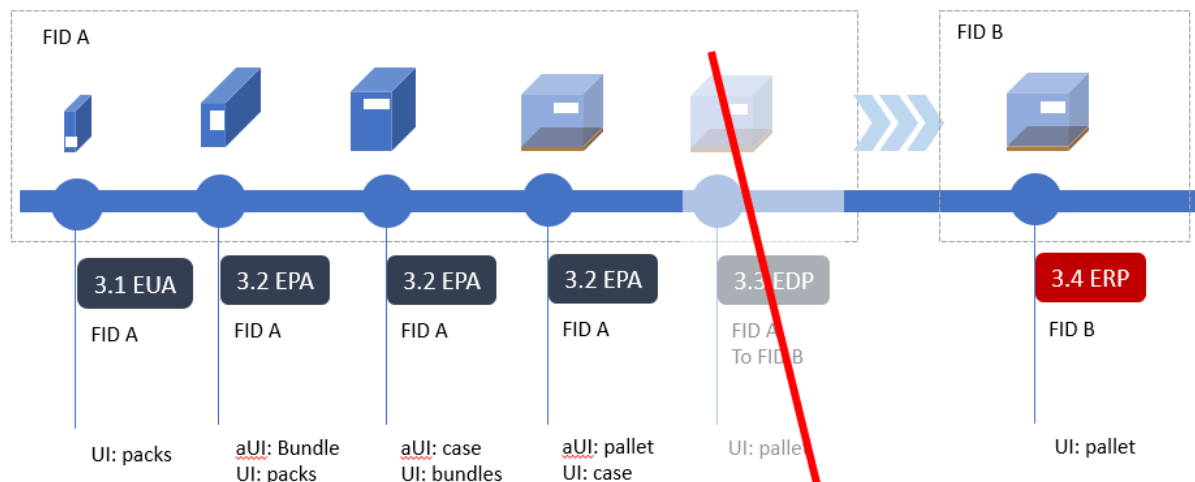
Figure 12 Arrival sequence error ARRIVAL_NOTALLOWED

UI_SEQUENCE_ERROR

The error is generated when UIs scanned at Arrival are not identical to those scanned in the prior Dispatch/Transloading event.

According to principle 5, the UIs must be scanned at Arrival at the highest level of aggregation, before being implicitly or explicitly disaggregated.

If the error is generated when Arrival is reported at the highest available aggregation level, the previous Economic Operator should be contacted to verify at which level he reported the Dispatch/Transloading event.

Exception: Arrivals of type return will not generate this error because implicit disaggregation is in that case permitted.

Example of ERP (3.4) rejected because it is not reported at the right level of aggregation:
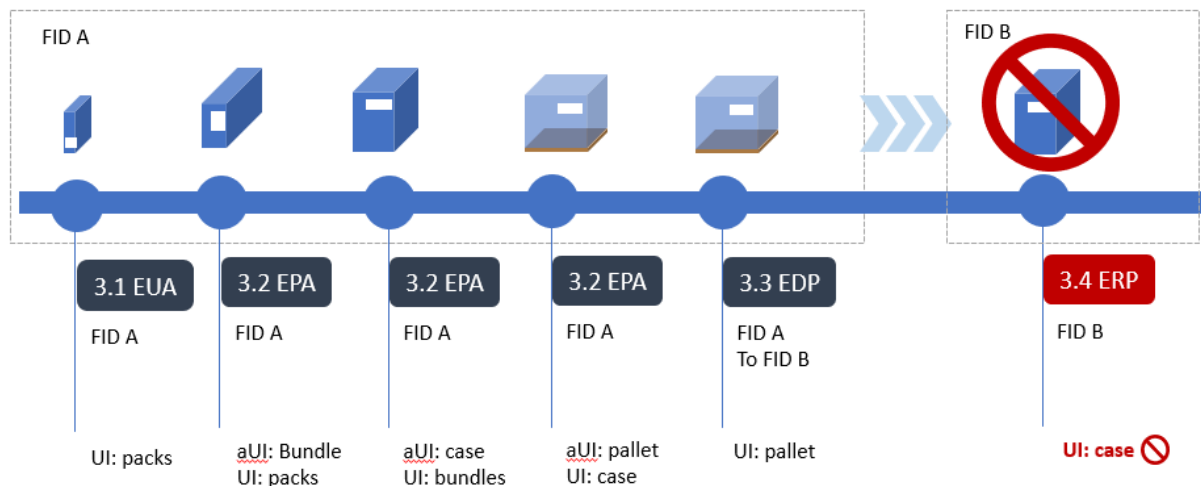
Figure 13 Arrival sequence error UI_SEQUENCE_ERROR

6.8.4.5 Implicit Disaggregation Trigger

The Arrival of type Return (Product_Return = "true") ERP (3.4) can trigger an implicit disaggregation when a child UI is identified as part of the event.

Note that an arrival message that contains a child UI will cause a sequence validation error UI_SEQUENCE_ERROR if the Product_Return flag is set to false.

6.8.4.6 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "F_ID": "QCUKT<1AB020054000049",
  "Event_Time": "19032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Product_Return": "true",
  "UI_Type": 1,
  "upUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110", "
DANXXXXXXXXXXXX2PR012345678919030110" ],
  "aUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110" ],
  "Arrival_comment": "Comments",
  "Message_Type": "ERP"
}
```

6.8.4.7 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "ERP",
    "Error": false,
```

```
    "Errors": null,

    "Checksum": "G6HF5H"

}
```

6.8.4.8 Error response sample

HTTP status	Error Code	Error Description
<< Common response code >>		
400	MULTIPLE_UI	VAL_UI_MULT_MSG
400	UI_NOT_EXIST UI_NOT_VALID	VAL_UI_EXIST_UPUI
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	UI_ALREADY_DISAGGREGATED	VAL_UI_ORD_DISAGG or VAL_UI_ORD_IMPLDISAGG
400	ARRIVAL_NOTALLOWED	VAL_UI_ORD_ARRIVAL or VAL_UI_ORD_ARRIVAL_RETURN
400	UI_SEQUENCE_ERROR	VAL_UI_ORD_SEQUENCE

6.8.5 ETL – (3.5) Trans-loading

6.8.5.1 Description

Report that UIs have been moved from one transport mechanism (vehicle) to another without being stored at a facility in-between.

