



**Working Draft 1.0**

**API Developer Guide**

**Product Inventory Management**

**October 2018**

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

## Disclaimer

This guide describes a set of APIs that may be implemented on one or more Interface Reference Points (IRPs) of the LSO Reference Architecture, as defined in MEF 55. This guide is not a MEF specification and is not based on a MEF consensus-driven technical specification development process.

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

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

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

- (a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

- (b) any warranty or representation that any MEF member will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

- (c) any form of relationship between any MEF member and the recipient or user of this document.

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

© MEF Forum 2018. All Rights Reserved

54

## Table of Contents

55	1.	List of Contributing Members .....	1
56	2.	Abstract .....	1
57	3.	Scope .....	1
58	4.	Resource Models .....	2
59	4.1	Product Inventory Resource Model .....	2
60	5.	State Diagrams .....	3
61	5.1	Product Inventory State Machine .....	3
62	6.	Notifications .....	3
63	7.	JSON Representation Samples .....	4
64	8.	API Interactions .....	6
65	8.1	Sequence Diagram .....	6
66	8.2	Operations .....	7
67	8.2.1	Retrieve Product .....	7
68	8.2.2	Retrieve a Single Product Order by Order Identifier .....	9
69	9.	Appendix A - Relationship to Product Ordering API .....	12
70	9.1	Product Order(s) & Product Inventory sync .....	12
71	9.2	Relationship between Order Item Status and Product Status .....	14

72

## List of Figures

73	Figure 1 – Product Inventory Resource Model.....	2
74	Figure 2 – Product Inventory State Machine .....	3
75	Figure 3 – Product Inventory API Interactions.....	6

76

## List of Tables

77	Table 1 – Contributing Member Companies .....	1
78	Table 2 – Product State Values .....	3
79	Table 3 – Product Order Example .....	14
80	Table 4 – Install Product Order Relationship .....	15
81	Table 5 – Suspend Product Order Relationship.....	16
82	Table 6 – Reactivate Product Order Relationship.....	16
83	Table 7 – Disconnect Product Order Relationship .....	17
84	Table 8 – Change Product Order Relationship .....	18

85

## 1. List of Contributing Members

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

Member Company

Table 1 – Contributing Member Companies

## 2. Abstract

This API Guide is intended to help software developers to implement the MEF Product Inventory API. The API swagger is available on the MEF [GitHub](#). This document covers the following topics:

- Product Inventory API Resource Model
- Product Resource Lifecycle
- Product Inventory API Notifications
- Product Resource JSON Representation
- Description of all API Operations

## 3. Scope

The scope of this API guide covers project the following product inventory functions:

- Retrieving Product Inventory instances based upon a filter of Product attributes
- Retrieving a specific Product Inventory instance based upon a Product identifier

Specifically, out of scope for this release are the following functions:

- Creating a Product instance in an inventory system
- Completely updating a Product instance in an inventory system
- Partially modifying a Product instance in an inventory system
- Deleting a Product instance from an inventory system

## 4. Resource Models

### 4.1 Product Inventory Resource Model

The Product Inventory REST resource model is as follows:

