MEF

1

2

3 **Working Draft**

4 **MEF W125.0.1**

5

6 **Amendment to MEF 125: LSO Cantata and LSO**
7 **Sonata - Subscriber Ethernet**

8

9 **January 2023**

10

11 **This draft represents MEF work in progress and**
12 **is subject to change.**

13 EXPORT CONTROL: This document contains technical data. The download, export, re-export or
14 disclosure of the technical data contained in this document may be restricted by applicable U.S. or
15 foreign export laws, regulations and rules and/or applicable U.S. or foreign sanctions ("Export
16 Control Laws or Sanctions"). You agree that you are solely responsible for determining whether
17 any Export Control Laws or Sanctions may apply to your download, export, reexport or disclosure
18 of this document, and for obtaining (if available) any required U.S. or foreign export or reexport
19 licenses and/or other required authorizations.
20

21 Disclaimer

22 © MEF Forum 2022. All Rights Reserved.

23 The information in this publication is freely available for reproduction and use by any recipient
24 and is believed to be accurate as of its publication date. Such information is subject to change
25 without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume
26 responsibility to update or correct any information in this publication. No representation or war-
27 ranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applica-
28 bility of any information contained herein and no liability of any kind shall be assumed by MEF
29 as a result of reliance upon such information.

30 The information contained herein is intended to be used without modification by the recipient or
31 user of this document. MEF is not responsible or liable for any modifications to this document
32 made by any other party.

33 The receipt or any use of this document or its contents does not in any way create, by implication
34 or otherwise:

35   a) any express or implied license or right to or under any patent, copyright, trademark or
36   trade secret rights held or claimed by any MEF member which are or may be associated
37   with the ideas, techniques, concepts or expressions contained herein; nor
38   b) any warranty or representation that any MEF members will announce any product(s)
39   and/or service(s) related thereto, or if such announcements are made, that such an-
40   nounced product(s) and/or service(s) embody any or all of the ideas, technologies, or
41   concepts contained herein; nor
42   c) any form of relationship between any MEF member and the recipient or user of this
43   document.

44 Implementation or use of specific MEF standards, specifications, or recommendations will be vol-
45 untary, and no Member shall be obliged to implement them by virtue of participation in MEF
46 Forum. MEF is a non-profit international organization to enable the development and worldwide
47 adoption of agile, assured and orchestrated network services. MEF does not, expressly or other-
48 wise, endorse or promote any specific products or services.

49

Table of Contents

List of Figures

1 List of Contributing Members

The following members of the MEF participated in the development of this document and have requested to be included in this list.

-

2 Abstract

The Amendment to MEF 125 LSO Cantata and LSO Sonata - Subscriber Ethernet Product Schemas and Developer Guide is a supplement of technical standard MEF 125 [3]. The purpose of this amendment is addition the following information:

- presentation of different Subscriber Ethernet configurations

- show the basic differences between Subscriber Ethernet technologies

- discuss common modifications

- provide examples for actions (add, modify, delete)

- deliver basic APIs steps walkthrough to order a Subscriber Ethernet product

175 **3 Introduction**

176 This document delivers only informative New Appendix A and provides Postman collection ex-
177 amples.

4   Changes to section 2 Abstract

Add the following paragraph to the end of section 2:

The Postman file is included in the GitHub repository and contains informative examples illustrating use of the Subscriber Ethernet payloads. This file is not part of this standard but is referred to in Appendix A.

- documentation\productSchema\carrierEthernet\MEF 125.0.1 - Appendix A.postman_collection.json

5   New Appendix A

Insert the content below into the document as Appendix A.

Appendix A   Usage examples (Informative)

This appendix aims to provide an extensive set of examples to cover:
- different configuration variants (Use Cases 2 and 3)

- basic all APIs steps walkthrough to order a Subscriber Ethernet product (Use Cases 1, 2, 4, 5, 6)

- common modifications (Use Cases 7-11, 13)

- deletion of a product (Use Case 12)

The examples are delivered in two forms:
- as part of this document – to allow comments and rich explanation

- as a Postman collection – for ease of use in testing.

The following terms are used in Appendix A:

- EPL -ethernetPrivateLineEvc

- EVPL – ethernetVirtualPrivateLineEvc

- EP-LAN – ethernetPrivateLineEvc

- EP-LAN ENDPOINT - ethernetPrivateLanEvcEp

- EVP-LAN – ethernetVirtualPrivateLanEvc

- EVP-LAN ENDPOINT ethernetVirtualPrivateLanEvcEp

- EP-TREE – ethernetPrivateTreeEvc

- EP-TREE ENDOPINT - ethernetPrivateTreeEvcEp

206   • EVP-TREE – ethernetVirtualPrivateTreeEvc

207   • EVP-TREE ENDOPINT - ethernetVirtualPrivateTreeEvcEp

208   • UNI – carrierEthernetSubscriberUni

A.1   High-Level flow

210 The Cantata and Sonata Interface Reference Points are formed from a set of APIs the serve differ-
211 ent functions in the end-to-end flow. Figure A2-1 shows all of the functions and their sequence.

Figure A2-1 – Cantata and Sonata End-to-End Function Flow

214

215   • Address Validation - allows the Buyer to retrieve address information from the Seller,
216   including exact formats, for addresses known to the Seller.

217   • Site Retrieval - allows the Buyer to retrieve Service Site information including exact
218   formats for Service Sites known to the Seller.

219   • Product Offering Qualification (POQ) - allows the Buyer to check whether the Seller
220   can deliver a product or set of products from among their product offerings at the ge-
221   ographic address or a service site specified by the Buyer; or modify a previously pur-
222   chased product.

223   • Quote - allows the Buyer to submit a request to find out how much the installation of
224   an instance of a Product Offering, an update to an existing Product, or a disconnect of
225   an existing Product will cost.

226   • Product Order - allows the Buyer to request the Seller to initiate and complete the
227   fulfillment process of an installation of a Product Offering, an update to an existing
228   Product, or a disconnect of an existing Product at the address defined by the Buyer.

229 • Product Inventory - allows the Buyer to retrieve the information about existing Product
230 instances from Seller's Product Inventory.

231 • Trouble Ticketing - allows the Buyer to create, retrieve, and update Trouble Tickets
232 as well as receive notifications about Incidents' and Trouble Tickets' updates. This
233 allows managing issues and situations that are not part of normal operations of the
234 Product provided by the Seller.

235 All of the above-mentioned APIs are provided in the SDK together with accompanying Developer
236 Guides. Please refer to these documents for more details and examples of particular functional
237 APIs.

A.2 Integration of product specifications into the APIs.

239 The above-mentioned APIs are product-agnostic in the meaning that they serve as a business in-
240 teraction level between the Buyer and the Seller, and they do not contain any product-specific
241 information in their specifications. In order to pass the product-specific information, an extension
242 pattern must be used. This applies to four APIs that carry product-specific information: POQ,
243 Quote, Product Order, and Product Inventory.
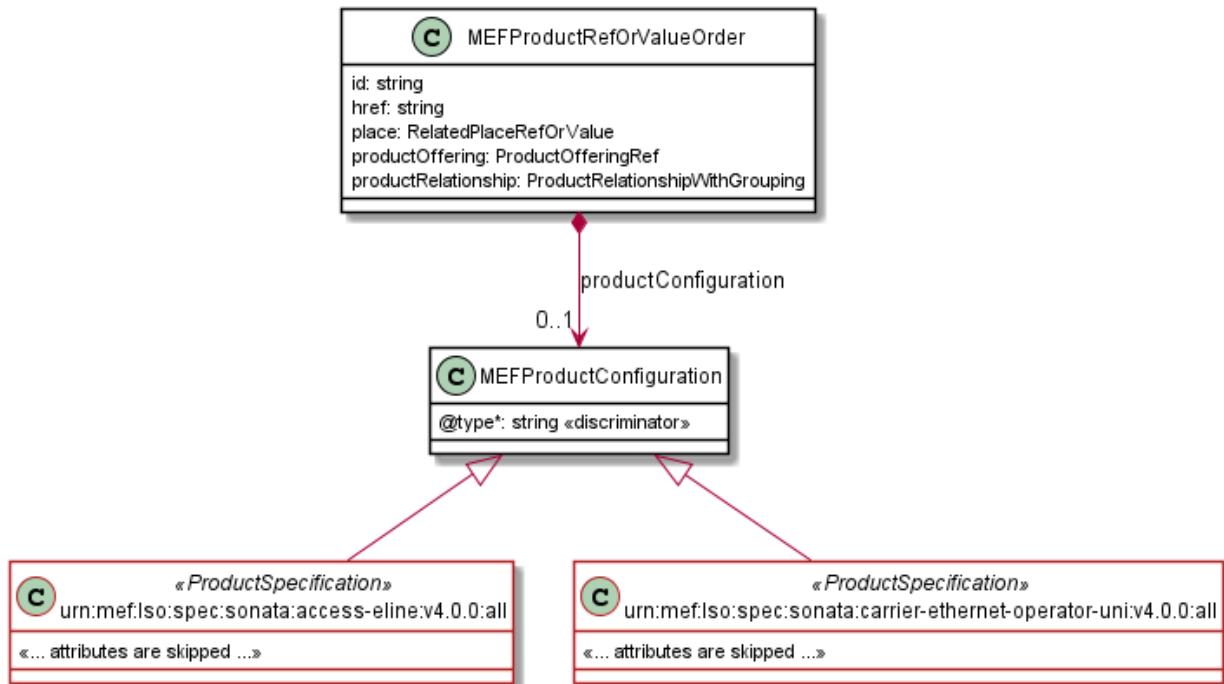
244 The extension hosting type in the API data model is "MEFProductConfiguration". The "@type"
245 attribute of that type must be set to a value that uniquely identifies the product specification (Figure
246 A2-2). A unique identifier for MEF standard product specifications is in URN format and is as-
247 signed by MEF. This identifier is provided as root schema "$id" and in product specification doc-
248 umentation. In this case, this will be one of:

249 • urn:mef:lso:spec:cantata-sonata:epl-evc:v1.0.0:all

250 • urn:mef:lso:spec:cantata-sonata:evpl-evc:v1.0.0:all

251 • urn:mef:lso:spec:cantata-sonata:eplan-evc:v1.0.0:all

252 • urn:mef:lso:spec:cantata-sonata:evplan-evc:v1.0.0:all

253 • urn:mef:lso:spec:cantata-sonata:eptree-evc:v1.0.0:all

254 • urn:mef:lso:spec:cantata-sonata:evptree-evc:v1.0.0:all

255 • urn:mef:lso:spec:cantata-sonata:eplan-evc-endpoint:v1.0.0:all

256 • urn:mef:lso:spec:cantata-sonata:evplan-evc-endpoint:v1.0.0:all

257 • urn:mef:lso:spec:cantata-sonata:eptree-evc-endpoint:v1.0.0:all

258 • urn:mef:lso:spec:cantata-sonata:evptree-evc-endpoint:v1.0.0:all

259 • urn:mef:lso:spec:cantata-sonata:carrier-ethernet-subscriber-uni:v1.0.0:all

260 Use of non-MEF standard product definitions is allowed. In such a case the schema identifier must
261 be agreed upon between the Buyer and the Seller.

262

263

Figure A2-2 – The Extension Pattern

264

265 Product specifications are provided as Json schemas without the "MEFProductConfiguration" con-
266 text. Product-specific attributes are introduced via the "MEFProductRefOrValue" (defined by the
267 Buyer). This entity has the "productConfiguration" attribute of type "MEFProductConfiguration"
268 which is used as an extension point for product-specific attributes. The example result of such
269 binding in a request payload may look like this (for POQ):

```
270  {
271      "instantSyncQualification": true,
272      "externalId": "BuyerPoq-00001",
273      "provideAlternative": false,
274      "projectId": "BuyerProjectX",
275      "productOfferingQualificationItem": [
276        {
277          "id": "item-001",
278          "action": "add",
279          "product": {
280            "productOffering": {
281              "id": "000073"
282            },
283            "productConfiguration": {
284              "@type": "urn:mef:lso:spec:cantata-sonata:epl-evc:v1.0.0:all"
285              "listOfCosNames": ["low"],
286              "availableMegLevel": "6",
287              "carrierEthernetSls": [],
288              "maximumFrameSize": 1522,
289              "evcEndPointA": {},
290              "evcEndPointZ": {}
291  <<the rest of the attributes omitted>>
```

POQ API part

Subscriber Ethernet Product part

```
292              ...
293          }
294        }
295      }
296    ]
297  }
```

A.3   action: add

This section guides through all the steps of Sonata and Cantata APIs that need to be performed in order to successfully order a Subscriber Ethernet product.

Note: Sellers are free to mandate some of these steps.

Note: As the examples of particular steps in many cases will replicate the product-specific infor-mation, in some of the snippets some parts of it will be omitted for better readability.

There are common rules for all request items for creation requests (POQ, Quote, Order):

- "item.action" must be set to "add"

- "item.product.id" must not be provided

- "product.productConfiguration" must contain all desired configurations.

A.3.1   Use Case 1: Address Validation

For detailed guidance on how to use the Address Validation API, please refer to MEF 121 [6].

The first step of the process is the Address Validation. The aim of this step is to align the address representation between the Buyer and the Seller. This is to overcome the very common problem of different address representation in various countries and systems. The Buyer sends a represen-tation of the address that is intended to be used in further steps (most likely an installation place). The question is "Dear Seller – do you recognize and understand this address?". Additionally, the Buyer may also ask the Seller to provide alternatives if there is no clear match. The Seller provides a response where in the "bestMatchGeographicAddress" (if found) a matching address is provided with an id that can be used in further steps to avoid the need for Address resolution.

Note: It is not mandatory for the Seller to provide the Id of the returned Address, yet it is recom-mended.

Note: The Seller's response might come with some enhancements in the Address. It is up to the Seller's discretion what makes the best match and an alternative.

The Buyer in the request places one of 4 possible representations of the Address (FieldedAddress, FormattedAddress, MEFGeographicPoint, or GeographicAddressLabel). The following Figure and snippet present an example request:

@type	FieldedAddress
streetNr	20
streetNrSuffix	14
streetName	Example
city	New York
postcode	10279
country	United States

provideAlternative	☐ false
submittedGeographicAddress	●

326

Figure A2-3 – UC1: Address Validation request

328   Example Address Validation Request:
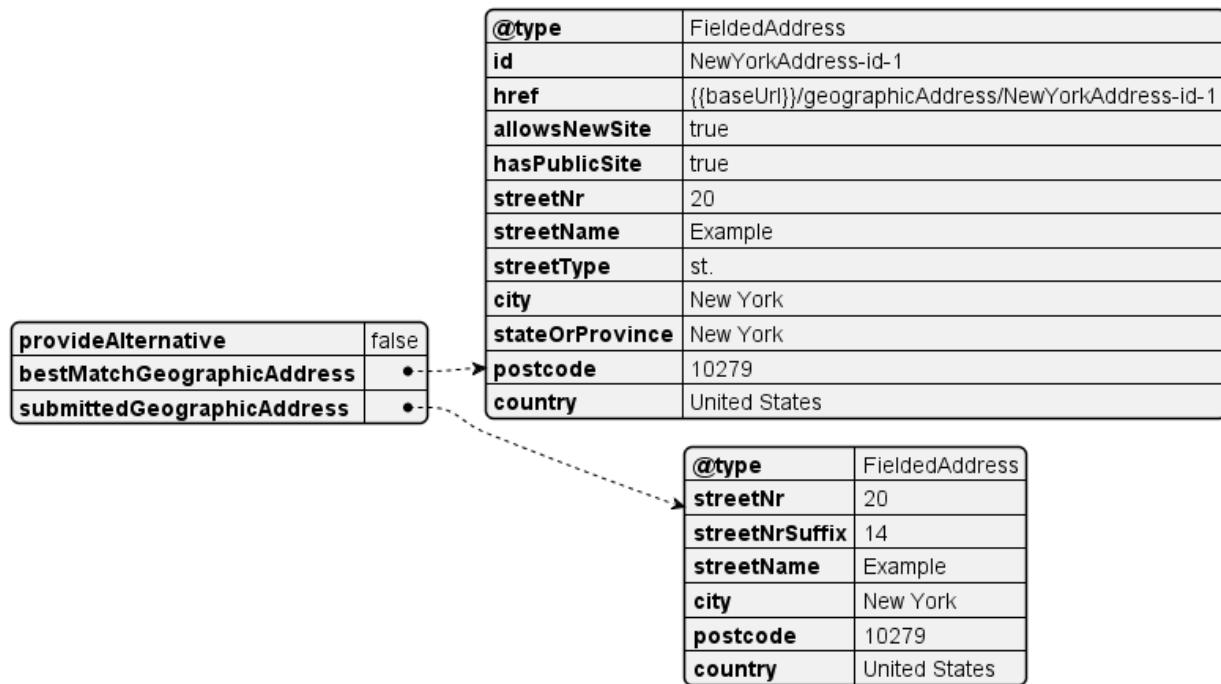
```
329  {
330     "provideAlternative": false,
331     "submittedGeographicAddress": {
332        "@type": "FieldedAddress",
333        "streetNr": "20",
334        "streetNrSuffix": "14",
335        "streetName": "Example",
336        "city": "New York",
337        "postcode": "10279",
338        "country": "United States"
339     }
340  }
```

341

342   In the response, the Seller repeats the submitted address for reference and populates the "best-
343   MatchGeographicAddress" and/or the "alternateGeographicAddress". In the example, the Seller
344   matches the best match address, which has a little more details than the one in the request. The
345   Seller also provides the address id ("NewYorkAddress-id-1") that the Buyer will refer to in later
346   steps.

347   *Note:* The identifiers will most likely be some kind of technical ids to provide uniqueness. In all
348   examples, the identifiers are shortened and made human-readable to make it easier to read and
349   match across the use cases.

@type	FieldedAddress
id	NewYorkAddress-id-1
href	{{baseUrl}}/geographicAddress/NewYorkAddress-id-1
allowsNewSite	true
hasPublicSite	true
streetNr	20
streetName	Example
streetType	st.
city	New York
stateOrProvince	New York
postcode	10279
country	United States

provideAlternative	false
bestMatchGeographicAddress	●
submittedGeographicAddress	●

@type	FieldedAddress
streetNr	20
streetNrSuffix	14
streetName	Example
city	New York
postcode	10279
country	United States

350

Figure A2-4 – UC1: Address Validation response

352

Seller's response:

```
354   {
355     "provideAlternative": "false",
356     "bestMatchGeographicAddress": {
357       "@type": "FieldedAddress",
358       "id": "NewYorkAddress-id-1",
359       "href": "{{baseUrl}}/geographicAddress/NewYorkAddress-id-1",
360       "allowsNewSite": "true",
361       "hasPublicSite": "true",
362       "streetNr": "20",
363       "streetName": "Example",
364       "streetType": "st.",
365       "city": "New York",
366       "stateOrProvince": "New York",
367       "postcode": "10279",
368       "country": "United States"
369     },
370     "submittedGeographicAddress": {
371       "@type": "FieldedAddress",
372       "streetNr": "20",
373       "streetNrSuffix": "14",
374       "streetName": "Example",
375       "city": "New York",
376       "postcode": "10279",
377       "country": "United States"
378     }
379   }
```

380   **A.3.2   Use Case 2a: POQ - new EPL, new UNIs, low class of service**

381   For detailed guidance on how to use the Product Offering Qualification API, please refer to MEF
382   87 [4].

383 The Product Offering Qualification step is designed for the Buyer to ask the question "Dear Seller,
384 are you able to provide a certain product (based on "productOffering" and "productConfigura-
385 tion") at a given location"? The Seller responds with qualification confidence:

386 • green - The Seller has high confidence that this Product can be delivered,

387 • yellow - The Seller believes they can deliver the Product but is not highly confident,

388 • red - The Seller cannot deliver the Product as specified.

389 In case of yellow or red, additionally, the Seller may return (if requested) an alternative Product
390 Offering, that might fulfill the Buyer's needs.

391 It is very important to understand the pattern of integrating the product configuration (so-called
392 "payload") with the functional product-agnostic API ("envelope"). As explained in MEF 125 [3],
393 the EPL product model is composed of 2 elements (products):

394 • the EPL itself. It contains the "evcEndPointA" and "evcEndPointZ" attributes, which
395 carry some endpoint configuration information, yet these are no references to the UNI
396 products.

397 • the UNIs

398 The information about one single product is carried within the Product Offering Qualification
399 (POQ) API by a single "productOfferingQualificationItem" being a subject to qualification. One
400 POQ Request can carry more than one POQ Items, that may or may not be related to each other.

401 There are 2 ways to reference products:

402 • existing Products – present in the Inventory at the moment of issuing the request, to
403 which the Buyer has the "product.id". These must be referenced by "productOffer-
404 ingQualificationItem.product.productRelationship" with appropriate "product.id" and
405 "relationshipType". The Product Specification defines what roles must be used during
406 referencing other products as specified in Chapter 13.

407 • newly created or modified products – ones being created or modified by other POQ
408 Item in the same POQ request, so there is a relation between the Items within a POQ.
409 These must be referenced using the "productOfferingQualificationItem.qualifica-
410 tionItemRelationship" by the target Item "id" and the "relationshipType" (CON-
411 NECTS_TO_UNI_A) and (CONNECTS_TO_UNI_Z).

412 In this use case, both the EPL and the UNIs products are created or, to be more precise, a request
413 to qualify if the creation of both of them is possible. Since 3 products are being subject to qualifi-
414 cation, the POQ request contains 3 items with "action=add". The EPL POQ Item has 2 relations:

415 • to the first UNI (NewYork_UNI), which is being qualified in the same request – by
416 "productOfferingQualificationItem.qualificationItemRelationship"

417 • to the second UNI (Washington_UNI), which is being qualified in the same request –
418 by "productOfferingQualificationItem.qualificationItemRelationship"

419 An instance diagram in Figure A2-5 shows an extracted part from the request, to present the most
420 important integration-related attributes. The product configuration attached to a POQ request is
421 highlighted with green color, and the product relations are highlighted with a bold font.

422

423
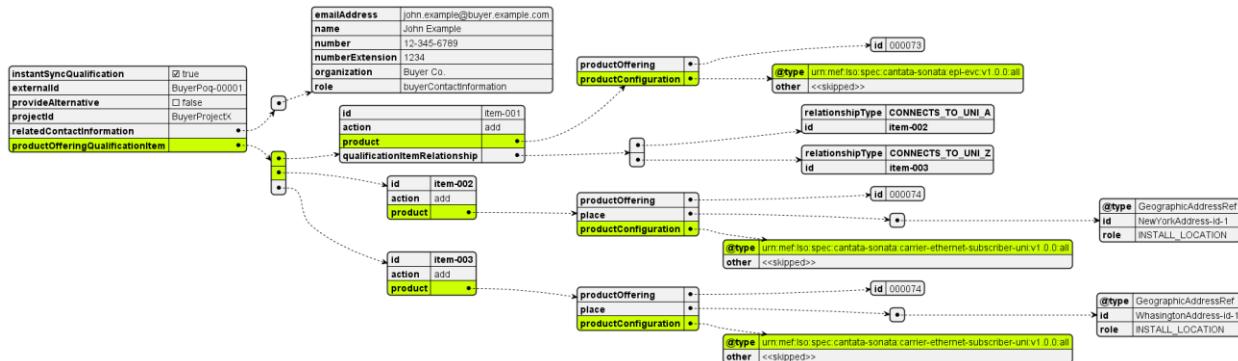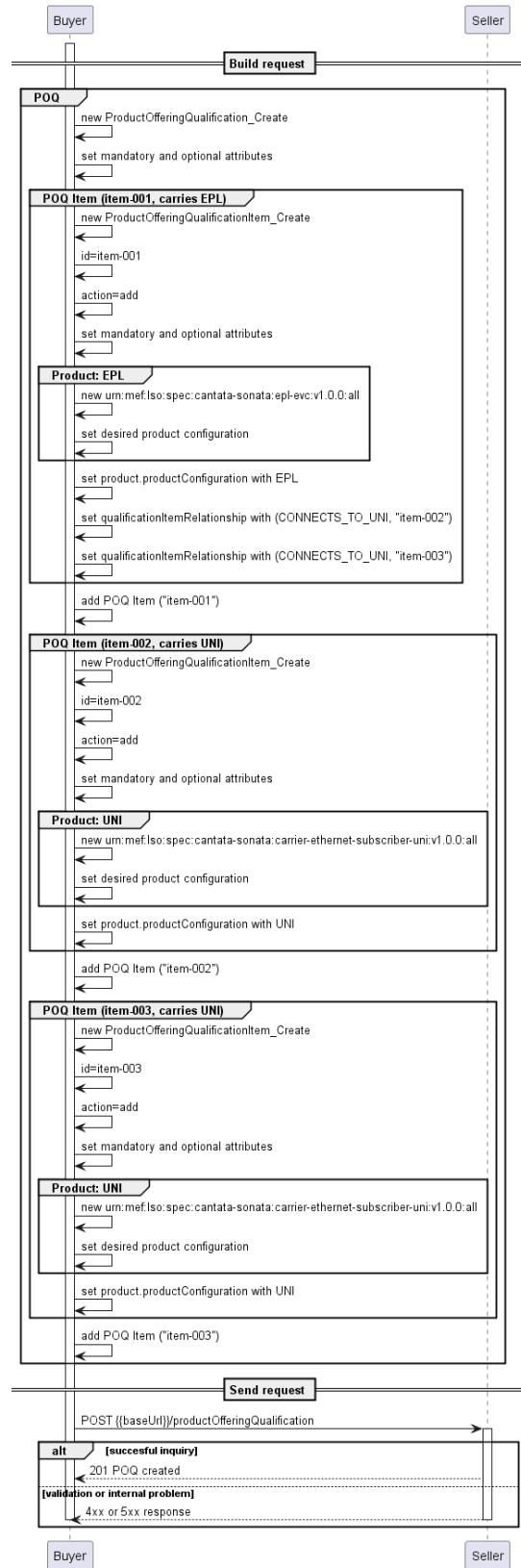
424 **Figure A2-5 – UC2a: POQ Request, envelope part**

425 The sequence diagram below (Figure A2-6) shows a set of logical steps of building the POQ re-
426 quest:

Figure A2-6 – UC2a: POQ request building steps

429 The products' integration with the API is covered. Let's go to some details of the products' con-
430 figuration. The setup of the Use Case 2 is presented in Figure A2-7.

