



Working Draft
MEF W137 v0.1

LSO Cantata and LSO Sonata Appointment Management API - Developer Guide

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

May 2022

Disclaimer

© MEF Forum 2022. All Rights Reserved.

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, specifications or recommendations 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.

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

Copyright

© MEF Forum 2022. Any reproduction of this document, or any portion thereof, shall contain the following statement: "Reproduced with permission of MEF Forum." No user of this document is authorized to modify any of the information contained herein.

Table of Contents

- [List of Contributing Members](#)
- [1. Abstract](#)
- [2. Terminology and Abbreviations](#)
- [3. Compliance Levels](#)
- [4. Introduction](#)
 - [4.1. Conventions in the Document](#)
 - [4.2. Relation to Other Documents](#)
 - [4.3. Approach](#)
 - [4.4. High-Level Flow](#)
- [5. API Description](#)
 - [5.1. High-level use cases](#)
 - [5.2. API Endpoint and Operation Description](#)
 - [5.2.1. Appointment API Endpoints](#)
 - [5.2.2. WorkOrder API Endpoints](#)
 - [5.3. Specifying the Buyer ID and the Seller ID](#)
 - [5.4. Model Structural Validation](#)
 - [5.5. Security Considerations](#)
- [6. API Interactions and Flows](#)
 - [6.1. Use case 7: Search Appointment Timeslot](#)
 - [6.1.1. Interaction flow](#)
 - [6.1.1. Create Appointment Timeslot - Request](#)
 - [6.1.2. Create Appointment Timeslot - Response](#)
 - [6.2. Use Case 8: Create Appointment](#)
 - [6.2.1. Interaction flow](#)
 - [6.2.2. Create Appointment - Request](#)
 - [6.2.3. Create Appointment - Response](#)
 - [6.2.4. Appointment - Lifecycle](#)
 - [6.3. Use Case 9: Retrieve Appointment List](#)
 - [6.4. Use Case 10: Retrieve Appointment by Appointment Identifier](#)
 - [6.5. Use Case 11: Patch Appointment by Appointment Identifier](#)
 - [6.6. Use case 12: Cancel Appointment by Appointment Identifier](#)
 - [6.7. Use Case 13: Retrieve WorkOrder List](#)
 - [6.8. Use Case 14: Retrieve WorkOrder by WorkOrder Identifier](#)
 - [6.9. Use case 19: Register for Appointment Notifications](#)
 - [6.10. Use case 20: Send Appointment Notification](#)
 - [6.11. Use case 21: Register for WorkOrder Notifications](#)
 - [6.12. Use case 22: Send WorkOrder Notification](#)
- [7. API Details](#)
 - [7.1. API patterns](#)
 - [7.1.1. Indicating errors](#)

- 7.1.2. Response pagination
- 7.2. Management API Data model
 - 7.2.1. Appointment
 - 7.2.2. WorkOrder
 - 7.2.3. Common
 - 7.2.4. Notification registration
- 7.3. Notification API Data Model
 - 7.3.1. Common Notification.
 - 7.3.2. Appointment Notification.
 - 7.3.3. Workorder Notification.
- 8. References

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.

Member

Table 1. Contributing Members

1. Abstract

This standard is intended to assist implementation of the Appointment functionality defined for the LSO Cantata and LSO Sonata Interface Reference Points (IRPs), for which requirements and use cases are defined in MEF 113 *Trouble Ticketing Requirements and Use Cases* [MEF113]. This standard consists of this document and complementary API definitions for:

- Appointment Management and Appointment Notification.
- WorkOrder Management and WorkOrder Notification.

This standard normatively incorporates the following files by reference as if they were part of this document, from the GitHub repository

<https://github.com/MEF-GIT/MEF-LSO-Sonata-SDK>

- `/productApi/workforce/appointment/appointmentManagement.api.yaml`
- `/productApi/workforce/appointment/appointmentNotification.api.yaml`
- `/productApi/workforce/appointment/workorderManagement.api.yaml`
- `/productApi/workforce/appointment/workorderNotification.api.yaml`

<https://github.com/MEF-GIT/MEF-LSO-Cantata-SDK>

- `/productApi/workforce/appointment/appointmentManagement.api.yaml`
- `/productApi/workforce/appointment/appointmentNotification.api.yaml`
- `/productApi/workforce/appointment/workorderManagement.api.yaml`
- `/productApi/workforce/appointment/workorderNotification.api.yaml`

The Appointment API is defined using OpenAPI 3.0 [OAS-V3]

2. Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions of terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling, in other MEF or external documents.

Term	Description	Reference
Application Program Interface (API)	In the context of LSO, API describes one of the Management Interface Reference Points based on the requirements specified in an Interface Profile, along with a data model, the protocol that defines operations on the data and the encoding format used to encode data according to the data model. In this document, API is used synonymously with REST API	[MEF55.1]
Appointment	In the context of this document, denotes an arrangement between the Buyer and Seller for a Seller Technician to meet with the Buyer at a specified time and place.	[TMF646]
Buyer	In the context of this document, denotes the organization or individual acting as the customer in a transaction over a Cantata (Customer <-> Service Provider) or Sonata (Service Provider <-> Partner) Interface	This document; adapted from [MEF80]
Incident	An entry within a Seller's tracking system created by the context of this document, denotes a situation that is not part of normal operation Seller, which contains information about a Situation in the Seller's network that has a possible negative impact on the operability of the network on a Product for one or more Buyers	[MEF113]
Issue	In the context of this document, denotes a problem with a Product as experienced by the Buyer that is not part of normal operation.	[MEF113]
Notification	A message sent from the Seller to the Buyer to inform about an event that has occurred in regard to a specific instance of a Ticket or an Incident	[MEF113]
Requesting Entity	The business organization that is acting on behalf of one or more Buyers. In the most common case, the Requesting Entity represents only one Buyer and these terms are then synonymous	[MEF79]
Responding Entity	The business organization that is acting on behalf of one or more Sellers. In the most common case, the Responding	[MEF79]

	Entity represents only one Seller and these terms are then synonymous	
REST API	Representational State Transfer. REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems.	[REST]
Seller	In the context of this document, denotes the organization acting as the supplier in a transaction over a Cantata (Customer <-> Service Provider) or Sonata (Service Provider <-> Partner) Interface	This document; adapted from [MEF80]
Search Time Slot	This task resource is used to retrieve available time slots. One of this available time slot is after used to create or reschedule an appointment.	[TMF646]
Situation	In the context of this document, denotes a problem that is not part of normal operation in the Seller's network that has a possible negative impact on the operability of a Product for one or more Buyers	[MEF113]
Ticket	An entry within a Seller's tracking system created by the Buyer (or a third party on behalf of the Buyer), which contains information about an Issue impacting normal operation of a Product, along with support interventions made by technical support staff, or third parties	[MEF113]
WorkOrder	In the context of this document, denotes a set of tasks to be confirmed and performed under the responsibility of a Technician at a given location	[MEF113]

3. Compliance Levels

The key words **"MUST"**, **"MUST NOT"**, **"REQUIRED"**, **"SHALL"**, **"SHALL NOT"**, **"SHOULD"**, **"SHOULD NOT"**, **"RECOMMENDED"**, **"NOT RECOMMENDED"**, **"MAY"**, and **"OPTIONAL"** in this document are to be interpreted as described in BCP 14 (RFC 2119 [\[RFC2119\]](#), RFC 8174 [\[RFC8174\]](#)) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD**

NOT) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words MAY or OPTIONAL) are labeled as **[Ox]** for optional.

A paragraph preceded by **[CRa]<** specifies a conditional mandatory requirement that **MUST** be followed if the condition(s) following the "<" have been met. For example, "**[CR1]<[D38]**" indicates that Conditional Mandatory Requirement 1 must be followed if Desirable Requirement 38 has been met. A paragraph preceded by **[CDb]<** specifies a Conditional Desirable Requirement that **SHOULD** be followed if the condition(s) following the "<" have been met. A paragraph preceded by ****[COc]<**** specifies a Conditional Optional Requirement that **MAY** be followed if the condition(s) following the "<" have been met.

4. Introduction

The Appointment and WorkOrder API allow the Buyer to create, retrieve, and update Appointment and WorkOrder as well as receive notifications and updates. This allows managing issues and situations that are not part of normal operations of the Product provided by the Seller.

This standard specification document describes the Application Programming Interface (API) for Appointment and WorkOrder functionality of the LSO Cantata Interface Reference Point (IRP) and LSO Sonata IRP as defined in the *MEF 55.1 Lifecycle Service Orchestration (LSO): Reference Architecture and Framework* [MEF55.1]. The LSO Reference Architecture is shown in Figure 1 with both IRPs highlighted.

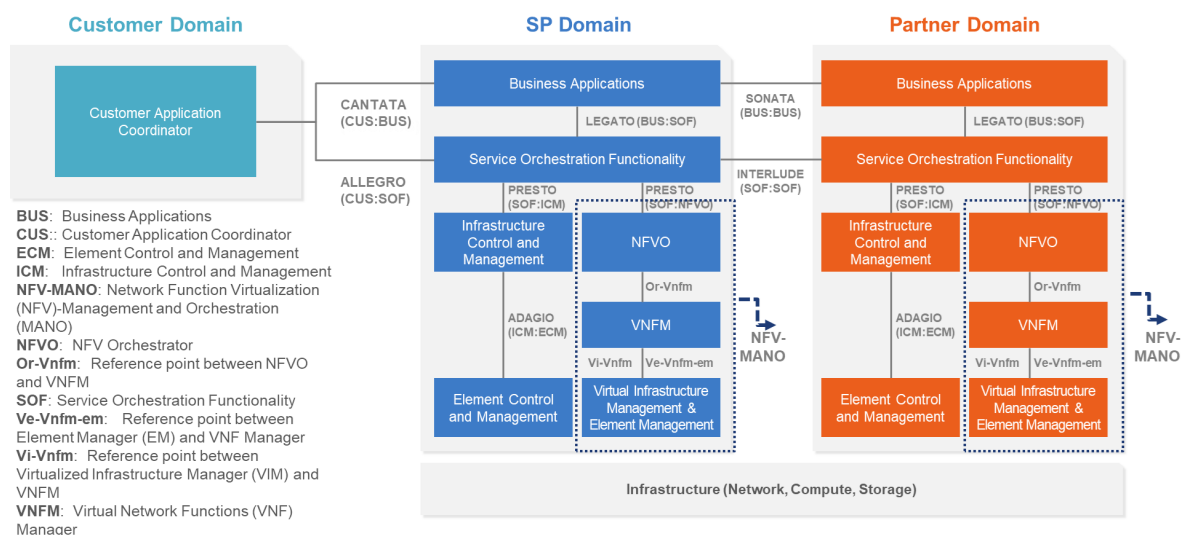


Figure 1. The LSO Reference Architecture

Cantata and Sonata IRPs define pre-ordering and ordering functionalities that allow an automated exchange of information between business applications of the Buyer (Customer or Service Provider) and Seller (Service Provider or Partner) Domains. Those are:

- Product Catalog
- Address Validation
- Site Retrieval
- Product Offering Qualification
- Product Quote
- Product Inventory
- Product Ordering
- Trouble Ticketing
- Appointment
- WorkOrder
- Billing

The business requirements and use cases for Appointment and WorkOrder are defined in MEF W113 *Trouble Ticketing Requirements and Use Cases* [MEF113]. MEF W113 defines use cases that cover Trouble Ticket, Incident, Appointment and WorkOrder. This API and Developer Guide covers the Appointment and WorkOrder related use cases. The Trouble Ticket and Incident use cases are covered by MEF124 LSO Cantata and LSO Sonata Trouble Ticket Management API - Developer Guide.

This document is structured as follows:

- [Chapter 4](#) provides an introduction to Appointment and WorkOrder and its description in a broader context of Cantata and Sonata and their corresponding SDKs.
- [Chapter 5](#) gives an overview of endpoints, resource model and design patterns.
- Use cases and flows are presented in [Chapter 6](#).
- And finally, [Chapter 7](#) complements previous sections with a detailed API description.

4.1. Conventions in the Document

- Code samples are formatted using code blocks. When notation `<< some text >>` is used in the payload sample it indicates that a comment is provided instead of an example value and it might not comply with the OpenAPI definition.
- Model definitions are formatted as in-line code (e.g. `Appointment`).
- In UML diagrams the default cardinality of associations is `0..1`. Other cardinality markers are compliant with the UML standard.
- In the API details tables and UML diagrams required attributes are marked with a `*` next to their names.
- In UML sequence diagrams `{{variable}}` notation is used to indicate a variable to be substituted with a correct value.

4.2. Relation to Other Documents

This API implements the Appointment and WorkOrder related requirements and use cases that are defined in [MEF113]. The API definition builds on *TMF646 Appointment Management API REST Specification R19.0.1* [TMF646] and *TMF697 Work Order Management API* [TMF697]. It is also related to *Cantata and LSO Sonata Trouble Ticket Management API - Developer Guide* [MEF124], which covers outstanding Trouble Ticket use cases are described in MEF 113.

4.3. Approach

As presented in Figure 2, both Cantata and Sonata API frameworks consists of three structural components:

- Generic API framework
- Product-independent information (Function-specific information and Function-specific operations)
- Product-specific information (MEF product specification data model)

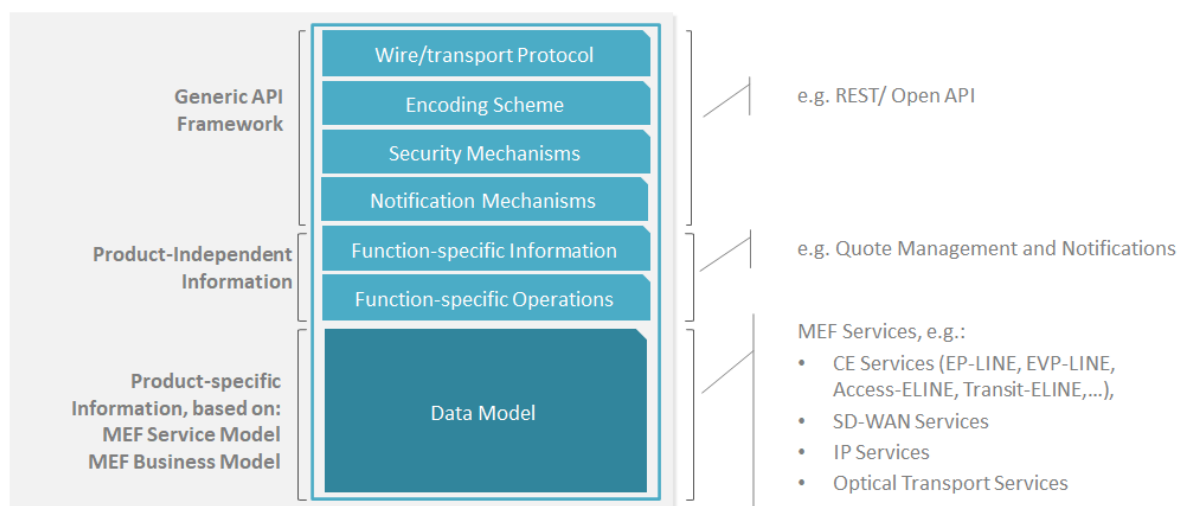


Figure 2. Cantata and Sonata API framework

The essential concept behind the framework is to decouple the common structure, information and operations from the specific product information content.

Firstly, the Generic API Framework defines a set of design rules and patterns that are applied across all Cantata or Sonata APIs.

Secondly, the product-independent information of the framework focuses on a model of a particular Cantata or Sonata functionality and is agnostic to any of the product specifications.

Finally, the product-specific information part of the framework focuses on MEF product specifications that define business-relevant attributes and requirements for trading MEF subscriber and MEF operator services.

The Appointment and WorkOrder are product-agnostic in their nature and are not intended to carry any product-specific payloads. They only reference product from the inventory by

id. It operates using the Generic API Framework and the Function-specific Information and Operations.

4.4. High-Level Flow

The Appointment and WorkOrder are part of a broader Cantata and Sonata End-to-End flow. Figure 3. below shows a high-level diagram to get a good understanding of the whole process and Appointment and WorkOrder position within it.