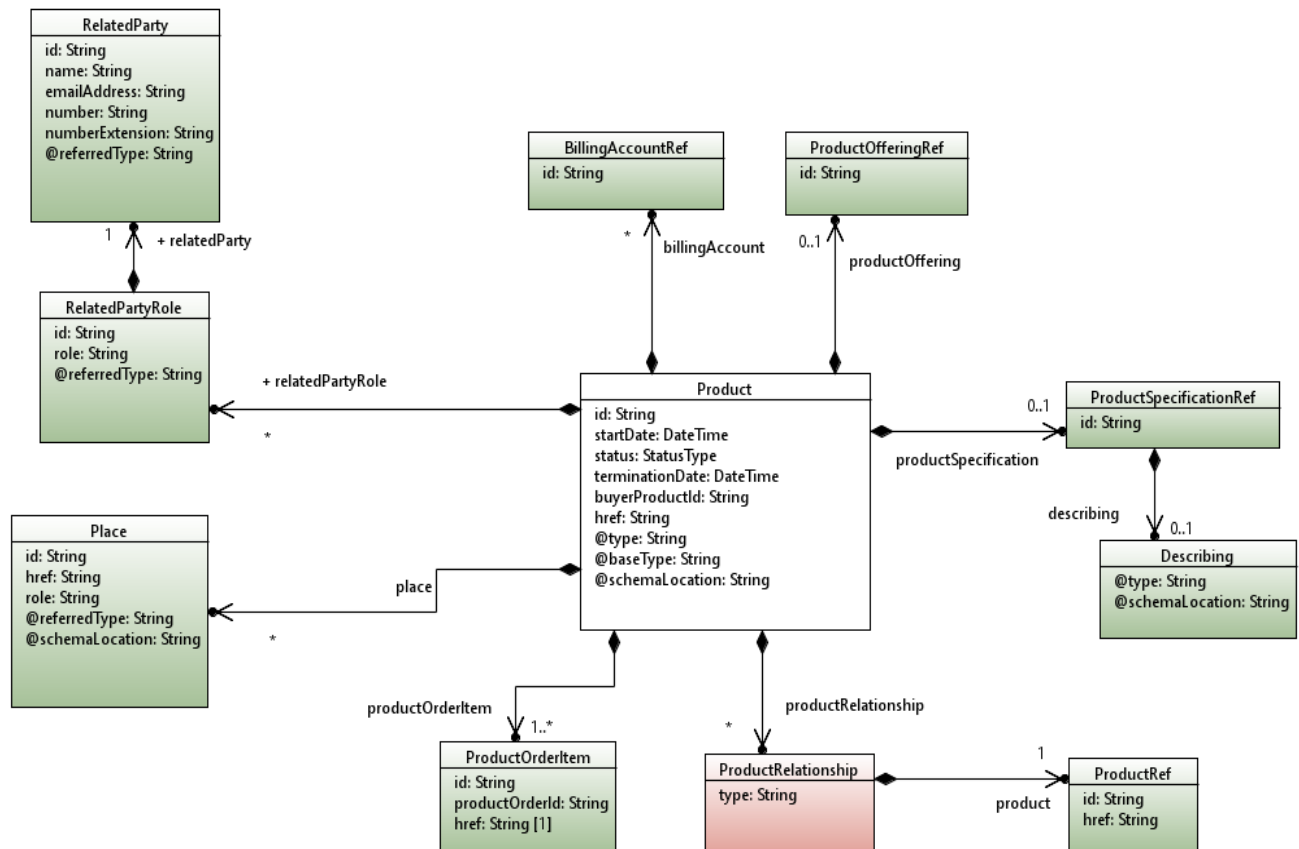


Figure 1 – Product Inventory Resource Model

Color coding scheme:

- White box: API main resource
- Pink box: API sub resource(s)
- Green boxes: API related/referred resource(s)