431
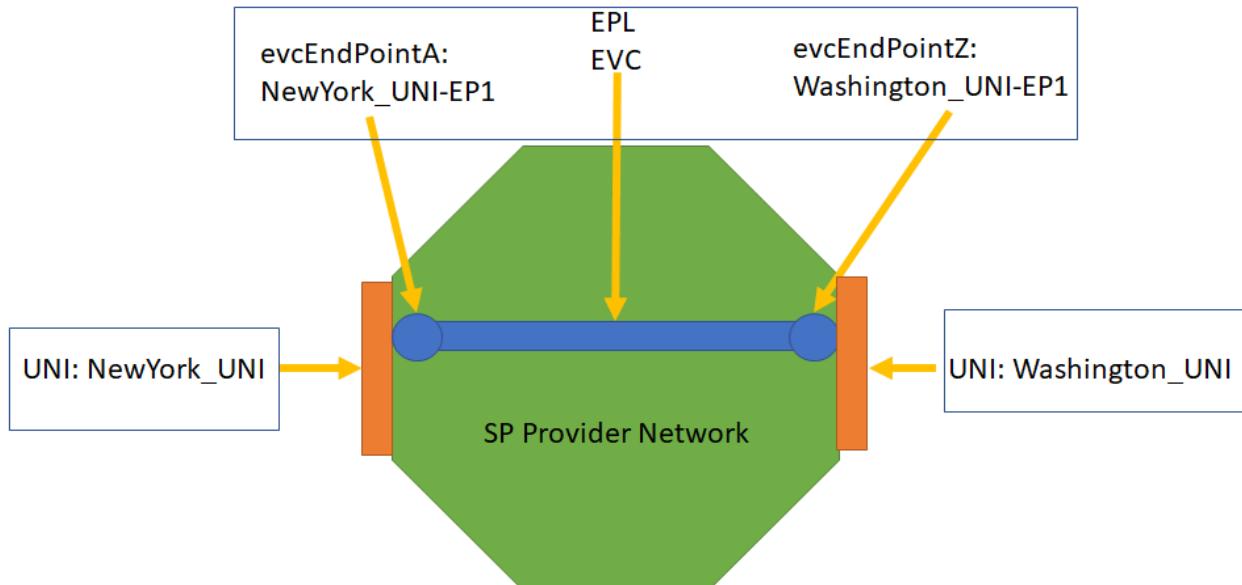
Figure A2-7 – UC2a: EPL Setup Diagram

432

433 This setup involves:

434 • Creation of the UNIs

435 o place: New York (Address id acquired in Use Case 1)

436 o id="NewYork_UNI"

437 o place: Washington (Address id acquired in Use Case 1)

438 o id="Washington_UNI"

439 • Creation of the EPL including:

440 o configuration of a new UNI Endpoint with id="NewYork_UNI-EP1", at the
441 UNI with id="NewYork_UNI", which is also created within the same request.

442 o configuration of a new UNI Endpoint with id="Washington_UNI-EP1", at the
443 UNI with id=" Washington_UNI _UNI", which is also created within the same
444 request.
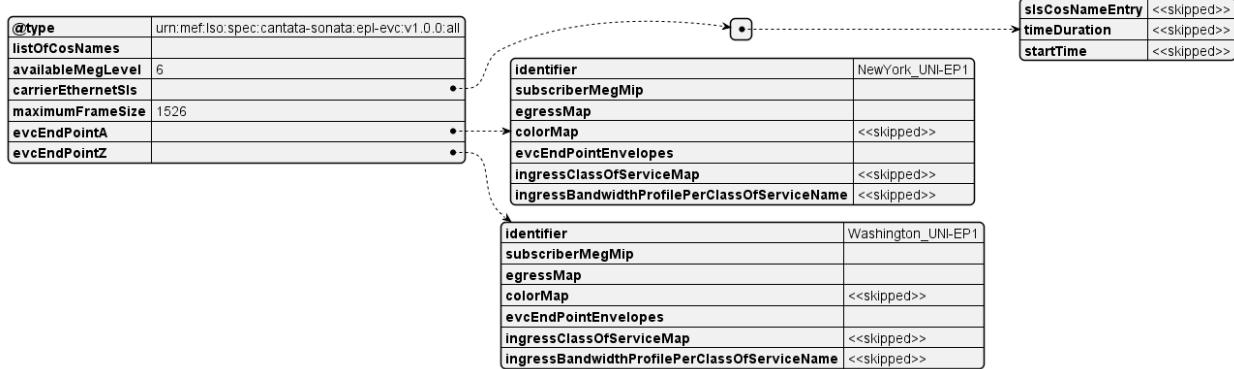
445 The diagram aggregates the scope of a particular product configuration into rectangles. This is to
446 stress that the UNI endpoints are parts of the EPL configuration. They are not individual orderable
447 products (this is the case in point-to-point connections).

448 The instance diagram for the whole EPL configuration is too big to be presented as a whole so it
449 is split and presented in parts. Figure A2-8 shows the basic EPL attributes. This diagram as at-
450 tached to Figure A2-5 as the node with "@type=urn:mef:lso:spec:cantata-sonata:epl-

451 evc:v1.0.0:all". The attributes that are skipped on this level are marked with a "<<skipped>>"
452 label and will be presented on the next diagrams.

453

454
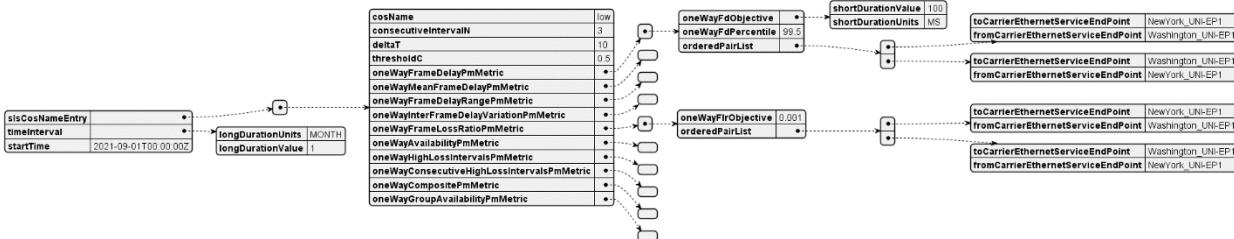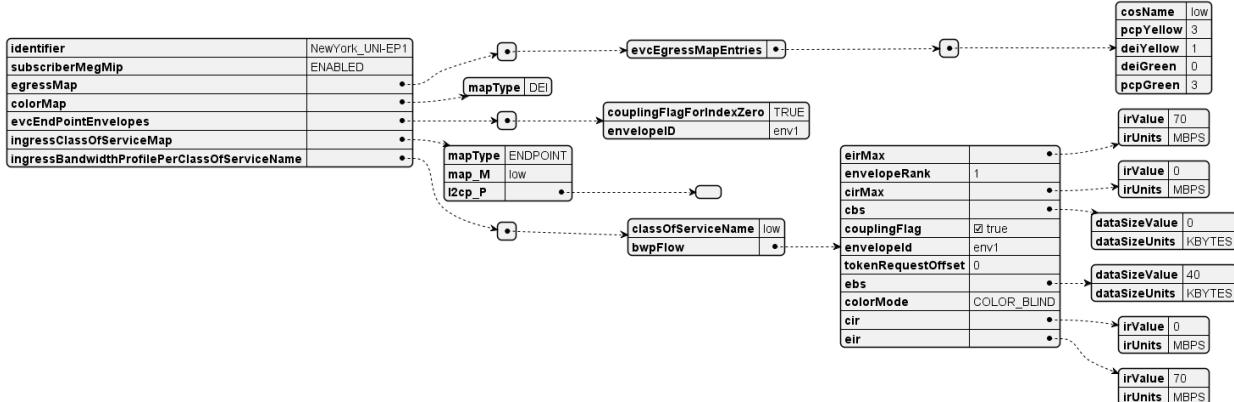
Figure A2-8 – UC2a: EPL basic attributes

456 The structures defining the "carrierEthernetSls" and the "evcEndPointA" are complex and pre-
457 sented in the following figures:

458

Figure A2-9 – UC2a: EPL Carrier Ethernet SLS

460
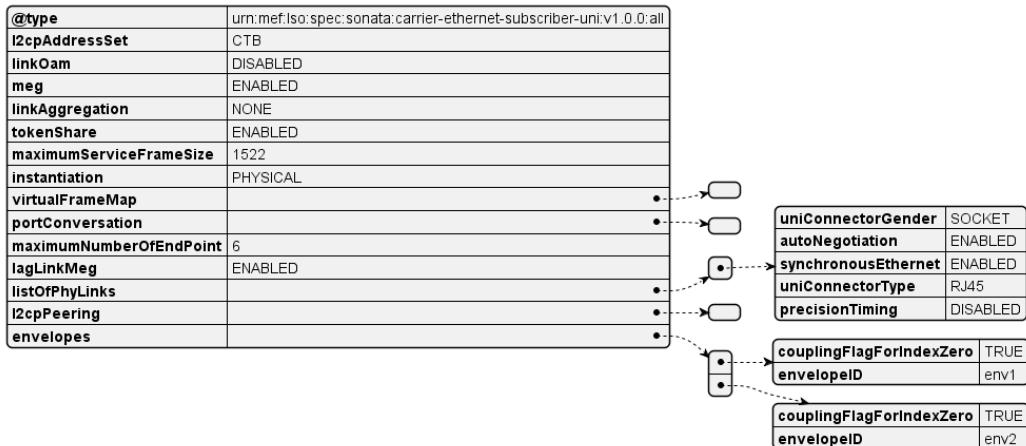
Figure A2-10 – UC2a: EPL UNI Endpoint

462 The last figure in this use case presents the UNI product configuration.

463

464
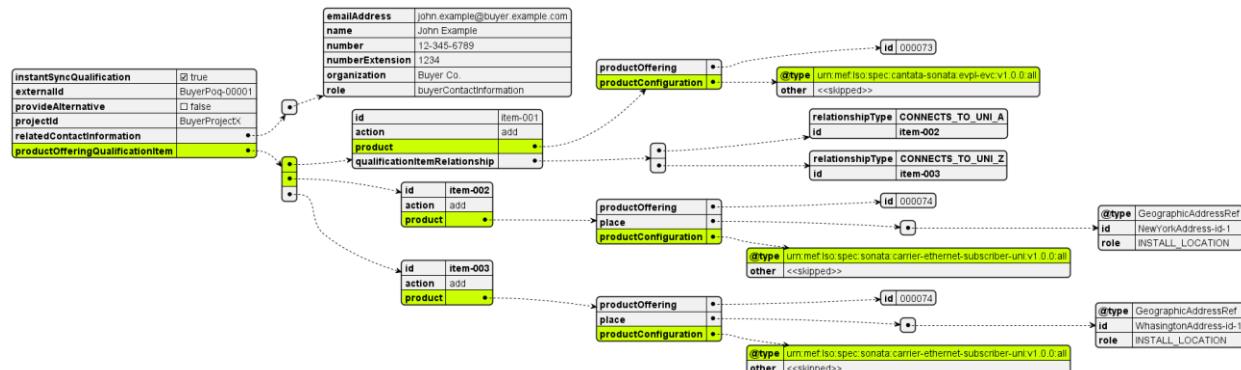
<div align="center">**Figure A2-11 – UC2a: UNI**</div>

A.3.3   Use Case 2b: POQ - new EVPL, new UNIs, low class of service

466
467
Detailed description of POQ use case is located in A.3.2 part. This section will describe the unique features of the EVPL technology.

468
469
EPL and EVPL are very similar technologies. However, there are a few differences at the connection attribute level. They will be highlighted in the following diagrams.

470
471
472
An instance diagram in Figure A2-12 shows an extracted part from the request, to present the most important integration-related attributes. The product configuration attached to a POQ request is highlighted with green color, and the product relations are highlighted with a bold font.

473

474
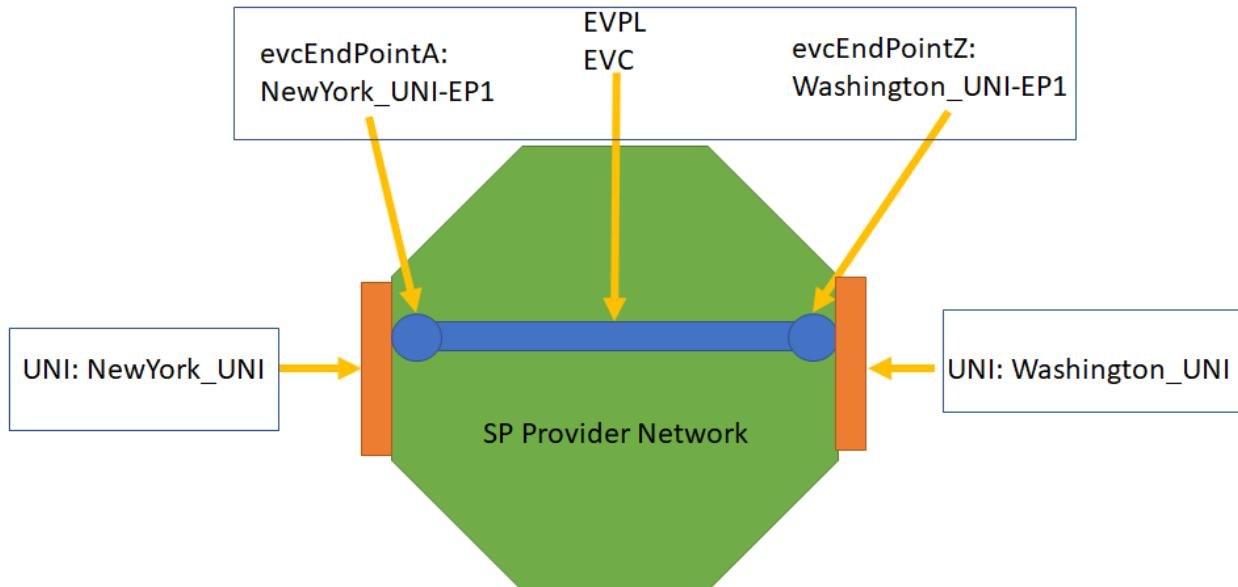
<div align="center">**Figure A2-12 – UC2b: POQ Request, envelope part**</div>

475
476
The products' integration with the API is covered. Let's go to some details of the products' configuration. The setup of the Use Case 2b is presented in Figure A2-13.

477

Figure A2-13 – UC2b: EVPL Setup Diagram
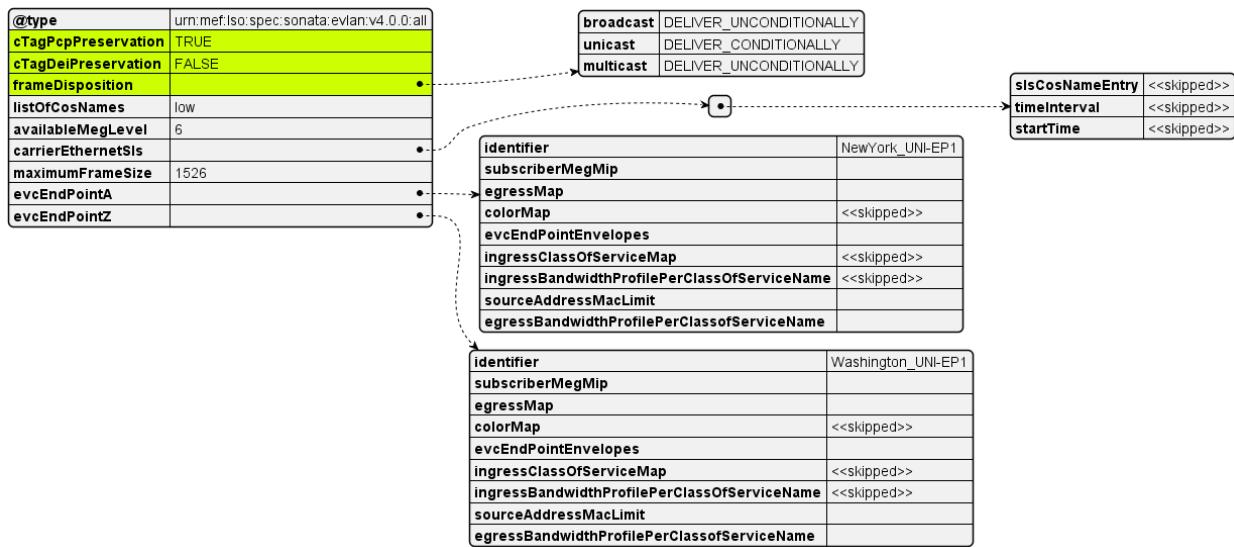
479 This setup involves:
480   • Creation of two the UNIs
481   o place: New York (Address id acquired in Use Case 1)
482   o id="NewYork_UNI"
483   o place: Washington (Address id acquired in Use Case 1)
484   o id="Washington_UNI"
485   • Creation of the EVPL, including:
486   o configuration of a new UNI Endpoint with id="NewYork_UNI-EP1", at the
487   UNI with id="NewYork_UNI", which is also created within the same request.
488   o configuration of a new UNI Endpoint with id=" Washington _UNI-EP1", at
489   the UNI with id=" Washington _UNI", which is also created within the same
490   request.

491 The diagram aggregates the scope of a particular product configuration into rectangles. This is to
492 stress that the UNI Endpoints are parts of the EVPL configuration. They are not individual order-
493 able products (this is the case in point-to-point connections).

494 The instance diagram for the whole EVPL configuration is too big to be presented as a whole so it
495 is split and presented in parts. Figure A2-14 shows the basic EVPL attributes. This diagram as
496 attached to Figure A2-12 as the node with "@type=urn:mef:lso:spec:cantata-sonata:evpl-
497 evc:v1.0.0:all". The attributes that are skipped on this level are marked with a "<<skipped>>"
498 label and will be presented on the next diagrams.

499

| @type	urn:mef:lso:spec:sonata:evlan:v4.0.0:all
cTagPcpPreservation	TRUE
cTagDeiPreservation	FALSE
frameDisposition	●
listOfCosNames	low
availableMegLevel	6
carrierEthernetSls	●
maximumFrameSize	1526
evcEndPointA	●
evcEndPointZ	●

broadcast	DELIVER_UNCONDITIONALLY
unicast	DELIVER_CONDITIONALLY
multicast	DELIVER_UNCONDITIONALLY

slsCosNameEntry	<<skipped>>
timeInterval	<<skipped>>
startTime	<<skipped>>

identifier	NewYork_UNI-EP1
subscriberMegMip	
egressMap	
colorMap	<<skipped>>
evcEndPointEnvelopes	
ingressClassOfServiceMap	<<skipped>>
ingressBandwidthProfilePerClassOfServiceName	<<skipped>>
sourceAddressMacLimit	
egressBandwidthProfilePerClassofServiceName	

identifier	Washington_UNI-EP1
subscriberMegMip	
egressMap	
colorMap	<<skipped>>
evcEndPointEnvelopes	
ingressClassOfServiceMap	<<skipped>>
ingressBandwidthProfilePerClassOfServiceName	<<skipped>>
sourceAddressMacLimit	
egressBandwidthProfilePerClassofServiceName	

500

501 **Figure A2-14 – UC2b: EVPL basic attributes**

502 The structures defining the UNI Endpoint are complex and presented in the following figures:

identifier	NewYork_UNI-EP1
subscriberMegMip	ENABLED
egressMap	●
colorMap	●
evcEndPointEnvelopes	●
ingressClassOfServiceMap	●
ingressBandwidthProfilePerClassOfServiceName	●
evcEndPointMap	●
sourceAddressMacLimit	●
egressBandwidthProfilePerClassofServiceName	●

