dentsu

TRACKING

Dentsu Aegis Network

DATA DICTIONARY 1.4.2 OPTIONAL TECHNICAL FEATURES FOR PRIMARY PROVIDERS

This document describes the optional functionalities to help the primary repositories meet the technical modalities of event reporting.

Summary of changes

Date	Version	Done by	Comment
20.04.2020	0.1	Dentsu Aegis Network	Internal Draft
27.05.2020	0.2	Dentsu Aegis Network	Update of the validation
17.06.2020	0.3		Removal of the envelop message proposal
06.07.2020	0.4	Dentsu Aegis Network	Updated MTL validation scope. Maintenance Service scope.
10.07.2020	0.5	Dentsu Aegis Network	Update http status to 202 Remove the service maintenance.
20.07.2020	0.6	Dentsu Aegis Network	System Reception Timestamp.
25.07.2020	1.0	Dentsu Aegis Network	
13.08.2020	1.1	Dentsu Aegis Network	Document Split Clerical error regarding the Checksum Data_List and definition of the Date format

Distribution

Date	Version	Submitted to
19.06.2020	0.3	Primary Providers
09.07.2020	0.4	Primary Providers
10.07.2020	0.5	Primary Providers
20.07.2020	0.6	Primary Providers
25.07.2020	1.0	Primary Providers
13.08.2020	1.1	Publication

Confidentiality Statement
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 Aegis Network.

Disclaimer
The information contained in this document is information and should not be used as a formal specification nor as implementation requirements.

Table of Contents

1 Introduction

This document defines the optional functionalities to help the primary repositories meet the technical modalities of event reporting.

2 Optional Message Time Sequence Validation

2.1 Overview

The out of sequence of an aggregation messages (EPA 3.2) can lead to the inaccurate display of the content of an aggregate UI.
The repacking activity can be caracterised in to two scenarii

- Repacking of a UI
- Reuse of an aUI

In order ensure proper sequencing of the aggregation, an additional validaton on the messages timestamp will be added.

2.2 System Reception Timestamp

In some cases, the manufacturer systems can generate burst of messages. A number of messages can be produced during the same second and therefor will have the same EventTime and the same MessageTimeLong.

In order to implement efficiently the sequence validation controls, the System Reception_Time at a millisecond precision will be added.
The Reception_Time will be recorded and added by the Primary repository as an Optional message.

2.2.1 Basic information block concerning the request

Basic information block concerning the request - schema					
Field	**Description**	**Data Type**	**Cardinality**	**Priority**	**Values**
Message_Type	The identifier of the type of message	Text	S	M	See above types of messages list
Code	The internal code of acknowledgment of the message. Used for recall too.	Text	S	M	property is nullable
RejectionData	The failure data recorded in the primary should the validation fail.	FailureData (See below table)	S	O	This should only be filled if the primary validation fails.
Reception_Time	System reception Time	Time(ms)	S	O	

Time(ms) format
FORMAT : yyyy-MM-ddTHH:mm:ss.fffZ
SAMPLE : 2020-08-13T16:01:34.477Z

2.3 System Reception Timestamp transmission

2.3.1 Primary to secondary

The primary repository adds the Reception_Time in the message transmissted to the secondary repository.

2.3.2 Router to Primary

The router will not forward the Reception_Time to the primary as this timestamp is optional.

2.4 Message Time Long vs Reception Timestamp

The Reception_Time is a millisecond precision timestamp allowing for an efficient sequence validation. The Reception_Time is optional and could in some cases not be present in the message. As fallback, the Secondary will rely on the Message_Time_Long.

2.5 Repacking scenarii

2.5.1 Repacking of a UI

- M1 EPA (Aggregation) Pack 1 to Bundle 1 (EventTime=T1, MT=T1)
- M2 EPA (Aggregation) Pack 1 to Bundle 2 corresponds to a repacking process (EventTime=T2, MT=T2)

The transmission of the messages M1 and M2 out of sequence will currently lead to an inaccurate update of the content of the Bundle 1 and 2.