## 5. State Diagrams

### 5.1 Product Inventory State Machine

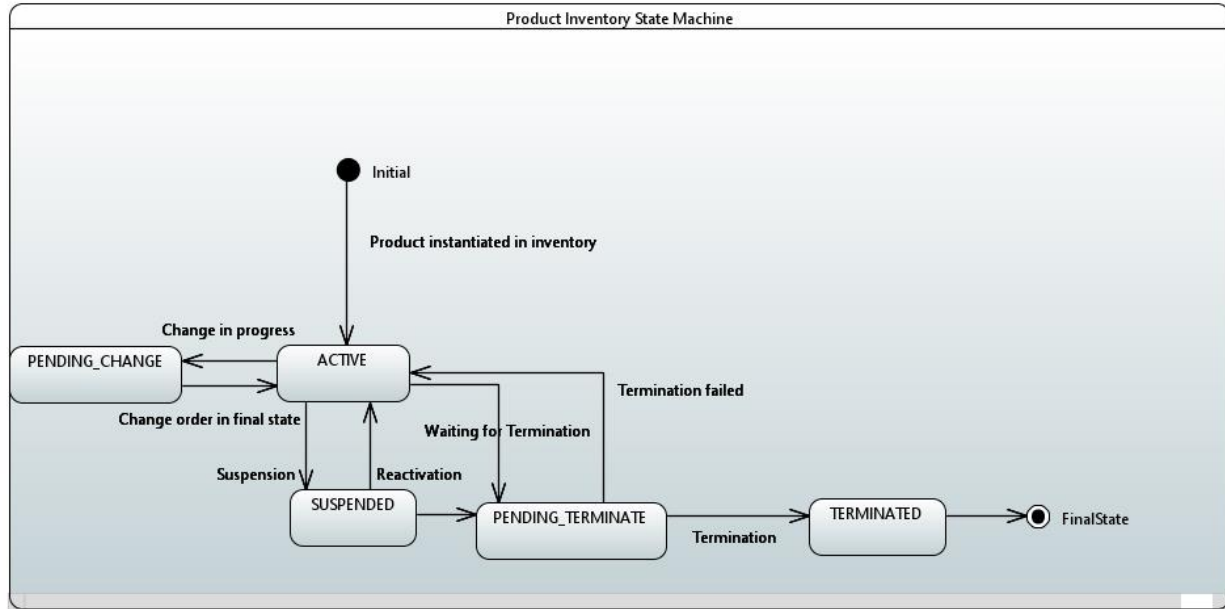


Figure 2 – Product Inventory State Machine

The definitions of the various product states are as follows:

State	Description
ACTIVE	The product has been successfully installed or has transitioned from a PENDING_CHANGE, SUSPENDED, or PENDING_TERMINATE state.
PENDING_CHANGE	The product was previously ACTIVE and a product order to change the product is in progress.
SUSPENDED	A product has been successfully suspended.
PENDING_TERMINATE	The product is in the process of being terminated via a disconnect product order.
TERMINATED	The product has been successfully terminated via a disconnect product order.

Table 2 – Product State Values

## 6. Notifications

No notifications are defined for this API.

## 7. JSON Representation Samples

The following is a JSON representation of an example of a "Product" resource object.

This example illustrates a UNI product – Let's assume that the UNI was created in a previous order and then sometime later was modified by another order.

```
{
  "id": "1546-45f6-7sqs-45th",
  "status": "ACTIVE",
  "startDate": "2018-02-02T08:52:15.705Z",
  "href":
"/serverRoot/api/mef/productInventoryManagement/v1/product/1546-45f6-7sqs-45th",
  "buyerProductId": "buyProd012jk364",
  "place": [
    {
      "id": "12df-zz56-hjy5-rr56",
      "role": "UNI Site",
      "@referredType": "site"
    }
  ],
  "productOffering": {
    "id": "OrangeUNI"
  },
  "productSpecification": {
    "id": "UNISpec",
    "describing": {
      "@type": "UNISpec",
      "@schemaLocation": "{MEF Product Spec
Location}/UNISpec.json"
    }
  },
  "/*": "This is where product spec characteristic values would be
provided",
  "physicalLayer": "10GBASE-SW",
  "synchronousModeEnabled": false,
  "numberOfLinks": 1,
  "billingAccount": [
    {
      "id": "7891-23654-dfre-ty55"
    }
  ],
  "relatedPartyRole": [
    {
      "role": "Technical Contact",
      "relatedParty": {
        "name": "Jean Pontus",
        "number": "689741456",
        "numberExtension": "401",
        "emailAddress": "jean@orange.com",
        "@referredType": "individual"
      }
    }
  ],
}
```



```
{
  "role": "Implementation Contact",
  "relatedParty": {
    "name": "Jean Pontus",
    "number": "689741456",
    "numberExtension": "401",
    "emailAddress": "jean@orange.com",
    "@REFERREDType": "individual"
  }
},
{
  "role": "Site Contact",
  "relatedParty": {
    "name": "Jean Pontus",
    "number": "689741456",
    "numberExtension": "401",
    "emailAddress": "jean@orange.com",
    "@REFERREDType": "individual"
  }
}
],
"productOrderItem": [
  {
    "id": "1236-tred-ty56-78uu",
    "href":
"/serverRoot/api/mef/productOrderManagement/v1/productOrder/1236-tred-ty56-78uu",
    "productOrderItem": "1"
  },
  {
    "id": "1aaz6-gfff-sd56-sf22",
    "href":
"/serverRoot/api/mef/productOrderManagement/v1/productOrder/1aaz6-gfff-sd56-sf22",
    "productOrderItem": "3"
  }
],
"@type": "MEFproduct"
}
```