503

504 **Figure A2-15 – UC2b: EVPL UNI Endpoint**

505 POQ Request example:
506 {
507   "instantSyncQualification": true,
508   "externalId": "BuyerPoq-00001",
509   "provideAlternative": false,
510   "projectId": "BuyerProjectX",
511   "relatedContactInformation": [
512   {
513   "emailAddress": "john.example@buyer.com",
514   "name": "John Example",
515   "number": "12-345-6789",
516   "numberExtension": "1234",
517   "organization": "Buyer Co.",

```
518            "role": "buyerContactInformation"
519          }
520        ],
521        "productOfferingQualificationItem": [
522          {
523            "id": "item-001",
524            "action": "add",
525            "qualificationItemRelationship": [
526              {
527                "relationshipType": "CONNECTS_TO_UNI_A",
528                "id": "item-002"
529              },
530              {
531                "relationshipType": "CONNECTS_TO_UNI_Z",
532                "id": "item-003"
533              }
534            ],
535            "product": {
536              "productOffering": {
537                "id": "000073"
538              },
539              "productConfiguration": {
540                "@type": "urn:mef:lso:spec:cantata-sonata:evpl-evc:v1.0.0:all",
541                "cTagPcpPreservation": "TRUE",
542                "cTagDeiPreservation": "FALSE",
543                "frameDisposition": {
544                  "broadcast": "DELIVER_UNCONDITIONALLY",
545                  "unicast": "DELIVER_CONDITIONALLY",
546                  "multicast": "DELIVER_UNCONDITIONALLY"
547                },
548                "listOfCosNames": ["low"],
549                "availableMegLevel": "6",
550                "carrierEthernetSls": [
551                  {
552                    "slsCosNameEntry": [
553                      {
554                        "cosName": "low",
555                        "consecutiveIntervalN": 3,
556                        "deltaT": 10,
557                        "thresholdC": 0.5,
558                        "oneWayFrameDelayPmMetric": [
559                          {
560                            "oneWayFdObjective": {
561                              "shortDurationValue": 100,
562                              "shortDurationUnits": "MS"
563                            },
```

```
564                             "oneWayFdPercentile": 99.5,
565                             "orderedPairList": [
566                               {
567                                 "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
568                                 "fromCarrierEthernetServiceEndPoint": "Washington_UNI-
569    EP1"
570                               },
571                               {
572                                 "toCarrierEthernetServiceEndPoint": "Washington_UNI-
573    EP1",
574                                 "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
575                               }
576                             ]
577                           }
578                         ],
579                         "oneWayMeanFrameDelayPmMetric": [],
580                         "oneWayFrameDelayRangePmMetric": [],
581                         "oneWayInterFrameDelayVariationPmMetric": [],
582                         "oneWayFrameLossRatioPmMetric": [
583                           {
584                             "oneWayFlrObjective": 0.001,
585                             "orderedPairList": [
586                               {
587                                 "toCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1",
588                                 "fromCarrierEthernetServiceEndPoint": "Washington_UNI-
589    EP1"
590                               },
591                               {
592                                 "toCarrierEthernetServiceEndPoint": "Washington_UNI-
593    EP1",
594                                 "fromCarrierEthernetServiceEndPoint": "NewYork_UNI-EP1"
595                               }
596                             ]
597                           }
598                         ],
599                         "oneWayAvailabilityPmMetric": [],
600                         "oneWayHighLossIntervalsPmMetric": [],
601                         "oneWayConsecutiveHighLossIntervalsPmMetric": [],
602                         "oneWayCompositePmMetric": [],
603                         "oneWayGroupAvailabilityPmMetric": []
604                       }
605                     ],
606                     "timeDuration": {
607                       "timeDurationValue": 1,
608                       "timeDurationUnits": "MONTH"
609                     },
```

```
610                    "startTime": "2022-10-12T00:00:00Z"
611                 }
612              ],
613              "maximumFrameSize": 1522,
614              "evcEndPointA": {
615                 "identifier": "NewYork_UNI-EP1",
616                 "subscriberMegMip": "ENABLED",
617                 "egressMap": [
618                    {
619                       "evcEgressMapEntries": [
620                          {
621                             "cosName": "low",
622                             "pcpYellow": 3,
623                             "deiYellow": 1,
624                             "deiGreen": 0,
625                             "pcpGreen": 3
626                          }
627                       ]
628                    }
629                 ],
630                 "colorMap": {
631                    "mapType": "DEI"
632                 },
633                 "evcEndPointEnvelopes": [
634                    {
635                       "couplingFlagForIndexZero": "TRUE",
636                       "envelopeID": "env1"
637                    }
638                 ],
639                 "ingressClassOfServiceMap": {
640                    "mapType": "ENDPOINT",
641                    "map_M": "low",
642                    "l2cp_P": []
643                 },
644                 "ingressBandwidthProfilePerClassOfServiceName": [
645                    {
646                       "classOfServiceName": "low",
647                       "bwpFlow": {
648                          "eirMax": {
649                             "irValue": 70,
650                             "irUnits": "MBPS"
651                          },
652                          "envelopeRank": 1,
653                          "cirMax": {
654                             "irValue": 0,
655                             "irUnits": "MBPS"
```

```
656              },
657              "cbs": {
658                "dataSizeValue": 0,
659                "dataSizeUnits": "KBYTES"
660              },
661              "couplingFlag": true,
662              "envelopeId": "env1",
663              "tokenRequestOffset": 0,
664              "ebs": {
665                "dataSizeValue": 40,
666                "dataSizeUnits": "KBYTES"
667              },
668              "colorMode": "COLOR_BLIND",
669              "cir": {
670                "irValue": 0,
671                "irUnits": "MBPS"
672              },
673              "eir": {
674                "irValue": 70,
675                "irUnits": "MBPS"
676              }
677            }
678          }
679        ],
680        "evcEndPointMap": [
681          {
682            "vlanId": [2, 3],
683            "vlanType": "LIST"
684          }
685        ],
686        "sourceAddressMacLimit": [],
687        "egressBandwidthProfilePerClassofServiceName": []
688      },
689      "evcEndPointZ": {
690        "identifier": "Washington_UNI-EP1",
691        "subscriberMegMip": "ENABLED",
692        "egressMap": [
693          {
694            "evcEgressMapEntries": [
695              {
696                "cosName": "low",
697                "pcpYellow": 3,
698                "deiYellow": 1,
699                "deiGreen": 0,
700                "pcpGreen": 3
701              }
```

```
702                        ]
703                      }
704                    ],
705                    "colorMap": {
706                      "mapType": "DEI"
707                    },
708                    "evcEndPointEnvelopes": [
709                      {
710                        "couplingFlagForIndexZero": "TRUE",
711                        "envelopeID": "env1"
712                      }
713                    ],
714                    "ingressClassOfServiceMap": {
715                      "mapType": "ENDPOINT",
716                      "map_M": "low",
717                      "l2cp_P": []
718                    },
719                    "ingressBandwidthProfilePerClassOfServiceName": [
720                      {
721                        "classOfServiceName": "low",
722                        "bwpFlow": {
723                          "eirMax": {
724                            "irValue": 70,
725                            "irUnits": "MBPS"
726                          },
727                          "envelopeRank": 1,
728                          "cirMax": {
729                            "irValue": 0,
730                            "irUnits": "MBPS"
731                          },
732                          "cbs": {
733                            "dataSizeValue": 0,
734                            "dataSizeUnits": "KBYTES"
735                          },
736                          "couplingFlag": true,
737                          "envelopeId": "env1",
738                          "tokenRequestOffset": 0,
739                          "ebs": {
740                            "dataSizeValue": 40,
741                            "dataSizeUnits": "KBYTES"
742                          },
743                          "colorMode": "COLOR_BLIND",
744                          "cir": {
745                            "irValue": 0,
746                            "irUnits": "MBPS"
747                          },
```

```json
748                    "eir": {
749                       "irValue": 70,
750                       "irUnits": "MBPS"
751                    }
752                  }
753                }
754              ],
755              "evcEndPointMap": [
756                {
757                   "vlanId": [2, 3],
758                   "vlanType": "LIST"
759                }
760              ],
761              "sourceAddressMacLimit": [],
762              "egressBandwidthProfilePerClassofServiceName": []
763            }
764          }
765        }
766      },
767      {
768        "id": "item-002",
769        "action": "add",
770        "product": {
771          "productOffering": {
772            "id": "000074"
773          },
774          "place": [
775            {
776               "@type": "GeographicAddressRef",
777               "id": "NewYorkAddress-id-1",
778               "role": "INSTALL_LOCATION"
779            }
780          ],
781          "relatedContactInformation": [
782            {
783               "number": "+12-345-678-90",
784               "emailAddress": "LocationContact@example.com",
785               "role": "locationContact",
786               "name": "Location Contact"
787            }
788          ],
789          "productConfiguration": {
790            "@type": "urn:mef:lso:spec:sonata:carrier-ethernet-subscriber-
791  uni:v1.0.0:all",
792            "listOfPhyLinks": [
793               {
```

```
794              "uniConnectorGender": "SOCKET",
795              "autoNegotiation": "ENABLED",
796              "synchronousEthernet": "ENABLED",
797              "uniConnectorType": "RJ45",
798              "precisionTiming": "DISABLED"
799            }
800          ],
801          "virtualFrameMap": [],
802          "portConversation": [],
803          "maximumNumberOfEndPoint": 6,
804          "lagLinkMeg": "ENABLED",
805          "l2cpAddressSet": "CTB",
806          "linkOam": "DISABLED",
807          "meg": "ENABLED",
808          "linkAggregation": "NONE",
809          "l2cpPeering": [],
810          "maximumNumberOfCtagVlanIdsPerEndPoint": 4094,
811          "tokenShare": "ENABLED",
812          "maximumServiceFrameSize": 1522,
813          "envelopes": [
814            {
815              "couplingFlagForIndexZero": "TRUE",
816              "envelopeID": "env1"
817            },
818            {
819              "couplingFlagForIndexZero": "TRUE",
820              "envelopeID": "env2"
821            }
822          ],
823          "instantiation": "PHYSICAL"
824        }
825      }
826    },
827    {
828      "id": "item-003",
829      "action": "add",
830      "product": {
831        "productOffering": {
832          "id": "000074"
833        },
834        "place": [
835          {
836            "@type": "GeographicAddressRef",
837            "id": "WashingtonAddress-id-1",
838            "role": "INSTALL_LOCATION"
839          }
```

```
840            ],
841         "relatedContactInformation": [
842           {
843              "number": "+12-345-678-90",
844              "emailAddress": "LocationContact@example.com",
845              "role": "locationContact",
846              "name": "Location Contact"
847           }
848         ],
849         "productConfiguration": {
850           "@type": "urn:mef:lso:spec:sonata:carrier-ethernet-subscriber-
851    uni:v1.0.0:all",
852           "listOfPhyLinks": [
853             {
854                "uniConnectorGender": "SOCKET",
855                "autoNegotiation": "ENABLED",
856                "synchronousEthernet": "ENABLED",
857                "uniConnectorType": "RJ45",
858                "precisionTiming": "DISABLED"
859             }
860           ],
861           "virtualFrameMap": [],
862           "portConversation": [],
863           "maximumNumberOfEndPoint": 6,
864           "lagLinkMeg": "ENABLED",
865           "l2cpAddressSet": "CTB",
866           "linkOam": "DISABLED",
867           "meg": "ENABLED",
868           "linkAggregation": "NONE",
869           "l2cpPeering": [],
870           "maximumNumberOfCtagVlanIdsPerEndPoint": 4094,
871           "tokenShare": "ENABLED",
872           "maximumServiceFrameSize": 1522,
873           "envelopes": [
874             {
875                "couplingFlagForIndexZero": "TRUE",
876                "envelopeID": "env1"
877             },
878             {
879                "couplingFlagForIndexZero": "TRUE",
880                "envelopeID": "env2"
881             }
882           ],
883           "instantiation": "PHYSICAL"
884         }
885       }
```

886   }
887   ]
888   }
889

A.3.4   Use Case 2c: POQ - new EP-LAN, new UNIs, new ENDPOINTs low class of service

890

891   Detailed description of POQ use case is located in A.3.2 part. This section will describe the unique
892   features of the EP-LAN technology.

893   It is very important to understand the pattern of integrating the product configuration (so-called
894   "payload") with the functional product-agnostic API ("envelope"). As explained in chapter MEF
895   125 [3]**Błąd! Nie można odnaleźć źródła odwołania.**, the EP-LAN product model is composed
896   of 3 elements (products):
897   • the EP-LAN

898   • the UNI

899   • the ENDPOINT (evcEndPoint)

900   The information about one single product is carried within the Product Offering Qualification
901   (POQ) API by a single "productOfferingQualificationItem" being a subject to qualification. One
902   POQ Request can carry more than one POQ Items, that may or may not be related to each other.
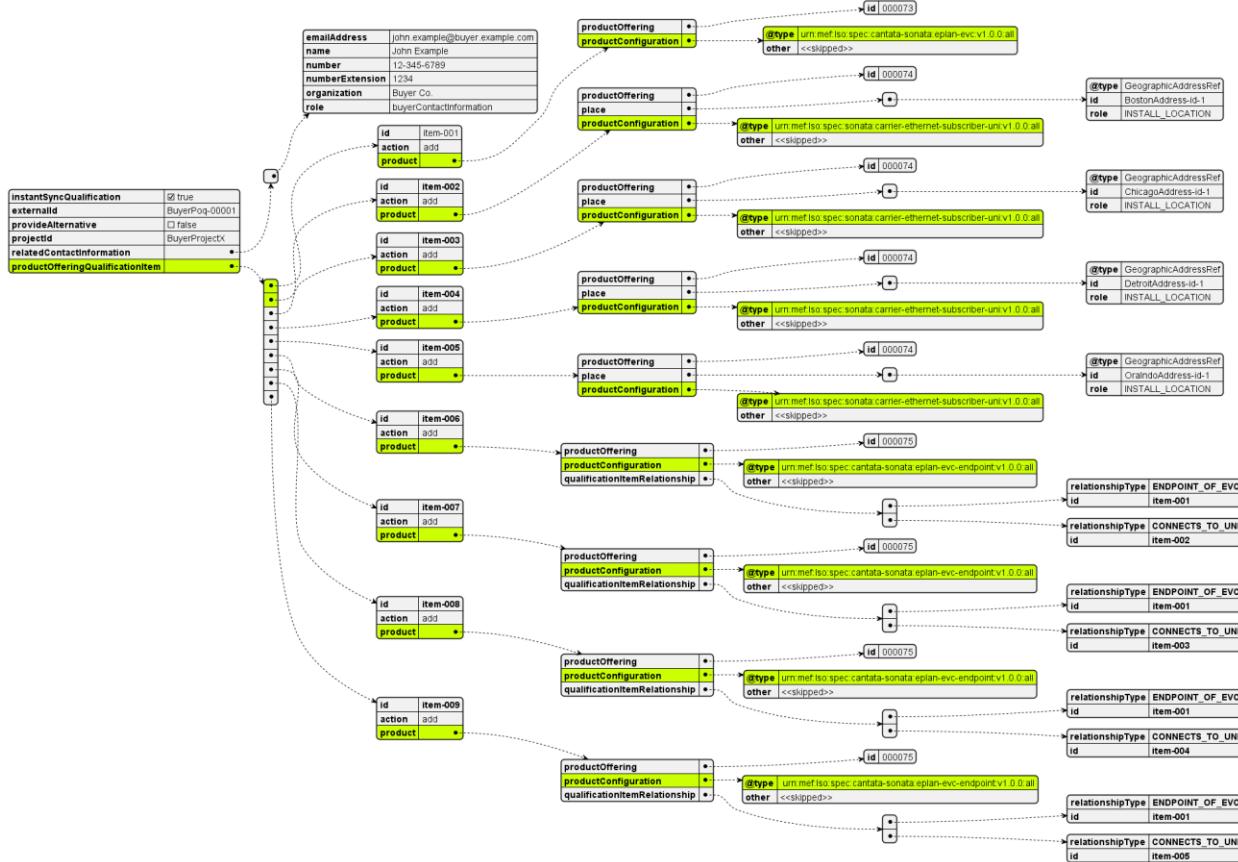
903   In this use case, both the EP-LAN (EVC), UNI and ENDPOINT (evcEndPoint) products are cre-
904   ated or, to be more precise, a request to qualify if the creation of both of them is possible.
905   • to the UNIs and ENDPOINTs (evcEndPoint), which are being qualified in the same
906   request – by
907   "productOfferingQualificationItem.qualificationItemRelationship"

908   An instance diagram in Figure A2-16 shows an extracted part from the request, to present the most
909   important integration-related attributes. The product configuration attached to a POQ request is
910   highlighted with green color, and the product relations are highlighted with a bold font.

911
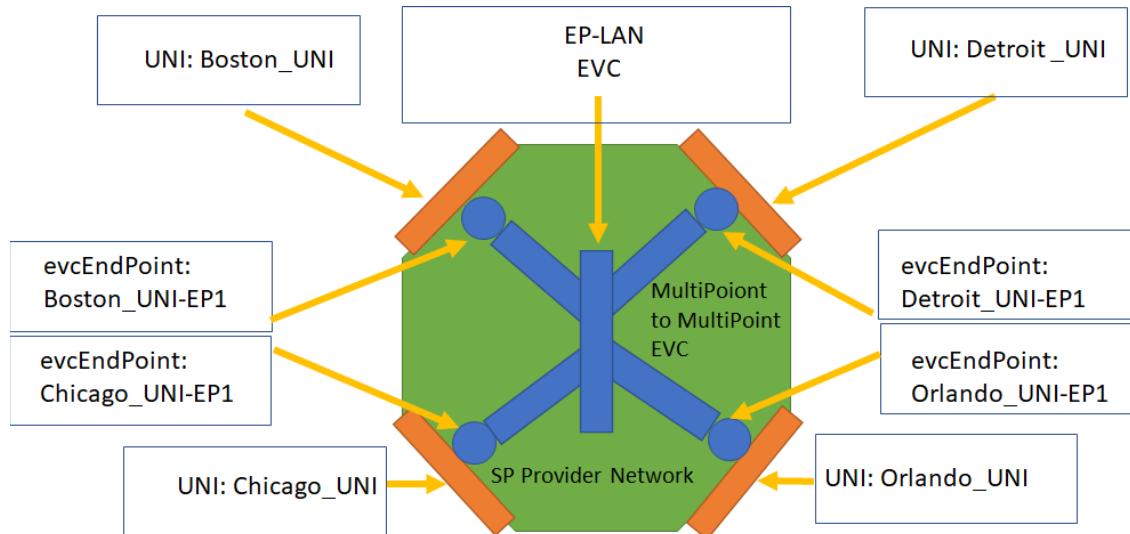
912

913 <p style="text-align:center">**Figure A2-16 – UC2c: POQ Request, envelope part**</p>

914 The products' integration with the API is covered. Let's go to some details of the products' con-
915 figuration. The setup of the Use Case 2c is presented in Figure A2-17.

916

917 <p style="text-align:center">**Figure A2-17 – UC2c: EP-LAN Setup Diagram**</p>

918 This setup involves:

919   •   Creation of the EP-LAN,

920   •   Creation of the UNIs

921   o   place: Boston (Address id acquired in Use Case 1)

922   o   id="Boston_UNI"

923   o   place: Chicago

924   o   id=" Chicago_UNI"

925   o   place: Detroit

926   o   id=" Detroit _UNI"

927   o   place: Orlando

928   o   id="Orlando_UNI"

929   •   Creation of the ENDPOINTs (evcEndPoint):

930   o   id="Boston_UNI-EP1"

931   o   id="Chicago_UNI-EP1"

932   o   id="Detroit_UNI-EP1"

933   o   id="Orlando_UNI-EP1"

934 The diagram aggregates the scope of a particular product configuration into rectangles. However,
935 unlike to EPL configuration, the ENDPOINTs (evcEndPoint) are separated products.

936 The instance diagram for the whole EP-LAN configuration is too big to be presented as a whole
937 so it is split and presented in parts. Figure A2-18 shows the basic EP-LAN attributes. This diagram
938 as attached to Figure A2-16 as the node with "@type=urn:mef:lso:spec:cantata-sonata:eplan-
939 evc:v1.0.0:all". The attributes that are skipped on this level are marked with a "<<skipped>>"
940 label and will be presented on the next diagrams.

941
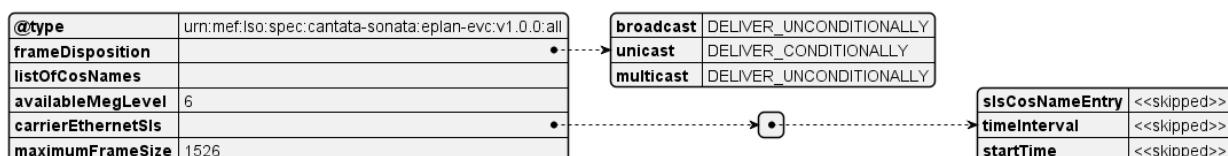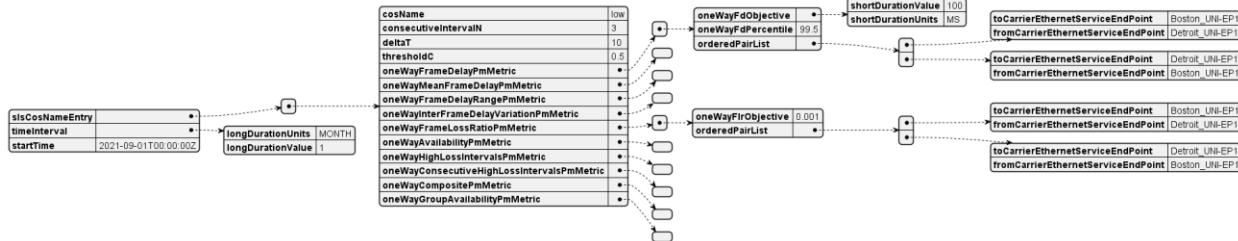
943 **Figure A2-18 – UC2c: EP-LAN basic attributes**

944 The structures defining the "carrierEthernetSls" and "evcEndPoint" are complex and presented in
945 the following figures:

946

947


948
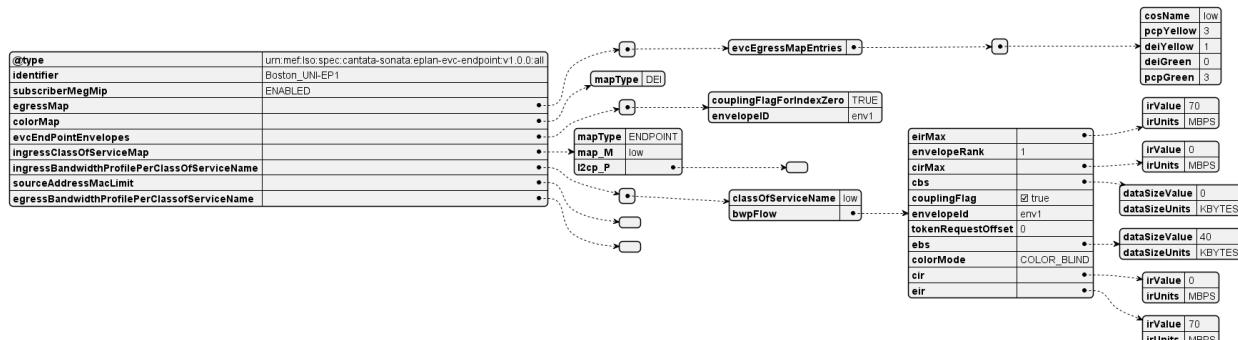
949 **Figure A2-19 – UC2c: EP-LAN Carrier Ethernet SLS**

950

951
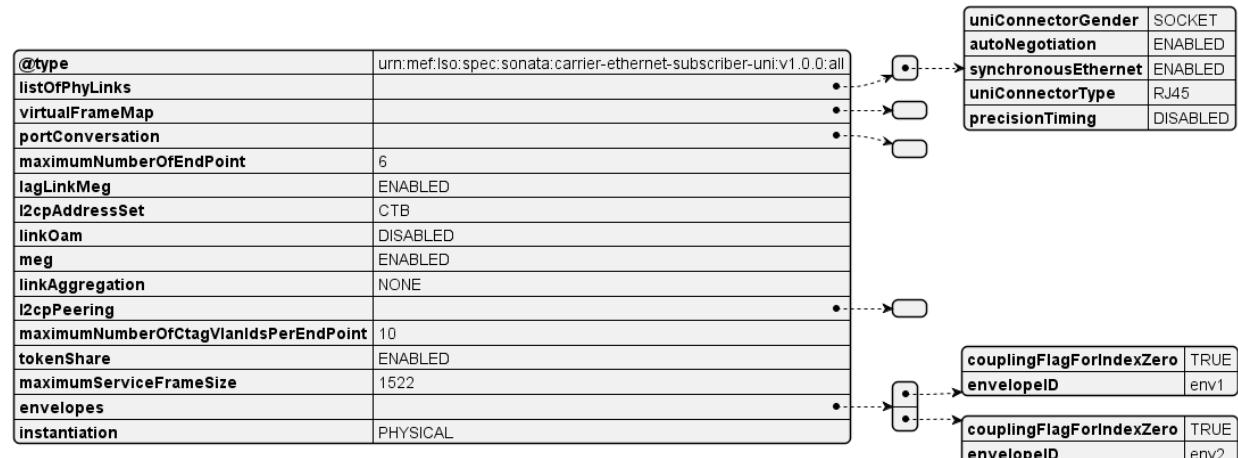
952 **Figure A2-20 – UC2c: EP-LAN ENDPOINT (evcEndPoint)**

953 The last figure in this use case presents the UNI product configuration.

954

955

956 **Figure A2-21 – UC2c: Subscriber Ethernet UNI**
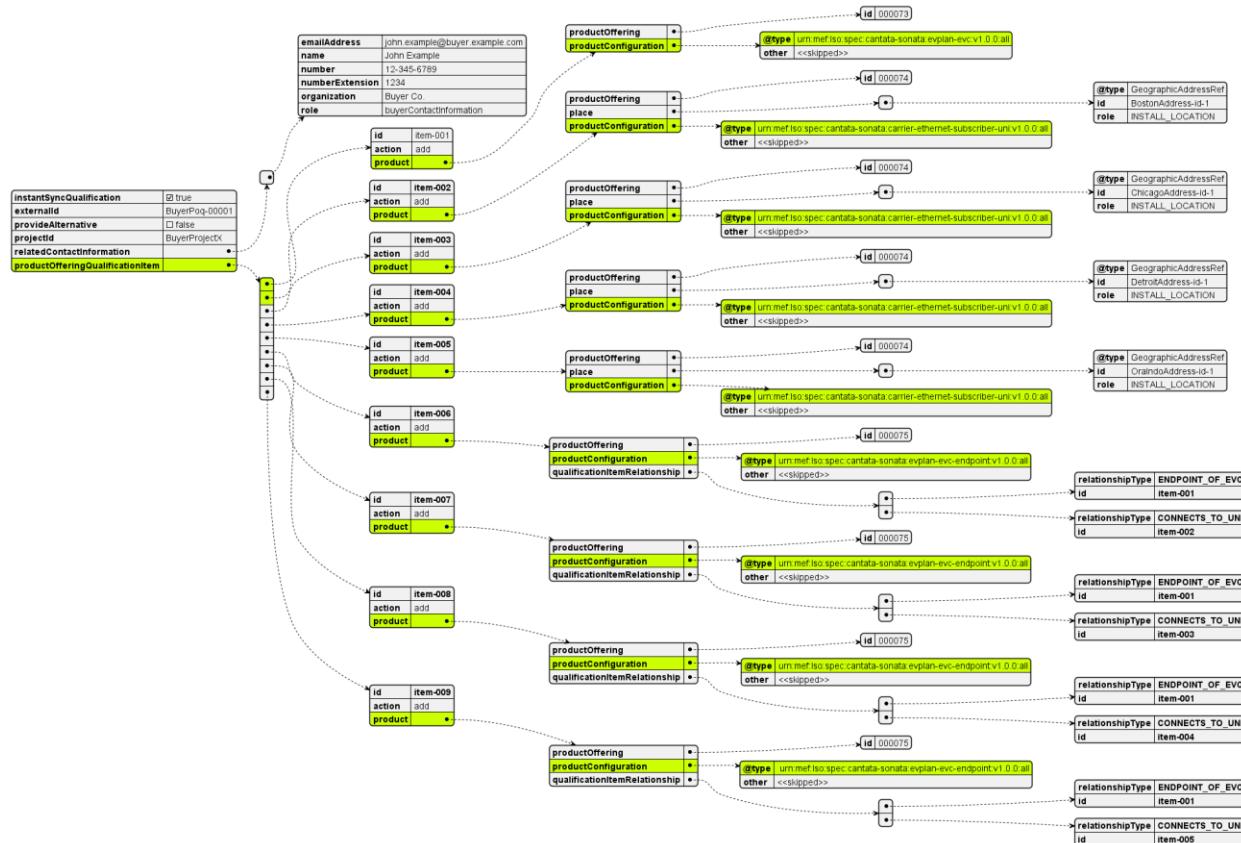
957 ### A.3.5 Use Case 2d: POQ - new EVP-LAN, new UNIs, new ENDPOINTs low class of service

958 Detailed description of POQ use case is located in A.3.2 part. This section will describe the unique
959 features of the EVP-LAN technology.

960 EP-LAN and EVP-LAN are very similar technologies. However, there are a few differences at the
961 connection attribute level. They will be highlighted in the following diagrams.

962 An instance diagram in Figure A2-22 shows an extracted part from the request, to present the most
963 important integration-related attributes. The product configuration attached to a POQ request is
964 highlighted with green color, and the product relations are highlighted with a bold font.

965

966

967 **Figure A2-22 – UC2d: POQ Request, envelope part**

968 The products' integration with the API is covered. Let's go to some details of the products' con-
969 figuration. The setup of the Use Case 2d is presented in Figure A2-23.