6.8.5.2 Description of the fields

Trans-loading event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = ETL
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Event_Time	Intended time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
Destination_ID1	Indication if the destination facility is located in the UK	Integer	S	M	0 – No 1 – Yes
Destination_ID2	Destination facility identifier code	FID	S	M, if Destination_ID 1 = 1	
Destination_ID3	Destination facility's full address	Text(5000)	S	M, if Destination_ID 1 = 0	
Destination_ID3_Address_Name	Destination facility's full address - Name part of the Address	Text(5000)	S	O	
Destination_ID3_Address_StreetOne	Destination facility's full address - Street part of the Address	Text(5000)	S	M, if Destination_ID 1 = 0	
Destination_ID3_Address_StreetTwo	Destination facility's full address - Second Element of the Street part of the Address	Text(5000)	S	O	
Destination_ID3_Address_City	Destination facility's full address - City	Text(5000)	S	M, if Destination_ID 1 = 0	
Destination_ID3_Address_PostCode	Destination facility's full address - PostalCode information	Text(5000)	S	O	
Transport_mode	Mode of transport to which the product is trans-loaded, see: Commission Regulation	Integer	S	M	See TransportMode

Trans-loading event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
	(EC) No 684/2009, Annex II, Code List 7				
Transport_vehicle	Identification of the vehicle (i.e. number plates, train number, plane/flight number, ship name or other identification)	Text(5000)	S	M	
Transport_cont1	Indication if the transport is containerised and uses an individual transport unit code (e.g. SSCC)	Boolean	S	M	0 – No 1 – Yes
Transport_cont2	Individual transport unit code of the container	ITU	S	M, if Transport_cont1 = 1	
EMCS	Dispatch under the Excise Movement and Control System (EMCS)	Boolean	S	M	0 – No 1 – Yes
EMCS_ARC	Administrative Reference Code (ARC)	ARC	S	M, if EMCS = 1	
UI_Type	Identification of UI types subject to the trans-loading (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs subject to the trans-loading	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs subject to the trans-loading	aUI	M	M, if UI_Type = 2 or 3	
Transloading_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.5.2.1 Response

Trans-loading event – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = ETL

6.8.5.3 Business validation

	ETL (3.5)
Technical validation	
VAL_UI_MULT_MSG	upUIs, aUIs
Business rule validation	
UI creation	
VAL_UI_EXIST_UPUI	upUIs
VAL_UI_EXIST_AUI	aUIs
VAL_UI_EXIST_UPUI_SEQ	upUIs
VAL_UI_EXIST_AUI_SEQ	aUIs
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	Destination_ID2
VAL_ENT_ACTIVE_EOID	EO_ID
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	upUIs, aUIs
VAL_UI_ORD_DISAGG	aUIs
VAL_UI_ORD_IMPLDISAGG	aUIs
Message Timing	
VAL_EVT_TIME	Event_Time

6.8.5.4 Sequence validation

Message Received	EDP 3.3 (type 1) Export	EDP 3.3 (type 2)	EDP 3.3 (types 3) VM	EDP 3.3 (type 4) VV	ETL 3.5	ETL 3.5 Export
ETL 3.5	No	Yes	No	No	Yes	No
ETL 3.5 (Export)	Yes	No	No	No	No	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

ETL (3.5) can only be preceded by EDP (3.3) of type 1 or 2 or another ETL (3.5)

The ETL (3.5) event is not subject to any location validation

6.8.5.5 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "Event_Time": "19032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Destination_ID1": 1,
  "Destination_ID2": "FGHZ7G",
  "Destination_ID3": "",
  "Transport_mode": 1,
  "Transport_vehicle": 1,
  "Transport_cont1": 1,
  "Transport_cont2": "code",
  "EMCS": 1,
  "EMCS_ARC": "ref",
  "UI_Type": 1,
  "upUIs": [ "DANXXXXXXXXXXXXX1PR012345678919030110", "
DANXXXXXXXXXXXXX2PR012345678919030110" ],
  "aUIs": [ "DANXXXXXXXXXXXX10FA00000119030110" ],
  "Transloading_comment": "Comments",
```

```
    "Message_Type": "ETL"
}
```

6.8.5.6 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",
    "Message_Type": "ETL",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

6.8.5.7 Error response sample

HTTP status		
<< Common response code >>		
400	MULTIPLE_UI	VAL_UI_MULT_MSG
400	UI_NOT_EXIST UI_NOT_VALID	VAL_UI_EXIST_UPUI
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	UI_ALREADY_DISAGGREGATED	VAL_UI_ORD_DISAGG or VAL_UI_ORD_IMPLDISAGG
299	SHIPMENT_WITHIN_24_HOURS	VAL_EVT_TIME

6.8.6 EUD – (3.6) Disaggregation of aggregated level UIs

6.8.6.1 Description

Event showing that an aggregation no longer exists.

6.8.6.2 Description of the fields

aUI disaggregation event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EUD
EO_ID	Economic operator's identifier	EOID	S	M	
F_ID	Facility's identifier	FID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
aUI	Aggregated level UI subject to disaggregation	aUI	S	M	
disaUI_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.6.2.1 Response

aUI disaggregation event– response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EUD

6.8.6.3 Business validation

	EUD (3.6)
Business rule validation	
UI creation	
VAL_UI_EXIST_AUI	aUI
VAL_UI_EXIST_AUI_SEQ	aUI
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID
VAL_ENT_ACTIVE_EOID	EO_ID
Sequence Validation	
VAL_UI_ORD_DEACTIVATED	aUI
VAL_UI_ORD_AGG_FID	aUI (ONLY for aUI that have not been implicitly disaggregated)

6.8.6.4 Sequence validation

The following table represents the authorized transitions for a UI and specifically the previous message for the UI.

Message Received	EPA 3.2 parent UI	EPA 3.2 parent UI Import	EPA 3.2 Child	ERP 3.4	ERP 3.4 (Return)	EUD 3.6	EUD 3.6 (aUI implicitly disaggregated) - reuse of aUI
EUD 3.6	Yes	Yes	Yes	Yes	Yes	No	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed

Yes (with Location Validation)	Next Message allowed with location validation

6.8.6.4.1 Clarification of the location validation

The disaggregation event is subject to the location validation (VAL_UI_ORD_AGG_FID). In other words, the aUI that are disaggregated must be in the same facility.

There is an exception for aUIs that were already disaggregated implicitly.

6.8.6.4.2 Clarification on the impact of Implicit disaggregation

Reporting of a disaggregation event (EUD 3.6) for an aUI that was previously implicitly disaggregated (i.e., the reporting of an event that allows the triggering of the implicit disaggregation mechanism) and does not correspond to any physical movement. Therefore, the location validation control VAL_UI_ORD_AGG_FID will not be applied.
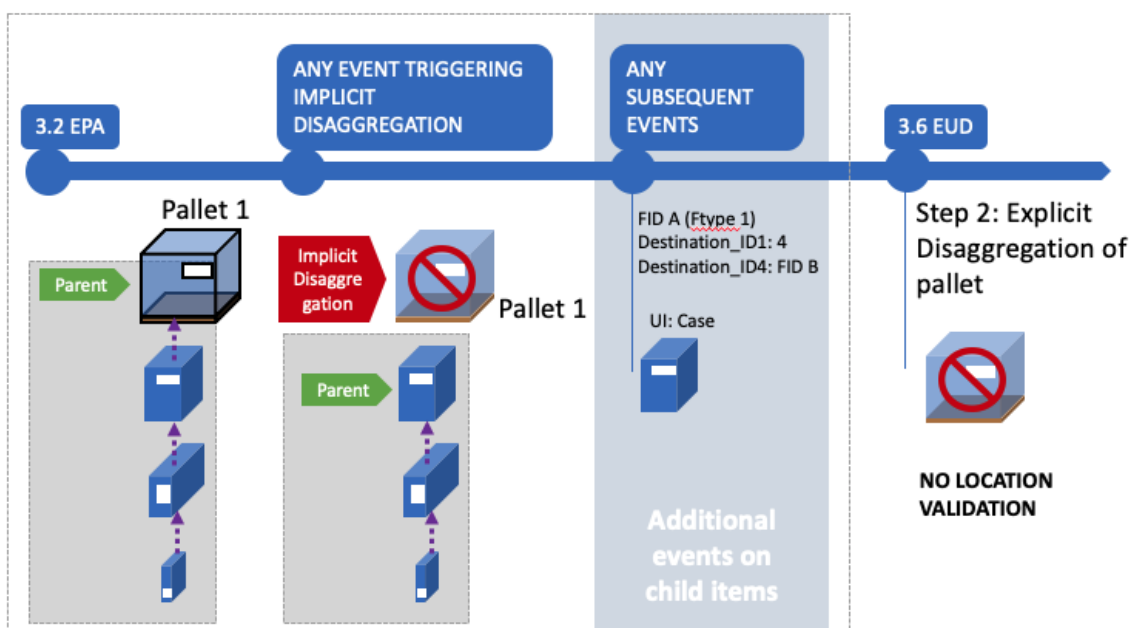
Figure 14 reporting of disaggregation event on aUI that are implicitly disaggregated.

6.8.6.5 Implicit disaggregation trigger

This event can trigger an implicit disaggregation when a child UI is identified as part of the event.

6.8.6.6 JSON Request sample

{