MSG Sequence	**MSG Type**	**UIDs**	**Comments**
1	EPA , MT=T2	Pack 1 to Bundle 2	
2	EPA, MT=T1	Pack 1 to Bundle 1	

With the validation of the MessageTimeLong, the message M1 will be rejected and the information of the Bundle 1 will correspond to the Pack 1

MSG Sequence	**MSG Type**	**UIDs**	**Primary ACK**	**Secondary ACK**	**Comments**
1	EPA , MT=T2	Pack 1 to Bundle 2	202	202	
2	EPA, MT=T1	Pack 1 to Bundle 1	400		

2.5.2 Reuse of an aUI

- M1 EPA (Aggregation 3.2) Pack 1 to Bundle 1 (EventTime=T1, MT=T1)
- M2 EUD (Disaggregation 3.6) Bundle 1 (EventTime=T2, MT=T2)
- M3 EPA (Aggregation 3.2) Pack 2 to Bundle 1 corresponds to a repacking process (EventTime=T3, MT=T3)

The transmission of the messages M1 and M2 out of sequence will currently lead to an inaccurate update of the content of the Bundle 1.

MSG Sequence	**MSG Type**	**UIDs**	**Comments**

1	EPA , MT=T3	Pack 2 to Bundle 1	
2	EPA, MT=T1	Pack 1 to Bundle 1	
3	EUD, MT=T2	Bundle 1	

The Bundle 1 will contain the information of Pack 1 and not the updated Pack 2.

In order to prevent out of sequence aggregation on the same UI to be reported, the validation on the Message Time Long will be added.

With the validation of the MessageTimeLong, the message M1 an M2 will be rejected and the information of the Bundle 1 will correspond to the Pack 2

MSG Sequence	**MSG Type**	**UIDs**	**Primary ACK**	**Secondary ACK**	**Comments**
1	EPA , MT=T3	Pack 2 to Bundle 1	202	202	
2	EPA, MT=T1	Pack 1 to Bundle 1	400		
3	EUD MT=T2	Bundle 1	400		

2.5.3 VAL_UI_MT on the Aggregation and Disaggregation

2.5.4 Aggregation parent UI

Upon the reception of the aggregation event (EPA 3.2). The repository will check for the **parent aUI** if there is any other aggregation events (EPA 3.2) and check if the current event is the lastest for the parent aUI.
Step 1: list all aggregation events (EPA 3.2) for the parent aUI.
Step 2: check if the aggregation event (EPA 3.2) is the latest aggregation event based on the Reception Time or Message Time Long (MTL).
The control passes if no aggregation event (EPA 3.2) for the parent aUI was previsouly reported with an MTL that is posterior was found.

2.5.5 Aggregation Child UI

Upon the reception of the aggregation event (EPA 3.2). The repository will check for the **Child UI** if there is any other aggregation events (EPA 3.2) and check if the current event is the lastest for that UI.

2.6 Error Code

The VAL_UI_MT return a OUT_OF_SEQUENCE

	Error Code	**http Status**
Sequence validation		
VAL_UI_MT	OUT_OF_SEQUENCE	400

2.6.1 Validation responsibility

	IRU (2.1)	IRA (2.2)	IDA (2.3)	EUA (3.1)	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)
Sequence validation														
VAL_UI_MT					X				X					

	Primary repository	**Router**	**Secondary repository**
Technical validation			
VAL_UI_MT	O	X	X

X. Mandatory
O. optional

2.7 Message Time validation implementation

2.7.1 Primary repository and Router

The control is implemented on the Message Time Long.

2.7.2 Secondary repository

The secondary repository will perform the validation using the Reception Time if available. As a fallback the control will be implemented on the Message Time Long.

2.8 Special processing of technical historical data.

2.8.1 Context

In some exceptional cases, if the primary fails to report some events, the secondary validation will prevent the primary to report these events aposteriori.

The following edge cases have been identified.

- MAINT_01: Repacking scenario
- MAINT_02: Historical Transloading scenario
- MAINT_03: Arrival before Deactivation

In order to allow the secondary to be a copy of the primary repositories, the secondary repository will perform a limited update of the meta data.
Only the event list will be updated
The metadata used for the validation (state and location) will not be updated in these specific cases.

The secondary will add specific audit trail during the processing of these specific cases.
The secondary will add specific information to the event allowing the competent authority to be informed of this specific processing.

Note that any other cases not exactly matching these scenarii will be rejected. (http status 400)

The primary repository will receive a successful answer (http status 202) and will not have to resend the message.