970
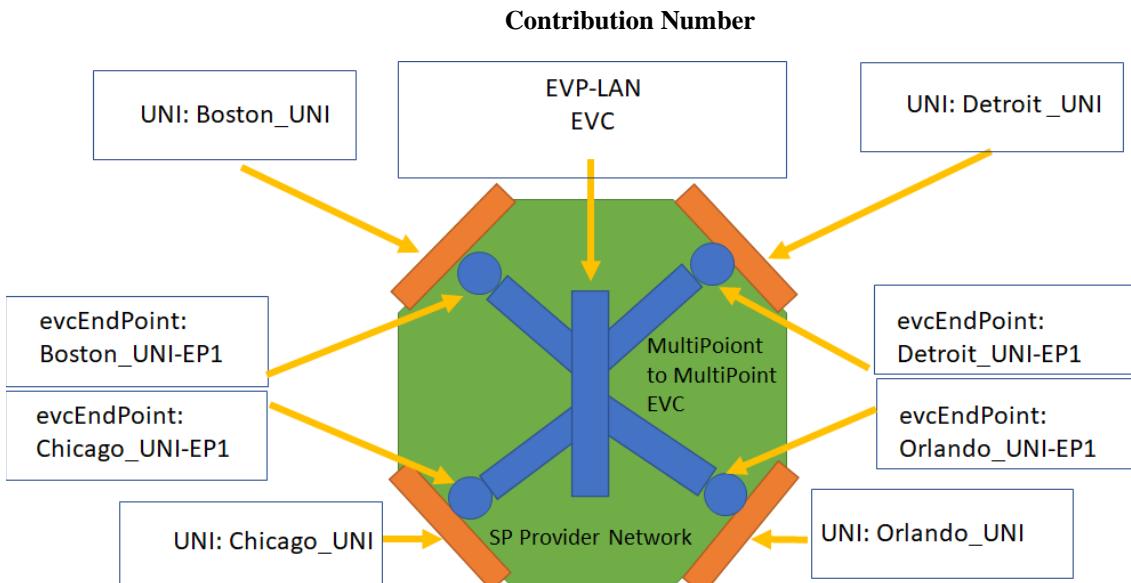
Figure A2-23 – UC2d: EVP-LAN Setup Diagram

This setup involves:

- Creation of the EP-LAN,

- Creation of the UNIs

    o place: Boston (Address id acquired in Use Case 1)

    o id="Boston_UNI"

    o place: Chicago

    o id="Chicago_UNI"

    o place: Detroit

    o id=" Detroit _UNI"

    o place: Orlando

    o id="Orlando_UNI"

- Creation of the ENDPOINTs (evcEndPoint)

    o id="Boston_UNI-EP1"

    o id="Chicago_UNI-EP1"

    o id="Detroit_UNI-EP1"

    o id="Orlando_UNI-EP1"

The diagram aggregates the scope of a particular product configuration into rectangles. However, unlike to EPL configuration, the ENDPOINTs (evcEndPoint) are separated products.

990 The instance diagram for the whole EVP-LAN configuration is too big to be presented as a whole
991 so it is split and presented in parts. Figure A2-24 shows the basic EVP-LAN attributes. This dia-
992 gram as attached to Figure A2-22 as the node with "@type=urn:mef:lso:spec:cantata-so-
993 nata:evplan-evc:v1.0.0:all". The attributes that are skipped on this level are marked with a

994

995
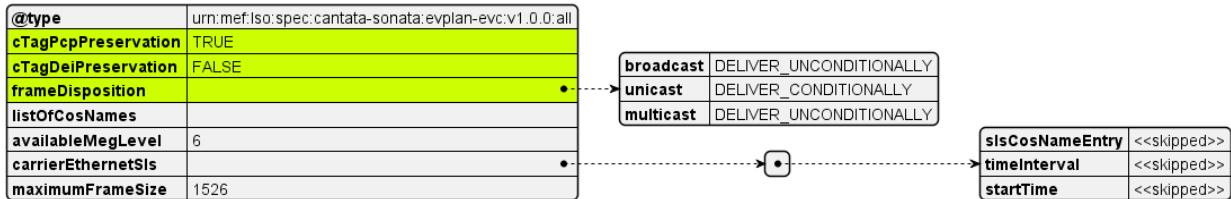
996 <p align="center">**Figure A2-24 – UC2d: EVP-LAN basic attributes**</p>

997 The structures defining the "evcEndPoint" are complex and presented in the following figures:

998
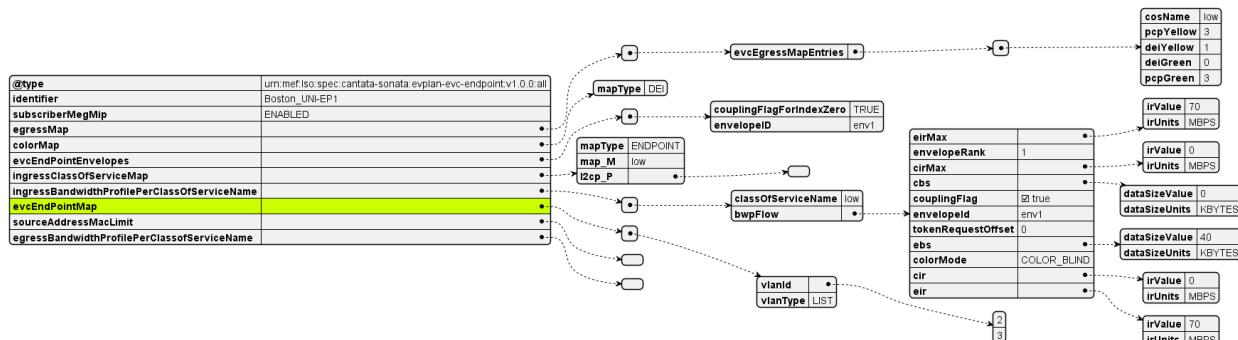
999 <p align="center">**Figure A2-25 – UC2d: EVP-LAN Endpoint (evcEndPoint)**</p>

1000 POQ Request example:

```
1001  {
1002    "instantSyncQualification": true,
1003    "externalId": "BuyerPoq-00001",
1004    "provideAlternative": false,
1005    "projectId": "BuyerProjectX",
1006    "relatedContactInformation": [
1007      {
1008        "emailAddress": "john.example@buyer.com",
1009        "name": "John Example",
1010        "number": "12-345-6789",
1011        "numberExtension": "1234",
1012        "organization": "Buyer Co.",
1013        "role": "buyerContactInformation"
1014      }
1015    ],
1016    "productOfferingQualificationItem": [
1017      {
1018        "id": "item-001",
1019        "action": "add",
```

```
1020          "product": {
1021            "productOffering": {
1022              "id": "000073"
1023            },
1024            "productConfiguration": {
1025              "@type": "urn:mef:lso:spec:cantata-sonata:evplan-evc:v1.0.0:all",
1026              "cTagPcpPreservation": "TRUE",
1027              "cTagDeiPreservation": "FALSE",
1028              "frameDisposition": {
1029                "broadcast": "DELIVER_UNCONDITIONALLY",
1030                "unicast": "DELIVER_CONDITIONALLY",
1031                "multicast": "DELIVER_UNCONDITIONALLY"
1032              },
1033              "listOfCosNames": ["low"],
1034              "availableMegLevel": "6",
1035              "carrierEthernetSls": [
1036                {
1037                  "slsCosNameEntry": [
1038                    {
1039                      "cosName": "low",
1040                      "consecutiveIntervalN": 3,
1041                      "deltaT": 10,
1042                      "thresholdC": 0.5,
1043                      "oneWayFrameDelayPmMetric": [
1044                        {
1045                          "oneWayFdObjective": {
1046                            "shortDurationValue": 100,
1047                            "shortDurationUnits": "MS"
1048                          },
1049                          "oneWayFdPercentile": 99.5,
1050                          "orderedPairList": [
1051                            {
1052                              "toCarrierEthernetServiceEndPoint": "Boston_UNI-EP1",
1053                              "fromCarrierEthernetServiceEndPoint": "Detroit_UNI-EP1"
1054                            },
1055                            {
1056                              "toCarrierEthernetServiceEndPoint": "Detroit_UNI-EP1",
1057                              "fromCarrierEthernetServiceEndPoint": "Boston_UNI-EP1"
1058                            }
1059                          ]
1060                        }
1061                      ],
1062                      "oneWayMeanFrameDelayPmMetric": [],
1063                      "oneWayFrameDelayRangePmMetric": [],
1064                      "oneWayInterFrameDelayVariationPmMetric": [],
1065                      "oneWayFrameLossRatioPmMetric": [
```

```
1066                              {
1067                                  "oneWayFlrObjective": 0.001,
1068                                  "orderedPairList": [
1069                                    {
1070                                        "toCarrierEthernetServiceEndPoint": "Boston_UNI-EP1",
1071                                        "fromCarrierEthernetServiceEndPoint": "Detroit_UNI-EP1"
1072                                    },
1073                                    {
1074                                        "toCarrierEthernetServiceEndPoint": "Detroit_UNI-EP1",
1075                                        "fromCarrierEthernetServiceEndPoint": "Boston_UNI-EP1"
1076                                    }
1077                                  ]
1078                              }
1079                            ],
1080                            "oneWayAvailabilityPmMetric": [],
1081                            "oneWayHighLossIntervalsPmMetric": [],
1082                            "oneWayConsecutiveHighLossIntervalsPmMetric": [],
1083                            "oneWayCompositePmMetric": [],
1084                            "oneWayGroupAvailabilityPmMetric": []
1085                          }
1086                        ],
1087                        "timeDuration": {
1088                          "timeDurationValue": 1,
1089                          "timeDurationUnits": "MONTH"
1090                        },
1091                        "startTime": "2022-10-12T00:00:00Z"
1092                      }
1093                    ],
1094                    "maximumFrameSize": 1522
1095                  }
1096                }
1097             },
1098             {
1099               "id": "item-002",
1100               "action": "add",
1101               "product": {
1102                 "productOffering": {
1103                   "id": "000074"
1104                 },
1105                 "place": [
1106                   {
1107                     "@type": "GeographicAddressRef",
1108                     "id": "BostonAddress-id-1",
1109                     "role": "INSTALL_LOCATION"
1110                   }
1111                 ],
```

```
1112          "relatedContactInformation": [
1113            {
1114              "number": "+12-345-678-90",
1115              "emailAddress": "LocationContact@example.com",
1116              "role": "locationContact",
1117              "name": "Location Contact"
1118            }
1119          ],
1120          "productConfiguration": {
1121            "@type": "urn:mef:lso:spec:sonata:carrier-ethernet-subscriber-
1122    uni:v1.0.0:all",
1123            "listOfPhyLinks": [
1124              {
1125                "uniConnectorGender": "SOCKET",
1126                "autoNegotiation": "ENABLED",
1127                "synchronousEthernet": "ENABLED",
1128                "uniConnectorType": "RJ45",
1129                "precisionTiming": "DISABLED"
1130              }
1131            ],
1132            "virtualFrameMap": [],
1133            "portConversation": [],
1134            "maximumNumberOfEndPoint": 6,
1135            "lagLinkMeg": "ENABLED",
1136            "l2cpAddressSet": "CTB",
1137            "linkOam": "DISABLED",
1138            "meg": "ENABLED",
1139            "linkAggregation": "NONE",
1140            "l2cpPeering": [],
1141            "maximumNumberOfCtagVlanIdsPerEndPoint": 4094,
1142            "tokenShare": "ENABLED",
1143            "maximumServiceFrameSize": 1522,
1144            "envelopes": [
1145              {
1146                "couplingFlagForIndexZero": "TRUE",
1147                "envelopeID": "env1"
1148              },
1149              {
1150                "couplingFlagForIndexZero": "TRUE",
1151                "envelopeID": "env2"
1152              }
1153            ],
1154            "instantiation": "PHYSICAL"
1155          }
1156        }
1157      },
```

```
1158           {
1159               "id": "item-006",
1160               "action": "add",
1161               "qualificationItemRelationship": [
1162                  {
1163                      "relationshipType": "ENDPOINT_OF_EVC",
1164                      "id": "item-001"
1165                  },
1166                  {
1167                      "relationshipType": "CONNECTS_TO_UNI",
1168                      "id": "item-002"
1169                  }
1170               ],
1171               "product": {
1172                 "productOffering": {
1173                     "id": "000075"
1174                 },
1175                 "productConfiguration": {
1176                     "@type": "urn:mef:lso:spec:cantata-sonata:evplan-evc-end-
1177        point:v1.0.0:all",
1178                     "identifier": "Boston_UNI-EP1",
1179                     "subscriberMegMip": "ENABLED",
1180                     "egressMap": [
1181                        {
1182                            "evcEgressMapEntries": [
1183                                {
1184                                    "cosName": "low",
1185                                    "pcpYellow": 3,
1186                                    "deiYellow": 1,
1187                                    "deiGreen": 0,
1188                                    "pcpGreen": 3
1189                                }
1190                            ]
1191                        }
1192                     ],
1193                     "colorMap": {
1194                         "mapType": "DEI"
1195                     },
1196                     "evcEndPointEnvelopes": [
1197                        {
1198                            "couplingFlagForIndexZero": "TRUE",
1199                            "envelopeID": "env1"
1200                        }
1201                     ],
1202                     "ingressClassOfServiceMap": {
1203                         "mapType": "ENDPOINT",
```

```
1204                    "map_M": "low",
1205                    "l2cp_P": []
1206                },
1207            "ingressBandwidthProfilePerClassOfServiceName": [
1208                {
1209                    "classOfServiceName": "low",
1210                    "bwpFlow": {
1211                        "eirMax": {
1212                            "irValue": 70,
1213                            "irUnits": "MBPS"
1214                        },
1215                        "envelopeRank": 1,
1216                        "cirMax": {
1217                            "irValue": 0,
1218                            "irUnits": "MBPS"
1219                        },
1220                        "cbs": {
1221                            "dataSizeValue": 0,
1222                            "dataSizeUnits": "KBYTES"
1223                        },
1224                        "couplingFlag": true,
1225                        "envelopeId": "env1",
1226                        "tokenRequestOffset": 0,
1227                        "ebs": {
1228                            "dataSizeValue": 40,
1229                            "dataSizeUnits": "KBYTES"
1230                        },
1231                        "colorMode": "COLOR_BLIND",
1232                        "cir": {
1233                            "irValue": 0,
1234                            "irUnits": "MBPS"
1235                        },
1236                        "eir": {
1237                            "irValue": 70,
1238                            "irUnits": "MBPS"
1239                        }
1240                    }
1241                }
1242            ],
1243            "evcEndPointMap": [
1244                {
1245                    "vlanId": [2, 3],
1246                    "vlanType": "LIST"
1247                }
1248            ],
1249            "sourceAddressMacLimit": [],
```

Page 41

```
1250              "egressBandwidthProfilePerClassofServiceName": []
1251            }
1252          }
1253        }
1254      ]
1255  }
1256


1257
```

1259   Detailed description of POQ use case is located in A.3.2 part. This section will describe the unique
1260   features of the particular technology.

1261   It is very important to understand the pattern of integrating the product configuration (so-called
1262   "payload") with the functional product-agnostic API ("envelope"). As explained in chapter MEF
1263   125 [3] the EP-TREE product model is composed of 2 elements (products):
1264   - the EP-TREE itself.

1265   - the UNI

1266   - the ENDPOINT (evcEndPoint)

1267   Additionally, EP-TREE has 2 types of ENDPOINT (evcEndPoint):

1268   - ROOT

1269   - LEAF

1270   The information about one single product is carried within the Product Offering Qualification
1271   (POQ) API by a single "productOfferingQualificationItem" being a subject to qualification. One
1272   POQ Request can carry more than one POQ Items, that may or may not be related to each other.
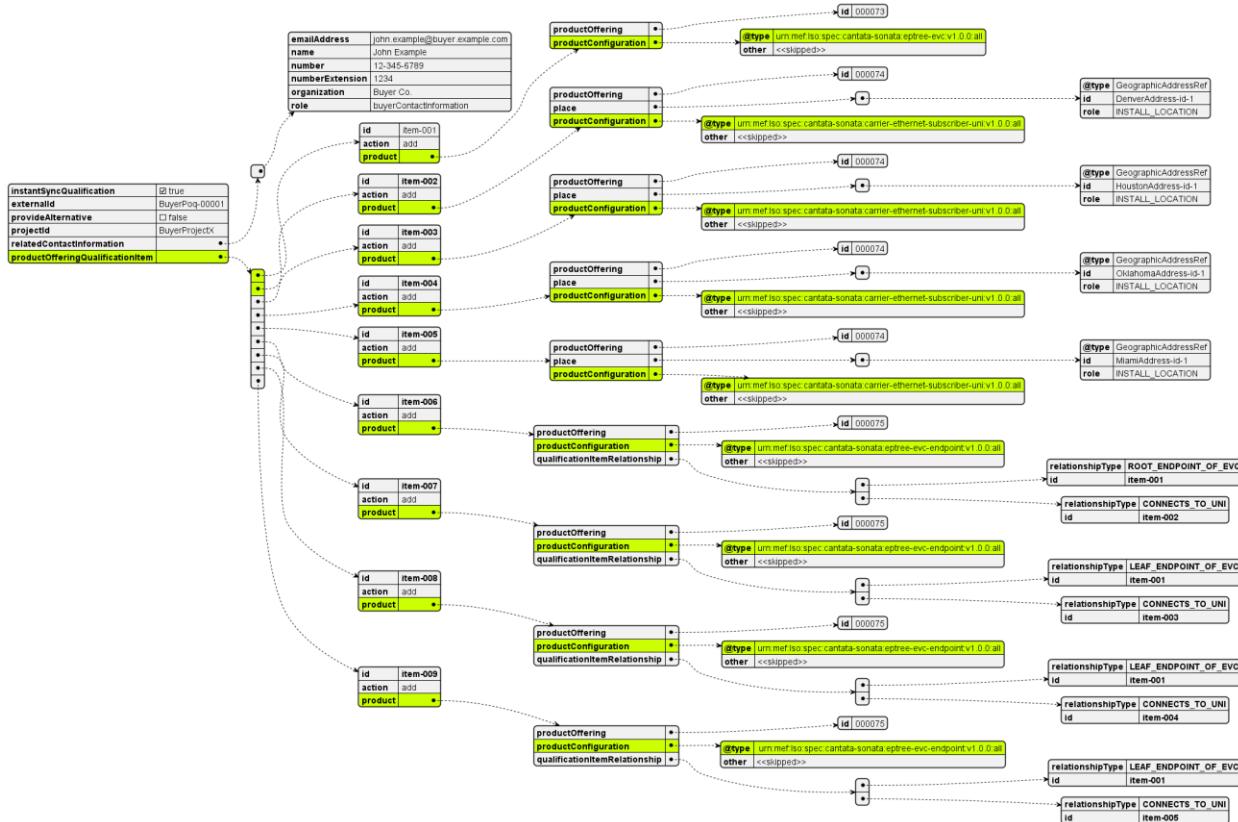
1273   In this use case, both the EP-TREE (EVC) and the UNI products are created or, to be more precise,
1274   a request to qualify if the creation of both of them is possible. Since 9 products are being subject
1275   to qualification, the POQ request contains 9 items with "action=add". The EP-TREE POQ Item
1276   has 8 relations:
1277   - to the UNIs, which is being qualified in the same request – by
1278   "productOfferingQualificationItem.qualificationItemRelationship"

1279   - to the ENDPOINTs (evcEndPoint), which is being qualified in the same request – by
1280   "productOfferingQualificationItem.qualificationItemRelationship

1281   An instance diagram in Figure A2-26 shows an extracted part from the request, to present the most
1282   important integration-related attributes. The product configuration attached to a POQ request is
1283   highlighted with green color, and the product relations are highlighted with a bold font.

1284
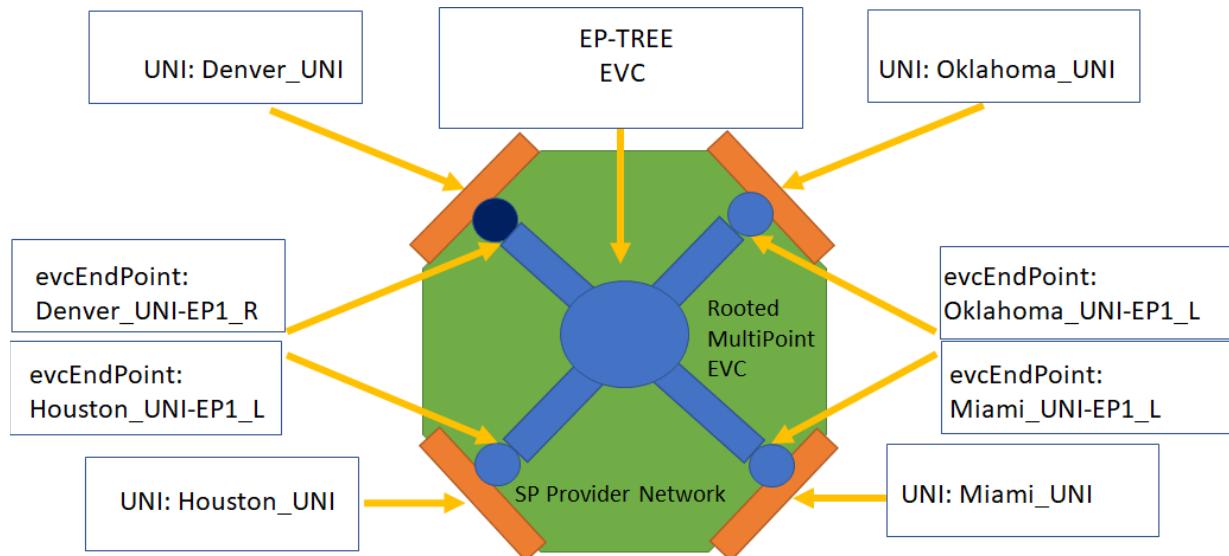
1285

1286

Figure A2-26 – UC2e: POQ Request, envelope part

1287 The products' integration with the API is covered. Let's go to some details of the products' con-
1288 figuration. The setup of the Use Case 2e is presented in Figure A2-27.

1289

1290

Figure A2-27 – UC2e: EP-TEE Setup Diagram

1291 This setup involves:
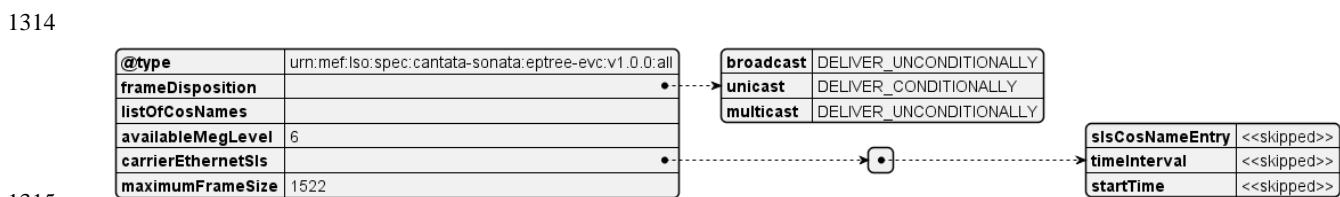
1292   •   Creation of the EP-TREE,

1293   •   Creation of the UNIs

1294   o   place: Denver (Address id acquired in Use Case 1)

1295   o   id="Denver_UNI"

1296   o   place: Houston

1297   o   id="Houston_UNI"

1298   o   place: Oklahoma

1299   o   id="Oklahoma_UNI"

1300   o   place: Miami

1301   o   id=" Miami _UNI"

1302   •   Creation of the ENDPOINTs (evcEndPoint):

1303   o   id=" **Denver_UNI-EP1_R**"

1304   o   id=" Houston _UNI-EP1_L"

1305   o   id=" Oklahoma_UNI-EP1_L"

1306   o   id=" Miami_UNI-EP1_L"

1307 The diagram aggregates the scope of a particular product configuration into rectangles. However,
1308 unlike to EPL configuration, the ENDPOINTs (evcEndPoint) are separated products.
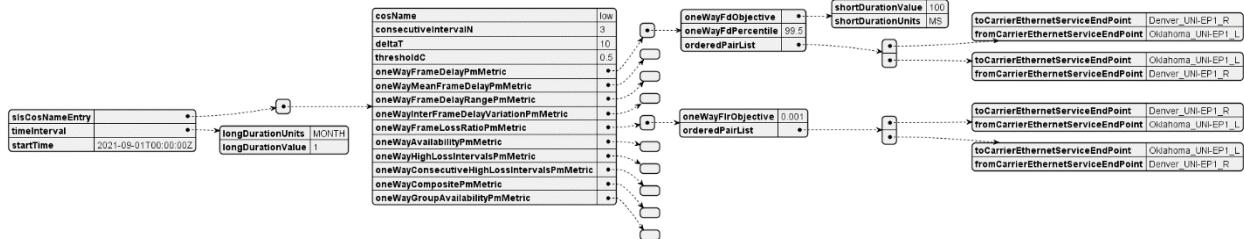
1309 The instance diagram for the whole EP-TREE, configuration is too big to be presented as a whole
1310 so it is split and presented in parts. Figure A2-28 shows the basic EP-TREE attributes. This dia-
1311 gram as attached to Figure A2-26 as the node with "@type=urn:mef:lso:spec:cantata-sonata:ep-
1312 tree-evc:v1.0.0:all". The attributes that are skipped on this level are marked with a "<<skipped>>"
1313 label and will be presented on the next diagrams.

1314

@type	urn:mef:lso:spec:cantata-sonata:eptree-evc:v1.0.0:all		broadcast	DELIVER_UNCONDITIONALLY
frameDisposition		•	unicast	DELIVER_CONDITIONALLY
listOfCosNames			multicast	DELIVER_UNCONDITIONALLY
availableMegLevel	6			
carrierEthernetSls		•		
maximumFrameSize	1522			

slsCosNameEntry	<<skipped>>
timeInterval	<<skipped>>
startTime	<<skipped>>

1315

1316 **Figure A2-28 – UC2e: EP-TREE basic attributes**

1317 The structures defining the "carrierEthernetSls" and "evcEndPoint" are complex and presented in
1318 the following figures:
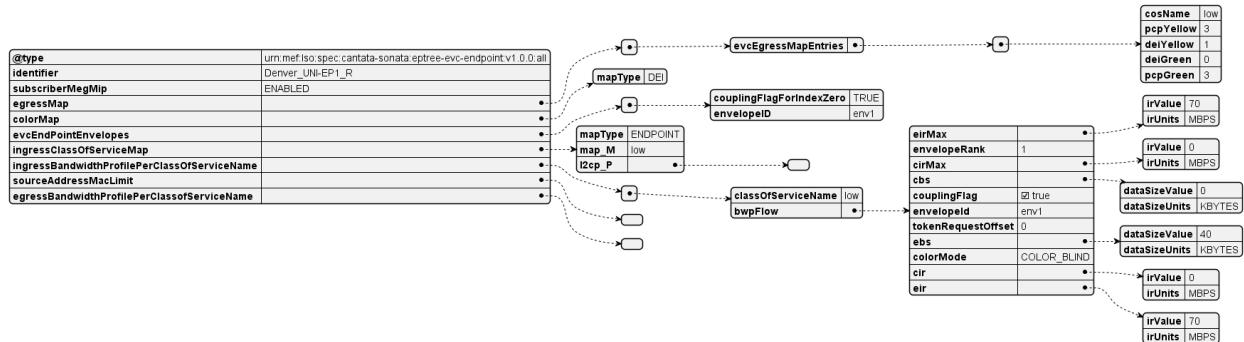
1319

1320
Figure A2-29 – UC2a: EP-TREE Carrier Ethernet SLS

1321

1322
Figure A2-30 – UC2e: EP-TREE ENDPOINT (evcEndPoint)

1323
A.3.7   Use Case 2f: POQ - new EVP-TREE, new UNIs, new ENDPOINTs low class of service

1324 Detailed description of POQ use case is located in A.3.2 part. This section will describe the unique
1325 features of the EVP-TREE technology.