```
    "EO_ID": "QCUKT+1AB020054",

    "F_ID": "QCUKT<1AB020054000049",

    "Event_Time": "19032014",

    "Message_Time_Long":"2022-03-20T14:16:45Z",

    "aUI": "DANXXXXXXXXXXX10FA00000119030110",

    "disaUI_comment": "Comments",

    "Message_Type": "EUD"

}
```

6.8.6.7 JSON Successful response sample

```
HTTP Status 202

{

    "Code": "873345b2-882f-4064-91f0-90669b46c30a",

    "Message_Type": "EUD",

    "Error": false,

    "Errors": null,

    "Checksum": "G6HF5H"

}
```

6.8.6.8 Error response sample

HTTP status	Error Code	Error Description
<< Common response code >>		
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	LOCATION_MISMATCH	VAL_UI_ORD_AGG_FID

6.8.7 EVR – (3.7) Report the delivery carried out with a vending van to retail outlet

6.8.7.1 Description

Event sent when UIs have been distributed to a retail outlet via a van delivery.

6.8.7.2 Description of the fields

Vending Van event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EVR
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
F_ID	Facility identifier code of retail outlet	FID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
UI_Type	Identification of UI types delivered (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs delivered	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs delivered	aUI	M	M, if UI_Type	

				= 2 or 3	
Delivery_comment	Comments by the reporting entity	Text(5000)	S	O	

6.8.7.2.1 Response

Vending Van event – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EVR

6.8.7.3 Business validation

	EVR (3.7)
Technical validation	
VAL_UI_MULT_MSG	upUIs, aUIs
Business rule validation	
UI creation	
VAL_UI_EXIST_UPUI	upUIs
VAL_UI_EXIST_AUI	aUIs
VAL_UI_EXIST_UPUI_SEQ	upUIs
VAL_UI_EXIST_AUI_SEQ	aUIs
Entity Validation	
VAL_ENT_EXIST_EOID	EO_ID
VAL_ENT_EXIST_FID	F_ID
VAL_ENT_ACTIVE_EOID	EO_ID

Sequence Validation	
VAL_UI_ORD_DEACTIVATED	upUIs, aUIs
VAL_UI_ORD_DISAGG	aUIs
VAL_UI_ORD_IMPLDISAGG	aUIs
Message Timing	
VAL_EVT_24H	Event_Time

6.8.7.4 Sequence Validation

The following table represents the authorized transitions for an UI and specifically the previous message associated with that UI.

Message Received	EDP 3.3 (type 1) Export	EDP 3.3 (type 2)	EDP 3.3 (types 3) VM	EDP 3.3 (type 4) VV
EVR 3.7	No	No	No	Yes

No	Next message not allowed for the UI (including different aggregation level)
Yes	Next Message allowed
Yes (with Location Validation)	Next Message allowed with location validation

For each UI in the event, an EVR must be preceded by a Dispatch event (EDP 3.3) of type 4. Any other combination will result in a Sequence Error.

Note that omitting the reporting of the Dispatch event (EDP 3.3) will result in a SEQUENCE_ERROR.

6.8.7.5 Implicit disaggregation trigger

This event can trigger an implicit disaggregation when a child UI is identified as part of the event.

Implicit Disaggregation occurring when Delivering to Retail Outlet (EVR 3.7)
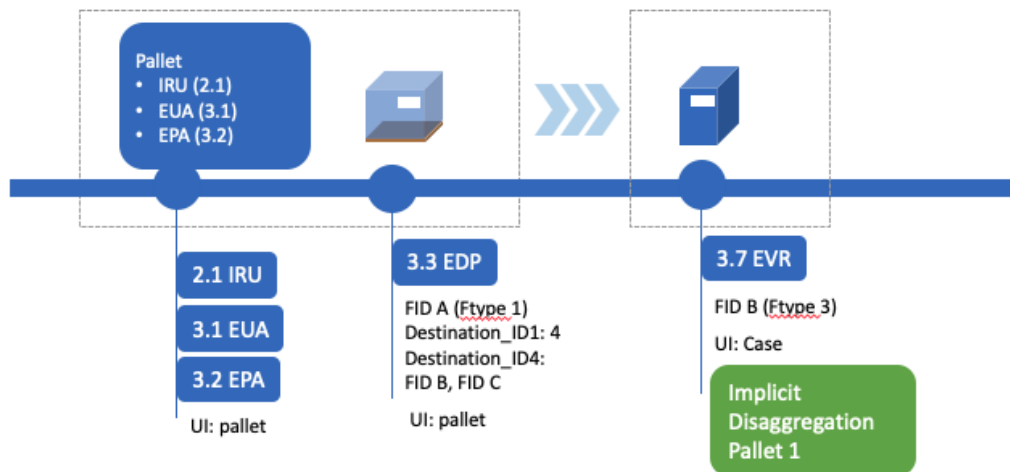
Figure 15 Implicit disaggregation trigger

6.8.7.6 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",

  "F_ID": "QCUKT<1AB020054000049",

  "Event_Time": "19032014",

  "Message_Time_Long":"2022-03-20T14:16:45Z",

  "UI_Type": 1,

  "upUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110", "
DANXXXXXXXXXXXX2PR012345678919030110" ],

  "aUIs": [ "DANXXXXXXXXXXXX10FA000000119030110" ],

  "Delivery_comment": "Comments",

  "Message_Type": "EVR"

}
```