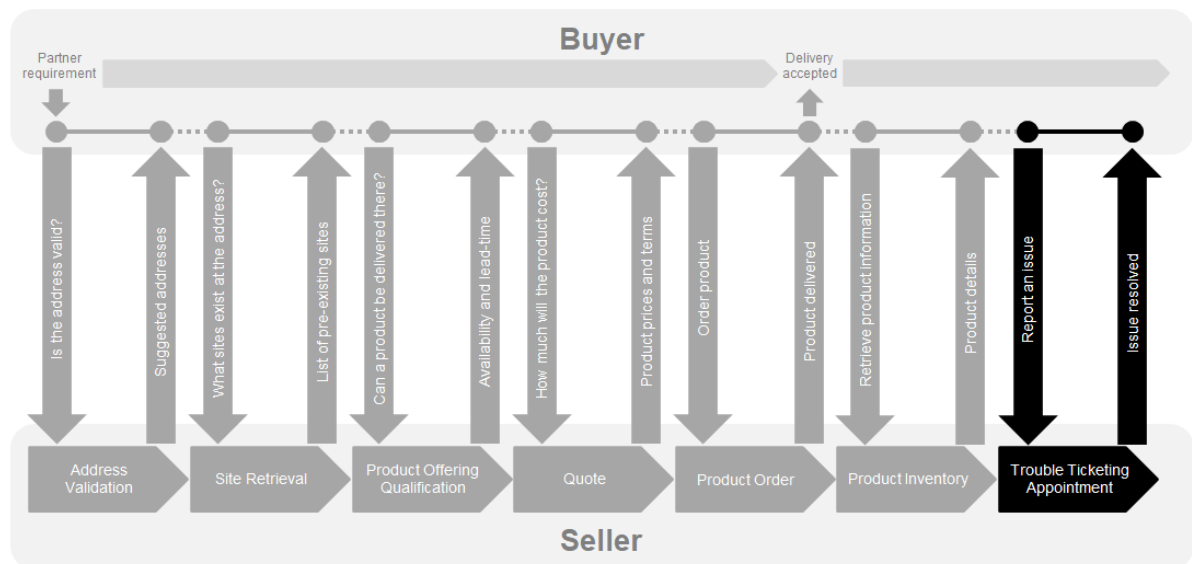


Figure 3. Cantata and Sonata End-to-End Function Flow

- Address Validation:
 - Allows the Buyer to retrieve address information from the Seller, including exact formats, for addresses known to the Seller.
- Site Retrieval:
 - Allows the Buyer to retrieve Geographic Site information including exact formats for Geographic Sites known to the Seller.
- Product Offering Qualification (POQ):
 - Allows the Buyer to check whether the Seller can deliver a product or set of products from among their product offerings at the geographic address or a Geographic Site specified by the Buyer; or modify a previously purchased product.
- Quote:
 - Allows the Buyer to submit a request to find out how much the installation of an instance of a Product Offering, an update to an existing Product, or a disconnect of an existing Product will cost.
- Product Order:
 - Allows the Buyer to request the Seller to initiate and complete the fulfillment process of an installation of a Product Offering, an update to an existing Product, or a disconnect of an existing Product at the address defined by the Buyer.
- Product Inventory:

- Allows the Buyer to retrieve the information about existing Product instances from Seller's Product Inventory.
- The Appointment and WorkOrder:
 - Allows the Buyer to create, retrieve, and update Appointment and WorkOrder as well as receive notifications about Appointment and WorkOrder' updates. This allows managing issues and situations that are not part of normal operations of the Product provided by the Seller. It also allows for WorkOrder and appointment management.

5. API Description

This section presents the API structure and design patterns. It starts with the high-level use cases diagram. Then it describes the REST endpoints with use case mapping. Next, it gives an overview of the API resource model.

5.1. High-level use cases

Figure 4 presents a high-level use case diagram as specified in MEF 113 [[MEF113](#)] in section 7. This picture aims to help understand the endpoint mapping. Use cases are described extensively in [chapter 6](#).

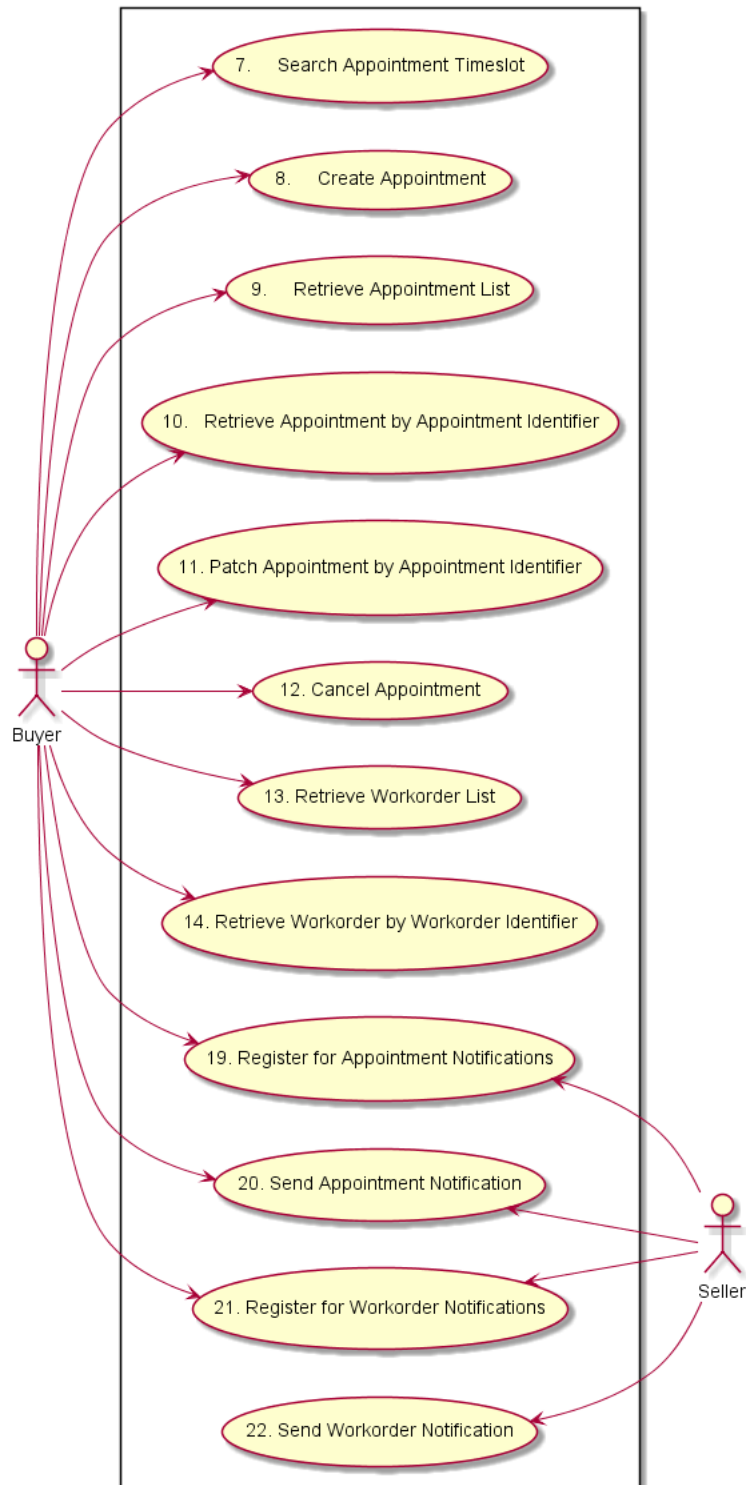


Figure 4. Use cases

5.2. API Endpoint and Operation Description

5.2.1. Appointment API Endpoints

5.2.1.1. Seller side Appointment API Endpoints

Base URL for Cantata: https://{serverBase}:{port}/{seller_prefix}/mefApi/cantata/appointment/v1/

Base URL for Sonata: https://{serverBase}:{port}/{seller_prefix}/mefApi/sonata/appointment/v1/

The following API endpoints are implemented by the Seller and allow the Buyer to create, retrieve, modify Appointment and register for Notifications. The endpoints and corresponding data model are defined in:

[/productApi/workforce/appointment/appointmentManagement.api.yaml](#).

Appointment Use cases:

API endpoint	Description	MEF 113 Use Case mapping
POST /searchTimeSlot	A request by the Buyer to find a set of available time slots for scheduling or rescheduling an Appointment for a WorkOrder with a Seller Technician.	UC 7: Search Appointment Timeslot
POST /appointment	A request by the Buyer to create an Appointment for a WorkOrder with a Seller Technician.	UC 8: Create Appointment
GET /appointment	The Buyer requests a list of Appointments from the Seller based on a set of specified filter criteria. The Seller returns a summarized list of Appointments.	UC 9: Retrieve Appointment List
GET /appointment/{{id}}	The Buyer requests detailed information about a single Appointment based on an Appointment Identifier.	UC 10: Retrieve Appointment by Appointment Identifier
PATCH /appointment/{{id}}	A request by the Buyer to patch or reschedule an Appointment for a WorkOrder with a Seller Technician.	UC 11: Patch Appointment by Appointment Identifier
POST /appointment/{{id}}/cancel	A request by the Buyer to cancel an Appointment for a WorkOrder with a Seller Technician.	UC 12: Cancel Appointment by Appointment Identifier
POST /hub	The Buyer requests to subscribe to notifications.	UC 19: Register for Appointment Notifications
GET /hub/{{id}}	A request initiated by the Buyer to retrieve the details of the notification subscription.	UC 19: Register for Appointment Notifications

API endpoint	Description	MEF 113 Use Case mapping
DELETE /hub/{{id}}	A request initiated by the Buyer to instruct the Seller to stop sending notifications.	UC 19. Register for Appointment Notifications

Table 2. Seller side mandatory Appointment API endpoints

[R1] The Seller **MUST** support Appointment API endpoints listed in Table 2. [MEF113 R3]

5.2.1.1. Buyer side Appointment API Endpoints

Base URL for Cantata: `https://{{serverBase}}:{{port}}`

`{{?/buyer_prefix}}/mefApi/cantata/appointmentNotification/v1/`

Base URL for Sonata: `https://{{serverBase}}:{{port}}`

`{{?/buyer_prefix}}/mefApi/sonata/appointmentNotification/v1/`

The following API Endpoints are used by the Seller to post notifications to registered listeners. The endpoints and corresponding data model are defined in

`/productApi/workforce/appointment/appointmentNotification.api.yaml.`

Appointment Use cases

API endpoint	Description	MEF 113 Use Case mapping
POST /listener/appointmentStatusChangeEvent	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following Appointment Notification Types have occurred: <code>AppointmentStatusChangeEvent</code>	UC 20. Send Appointment Notification
POST /listener/appointmentAttributeValueChangeEvent	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following Appointment Notification Types have occurred: <code>AppointmentAttributeValueChangeEvent</code>	UC 20. Send Appointment Notification

Table 3. Buyer side mandatory Appointment API endpoints

[R2] The Buyer **MUST** support Appointment API endpoints listed in Table 3. [MEF113 R3]

5.2.2. WorkOrder API Endpoints

5.2.2.1. Seller side WorkOrder API Endpoints

Base URL for Cantata: `https://{{serverBase}}:{{port}}`

`{{?/seller_prefix}}/mefApi/cantata/workOrderManagement/v1/`

Base URL for Sonata: `https://{{serverBase}}:{{port}}`

`{{?/seller_prefix}}/mefApi/sonata/workOrderManagement/v1/`

The following API endpoints are implemented by the Seller and allow the Buyer to retrieve WorkOrders and register for Notifications. The endpoints and corresponding data model are defined in:

`/productApi/workforce/workorder/workorderManagement.api.yaml.`

WorkOrder API Use cases:

API endpoint	Description	MEF 113 Use Case mapping
<code>GET /workOrder</code>	The Buyer requests a list of WorkOrders from the Seller based on a set of specified filter criteria. The Seller returns a summarized list of WorkOrders.	UC 13. Retrieve WorkOrder List
<code>GET /workOrder/{{id}}</code>	The Buyer requests detailed information about a WorkOrder based on a WorkOrder Identifier.	UC 14. Retrieve WorkOrder by WorkOrder Identifier
<code>POST /hub</code>	The Buyer requests to subscribe to notifications.	UC 21: Register for WorkOrder Notifications
<code>GET /hub/{{id}}</code>	A request initiated by the Buyer to retrieve the details of the notification subscription.	UC 21: Register for WorkOrder Notifications
<code>DELETE /hub/{{id}}</code>	A request initiated by the Buyer to instruct the Seller to stop sending notifications.	UC 21. Register for WorkOrder Notifications

Table 4. Seller side mandatory WorkOrder API endpoints

[R3] The Seller **MUST** support Workorder API endpoints listed in Table 4. [MEF113 R4]

5.2.2.2. Buyer side WorkOrder API Endpoints

Base URL for Cantata: `https://{{serverBase}}:{{port}}`

`{{?/buyer_prefix}}/mefApi/cantata/workOrderNotification/v1/`

Base URL for Sonata: `https://{{serverBase}}:{{port}}`

`{{?/buyer_prefix}}/mefApi/sonata/workOrderNotification/v1/`

The following API endpoints are implemented by the Seller and allow the Buyer to register and send WorkOrder Notifications. The endpoints and corresponding data model are defined in:

`/productApi/workforce/appointment/workOrderNotification.api.yaml.`

WorkOrder API Use cases

API endpoint	Description	MEF 113 Use Case mapping
<code>POST /listener/workOrderCreateEvent</code>	The The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following WorkOrder Notification Types has occurred: <code>workOrderCreateEvent</code>	UC 22. Send WorkOrder Notification
<code>POST /listener/workOrderAttributeValueChangeEvent</code>	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following WorkOrder Notification Types has occurred: <code>workOrderStateChangeEvent</code>	UC 22. Send WorkOrder Notification
<code>POST /listener/workOrderAppointmentRequiredEvent</code>	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following WorkOrder Notification Types has occurred: <code>workOrderAppointmentRequiredEvent</code>	UC 22. Send WorkOrder Notification

Table 5. Buyer side mandatory WorkOrder API endpoints

[R4] The Buyer **MUST** support WorkOrder API endpoints listed in Table 5. [MEF113 R4]

5.3. Specifying the Buyer ID and the Seller ID

A business entity willing to represent multiple Buyers or multiple Sellers must follow requirements of MEF 79 [MEF79] chapter 8.8, which states:

For requests of all types, there is a business entity that is initiating an Operation (called a Requesting Entity) and a business entity that is responding to this request

(called the Responding Entity). In the simplest case, the Requesting Entity is the Buyer and the Responding Entity is the Seller. However, in some cases, the Requesting Entity may represent more than one Buyer and similarly, the Responding Entity may represent more than one Seller.

While it is outside the scope of this specification, it is assumed that the Requesting Entity and the Responding Entity are aware of each other and can authenticate requests initiated by the other party. It is further assumed that both the Buying Entity and the Requesting Entity know:

- a) the list of Buyers the Requesting Entity represents when interacting with this Responding Entity; and
- b) the list of Sellers that this Responding Entity represents to this Requesting Entity.

In the API the `buyerId` and `sellerId` are represented as query parameters in each operation defined in `appointmentManagement.api.yaml` and as attributes of events as described in `appointmentNotification.api.yaml`.

[R5] If the Requesting Entity has the authority to represent more than one Buyer the request **MUST** include `buyerId` query parameter that identifies the Buyer being represented. [MEF79 R80]

[R6] If the Requesting Entity represents precisely one Buyer with the Responding Entity, the request **MUST NOT** specify the `buyerId`. [MEF79 R81]

[R7] If the Responding Entity represents more than one Seller to this Buyer the request **MUST** include `sellerId` query parameter that identifies the Seller with whom this request is associated. [MEF79 R82]

[R8] If the Responding Entity represents precisely one Seller to this Buyer, the request **MUST NOT** specify the `sellerId`. [MEF79 R83]

[R9] If `buyerId` or `sellerId` attributes were specified in the request same attributes **MUST** be used in the notification payload.

5.4. Model Structural Validation

The structure of the HTTP payloads exchanged via The Appointment and WorkOrder API endpoints is defined using OpenAPI version 3.0.

[R10] Implementations **MUST** use payloads that conform to these definitions.

5.5. Security Considerations

There must be an authentication mechanism whereby a Seller can be assured who a Buyer is and vice-versa. There must also be authorization mechanisms in place to control what a particular Buyer or Seller is allowed to do and what information may be obtained. However, the definition of the exact security mechanism and configuration is outside the scope of this document. It is being worked on by a separate MEF Project (MEF W128).

6. API Interactions and Flows

This section provides a detailed insight into the API functionality, use cases, and flows. It starts with Figure 5 and Table 6, presenting a list and short description of all business use cases then presents the variants of end-to-end interaction flows, and in following subchapters describes the API usage flow and examples for each of the use cases.