1326 EP-TREE and EVP-TREE are very similar technologies. However, there are a few differences at
1327 the connection attribute level. They will be highlighted in the following diagrams.

1328 It is very important to understand the pattern of integrating the product configuration (so-called
1329 "payload") with the functional product-agnostic API ("envelope"). As explained in chapter MEF
1330 125 [3], the EVP-TREE product model is composed of 2 elements (products):
1331   • the EVP-TREE itself.

1332   • the UNI

1333   • the ENDPOINT (evcEndPoint)
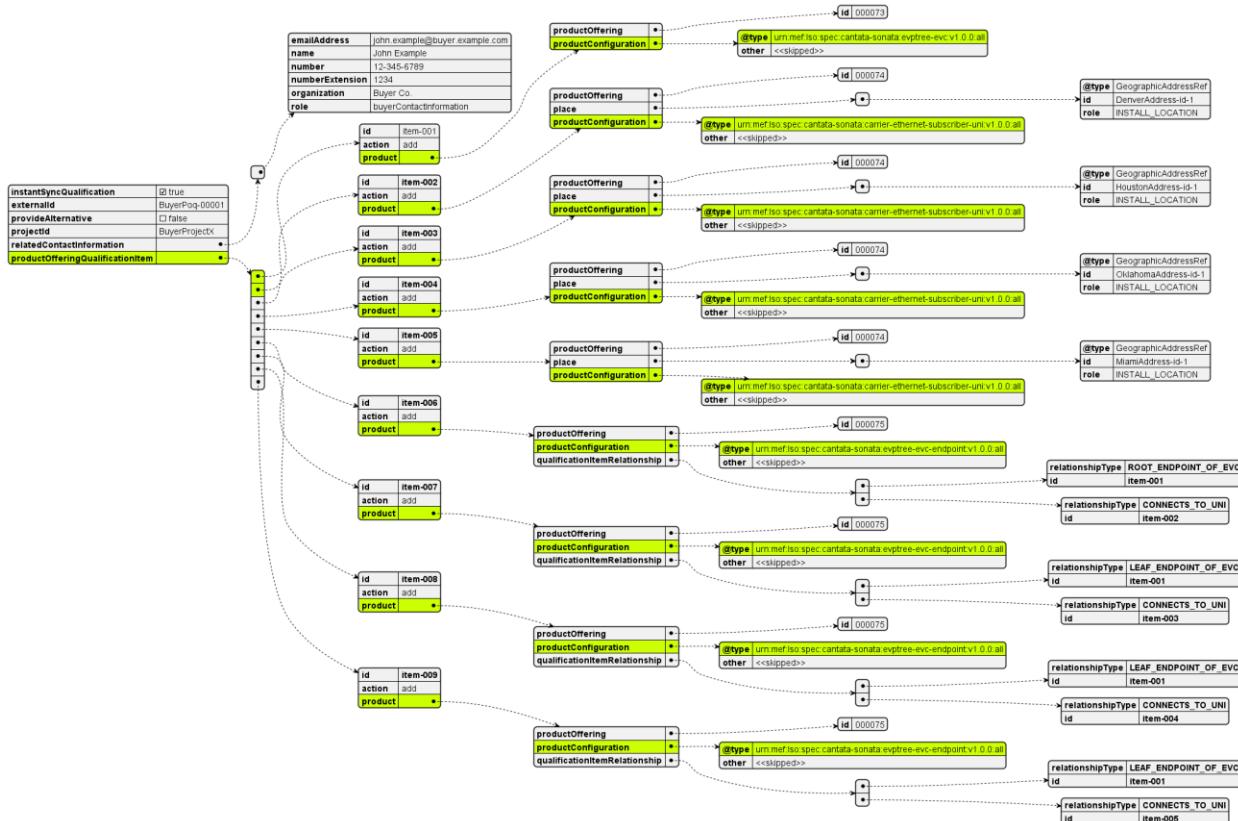
1334 The same as EP-TREE, EVP-TREE has 2 types of ENDPOINT (evcEndPoint):

1335   • ROOT

1336   • LEAF

1337 An instance diagram in Figure A2-31shows an extracted part from the request, to present the most
1338 important integration-related attributes. The product configuration attached to a POQ request is
1339 highlighted with green color, and the product relations are highlighted with a bold font.

1340

Figure A2-31 – UC2f: POQ Request, envelope part

The products' integration with the API is covered. Let's go to some details of the products' configuration. The setup of the Use Case 2f is presented in Figure A2-32.
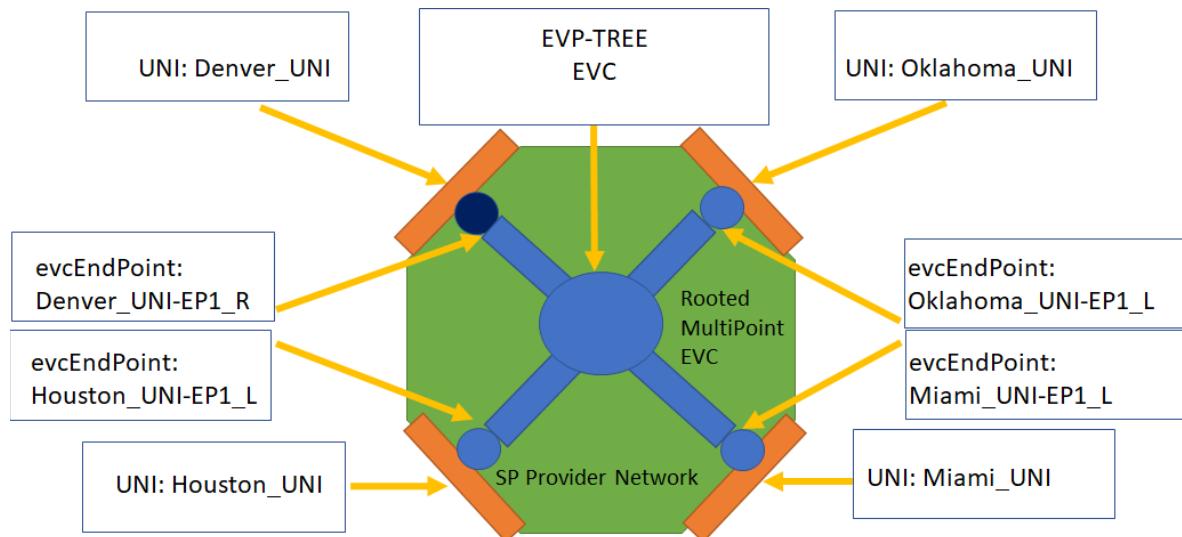
1344

Figure A2-32 – UC2f: EVP-TREE Setup Diagram

This setup involves:
- Creation of the EVP-TREE,

1348   •   Creation of the UNIs

1349   o   place: Denver (Address id acquired in Use Case 1)

1350   o   id=”Denver_UNI”

1351   o   place: Houston

1352   o   id=”Houston_UNI”

1353   o   place: Oklahoma

1354   o   id=”Oklahoma _UNI”

1355   o   place: Miami

1356   o   id=” Miami _UNI”

1357   •   Creation of the ENDPOINTs (evcEndPoint):

1358   o   **id=” Denver_UNI -EP1_R”**

1359   o   id=” Houston _UNI-EP1_L”

1360   o   id=” Oklahoma _UNI-EP1_L”

1361   o   id=” Miami _UNI-EP1_L”

1362 The diagram aggregates the scope of a particular product configuration into rectangles. However,
1363 unlike to EPL configuration, the EVP-TREE ENDPOINTs (evcEndPoint) are separated products.
1364 The instance diagram for the whole EVP-TREE configuration is too big to be presented as a whole
1365 so it is split and presented in parts. Figure A2-33 shows the basic EVP-TREE attributes. This
1366 diagram as attached to Figure A2-31 as the node with "@type=urn:mef:lso:spec:cantata-so-
1367 nata:evptree-evc:v1.0.0:all". The attributes that are skipped on this level are marked with a
1368 "<<skipped>>" label and will be presented on the next diagrams.

1369

1370
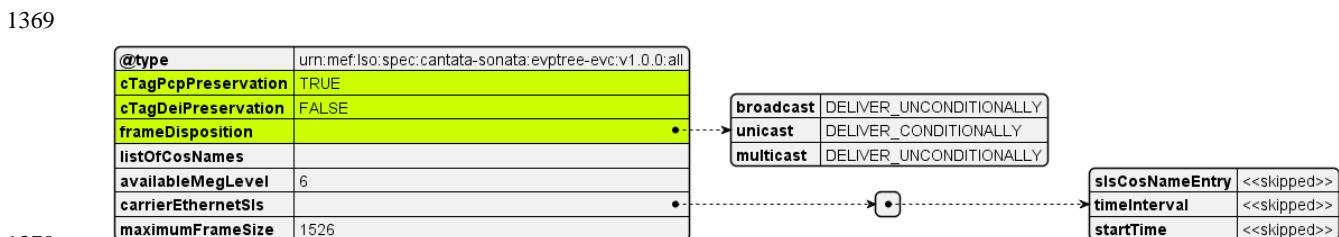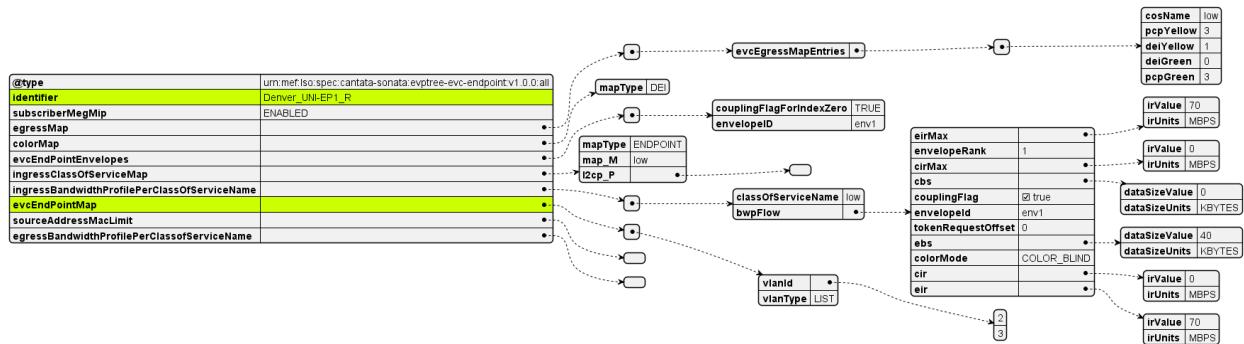
1371 **Figure A2-33 – UC2f: EVP-TREE basic attributes**

1372 The structures defining the ENDPOINT (evcEndPoint) are complex and presented in the following
1373 figures:

1374

1375

1376 <p style="text-align:center">Figure A2-34 – UC2f: EVP-TREE ENDPOINT (evcEndPoint)</p>

1377 POQ Request example:
1378 {
1379   "instantSyncQualification": true,
1380   "externalId": "BuyerPoq-00001",
1381   "provideAlternative": false,
1382   "projectId": "BuyerProjectX",
1383   "relatedContactInformation": [
1384   {
1385   "emailAddress": "john.example@buyer.com",
1386   "name": "John Example",
1387   "number": "12-345-6789",
1388   "numberExtension": "1234",
1389   "organization": "Buyer Co.",
1390   "role": "buyerContactInformation"
1391   }
1392   ],
1393   "productOfferingQualificationItem": [
1394   {
1395   "id": "item-001",
1396   "action": "add",
1397   "product": {
1398   "productOffering": {
1399   "id": "000073"
1400   },
1401   "productConfiguration": {
1402   "@type": "urn:mef:lso:spec:cantata-sonata:evptree-evc:v1.0.0:all",
1403   "cTagPcpPreservation": "TRUE",
1404   "cTagDeiPreservation": "FALSE",
1405   "frameDisposition": {
1406   "broadcast": "DELIVER_UNCONDITIONALLY",
1407   "unicast": "DELIVER_CONDITIONALLY",
1408   "multicast": "DELIVER_UNCONDITIONALLY"
1409   },
1410   "listOfCosNames": ["low"],

```
1411              "availableMegLevel": "6",
1412              "carrierEthernetSls": [
1413                {
1414                   "slsCosNameEntry": [
1415                     {
1416                        "cosName": "low",
1417                        "consecutiveIntervalN": 3,
1418                        "deltaT": 10,
1419                        "thresholdC": 0.5,
1420                        "oneWayFrameDelayPmMetric": [
1421                          {
1422                             "oneWayFdObjective": {
1423                                "shortDurationValue": 100,
1424                                "shortDurationUnits": "MS"
1425                             },
1426                             "oneWayFdPercentile": 99.5,
1427                             "orderedPairList": [
1428                               {
1429                                  "toCarrierEthernetServiceEndPoint": "Denver_UNI-EP1_R",
1430                                  "fromCarrierEthernetServiceEndPoint": "Oklahoma_UNI-
1431      EP1_L"
1432                               },
1433                               {
1434                                  "toCarrierEthernetServiceEndPoint": "Oklahoma_UNI-
1435      EP1_L",
1436                                  "fromCarrierEthernetServiceEndPoint": "Denver_UNI-
1437      EP1_R"
1438                               }
1439                             ]
1440                          }
1441                        ],
1442                        "oneWayMeanFrameDelayPmMetric": [],
1443                        "oneWayFrameDelayRangePmMetric": [],
1444                        "oneWayInterFrameDelayVariationPmMetric": [],
1445                        "oneWayFrameLossRatioPmMetric": [
1446                          {
1447                             "oneWayFlrObjective": 0.001,
1448                             "orderedPairList": [
1449                               {
1450                                  "toCarrierEthernetServiceEndPoint": "Denver_UNI-EP1_R",
1451                                  "fromCarrierEthernetServiceEndPoint": "Oklahoma_UNI-
1452      EP1_L"
1453                               },
1454                               {
1455                                  "toCarrierEthernetServiceEndPoint": "Oklahoma_UNI-
1456      EP1_L",
```

```
1457                              "fromCarrierEthernetServiceEndPoint": "Denver_UNI-
1458     EP1_R"
1459                              }
1460                          ]
1461                      }
1462                  ],
1463                  "oneWayAvailabilityPmMetric": [],
1464                  "oneWayHighLossIntervalsPmMetric": [],
1465                  "oneWayConsecutiveHighLossIntervalsPmMetric": [],
1466                  "oneWayCompositePmMetric": [],
1467                  "oneWayGroupAvailabilityPmMetric": []
1468              }
1469          ],
1470          "timeDuration": {
1471              "timeDurationValue": 1,
1472              "timeDurationUnits": "MONTH"
1473          },
1474          "startTime": "2022-10-12T00:00:00Z"
1475      }
1476      ],
1477      "maximumFrameSize": 1522
1478      }
1479      }
1480  },
1481  {
1482      "id": "item-002",
1483      "action": "add",
1484      "product": {
1485          "productOffering": {
1486              "id": "000074"
1487          },
1488          "place": [
1489              {
1490                  "@type": "GeographicAddressRef",
1491                  "id": "DenverAddress-id-1",
1492                  "role": "INSTALL_LOCATION"
1493              }
1494          ],
1495          "relatedContactInformation": [
1496              {
1497                  "number": "+12-345-678-90",
1498                  "emailAddress": "LocationContact@example.com",
1499                  "role": "locationContact",
1500                  "name": "Location Contact"
1501              }
1502          ],
```

```
1503            "productConfiguration": {
1504                "@type": "urn:mef:lso:spec:sonata:carrier-ethernet-subscriber-
1505        uni:v1.0.0:all",
1506                "listOfPhyLinks": [
1507                  {
1508                    "uniConnectorGender": "SOCKET",
1509                    "autoNegotiation": "ENABLED",
1510                    "synchronousEthernet": "ENABLED",
1511                    "uniConnectorType": "RJ45",
1512                    "precisionTiming": "DISABLED"
1513                  }
1514                ],
1515                "virtualFrameMap": [],
1516                "portConversation": [],
1517                "maximumNumberOfEndPoint": 6,
1518                "lagLinkMeg": "ENABLED",
1519                "l2cpAddressSet": "CTB",
1520                "linkOam": "DISABLED",
1521                "meg": "ENABLED",
1522                "linkAggregation": "NONE",
1523                "l2cpPeering": [],
1524                "maximumNumberOfCtagVlanIdsPerEndPoint": 4094,
1525                "tokenShare": "ENABLED",
1526                "maximumServiceFrameSize": 1522,
1527                "envelopes": [
1528                  {
1529                    "couplingFlagForIndexZero": "TRUE",
1530                    "envelopeID": "env1"
1531                  },
1532                  {
1533                    "couplingFlagForIndexZero": "TRUE",
1534                    "envelopeID": "env2"
1535                  }
1536                ],
1537                "instantiation": "PHYSICAL"
1538            }
1539          }
1540        },
1541        {
1542          "id": "item-006",
1543          "action": "add",
1544          "qualificationItemRelationship": [
1545            {
1546              "relationshipType": "ROOT_ENDPOINT_OF_EVC",
1547              "id": "item-001"
1548            },
```

```
1549              {
1550                  "relationshipType": "CONNECTS_TO_UNI",
1551                  "id": "item-002"
1552              }
1553          ],
1554          "product": {
1555              "productOffering": {
1556                  "id": "000075"
1557              },
1558              "productConfiguration": {
1559                  "@type": "urn:mef:lso:spec:cantata-sonata:evplan-evc-end-
1560      point:v1.0.0:all",
1561                  "identifier": "Denver_UNI-EP1_R",
1562                  "subscriberMegMip": "ENABLED",
1563                  "egressMap": [
1564                      {
1565                          "evcEgressMapEntries": [
1566                              {
1567                                  "cosName": "low",
1568                                  "pcpYellow": 3,
1569                                  "deiYellow": 1,
1570                                  "deiGreen": 0,
1571                                  "pcpGreen": 3
1572                              }
1573                          ]
1574                      }
1575                  ],
1576                  "colorMap": {
1577                      "mapType": "DEI"
1578                  },
1579                  "evcEndPointEnvelopes": [
1580                      {
1581                          "couplingFlagForIndexZero": "TRUE",
1582                          "envelopeID": "env1"
1583                      }
1584                  ],
1585                  "ingressClassOfServiceMap": {
1586                      "mapType": "ENDPOINT",
1587                      "map_M": "low",
1588                      "l2cp_P": []
1589                  },
1590                  "ingressBandwidthProfilePerClassOfServiceName": [
1591                      {
1592                          "classOfServiceName": "low",
1593                          "bwpFlow": {
1594                              "eirMax": {
```

```
1595                      "irValue": 70,
1596                      "irUnits": "MBPS"
1597                    },
1598                  "envelopeRank": 1,
1599                  "cirMax": {
1600                      "irValue": 0,
1601                      "irUnits": "MBPS"
1602                    },
1603                  "cbs": {
1604                      "dataSizeValue": 0,
1605                      "dataSizeUnits": "KBYTES"
1606                    },
1607                  "couplingFlag": true,
1608                  "envelopeId": "env1",
1609                  "tokenRequestOffset": 0,
1610                  "ebs": {
1611                      "dataSizeValue": 40,
1612                      "dataSizeUnits": "KBYTES"
1613                    },
1614                  "colorMode": "COLOR_BLIND",
1615                  "cir": {
1616                      "irValue": 0,
1617                      "irUnits": "MBPS"
1618                    },
1619                  "eir": {
1620                      "irValue": 70,
1621                      "irUnits": "MBPS"
1622                    }
1623                }
1624              }
1625            ],
1626          "evcEndPointMap": [
1627            {
1628                "vlanId": [2, 3],
1629                "vlanType": "LIST"
1630            }
1631          ],
1632          "sourceAddressMacLimit": [],
1633          "egressBandwidthProfilePerClassofServiceName": []
1634        }
1635      }
1636    }
1637  ]
1638 }
1639
```

1640
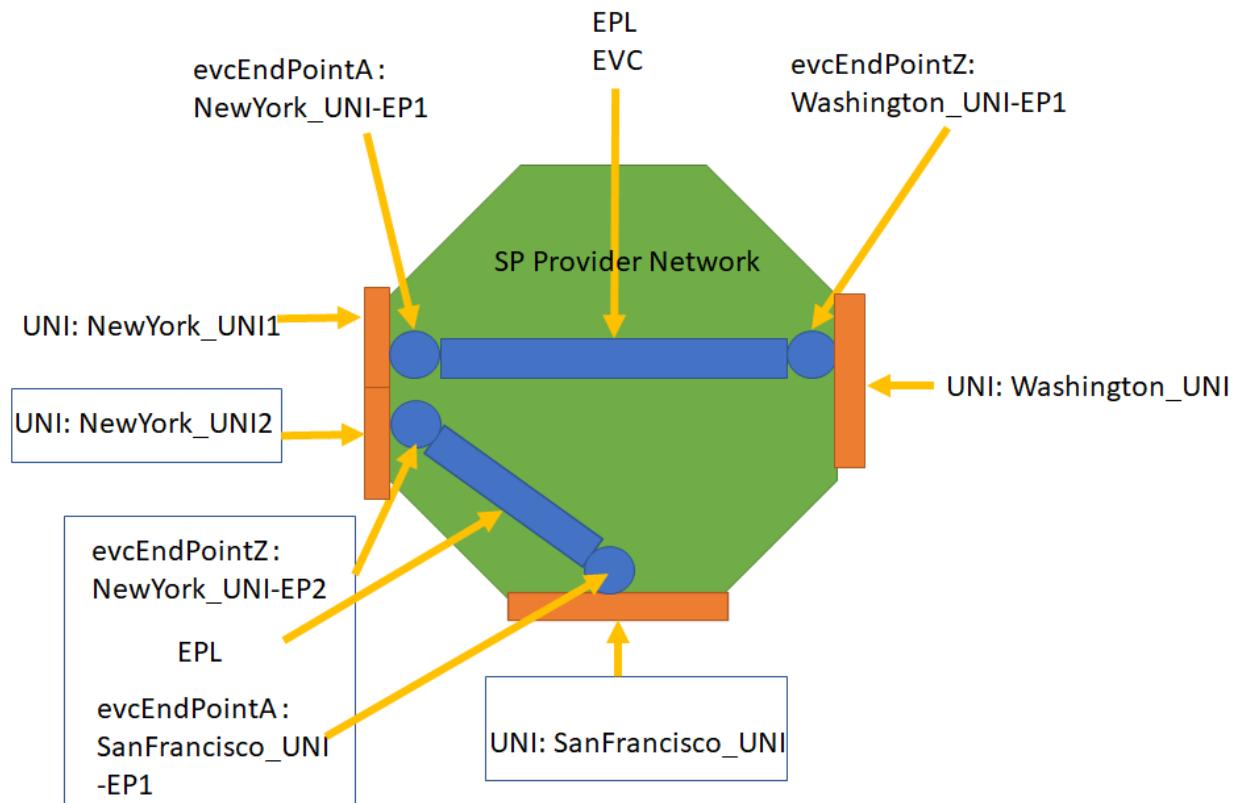
A.3.8   Use Case 3a: POQ - new EPL, new 2 UNIs, low class of service

1642   This use case presents the same step (POQ) as Use Case 2a but with two differences:

1643   • 2 new UNI are created

1644   • a more complex configuration of an EPL is presented.

1645   The setup of the Use Case 3a is presented in Figure A2-35:

1646

1647   **Figure A2-35 – UC3a: EPL – modified setup diagram**

1648   This setup involves:

1649   • Creation of 2 new UNI

1650   o configuration of a new UNI with id="NewYork_UNI2",

1651   o configuration of a new UNI with id="SanFrancisco_UNI",

1652   • Creation of the new EPL, including:

1653   o configuration of a new UNI Endpoint with id="NewYork_UNI-EP2", at the new
1654   UNI with id="NewYork_UNI2"

1655   ○ configuration of a new UNI Endpoint with id="SanFrancisco_UNI-EP1", at the
1656   new UNI with id="SanFrancisco_UNI"

1657   EPL with new configuration is presented on POQ request example in Figure A2-36:

1658
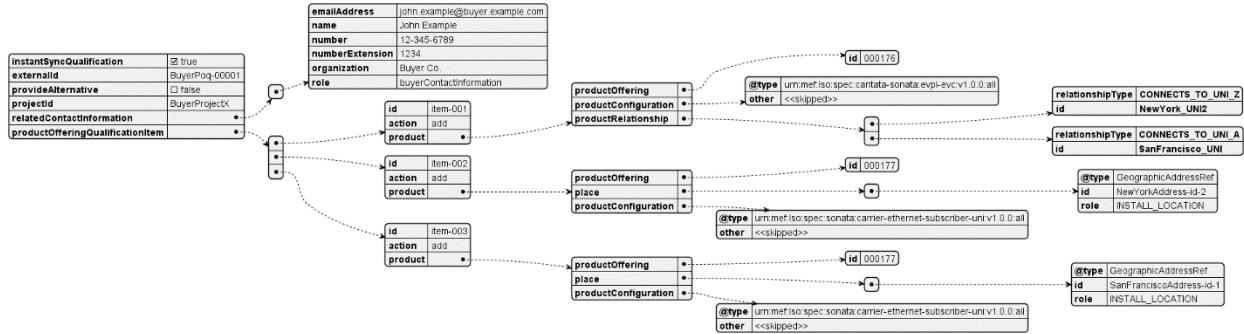
1659   **Figure A2-36 – UC3a: EPL relationships**

1660   **A.3.9 Use Case 3b: POQ - new EVPL, existing UNI and 1 new UNI, low class of service**

1661   This use case presents the same step (POQ) as Use Case 2b but with two differences:

1662   • an existing UNI is being used along with 1 new UNI created,

1663   • a more complex configuration of an EVPL is presented.