6.8.7.7 JSON Successful response sample

```
HTTP Status 202
{
   "Code": "873345b2-882f-4064-91f0-90669b46c30a",

    "Message_Type": "EVR",

    "Error": false,

    "Errors": null,

    "Checksum": "G6HF5H"

}
```

6.8.7.8 Error response sample

HTTP status		
<< Common response code >>		
400	MULTIPLE_UI	VAL_UI_MULT_MSG
400	UI_NOT_EXIST UI_NOT_VALID	VAL_UI_EXIST_UPUI
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI
400	UI_NOT_VALID	VAL_UI_EXIST_UPUI_SEQ
400	UI_NOT_EXIST	VAL_UI_EXIST_AUI_SEQ
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_EOID
400	FID_NOT_EXIST_OR_ACTIVE	VAL_ENT_EXIST_FID
400	EOID_NOT_EXIST_OR_ACTIVE	VAL_ENT_ACTIVE_EOID
400	UI_DEACTIVATED	VAL_UI_ORD_DEACTIVATED
400	UI_ALREADY_DISAGGREGATED	VAL_UI_ORD_DISAGG or VAL_UI_ORD_IMPLDISAGG
299	SHIPMENT_WITHIN_24_HOURS	VAL_EVT_24H

6.9 Reporting transactional events (trade information)

6.9.1 EIV – (4.1) Issuing of the invoice

6.9.1.1 Description.

Report invoice details for an UI.

6.9.1.2 Description of the fields

Invoice reporting					
Field	Description	Data Type	Cardinality	Priority	Values
BasicInfo_Req	Block of basic information elements	Component << Basic	S	M	Message_Type = EIV

Invoice reporting					
Field	**Description**	**Data Type**	**Cardin ality**	**Priority**	**Values**
		Information Request >>			
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	
Invoice_Type1	Type of the invoice	Integer	S	M	See InvoiceType
Invoice_Type2	Description of the other type of the invoice	Text(5000)	S	M, if Invoice_ Type1 = 3	
Invoice_Number	Number of the invoice	Text(5000)	S	M	
Invoice_Date	Date of the invoice	Date	S	M	
Invoice_Seller	Identity of the seller	EOID	S	M	
Invoice_Buyer1	Identification if the buyer is located in the UK	Boolean	S	M	0 – No 1 – Yes
Invoice_Buyer2	Identity of the buyer	EOID	S	M, if Invoice_ Buyer1 = 1	
Buyer_Name	Buyer's registered legal name	Text(5000)	S	M, if Invoice_ Buyer1 = 0	
Buyer_Address	Buyer's address	Text(5000)	S	M, if Invoice_ Buyer1 = 0	
Buyer_Address_Name	Buyer's address - Name part of the Address	Text(5000)	S	O	
Buyer_Address_Street One	Buyer's address - Street part of the Address	Text(5000)	S	M, if Invoice_ Buyer1 = 0	
Buyer_Address_StreetT wo	Buyer's address - Second Element of the Street part of the Address	Text(5000)	S	O	

Invoice reporting					
Field	**Description**	**Data Type**	**Cardin ality**	**Priority**	**Values**
Buyer_Address_City	Buyer's address - City	Text(5000)	S	M, if Invoice_ Buyer1 = 0	
Buyer_Address_PostCo de	Buyer's address - PostalCode information	Text(5000)	S	O	
Buyer_CountryReg	Buyer's country of registration	Country	S	M, if Invoice_ Buyer1 = 0	
Buyer_TAX_N	Buyer's tax registration number	Text(5000)	S	M, if Invoice_ Buyer1 = 0	
First_Seller_UK	Identification if the invoice is issued by the first seller in the UK, i.e. the UK manufacturer or the importer, and the product is destined for the UK market	Boolean	S	M	0 – No 1 – Yes
Product_Items_1	List of TPIDs corresponding to the product items listed on the invoice	TPID	M	M, if First_Sell er_UK = 1	
Product_Items_2	List of product numbers corresponding to the product items listed on the invoice (in the same order as product_Items_1)	PN	M	M, if First_Sell er_UK = 1	
Product_Price	Net unit packet price per each pair of TPID and product number (in the same order as product_Items_1)	Decimal	M	M, if First_Sell er_UK = 1	
Invoice_Net	Total net amount of the invoice	Decimal	S	M	
Invoice_Currency	Currency of the invoice	Currency	S	M	
UI_Type	Identification of UI types covered by the invoice (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and

Invoice reporting					
Field	**Description**	**Data Type**	**Cardin ality**	**Priority**	**Values**
					aggregated level UIs
upUIs	List of unit packet level UIs covered by the invoice	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs covered by the invoice	aUI	M	M, if UI_Type = 2 or 3	
Invoice_comment	Comments by the reporting entity	Text(5000)	S	O	

6.9.1.2.1 Response

Invoice reporting– response					
Field	**Description**	**Data Type**	**Cardinality**	**Priorit y**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EIV