137

138

## 8. API Interactions

### 8.1 Sequence Diagram

The product interaction model is described below:

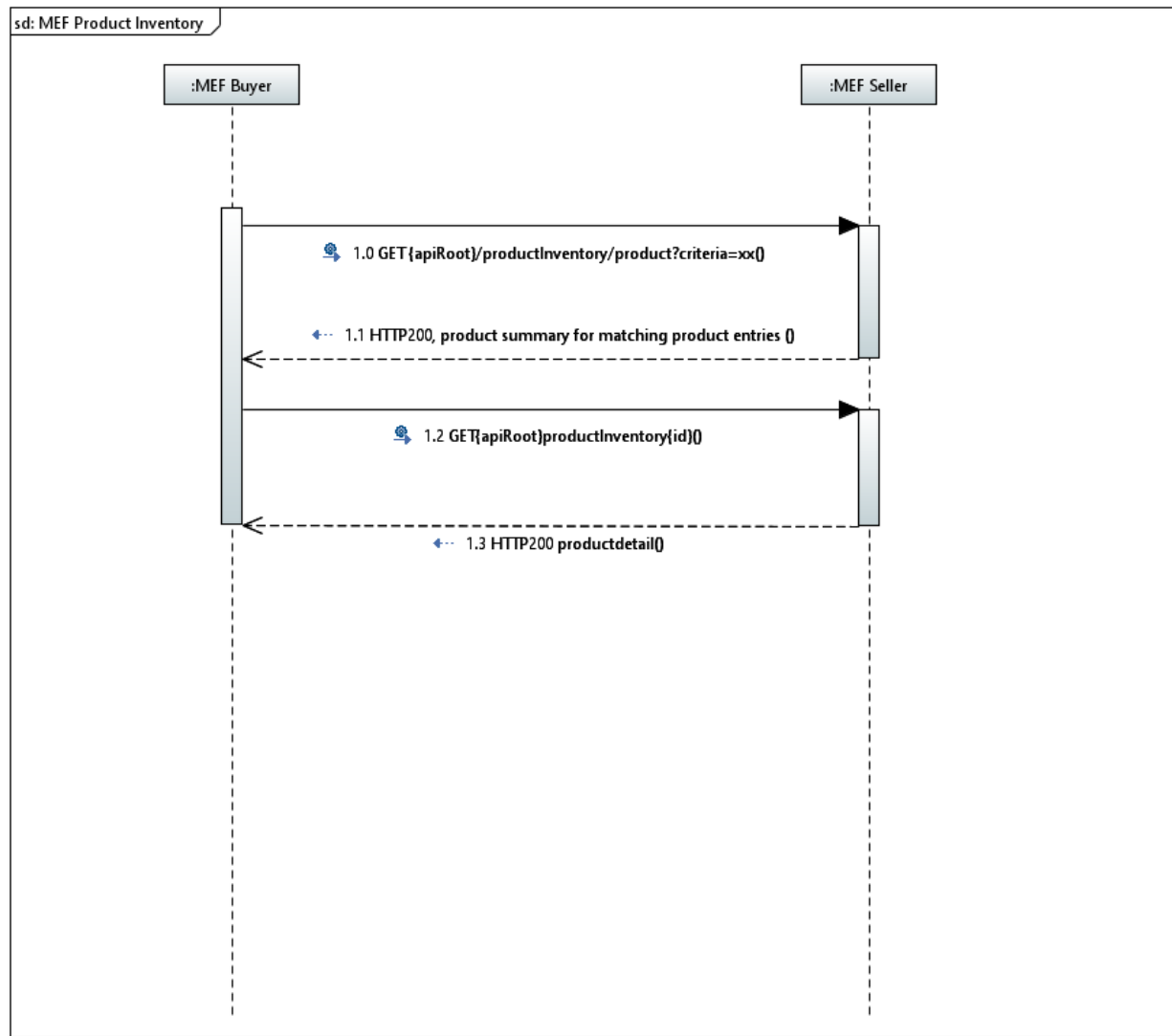


Figure 3 – Product Inventory API Interactions

## 8.2 Operations

### 8.2.1 Retrieve Product

GET /product?{filtering}

#### Description

This operation is used to retrieve product (s) corresponding to search criteria.