2.8.2 MAINT_01: Repacking scenario

The case of out of sequence of repacking process leading the original aggregation message and the subsequent disaggregation messages of being accepted.

Message supported:

- aggregation message (EPA),
- disaggregation message (EUD).

Controls

- Messages must have been rejected by the regular endpoint.
- Control on the existance of the aUI
- Control on the Reception Time or Message Time Long

2.8.3 MAINT_02: Historical Transloading scenario

ETL message

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EDP	Any	202	202	
2	ETL	Same as in EDP	202	TIMEOUT	Message is not transmitted to Secondary due to technical reasons
3	ERP	Same as in EDP	202	202	

Message supported:

- transloading message (ETL 3.5),

Controls.

- Control on the MessageTimeLong that ETL was sent after an EDP (3.3)
- Control on the MessageTimeLong that ETL was sent before an ERP (3.4)

2.8.4 MAINT_03: Arrival before Deactivation

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EDP (3.3)	Any	202	202	

2	ERP (3.4)		202	TIMEOUT	Message is not transmitted to Secondary due to technical reasons
3	IDA (2.3)		202	202	

Message supported:

- Arrival message (ERP 3.4),

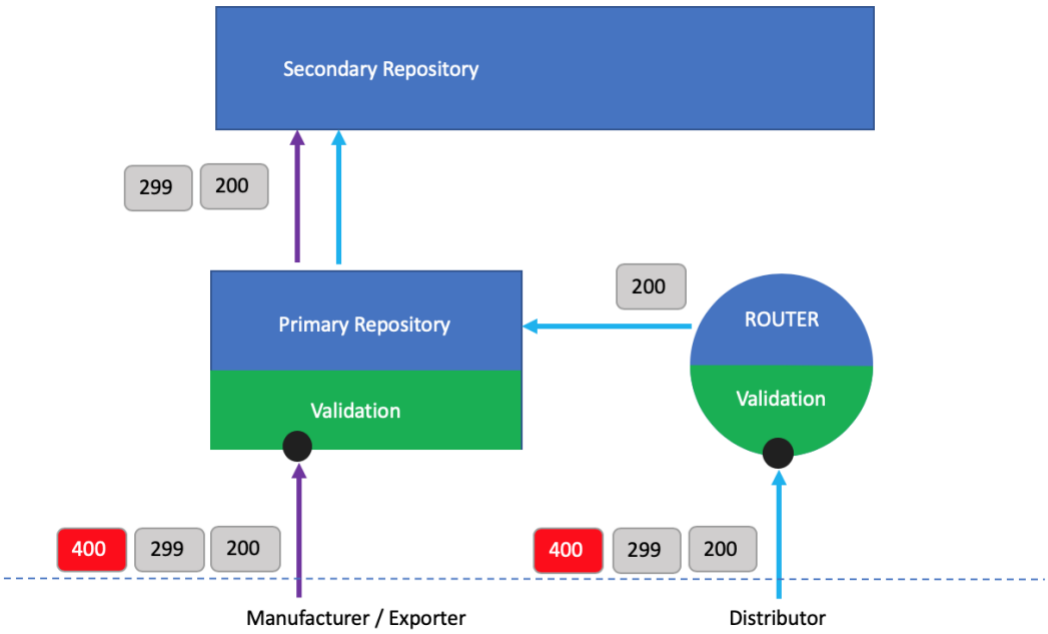
Controls.