MEF 113 defines use cases related to three domains:

- Trouble Ticket
- WorkOrder
- Appointment

Figure 5 presents an example of an end-to-end flow that shows dependencies between all the domains:

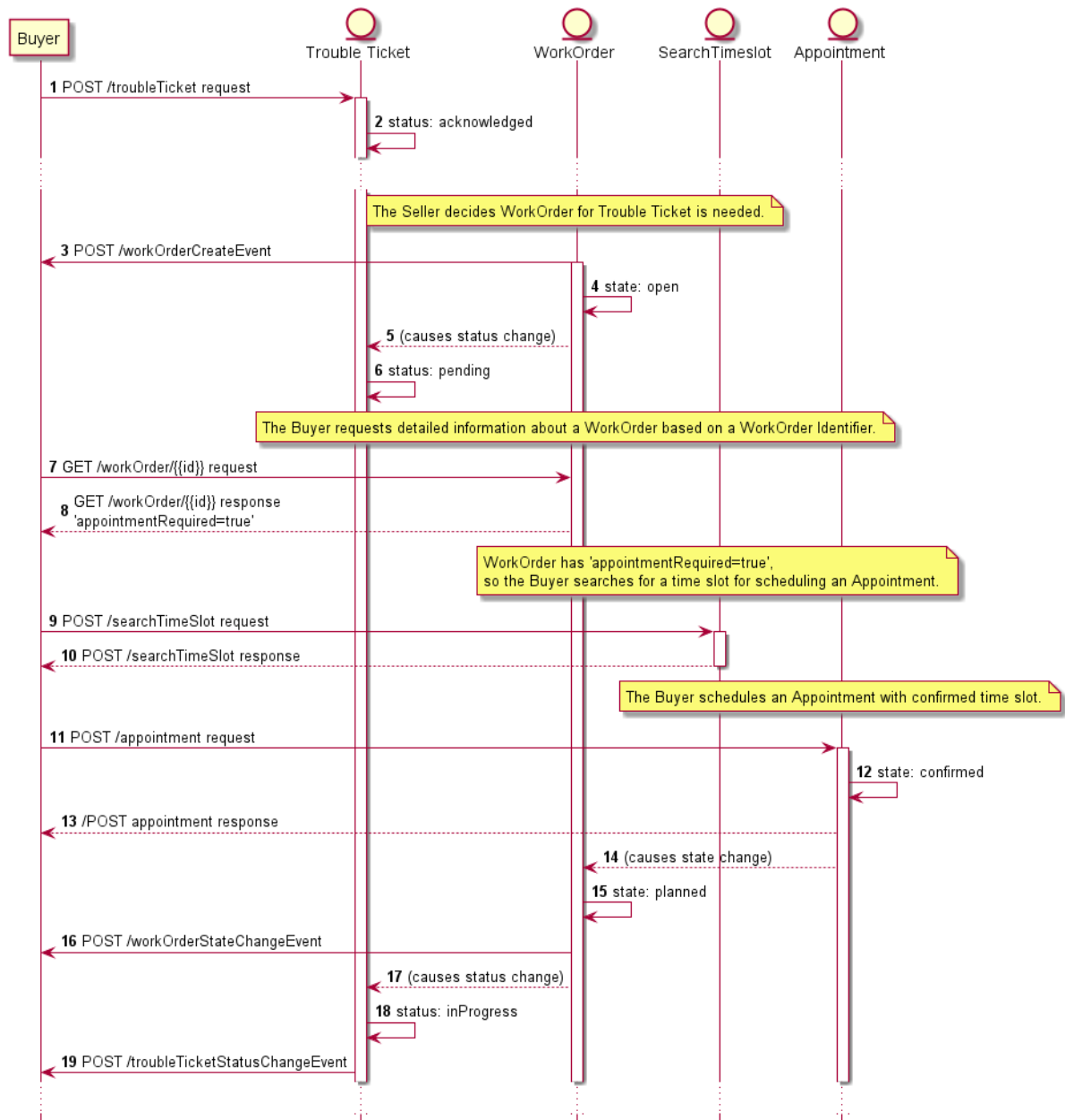


Figure 5. End-to-End API Flows

- (1) The Buyer experiences the issue in the network and creates the Trouble Ticket.
- (2) The Seller creates the Trouble Ticket and sets the status: **acknowledged**.
- The Seller decides that a WorkOrder with Appointment is needed to resolve the issue.
- The Seller creates a Workorder in state **open** (4) and sends a **workOrderCreateEvent** (3).
- (7-8) The Buyer requests detailed information about the WorkOrder.
- (9) The Buyer proposes time slots for scheduling an Appointment, if the WorkOrder requires the Appointment (the parameter set to **appointmentRequired=true**)
- (10) The Seller responds with the list of available time slots.
- (11) The Buyer schedules an Appointment with agreed time slot.
- (12) The Buyer sets the Appointment status to **confirmed**.
- (14-15) Appointment creation causes the WorkOrder state change to **planned**
- (17-18) WorkOrder state change to **planned** causes the Trouble Ticket status change back to **in_progress**.

Use Case #	Use Case Name	Use Case Description
7	Search Appointment Timeslot	A request by the Buyer to find a set of available time slots for scheduling or rescheduling an Appointment for a WorkOrder with a Seller Technician.
8	Create Appointment List	A request by the Buyer to create an Appointment for a WorkOrder with a Seller Technician.
9	Retrieve Appointment List	The Buyer requests a list of Appointments from the Seller based on a set of specified filter criteria. The Seller returns a summarized list of Appointments.
10	Retrieve Appointment by Appointment Identifier	The Buyer requests detailed information about a single Appointment based on an Appointment Identifier.
11	Patch Appointment by Appointment Identifier	A request by the Buyer to patch or reschedule an Appointment for a WorkOrder with a Seller Technician.
12	Cancel Appointment by Appointment Identifier	A request by the Buyer to cancel an Appointment for a WorkOrder with a Seller Technician.
13	Retrieve WorkOrder List	The Buyer requests a list of Work orders from the Seller based on a set of specified filter criteria. The Seller returns a summarized list of Work orders.
14	Retrieve WorkOrder by WorkOrder Identifier	The Buyer requests detailed information about a WorkOrder based on a WorkOrder Identifier.
19	Register for Appointment Notifications	The Buyer requests to subscribe to Appointment Notifications.

Use Case #	Use Case Name	Use Case Description
20	Send Appointment Notification	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following Appointment Notification Types has occurred: <code>appointmentAttributeValueChangeEvent</code> , <code>appointmentStatusChangeEvent</code>
21	Register for WorkOrder Notifications	The Buyer requests to subscribe to WorkOrder Notifications.
22	Send WorkOrder Notification Identifier	The Seller sends a notification regarding an Appointment to the Buyer indicating one of the following WorkOrder Notification Types has occurred: <code>workOrderCreateEvent</code> , <code>workOrderStateChangeEvent</code> , <code>workOrderAppointmentRequiredEvent</code>

Table 6. Use cases description

The detailed business requirements of each of the use cases are described in sections 7 and 8 of MEF 113 [MEF113].

6.1. Use case 7: Search Appointment Timeslot

This is the initial step for Appointment processing.

6.1.1. Interaction flow

The flow of this use case is very simple and is described in Figure 6.

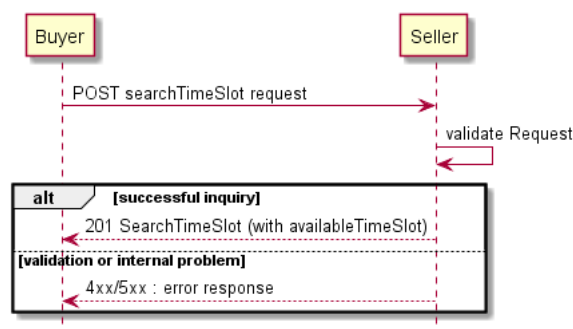


Figure 6. Use Case 7: Appointment Timeslot create request flow

The Buyer sends a request with a `SearchTimeSlot_Create` type in the body.

6.1.1. Create Appointment Timeslot - Request

Figure 7 presents the data model of the Appointment Timeslot. The model of the request message (`SearchTimeSlot_Create`) is a subset of the `SearchTimeSlot` model and contains only attributes that can (or must) be set by the Buyer. The Seller then enriches the entity in the response with additional information.

The full list of attributes is available in [Section 7](#) and in the API specification which is an integral part of this standard.

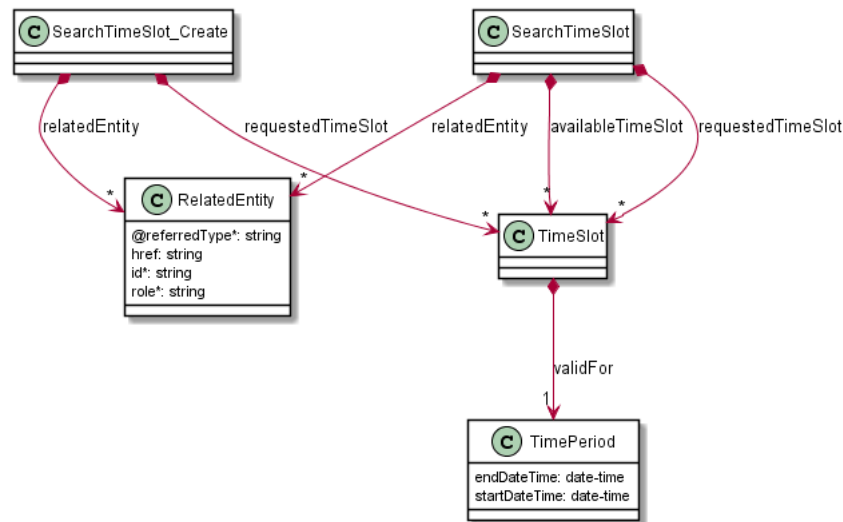


Figure 7. Create Appointment Timeslot Model

The snippet below presents an example of the Create Appointment Timeslot Request:

`SearchTimeSlot` Create

```

{
  "requestedTimeSlot": [
    {
      "validFor": {
        "startDateTime": "2022-05-19T08:00:00.170Z",
        "endDateTime": "2022-05-21T16:00:00.170Z"
      }
    }
  ],
  "relatedEntity": [
    {
      "id": "01494079-6c79-4a25-83f7-48284196d44d",
      "role": "parentWorkOrder",
      "@referredType": "WorkOrder"
    }
  ]
}

```

[R11] The Buyer's request **MUST** include the following attributes: [MEF113 R78]

- `relatedEntity`
- `requestedTimeSlot`

6.1.2. Create Appointment Timeslot - Response

The Seller responds with a `SearchTimeSlot` type, which adds some attributes to the `SearchTimeSlot_Create` that was used in the Buyer's request.

Note: The term "Seller Response Code" used in the Business Requirements maps to HTTP response code, where `2xx` indicates *Success* and `4xx` or `5xx` indicate *Failure*.

The following snippet presents the Seller's response. It has the same structure as in the retrieve by identifier operation.

```
{
  "availableTimeSlot": [
    {
      "validFor": {
        "startDateTime": "2022-05-21T08:00:00.125Z",
        "endDateTime": "2022-05-21T16:00:00.125Z"
      }
    }
  ],
  "requestedTimeSlot": [
    {<< as provided by the Buyer >>
      "validFor": {
        "startDateTime": "2022-05-19T08:00:00.170Z",
        "endDateTime": "2022-05-21T16:00:00.170Z"
      }
    }
  ],
  "relatedEntity": [
    {<< as provided by the Buyer >>
      "id": "01494081-6c79-4a25-42f7-48284196d55d",
      "role": "parentWorkOrder",
      "@referredType": "WorkOrder"
    }
  ]
}
```

[R12] The Seller **MUST** return an error `Error422` if any of the included attributes in the Buyer's request are invalid. [MEF113 R81]

[R13] The Seller's response **MUST** return an empty list of `availableTimeSlot` if no Seller resources are available for an appointment that falls within any of the `requestedTimeSlot`. [MEF113 R84]

[R14] The Seller's response **MUST** include one or more `availableTimeSlot`, if Seller resources are available for an appointment that falls within the `requestedTimeSlot`. [MEF113 R85]

6.2. Use Case 8: Create Appointment

This is the initial step for Appointment processing.

6.2.1. Interaction flow

The flow of this use case is very simple and is described in Figure 8.

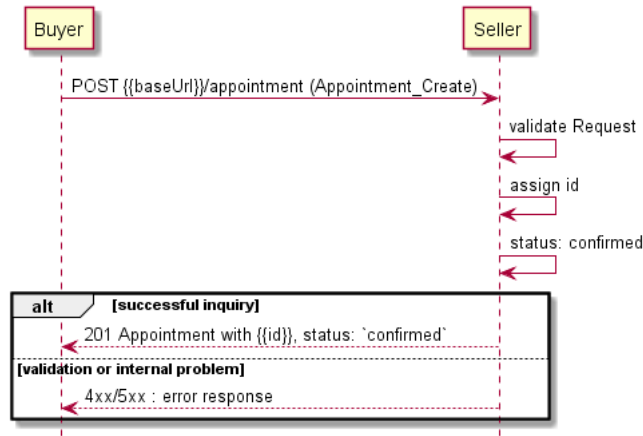


Figure 8. Use Case 8: Appointment create request flow

The Buyer sends a request with a `Appointment_Create` type in the body. The Seller performs request validation, assigns an `id`, and returns `Appointment` type in the response body, with a `status` set to `confirmed`. From this point, the Appointment is ready for further processing. The Buyer must track the progress of the process by subscribing for notifications (see [chapter 6.9](#)). The flow example with the use of Notifications is presented in Figure 9.

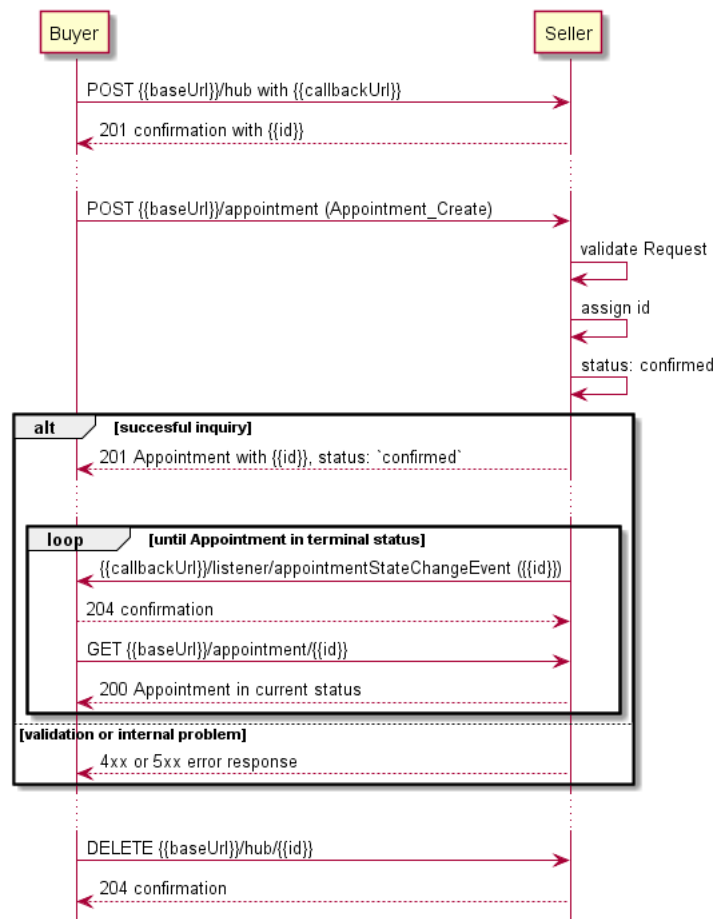


Figure 9. Appointment progress tracking - Notifications

Note: The context of notifications is not a part of the considered use case itself. It is presented to show the big picture of end-to-end flow. This applies also to all further use case flow diagrams with notifications.

6.2.2. Create Appointment - Request

Figure 10 presents the data model of the Appointment. The model of the request message (`Appointment_Create`) is a subset of the `Appointment` model and contains only attributes that can (or must) be set by the Buyer. The Seller then enriches the entity in the response with additional information.

The full list of attributes is available in [Section 7](#) and in the API specification which is an integral part of this standard.

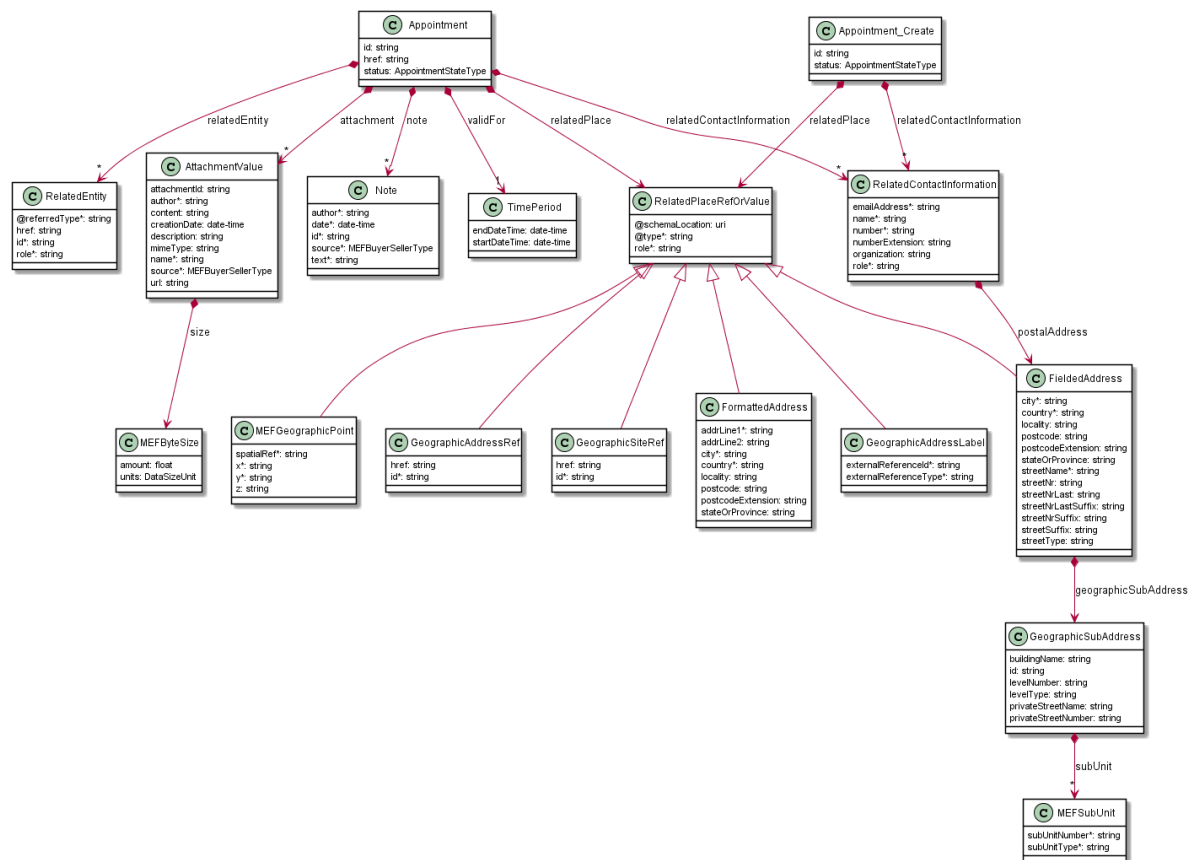


Figure 10. Create Appointment Model

The snippet below presents an example of the Create Appointment Request:

Appointment Create

```
{
  "relatedContactInformation": [
    {
      "emailAddress": "john.example@example.com",
      "name": "John Example",
      "number": "+12-345-678-90",
      "organization": "Buyer Example Co.",
      "role": "buyerAppointmentContact"
    }
  ],
  "relatedPlace": [
    {
      "@type": "GeographicAddressRef",
      "id": "00000000-0000-0030-0305-873500002010",
      "role": "APPOINTMENT_LOCATION"
    }
  ]
}
```

```

    }
  ],
  "validFor": {<< as provided in availableTimeSlot >>
    "startDateTime": "2022-05-21T08:00:00.125Z",
    "endDateTime": "2022-05-21T16:00:00.125Z"
  },
  "workOrder": [
    {
      "id": "01494081-6c79-4a25-42f7-48284196d55d"
    }
  ]
},
"attachment": [
  {
    "attachmentId": "att-001",
    "author": "John Example",
    "creationDate": "2022-05-19T10:04:18.812Z",
    "description": "Print screen from the assurance system",
    "mimeType": "image/jpeg",
    "name": "Alarm",
    "size": {
      "amount": 5.3,
      "units": "MBYTES"
    }
  }
],
"note": [
  {<< as provided by the Buyer >>
    "id": "note-1",
    "author": "John Example",
    "date": "2021-06-02T14:25:11.090Z",
    "source": "buyer",
    "text": "Couldn't reach the support on phone."{
  ]
}

```

[R15] The Buyer's request **MUST** include the following attributes: [MEF113 R86]

- `relatedContactInformation` - item with `role=buyerAppointmentContact`
- `relatedPlace`
- `workOrder`
- `validfor`

[O1] The Buyer's request **MAY** include the `note` and `attachment` attributes: [MEF113 O16]

6.2.3. Create Appointment - Response

The Seller responds with a `Appointment` type, which adds some attributes to the `Appointment_Create` that was used in the Buyer's request.

```

{
  "id": "00000000-4444-5555-6666-000000000987",
  "href": "{baseUri}/appointment/00000000-4444-5555-6666-000000000987",
  "attachment": [
    {
      "attachmentId": "att-001",
      "author": "John Example",
      "creationDate": "2022-05-19T10:04:18.812Z",
      "description": "Print screen from the assurance system",
      "mimeType": "image/jpeg",
      "name": "Alarm",
      "size": {
        "amount": 5.3,
        "units": "MBYTES"
      },
      "source": "buyer",
      "url": "https://example.com/documents/00000000-0000-1111-2222-000000001111"
    }
  ]
}

```

```

],
"note": [
  {<< as provided by the Buyer >>
    "id": "note-1",
    "author": "John Example",
    "date": "2021-06-02T14:25:11.090Z",
    "source": "buyer",
    "text": "Couldn't reach the support on phone."
  }
],
"workOrder": [
  {<< as provided by the Buyer >>
    "id": "01494081-6c79-4a25-42f7-48284196d55d"
  }
],
"relatedPlace": {
  "@type": "GeographicAddressRef",
  "id": "00000000-0000-0030-0305-873500002010",
  "role": "APPOINTMENT_LOCATION"
},
"status": "confirmed",
"validFor": {
  "startDateTime": "2022-05-21T08:00:00.125Z",
  "endDateTime": "2022-05-21T16:00:00.125Z"
},
"relatedContactInformation": [
  {<< as provided by the Buyer >>
    "emailAddress": "john.example@example.com",
    "name": "John Example",
    "number": "+12-345-678-90",
    "organization": "Buyer Example Co.",
    "role": "buyerAppointmentContact",
  }
]
}

```

The response to the create request does not contain all possible attributes.

[R16] The Seller **MUST** return an error `Error422` if the `validFor` is not valid. [MEF113 R90]

[R17] The Seller **MUST** return an error `Error422` if a Seller technician is not available for an appointment at the `relatedPlace` for the specified `requestedTimeSlot`. [MEF113 R91]

[R18] The Seller's response **MUST** include the Buyer following attributes. [MEF113 R92-R93]

- `id`
- `relatedContactInformation` - item with `role=buyerAppointmentContact`
- `relatedContactInformation` - item with `role=appointmentPlaceContact`
- `relatedContactInformation` - item with `role=sellerAppointmentContact`
- `relatedPlace`
- `status`
- `workOrder`
- `validFor`

[O2] The Seller's response **MAY** add the `note` and `attachment` attributes. [MEF113 O17]

[R19] The Seller **MUST** set the appointment `status` to `confirmed`. [MEF113 R91]

[R20] The Seller **MUST** return an error `Error422` if the `workOrder` references a workOrder that contains a reference to an appointment with `status` of `confirmed`. [MEF113 R95]

[R21] The Seller **MUST** return an error `Error422` if the related workOrder `state` is not `open`.

[MEF113 R96]

[R22] After creating the Appointment instance, the Seller **MUST** add the `appointment.id` to the `appointment` attribute of the related WorkOrder. [MEF113 R97]

[R23] The Seller **MUST** set the `appointmentRequired` attribute to `false` in the related WorkOrder.

[MEF113 R98]

[R24] The Seller **MUST** set the related WorkOrder `state` to `planned`. [MEF113 R99]

[R25] The Seller **MUST** update the `status` of the Trouble Ticket, for which this Appointment was created to `in_progress`. [MEF113 R100]

6.2.4. Appointment - Lifecycle

Figure 11 presents the Appointment state machine:

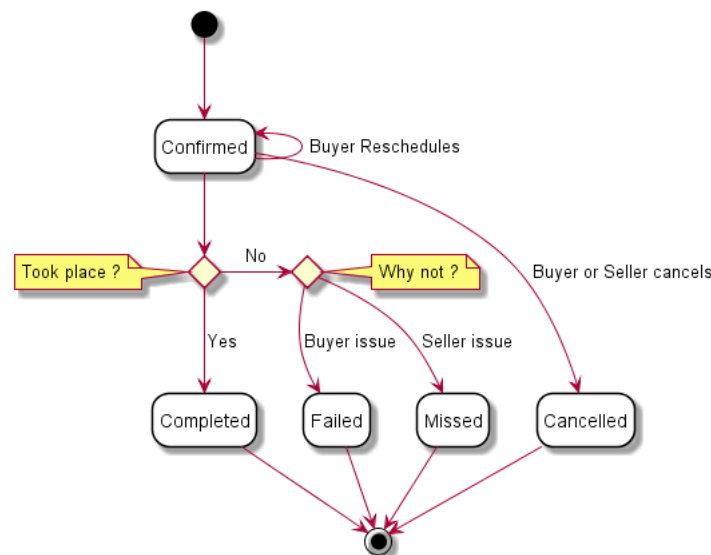


Figure 11. Appointment State Machine

After receiving the request, the Seller performs a validation of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `Appointment` in `confirmed` status. Then the Seller starts working on resolving the issue and moves the Appointment to `completed` state. The Appointment changes a status to `missed` when it didn't take place because of the Seller issue or to `failed` if caused the Buyer issue. Buyer or Seller can `cancel` the Appointment.

Table 5 presents the mapping between the API `status` names (aligned with TMF) and the MEF 113 naming, together with statuses' description.

status	MEF 113 name	Description
--------	-----------------	-------------

status	MEF 113 name	Description
confirmed	SCHEDULED	The Buyer has negotiated and confirmed the Appointment with the Seller (scheduled).
cancelled	CANCELLED	The Appointment was cancelled by the Buyer or the Seller determined that an Appointment was not required. This is a terminal state.
missed	MISSED	The Appointment did not take place, because of a Seller related issue. For example, no Seller Technician was available on the date of the appointment. .
failed	FAILED	The Appointment did not take place, because of an issue with the Buyer. For example, Seller Technician was unable to get to the Appointment due to an incorrect location or unable to get access to the Buyer's site. This is a terminal state.
completed	COMPLETED	The Appointment took place as confirmed. This is a terminal state.

Table 7: Appointment statuses

[R26] The Seller **MUST** support all Appointment `status` and associated transitions. [MEF113 R192]

[R27] The Seller **MUST** send an `AppointmentStatusChangeEvent` whenever an Appointment `status` change has occurred. [MEF113 R193]

[R28] The Seller **MUST** send an `AppointmentAttributeValueChangeEvent` whenever the Seller has updated any of the following attributes: [MEF113 R194]

- `attachment`
- `note`
- `relatedContactInformation`

6.3. Use Case 9: Retrieve Appointment List

[O3] The Buyer **MAY** use `GET /appointment` operation with the following filter criteria : [MEF113 O18]

- `workOrderId`
- `geographicSiteId`
- `geographicAddressId`
- `status`

- `validFor.gt`
- `validFor.lt`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. The filtering and pagination attributes must be specified in URI query format [RFC3986](#). Section [7.1.2](#). provides details about the implementation of pagination mechanism.

```
https://serverRoot/mefApi/sonata/appointment/v1/appointment?status=confirmed&limit=10&offset=0
```

The example above shows a Buyer's request to get all Appointments that are in the `confirmed` status. Additionally, the Buyer asks only for a first (`offset=0`) pack of 10 results (`limit=0`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `Appointment_Find` objects matching the criteria. To get more details (e.g. the item level information), the Buyer has to query a specific `Appointment` by `id`.

[R29] The Seller **MUST** return the following attributes: [MEF113 R103]

- `id`
- `href`
- `workOrder`
- `relatedPlace`
- `status`
- `validFor`

[R30] The Seller response **MUST** include all Appointments with an `validFor` that partially or completely overlaps with the `validFor.gt` and `validFor.lt` filter criteria. [MEF113 R104]

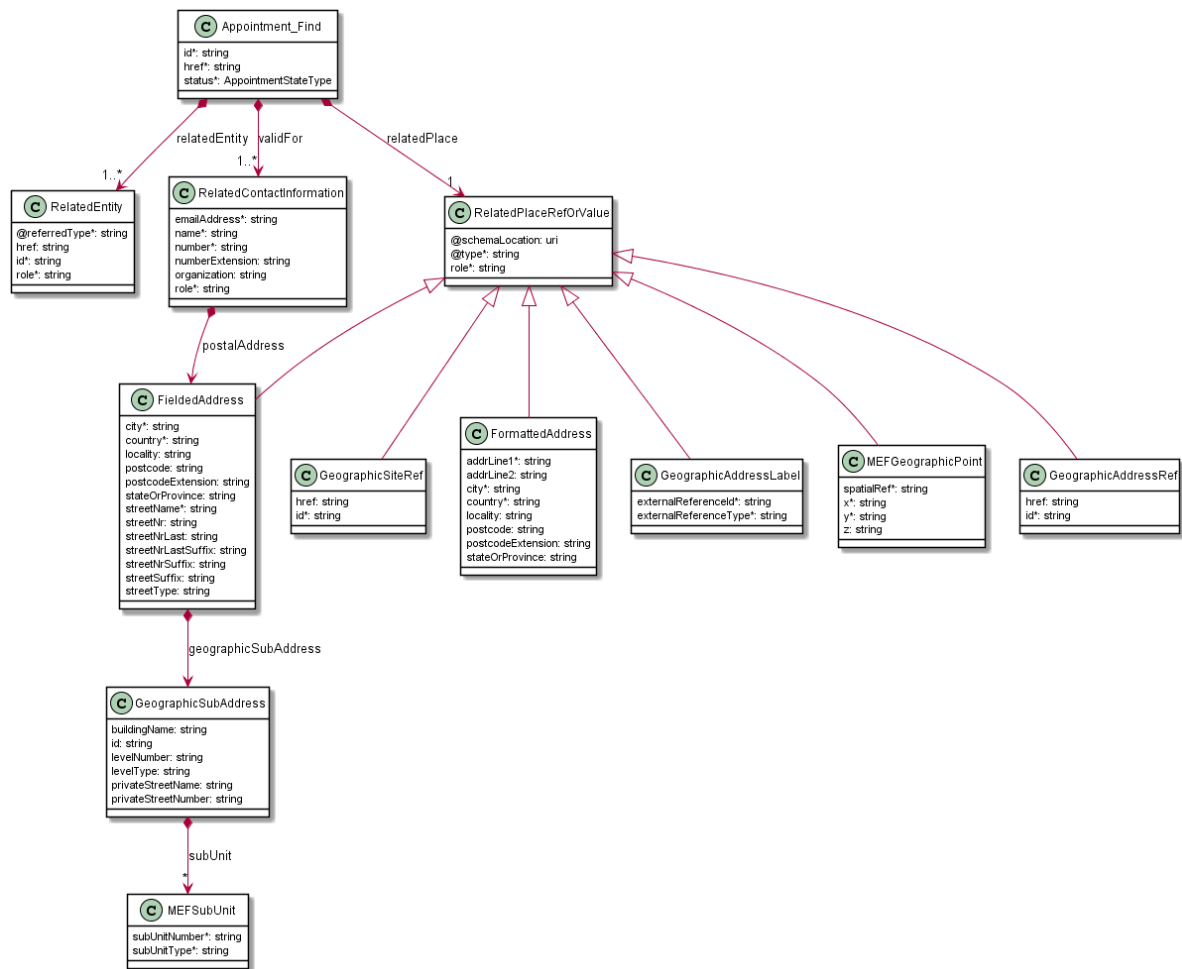


Figure 12. Use Case 9: Retrieve Appointment List - Model

6.4. Use Case 10: Retrieve Appointment by Appointment Identifier

The Buyer can get detailed information about the Appointment from the Seller by using a `GET /appointment/{id}` operation.

[R31] The Buyer **MUST** include the `id`. [MEF113 R105]

[R32] In case `id` does not allow to find a `Appointment` instance in Seller's system, an error `Error404` **MUST** be returned. [MEF113 R107]

[R33] The Seller's response **MUST** include the following attributes: [MEF113 R109]

- `id`
- `href`
- `attachment`
- `note`
- `relatedPlace`
- `relatedContactInformation`
- `status`

- **validfor**
- **workOrder**

[R34] The Seller's response **MUST** include all of the following optional attributes, if they were set by the Buyer or the Seller: [MEF113 R110]

- **attachment**
- **note**

6.5. Use Case 11: Patch Appointment by Appointment Identifier

The update operation is realized with the use of the REST PATCH operation. For that purpose, a specialized type **Appointment_Update** is provided. It consists of attributes limited to a subset that includes only the Buyer updatable attributes.

The PATCH usage recommendation follows TMF 621 json/merge (<https://tools.ietf.org/html/rfc7386>).

Figure 13 presents the model used in the PATCH request. The Seller responds with a **Appointment** type.

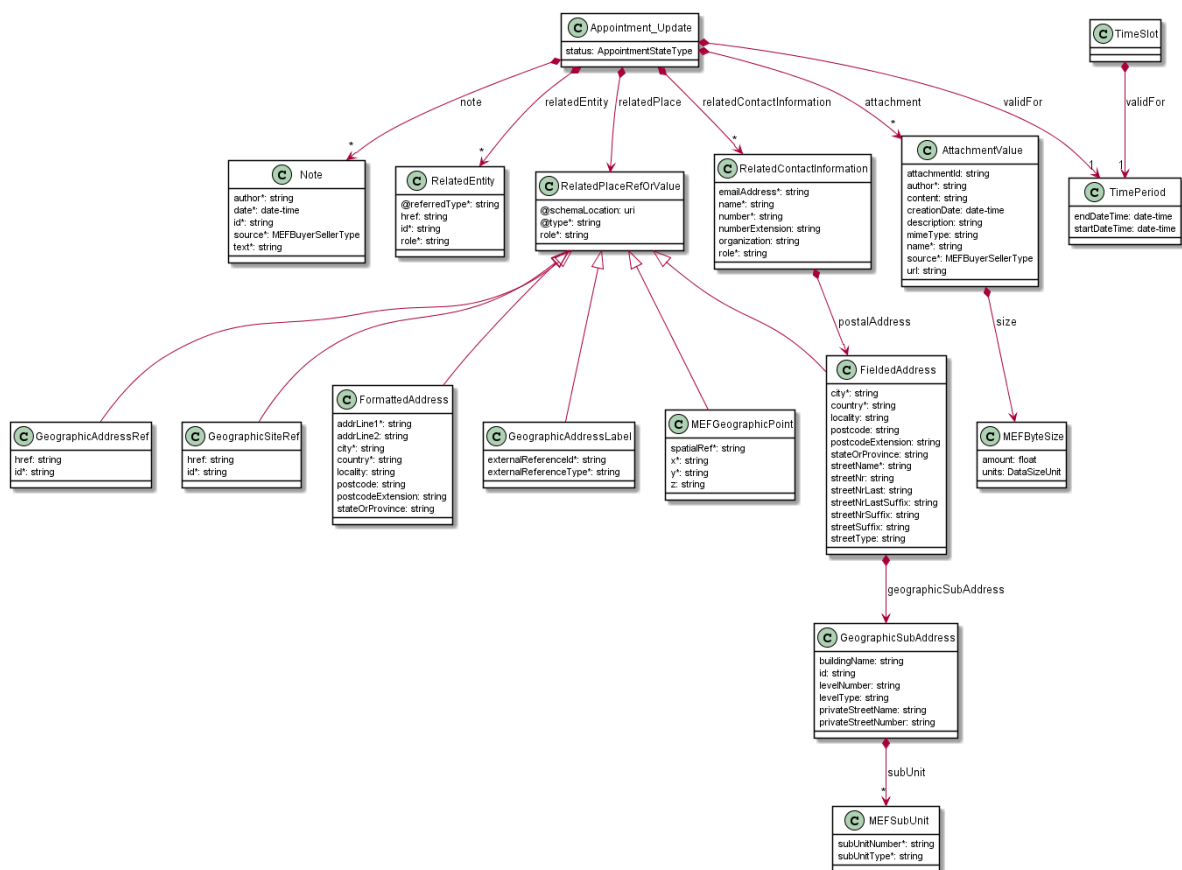


Figure 13. Patch request Model

[R35] The Buyer **MUST** include at least one of the following attributes: [MEF113 R112]

- attachment
- note
- relatedContactInformation - item with role=buyerAppointmentContact
- relatedContactInformation - item with role=appointmentPlaceContact
- validfor

[R36] The Buyer **MUST NOT** change the `workOrder` attribute. [MEF113 R113]

[R37] The Seller **MUST** return an error `Error404` if the Appointment with the `id` is not found. [MEF113 R115]

[R38] The Seller **MUST** return an error `Error422` if the `status` of the Appointment to be updated is not `confirmed`. [MEF113 R116]

[R39] If `validfor` was included by the Buyer, the Seller **MUST** return an error `Error422` if the `validfor` is not valid. [MEF113 R117]

[R40] If an `validfor` was included by the Buyer, the Seller **MUST** return an error `Error422` if a Seller technician is not available for an appointment at the `relatedPlace` for the specified `availableTimeSlot`. [MEF113 R118]

[R41] To add a new `note` the Buyer **MUST** append it to the existing `note` list. [MEF113 R119]

[R42] To add a new `attachment` the Buyer **MUST** append it to the existing `attachment` list. [MEF113 R120]

The example below shows a request to add a new `note` (existing cannot be modified or deleted):

```
{
  "note": [
    {<<previously existing>>
      "id": "note-1",
      "author": "John Example",
      "date": "2021-06-02T14:25:11.090Z",
      "source": "buyer",
      "text": "Couldn't reach the support on phone."
    },
    {<<added new note>>
      "id": "note-2",
      "author": "Kate Example",
      "date": "2021-06-02T19:25:11.090Z",
      "source": "buyer",
      "text": "Support reached after 5 hours"
    }
  ]
}
```

6.6. Use case 12: Cancel Appointment by Appointment Identifier

The Buyer may request to Cancel a Appointment by using `POST /appointment/{id}/cancel` endpoint. This operation only requires providing the `id` in the path and has an empty 204 confirmation response.