6.9.1.3 JSON Request sample

```
{
  "EO_ID": "QCUKR+1AB020054",
  "Event_Time": "19032014",
  "Message_Time_Long":"2019-03-20T14:16:45Z",
  "Invoice_Type1": 1,
  "Invoice_Type2": "other type",
  "Invoice_Number": "INV000001",
  "Invoice_Date": "2018-08-23T07:32:20.7878086+00:00",
  "Invoice_Seller": "SellerId",
  "Invoice_Buyer1": false,
  "Invoice_Buyer2": null,
  "Buyer_Name": "Buyer1",
  "Buyer_Address": "BuyerAddress",
  "Buyer_CountryReg": "LU",
  "Buyer_TAX_N": "TAX0001",
  "First_Seller_UK: 1,
  "Product_Items_1": [ "11111-1111111","11111-1111112" ],
  "Product_Items_2": [ "01234567891234","01234567891235"  ],
  "Product_Price": [ "16.99", "19.99" ],
```

```
  "Invoice_Net": 10099.99,

  "Invoice_Currency": "GBP",

  "UI_Type": 1,

  "upUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110",
"DANXXXXXXXXXXXX1PR012345678919030110" ],

  "aUIs": [ "DANXXXXXXXXXXX10FA000000119030110" ],

  "Invoice_comment": "Comments",

  "Message_Type": "EIV"

}
```

6.9.1.4 JSON Successful response sample

```
HTTP Status 202
{
   "Code": "873345b2-882f-4064-91f0-90669b46c30a",

   "Message_Type": "EIV",

   "Error": false,

   "Errors": null,

   "Checksum": "G6HF5H"

}
```

6.9.1.5 Error response sample

Processing errors

HTTP status	Error Code	Error Description
<< Common response code >>		

6.9.2 EPO – (4.2) Issuing of the order number

6.9.2.1 Description

Reports a purchase order event for an UI.

6.9.2.2 Description of the fields

Purchase order event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EPO

Purchase order event					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_l ong	Message sending Time	Time(L)	S	M	
Order_Number	Number of the purchase order	Text	S	M	
Order_Date	Date of the purchase order	Date	S	M	
UI_Type	Identification of UI types covered by the purchase order (recorded at the highest level of available aggregation)	Integer	S	M	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs covered by the purchase order	upUI(L)	M	M, if UI_Type = 1 or 3	
aUIs	List of aggregated level UIs covered by the purchase order	aUI	M	M, if UI_Type = 2 or 3	
Order_comment	Description of the reason for delayed recording of the purchase order	Text(5000)	S	O	

6.9.2.2.1 Response

Purchase order – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority y**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = EPO

6.9.2.3 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",
  "Event_Time": "22032014",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "Order_Number": "1234",
```

```
  "Order_Date": "2022-08-23T07:32:20.7878086+00:00",

  "UI_Type": 1,

  "upUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110",
"DANXXXXXXXXXXXX1PR012345678919030110" ],

  "aUIs": [ "DANXXXXXXXXXXXX10FA000000119030110" ],

  "Order_comment": "Comments",

  "Message_Type": "EPO"
}
```

6.9.2.4 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",

    "Message_Type": "EPO",

    "Error": false,

    "Errors": null,

    "Checksum": "G6HF5H"
}
```

6.9.2.5 Error response sample

Processing errors

HTTP status	Error Code	Error Description
<< Common response code >>		

6.9.3 EPR – (4.3) Receipt of the payment

6.9.3.1 Description

Adds a payment record event to a UI.

6.9.3.2 Description of the fields

		Payment record event			
Field	**Description**	**Data Type**	**Cardinal ity**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = EPR
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Event_Time	Time of event occurrence	Time(s)	S	M	
Message_Time_lo ng	Message sending Time	Time(L)	S	M	
Payment_Date	Date of the payment receipt	Date	S	M	
Payment_Type	Type of payment	Integer	S	M	See PaymentType
Payment_Amount	Amount of the payment	Decimal	S	M	
Payment_Currenc y	Currency of the payment	Currency	S	M	
Payment_Payer1	Identification if the payer is located in the UK	Boolean	S	M	0 – No 1 – Yes
Payment_Payer2	Identity of the payer	EOID	S	M, if Payment_P ayer1 = 1	
Payer_Name	Payer's registered legal name	Text(5000)	S	M, if Payment_P ayer1= 0	
Payer_Address	Buyer's address	Text(5000)	S	M, if Payment_P ayer1= 0	
Payer_Address_N ame	Buyer's address - Name part of the Address	Text(5000)	S	O	
Payer_Address_St reetOne	Buyer's address - Street part of the Address	Text(5000)	S	M, if Payment_P ayer1= 0	
Payer_Address_St reetTwo	Payer's address - Second Element of the Street part of the Address	Text(5000)	S	O	
Payer_Address_Ci ty	Payer's address - City	Text(5000)	S	M, if Payment_P ayer1= 0	

Payment record event					
Field	**Description**	**Data Type**	**Cardinal ity**	**Priority**	**Values**
Payer_Address_P ostCode	Payer's address - PostalCode information	Text(5000)	S	O	
Payer_CountryRe g	Payer's country of registration	Country	S	M, if Payment_P ayer1 = 0	
Payer_TAX_N	Payer's tax registration number	Text(5000)	S	M, if Payment_P ayer1 = 0	
Payment_Recipien t	Identity of the recipient	EIOD	S	M	
Payment_Invoice	Indication if the payment corresponds to the existing invoice	Boolean	S	M	0 – No 1 – Yes
Invoice_Paid	Number of the invoice paid with the payment	Text(5000)	S	M, if Payment_In voice = 1	
UI_Type	Identification of UI types covered by the payment (recorded at the highest level of available aggregation)	Integer	S	M, if Payment_In voice = 0	1 – only unit packet level UIs 2 – only aggregated level UIs 3 – both unit packet and aggregated level UIs
upUIs	List of unit packet level UIs covered by the payment	upUI(L)	M	M, if AND Payment_In voice = 0 UI_Type = 1 or 3	
aUIs	List of aggregated level UIs covered by the payment	aUI	M	M, if AND Payment_In voice = 0 UI_Type = 2 or 3	
Payment_comme nt	Comments by the reporting entity	Text(5000)	S	O	

6.9.3.2.1 Response

<table>
<tr><th colspan="6">Payment record – response</th></tr>
<tr><th>Field</th><th>Description</th><th>Data Type</th><th>Cardinality</th><th>Priority y</th><th>Values</th></tr>
<tr><td>BasicInfo_Resp</td><td>Block of basic information elements</td><td>Component << Basic Information Response >></td><td>S</td><td>M</td><td>Message_Type = EPR</td></tr>
</table>