- Control

3 Hard Validation - From Warning 299 to Errors 400 on the Secondary interface

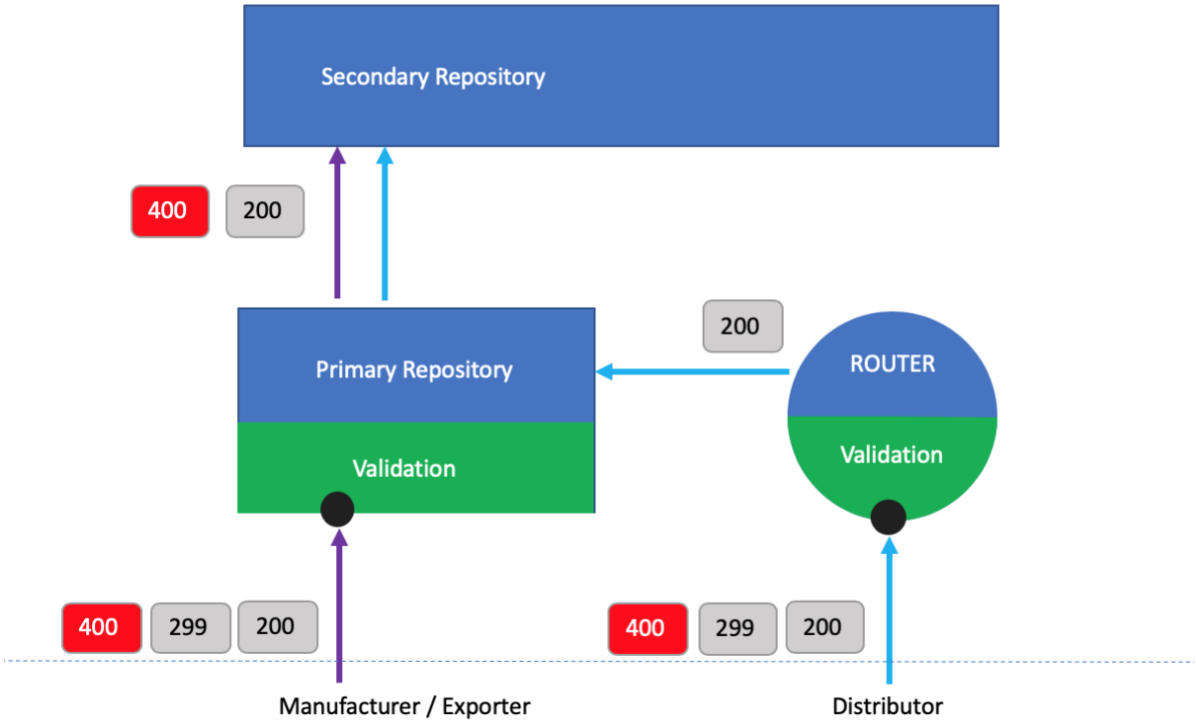
3.1 Context

As of the current protocol, the interface between primary repository and secondary repository accepts out of sequence messages or validation failures. In these cases, the secondary repository returns a 299 Warning to the primary. Upon processing on the warning messages, UI level Metadata is updated at the EU Secondary level. The Meta data is later used to perform the validation



3.2 Proposal on the protocol update between Primary and Secondary repostory

In case the EU Secondary faces a sequence control error, the proposal consists in changing the WARNING (HTTP Code 299) to an ERROR (HTTP Code 400).
The Error response contains the details of the validation error.
The original message is stored on the audit repository.



3.2.1 Handling of validation error 400 on the Secondary.
The Meta data associated with the UI is not updated upon reception of rejected messages (error 400).

3.2.2 Rejected message retransmission
If a message is rejected by the secondary repository, the primary repository can retransmit the same message (same payload and same recallcode) as an attempt to get it accepted.

If there are no new messages between the initial attempt and the retry, the outcome will be an error 400, so the recommended approach is to retry the message after some other messages have been sent and accepted in between which will update the metadata of the UIs of the rejected message to be in a position where the retried message can be accepted.

3.2.3 Validation and Error Code – Primary to Secondary Repository

	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	400

	EXCISE_NUMBER_NOT_VALID NON_COMPATIBLE_UIS NOT_THE_SAME_NUMBER_OF_ITEMS	
VAL_MSG_XML	FAILED_VALIDATION	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_MSG_CODE_DUPLICATE	FAILED_VALIDATION	400
VAL_UI_MULT_MSG	MULTIPLE_UID 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_VALID	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_EXIST_MID	MID_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
VAL_ENT_ACTIVE_MID	MID_NOT_EXIST_OR_ACTIVE	400
VAL_ENT_REL_EOID_FID	FID_NOT_RELATED_TO_EOID	400
VAL_ENT_REL_ FID_MID	MID_NOT_RELATED_TO_FID	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
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
Message Timing		
VAL_EVT_24H	OPERATION_WITHIN_24_HOURS	202
VAL_EVT_TIME	SHIPMENT_WITHIN_24_HOURS	202
Recall		
VAL_RECALL_EXIST	CODE_NOT_EXIST CODE_NOT_UNIQUE	400

