

UK VDS LOGISTIC TRACEABILITY

API Technical Specifications

Version: v.1.7.0

Date: 10.09.2026

PUBLIC



Enabling trust

DOCUMENT HISTORY

VERSION	DATE	AUTHOR	COMMENTS
v.1.1	24.06.2025	C. York	First draft
v.1.2	07.04.2026	C. York	Updated to add specs to cover: 1) the endpoint to exchange the username/password for a token and 2) the endpoints to download the codes.
v.1.3	16.04.2026	Z. Belfkih	Added the GS1 Formats in new section 7.9
v.1.4	11.05.2026	J. Gardiol	Clarified expected values for readPoint and bizLocation in events. Improved examples.
v.1.5	28.05.2026	O. Ludwig	Adaptation for UK Vaping Duty Stamps solution. In the examples, replacement of lote code by HRC <i>urn:lote:id:pack:WAA1a0preri7fi2LFnap9gH06022514</i> by <i>urn:epc:id:ext:TSA/AHMEXRM02310</i>
v.1.5.1	12.06.2026	O. Ludwig	Corrections and complementary info on product codes.
v.1.6.0	03.07.2026	J. Gardiol	Removed section 7.9 Added URN format for UID-SSCC, added UID-SGTIN
v.1.6.1	22.07.2026	K. Pisnia	Added section on Void shipping event. Remove commissioning of UIPR (individual products)
v.1.7.0	10.09.2026	J. Gardiol & O.Ludwig	Enhanced documentation with added sections: Code State Machine (3) Anomalies (4) UID-EXT (8.2) – for product codes Identifiers (9) Connection Failure Handling and Retry Strategy (11) Updated event names: Decommissioning → Destroy Sales → Release for Consumption (10.8) Added Event: Error Declaration (10.10)

All material in these pages, including text, layout, presentation, logos, icons, photos and all other artwork including any derivative work, is the intellectual property of SICPA (for the purpose of this document the word "SICPA" shall mean SICPA HOLDING SA or any of its parent or affiliate companies) unless otherwise stated, and is protected by trademark, patent or copyright. No reproduction, derivative, or commercial use of any material is authorized without the prior express written permission by SICPA. Information contained in, or derived from, these pages shall not be used for development, production, marketing of any product or service, or for any other purpose, without the express prior written permission of SICPA.

TABLE OF CONTENTS

1 PREFACE	4
1.1 PURPOSE	4
1.2 INTENDED AUDIENCE	4
2 CAPTURING EVENTS.....	5
2.2 Capturing API	6
2.3 Events processing service.....	6
2.4 Anomalies processing service	6
3 CODE STATE MACHINE	7
4 ANOMALIES	8
5 CAPTURING API ENDPOINTS	9
5.1 Capture event.....	9
5.2 Event status	10
6 EVENT STATUSES.....	12
7 ERRORS	13
7.1 Capturing API errors	13
7.2 Events processing service errors	15
8 CODE FORMATS	17
8.1 UID-EXT (Product – UK VDS Stamp).....	17
8.2 UID-AG (Generated by UK VDS LOTE).....	17
8.3 UID-SSCC (Generated by Economic operator)	18
8.4 UID-SGTIN (Generated by Economic operator)	19
9 IDENTIFIERS.....	21
9.1 Site / Facility	21
9.2 Line / Machine	21
9.3 SKU / Product	21
10 EVENTS.....	22
10.1 Commissioning.....	22
10.2 Aggregation.....	24
10.3 Disaggregation	26
10.4 Shipping	27
10.5 Void shipping	28
10.6 Receiving.....	29
10.7 Decommissioning.....	31
10.8 RELEASE FOR CONSUMPTION (SALE).....	32
10.9 Return from customer.....	33
10.10 Void shipping.....	Error! Bookmark not defined.
10.11 Error declaration	34
11 CONNECTION FAILURE HANDLING AND RETRY STRATEGY.....	35

1 PREFACE

1.1 PURPOSE

This manual provides documentation for the RESTful Application Programming Interface (API) of the UK Vaping Duty Stamps solution for logistics traceability in the United Kingdom, hereafter called **UK VDS LOTE**. This API enables integration with your existing systems and applications, allowing you to automate sending events to UK VDS LOTE.

1.2 INTENDED AUDIENCE

This manual is designed for:

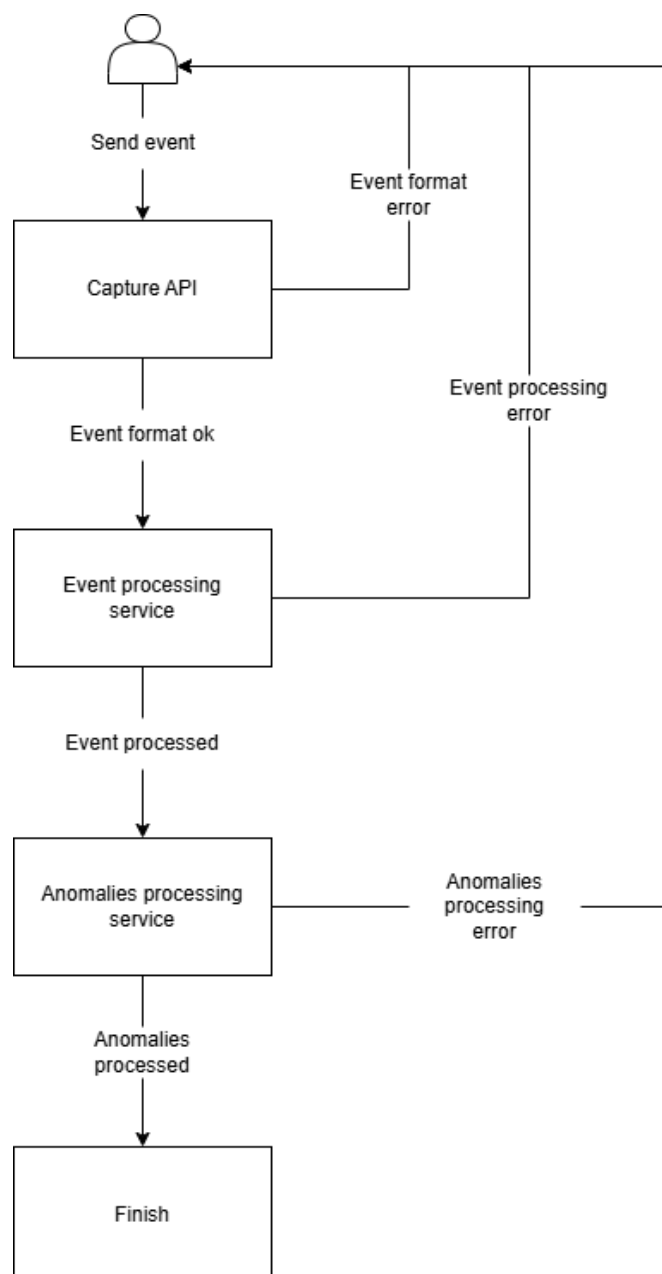
- **Business Analysts:** Analysts who need to understand the API's capabilities to support business processes.
- **Developers:** Software engineers and developers who will integrate the API into their applications.
- **IT Professionals:** IT administrators and system integrators responsible for deploying and managing the API.