1664   The setup of the Use Case 3 is presented in Figure A2-37:

EVPL
EVC

evcEndPointA:
NewYork_UNI-EP1

evcEndPointZ:
Washington_UNI-EP1

SP Provider Network

UNI: NewYork_UNI

UNI: Washington_UNI

evcEndPointZ:
NewYork_UNI-EP2

EVPL

evcEndPointA:
SanFrancisco_UNI
-EP1

UNI: SanFrancisco_UNI

1665
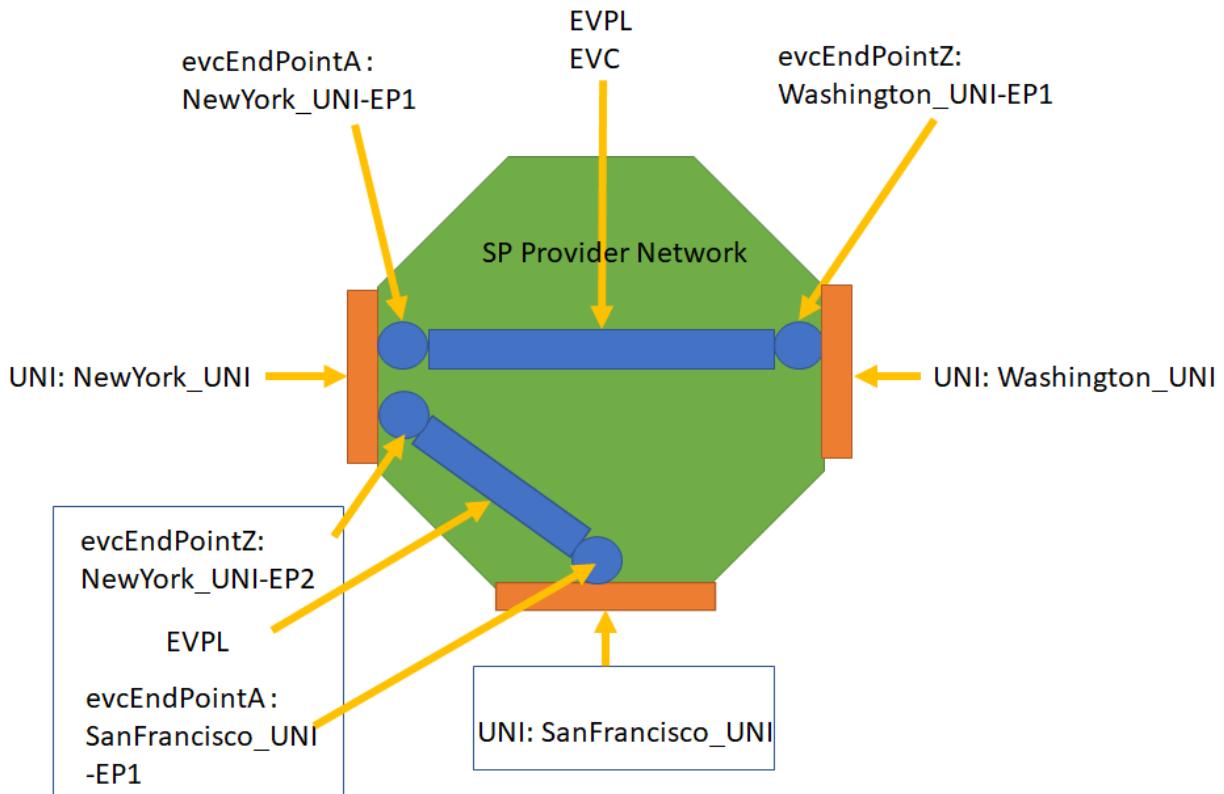
Figure A2-37 – UC3b: EVP – modified Setup diagram
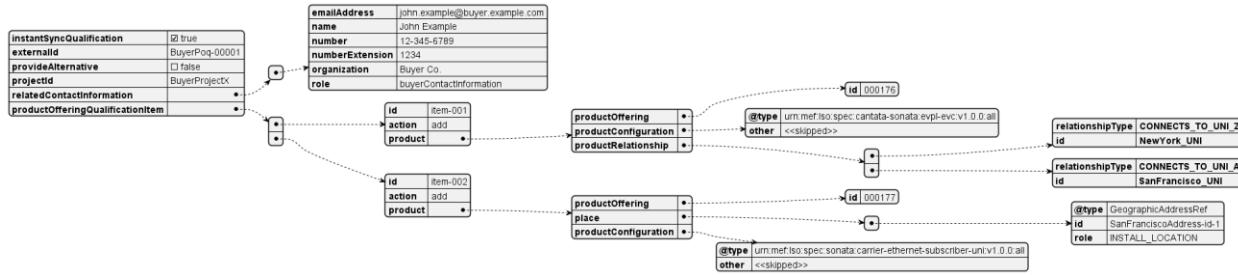
1667 This setup involves:

1668 • Creation of 1 new UNI

1669 o configuration of a new UNI with id="SanFrancisco_UNI",

1670 • Creation of the new EVPL, including:

1671 o configuration of a new UNI Endpoint with id="NewYork_UNI-EP2", at the al-
1672 ready existing UNI with id="NewYork_UNI", the one that was created in Use Case
1673 2 (assuming it was successfully ordered)

1674 o configuration of a new UNI Endpoint with id="SanFrancisco_UNI-EP1", at the
1675 new UNI with id="SanFrancisco_UNI"

1676 EPL with new configuration is presented on POQ request example in Figure A2-38:

1677
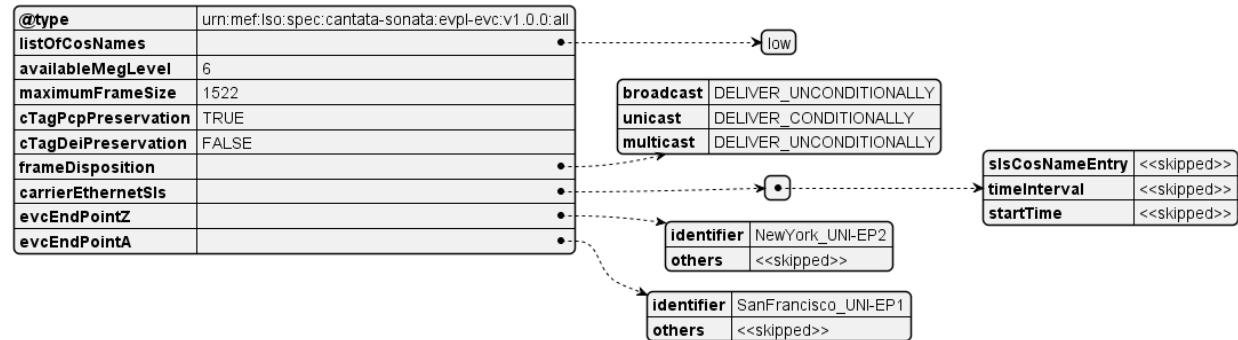
1678

1679 **Figure A2-38 – UC3b: EVPL relationships**

1680 The configuration of the UNI endpoint introduces a configuration with two classes of service: low
1681 and high. The difference is first noticed at the root EVPL type configuration by having a list in the
1682 "listOfCosNames" attribute:

1683
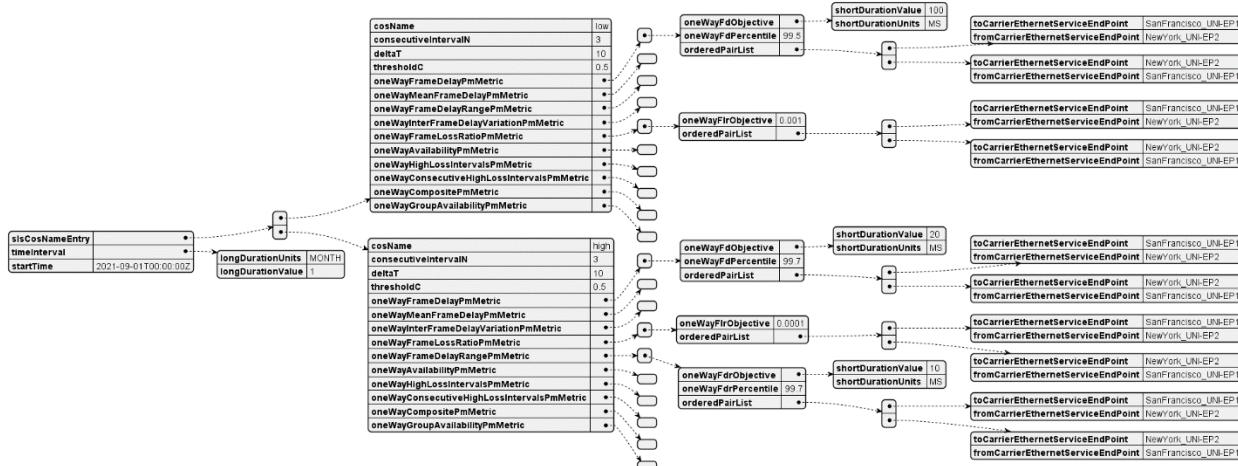
1684

1685 **Figure A2-39 – UC3b: EVPL basic attributes**

1686 Next, the "carrierEthernetSls" provides the Service Level Specification by defining the perfor-
1687 mance metrics per class of service. Notice the different metrics and values per low and high ones.

1688
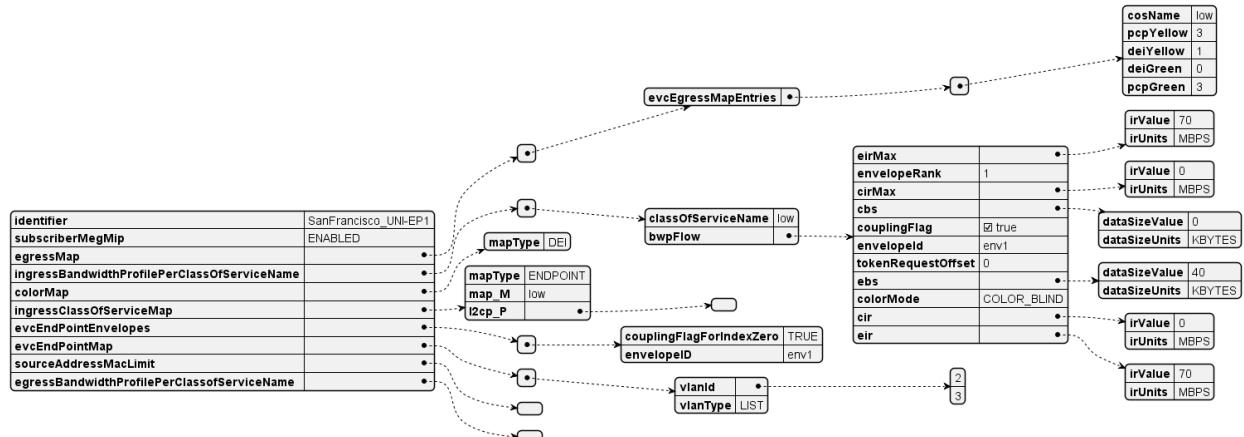
1689

1690 **Figure A2-40 – UC3b: EVPL Carrier Ethernet SLS**

1691

Figure A2-41 – UC3b: EVPL UNI Endpoint

1692

1693 The bandwidth profiles defined for both classes of service share the same envelope "env2" which
1694 means they share the same bandwidth "pipe". The "high" profile defines the "cirMax" and
1695 "eirMax" on the same level as corresponding "cir" and "eir" values, which is 50 MBPS. This
1696 means that the 70 MBPS bandwidth is guaranteed for this profile. The "low" profile defines the
1697 "cir" and "cirMax" on the level of 0 MBPS which means no bandwidth is guaranteed. The values
1698 of and the "eir" of 40 MBPS that a maximum of 40 MBPS can be used when free and, because
1699 "eirMax" is set to 70 MBPS, additional 30 MBPS from the "high" CoS reservation can be used,
1700 when the traffic is not utilizing the full guaranteed 50 MBPS.

1701 Note that the names of the classes of service – "low" and "high" are used several times across the
1702 payload and must match respectively in all of the places.

1703 The request example, as huge and repeating can be found in the attached postman collection.

1704 **A.3.10 Use Case 4a: Quote EPL**

1705 For detailed guidance on how to use the Quote Management API, please refer to MEF 115 [5].
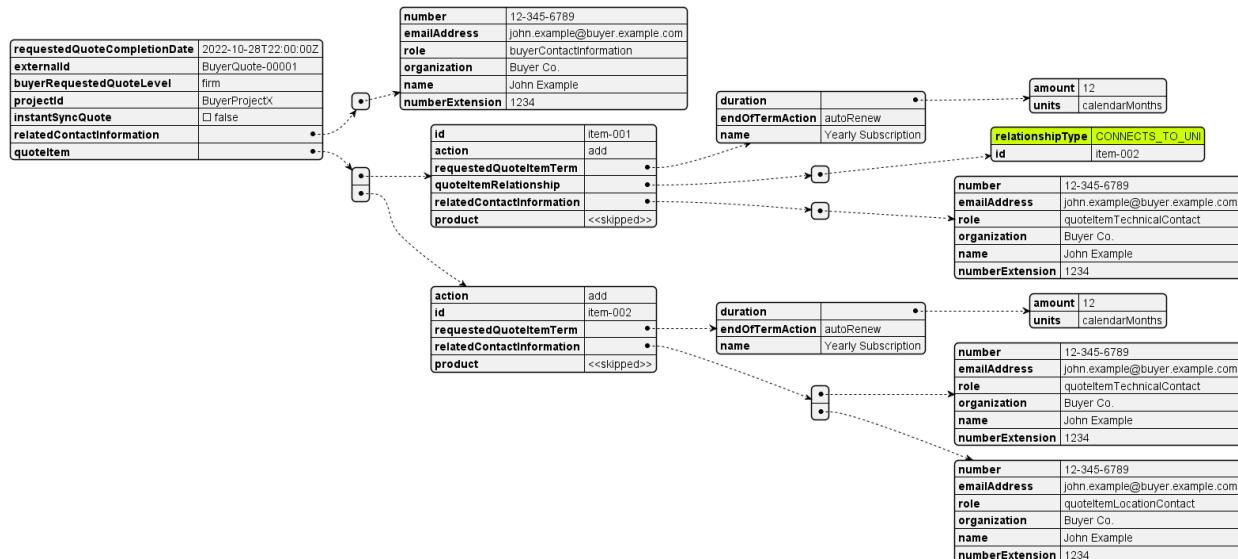
1706 The aim of the Quote step is to allow the Buyer to submit a request to find out how much the
1707 installation of an instance of a Product Offering, an update to an existing Product, or a disconnect
1708 of an existing Product will cost.

1709 This use case is the next step after use case 2. It asks for a quotation of the installation of the EPL
1710 and UNI products, with configuration as described in use case 2a (A.3.2).

1711 The Quote API carries product information exactly the same way as the POQ. The same steps in
1712 request building and rules of referencing existing products or ones in the same request, as described
1713 in section A.3.2, apply.

1714 Figure A2-42presents a diagram of a Quote request, with product information skipped.

1716 **Figure A2-42 – UC4a: EPL Quote Request**

1717 The most important attributes to set in the quote request are:

1718 "instantSyncQuote" – to state the preference of receiving an instant (synchronous) response or a
1719 deferred (asynchronous) one. In the latter case, the Seller only sends back an acknowledge re-
1720 sponse and proceeds with the quotation. The Buyer may choose to register for notification or per-
1721 form a periodical poll.

1722 "requestedQuoteCompletionDate" – If an instant response is not required this specifies the re-
1723 quested response time.

1724 "buyerRequestedQuoteLevel" - 3 different types of quotes are managed:

1725   - **Budgetary**: A Quote that is provided quickly and with very little analysis such that the
1726   Buyer can get an idea of how much the requested Product Offering could cost. Any
1727   charges specified are subject to change.

1728   - **Firm - Subject to Feasibility Check**: A Quote that is provided to the Buyer based on
1729   some, but not a complete, pre-order analysis. At this stage, the Seller may not be willing
1730   to perform any further work on the Quote and requests that the Buyer use the Firm –
1731   Subject to Feasibility Check Quote to proceed to the Order process. Ordering is possible
1732   based on the Firm – Subject to Feasibility Check Quote with some stipulations as to how
1733   cost identified during delivery is addressed. The Monthly Recurring Charges specified in
1734   the Quote Response are final. Non-Recurring Charges specified in the Quote Response
1735   are subject to change and new Non-Recurring Charges may be identified during fulfill-
1736   ment.

1737   - **Firm**: A Quote provided to the Buyer based on complete pre-order analysis. All Monthly
1738   Recurring Charges and Non-Recurring Charges specified on a Firm Quote are committed.
1739   A Firm Quote may expire at some date specified by the Seller.

1740 "requestedQuoteItemTerm" – to specify the term (also known as commitment)

1741 In the response, the Seller confirms (most likely) the "quoteLevel", "quoteItemTerm" and provides
1742 a price per each quote item. An example of price specification is shown below:

```
1743        "quoteItemPrice": [
1744            {
1745                "name": "Monthly Plan 25",
1746                "priceType": "recurring",
1747                "recurringChargePeriod": "month",
1748                "price": {
1749                    "taxRate": 16,
1750                    "dutyFreeAmount": {
1751                        "unit": "EUR",
1752                        "value": 25,
1753                    },
1754                    "taxIncludedAmount": {
1755                        "unit": "EUR",
1756                        "value": 29,
1757                    },
1758                },
1759            }
1760        ],
```

1761 Note: The Seller may require the Buyer to perform POQ prior to sending a Quote request.
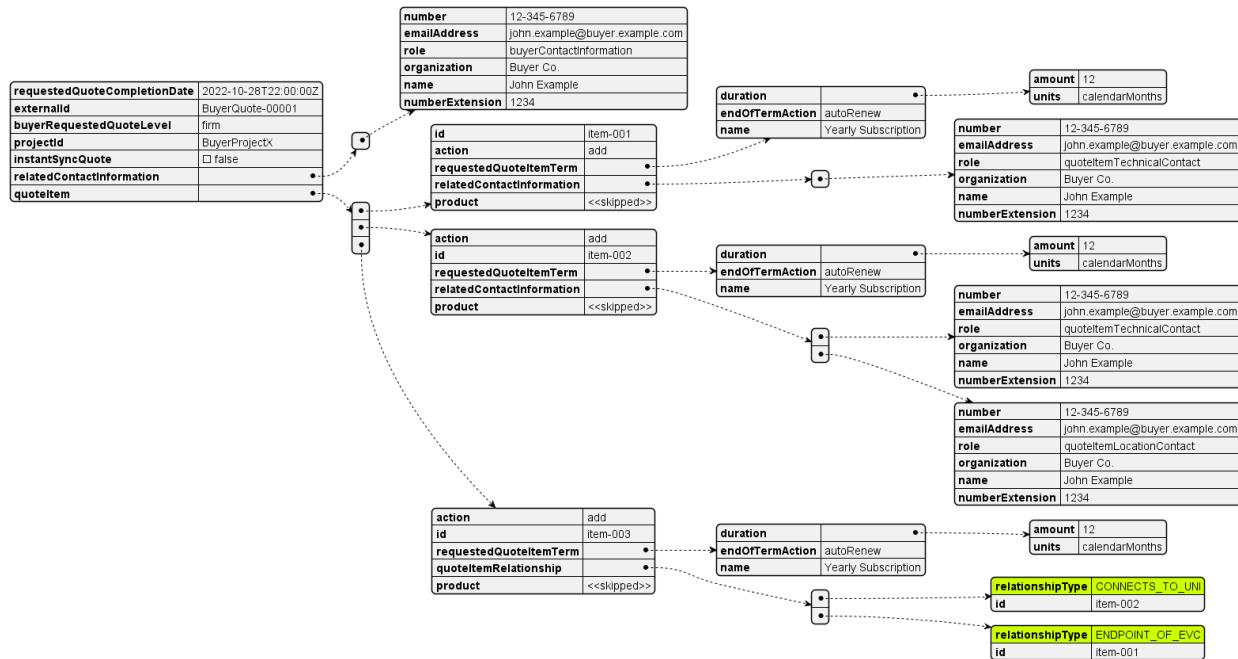
1762 **A.3.11 Use Case 4b: Quote EVP-LAN**

1763 Detailed description of "Quote" use case is located in A.3.10 part. This section will describe the
1764 unique features of the EVP-LAN technology.

1765 EVP-LAN has 2 additional products such as UNI and ENPOINT. These products are involved in
1766 this use case for EVP-LAN.

1767 Figure A2-43 presents a diagram of a Quote request, with product information skipped.

1768

Figure A2-43 – UC4b: EVP-LAN Quote Request

1769

A.3.12 Use Case 4c: Quote EP-TREE

1770

1771 Detailed description of "Quote" use case is located in A.3.10 part. This section will describe the
1772 unique features of the EP-TREE technology.
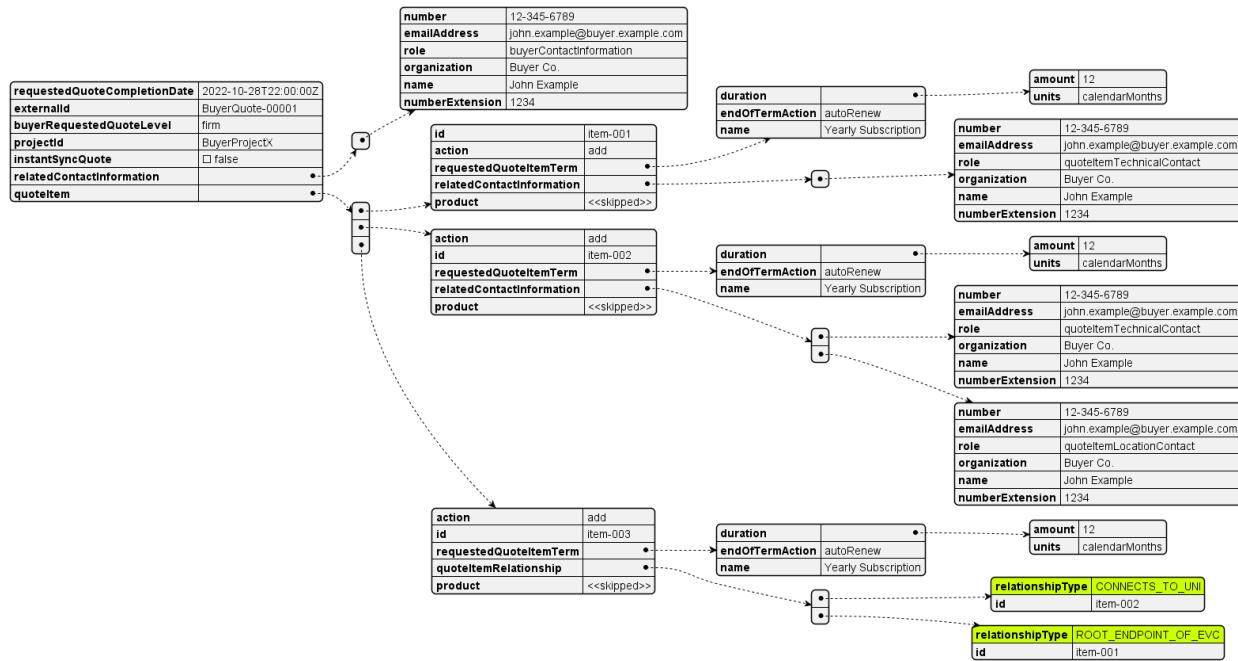
1773 EP-TREE has 2 additional products such as UNI and ENPOINT (evcEndPoint). These products
1774 are involved in this use case for EP-TREE.