VAL_RECALL_LAST	RECALL_NOT_LAST_EVENT	400

3.3 Validation Responsibility

	Primary repository	Router	Secondary repository
Technical validation			
VAL_SEC_HASH	400	400	400
VAL_SEC_TOKEN	401	401	401
VAL_MSG_JSON	400	400	400
VAL_MSG_XML	400	400	400
VAL_MSG_TYPE	400	400	400
VAL_FIE_MAN	400	400	400
VAL_FIE_FORMAT	400	400	400
VAL_FIE_REF	400	400	400
VAL_MSG_DUPLICATE	400	400	400
VAL_MSG_CODE_DUPLICATE	400		400
Business rule validation			
VAL_UI_MULT_MSG	400	400	400
VAL_UI_EXIST_APP	400		400
VAL_UI_DUPLICATE_APP	400		400
VAL_UI_FID_APP	400		400
VAL_UI_EXIST_UPUI	400	400	400
VAL_UI_EXIST_AUI	400	400	400
VAL_UI_EXIST_UPUI_SEQ	400	400	400
VAL_UI_EXIST_AUI_SEQ	400	400	400
VAL_UI_EXPIRY	400		400
VAL_UI_ORD_REACTIVATION	400	400	400
VAL_UI_ORD_DEACTIVATED	400	400	400
VAL_UI_ORD_AGG_MULT	400	400	400
VAL_UI_ORD_DISAGG	400	400	400
VAL_UI_ORD_IMPLDISAGG	400	400	400
VAL_UI_ORD_AGG_FID	400	400	400
VAL_UI_ORD_ARRIVAL	400	400	400
VAL_UI_ORD_ARRIVAL_RETURN	400	400	400
VAL_UI_ORD_DISPATCH	400	400	400
VAL_EVT_24H	299	299	
VAL_EVT_TIME	299	299	
VAL_ENT_EXIST_EOID		400	400
VAL_ENT_EXIST_FID		400	400
VAL_ENT_EXIST_MID		400	400
VAL_ENT_ACTIVE_EOID		400	400
VAL_ENT_ACTIVE_FID		400	400
VAL_ENT_ACTIVE_MID		400	400
VAL_ENT_REL_EOID_FID		400	400
VAL_ENT_REL_ FID_MID		400	400
VAL_RECALL_EXIST	400	400	400
VAL_RECALL_LAST	400	400	400

3.4 Impact

3.4.1 On primary repositories

The hard validation enforcement is a simplification of the communication meant to assist the primary repository to achieve compliance with the technical specifications (v. 1.4.2).
The primary repositories must implement the management of validation error 400 messages.
The current (version 1.4.2) requires the primary provider to already review and analyse the warning messages (http status 299)

3.4.2 On the secondary repository

The support of the 400 and therefore retransmission of messages will increase the processing and message storage.

3.5 Example Use Cases

Multiple aggregation and disaggregation
Recall on primary side

3.5.1 Use case 1:

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EDP	Any	202	202	
2	ETL	Same as in EDP	202	TIMEOUT	Message is not transmitted to Secondary due to technical reasons
3	ERP	Same as in EDP	202	202	

Resolution

The Transoaling message is VALID and cannot be transmitted to the Secondary Repository because the original transmission timed out and there was an ERP positively ackownledge afterwards.

The ETL message will trigger a sequence validation error.

To solve this situation, the ETL message has to be transmitted to the Mainteinance Endpoint so it is reflected in the history of the UI.

The metadata associated with the UI is the same regardless if the ETL has been received or not and no further action is needed.