The response will be a product summary.

Only the following attributes can be used as search criteria

- status
- productSpecification.id
- productOffering.id
- relatedPartyId Role (role + relatedPartyId)
- relatedPartyRoleId
- buyerProductId
- placeId +placeType
- buyerProductId
- offset (requested index for start of resources to be provided in response requested by client)
- limit (Requested number of resources to be provided in response requested by client)

Only the following attributes will be retrieved in the summary view:

- id
- href
- status
- startDate
- productOffering.id
- productSpecification.id
- buyerProductId

#### Behavior

- Returns HTTP/1.1 status code 200 if the request was successful
- To be noted that if not product match the request criteria, an empty list is retrieved (no error code)

180 Otherwise:

400	Bad Request
401	Unauthorized
403	Forbidden
422	Unprocessable entity
503	Service Unavailable

181 Specific business errors will be encapsulated in HTTP Response 422 Unprocessable entity:

- 182     • 100: Too many records retrieved – please restrict requested parameter value(s)
- 183     • 101: Incomplete request – if relatedPartyId is filled, a role must also be filled.
- 184     • 102: Incomplete request – if role id filled, a relatedPartyRole must be filled.

## 185 Usage Samples

### 186 Request:

```
GET /serverRoot/api/mef/productInventoryManagement/v1/product?status=ACTIVE
Accept: application/json
```

### 187 Response:

```
188 200
189 [
190   {
191     "id": "1546-45f6-7sqs-45th",
192     "status": "ACTIVE",
193     "startDate": "2018-02-02T16:09:36.032Z",
194     "href": "/serverRoot/api/mef
195 /productInventoryManagement/v1/product/1546-45f6-7sqs-45th",
196     "productOffering": {
197       "id": "OrangeUNI"
198     },
199     "productSpecification": {
200       "id": "UNISpec"
201     },
202     "buyerProductId": "012jk364"
203   },
204   {
205     "id": "sd22-aa45-98ty-2236",
206     "status": "ACTIVE",
207     "startDate": "2018-02-02T16:09:37.089Z",
208     "href": "/serverRoot/api/mef
209 /productInventoryManagement/v1/product/sd22-aa45-98ty-2236",
210     "productOffering": {
211       "id": "OrangeELine"
212     },
213     "productSpecification": {
214       "id": "eLineSpec"
215     },
216   }
217 ]
```



```
216         "buyerProductId": "435GHy"  
217     }  
218 ]
```

## 219 8.2.2 Retrieve a Single Product Order by Order Identifier

220  
221 GET /product/{productId}

### 222 Description

223 This operation is used to retrieve one product instance

224 The response will be a product resource representation with all the attributes

### 225 Behavior

- 226 • Returns HTTP/1.1 status code 200 if the request was successful

227 Otherwise:

400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
503	Service Unavailable

### 228 Usage Samples

#### 229 Request:

```
GET /serverRoot/api/mef/productInventoryManagement/v1/product/1546-45f6-7sqs-45th  
Accept: application/json
```