The sequence diagram below presents this use case in detail.

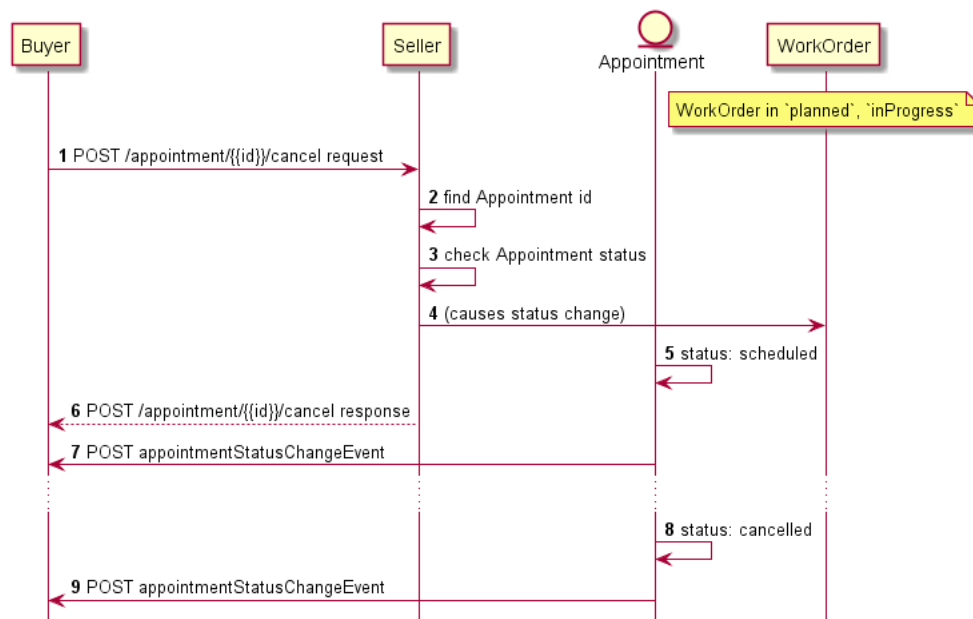


Figure 14. Cancel Appointment Flow

The Seller verifies the request, then searches for a Appointment to be cancelled by given `id`. If found, the status is verified `confirmed`. If everything is verified correctly, the Seller moves the appointment to the `assessingCancellation` status, sends a successful response to a cancellation request, and starts assessing the cancellation process for the appointment.

[R43] The Seller **MUST** return an error `Error404` if the Appointment `id` is not found. [MEF113 R125]

[R44] The Seller **MUST** return an error `Error422` if the Appointment `status` is not `confirmed`. [MEF113 R126]

[R45] The Seller **MUST** return an error `Error422` if the related WorkOrder `state` is not `planned`. [MEF113 R127]

[R46] The Seller **MUST** set the `appointmentRequired` attribute to `true` in the related WorkOrder. [MEF113 R128]

[R47] The Seller **MUST** set the related workorder `state` to `open`. [MEF113 R129]

[R48] The Seller **MUST** set the Appointment `status` to `canceled`. [MEF113 R1130]

6.7. Use Case 13: Retrieve WorkOrder List

[O4] The Buyer **MAY** use `GET /workOrder` operation with desired filtering criteria: [MEF113 O20]

- `appointmentRequired`
- `geographicalSiteId`
- `geographicalAddressId`
- `relatedEntity`
- `state`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. The filtering and pagination attributes must be specified in URI query format [RFC3986](#). Section [7.1.2](#). provides details about the implementation of pagination mechanism.

```
https://serverRoot/mefApi/sonata/appointment/v1/appointment?
status=inProgress&priority=critical&limit=10&offset=0
```

The example above shows a Buyer's request to get all WorkOrder that are in the `inProgress`. Additionally, the Buyer asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `WorkOrder_Find` objects matching the criteria. To get more details (e.g. the item level information), the Buyer has to query a specific `WorkOrder` by `id`.

[R49] The Seller **MUST** respond to a `WorkOrder_Find` containing the workOrder attributes that matches the Buyer's filter criteria: [MEF113 R133]

- `appointmentRequired`
- `id`
- `place`
- `relatedEntity`
- `state`

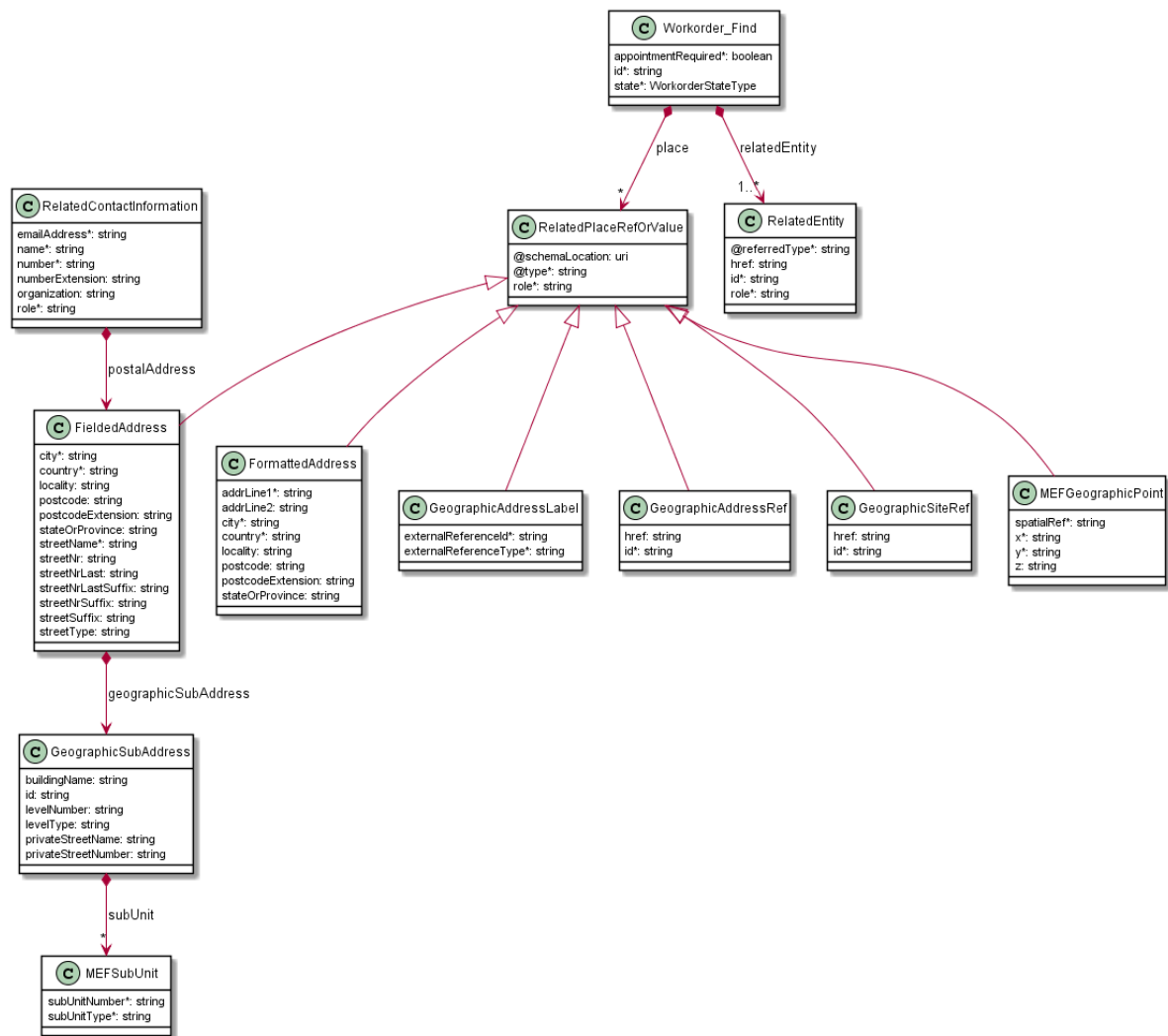


Figure 15. Use Case 13: Retrieve WorkOrder List - Model

6.8. Use Case 14: Retrieve WorkOrder by WorkOrder Identifier

The Buyer can retrieve detailed information about the WorkOrder from the Seller by using a `GET /workOrder/{id}` operation.

[R50] In case `id` does not allow to find a `WorkOrder` instance, an error `Error404` **MUST** be returned. [MEF113 R136]

[R51] The Seller's response **MUST** include the following WorkOrder attributes. [MEF113 R138]

- `appointmentRequired`
- `id`
- `place`
- `relatedContactInformation` - item with `role=technician`
- `relatedEntity`
- `state`

- task

[R52] The Seller **MUST** include all of the optional WorkOrder attributes, if they were set by the Buyer or Seller: [MEF113 R139]

- appointment
- note
- relatedContactInformation - item with role=technicalContact

The following Figure presents the model of the WorkOrder.

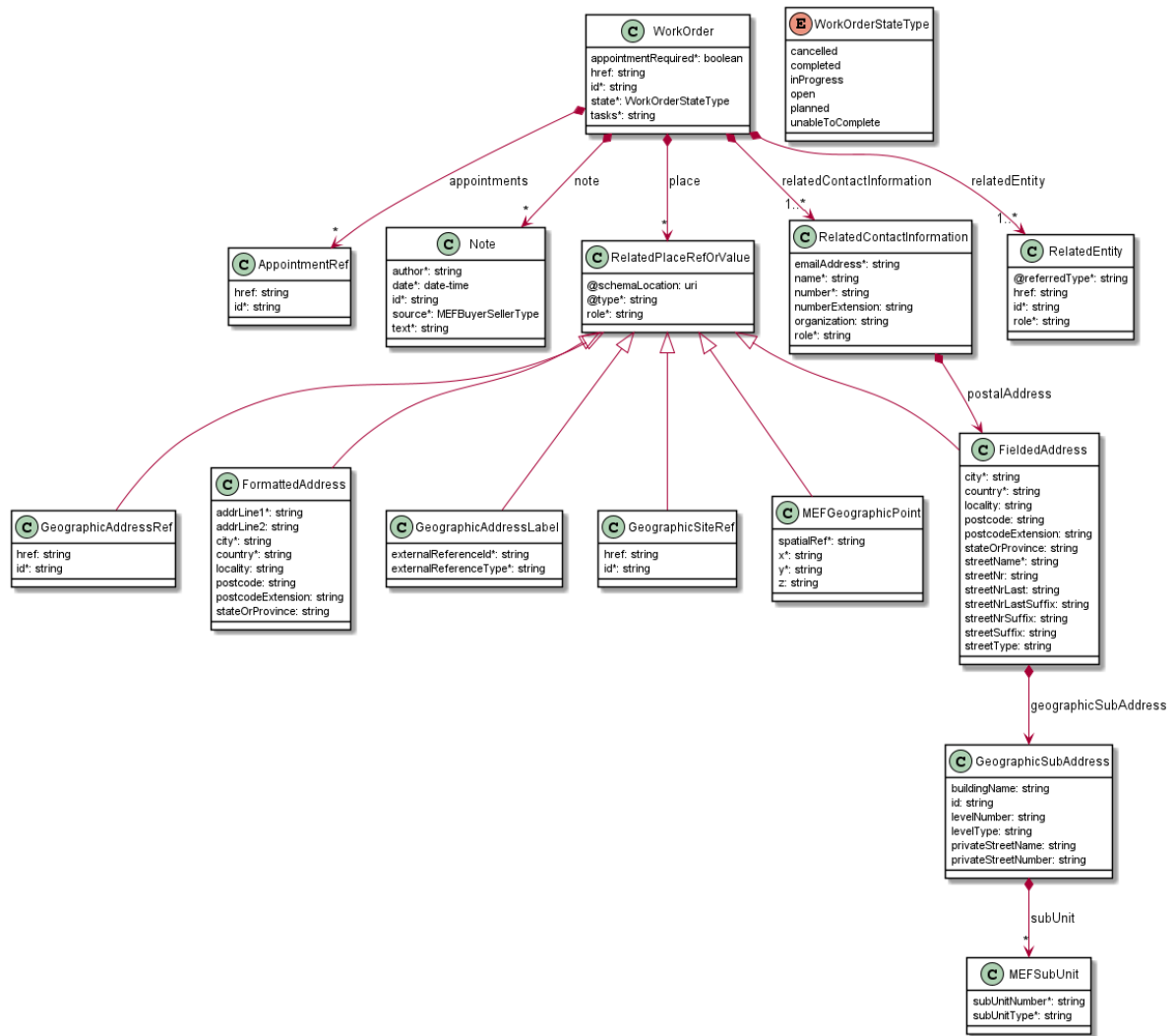


Figure 16. Use Case 14: WorkOrder - Model

Table 8 presents the mapping between the API **state** names and the MEF 113 naming, together with their description.

state	MEF 113 name	Description
completed	COMPLETED	Technician responsible for the WorkOrder has successfully completed all the assigned Tasks.

state	MEF 113 name	Description
cancelled	CANCELLED	The WorkOrder has been cancelled by the Seller or due to the Buyer requesting to cancel the Ticket.
inProgress	IN_PROGRESS	Technician responsible for the WorkOrder has been assigned and started one or more of the assigned Tasks.
open	OPEN	A workOrder was initiated by the Seller to be assigned to a Technician responsible for resolving the Ticket.
planned	PLANNED	The WorkOrder has been given an execution date for resolving one or more Tasks.
unableToComplete	UNABLE_TO_COMPLETE	The Technician responsible for the workOrder was unable to complete one or more of the assigned Tasks, because of additional skills or information is required. Additional tasks are required to resolve the ticket and a new workOrder needs to be opened.