3.5.2 Use case 2
Manufacturer to Primary

MSG Sequence	MSG Type	UIDs	Primary ACK		
0	EUA	Pack activation	202		
1	EPA	Pack to Bundle	202		
2	EPA	Bundle to Case	202		
3	EPA	Case to Pallet	202		

Primary to Secondary scenarios:

Scenario A)

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EUA	Pack activation	202	202	
2	EPA	Pack to Bundle	202	202	
3	EPA	Case to Pallet	202	400	Rejected as Cases do not exist
4	EPA	Bundle to Case	202	202	
5	EPA	Case to Pallet	-	202	Now the message is accepted because Cases exist

Scenario B)

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments

1	EPA	Pack to bundle	202	400	Packs are not activated
2	EPA	Case to Pallet	202	400	Rejected as Cases do not exist
3	EPA	Bundle to Case	202	400	Bundles do not exist
4	EUA	Pack activation	202	202	
5	EPA	Case to Pallet	-	400	Case do not exist
6	EPA	Pack to bundle	-	202	
7	EPA	Bundle to Case	-	202	
8	EPA	Case to Pallet	-	202	

Resolution

For a given hierarchy creation, as long as the Manufacturer has repoted the complete hierarchy (EUA + Multiple EPAs) to the Primary Repository, with the 400s returned by the Secondary Repository and a retry mechanism the hierarchy will be successfully created eventually.

3.5.3 Use Case 3

MSG Sequence	**MSG Type**	**UIDs**	**Primary ACK**		
1	EUA	Pack activation	202		
2	EPA	Pack to Bundle	202		
3	EPA_a	Bundle to Case	202		
4	RCL	Recall Bundle to Case	202		
5	EPA	Case to Pallet	400		
6	EPA_b	Bundle to Case	202		
7	EPA	Case to Pallet	202		

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EUA	Pack activation	202	202	
2	EPA	Pack to Bundle	202	202	
3	EPA_a	Bundle to Case (UID_A)	202	TIMEOUT	
4	RCL	Recall Bundle to Case (EPA_a)	202	400	Recall Code does not exist
5	EPA	Case to Pallet	400	**202**	Message has to be reported as REJECTED to the Secondary
6	EPA_b	Bundle to Case (UID_B)	202	202	UI history inconsistent
7	RCL	Recall Bundle to Case	-	400	Recall code does not exist
8	EPA_a	Bundle to Case (UID_B)	-	202	Implicit disagregation UID_A - INCONSISTENCY
9	RCL	Recall Bundle to Case (EPA_a)	-	202	Issue resolved because EPA_a is rolled back. **High risk**.

Resolution:

Even though the end result can be similar to the Primary metadata and hierarchy status, allowing these inconsistencies when they can be prevented beforehad is a risk the protocol should avoid taking

This why for the Hierarchy use cases the message time long will be taken into account to not update the metadata of hierarchical relationship between UIs.

- In the case of EPAs, when the Message Time Long of the latest EPA received for a given CHILDREN UI is OLDER than the latest known

Message Time Long for the latest EPA for that UI, the message will be recorded but will not update the metadata.

Therefore the proposed implementation for these use case taking the MTL (Message Time Long) into account would be:

MSG Sequence	MTL	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	T0	EUA	Pack activation	202	202	
2	T1	EPA	Pack to Bundle	202	202	
3	T2	EPA_a	Bundle to Case (UID_A)	202	TIMEOUT	
4	T3	RCL	Recall Bundle to Case (EPA_a)	202	400	Recall Code does not exist
5	T4	EPA	Case to Pallet	400	**202**	Message has to be reported as REJECTED to the Secondary
6	T5	EPA_b	Bundle to Case (UID_B)	202	202	UI history inconsistent
7	T3	RCL	Recall Bundle to Case	-	400	Recall code does not exist
8	T2	EPA_a	Bundle to Case (UID_B)	-	**202**	For the Bundles, the latest known EPA is “T5” > T2, therefore message is

						recorded but no hierarchy is updated
9		RCL	Recall Bundle to Case (EPA_a)	-	202	In the end message is recalled

3.5.4 Use case 3

Scenario : The multiple level aggregation received successfully on the Primary repository.

- M1 : EPA 3.2 Pack to bundle MT=T1
- M2 : EPA 3.2 bundle to MC MT=T2
- M3 : EPA 3.2 MC to pallet MT=T3.

messages are not sent in the right order by the primary
Messages not in the right order are not considered.

- M2 : EPA 3.2 bundle to MC MT=T2
- M1 : EPA 3.2 Pack to bundle MT=T1
- M3 : EPA 3.2 MC to pallet MT=T3.