2 CAPTURING EVENTS

Event capturing in UK VDS LOTE is done in 3 separate processes:

- Capturing API - synchronous
- Event processing service - asynchronous
- Anomalies processing service - asynchronous

Diagram below describes the process.



2.2 CAPTURING API

Events are sent to UK VDS LOTE using a single API endpoint for capturing any event.

Upon receiving the event, API will synchronously check basic formatting of the event message and either accept or reject the event with error message sent in response.

Events that result in error in this stage will be rejected from the system and will not be recorded as valid events in the events history, except for the records in audit trail and error logs.

2.3 EVENTS PROCESSING SERVICE

In case that event message is validated and accepted by API, it will then be processed asynchronously by the events processing service.

This service performs all the logical checks. In case there are any errors, event processing will be finished with an error status.

Such events will be recorded in events history but will not result in any of data recorded in UK VDS LOTE.

Events that are processed successfully will result in data recorded in UK VDS LOTE.

2.4 ANOMALIES PROCESSING SERVICE

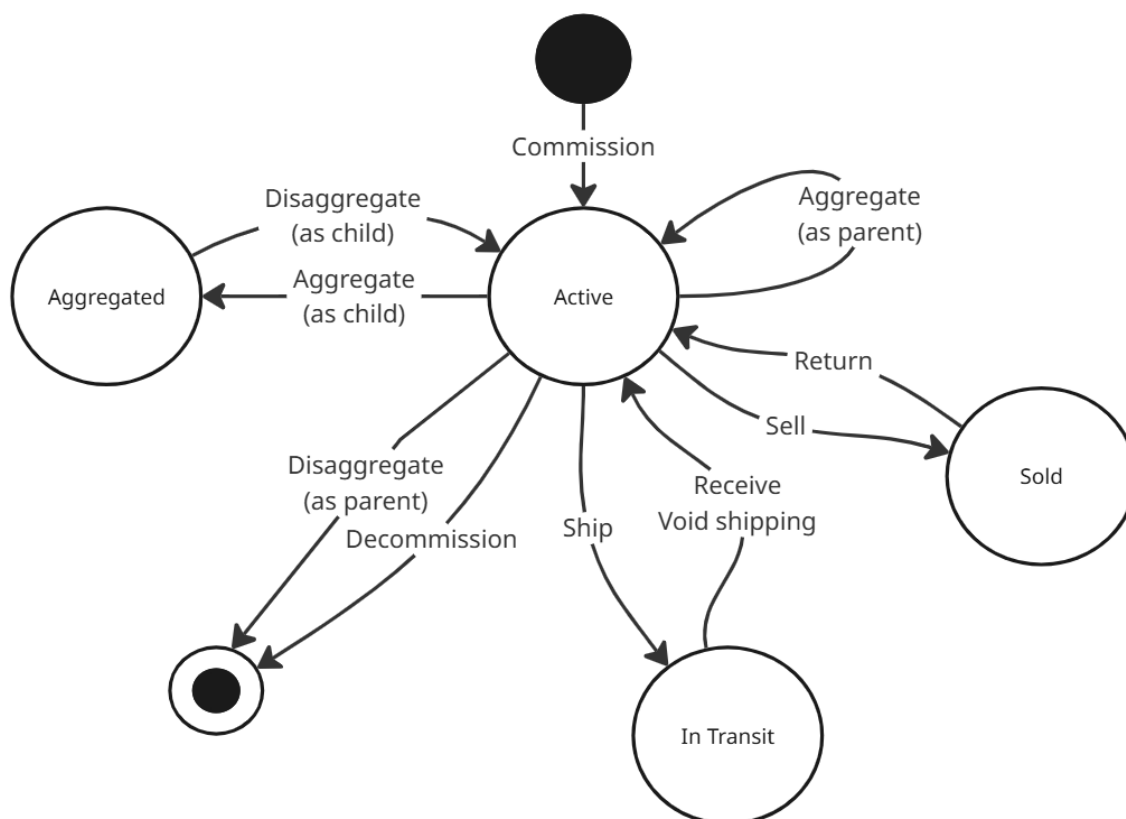
Events that are successfully processed are sent to anomalies processing service.

This service will mark invalid events order if found and record them as anomalies in the system.

It will also resolve any anomalies that are invalidated by receiving missing events from previous checks.

3 CODE STATE MACHINE

The various codes for products and aggregation packages follow the same general state machine based on the events that are sent.



- Codes start existing in the system with the commissioning event. Referring to any code that does not have a commissioning event will lead to an anomaly
- The diagram shows the allowed events based on the current state (outgoing arrows). Declaring any other event will lead to an anomaly
- Disaggregating a parent code or decommissioning any code (via destroy/lost/stolen/waste declaration) is final, the code may not be used in any event anymore
- All events can be canceled with an error declaration (see “Error declaration” section)

4 ANOMALIES

UK VDS LOTE is quite permissive in terms of events it accepts. The goal is to record any declaration that isn't malformed and refers to known identifiers. Business rules and coherence of events are enforced with anomalies.

There are 3 main categories of events:

- **Missing event:** These indicate that an event declaration is likely missing. Assuming the event declaration is correct, declaring the missing event should resolve the anomaly
- **Event data mismatch:** Some events should have matching data in their declarations. Shipping and receiving events should declare the same source and destination for example. Fixing these requires an error declaration and declaring the event with correct data.
- **Incorrect event location:** All events require declaring a location (Site/Facility identifiers) that should correspond to where the system thinks the codes are currently located. The only way a code can change location is a shipment (shipping/receiving events) or a sale (return may declared at another location than the sale event).