6.9.3.3 JSON Request sample

```
{
  "EO_ID": "QCUKT+1AB020054",

  "Event_Time": "22032014",

  "Message_Time_Long":"2022-03-20T14:16:45Z",

  "Payment_Date": "2022-08-23T07:32:20.7878086+00:00",

  "Payment_Type": 1,

  "InvoiceType": 1,

  "UI_Type": 1,

  "Payment_Amount": 1.99,

  "Payment_Currency": "GBP",

  "Payment_Payer1": true,

  "Payment_Payer2": "PayerId",

  "Payer_Name": "PayerNmae",

  "Payer_Address": "Address",

  "Payer_CountryReg": "UK",

  "Payer_TAX_N": "TaxId",

  "Payment_Recipient": "PaymentRecipient",

  "Payment_Invoice": 1,

  "Invoice_Paid":  "test" ,

  "upUIs": [ "DANXXXXXXXXXXXX1PR012345678919030110",
"DANXXXXXXXXXXXX2PR012345678919030110" ],

  "aUIs": [ "DANXXXXXXXXXXXX10FA00000119030110", "DANXXXXXXXXXXXX20FA00000119030110"
],

  "Payment_comment": "Comments",

  "Message_Type": "EPR"

}
```

6.9.3.4 JSON Successful response sample

```
HTTP Status 202
```

```
{
    "Code": "873345b2-882f-4064-91f0-90669b46c30a",

    "Message_Type": "EPR",

    "Error": false,

    "Errors": null,

    "Checksum": "G6HF5H"
}
```

6.9.3.5 Error response sample

Processing errors

HTTP status		
<< Common response code >>		

6.10 Recall

6.10.1 RCL – (5.0) Recalls of requests, operational and transactional messages

6.10.1.1 Description

Given a recall id ("Code" in the return of any message) The caller can mark that event invalid.

This is possible for message types 2-1, 2-2, 3-A1 (code pairing), 3-1 to 3-7, 4-1, 4-2 and 4-3.

6.10.1.2 Description of the fields

Recall – request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Req	Block of basic information elements	Component << Basic Information Request >>	S	M	Message_Type = RCL
EO_ID	Economic operator identifier code of the submitting entity	EOID	S	M	
Message_Time_long	Message sending Time	Time(L)	S	M	

Recall – request					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Recall_CODE	Message recall code provided to the message sender in the acknowledgement of the original message to be recalled	UUID	S	M	
Recall_Reason 1	Reason for recalling the original message	Integer	S	M	See RecallReasonType
Recall_Reason 2	Description of the reason for recalling the original message	Text(5000)	S	M, if Recall_ Reason 1 = 3 (other reason)	
Recall_Reason 3	Any additional explanations on the reason for recalling the original message	Text(5000)	S	O	

6.10.1.2.1 Response

Recall – response					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
BasicInfo_Resp	Block of basic information elements	Component << Basic Information Response >>	S	M	Message_Type = RCL

6.10.1.3 JSON Request sample

```
{
  "EO_ID":"QCUKT+1AB020054",
  "Message_Time_Long":"2022-03-20T14:16:45Z",
  "RecallReason1":1,
  "RecallReason2":1,
  "RecallReason3":"Comments",
  "Message_Type":"RCL",
  "Recall_Code": "873345b2-882f-4064-91f0-90669b46c30a"
}
```

Note that "Recall_Code" is a reference to the message that it is wished to be recalled.

6.10.1.4 JSON Successful response sample

```
HTTP Status 202
{
    "Code": "873345b2-882f-4064-91f0-111122222222",
    "Message_Type": "RCL",
    "Error": false,
    "Errors": null,
    "Checksum": "G6HF5H"
}
```

Note that "Code" is the Recall Code of the Recall Message, provided by the UK Gateway.

6.10.1.5 Error response sample

HTTP status	Error Code	Error Description
<< Common response code >>		
400	RECALL_NOT_LAST_EVENT	Recall code not found: ERROR: The provided recall code does not exist. Recall code not the last event: ERROR: Please note that a recall can only be performed on valid messages that refer to UIs which were not later used in other messages.

7 List of Error Codes

7.1 Security errors

HTTP status	Error Code	Text Description
401	INVALID_OR_EXPIRED_TOKEN Related control: VAL_SEC_TOKEN	Error Descr: The incoming token is not valid or expired Comment: The security token has expired and should be renewed.

7.2 Processing errors

HTTP status	Error Code	Text Description
400	FAILED_VALIDATION Related control: VAL_FIE_REF	Error Descr: The field <XXX> should contain a valid <YYY> Error Descr for Circular Reference issue: The message contains UI values that form a circular reference Eg: The value for the field Aggregation_Type (**XXX**) is not in the defined set of values for AggregationType (**YYY**) (1 2 or 3) Comments: The values must match the values included in the set defined in the Data Dictionary. Concerning circular reference error: the UI mentioned in the event message is a parent of another UI present in the same message.
400	REQUIRED_FIELD_FAILED_VALIDATION Related controls: VAL_FIE_MAN VAL_MSG_JSON	Error Descr: The field <XXX> is required. VAL_FIE_MAN: Data missing in Mandatory field.

			Eg: field = ""
			VAL_MSG_JSON:
			Missing mandatory field.
			Eg: field = null or not present in the JSON
400	INVALID_MESSAGE_TYPE Related control: VAL_MSG_TYPE		Error Descr: Message type is unknown Comment: The type of Message you are using is not present in the Data Dictionary.
400	INVALID_SIGNATURE Related control: VAL_SEC_HASH		Error Descr: Hash information not matching the message signature Comment: The validation of the HASH of the body of the message doesn't match the transmitted HASH information in the header X-OriginalHash.
400	MAX_LENGTH_FAILED_VALIDATION Related control: VAL_MSG_JSON		Error Descr: The field <XXX> should be a value with maximum length of <Y> Comment: The message doesn't follow the specifications defined in the Data Dictionary. The number of characters must remain under the max length.
400	MIN_LENGTH_FAILED_VALIDATION Related control: VAL_MSG_JSON		Error Descr: The field <XXX> should be a value with minimum length of <Y> Comment: The message doesn't follow the specifications defined in the Data Dictionary. The number of characters must remain above the min length.
400	ENTRY_LENGTH_FAILED_VALIDATION Related control: VAL_MSG_JSON		Error Descr: The field <XXX> should be a 2-dimensional array where each row contains <Y> elements