Table 8: WorkOrder states

Figure 17 presents the WorkOrder state machine:

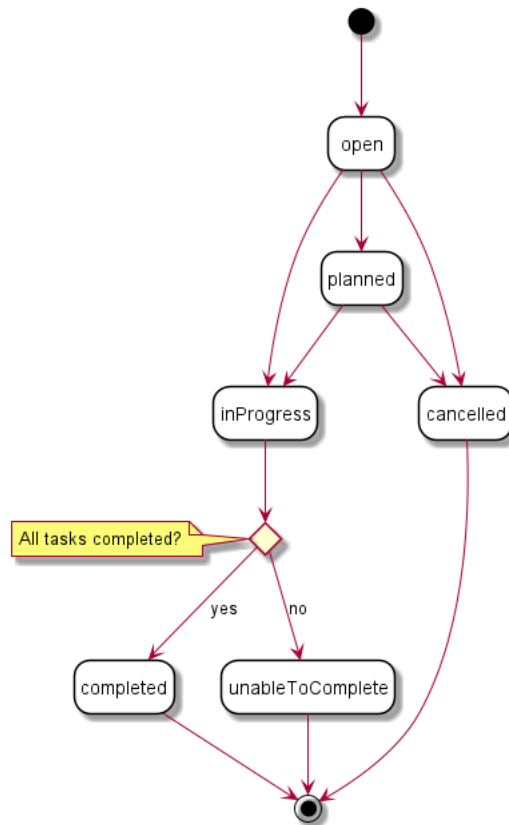


Figure 17. WorkOrder State Machine

[R53] The Seller **MUST** support all **state** values and their associated transitions. [MEF113 R181]

Below you can find a snippet with a WorkOrder example:

```

{
  "id": "98765432-9876-5432-0000-000000000055",
  "appointmentRequired": false,
  "relatedContactInformation": [
    {
      "emailAddress": "technical.contact@example.com",
      "name": "Technical Contact",
      "number": "+12-345-678-91",
      "organization": "Seller Example Co.",
      "role": "technicalContact"
    }
  ],
  "relatedEntity": [
    {
      "id": "01494064-6c79-4a32-83f7-53284196d35d",
      "role": "parentTroubleTicket",
      "@referredType": "TroubleTicket"
    }
  ],
  "state": "open",
  "task": ["Replace the broken SFP", "Perform OTDR"]
}

```

[R54] The Seller **MUST** send a **workOrderCreateEvent** whenever a new WorkOrder has been created. [MEF113 R182]

[R55] The Seller **MUST** send a **workOrderStateChangeEvent** whenever a WorkOrder **state** change has occurred. [MEF113 R183]

[R56] The Seller **MUST** send a `workOrderAppointmentRequiredEvent` when the Seller sets the `appointmentRequired` attribute in the `WorkOrder` to `true`. [MEF113 R184]

[R57] If the `appointmentRequired` attribute in a `WorkOrder` is `true`, the Buyer **MUST** schedule an Appointment using a Search Appointment request followed by a Create Appointment request before the Seller is able to continue processing the associated Ticket. [MEF113 R185]

6.9. Use case 19: Register for Appointment Notifications

To register for notifications the Buyer uses the `registerListener` operation from the API: `POST /hub`. The request model contains only 2 attributes:

- `callback` - mandatory, to provide the callback address the events will be notified to,
- `query` - optional, to provide the required types of event.

The usage of a combination of these attributes fulfills the [ME113 R125], [ME113 R126], [ME113 R127] requirements.

By using a simple request:

```
{
  "callback": "https://buyer.com/listenerEndpoint"
}
```

The Buyer subscribes for notification of all types of events. Those are:

- `appointmentAttributeValueChangeEvent`
- `appointmentStatusChangeEvent`

If the Buyer wishes to subscribe to 2 different types of events, there are 2 possible syntax variants [TMF630]:

```
eventType=appointmentAttributeValueChangeEvent,appointmentStatusChangeEvent
```

or

```
eventType=appointmentAttributeValueChangeEvent&appointmentStatusChangeEvent
```

The `query` formatting complies to RCF3986 [RFC3986](#). According to it, every attribute defined in the Event model (from notification API) can be used in the `query`. However, this standard requires only `eventType` attribute to be supported.

6.10. Use case 20: Send Appointment Notification

Notifications are used to asynchronously inform the Buyer about the respective objects and attributes changes. The Seller's synchronous response to a Appointment create requests are considered to act as a Create Notification so there is no explicit respective Create Notification type. The next notification must be sent when the state changes compared to the previously sent one.

[R58] The Seller **MUST NOT** send Appointment Notifications for an Appointment Notification Type to a Buyer who has not registered for the Appointment Notification Type. [MEF113 R164]

[R59] The Seller **MUST** send Appointment Notifications to a Buyer who has registered for the Appointment Notification Type. [MEF113 R165]

The Figure below shows all entities involved in the Notification use cases.

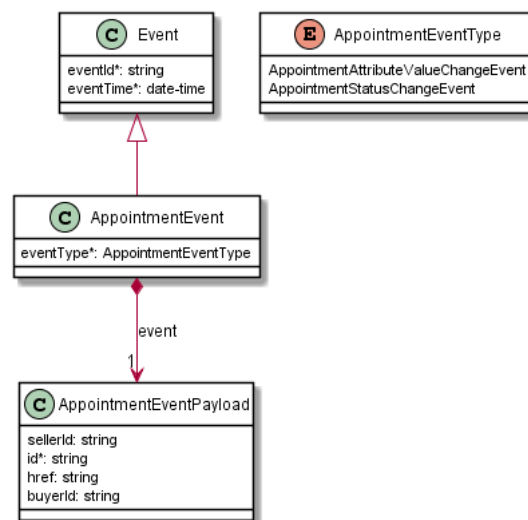


Figure 18. Use Case 20. Appointment Notification Data Model

Note: the body of the event carries only the source object's **id**. The Buyer needs to query it later by **id** to get details.

To stop receiving events, the Buyer has to use the **unregisterListener** operation from the **DELETE /hub/{id}** endpoint. The **id** is the identifier received from the Seller during the listener registration.

The table below presents the mapping between the API Notification types' names and the ones in MEF 113. The inconsistencies are caused by using the TMF event types as the base for this API.

API name	MEF 113 name
appointmentAttributeValueChangedEvent	APPOINTMENT_UPDATE
appointmentStatusChangeEvent	APPOINTMENT_STATUS_CHANGE

Table 9. Appointment Notification types mapping

6.11. Use case 21: Register for WorkOrder Notifications

To register for notifications the Buyer uses the `registerListener` operation from the API: `POST /hub`. The request model contains only 2 attributes:

- `callback` - mandatory, to provide the callback address the events will be notified to,
- `query` - optional, to provide the required types of event.

The usage of a combination of these attributes fulfills the [ME113 R125], [ME113 R126], [ME113 R127] requirements.

By using a simple request:

```
{
  "callback": "https://buyer.com/listenerEndpoint"
}
```

The Buyer subscribes for notification of all types of events. Those are:

- `workOrderCreateEvent`
- `workOrderStateChangeEvent`
- `workOrderAppointmentRequiredEvent`

If the Buyer wishes to subscribe to 3 different types of events, there are 3 possible syntax variants [TMF630]:

```
eventType=workOrderCreateEvent,workOrderStateChangeEvent,workOrderAppointmentRequiredEvent
```

or

```
eventType=workOrderCreateEvent&workOrderStateChangeEvent&workOrderAppointmentRequiredEvent
```

The `query` formatting complies to RCF3986 [RFC3986](#). According to it, every attribute defined in the Event model (from notification API) can be used in the `query`. However, this standard requires only `eventType` attribute to be supported.

6.12. Use case 22: Send WorkOrder Notification

Notifications are used to asynchronously inform the Buyer about the respective objects and attributes changes. The Seller's synchronous response to a WorkOrder create requests are considered to act as a Create Notification so there is no explicit respective Create Notification type. The next notification must be sent when the state changes compared to the previously sent one.

[R60] The Seller **MUST NOT** send WorkOrder Notifications for a WorkOrder Notification Type to a Buyer who has not registered for the WorkOrder Notification Type. [MEF R170]

[R61] The Seller **MUST** send WorkOrder Notifications to a Buyer who has registered for the WorkOrder Notification Type. [MEF R171]

- `workOrderCreateEvent`
- `workOrderStateChangeEvent`
- `workOrderAppointmentRequiredEvent`

The Figure below shows all entities involved in the Notification use cases.

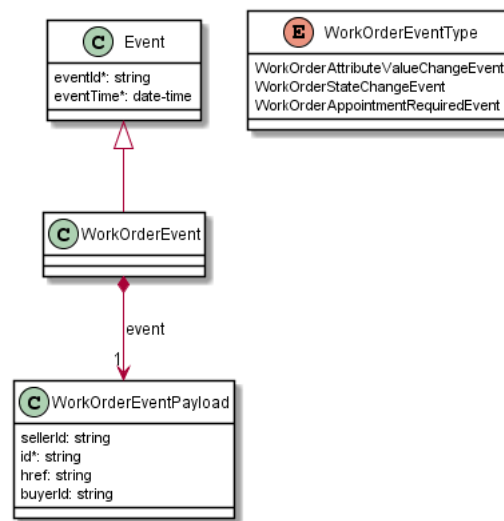


Figure 19. Use Case 22. WorkOrder Notification Data Model

Note: the body of the event carries only the source object's `id`. The Buyer needs to query it later by `id` to get details.

To stop receiving events, the Buyer has to use the `unregisterListener` operation from the `DELETE /hub/{id}` endpoint. The `id` is the identifier received from the Seller during the listener registration.

The table below presents the mapping between the API Notification types' names and the ones in MEF 113. The inconsistencies are caused by using the TMF event types as the base for this API.

API name	MEF 113 name
<code>workOrderCreateEvent</code>	WORKORDER_CREATE
<code>workOrderStateChangeEvent</code>	WORKORDER_STATE_CHANGE
<code>workOrderAppointmentRequiredEvent</code>	WORKORDER_APPOINTMENT_REQUIRED

Table 10. WorkOrder Notification types mapping

7. API Details

7.1. API patterns

7.1.1. Indicating errors

Erroneous situations are indicated by appropriate HTTP responses. An error response is indicated by HTTP status 4xx (for client errors) or 5xx (for server errors) and appropriate response payload. The Product Order API uses the error responses as depicted and described below.

Implementations can use HTTP error codes not specified in this standard in compliance with rules defined in RFC 7231 [RFC7231]. In such a case, the error message body structure might be aligned with the `Error`.

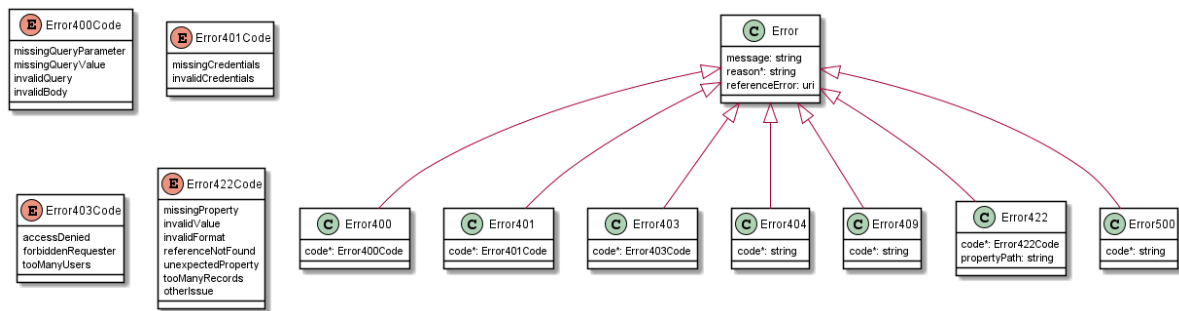


Figure 20. Data model types to represent an erroneous response

7.1.1.1. Type Error

Description: Standard Class used to describe API response error Not intended to be used directly. The `code` in the HTTP header is used as a discriminator for the type of error returned in runtime.

Name	Type	Description
message	string	Text that provides mode details and corrective actions related to the error. This can be shown to a client user.
reason*	string	Text that explains the reason for the error. This can be shown to a client user.
referenceError	uri	URL pointing to documentation describing the error

7.1.1.2. Type Error400

Description: Bad Request. (<https://tools.ietf.org/html/rfc7231#section-6.5.1>)

Inherits from:

- [Error](#)

Name	Type	Description
		One of the following error codes:
		- missingQueryParameter: The URI is missing a required query-string parameter
code*	Error400Code	- missingQueryValue: The URI is missing a required query-string parameter value
		- invalidQuery: The query section of the URI is invalid.
		- invalidBody: The request has an invalid body

7.1.1.3. [enum](#) Error400Code

Description: One of the following error codes:

- missingQueryParameter: The URI is missing a required query-string parameter
- missingQueryValue: The URI is missing a required query-string parameter value
- invalidQuery: The query section of the URI is invalid.
- invalidBody: The request has an invalid body

7.1.1.4. Type Error401

Description: Unauthorized. (<https://tools.ietf.org/html/rfc7235#section-3.1>)

Inherits from:

- [Error](#)

Name	Type	Description
		One of the following error codes:
code*	Error401Code	- missingCredentials: No credentials provided.
		- invalidCredentials: Provided credentials are invalid or expired

7.1.1.5. [enum](#) Error401Code

Description: One of the following error codes:

- missingCredentials: No credentials provided.
- invalidCredentials: Provided credentials are invalid or expired

7.1.1.6. Type Error403

Description: Forbidden. This code indicates that the server understood the request but refuses to authorize it. (<https://tools.ietf.org/html/rfc7231#section-6.5.3>)

Inherits from:

- [Error](#)

Name	Type	Description
code*	Error403Code	<p>This code indicates that the server understood the request but refuses to authorize it because of one of the following error codes:</p> <ul style="list-style-type: none">- accessDenied: Access denied- forbiddenRequester: Forbidden requester- tooManyUsers: Too many users

7.1.1.7. [enum](#) **Error403Code**

Description: This code indicates that the server understood the request but refuses to authorize it because of one of the following error codes:

- accessDenied: Access denied
- forbiddenRequester: Forbidden requester
- tooManyUsers: Too many users

7.1.1.8. **Type Error404**

Description: Resource for the requested path not found.
(<https://tools.ietf.org/html/rfc7231#section-6.5.4>)

Inherits from:

- [Error](#)

Name	Type	Description
code*	string	<p>The following error code:</p> <ul style="list-style-type: none">- notFound: A current representation for the target resource not found

7.1.1.9. **Type Error409**

Description: Conflict (<https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.8>)

Inherits from:

- [Error](#)

Name	Type	Description
------	------	-------------

Name	Type	Description
------	------	-------------

code*	string	The following error code: - conflict: The client has provided a value whose semantics are not appropriate for the property.
-------	--------	---

7.1.1.10. Type Error422

The response for HTTP status 422 is a list of elements that are structured using the Error422 data type. Each list item describes a business validation problem. This type introduces the propertyPath attribute which points to the erroneous property of the request, so that the Buyer may fix it easier. It is highly recommended that this property should be used, yet remains optional because it might be hard to implement.

Description: Unprocessable entity due to a business validation problem.
(<https://tools.ietf.org/html/rfc4918#section-11.2>)

Inherits from:

- Error

Name	Type	Description
		One of the following error codes:
		- missingProperty: The property the Seller has expected is not present in the payload
		- invalidValue: The property has an incorrect value
		- invalidFormat: The property value does not comply with the expected value format
code*	Error422Code	- referenceNotFound: The object referenced by the property cannot be identified in the Seller system
		- unexpectedProperty: Additional property, not expected by the Seller has been provided
		- tooManyRecords: the number of records to be provided in the response exceeds the Seller's threshold.
		- otherIssue: Other problem was identified (detailed information provided in a reason)
propertyPath	string	A pointer to a particular property of the payload that caused the validation issue. It is highly recommended that this property should be used. Defined using JavaScript Object Notation (JSON) Pointer (https://tools.ietf.org/html/rfc6901).

7.1.1.11. enum Error422Code

Description: One of the following error codes:

- **missingProperty:** The property the Seller has expected is not present in the payload
- **invalidValue:** The property has an incorrect value
- **invalidFormat:** The property value does not comply with the expected value format
- **referenceNotFound:** The object referenced by the property cannot be identified in the Seller system
- **unexpectedProperty:** Additional property, not expected by the Seller has been provided
- **tooManyRecords:** the number of records to be provided in the response exceeds the Seller's threshold.
- **otherIssue:** Other problem was identified (detailed information provided in a reason)

7.1.1.12. Type Error500

Description: Internal Server Error. (<https://tools.ietf.org/html/rfc7231#section-6.6.1>)

Inherits from:

- [Error](#)

Name	Type	Description
		The following error code:
code*	string	- internalError: Internal server error - the server encountered an unexpected condition that prevented it from fulfilling the request.

The following error code:

code* string - internalError: Internal server error - the server encountered an unexpected condition that prevented it from fulfilling the request.

7.1.1.13. Type Error501

Description: Not Implemented. Used in case Seller is not supporting an optional operation (<https://tools.ietf.org/html/rfc7231#section-6.6.2>)

Inherits from:

- [Error](#)

Name	Type	Description
		The following error code:
code*	string	- notImplemented: Method not supported by the server

The following error code:

code* string - notImplemented: Method not supported by the server

7.1.2. Response pagination

A response to retrieve a list of results (e.g. [GET /productOfferingQualification](#)) can be paginated. The Buyer can specify following query attributes related to pagination:

- [limit](#) - number of expected list items
- [offset](#) - offset of the first element in the result list

The Seller returns a list of elements that comply with the requested `limit`. If the requested `limit` is higher than the supported list size the smaller list result is returned. In that case, the size of the result is returned in the header attribute `X-Result-Count`. The Seller can indicate that there are additional results available using:

- `X-Total-Count` header attribute with the total number of available results
- `X-Pagination-Throttled` header set to `true`

[R62] Seller **MUST** use either `x-Total-Count` or `x-Pagination-Throttled` to indicate that the page was truncated and additional results are available.

7.2. Management API Data model

7.2.1. Appointment

Figure 18 presents the whole Appointment Management data model the data types, requirements related to them and mapping to MEF 113 specifications are discussed later in this section.