Category	Anomaly
Missing event	Missing commissioning
Missing event	Parent missing disaggregation
Missing event	Shipping missing before receiving
Missing event	Receiving event missing after shipping event
Missing event	Missing return from customer after sale
Missing event	Return from customer not after sale
Event data mismatch	Invoice mismatch between shipping source and receiving destination
Event data mismatch	Source and destination mismatch between shipping and receiving events
Event data mismatch	Return from customer invoice mismatch
Invalid event sequence	Destroy event not permitted
Incorrect event location	Aggregation not at last location
Incorrect event location	Shipping not at last event location
Incorrect event location	Destruction not at last event location
Incorrect event location	Sale not at last event location

5 CAPTURING API ENDPOINTS

5.1 CAPTURE EVENT

Method name	Capture event		
Method description	This endpoint is used for capturing any type of event in XML format in UK VDS LOTE system. Errors displayed from Capturing API errors Endpoint requires User's API-key in header. The API key will be provided by SICPA. It will be a string between 20 and 40 characters long.		
Method type	POST		
Method address	/captureEvent		
Headers	Content-Type: application/xml API-Key: {{API-KEY}}		
Request body XML	EPCIS event XML. See examples and definitions .		
Response XML (success)	<pre><?xml version="1.0" encoding="UTF-8" ?> <response> <status></status> <errorCode></errorCode> <errorMessage></errorMessage> <event_id></event_id> <timestamp></timestamp> </response></pre>		
Response XML (error)	<pre><?xml version="1.0" encoding="UTF-8" ?> <response> <status></status> <errorCode></errorCode> <errorMessage></errorMessage> <errorDescription></errorDescription> <errorTrace></errorTrace> <timestamp></timestamp> </response></pre>		
Response Fields description	Field	Field type	Field description
	status	Text	General request status
	errorCode	Integer	Error code, 0 if success
	errorMessage	Text	Error message title, empty if success
	errorDescription	Text	Error description, not returned if success
	errorTrace	Text	Error log information, not returned if success
	event_id	UUID	UUID of event, returned if success
	timestamp	Text	Date and time stamp of response
Version	1.0		

5.2 EVENT STATUS

Method name	Event Status
Method description	This endpoint is used for fetching information about the status of event with {{event_id}} Endpoint requires User's API-key in header
Method type	GET
Method address	/captureEvent/status/{{event_id}}
Headers	Content-Type: application/xml API-Key: {{API-KEY}}
Request body	None
Response JSON	<pre>{ "data": { "created_ts": "2025-05-27T22:43:02.000000+02:00", "created_users_id": 10, "created_users_username": "username", "date_processed": "2025-05-27T22:43:08.000000+02:00", "date_uploaded": "2025-05-27T22:43:02.000000+02:00", "description": "", "eo_id": "EOID", "event_body": "<?xml version=\"1.0\"?>\r\n<epcis:EPCISDocument xmlns:epcis=\"urn:epcglobal:epcis:xsd:1\" xmlns:cbvmd=\"urn:epcglobal:cbv:mda\" xmlns:lote=\"urn:sicpa:lote\" schemaVersion=\"1.2\" creationDate=\"2025-06- 03T22:42:01.552Z\">\r\n <EPCISBody>\r\n <EventList>\r\n <ObjectEvent>\r\n <eventTime>2025-06-03T22:42:01.552Z</eventTime>\r\n <recordTime>2025-06-03T22:42:01.552Z</recordTime>\r\n <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>\r\n <baseExtension>\r\n <eventID>urn:uuid:1cfe4f8b- cd15-4d7d-98b3-5744cf1168f4</eventID>\r\n </baseExtension>\r\n <epcList>\r\n <epc>urn:epc:id:ext:TSA/AHMEXRM02310</epc>\r\n </epcList>\r\n <action>DELETE</action>\r\n <bizStep>urn:epcglobal:cbv:bizstep:destroying</bizStep>\r\n <disposition>urn:epcglobal:cbv:disp:destroyed</disposition>\r\n <readPoint>\r\n <id>urn:lote:id:loc:WAA1ades2</id>\r\n </readPoint>\r\n <bizLocation>\r\n <id>urn:lote:id:loc:WAA1ades2</id>\r\n </bizLocation>\r\n </ObjectEvent>\r\n </EventList>\r\n </EPCISBody>\r\n </epcis:EPCISDocument>", "event_status": 60, "event_status_name": "finished processing anomalies ok", "event_ts": "2025-05-27T08:30:16.000000+02:00", "event_type": 50, "event_type_name": "receiving", "event_uid": "4347d442-282a-4e45-a485-22a73575fb61", "id": 348 } }</pre>

Method name	Event Status		
	<pre> }, "errorCode": 0, "errorMessage": "", "requestID": "", "status": "success", "timestamp": "2025-05-27T22:46:46.285806+02:00" } </pre>		
Response Fields description	Field	Field type	Field description
	status	Text	General request status
	errorCode	Integer	Error code, 0 if success
	errorMessage	Text	Error message title, empty if success
	errorDescription	Text	Error description, not returned if success
	errorTrace	Text	Error log information, not returned if success
	data	JSON object	Event details
	data.created_ts	Timestamp	Timestamp when event was captured
	data.created_users_id	Integer	ID of the user who sent event
	data.created_users_username	Text	Username of the user who sent event
	data.date_processed	Timestamp	Timestamp when event was processed
	data.date_uploaded	Timestamp	Timestamp when event was uploaded
	data.description	Text	Empty if no errors. In case of errors, displays JSON formatted string with errors from Events processing service errors .
	data.eo_id	Integer	EO_ID of the user who sent event
	data.event_body	Text	Event document XML
	data.event_status	Integer	Event status ID
	data.event_status_name	Text	Event status name
	data.event_ts	Timestamp	Event time
	data.event_type	Integer	Event type ID
	data.event_type_name	Text	Event type name
	data.event_uid	UUID	Event UUID
	data.id	Integer	Event ID
Version	1.0		

6 EVENT STATUSES

Event status code	Event Status	Description
10	uploaded	Event has processed by Capturing API and accepted for processing
20	processing	Event is being processed by Events processing service
30	finished processing ok	Event has been processed by Events processing service successfully. Event is available in Track and Trace. Event is sent to Anomalies processing service
40	error in processing	Events processing service detected error. Event is not available in Track and Trace, processing finished
50	processing anomalies	Event is being processed by Anomalies processing service
60	finished processing anomalies ok	Event has been processed by Anomalies processing service successfully. Processing finished
70	error in processing anomalies	There was an error in Anomalies processing service.

7 ERRORS

7.1 CAPTURING API ERRORS

Errors returned synchronously by Capturing API endpoint.