1775 Additionally EP-TREE has 2 types of ENDPOINT (evcEndPoint):

1776 • ROOT

1777 • LEAF
1778

1779 Figure A2-44 presents a diagram of a Quote request, with product information skipped.

1780

<p style="text-align:center">**Figure A2-44 – UC4c:EP-TREE Quote Request**</p>

1782 ## A.3.13 Use Case 5a: Product Order EPL
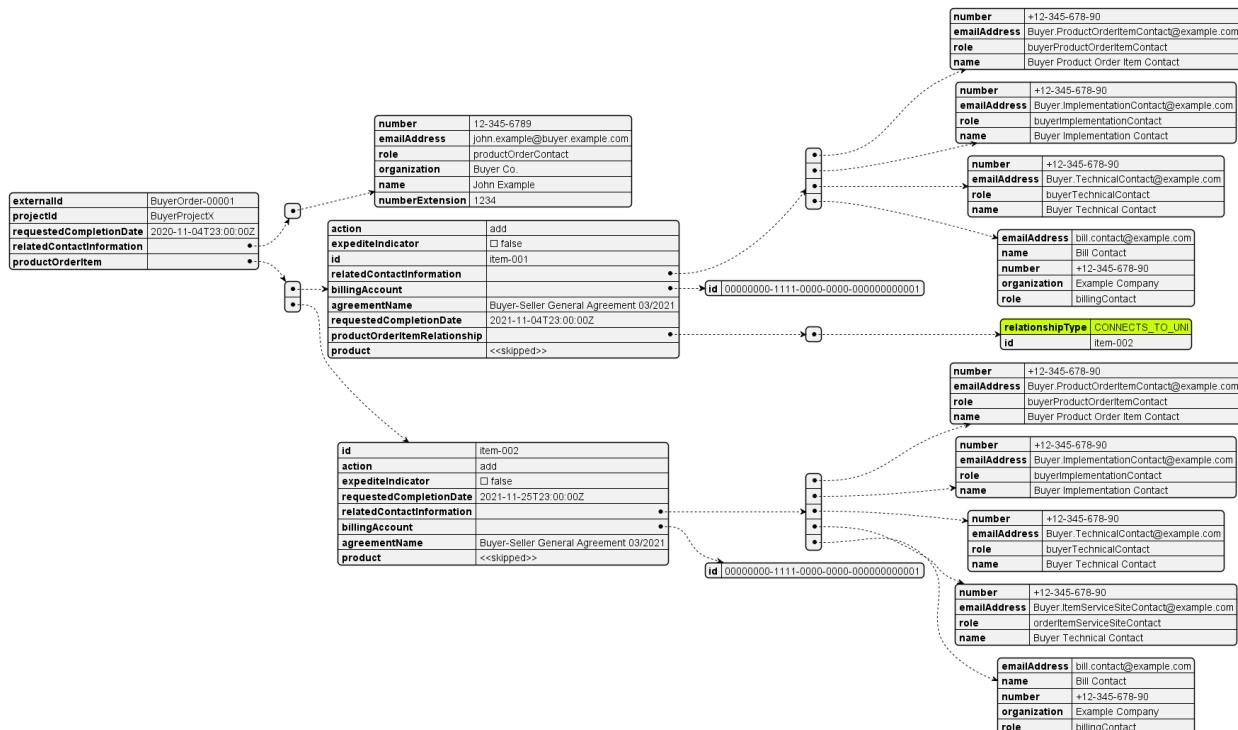
1783 Product Order allows the Buyer to request the Seller to initiate and complete the fulfillment process
1784 of an installation of a Product Offering, an update to an existing Product, or a disconnect of an
1785 existing Product at the address defined by the Buyer.

1786 This use case is the next step after use case 4a. It places an order for the installation of the EPL
1787 and UNI products, which were qualified and quoted in use cases 2 and 4.

1788 The Order API carries product information exactly the same way as the POQ and Quote. The same
1789 steps in request building and rules of referencing existing products or ones in the same request, as
1790 described in section A.3.2, apply.

1791 An example Product Order request can be found in the postman collection. Figure A2-45 presents
1792 it with product information skipped for readability.

1793

1794 **Figure A2-45 – UC5a: EPL Product Order request**

1795 Again, there are a few attributes to be set by the Seller in the request like "requestedComple-
1796 tionDate", "expediteIndicator" or "billingAccount" together with required contact information.

1797 The Seller responds with an acknowledge confirmation and then starts processing the order. The
1798 order fulfillment process is longer than a simple request-response one of the previous steps (POQ,
1799 Quote) and the state machine is more complex. The process may also be more interactive due to
1800 charge negotiation, possible request updates, etc.

1801 Product order API offers much more use cases like updating, expediting, or canceling an order
1802 request and additional charge negotiation.
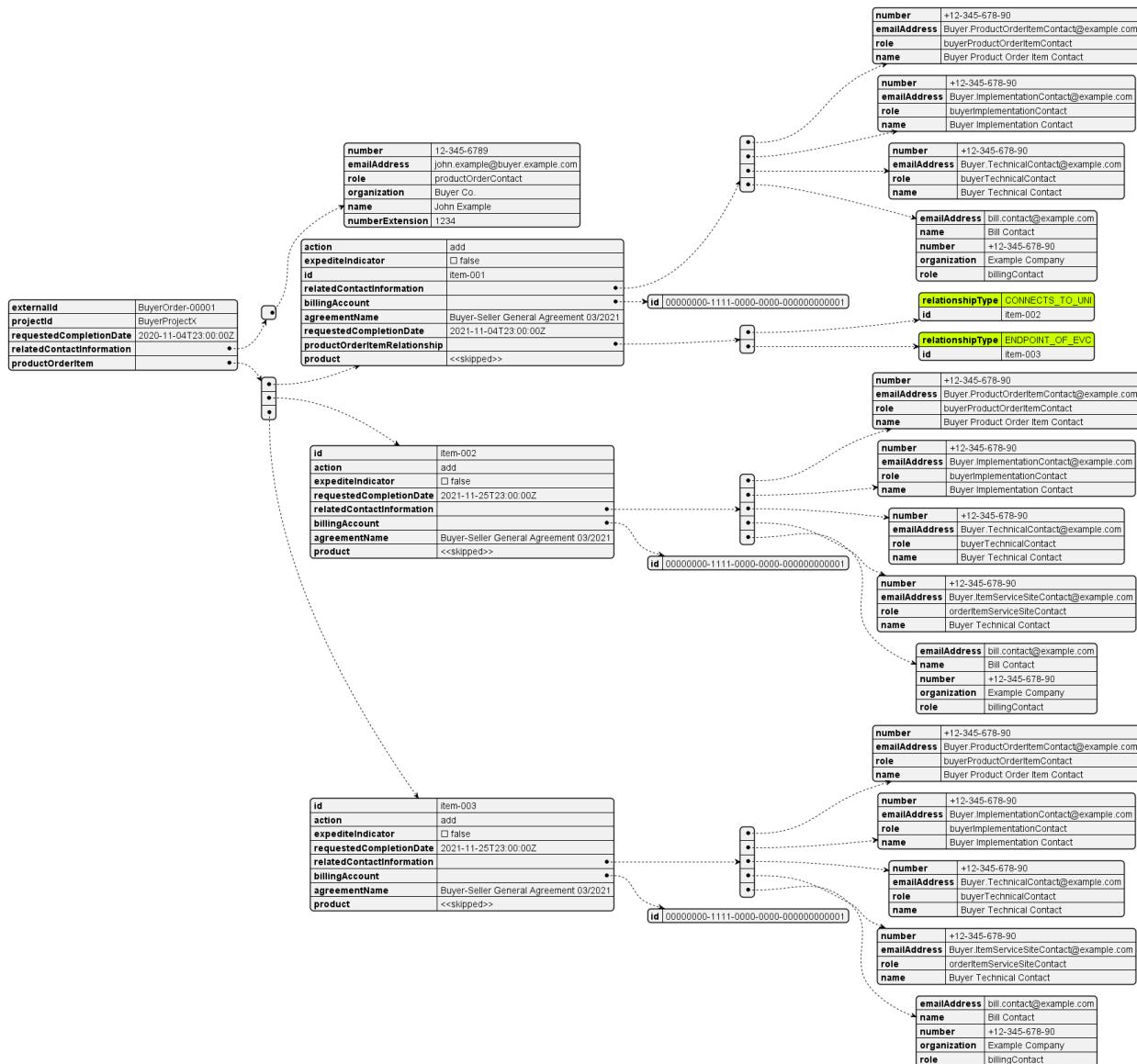
1803 **A.3.14 Use Case 5b: Product Order EVP-LAN**

1804 Detailed description of "Product Order" use case is located in A.3.13 part. This section will de-
1805 scribe the unique features of the EVP-LAN technology.

1806 This use case is the next step after use case 4b. It places an order for the installation of the, EVP-
1807 LAN, UNI and ENDPOPINT products, which were qualified and quoted in use cases 2 and 4.

1808 An example Product Order request can be found in the postman collection. Figure A2-46 presents
1809 it with product information skipped for readability.

1810

1811 <center>**Figure A2-46 – UC5b: EVP-LAN Product Order request**</center>

1812 **A.3.15 Use Case 5c: Product Order EP-TREE**

1813 Detailed description of "Quote" use case is located in A.3.10 part. This section will describe the
1814 unique features of the EP-TREE technology.
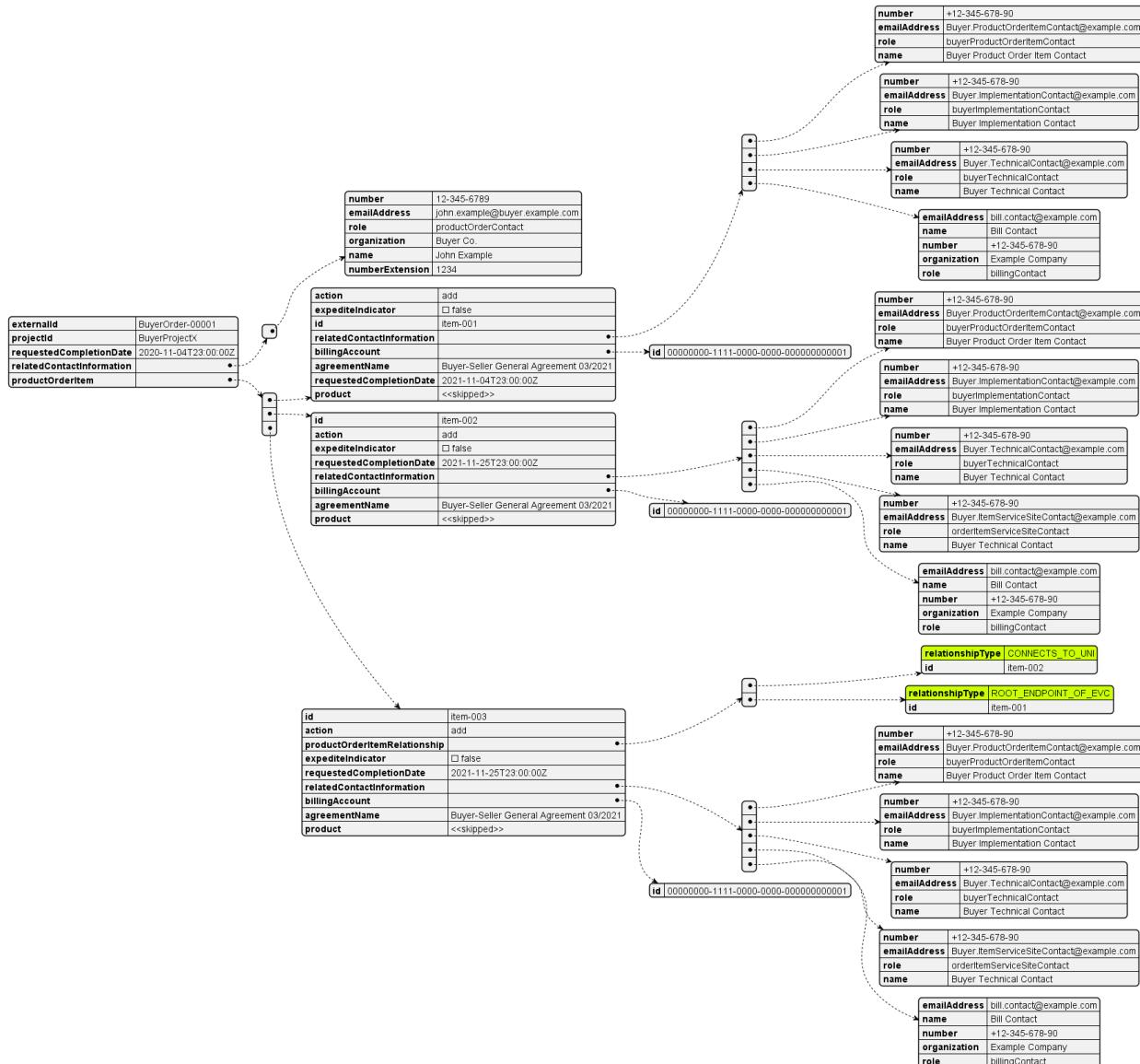
1815 This use case is the next step after use case 4c. It places an order for the installation of the EP-
1816 TREE and UNI products and ENDPOPINT products, which were qualified and quoted in use cases
1817 2 and 4.

1818 An example Product Order request can be found in the postman collection. Figure A2-47 presents
1819 it with product information skipped for readability.

Figure A2-47 – UC5c: EP-TREE Product Order request

A.4   action: modify

The mechanism of building a modification request for both envelope and payload for all steps are practically the same as for the create request.

The differences are in the following common rules (POQ, Quote, Order):

- "item.action" must be set to "modify"

- "item.product.id" of the product to be updated must be provided

- "product.productConfiguration" must contain all desired configuration (not only the updated values)
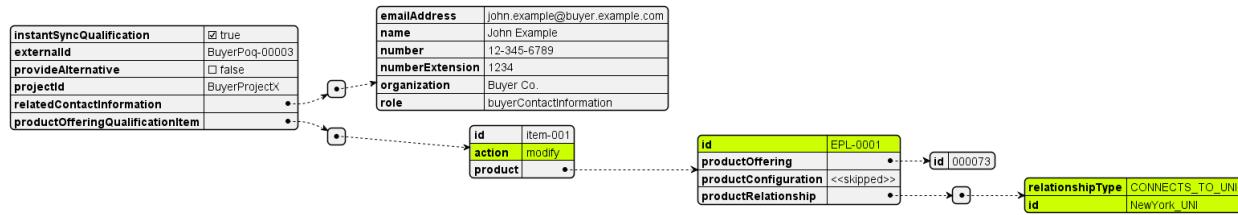
1830   -   "product.productOffering" must not be changed

1831   -   The Subscriber Ethernet Products do not allow "product.productRelationship", and "prod-
1832   uct.place" to be changed.

A.4.1   Use Case 6a: POQ: Bandwidth change EPL

1834 Use cases 6a, 7a, and 8a present POQ, Quote, and Order for an EPL bandwidth change. The change
1835 is made only for the attributes of the EPL product, so requests contain only one item (UNI product
1836 is not modified). The change is made by updating the "eir.irValue" and "eirMax.irValue" from 70
1837 to 100.

1838 Note that since there is no accompanying item with the UNI, the relationship information "CON-
1839 NECTS_TO_UNI" must be provided with the use of "product.productRelationship" attribute to
1840 point to the existing UNI instance with "id"="NewYork_UNI"

1841 The diagram below shows a POQ request for modification, highlighting the changes compared to
1842 the creation request.
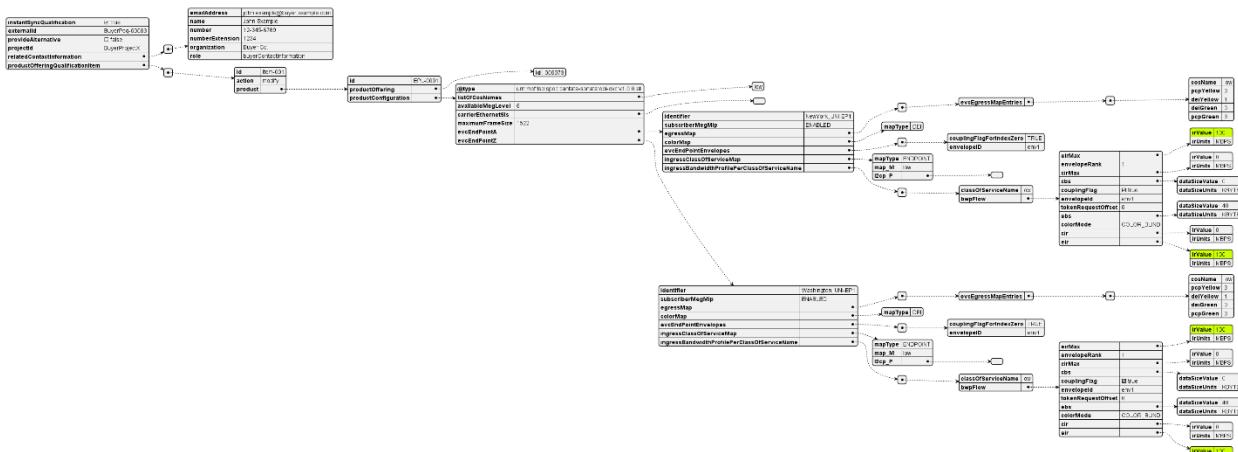
1843

1844

1845   **Figure A2-48 – UC6a: EPL POQ modify request**

1846

1847

1848   **Figure A2-49 – UC6a: EPL modified attributes**

1849 A full example request can be found in the attached postman collection.

A.4.2   Use Case 6b: POQ: Add UNI and Endpoint EVP-LAN

1851 Use cases 6b, 7b, and 8b present POQ, Quote, and Order for an EVP-LAN for adding new UNI
1852 and ENDPOINT. The change is made to add a new UNI and ENDPOINT colored in light blue in
1853 Figure A2-50 below

1854 Note that since there is no accompanying item with the UNI, the relationship information "CON-
1855 NECTS_TO_UNI" must be provided with the use of "productOfferingQualificationItem.qualifi-
1856 cationItemRelationship" attribute to point to the new UNI instance with "id"=" Philadelphia
1857 _UNI".

1858 The same situation is related to ENDPOINT, the relationship information "END-
1859 POINT_OF_EVC" must be provided with the use of "productOfferingQualificationItem.qualifi-
1860 cationItemRelationship" attribute to point to the new UNI instance with "id"=" Philadelphia
1861 _UNI_EP1".
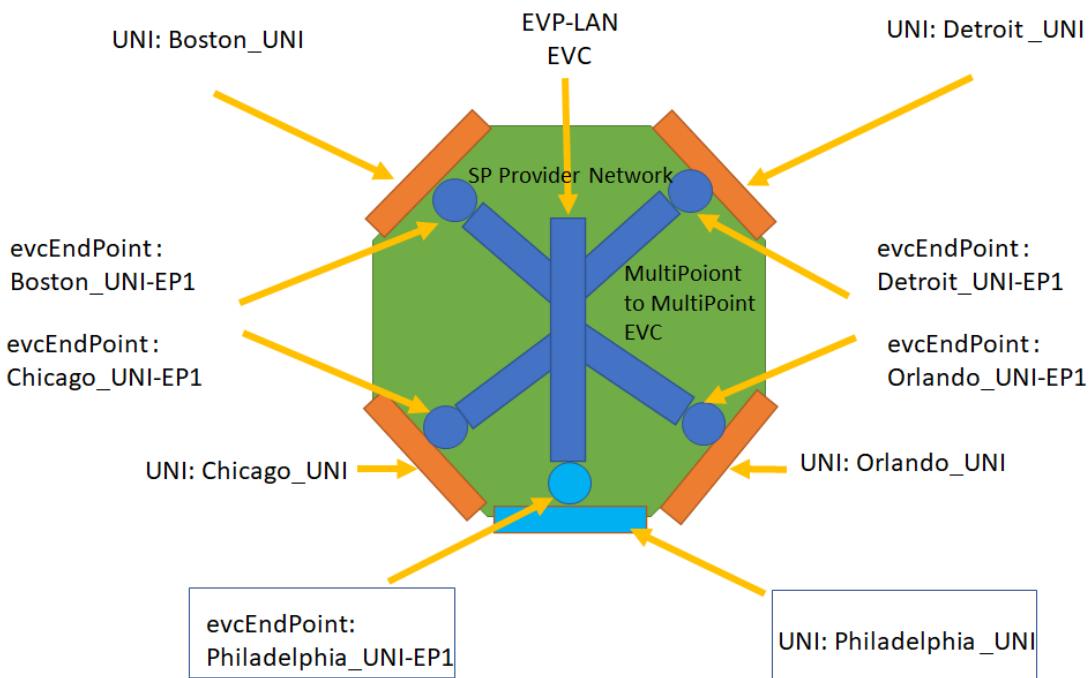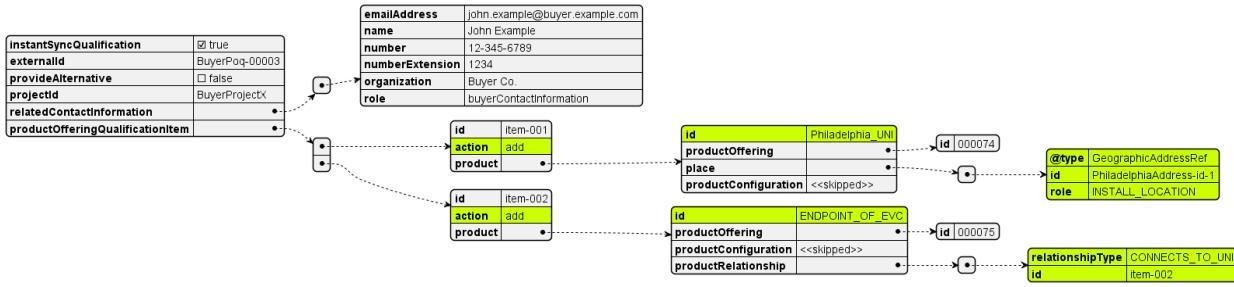
1862

1863 **Figure A2-50 – UC6b: EVP-LAN Add products diagram**

1864 The diagram below shows a POQ request for modification, highlighting the changes compared to
1865 the creation request.

1866

Figure A2-51 – UC6b: EVP-LAN POQ modify request
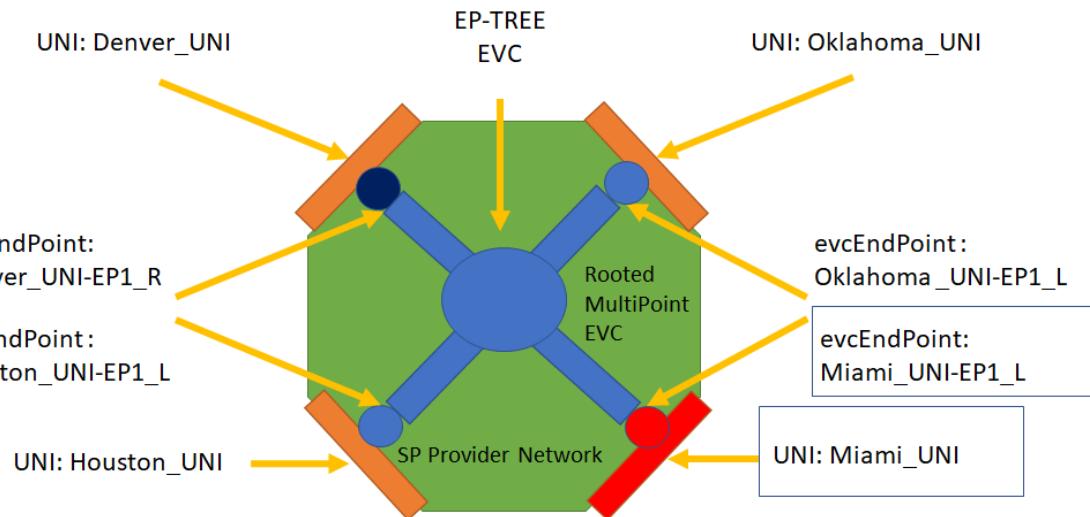
1868   A full example request can be found in the attached postman collection.

1869   ### A.4.3   Use Case 6c: POQ: Remove UNI and Endpoint EP-TREE

1870   Use cases 6c, 7c, and 8c present POQ, Quote, and Order for EP-TREE for removing UNI and
1871   ENDPOINT (evcEndPoint). The change is made to remove a particular UNI and ENDPOINT
1872   (evcEndPoint) shown in red in Figure A2-52.

1873   Note that since there is no accompanying item with the UNI, the relationship information "CON-
1874   NECTS_TO_UNI" must be provided with the use of "productOfferingQualificationItem.qualifi-
1875   cationItemRelationship" attribute to point to the new UNI instance with "id"="Miami_UNI".