			Comment: The message doesn't follow the specifications defined in the Data Dictionary.
400	INVALID_INPUT_FORMAT Related controls: VAL_MSG_JSON VAL_FIE_FORMAT		Error Descr: see examples below Example for 3.3 message (dispatch - EDP): n/a is a permitted value for the field 'Transport_vehicle' only if Transport_mode = 0 Example for recall messages: Please note that a recall cannot be performed on <XXX> messages. Comments: The message doesn't follow the specifications defined in the Data Dictionary. The body of the message contains at least one field in wrong format or does not correspond to a valid JSON message. Recall messages in particular cannot be performed on 2.1 messages (IRU), 2.2 messages (IRA) and 2.3 messages (IDA)
400	PAYLOAD_NOT_UNIQUE Related control: VAL_MSG_DUPLICATE		Error Descr: The message should contain a payload which was not previously used Comment: The system already processed the same payload delivered in an earlier message. You cannot resend the same payload.
400	EXCISE_NUMBER_NOT_VALID Related control: VAL_MSG_JSON		Error Descr: The field 'EO_ExciseNumber2' should contain a valid excise number Comment: The format of the field EO_ExciseNumber2 doesn't match the Data Dictionary.
400	NON_COMPATIBLE_UIS		Error Descr: the field 'upUI_2' should be compatible with 'upUI_1'

	Related control: VAL_MSG_JSON		Comment: Activation failed as ordered list of UIs with timestamp, did not match short UIs.
400	NOT_THE_SAME_NUMBER_OF_ITEMS Related control: VAL_MSG_JSON		Error Descr: For 3.1 message (activation - EUA): The field 'upUI_1' should contain the same number of items as 'upUIs_2' For 4.1 message (invoice - EIV): The field 'Product_Items_2' should contain the same number of items as 'ProductIdentifiers' And The field 'Product_Price' should contain the same number of items as 'ProductIdentifiers' Comments: 3.1 message (activation - EUA): Activation failed as number of UI with timestamp, did not same number as short UIs. 4.1 message (invoice - EIV): message failed as the items contained int the fields 'Product_Items_2' and/or 'Product_Price' are/is not the same as the number of items in 'ProductIdentifiers'.
400	MULTIPLE_UI Related control: VAL_UI_MULT_MSG		Error Descr: The field 'upUI'/'aUI' contains duplicate values Comment: Multiple duplicate UI present in the message lists. Message must contain only one occurrence of the same UI.
500	SYSTEM_ERROR		Error descr: Null Comment: The internal error ID should be provided to Dentsu support if required.

7.3 Validation Warning

HTTP status	Error Code	Text Description
299	OPERATION_WITHIN_24_HOURS Related control: VAL_EVT_24H	Warning Descr: Reporting events should be performed within 24 hours of the occurrence of the event (except dispatch and trans-loading events) Comment: You received this warning because this message has been reported late, i.e. more than 24 hours after the event time. Please note that the reporting time frame will be reduced to 3 hours starting from May 2028.
299	SHIPMENT_WITHIN_24_HOURS Related control: VAL_EVT_TIME	Warning Descr: The date/Time provided in the field 'Event_Time' should not be more than 24 hours ahead of the actual reporting time Comment: Dispatch and transloading events have to be reported within a time frame of 24 hours prior to the occurrence of the movement. Control is based on the "actual date – Event_Time" time difference.
299	UI_SEQUENCE_WARNING Related control: VAL_UI_ORD_SEQUENCE_WARNING	Warning Descr: <action> is not expected nor allowed when state is generated/deactivated/implicitly disaggregated Error Data: list of UI Comment: This error happens when you are trying to generate upUIs which are already existing in the UK repository, or to deactivate upUIs/aUIs already deactivated, or to deactivate aUIs already implicitly disaggregated.
299	UI_NOT_EXIST Related controls: VAL_UI_EXIST_UPUI VAL_UI_EXIST_AUI	Warning Descr: The field 'upUIs/aUIs' must contain elements that are already recorded and in one of the following states: Activated, Generated Error Data: list of UI

		Comment:
		This error is implemented for logistic actions, excluding UI activation.
		The most common reasons causing this message to occur are: a) the UI in question is a pre TPD/legacy UI never reported to the UK repository, b) the UI in question is a new UI not yet reported to the UK repository, c) The UI in question has not been encoded or decoded in line with the existing formatting instructions
		VAL_UI_EXIST_AUI
		Logistic action is not expected nor allowed if a UI does not exist (has not been part of an EPA message as a parent).
		VAL_UI_EXIST_UPUI
		Logistic action is not expected nor allowed if a UI does not exist (has not been part of any IRU message).

7.4 Validation errors

HTTP status	Error Code	Text Description
400	UI_NOT_VALID Related controls: VAL_UI_EXIST_UPUI_SEQ VAL_UI_EXIST_AUI_SEQ	Error Descr: <action> is not expected nor allowed when the pack has not been part of an application or an aggregation Error Data: list of UI Comment: Action on upUI is not expected nor allowed when the upUI has not been applied. Action on aUI is not expected nor allowed when the aUI has not been aggregated

400	UIS_APPLICATION_ERROR Related controls: VAL_UI_EXIST_APP VAL_UI_DUPLICATE_APP	Error Descr for 3.1 message (activation - EUA): Unique Identifier application on unit pack is not expected nor allowed when pack does not exist or has been reported to be already applied Error descr for 2.3 message (deactivation - IDA): Deactivation of upUI/aUI is not expected nor allowed when upUI/aUI does not exist Error Data: list of UI Comment: this error is generated when trying to activate/deactivate UIs which are not recorded in the UK repository, or which have already received an application event.
400	UI_DEACTIVATED Related controls: VAL_UI_ORD_REACTIVATION VAL_UI_ORD_DEACTIVATED	Error Descr for 3.1 messages (activation - EUA): Unique identifier application on unit pack is not expected nor allowed once the unique identifier has been deactivated Error Descr for other logistic actions: <Action> is not expected nor allowed once the unique identifier has been deactivated Error Data: list of UI Comment: The action you want to perform is not expected nor allowed after the UI has been deactivated with the 2.3 message.
400	MULTIPLE_AGGREGATION Related control: VAL_UI_ORD_AGG_MULT	Error Descr: Aggregation is not expected nor allowed when the pack is considered as already aggregated or implicitly disaggregated Error Data: list of UI Comment: Multiple aggregation identified for an aUI (as a parent) without having an explicit disaggregation of this aUI.
400	UI_ALREADY_DISAGGREGATED Related controls:	Error Descr: <action> is not expected nor allowed once the pack has been made explicitly available for aggregation after disaggregation