Status code	errorCode	errorMessage	errorDescription	Description
405	100405	405 Method Not Allowed: The method is not allowed for the requested URL. URL::URL		Endpoint method not allowed
404	100404	404 Not Found: The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again. URL::URL		Endpoint does not exist
500	100500	Unexpected system error		Unexpected system error
400	100400	Bad request	For EPCIS document mimetype must be XML	Event is not XML document
400	9999999	InternalToolsError.message	InternalToolsError.description	
400	9999999	General Error	(Exception)	
400	9999999	MySQL DatabaseError.	Event {UUID} already exists!	Event with UUID already exists in the system
400	100501	Error parsing EPCIS document.	EPCIS schema version must be '1.2'.	Event is not EPCIS 1.2 document
400	100501	Error parsing EPCIS document.	Error parsing 'eventTime'. Missing or not valid format.	eventTime is not valid
400	100501	Error parsing EPCIS document.	Error parsing 'captureTime'	captureTime is not valid
400	100501	Error parsing EPCIS document.	Time {eventTime}: [eventTimeTimestamp] is greater than time {captureTime}: [captureTimeTimestamp], Using time_delta: {300}	eventTime is greater than captureTime using time delta
400	100501	Error parsing EPCIS document.	Batch object is not supported.	
400	100501	Error parsing EPCIS document.	URN type is not supported. Only supported 'epc' type is 'id'.	URN is invalid. Must start with "id"
400	100501	Error parsing EPCIS document.	Only supported 'lote' type is 'pack', 'agg' and 'loc'.	Invalid type for lote URN
400	100501	Error parsing EPCIS document.	Only supported 'lote' type is 'id'.	Lote URN must start with "id"
400	100501	Error parsing EPCIS document.	Missing event location. Unable to find 'bizLocation' or 'readPoint'	Event location missing
400	100501	Error parsing EPCIS document.	Missing 'baseExtension:eventID' in event.	Event ID missing
400	100501	Error parsing EPCIS document.	Invalid 'eventTimeZoneOffset' in EPCIS document. Must be between '-18:00' and '+18:00'.	eventTimeZoneOffset out of range



Status code	errorCode	errorMessage	errorDescription	Description
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in Commissioning event. Only 'active' is supported.	Commissioning disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in Commissioning event. Only 'add' is supported.	Commissioning action invalid
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in 'destroy' business step. Only 'stolen, expired, damaged, destroyed' are supported.	Destroy disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in 'destroy' business step. Only 'delete' is supported.	Destroy action invalid
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in Shipping event. Only 'in_transit' is supported.	Shipping disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in Shipping event. Only 'observe' is supported.	Shipping action invalid
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in Receiving business step. Only 'in_progress' and 'returned' are supported.	Receiving disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in Receiving business step. Only 'observe' is supported.	Receiving action invalid
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in Sale event. Only 'retail_sold' is supported.	Sale disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in Sale event. Only 'observe' is supported.	Sale action invalid
400	100501	Error parsing EPCIS document.	Unsupported bizStep[{}] in object event.	bizStep invalid
400	100501	Error parsing EPCIS document.	Unsupported 'bizStep' in Aggregation/Disaggregation event. Only 'packing' and 'unpacking' are supported.	Aggregation/Disaggregation bizStep invalid
400	100501	Error parsing EPCIS document.	Unsupported 'disposition' in Aggregation/Disaggregation event. Only 'in_progress' is supported	Aggregation/Disaggregation disposition invalid
400	100501	Error parsing EPCIS document.	Unsupported 'action' in Aggregation/Disaggregation event. Only 'add' and 'delete' are supported.	Aggregation/Disaggregation action invalid
400	100501	Error parsing EPCIS document.	Invalid type of 'bizTransaction' in EPCIS document. Must be 'urn:epcglobal:cbv:btt:inv'	Invoice tag invalid
400	100501	Error parsing EPCIS document.	Missing 'intendedRoute' in EPCIS document.	Missing intendedRoute in shipping, receiving
400	100501	Error parsing EPCIS document.	Missing 'customer' in EPCIS document.	Missing customer in sale, return
400	100501	Error parsing EPCIS document.	Only one 'extension.sourceList' is allowed in EPCIS document.	Multiple sources in event
400	100501	Error parsing EPCIS document.	Only one 'extension.destinationList' is allowed in EPCIS document.	Multiple destinations in event
400	100501	Error parsing EPCIS document.	Invalid type of 'extension.sourceList' in EPCIS document. Must be 'urn:epcglobal:cbv:sdt:location'	Invalid sourceList
400	100501	Error parsing EPCIS document.	Invalid type of 'extension.destinationList' in EPCIS document. Must be 'urn:epcglobal:cbv:sdt:location'	Invalid destinationList
400	100501	Error parsing EPCIS document.	Missing or empty list 'epcList' in EPCIS document.	Missing EPC list in Commissioning, Destroy, Shipping, Receiving, Sale, Return
400	100501	Error parsing EPCIS document.	Missing or empty list 'childEPCIs' in EPCIS document.	Missing children list in Aggregation, Disaggregation
400	100501	Error parsing EPCIS document.	Unable to convert string[{TIMESTAMP}] to datetime	Date time invalid
400	100501	Error parsing EPCIS document.	Content error. Body is empty or missing 'event' tag.	Invalid event
400	100501	Error parsing EPCIS document.	Content error. Source and destination location are the same.	Source and destination are same

Status code	errorCode	errorMessage	errorDescription	Description
400	100501	Error parsing EPCIS document.	Error parsing Commissioning Event.	
400	100501	Error parsing EPCIS document.	Error parsing Aggregation Event.	
400	100501	Error parsing EPCIS document.	Error parsing Disaggregation Event.	
400	100501	Error parsing EPCIS document.	Error parsing Shipping Event.	
400	100501	Error parsing EPCIS document.	Error parsing Sale Event.	
400	100501	Error parsing EPCIS document.	Error parsing Receiving Event.	
400	100501	Error parsing EPCIS document.	Error parsing Return From Customer Event.	
400	100501	Error parsing EPCIS document.	Error parsing Destroy Event.	