MSG Sequence	**MSG Type**	**UIDs**	**Secondary ACK**	**Comments**
1	EPA MT=T2	bundle to MC	400	The UI doesn't exist
2	EPA MT=T1	Pack to Bundle	202	
3	EPA MT=T3	MC to pallet	400	
4	EPA MT=T2	bundle to MC	202	
5	EPA MT=T3	MC to pallet	202	

3.5.5 Use case 4

Scenario : wrong aggregation message from manufacturer

- M1 : EPA 3.2 unit U1 to bundle B1, MT=T2
- M2 : EPA 3.2 bundle B1 to MC MC1, MT=T3
- M3 : EPA 3.2 MC MC1 to pallet P1, MT=T4.
- M4 : EPA 3.2 unit U2 to bundle B1, MT=T1

The M4 should be rejected by the Primary repository.

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EPA MT=T2	Pack to Bundle	202	202	
2	EPA MT=T3	Bundle to Case	202	202	
3	EPA MT=T4	Case to Pallet	202	202	
4	EPA MT=T1	Pack to Bundle	400		The Primary will rely and the MTL validation to reject the message M4

Resolution:
Since the root cause is the miss reporting of the aggregation messages by the manufacturer. The manufacturer should use the RECALL mechanim to correct the data.

3.5.6 Use case 5

M1 : M3.2 Pack U1 to bundle B1, MT=T1
M2 : M3.2 bundle B1 to MC MC1, MT=T3
M3 : M3.2 MC MC1 to pallet P1, MT=T4.
M4 : M3.3 Pallet P1, MT=T5.
M5 : M3.3 unit MC MC1 , MT=T2 ,

MSG Sequence	MSG Type	UIDs	Primary ACK	Secondary ACK	Comments
1	EPA, MT=T1	Pack to Bundle	202	202	
2	EPA, MT=T3	bundle B1 to MC MC1	202	202	
3	EPA, MT=T4	MC MC1 to pallet P1	202	202	
4	EDP, MT=T5	Pallet P1	202	202	
5	EDP, MT=T2	unit MC MC1	400		

3.6 Repacking

The repacking process is a common business practice. This activity can be perfomed at the manufacturer side as well as at the distribution side.

In order to prevent out of sequence aggregation on the same UI to be reported, the validation on the Message Time Long or Reception Time is added.

- M1 EPA (Aggregation) Pack 1 to Bundle 1 (EventTime=T1, MT=T1)
- M2 EUD (Disaggregation) Bundle 1 (EventTime=T2, MT=T2)
- M3 EPA (Aggregation) Pack 2 to Bundle 1 corresponds to a repacking process (EventTime=T3, MT=T3)

With the validation of the MessageTimeLong, the message M1 will be rejected and the information of the Bundle 1 will correspond to the Pack 2

MSG Sequence	**MSG Type**	**UIDs**	**Primary ACK**	**Secondary ACK**	**Comments**
1	EUD , MT=T2	Bundle 1	400		
2	EPA , MT=T3	Pack 2 to Bundle 1	202	202	
3	EPA, MT=T1	Pack 1 to Bundle 1	400		

4 Removal of the out of Sequence support on the Secondary repository

4.1 Context

Some primary providers raised the concern that they can not ensure the proper sequence of messages within their repositories in the established timeframe. In order to protect the functioning of the Secondary Repository and the Router and to accommodate the shortcoming of some primary providers and in agreement with the EU Commission we have decided to implement the "out-of-sequence" correction mechanism for the messages related to the production events (messages 3.1 and 3.2).

The "out-of-sequence" correction mechanism is an exception mechanism that is implemented on a temporary basis.

For the sake of clarity, the out of sequence message due to product movement (3.3, 3.4, 3.5, 3.6 and 3.7) are not in the scope of "out-of-sequence" correction mechanism. The failure to report these events in sequence will cause validation errors on the router.

4.2 Action

This feature will be removed

M1 : M3.2 unit U1 to bundle B1, MTL=T1
M3 : M3.2 MC MC1 to pallet P1, MTL=T3
M2 : M3.2 bundle B1 to MC MC1, MTL=T2

MSG Sequence	**MSG Type**	**UIDs**	**Secondary ACK**	**Comments**
1	EPA MTL=T1	Pack to Bundle	202	
2	EPA MTL=T3	Case to Pallet	400	
3	EPA MTL=T2	Bundle to Case	202	