		VAL_UI_ORD_DISAGG	
		VAL_UI_ORD_IMPLDISAGG	Error Data: list of UI
			Comment: an aUI that has been disaggregated (explicitly or implicitly) cannot be part of any product movement prior of being aggregated.
400	LOCATION_MISMATCH		Error Descr: the FID must match with the location state of the reported unique identifiers
	Related controls:		
	VAL_UI_ORD_AGG_FID		Error Data: list of UI
	VAL_UI_ORD_DISPATCH		
			Comment:
			VAL_UI_ORD_AGG_FID
			All the goods must have been produced or reported to be in stock in the location where they are aggregated / disaggregated.
			VAL_UI_ORD_DISPATCH
			All the goods must have been produced or reported to be in stock in the location from which they are i dispatched.
400	FID_MISMATCH		Error Descr: the FID must match the FID specified in the meta data of unique identifiers
	Related Control:		
	VAL_UI_FID_APP		Comment: UI application in this location is not expected nor allowed as this location is not the one of the 2.1 message.
			Error Data: list of UI
400	ARRIVAL_NOTALLOWED		Error Descr: <action> not expected nor allowed when the pack is considered as in stock within EU facility following an application, arrival or return.
	Related Control:		
	VAL_UI_ORD_ARRIVAL		
	VAL_UI_ORD_ARRIVAL_RETURN		Error Data: list of UI
			Comment: arrival in UK facility is not expected nor allowed when the UI is considered as 'in

			stock' in a non-retail facility, meaning that a UI must have been part of a prior reported dispatch or transloading event before having an arrival.
			It is not allowed to arrive multiple times the same UIs.
			Exceptions:
			Imported products are allowed for arrival in UK facility without having any prior dispatch or transloading
400	UI_SEQUENCE_ERROR Related control: VAL_UI_ORD_SEQUENCE		Error descr: \<action> is not expected nor allowed when/once \<state> Error Data: list of UI Comment: this is a generic sequence validation error caused by a message sent out of the permitted sequence. In particular, an implicit disaggregation in transit is not allowed, meaning that the reporting of an arrival must be done with reference to the same UI(s) as reported for the purpose of preceding dispatch/transloading. Example for a 3.3 message (dispatch): *Dispatch from UK for delivery to retail destination is not expected nor allowed when the pack has been reported as dispatched for delivery to retail destination (DISPATCHED_UK_FIXED_QT_RETAIL/EDP_UK_FIXED_QT_RETAIL)* In this case, the dispatch is failing because it is including UIs which are in state dispatched Example for a 3.7 message (report of delivery through Vending Van) *Report of EU delivery with a vending van to retail outlet destination is not expected nor allowed when the pack has not been previously reported as dispatched for VV delivery (DISPATCHED_UK_FIXED_QT_RETAIL/EVR)*

			In this case, the delivery through Vending Van is not allowed because the previous event is a dispatch to a facility with delivery of fixed quantity. It should have been a dispatch with delivery with Vending Van.
400	UI_EXPIRED Related control: VAL_UI_EXPIRY		Error Descr: Some or all unique identifiers listed in the message have expired Error Data: list of UI Comment: Validation if the application or the aggregation date doesn't exceed the 6 months period after the generation of unique identifiers by the ID issuers.
400	EOID_NOT_EXIST_OR_ACTIVE Related controls: VAL_ENT_EXIST_EOID VAL_ENT_ACTIVE_EOID		Error Descr: EOID mentioned in the field 'EO_ID' is not marked as active in the repository Error Data: EOID Comment: The field 'EO_ID' must contain elements that are already recorded and active.
400	FID_NOT_EXIST_OR_ACTIVE Related controls: VAL_ENT_EXIST_FID VAL_ENT_ACTIVE_FID		Error Descr: FID mentioned in the field 'F_ID' is not marked as active in the repository Error Data: FID Comment: The field 'F_ID' must contain elements that are already recorded and active.
400	CODE_NOT_UNIQUE Related controls: VAL_RECALL_EXIST		Error Descr: The field 'Recall_Code' must contain a value which was not previously used Comment: The recall code provided has been used before. For recall messages, you cannot recall twice on the same recall code if the previous recall message succeeded.
400	CODE_NOT_EXIST Related control: VAL_RECALL_EXIST		Error Descr: The field 'Recall_Code' must contain elements that are already recorded Comment: The recall code provided has not been found into the UK repository.

400	RECALL_NOT_LAST_EVENT Related control: VAL_RECALL_LAST	Error Descr: Please note that a recall can only be performed on valid messages that referred to UIs which were not later used in other messages. Error Data: list of pair (UI @ Previous RecallCode) Comment: recalls can only be performed on last event.
400	PRINTED_CODES_ALREADY_US ED Related control: VAL_PRINTED_CODES_ALREADY _USED	Error Descr: The printed code has already been paired with another UI. Error Data: list of UI
400	PAIRED_CODES_ALREADY_USE D Related control: VAL_PAIRED_CODES_ALREADY_ USED	Error Descr: The code to be paired has already been paired with another UI. Error Data: list of UI

8 Endpoints

The following URLs are the intended system URLs but Dentsu reserves it's right to change them prior to the go-live. If this happens, Dentsu will communicate the change with sufficient time through the document center located at: https://uk-trackandtrace.dentsuaegistracking.com/

The PRE-PROD (Public QA/Integration environment) is expected to be available as of 1st of May 2022

The PROD URL (Production URL) is expected to be available as of 1st of July 2022.

Service	PRE-PROD URL (Public QA/Integration environment)
UK ID Issuer API	https://api.idissuer.qa-uk-trackandtrace-dentsutracking.com
UK ID Issuer API (Auth)	https://auth.qa-uk-trackandtrace-dentsutracking.com
UK ID Issuer Portal	https://idissuer.qa-uk-trackandtrace-dentsutracking.com
UK Gateway (JSON)	https://api.gateway.qa-uk-trackandtrace-dentsutracking.com
UK Gateway (Auth)	https://auth.qa-uk-trackandtrace-dentsutracking.com
Service	**PROD URL (Production URL)**
UK ID Issuer API	https://api.idissuer.uk-trackandtrace-dentsutracking.com
UK ID Issuer API (Auth)	https://auth.uk-trackandtrace-dentsutracking.com
UK ID Issuer Portal	https://idissuer.uk-trackandtrace-dentsutracking.com
UK Gateway (JSON)	https://api.gateway.uk-trackandtrace-dentsutracking.com
UK Gateway (Auth)	https://auth.uk-trackandtrace-dentsutracking.comhttps://auth.uk-trackandtrace-dentsutracking.com/

9 List of Standards

1	OAuth 2	https://www.oauth.com/oauth2-servers/access-tokens/client-credentials/
2	ISO/IEC 9834-8:2014 Information technology -- Procedures for the operation of object identifier registration authorities -- Part 8: Generation of universally Unique identifier (UI) (UUIs) and their use in object identifiers	https://www.iso.org/standard/62795.html
3	The JavaScript Object Notation (JSON) Data Interchange Format Internet Engineering Task Force (IETF) Request for Comments: 8259	https://tools.ietf.org/html/rfc8259
4	UUID	https://www.ietf.org/rfc/rfc4122.txt