7.2 EVENTS PROCESSING SERVICE ERRORS

Errors display in Event status endpoint.

errorCode	errorMessage	errorDescription	Description
1001	Unauthorised user!		User API key is invalid or has no rights on objects or locations
10004	Error getting location name	Error getting location name for location id '{LOCATIONID}'. Missing from master data records	Location does not exist or user does not have rights
10004	Error getting location name	General error getting location from database:: [{}]	Unhandled error getting locations
10002	Error getting country name	Error getting country name for country id '{COUNTRYID}'. Missing from master data records	
10002	Error getting country name	General error getting country from database:: [{}]	Unhandled error getting country
10005	Error getting machine info	Error getting machine info for location id '{MACHINEID}'. Missing from master data records	Machine does not exist or user does not have rights
10005	Error getting machine info	General error getting machine info from database:: [{}]	Unhandled error getting machine
10003	Error getting product name	Error getting product name for product id '{SKU}'. Missing from master data records	Product does not exist or user does not have rights
10003	Error getting product name	General error getting product from database:: [{}]	Unhandled error getting product
9999999	Error parsing URN code	Error parsing code '{CODE}'. Check documentation for valid format	Invalid EPC
9999999	Invalid URN code	Expected urn:lote:id:loc:xxxxx but got {}	Invalid location URN
9999999	Invalid URN code	Expected urn:lote:epcis:bt:inv:xxxxx but got {}	Invalid invoice URN
9999999	Invalid URN code	Expected urn:lote:name:custxxxxx but got {}	Invalid customer URN



errorCode	errorMessage	errorDescription	Description
9999999	Unable to set all PACK level objects	List of object missing in database or in wrong status: {LIST}	Invalid or missing EPCs is event
9999999	Unable to set all PACK level objects	Duplicate IDs found in the list: {LIST}	Duplicates found in EPCs list
9999999	Unable to set all PACK level objects	Different full codes found in the list: {LIST}	Serial numbers used in more than one UID-PR
9999999	Error parsing event.	eventID must be UUID version 4.	Invalid event UUID
9999999	Error parsing event.	eventID must be defined and start with 'urn:uuid:'.	Invalid event UUID
9999999	Error processing object.	'ReadPoint' must be the same as UID-MA extracted from EPC: readPoint[{READPOINT}], UID_MA[{UID-MA}].	In commissioning,UID-PRs must contain machine ID same as in read point.
9999999	Error processing object	All objects must be of same type. In your event we have: pack[{}], aggregation[{}], external_code[{}], SSCC[{}].	Commissioning is allowed only on same UID type
9999999	Invalid children object	Children object must be empty in Disaggregation event.	
9999999	Invalid children object	Not allowed children object to be the same as parent object	Child cannot be the same as parent
9999999	Invalid children object type	Not allowed children object type SSCC in Aggregation event	SSCC cannot be child
9999999	Invalid parent object type	Not allowed parent object type PACK in Aggregation event	UID-PR cannot be parent
9999999	Error executing items_update_status_insert	Error processing event '{}'. Error[{}]: {}	
9999999	Error executing items_update_status_insert	General error executing items_update_status_insert:: [{}]	
9999999	Error executing evn_offline_upload_data	Error processing event '{}'. Error code: {}	
9999999	Error executing evn_offline_upload_data	General error executing evn_offline_upload_data:: [{}]	

8 CODE FORMATS

8.1 UID-EXT (PRODUCT – UK VDS STAMP)

The UID-EXT is the content of the visible datamatrix on the SICPA tax stamp. It is composed of a prefix that depends on the stamp type (either TSA/ or DRY/) and a sequence of 12 alpha-numeric characters that are also printed in human-readable form (sometimes referred to as HRC, for Human Readable Code) on the stamp.

Example: TSA/4F7QDQ2VGV44

8.2 UID-AG (GENERATED BY UK VDS LOTE)

The UID-AG is based on ISO/IEC 15459:2014 standard and has the following format.

Block	A		B
Block Name	Unique Identification Code (UIC)	UID Prefix	Serial Number
Block Details	Authorised character set is [a-z][A-Z][0-9] → 62 possible characters		
Length	5 characters	1 character	20 characters
Example	WAA1a	1	WSXEDC34tf78H0ZTHTZT
Generated by	UK VDS LOTE		UK VDS LOTE

Block / Attribute	Value
Unique Identification Code (UIC)	WAA1a (constant value)
UID Prefix	1 (constant value for UID-AG)

8.3 UID-SSCC (GENERATED BY ECONOMIC OPERATOR)

The SSCC format is defined in ISO/IEC 15417: 2007. It is made-up of the following fields (18 characters excluding the GS1 Application Identifier):

Field	Description	Length	Value	Comment
GS1 Application Identifier		2 numeric char	Fixed: 00	"(" and ")" excluded from the field itself and only printed in the HRC of the Code 128
Extension Digit	GS1's IAC	1 numeric char	Variable: Between 0 and 9	0 to 9 are the IAC allocated to GS1 by the RA (according to ISO 15459-1:2014), hence this extension ID is the GS1's IAC
GS1 Company Prefix	CIN allocated by GS1 to the manufacturer (i.e. identity issuer)	N numeric char	Variable	
Serial Reference	Serial reference allocated to the transport unit by the manufacturer (i.e. identity issuer)	M numeric char	Variable	
Check Digit	Check Digit calculated on the basis of the preceding 17 digits	1 alphanumeric char	Variable	The Check Digit is calculated using the algorithm defined by GS1 [GENSPECS, 7.9.1]

Block / Attribute	Value
GS1 Application Identifier	00 (constant value)
Extension Digit	Variable: between 0 and 9
GS1 Company Prefix	CIN allocated by GS1 to the economic operator

EPCIS URN

In EPCIS messages, the UID-SSCC must be provided in URN format:

urn:epc:id:sscc:<company_prefix>.<extension_digit><serial_number>

Example value: urn:epc:id:sscc:8410100.0000000001

Important notes:

- Usage of UID-SSCC requires that the GS1 company prefix is registered in LOTE.

- It is up to the economic operator to manage the serial numbers to ensure that UID-SSCC is unique in the system.
- UID-SSCC may only be used for top-level aggregation units. UID-SSCCs may not be declared as children of another code in an aggregation event.

8.4 UID-SGTIN (GENERATED BY ECONOMIC OPERATOR)

The SGTIN format is defined in ISO/IEC 16022: 2006. It is made-up of the following fields:

Field	Description	Length	Value	Comment
<FNC1>	Special character mandated by GS1	N/A	N/A	
Application Identifier	Application Identifier used as GTIN prefix	2 numeric chars	Fixed: 01	
GTIN-14	GTIN of the SKU as recorder in GS1	14 numeric chars	Variable	GTIN-14 or EAN-13 prefixed by "0".
Application Identifier	Application Identifier used as Serial Number Prefix	2 numeric chars	Fixed: 21	
Serial Number	Serial Number from GS1's 82 character set	Variable (max 20 char)	Variable	
Application Identifier	Application Identifier used as extra string prefix for additional data that the manufacturer wants to add	3 or 2 numeric chars	Fixed: 240 or 10	Optional.
Extra string	Additional data that the manufacturer wants to add	Variable	Variable	Optional.