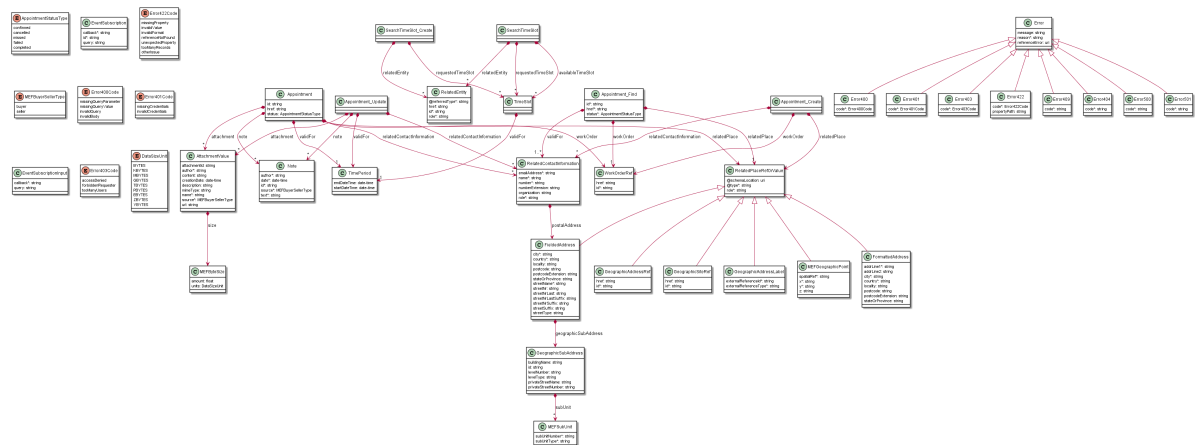


Figure 21. Appointment Management Data Model

7.2.1.1. Type Appointment

Description: An Appointment is an arrangement to do something or meet someone at a particular time, at a place (for face to face appointment) or in a contact medium (for phone appointment).

Name	Type	Description	MEF
id	string	Unique identifier of the appointment	Appointment Identifier
href	string	Unique URI used to access to the appointment resource	Not represented in MEF

Name	Type	Description	MEF
attachment	AttachmentValue[]	Attachments to the Appointment, such as a file, screen shot or embedded content	Appointment Attach
note	Note[]	Notes describing the purpose of and the results of the Appointment	Appointment Notes
relatedPlace	RelatedPlaceRefOrValue	The location of the Appointment.	Appointment Place
status	AppointmentStatusType	The state of the Appointment	Appointment State
validFor*	TimePeriod	The Date and Time interval the Seller Technician is scheduled to arrive at the Appointment.	Appointment Timesl
relatedContactInformation	RelatedContactInformation[]	<p>Party playing a role for this Appointment. The 'role' is to specify the type of contact as specified in MEF 113:</p> <p>Appointment Place Contact (role=appointmentPlaceContact) - The site contact(s) which the Seller Technician may need to contact in order to get access to the Appointment Place during the Appointment. This could be an end-user, security personnel or any authorized person Buyer Appointment Contact (role=buyerAppointmentContact)- The Buyer contact(s) assigned to and responsible for the Appointment. Seller Appointment Contact (role=sellerAppointmentContact) - The Seller contact(s) assigned to and responsible for the Appointment.</p>	Appointment Place Contact Buyer Appointment Contact Seller Appointment Contact
workOrder	WorkOrderRef	A reference to a WorkOrder for which the Appointment is created	WorkC

7.2.1.2. Type Appointment_Create

Description: An Appointment is an arrangement to do something or meet someone at a particular time, at a place (for face to face appointment) or in a contact medium (for phone appointment).

Name	Type	Description	MEF 113
relatedPlace	RelatedPlaceRefOrValue	Identifies the postal address of the person or office to be contacted.	Appointment Place
		Party playing a role for this Appointment. The 'role' is to specify the type of contact as specified in MEF 113: Appointment Place Contact (role=appointmentPlaceContact) - The site contact(s) which the Seller Technician may need to contact in order to get access to the Appointment Place during the Appointment. This could be an end-user, security personnel or any authorized person Buyer Appointment Contact (role=buyerAppointmentContact)- The Buyer contact(s) assigned to and responsible for the Appointment. Seller Appointment Contact (role=sellerAppointmentContact) - The Seller contact(s) assigned to and responsible for the Appointment.	Appointment Place Contact Buyer Appointment Contact Seller Appointment Contact
relatedContactInformation	RelatedContactInformation[]		
workOrder	WorkOrderRef	A reference to a WorkOrder for which the Appointment is created	WorkOrder

7.2.1.3. Type Appointment_Find

Description: An Appointment is an arrangement to do something or meet someone at a particular time, at a place (for face to face appointment) or in a contact medium (for phone appointment).

Name	Type	Description	MEF 113
------	------	-------------	---------

Name	Type	Description	MEF 113
id*	string	Unique identifier of the appointment	Appointment Identifier
href*	string	Unique URI used to access to the appointment resource	Not represented in MEF 113
relatedPlace*	RelatedPlaceRefOrValue	The location of the Appointment.	Appointment Place
status*	AppointmentStatusType	The state of the Appointment.	Appointment State
validFor*	TimePeriod[]	The Date and Time interval the Seller Technician is scheduled to arrive at the Appointment	Appointment Timeslot
workOrder*	WorkOrderRef	A reference to a WorkOrder for which the Appointment is created.	WorkOrder

7.2.1.4. Type Appointment_Update

Description: An Appointment is an arrangement to do something or meet someone at a particular time, at a place (for face to face appointment) or in a contact medium (for phone appointment).

Name	Type	Description	MEF 113
attachment	AttachmentValue[]	Attachments to the Appointment, such as a file, screen shot or embedded content.	Appointment Attachments
note	Note[]	Notes describing the purpose of and the results of the Appointment.	Appointment Notes
validFor	TimePeriod	The Date and Time interval the Seller Technician is scheduled to arrive at the Appointment	Appointment Timeslot

Name	Type	Description	MEF 113
relatedContactInformation	RelatedContactInformation[]	<p>Party playing a role for this Appointment. The 'role' is to specify the type of contact as specified in MEF 113:</p> <p>Appointment Place Contact (role=appointmentPlaceContact) - The site contact(s) which the Seller Technician may need to contact in order to get access to the Appointment Place during the Appointment. This could be an end-user, security personnel or any authorized person Buyer Appointment Contact (role=buyerAppointmentContact)- The Buyer contact(s) assigned to and responsible for the Appointment. Seller Appointment Contact (role=sellerAppointmentContact) - The Seller contact(s) assigned to and responsible for the Appointment.</p>	<p>Appointment Place Contact Buyer Appointment Contact Seller Appointment Contact</p>

7.2.1.5. enum AppointmentStatusType

Description: Valid values for the lifecycle status of the appointment

Value	MEF 113	Description
confirmed	SCHEDULED	The Buyer has negotiated and scheduled the Appointment with the Seller.
cancelled	CANCELLED	The Appointment was cancelled by the Buyer or the Seller determined that an Appointment was not required. This is a terminal state.
missed	MISSED	The Appointment did not take place, because of a Seller related issue. For example, no Seller Technician was available on the date of the appointment.

Value	MEF 113	Description
failed	FAILED	The Appointment did not take place, because of an issue with the Buyer. For example, Seller Technician was unable to get to the Appointment due to an incorrect location or unable to get access to the Buyer's site. This is a terminal state.
completed	COMPLETED	The Appointment took place as scheduled. This is a terminal state.

7.2.1.6. Type SearchTimeSlot

Description: This task resource is used to retrieve available time slots. One of this available time slot is after used to create or reschedule an appointment

Name	Type	Description	MEF 113
availableTimeSlot	TimeSlot[]	A set of time slots with availability of a Seller's Technician returned by the Seller, which the Buyer may select for creating or rescheduling an Appointment	Available Timeslots
requestedTimeSlot	TimeSlot[]	A set of preferred time slots the Buyer is re-requesting the Seller to verify for availability by a Seller's Technician at the Place referenced in the Appointment Related Entity. For example Monday thru Friday, or a set of specific time slots.	Requested Timeslots
relatedEntity	RelatedEntity[]	A Reference to the Related Entity the Appointment is for.	Appointment Related Entity

7.2.1.7. Type SearchTimeSlot_Create

Description: This task resource is used to retrieve available time slots. One of this available time slot is after used to create or reschedule an appointment

Name	Type	Description	MEF 113
------	------	-------------	---------

Name	Type	Description	MEF 113
requestedTimeSlot	TimeSlot[]	A set of preferred time slots the Buyer is re-requesting the Seller to verify for availability by a Seller's Technician at the Place referenced in the Appointment Related Entity. For example Monday thru Friday, or a set of specific time slots.	Requested Timeslots
relatedEntity	RelatedEntity[]	A Reference to the Related Entity the Appointment is for.	Appointment Related Entity

7.2.1.8. Type WorkOrderRef

Description: A reference to an WorkOrder resource.

Name	Type	Description	MEF 113
href	string	Hyperlink to the referenced WorkOrder.	Not represented in MEF 113
id*	string	Identifier of the referenced WorkOrder.	WorkOrder Identifier

7.2.2. WorkOrder

Figure 22 presents the whole WorkOrder Management data model the data types, requirements related to them and mapping to MEF 113 specifications are discussed later in this section.

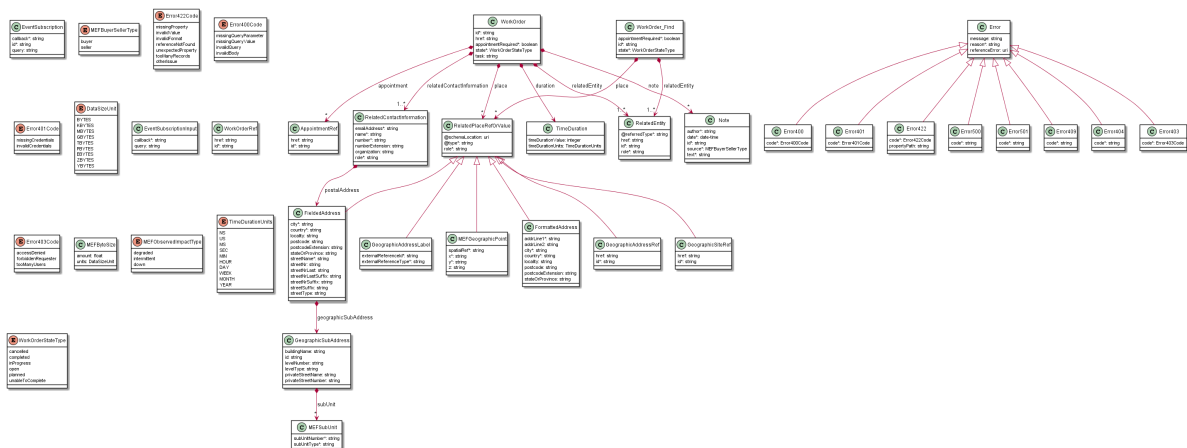


Figure 22. Workorder Management Data Model

7.2.2.1. Type WorkOrder

Description: A set of tasks to be scheduled and performed under the responsibility of a Seller Technician(s)

Name	Type	Description	MEF 113
id*	string	Unique (within the Seller domain) identifier for the WorkOrder.	Workorder Identifier
href	string	Hyperlink, a reference to the WorkOrder entity	Not represented in MEF 113
appointmentRequired*	boolean	The Seller requires the Buyer to schedule an Appointment. If set to true, the Seller is Requesting the Buyer to schedule an Appointment.	Appointment Required
appointment	AppointmentRef[]	A reference to a set of Appointments for the WorkOrder. A WorkOrder may contain only one open Appointment at a time (e.g. with state of 'scheduled').	Workorder Appointments
duration	TimeDuration	The anticipated amount of time the Seller Technician will be onsite.	Workorder Duration
note	Note[]	A set of unstructured comments or information associated to the WorkOrder	Workorder Notes

Name	Type	Description	MEF 113
place	RelatedPlaceRefOrValue[]	The location where the WorkOrder Tasks are to be performed. If an appointment is needed, this will also be the location where the Appointment takes place and includes the site contact which the Seller technician may need to get access to the Buyer's site during the Appointment. This could be an end-user, security personnel or any authorized person	Workorder Place
relatedContactInformation*	RelatedContactInformation[]	Party playing a role for this WorkOrder. The 'role' is to specify the type of contact as specified in MEF 113: Technical Contact ('role=technicalContact') - REQUIRED to be set by the Seller. The Seller technical contact responsible for the WorkOrder. Technician ('role=technician') - The Seller technician assigned to the WorkOrder and responsible for performing a set of tasks. In certain instances this could be a Buyer technician that works on behalf of the Seller.	Workorder Contact

Name	Type	Description	MEF 113
relatedEntity*	RelatedEntity[]	An entity that is related to the WorkOrder such as a service, a product, etc. The entity which the WorkOrder is associated with.	Workorder Related Entity
state*	WorkOrderStateType	The state of the Workorder.	Workorder State
task	string[]	A set of tasks to be performed under the responsibility of the Technician to fulfil the WorkOrder. Each item is a description of a specific task to be performed under the responsibility of the Technician.	Tasks

7.2.2.2. Type WorkOrder_Find

Description: A set of tasks to be scheduled and performed under the responsibility of a Seller Technician(s)

Name	Type	Description	MEF 113
appointmentRequired*	boolean	The Seller requires the Buyer to schedule an Appointment. If set to true, the Seller is Requesting the Buyer to schedule an Appointment.	Appointment Required
id*	string	Unique (within the Seller domain) identifier for the WorkOrder.	Workorder Identifier

Name	Type	Description	MEF 113
place	RelatedPlaceRefOrValue[]	The location where the WorkOrder Tasks are to be performed. If an appointment is needed, this will also be the location where the Appointment takes place and includes the site contact which the Seller technician may need to get access to the Buyer's site during the Appointment. This could be an end-user, security personnel or any authorized person	Workorder Place
relatedEntity*	RelatedEntity[]	An entity that is related to the WorkOrder such as a service, a product, etc. The entity which the WorkOrder is associated with.	Workorder Related Entity
state*	WorkOrderStateType	The state of the Workorder	Workorder State

7.2.2.3. enum WorkOrderStateType

Description: Possible values for the state of the WorkOrder

state	MEF 113 name	Description
completed	COMPLETED	The Seller Technician responsible for the WorkOrder has successfully completed all the assigned Tasks.
cancelled	CANCELLED	The WorkOrder has been cancelled by the Seller or due to the Buyer requesting to cancel the WorkOrder.

state	MEF 113 name	Description
inProgress	IN_PROGRESS	The Seller Technician responsible for the WorkOrder has been assigned and started one or more of the assigned Tasks.
open	OPEN	A WorkOrder was initiated by the Seller to be assigned to a Technician responsible for resolving the WorkOrder.
planned	PLANNED	The WorkOrder has been given an execution date for resolving one or more Tasks.
unableToComplete	UNABLE_TO_COMPLETE	The Seller Technician responsible for the WorkOrder was unable to complete one or more of the assigned Tasks, because additional skills or information is required. Additional tasks are required to resolve the WorkOrder and a new WorkOrder needs to be opened.

7.2.2.4. Type AppointmentRef

Description: A reference to an Appointment resource available through Appointment API.

Name	Type	Description	MEF 113
href	string	Hyperlink to the referenced Appointment. Hyperlink MAY be used by the Seller in responses. Hyperlink MUST be ignored by the Seller in case it is provided by the Buyer in a request	Not represented in MEF 113
id*	string	Identifier of the referenced Appointment.	Appointment Identifier

7.2.3. Common

Types described in this subsection are shared among two or more Cantata and Sonata APIs.

7.2.3.1. Type AttachmentValue

Description: Complements the description of an element (for instance a product) through video, pictures...

Name	Type	Description	MEF 113
attachmentId	string	locally unique identifier to distinguish items from the Attachment list.	Not represented in MEF 113
author*	string	Author of the Attachment	Attachment Author
content	string	The actual contents of the attachment object, if embedded, encoded as base64. Either url or (content and mimeType) attributes MUST be provided during creation.	Content
creationDate	date-time	The date the Attachment was added.	Attachment Date
description	string	A narrative text describing the content of the attachment	Description
mimeType	string	Attachment mime type such as extension file for video, picture and document	Mime Type
name*	string	The name of the attachment	Attachment Name
size	MEFByteSize	The size of the attachment.	MEFByteSize
source*	MEFBuyerSellerType	Indicates if the attachment was added by the Buyer or the Seller.	Attachment Source
url	string	URL where the attachment is located. Either url or (content and mimeType) attributes MUST be provided during creation.	URL

7.2.3.2. enum DataSizeUnit

Description: The unit of measure in the data size.

Value

BYTES

KBYTES

GBYTES

TBYTES

Value
PBYTES
EBYTES
ZBYTES
YBYTES

7.2.3.3. Type FieldedAddress

Description: A type of Address that has a discrete field and value for each type of boundary or identifier down to the lowest level of detail. For example "street number" is one field, "street name" is another field, etc. Reference: MEF 79 (Sn 8.9.2)

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
city*	string	The city that the address is in	City
country*	string	Country that the address is in	Country
geographicSubAddress	GeographicSubAddress	Additional fields used to specify an address, as detailed as possible.	Not represented in MEF 113
locality	string	The locality that the address is in	Locality
postcode	string	Descriptor for a postal delivery area, used to speed and simplify the delivery of mail (also known as zip code)	Postal Code
postcodeExtension	string	An extension of a postal code. E.g. the part following the dash in a US urban property address	Postal Code Extension
stateOrProvince	string	The State or Province that the address is in	State Or Province

Name	Type	Description	MEF 113
streetName*	string	Name of the street or other street type	Street Name
streetNr	string	Number identifying a specific property on a public street. It may be combined with streetNrLast for ranged addresses. MEF 79 defines it as required however as in certain countries it is not used we make it optional in API.	Street Number
streetNrLast	string	Last number in a range of street numbers allocated to a property	Street Number Last
streetNrLastSuffix	string	Last street number suffix for a ranged address	Street Number Suffix Last
streetNrSuffix	string	The first street number suffix	Street Number Suffix
streetSuffix	string	A modifier denoting a relative direction	Street Suffix
streetType	string	The type of street (e.g., alley, avenue, boulevard, brae, crescent, drive, highway, lane, terrace, parade, place, tarn, way, wharf)	Street Type

7.2.3.4. Type FormattedAddress

Description: A type of Address that has discrete fields for each type of boundary or identifier with the exception of street and more specific location details, which are combined into a maximum of two strings based on local postal addressing conventions.