#### 230 Response:

```
200  
  
{  
    "id": "1546-45f6-7sqs-45th",  
    "status": "ACTIVE",  
    "startDate": "2018-02-02T08:52:15.705Z",  
    "href":  
"/serverRoot/api/mef/productInventoryManagement/v1/product/1546-45f6-7sqs-45th",  
    "buyerProductId": "buyProd012jk364",  
    "place": [  
        {  
            "id": "12df-zz56-hjy5-rr56",  
            "role": "UNI Site",
```

```

        "@referredType": "site"
    },
    ],
    "productOffering": {
        "id": "OrangeUNI"
    },
    "productSpecification": {
        "id": "UNISpec",
        "describing": {
            "@type": "UNISpec",
            "@schemaLocation": "{MEF Product Spec
Location}/UNISpec.json"
        }
    },
    "///": "This is where product spec characteristic values would be
given",
    "physicalLayer": "10GBASE-SW",
    "synchronousModeEnabled": false,
    "numberOfLinks": 1,
    "billingAccount": [
        {
            "id": "7891-23654-dfre-ty55"
        }
    ],
    "relatedPartyRole": [
        {
            "role": "Technical Contact",
            "relatedParty": {
                "name": "Jean Pontus",
                "number": "689741456",
                "numberExtension": "401",
                "emailAddress": "jean@orange.com",
                "@referredType": "individual"
            }
        },
        {
            "role": "Implementation Contact",
            "relatedParty": {
                "name": "Jean Pontus",
                "number": "689741456",
                "numberExtension": "401",
                "emailAddress": "jean@orange.com",
                "@referredType": "individual"
            }
        },
        {
            "role": "Site Contact",
            "relatedParty": {
                "name": "Jean Pontus",
                "number": "689741456",
                "numberExtension": "401",
                "emailAddress": "jean@orange.com",
                "@referredType": "individual"
            }
        }
    ],
    "productOrderItem": [

```

```
{
  {
    "id": "1236-tred-ty56-78uu",
    "href":
"/serverRoot/api/mef/productOrderManagement/v1/productOrder/1236-tred-ty56-
78uu",
    "productOrderItem": "1"
  },
  {
    "id": "1aaz6-gfff-sd56-sf22",
    "href":
"/serverRoot/api/mef/productOrderManagement/v1/productOrder/1aaz6-gfff-
sd56-sf22",
    "productOrderItem": "3"
  }
],
"@type": "MEFproduct"
}
```

231

232

## 9. Appendix A - Relationship to Product Ordering API

### 9.1 Product Order(s) & Product Inventory sync

One golden rule: The product inventory is built and updated only through productOrder processing. Updating covers product characteristic changes and suspension, reactivation or termination of a product.

This means that a product is the result of a succession of order/order items that have created it and then modified it.

- A productOrder is a transaction that describes actions on product(s). An order Item describes one action on one product (or future one for 'INSTALL' order item action). Once fulfilled a productOrder has not any impact on the product.
- A product is a persistent object in SP Seller's database.
- The Buyer must be able to retrieve 'his' product in the Seller's database to use them in a productOrder to describe expected modifications.

Remark: only for administrative reasons (coming from the Seller) could the product inventory be directly updated. These specific use cases as inventory correction – are not part of this document as they are related to Seller internal processes.

The following table provides an example to clarify this synchronization:

Date	Event	ProductOrder	ProductInventory
Before			An UNI is existing (Id: 14789)– status Active  An ENNI is existing (id: 9963) – status active
09/07/2017	SP1 wants a new E-Line between UNI1 and ENNI		He uses GET product to retrieving product Id
		SP1 post an order for this E-Line between UNI14789 and ENNI9963	
		SP2 receives the order. This is the order #1  Order #1 in	



		Acknowledged status	
		SP2 begin delivery  Order #1 in ‘InProgress’ status	A E-Line product is created in the inventory in status pendingActive. E-Line Product has the id 55664
	E-Line is build!	Order #1 in ‘Completed’ status	The E-Line product 55664 is active status
11/8/2017	SP1 want to change E-Line bandwidth speed		He uses GET product to retrieving product Id (55664)
		SP1 post an order to modify E-Line 55664	
		SP2 receives the order. This is the order #9  Order #9 in Acknowledged status  <b>Note that order #9 has not any link with order #1 !!</b>	
		SP2 begin delivery  Order #9 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in Active  <i>(Sometime we use a pendingModification status – not in TMF state engine)</i>
	E-Line speed is modified	Order #9 in ‘Completed’ status	The E-Line product 55664 is active status with new attribute updated
5/7/2018	SP1 wants to suspend his E-Line		He uses GET product to retrieving product Id (55664)
		SP1 post an order to suspend E-Line55664	
		SP2 receives the order. This is the order #48  Order #48 in	