Block / Attribute	Value
GS1 Application Identifier	01 (constant value)
Extension Digit	Variable: between 0 and 9
GS1 Company Prefix	CIN allocated by GS1 to the economic operator

EPCIS URN format

In EPCIS messages, the UID-SGTIN must be provided in URN format:

urn:epc:id:sgtin:<gs1_company_prefix>.<extension digit><item_reference>.<serial_number>

Example value: urn:epc:id:sgtin:8410100.000001.ABCD01

Important notes:

- Usage of UID-SGTIN requires that the GS1 company prefix is registered in UK VDS LOTE.
- It is up to the economic operator to manage the serial numbers to ensure that UID-SGTIN is unique in the system.

9 IDENTIFIERS

9.1 SITE / FACILITY

Site or Facility identifiers in EPCIS messages are composed of 3 parts:

- A URN prefix, urn:lote:id:loc:
- A static prefix, WAA1a
- The site identifier, that can be found in the UK VDS Web UI, typically 1 or more digits. For example, “17”.

Full example: urn:lote:id:loc:WAA1a17

9.2 LINE / MACHINE

Line or Machine identifiers in EPCIS messages are composed of 3 parts:

- A URN prefix, urn:lote:id:loc:
- A static prefix, WAA1a
- The line identifier, that can be found in the UK VDS Web UI, typically 4 alpha-numeric characters. For example, “FG5R”.

Full example: urn:lote:id:loc:WAA1aFG5R

9.3 SKU / PRODUCT

SKU or product identifiers in EPCIS messages are composed of just the SKU identifier that can be found in the UK VDS Web UI, typically 4 alpha-numeric characters, for example “VY45”.

10EVENTS

10.1COMMISSIONING

Commissioning event is an event in which a new object is created with a corresponding ID in the system. This event is a prerequisite for all events that occur after it, because through this event IDs for products, logistics and aggregation IDs are generated, which are later used in the aggregation and movement of products through the supply chain (shipping, receiving).

In the examples below, the product codes such as TSA/AHMEXRM02310 are unique identifiers per product, that are generated by the VDS Solution at the coding stage. They are composed of a prefix meant to identify the type of stamp (in this example “TSA” that corresponds to Sef-Adhesive type, DRY will be used to identify dry labels) followed by "/" and a string of 12 alphanumeric characters forming the Human Readable Code (HRC). These data materialized in the form of a 12x26 Datamatrix 2D code printed on the stamp affixed on the product.

Commissioning of individual products is not shown and is not necessary. In the UK VDS context, the product code is the above-mentioned visible DMX visible code that contains the HRC. That same code is used by the system for activation of stamps. Activation then triggers commissioning (so-called auto-commissioning).

10.1.1 AGGREGATION LEVEL (UID-AG GENERATED BY UK VDS)

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2"
creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T00:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:32d018f0-24c7-4ee4-80c3-6453b64b7b23</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
        <disposition>urn:epcglobal:cbv:disp:active</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1aFG5R</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

10.1.2 AGGREGATION LEVEL (UID-SSCC GENERATED BY ECONOMIC OPERATOR)

The SSCC is the highest aggregation level and corresponds to the shipped item

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T00:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:32d018f0-24c7-4ee4-80c3-6453b64b7b23</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:epc:id:sscc:385123.04567891234</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
        <disposition>urn:epcglobal:cbv:disp:active</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1aFG5R</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

10.1.3 AGGREGATION LEVEL (UID-SGTIN GENERATED BY ECONOMIC OPERATOR)

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T00:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:32d018f0-24c7-4ee4-80c3-6453b64b7b23</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:epc:id:sgtin:8410100.000001.ABCD01</epc>
        </epcList>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
        <disposition>urn:epcglobal:cbv:disp:active</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1aFG5R</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T00:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	32d018f0-24c7-4ee4-80c3-6453b64b7b23
<epc>	Item	urn	<epc>urn:epc:id:ext:TSA/AHMEXRM02310</epc> <epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc> <epc>urn:epc:id:sscc:385123.04567891234</epc> <epc>urn:epc:id:sgtin:8410100.000001.ABCD01</epc>
<readPoint>	Machiiine where event is captured	urn	<id>urn:lote:id:loc:WAA1aFG5R</id>
<bizLocation>	Facility where event happened	urn	<id>urn:lote:id:loc:WAA1a17</id>

10.2 AGGREGATION

An aggregation event is an event where a hierarchy is created in product packaging, from basic packaging to the largest logistics packaging. Through this event, parent-child relationships are created between the objects created by the commissioning event.

```

<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2024-05-24T01:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:42d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <parentID>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</parentID>
        <childEPCs>
          <epc>urn:epc:id:ext:TSA/AHMEXRM02310</epc>
          <epc>urn:epc:id:ext:TSA/AHMEXRM02311</epc>
          <epc>urn:epc:id:ext:TSA/AHMEXRM02312</epc>
          <epc>urn:epc:id:ext:TSA/AHMEXRM02313</epc>
        </childEPCs>
        <action>ADD</action>
        <bizStep>urn:epcglobal:cbv:bizstep:packing</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1a17</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </AggregationEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>

```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T01:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	42d018f0-24c7-4ee4-80c3-6453b64b7b23

<parentID>	Parent in aggregation	urn	<parentID>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</parentID>
<childEPCs>	List of children in aggregation	List of urn	
<epc>	Item	urn	<epc> urn:epc:id:ext:TSA/AHMEXRM02310</epc> <epc> urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP4</epc>
<readPoint>	Facility where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<bizLocation>	Facility where event happened	urn	<id>urn:lote:id:loc:WAA1a17</id>

10.3 DISAGGREGATION

Disaggregation event is an event where a hierarchy is destroyed in product packaging, from basic packaging to the largest logistics packaging. Through this event, parent-child relationships are broken between the objects and parent object is being decommissioned. For children, parents are removed and status stays the same.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <AggregationEvent>
        <eventTime>2024-05-24T01:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:42d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <parentID>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</parentID>
        <childEPCs />
        <action>DELETE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:unpacking</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1a17</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </AggregationEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T01:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	42d018f0-24c7-4ee4-80c3-6453b64b7b23
<parentID>	Parent in aggregation	urn	<parentID>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</parentID>
<childEPCs>	List of children in aggregation	List of urn	
<epc>	Item	urn	<epc> urn:epc:id:ext:TSA/AHMEXRM02310</epc> <epc> urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP4</epc>
<readPoint>	Facility where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<bizLocation>	Facility where event happened	urn	<id>urn:lote:id:loc:WAA1a17</id>