Reference: MEF 79 (Sn 8.9.3)

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
addrLine1*	string	The first address line in a formatted address	Address Line 1
addrLine2	string	The second address line in a formatted address	Address Line 2
city*	string	The city that the address is in	City
country*	string	Country that the address is in	Country
locality	string	An area of defined or undefined boundaries within a local authority or other legislatively defined area, usually rural or semi-rural in nature	Locality
postcode	string	Descriptor for a postal delivery area, used to speed and simplify the delivery of mail (also known as ZIP code)	Postal Code
postcodeExtension	string	An extension of a postal code. E.g. the part following the dash in an US urban property address	Postal Code Extension
stateOrProvince	string	The State or Province that the address is in	State Or Province

7.2.3.5. Type GeographicAddressLabel

Description: A unique identifier controlled by a generally accepted independent administrative authority that specifies a fixed geographical location. Reference: MEF 79 (Sn 8.9.4)

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
externalReferenceId*	string	A reference to an address by id	Administrative Authority Address Label

Name	Type	Description	MEF 113
externalReferenceType*	string	Uniquely identifies the authority that specifies the addresses reference and/or its type (if the authority specifies more than one type of address). The value(s) to be used are to be agreed during the onboarding. For North American providers this would normally be CLLI (Common Language Location Identifier) code.	Administrative Authority

7.2.3.6. Type GeographicAddressRef

Description: A reference to a Geographic Address resource available through Address Validation API.

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
href	string	Hyperlink to the referenced GeographicAddress. Hyperlink MAY be used by the Seller in responses. Hyperlink MUST be ignored by the Seller in case it is provided by the Buyer in a request	Not represented in MEF 113
id*	string	Identifier of the referenced Geographic Address. This identifier is assigned during a successful address validation request (Geographic Address Validation API)	Not represented in MEF 113

7.2.3.7. Type GeographicSiteRef

Description: A reference to a Geographic Site resource available through Service Site API

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
href	string	Hyperlink to the referenced GeographicSite. Hyperlink MAY be used by the Seller in responses. Hyperlink MUST be ignored by the Seller in case it is provided by the Buyer in a request	Not represented in MEF 113

Name	Type	Description	MEF 113
id*	string	Identifier of the referenced Geographic Site.	Not represented in MEF 113

7.2.3.8. Type GeographicSubAddress

Description: Additional fields used to specify an address, as detailed as possible.

Name	Type	Description	MEF 113
buildingName	string	Allows for identification of places that require building name as part of addressing information	Building Name
id	string	Unique Identifier of the subAddress	Not represented in MEF 113
levelNumber	string	Used where a level type may be repeated e.g. BASEMENT 1, BASEMENT 2	Level Number
levelType	string	Describes level types within a building	Level Type
privateStreetName	string	"Private streets internal to a property (e.g. a university) may have internal names that are not recorded by the land title office	Private Street Name
privateStreetNumber	string	Private streets numbers internal to a private street	Private Street Number
subUnit	MEFSubUnit[]	Representation of a MEFSubUnit It is used for describing subunit within a subaddress e.g.BERTH, FLAT, PIER, SUITE, SHOP, TOWER, UNIT, WHARF.	Not represented in MEF 113

7.2.3.9. enum MEFBuyerSellerType

Description: An enumeration with buyer and seller values.

Value MEF 113

buyer BUYER

seller SELLER

7.2.3.10. Type MEFByteSize

Description: A size represented by value and Byte units

Name	Type	Description	MEF 113
amount	float	Numeric value in a given unit	Value
units	DataSizeUnit	Byte Unit	Unit

7.2.3.11. Type MEFGeographicPoint

Description: A MEFGeographicPoint defines a geographic point through coordinates.

Reference: MEF 79 (Sn 8.9.5)

Inherits from:

- [RelatedPlaceRefOrValue](#)

Name	Type	Description	MEF 113
spatialRef*	string	The spatial reference system used to determine the coordinates (e.g. "WGS84"). The system used and the value of this field are to be agreed upon during the onboarding process.	Spatial Reference
x*	string	The latitude expressed in the format specified by the `spacialRef`	Latitude
y*	string	The longitude expressed in the format specified by the `spacialRef`	Longitude
z	string	The elevation expressed in the format specified by the `spacialRef`	Elevation

7.2.3.12. Type MEFSubUnit

Description: Allows for sub unit identification

Name	Type	Description	MEF 113
------	------	-------------	---------

Name	Type	Description	MEF 113
subUnitNumber*	string	The discriminator used for the subunit, often just a simple number but may also be a range.	Sub Unit Name
subUnitType*	string	The type of subunit e.g.BERTH, FLAT, PIER, SUITE, SHOP, TOWER, UNIT, WHARF.	Sub Unit Type

7.2.3.13. Type Note

Description: Extra information about a given entity. Only useful in processes involving human interaction. Not applicable for an automated process.

Name	Type	Description	MEF 113
author*	string	Author of the note	Author
date*	date-time	Date of the note	Note Date
id*	string	Identifier of the note within its containing entity (may or may not be globally unique, depending on provider implementation)	Not represented in MEF 113
source*	MEFBuyerSellerType	Indicates if this Note was added by the Buyer or Seller.	Note Source
text*	string	Text of the note	Note Text

7.2.3.14. Type RelatedContactInformation

Description: Contact data for a person or organization that is involved in a given context. It is specified by the Seller (e.g. Seller Contact Information) or by the Buyer.

Name	Type	Description	MEF 113
emailAddress*	string	Email address	Contact email Address
name*	string	Name of the contact	Contact Name
number*	string	Phone number	Contract Phone Number
numberExtension	string	Phone number extension	Contract Phone Number Extension"

Name	Type	Description	MEF 113
organization	string	The organization or company that the contact belongs to	Contact Organization
postalAddress	FieldedAddress	Identifies the postal address of the person or office to be contacted.	Contact Postal Address
role*	string	A role the party plays in a given context.	Not represented in MEF 113

7.2.3.15. Type RelatedEntity

Description: A reference to an entity, where the type of the entity is not known in advance.

Name	Type	Description	MEF 113
@referredType*	string	The actual type of the target instance when needed for disambiguation.	Not represented in MEF 113
href	string	Reference of the related entity.	Not represented in MEF 113
id*	string	Unique identifier of a related entity.	Product Identifier
role*	string	The role of an entity.	Not represented in MEF 113

7.2.3.16. Type RelatedPlaceRefOrValue

Description: Defines the Place (Address or Site) by reference or by value.

Name	Type	Description	MEF 113
@schemaLocation	uri	A URI to a JSON-Schema file that defines additional attributes and relationships. May be used to define additional related place types. Usage of this attribute must be agreed upon between Buyer and Seller.	Not represented in MEF 113

Name	Type	Description	MEF 113
@type*	string	This field is used as a discriminator and is used between different place representations. This type might discriminate for additional related place as defined in '@schemaLocation'.	Not represented in MEF 113
role*	string	Role of this place	RelatedPlaceRefOrValue

7.2.3.17. **enum** TimeDurationUnits

Description: The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.

Value

NS

US

MS

SEC

MIN

HOURL

DAY

WEEK

MONTH

YEAR

7.2.3.18. Type TimePeriod

Description: A period of time, either as a deadline (endTime only) a startTime only, or both

Name	Type	Description	MEF 113
endTime	date-time	End of the time period, using IETC-RFC-3339 format	Not represented in MEF 113
startTime	date-time	Start of the time period, using IETC-RFC-3339 format. If you define a start, you must also define an end	Not represented in MEF 113

7.2.3.19. Type TimeSlot

Description:

Name	Type	Description	MEF 113
validFor*	TimePeriod	The Date and Time interval the Seller Technician is scheduled to arrive at the Appointment.	Appointment Timeslot

7.2.4. Notification registration

Notification registration and management are done through [/hub](#) API endpoint. The below sections describe data models related to this endpoint.

7.2.4.1. Type EventSubscription

Description: Sets the communication endpoint address the service instance must use to deliver notification information

Name	Type	Description	MEF 113
callback*	string	The value provided by the Buyer in 'EventSubscriptionInput' during notification registration	Notification Target Information
id*	string	An identifier of the event subscription assigned by the Seller when a resource is created.	Not represented in MEF 113
query	string	This attribute is used to define notification registration constraints.	List of Notification Event Types, Action

7.2.4.2. Type EventSubscriptionInput

Description: This class is used to register for Notifications.

Name	Type	Description
callback*	string	This callback value must be set to *host* property from Buyer Notification API (ap This property is appended with the base path and notification resource path specifie which notification is sent. E.g. for "callback": "http://buyer.com/listenerEndpoint", will be sent to: 'http://buyer.com/listenerEndpoint/mefApi/sonata/appointmentNotification/v2/listenerEndpoint'
query	string	This attribute is used to define to which type of events to register to. Example: "que AppointmentStatusChangeEvent". To subscribe for more than one event type, put tl 'eventType=AppointmentStatusChangeEvent,AppointmentResolvedEvent'. The po 'AppointmentEventType' in AppointmentNotification.api.yaml. An empty query is t ending in subscription for all event types.

7.3. Notification API Data Model

7.3.1. Common Notification.

7.3.1.1. Type Event

Description: Event class is used to describe information structure used for notification.

Name	Type	Description	MEF 113
eventId*	string	Id of the event	Not represented in MEF 113
eventTime*	date-time	Date time when the event occurred	Not represented in MEF 113

7.3.2. Appointment Notification.

Figure 19 presents the whole WorkOrder Management data model the data types, requirements related to them and mapping to MEF 113 specifications are discussed later in this section.

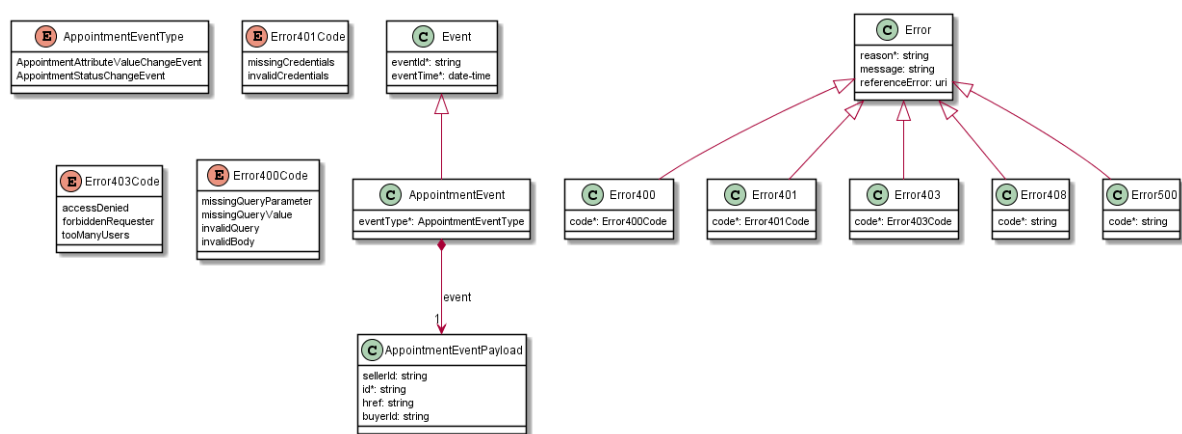


Figure 23. Appointment Notification Data Model

7.3.2.1. Type AppointmentEvent

Description:

Inherits from:

- [Event](#)

Name	Type	Description	MEF 113
eventType*	AppointmentEventType	Indicates the type of the event.	Notification Type

Name	Type	Description	MEF 113
event*	AppointmentEventPayload	A reference to the object that is source of the notification.	Not represented in MEF 113

7.3.2.2. Type AppointmentEventPayload

Description: The identifier of the Appointment being subject of this event.

Name	Type	Description	MEF 113
sellerId	string	The unique identifier of the organization that is acting as the Seller. MUST be specified in the request only when requester entity represents more than one Seller.	Not represented in MEF 113
id*	string	ID of the Appointment attributed by quoting system	Not represented in MEF 113
href	string	Hyperlink to access the Appointment	Not represented in MEF 113
buyerId	string	The unique identifier of the organization that is acting as the a Buyer. MUST be specified in the request only when the responding represents more than one Buyer.	Not represented in MEF 113

7.3.2.3. Enum AppointmentEventType

Description: Type of the Appointment Event

Value	MEF 113
appointmentAttributeValueChangeEvent	APPOINTMENT_UPDATE
appointmentStatusChangeEvent	APPOINTMENT_STATUS_CHANGE

7.3.3. Workorder Notification.

Figure 18 presents the whole WorkOrder Management data model the data types, requirements related to them and mapping to MEF 113 specifications are discussed later in this section.

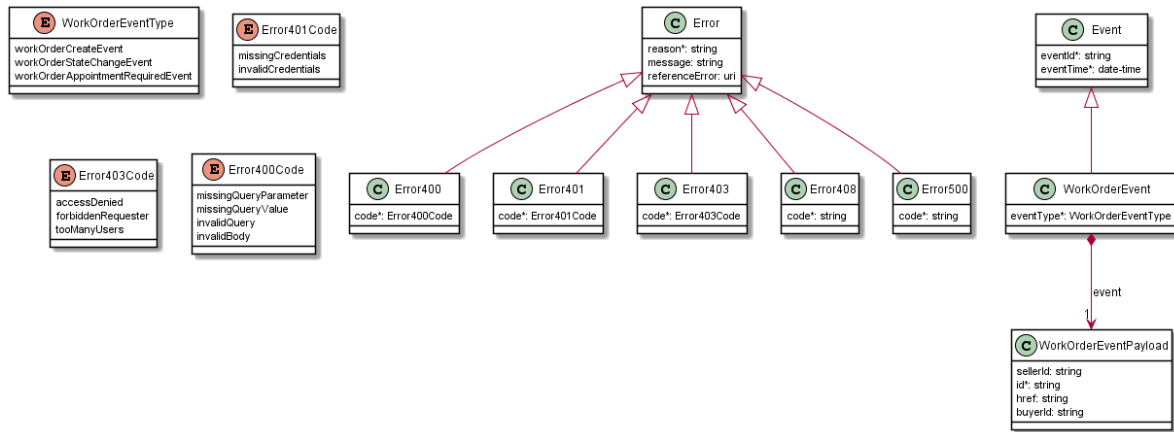


Figure 24. Workorder Notification Data Model

7.3.3.1. Type WorkOrderEvent

Description:

Inherits from:

- [Event](#)

Name	Type	Description	MEF 113
eventType*	WorkOrderEventType	Indicates the type of the event.	Notification Type
event*	WorkOrderEventPayload	A reference to the object that is source of the notification.	Not represented in MEF 113

7.3.3.2. Type WorkOrderEventPayload

Description: The identifier of the WorkOrder being subject of this event.

Name	Type	Description	MEF 113
sellerId	string	The unique identifier of the organization that is acting as the Seller. MUST be specified in the request only when requester entity represents more than one Seller.	Not represented in MEF 113
id*	string	ID of the WorkOrder attributed by quoting system	Not represented in MEF 113

Name	Type	Description	MEF 113
href	string	Hyperlink to access the WorkOrder	Not represented in MEF 113
buyerId	string	The unique identifier of the organization that is acting as the a Buyer. MUST be specified in the request only when the responding represents more than one Buyer.	Not represented in MEF 113

7.3.3.3. **Enum** WorkOrderEventType

Description: Type of the WorkOrder Event

Value	MEF 113
workOrderCreateEvent	WORKORDER_CREATE
workOrderStateChangeEvent	WORKORDER_STATE_CHANGE
workOrderAppointmentRequiredEvent	WORKORDER_APPOINTMENT_REQUIRED

8. References

- [OAS-v3] [Open API 3.0](#), February 2020
- [MEF55.1] [MEF 55.1](#), Lifecycle Service Orchestration (LSO): Reference Architecture and Framework, February 2021
- [MEF79] [MEF 79](#), Address, Service Site, and Product Offering Qualification Management, Requirements and Use Cases, November 2019
- [MEF80] [MEF 80](#), Quote Management Requirements and Use Cases, July 2021
- [MEF113] [MEF W113](#), Trouble Ticketing Business Requirements and Use Cases, January 2022, Draft Standard (R3)
- [MEF124] [MEF W124](#), LSO Cantata and LSO Sonata Trouble Ticket Management API - Developer Guide, June 2022
- [REST] [Chapter 5: Representational State Transfer \(REST\)](#), Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures (Ph.D.).
- [RFC2119] [RFC 2119](#), Key words for use in RFCs to Indicate Requirement Levels, March 1997
- [RFC3986] [RFC 3986](#), Uniform Resource Identifier (URI): Generic Syntax, January 2005
- [RFC8174] [RFC 8174](#), Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, May 2017
- [TMF630] [TMF 630](#), TMF630 API Design Guidelines 4.2.0

- [TMF646] [TMF 646](#), TMF646 Appointment Management API REST Specification R19.0.1
- [TMF697] [TMF 697](#), TMF697 Work Order Management API REST Specification v4.0.0