		Acknowledged status Note that order #48 has not any link with orders #1 or #9	
		SP2 begin delivery  Order #48 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in Active <i>(Sometime we use a pendingsuspension status – not in TMF state engine)</i>
	E-Line is suspended	Order #48 in ‘Completed’ status	The E-Line product 55664 is in suspended status
5/13/2018	SP1 wants to reactivate his E-Line		He uses GET product to retrieving product Id (β 55664)
		SP1 post an order to reactivate E-Line 55664	
		SP2 receives the order. This is the order #52  Order #52 in Acknowledged status	
		SP2 begin delivery  Order #52 in ‘InProgress’ status	The E-Line product #55664 in the inventory stays in suspended status
	E-Line is active	Order #52 in ‘Completed’ status	The E-Line product 55664 is in active status
.....			

Table 3 – Product Order Example

## 9.2 Relationship between Order Item Status and Product Status

There is a relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “INSTALL”, as described in the table below:

Order Item Status	Product Status	Comment
Acknowledged	NA	No product available through the API
Rejected	NA	No product available through the API
Pending	NA	No product available through the API
AssessingCancellation	NA	No product available through the API
PendingCancellation	NA	No product available through the API
Cancelled	NA	No product available through the API
InProgress	NA	No product available through the API
Configured	NA	No product available through the API
Held	NA	No product available through the API
Failed	NA	No product available through the API
Completed	Active	

Table 4 – Install Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “SUSPEND”. It is assumed the “Product” status is in an “Active” state when the order to suspend the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	Active	
AssessingCancellation	Active	
PendingCancellation	Active	
Cancelled	Active	

InProgress	Active	
Configured	Active	
Held	Active	
Failed	Active	The product order has failed so the product remains in an active status.
Completed	Suspended	Product is successfully suspended.

Table 5 – Suspend Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” on a product order is “REACTIVATE”. It is assumed the “Product” status is in a “Suspended” state when the order to suspend the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Suspended	Product already suspended
Rejected	Suspended	Rejection of product order item doesn’t change product status.
Pending	Suspended	
AssessingCancellation	Suspended	
PendingCancellation	Suspended	
Cancelled	Suspended	
InProgress	Suspended	
Configured	Suspended	
Held	Suspended	
Failed	Suspended	The product order item has failed so the product remains in Suspended status.
Completed	Active	Request to reactivate a previously suspended product is successful.

Table 6 – Reactivate Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” is “DISCONNECT”. It is assumed the “Product” status is in an “Active” state when the order to disconnect the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	PendingTerminate	
AssessingCancellation	PendingTerminate	
PendingCancellation	PendingTerminate	
Cancelled	Active	
InProgress	PendingTerminate	
Configured	PendingTerminate	
Held	PendingTerminate	
Failed	Active	As the disconnect has failed, the product status remains Active
Completed	Terminated	Disconnect is successful.

Table 7 – Disconnect Product Order Relationship

The following table represents the relationship between the “OrderItem” status and the “Product” status when the “orderItemAction” is “CHANGE”. It is assumed the “Product” status is in an “Active” state when the order to change the product is received.

Order Item Status	Product Status	Comment
Acknowledged	Active	Product already active
Rejected	Active	Rejection of product order item doesn’t change product status.
Pending	PendingChange	
AssessingCancellation	PendingChange	
PendingCancellation	PendingChange	
Cancelled	Active	
InProgress	PendingChange	

Configured	PendingChange	
Held	PendingChange	
Failed	Active	<p>As the change has failed, the product status transitions from PendingChange to its previous status of Active and the change does not go into effect.</p> <p>Note: because the order item failed, the buyer received a notification of failure</p>
Completed	Active	<p>The product status transitions from PendingChange to Active and the change goes into effect.</p> <p>Note: because the order item was successful , the buyer received a notification of success</p>

Table 8 – Change Product Order Relationship