10.4 SHIPPING

A shipping event is an event for sending products from one location to another location. With this event, products and their logistics packages move through the supply chain, with or without a change of ownership.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:lotec="urn:sicpa:lotec:xsd:1"
schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T02:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:52d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lotec:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
        <readPoint>
          <id>urn:lotec:id:loc:WAA1a17</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
type="urn:epcglobal:cbv:btt:inv">urn:lotec:epcis:bt:inv:12345</bizTransaction>
        </bizTransactionList>
        <extension>
          <sourceList>
            <source type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a17</source>
          </sourceList>
          <destinationList>
            <destination
type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a18</destination>
          </destinationList>
        </extension>
        <lotec:intendedRoute>Lausanne - Zurich - Zagreb</lotec:intendedRoute>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T02:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	52d018f0-24c7-4ee4-80c3-6453b64b7b23
<epcList>	List of shipped epcs	List of urn	
<epc>	Item	urn	<epc>urn:lotec:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
<readPoint>	Location where event is captured	urn	<id>urn:lotec:id:loc:WAA1a17</id>
<source>	Source location of shipping	urn	<source type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a17</source>
<destination>	Destination location of shipping	urn	<destination type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a18</destination>
<bizTransaction>	Invoice ID (optional)	urn	urn:lotec:epcis:bt:inv:12345
<lotec:intendedRoute>	Intended route	String	Lausanne – Zurich - Zagreb

10.5 VOID SHIPPING

The void shipping event can be used to cancel the shipping declaration of a code. For the shipping declaration to be identified correctly, all these fields must match: readPoint, source, destination, invoice ID. Any EPC declared in the shipping event can be referenced in the void shipping event.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:lote="urn:sicpa:lote:xsd:1"
schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T07:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:82d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lote:id:agg:WAA112NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:void_shipping</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA11FG5R</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
type="urn:epcglobal:cbv:bt:inv">urn:lote:epcis:bt:inv:12345</bizTransaction>
        </bizTransactionList>
        <extension>
          <sourceList>
            <source type="urn:epcglobal:cbv:sdt:location"> urn:lote:id:loc:WAA11FG5R
            </source>
          </sourceList>
          <destinationList>
            <destination
type="urn:epcglobal:cbv:sdt:location">urn:lote:id:loc:WAA11FGT6</destination>
          </destinationList>
        </extension>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T02:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	52d018f0-24c7-4ee4-80c3-6453b64b7b23
<epcList>	List of shipped eps	List of urn	
<epc>	Item	urn	<epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
<readPoint>	Location where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<source>	Source location of shipping	urn	<source type="urn:epcglobal:cbv:sdt:location">urn:lote:id:loc:WAA11FG5R</source>
<destination>	Destination location of shipping	urn	<destination type="urn:epcglobal:cbv:sdt:location">urn:lote:id:loc:WAA11FGT6</destination>
<bizTransaction>	Invoice ID (optional)	urn	urn:lote:epcis:bt:inv:12345

10.6 RECEIVING

A receiving event is an event for receiving products previously shipped in shipping event on destination location. With this event, paired with shipping event, products and their logistics packages move through the supply chain, with or without a change of ownership.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:lotec="urn:sicpa:lotec:xsd:1"
schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T03:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:62d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lotec:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
        <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
        <readPoint>
          <id>urn:lotec:id:loc:WAA1a18</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
type="urn:epcglobal:cbv:btt:inv">urn:lotec:epcis:bt:inv:12345</bizTransaction>
        </bizTransactionList>
        <extension>
          <sourceList>
            <source type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a17</source>
          </sourceList>
          <destinationList>
            <destination
type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a18</destination>
          </destinationList>
        </extension>
        <lotec:intendedRoute>Lausanne - Zurich - Zagreb</lotec:intendedRoute>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T03:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	62d018f0-24c7-4ee4-80c3-6453b64b7b23
<epcList>	List of shipped eps	List of urn	
<epc>	Item	urn	<epc>urn:epc:id:ext:TSA/AHMEEXRM02310</epc>
<readPoint>	Location where event is captured	urn	<id>urn:lotec:id:loc:WAA1a17</id>
<source>	Source location of shipping	urn	<source type="urn:epcglobal:cbv:sdt:location">urn:lotec:id:loc:WAA1a17</source>

Field	Description	Type	Example
<destination>	Destination location of shipping	urn	<destination type="urn:epcglobal:cbv:sdt:location"> urn:lote:id:loc:WAA1a18 </destination>
<bizTransaction>	Invoice ID (optional)	urn	urn:lote:epcis:bt:inv:12345
<lote:intendedRoute>	Intended route	String	Lausanne – Zurich - Zagreb

10.7 DECOMMISSIONING

Decommissioning event is an event by which an object that exists in the system is destroyed. When the action is executed on a product, this event means deletion of the object from the system. When the action is performed on an ID that indicates logistics packages and other packages of higher levels in the hierarchy, this event indicates the deletion of the parent-child relationship and any objects that are children of the destroyed object are also destroyed.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T04:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:72d018f0-24c7-4ee4-80c3-6453b64b7b23</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:epc:id:ext:TSA/AHMEEXRM02310</epc>
        </epcList>
        <action>DELETE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:destroying</bizStep>
        <disposition>urn:epcglobal:cbv:disp:destroyed</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1a17</id>
        </readPoint>
        <bizLocation>
          <id>urn:lote:id:loc:WAA1a17</id>
        </bizLocation>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T04:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	72d018f0-24c7-4ee4-80c3-6453b64b7b23
<epc>	Item	urn	<epc>urn:epc:id:ext:TSA/AHMEEXRM02310</epc>
<disposition>	Status of serial numbers after event is processed	urn	urn:epcglobal:cbv:disp:destroyed - destroy urn:epcglobal:cbv:disp:stolen - stolen urn:epcglobal:cbv:disp:expired - lost urn:epcglobal:cbv:disp:damaged - wasted
<readPoint>	Location where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<bizLocation>	Location where event happened	urn	<id>urn:lote:id:loc:WAA1a17</id>