1876   The same situation is related to ENDPOINT (evcEndPoint). The relationship information "END-
1877   POINT_OF_EVC" must be provided with the use of "productOfferingQualificationItem.qualifi-
1878   cationItemRelationship" attribute to point to the new UNI instance with "id"="Mi-
1879   ami_UNI_EP1_L".
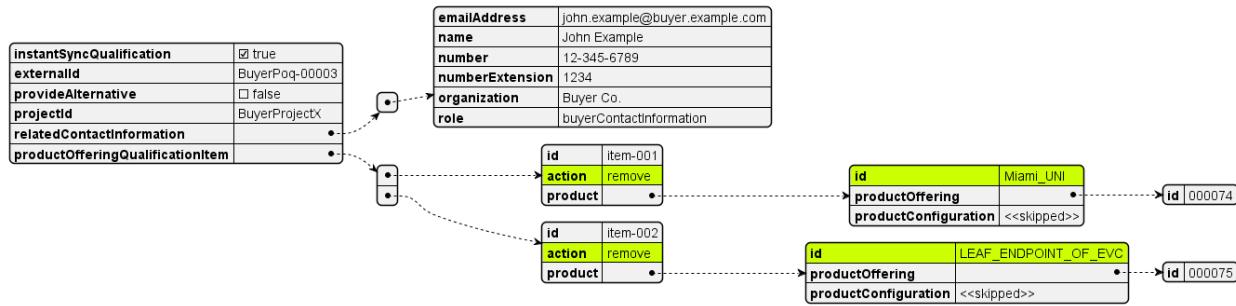
1880

Figure A2-52 – UC6c:EP-TREE Remove products diagram

1882   The diagram below shows a POQ request for modification, highlighting the changes compared to
1883   the creation request. (Figure A2-53)

1884
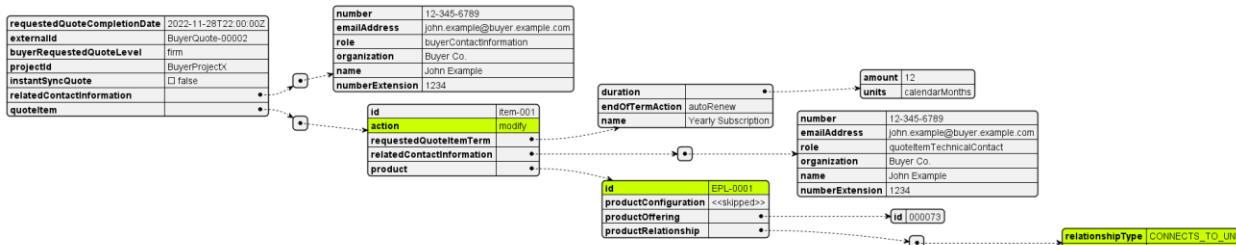
1885

1886   **Figure A2-53 – UC6c: EP-TREE POQ modify request**

1887   A full example request can be found in the attached postman collection.

1888   **A.4.4   Use Case 7a: Quote: Bandwidth change EPL**

1889   As the details of the product modification are already described in the previous chapter, this use
1890   case will only highlight the changes in the quote request, compared to the create request (Figure
1891   A2-54)
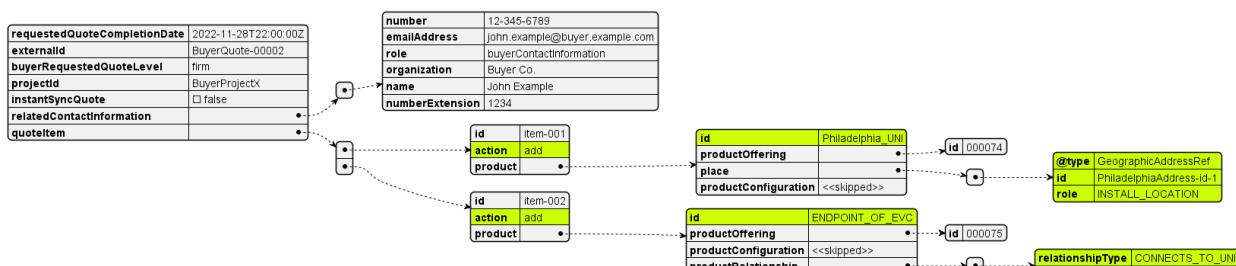
1892

1893   **Figure A2-54 – UC7a: EPL Quote modification request**

1894   A full example request can be found in the attached postman collection.

1895   **A.4.5   Use Case 7b: Quote: Add UNI and Endpoint EVP-LAN**

1896   As the details of the product modification are already described in the previous chapter, this use
1897   case will only highlight the changes in the quote request, compared to the create request (Figure
1898   A2-55)

1899
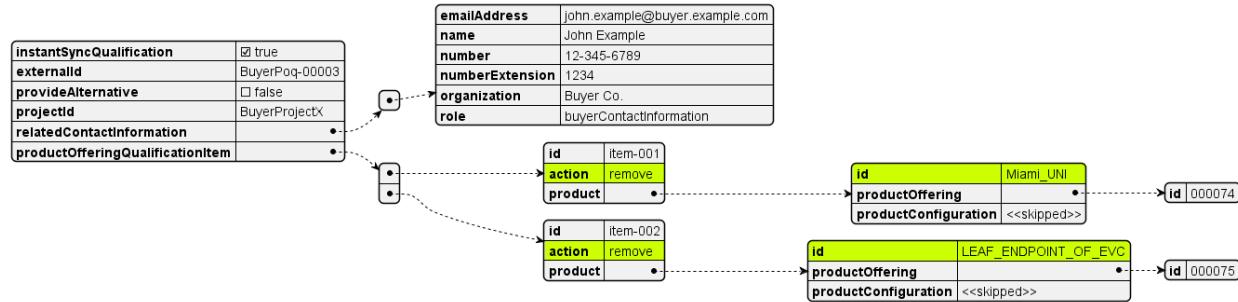
1900   **Figure A2-55 – UC7b: EVP-LAN Quote modification request**

1901   A full example request can be found in the attached postman collection.

1902 **A.4.6   Use Case 7c: Quote: Remove UNI and Endpoint EP-TREE**

1903 As the details of the product modification are already described in the previous chapter, this use
1904 case will only highlight the changes in the quote request, compared to the create request (Figure
1905 A2-56).
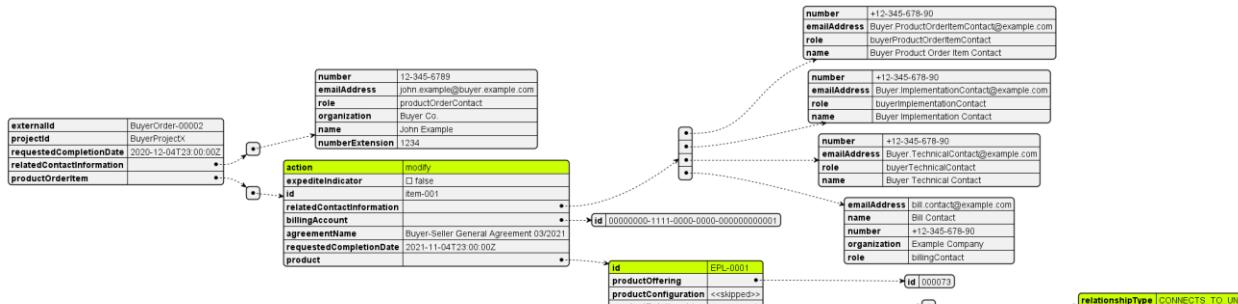
1906

1907 **Figure A2-56 – UC7c: EP-TREE Quote modification request**

1908 A full example request can be found in the attached postman collection.

1909 **A.4.7   Use Case 8a: Product Order: Bandwidth change EPL**

1910 All rules were described in the two above chapters. Figure A2-57 presents the Order request with
1911 highlighted changes:

1912

1913 **Figure A2-57 – UC8a: EPL Order modification request**

1914 A full example request can be found in the attached postman collection.

1915 **A.4.8   Use Case 8b: Product Order: Add Endpoint EVP-LAN**

1916 All rules were described in the two above chapters. Figure A2-58 presents the Order request with
1917 highlighted changes:

1918
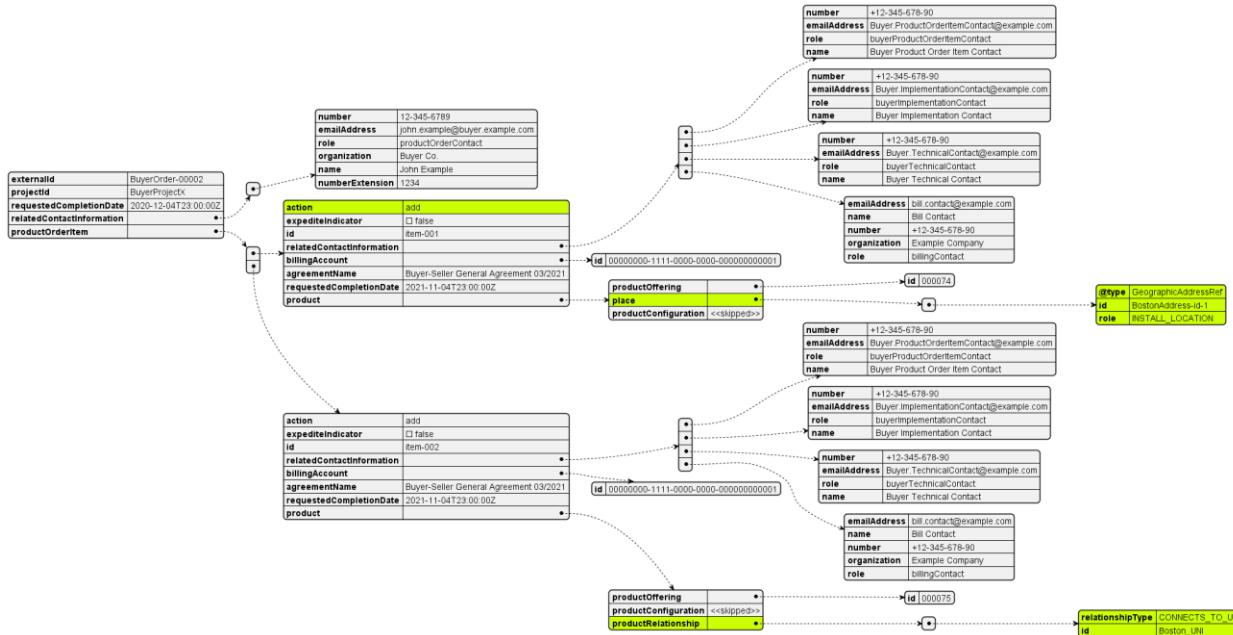
1919   **Figure A2-58 – UC8b: EVP-LAN Order modification request**

1920   A full example request can be found in the attached postman collection.

1921   **A.4.9   Use Case 8c: Product Order: Remove Endpoint EP-TREE**

1922   All rules were described in the two above chapters. Figure A2-59 presents the Order request with
1923   highlighted changes:

1924
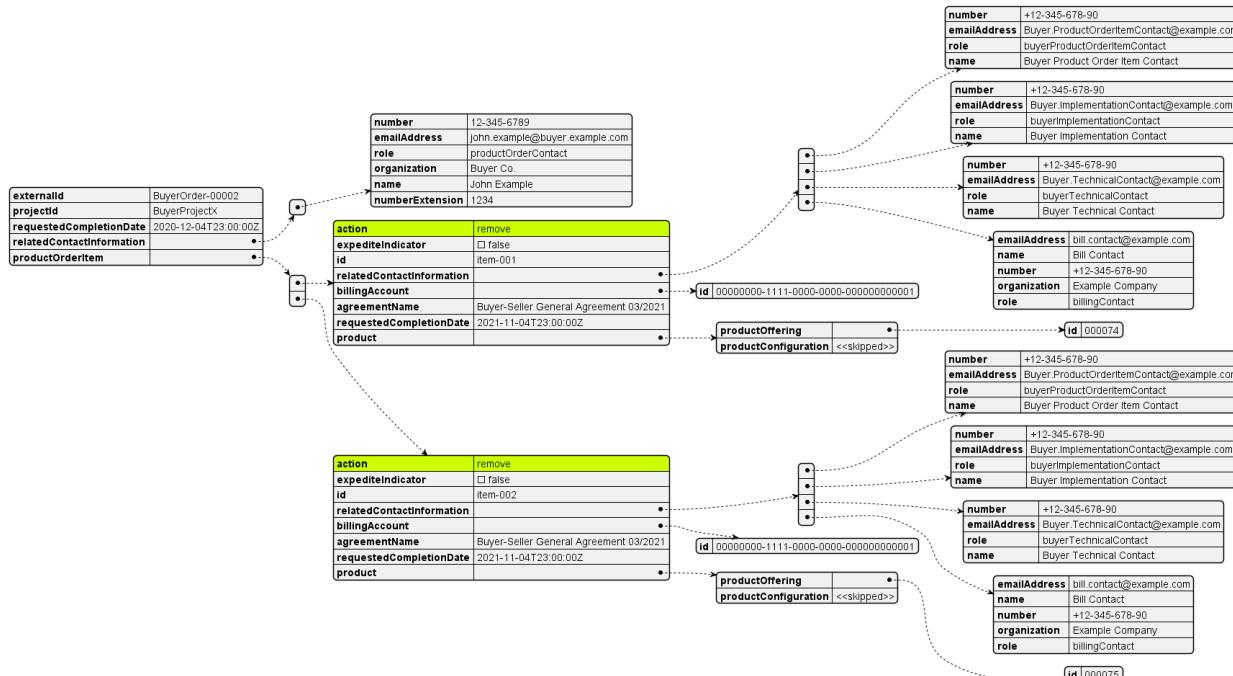
1925   **Figure A2-59 – UC8c: EP-TREE Order modification request**
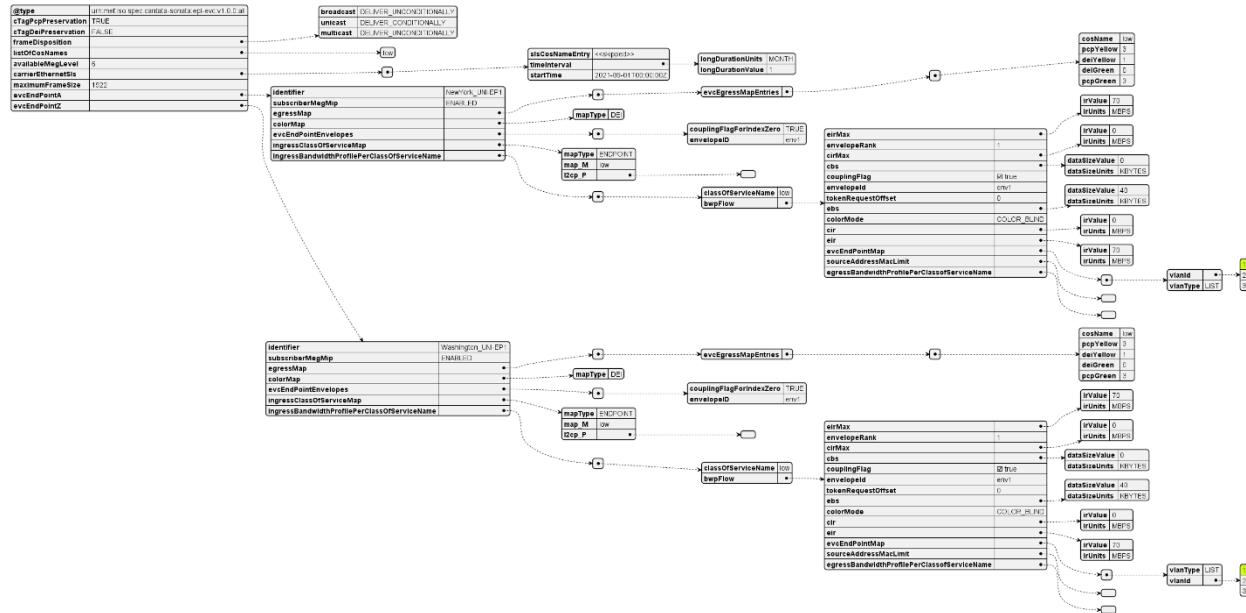
1926 A full example request can be found in the attached postman collection.

1927

A.4.10 Use Case 9a: Product Order: VLAN change at the UNI (EVPL)

1929 In this case, an order to enhance the list of VLAN IDs are mapped to the UNI End Point is enhanced
1930 from [1, 2] to [1, 2, 3]. This is done with the order request as above, but with a slightly modified
1931 product payload, which is presented in Figure A2-60:

1932

1933

1934 **Figure A2-60 – UC9: EVPL Order modification request**

1935 A full example request can be found in the attached postman collection.

A.5   action: delete

1937 Delete requests are for all steps are very straightforward, as they only carry the product "id".

1938 Following common rules apply for disconnect operation:

1939 - "item.action" must be set to "delete"

1940 - "item.product.id" of the product to be deleted must be provided

1941 - "product.productConfiguration" must not be provided

1942 - no other item attribute may be provided (except for optional "billingAccount" in Order)

A.5.1   Use Case 10a: Product Order: Delete both EPL and UNIs

Deletion of both EPL and UNIs products can be ordered with a request that is presented in Figure A2-61:
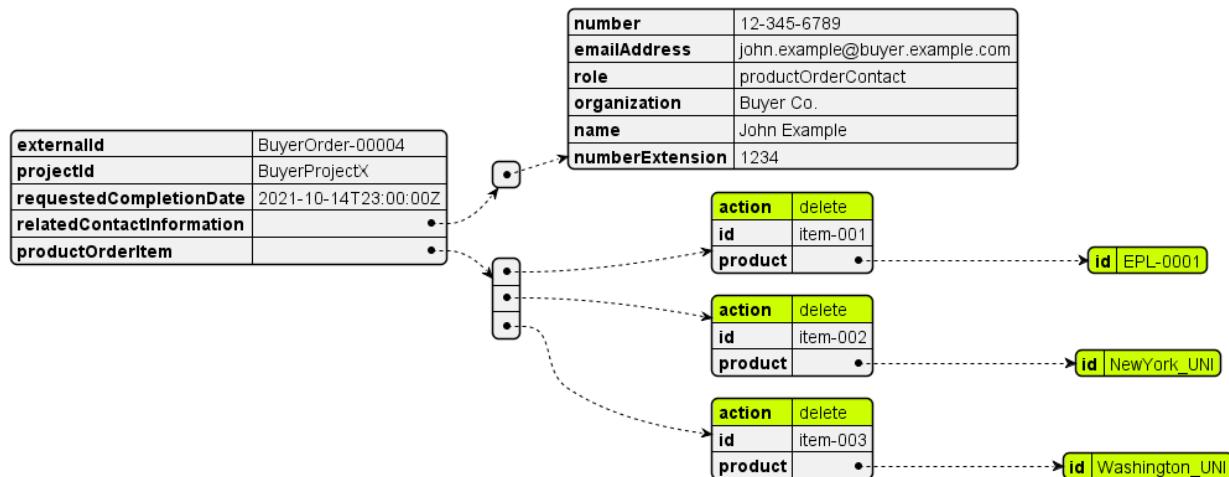
Figure A2-61 – UC10a: EPL Order deletion request

Note: A disconnect request may result in additional charges (if not quoted earlier).

JSON representation of this request:

```
{
    "externalId": "BuyerOrder-00004",
    "projectId": "BuyerProjectX",
    "requestedCompletionDate": "2021-10-14T23:00:00Z",
    "relatedContactInformation": [
        {
            "number": "12-345-6789",
            "emailAddress": "john.example@buyer.example.com",
            "role": "productOrderContact",
            "organization": "Buyer Co.",
            "name": "John Example",
            "numberExtension": "1234"
        }
    ],
    "productOrderItem": [
        {
            "action": "delete",
            "id": "item-001",
            "product": {
                "id": "EPL-0001"
            }
        },
        {
```

```
1973          "action": "delete",
1974          "id": "item-002",
1975          "product": {
1976             "id": "NewYork_UNI"
1977          }
1978       },
1979       {
1980          "action": "delete",
1981          "id": "item-003",
1982          "product": {
1983             "id": "Washington_UNI"
1984          }
1985       }
1986    ]
1987 }
1988
```

A.5.2 Use Case 10b: Product Order: Delete EVP-LAN UNI and ENDPOINT

Deletion of both EVP-LAN UNI and ENDPOINT (evcEndPoint) products can be ordered with a request that is presented in Figure A2-62:


Figure A2-62 – UC10b: EVP-LAN Order deletion request

Note: A disconnect request may result in additional charges (if not quoted earlier).

JSON representation of this request:

```
1996 {
1997    "externalId": "BuyerOrder-00004",
1998    "projectId": "BuyerProjectX",
1999    "requestedCompletionDate": "2021-10-14T23:00:00Z",
2000    "relatedContactInformation": [
2001       {
2002          "number": "12-345-6789",
```

```
2003            "emailAddress": "john.example@buyer.example.com",
2004            "role": "productOrderContact",
2005            "organization": "Buyer Co.",
2006            "name": "John Example",
2007            "numberExtension": "1234"
2008          }
2009       ],
2010       "productOrderItem": [
2011         {
2012            "action": "delete",
2013            "id": "item-001",
2014            "product": {
2015              "id": "EVPLAN-0001"
2016            }
2017         },
2018         {
2019            "action": "delete",
2020            "id": "item-002",
2021            "product": {
2022              "id": "Boston_UNI"
2023            }
2024         },
2025         {
2026            "action": "delete",
2027            "id": "item-002",
2028            "product": {
2029              "id": "Boston_UNI-EP1"
2030            }
2031         }
2032       ]
2033     }
```

2034   **A.5.3   Use Case IOc: Product Order: Delete EP-TREE UNI and ENDPOINT**

2035   Deletion of EP-TREE UNI and ENDPOINT (evcEndPoint) products can be ordered with a request
2036   that is presented in Figure A2-63:

2037
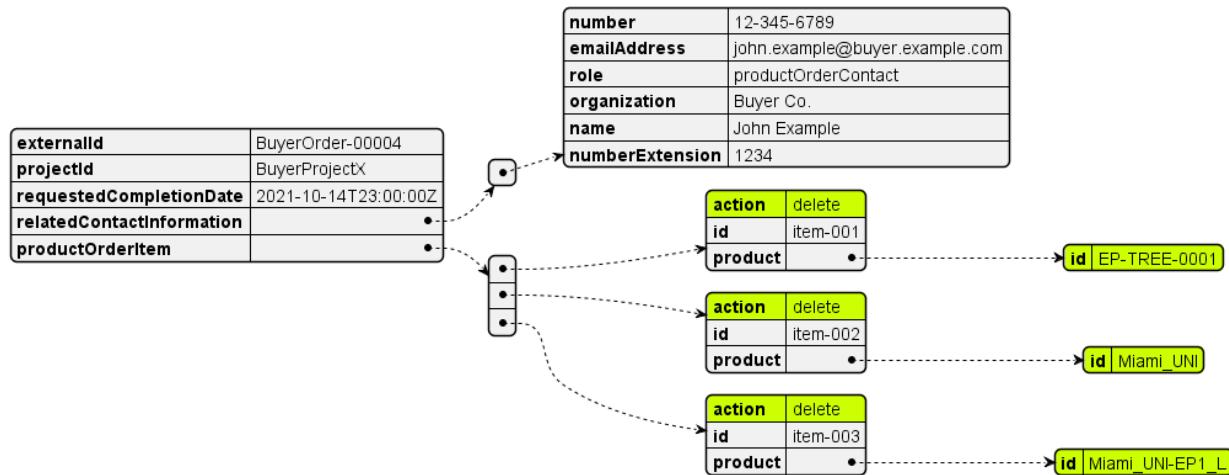
2038 **Figure A2-63 – UC10c: EP-TEE Order deletion request**

2039 Note: A disconnect request may result in additional charges (if not quoted earlier).

2040 **A.5.4   Use Case 11a: Move EPL to a different Location**

2041 The case of moving the office to another building cannot be realized by a single update of the
2042 "place" attribute of the UNI product.

- *The location and physical layer of a UNI cannot be changed once it is ordered; in-*
2043
2044 *stead, this is handled as an installation (UNI at the new location) and a disconnect*
2045 *(UNI at previous location), as there is often a requirement for a smooth transition with*
2046 *minimum downtime.*

2047 Nor it can be realized by updating EPL's product reference to another UNI. Chapter 11:

2048 *Changing the UNI Reference or the UNI Location is not supported for an* EPL, EVPL *Service. The*
2049 *value included in a Change request must be identical to the value in the Inventory.*

2050 So, the argument is both business and technical. In order to realize this use case, the following
2051 requests must be performed:

2052   1. Creation of new UNI at the new location

2053   2. Creation of a new EPL

2054   3. Deletion of an old EPL

2055   4. Deletion if an old UNI (optionally, if not used by other connections)

2056 Step 1 as potentially requiring physical installation should be performed earlier to prepare for a
2057 switchover. Steps 2 and 3 should be coordinated to assure minimal downtime.

2058 This use case as being built upon already described steps is not part of the attached postman col-
2059 lection.

A.5.5   Use Case 11b: Move EVP-LAN to a different Location

The case of moving the office to another building cannot be realized by a single update of the "place" attribute of the UNI product.
- *The location and physical layer of a UNI cannot be changed once it is ordered; instead, this is handled as an installation (UNI at the new location) and a disconnect (UNI at previous location), as there is often a requirement for a smooth transition with minimum downtime.*

Nor it can be realized by updating EVP-LAN product reference to another UNI. Chapter 11:

Changing the UNI Reference or the UNI Location is not supported for an EVP-LAN *Service. The value included in a Change request must be identical to the value in the Inventory.*

So, the argument is both business and technical. In order to realize this use case, the following requests must be performed:

1. Creation of new UNI at the new location

2. Creation of a new EVP-LAN

3. Deletion of an old EVP-LAN

4. Deletion if an old UNI (optionally, if not used by other connections)

5. Deletion if an old ENDPOINT (evcEndPoint) (optionally, if not used by other connections)

Step 1 as potentially requiring physical installation should be performed earlier to prepare for a switchover. Steps 2 and 3 should be coordinated to assure minimal downtime.

This use case as being built upon already described steps is not part of the attached postman collection.

A.5.6   Use Case 11c: Move EP-TREE to a different Location

The case of moving the office to another building cannot be realized by a single update of the "place" attribute of the UNI product.
- *The location and physical layer of a UNI cannot be changed once it is ordered; instead, this is handled as an installation (UNI at the new location) and a disconnect (UNI at previous location), as there is often a requirement for a smooth transition with minimum downtime.*

Nor it can be realized by updating EP-TREE product reference to another UNI. Chapter 11:

Changing the UNI Reference or the UNI Location is not supported for an EP-TREE *Service. The value included in a Change request must be identical to the value in the Inventory.*

So, the argument is both business and technical. In order to realize this use case, the following requests must be performed:

1. Creation of new UNI at the new location

2. Creation of a new EP-TREE

3. Deletion of an old EP-TREE,

4. Deletion if an old UNI (optionally, if not used by other connections)

5. Deletion if an old ENDPOINT (evcEndPoint) (optionally, if not used by other connections)

Step 1 as potentially requiring physical installation should be performed earlier to prepare for a switchover. Steps 2 and 3 should be coordinated to assure minimal downtime.

This use case as being built upon already described steps is not part of the attached postman collection.

6 References

[1] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997

[2] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by B Leiba, May 2017, Copyright © 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

2120   [3]   MEF 125, *LSO Cantata and LSO Sonata - Subscriber Ethernet Product Schemas and*
2121   *Developer Guide*, July 2022

2122   [4]   MEF 87, *LSO Cantata and LSO Sonata Product Offering Qualification API – Devel-*
2123   *oper Guide*, May 2022

2124   [5]   MEF 115, *LSO Cantata and LSO Sonata Quote Management API – Developer Guide*,
2125   May 2022

2126   [6]   MEF 121, *LSO Cantata and LSO Sonata Address Management API – Developer Guide*,
2127   May 2022

2128