10.8 RELEASE FOR CONSUMPTION (SALE)

Release for consumption event is an event for sending products from one location to another location which is not registered in the system. With this event, products and their logistics packages end their presence in the supply chain and must not reappear in the system unless returned from the customer. It is similar to shipping event but with a bizstep void_shipping.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:lote="urn:sicpa:lote:xsd:1"
schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T02:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:52d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lote:id:agg:WAA112NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:retail_selling</bizStep>
        <disposition>urn:epcglobal:cbv:disp:retail_sold</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1a17</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
type="urn:epcglobal:cbv:bt:inv">urn:lote:epcis:bt:inv:12345</bizTransaction>
        </bizTransactionList>
        <lote:customer>Some customer name</lote:customer>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T02:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	52d018f0-24c7-4ee4-80c3-6453b64b7b23
<epcList>	List of shipped eps	List of urn	
<epc>	Item	urn	<epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
<readPoint>	Location where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<bizTransaction>	Invoice ID (optional)	urn	urn:lote:epcis:bt:inv:12345
<lote:customer>	Customer	String	Some customer name

10.9 RETURN FROM CUSTOMER

Return from customer event is an event for sending products back from customers unknown location to location in system. With this event, products and their logistics packages are returned to the supply chain. Only serial numbers with status “sold” can be returned from customer. Event is similar to receiving event, with bizstep collecting.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:lote="urn:sicpa:lote:xsd:1"
schemaVersion="1.2" creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T07:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:82d018f0-24c7-4ee4-80c3-6453b64b7c24</eventID>
        </baseExtension>
        <epcList>
          <epc>urn:lote:id:agg:WAA112NM22PK9D90ERA783WP2</epc>
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
        <disposition>urn:epcglobal:cbv:disp:returned</disposition>
        <readPoint>
          <id>urn:lote:id:loc:WAA1a17</id>
        </readPoint>
        <bizTransactionList>
          <bizTransaction
type="urn:epcglobal:cbv:bt:inv">urn:lote:epcis:bt:inv:12345</bizTransaction>
        </bizTransactionList>
        <lote:customer>Some customer name</lote:customer>
      </ObjectEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<eventTime>	Time when event happened	Time Stamp	2024-05-24T02:00:00.000+02:00
<eventTimeZoneOffset>	Time Zone Offset to UTC	Integer	+02:00
<eventID>	Unique ID of event generated by source system – reference in source system	UUID	52d018f0-24c7-4ee4-80c3-6453b64b7b23
<epcList>	List of shipped epcs	List of urn	
<epc>	Item	urn	<epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
<readPoint>	Location where event is captured	urn	<id>urn:lote:id:loc:WAA1a17</id>
<bizTransaction>	Invoice ID (optional)	urn	urn:lote:epcis:bt:inv:12345
<lote:customer>	Customer	String	Some customer name

10.10 ERROR DECLARATION

In case some event contains errors or was sent by mistake, it can be declared as error and excluded from supply chain in a way that it will be displayed but not influencing regular course of events.

Any event can be declared as error by adding errorDeclaration to baseExtension.

When the captured event is declared as error, it must contain the exact same data in EPCIS document. The only difference is errorDeclaration section.

```
<?xml version="1.0"?>
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1" schemaVersion="1.2"
creationDate="2024-05-26T18:29:46.076Z">
  <EPCISBody>
    <EventList>
      <ObjectEvent>
        <eventTime>2024-05-24T00:00:00.000+02:00</eventTime>
        <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
        <baseExtension>
          <eventID>urn:uuid:32d018f0-24c7-4ee4-80c3-6453b64b7b23</eventID>
          <errorDeclaration>
            <declarationTime>2026-06-24T20:36:41.344Z</declarationTime>
            <reason>urn:epcglobal:cbv:er:incorrect_data</reason>
            <correctiveEventIDs>
              <correctiveEventID>urn:uuid:f8bf14a7-37d1-48ff-8c7f-
7dfec994b1b7</correctiveEventID>
            </correctiveEventIDs>
          </errorDeclaration>
        </baseExtension>
      <epcList>
        <epc>urn:lote:id:agg:WAA1a12NM22PK9D90ERA783WP2</epc>
      </epcList>
      <action>ADD</action>
      <bizStep>urn:epcglobal:cbv:bizstep:commissioning</bizStep>
      <disposition>urn:epcglobal:cbv:disp:active</disposition>
      <readPoint>
        <id>urn:lote:id:loc:WAA1aFG5R</id>
      </readPoint>
      <bizLocation>
        <id>urn:lote:id:loc:WAA1a17</id>
      </bizLocation>
    </ObjectEvent>
  </EventList>
</EPCISBody>
</epcis:EPCISDocument>
```

Field	Description	Type	Example
<errorDeclaration>	Error declaration		
<declarationTime>	Time when event was declared error	timestamp	2026-06-24T20:36:41.344Z
<reason>	Reason for declaring event error	urn	urn:epcglobal:cbv:er:incorrect_data urn:epcglobal:cbv:er:did not occur
<correctiveEventIDs>	List of corrective event UUIDs. Optional	List of UUIDs	
<correctiveEventID>	Corrective event UUID	UUID	urn:uuid:f8bf14a7-37d1-48ff-8c7f-7dfec994b1b7

11 CONNECTION FAILURE HANDLING AND RETRY STRATEGY

To promote robust and reliable integrations with the UK VDS LOTE Events API, manufacturers must adopt a resilient approach to event submission, minimizing data loss and preventing duplicate entries during network disruptions or API outages. The following upgraded strategy details best practices for buffering, retry logic, and duplicate prevention.

Recommended Procedure:

Buffering Unsent Events

- Temporary Local Buffering:

Any event submission to the /captureEvent endpoint that fails (i.e., due to connection errors, timeouts, or no response) must be placed in a secure, persistent local buffer. This may take the form of a file-based queue, in-memory store with durable backup, or lightweight local database.

- Comprehensive Metadata:

Each record in the buffer should include:

- o Event Payload: The original event data as sent.
- o Timestamp: Time when the event was first attempted.
- o Event Type: To facilitate categorization and debugging.
- o Retry Count: Number of retry attempts for that event.
- o Unique Identifier (UUID): To support accurate deduplication and status checks.

Retry Logic

- Automated Retry Initiation:

Initiate retry attempts once network connectivity or the LOTE Events API service is restored.

- Exponential Backoff & Maximum Limit:

Apply exponential backoff to incrementally increase wait times between retries—e.g., 1s, 2s, 4s, 8s—capping at a configurable maximum (e.g., 5 attempts). This reduces repeated load spikes during major outages.

- Consolidated Retry Logging:

Log all retry attempts with event metadata, reasons for retrying, success/failure status, and server responses. Ensure these logs are accessible for audit and troubleshooting.



SICPA SA
World Headquarters
Av de Florissant 41
1008 Prilly
Switzerland

Tel +41 21 627 55 55
Fax +41 21 627 57 27
info@sicpa.com
www.sicpa.com