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.

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. Description
  - 4.2. Conventions in the Document
  - 4.3. Relation to Other Documents
  - 4.4. Approach
  - 4.5. High-Level Flow
- 5. API Description
  - 5.1. High-level use cases
  - 5.2. Resource/endpoint Description
    - 5.2.1. Seller Side Endpoints
    - 5.2.2. Specifying the Buyer ID and the Seller ID
  - 5.3. API Resource Schema summary
    - 5.3.1. Fielded Address
    - 5.3.2. Formatted Address
    - 5.3.3. Geographic Point
    - 5.3.4. Geographic Address Label
  - 5.4. Model Structural Validation
  - 5.5. Security Considerations
- 6. API Interaction & Flows
  - 6.1. Use case 3: Retrieve Service Site List
  - 6.2. Use case 4: Retrieve Service Site by Identifier
- 7. API Details
  - 7.1. API patterns
    - 7.1.1. Indicating errors
  - 7.2. API Data model
    - 7.2.1. Geographic Site
    - 7.2.2. Geographic Address Representation
- 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.

<table>
<thead>
<tr>
<th>Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amartus</td>
</tr>
<tr>
<td>Colt</td>
</tr>
<tr>
<td>Lumen Technologies</td>
</tr>
<tr>
<td>NEC/Netcracker</td>
</tr>
<tr>
<td>Orange</td>
</tr>
<tr>
<td>Proximus</td>
</tr>
<tr>
<td>Spirent Communications</td>
</tr>
</tbody>
</table>

Table 1: Contributing Members

1. Abstract

This standard is intended to assist the implementation of the Site Retrieval functionality defined for the LSO Cantata and LSO Sonata Interface Reference Points (IRPs), for which requirements and use cases are defined in MEF 79 Address, Service Site, and Product Offering Qualification Management Requirements and Use Cases [MEF79] and MEF W79.0.2 Amendment to MEF 79: Address Validation [MEF79.0.2]. This standard consists of this document and complementary API definitions.

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

commit id: 2062c16db194adc5109d0b7c0578a1a9128c6471

• productApi/serviceability/site/geographicSiteManagement.api.yaml

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

commit id: fd4aad8d6417b6aed2fa4e2d4ffa9836648addb0

• productApi/serviceability/site/geographicSiteManagement.api.yaml
# 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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>A way of specifying an absolute fixed location on earth using pre-established boundary and identifier information such as country, city, postal code and street information.</td>
<td>[MEF79]</td>
</tr>
<tr>
<td>Application Program Interface (API)</td>
<td>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.</td>
<td>[MEF55.1]</td>
</tr>
<tr>
<td>Buyer</td>
<td>In the context of this document, denotes the organization or individual acting as the customer in a transaction over a Cantata (Customer &lt;-&gt; Service Provider) or Sonata (Service Provider &lt;-&gt; Partner) Interface.</td>
<td>This document; adapted from [MEF80]</td>
</tr>
<tr>
<td>Geographic Address Label</td>
<td>An identifier that is unique within the Administrative Authority that controls assignment of the label and that specifies a fixed location on earth.</td>
<td>[MEF79.0.2]</td>
</tr>
<tr>
<td>Fielded Address</td>
<td>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, &quot;street number&quot; is one field, &quot;street name&quot; is another field, etc.</td>
<td>[MEF79]</td>
</tr>
<tr>
<td>Formatted Address</td>
<td>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.</td>
<td>[MEF79]</td>
</tr>
<tr>
<td>OpenAPI</td>
<td>RESTful API Documentation Specification for machine-readable interface files for describing, producing, consuming, and visualizing RESTful web services.</td>
<td>[OAS-V3]</td>
</tr>
<tr>
<td>Requesting Entity</td>
<td>The business organization that is acting on behalf of one or more Buyers. In the most common case, the Requesting Entity is the Seller.</td>
<td>[MEF79]</td>
</tr>
<tr>
<td><strong>Responding Entity</strong></td>
<td>Entity represents only one Buyer and these terms are then synonymous.</td>
<td>[MEF79]</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>REST API</strong></td>
<td>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.</td>
<td>[REST]</td>
</tr>
<tr>
<td><strong>Seller</strong></td>
<td>In the context of this document, denotes the organization acting as the supplier in a transaction over a Cantata (Customer &lt;-&gt; Service Provider) or Sonata (Service Provider &lt;-&gt; Partner) Interface.</td>
<td>This document; adapted from [MEF80]</td>
</tr>
<tr>
<td><strong>Geographic Site</strong></td>
<td>A fixed physical location at which a Product can be installed. Its location can be described either with geocodes (Lat/Long information) or by association with an Address or Geographic Address Label. This association may include a Sub-address describing where within that Address or Geographic Address Label this particular Geographic Site is located.</td>
<td>[MEF79.0.2]</td>
</tr>
</tbody>
</table>

### 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.

### 4. Introduction
This standard specification document describes the Application Programming Interface (API) for Site Retrieval functionality of the LSO Cantata Interface Reference Point (IRP) and 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:

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

The business requirements and use cases for Site Retrieval are defined in Address, Service Site, and Product Offering Qualification Management Requirements and Use Cases (MEF 79) [MEF79] and Amendment to MEF 79: Address Validation (MEF W79.0.2) [MEF79.0.2].

This document focuses on implementation aspects of Site Retrieval functionality and is structured as follows:

- **Chapter 4** provides an introduction to Site Retrieval 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. Description

A Site usually represents a location where the Seller has already delivered one or more products. A Site identifier is assigned at some point by the Seller to reference the location. A Site may be reflected by one or more types of addresses.

The point of Site Retrieval API is to allow the Buyer to retrieve details about the location identified by a GeographicAddress.

4.2. 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. GeographicAddress).
• In UML diagrams the default cardinality of associations is 0..1. Other cardinality markers are complaint 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 indicates a variable to be substituted with a correct value.

4.3. Relation to Other Documents

The requirements and use cases for Site Retrieval functionality are defined in MEF 79 [MEF79] and MEF 79.0.2 [MEF79.0.2]. The API definition builds on TMF 674 API as specified by TMF674 Geographic Site Management API User Guide v4.0.1 [TMF674].

4.4. 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 Site Retrieval is product-agnostic in its nature and is not intended to carry any product-specific payloads. It operates using the Generic API Framework and the Function-specific Information and Operations.

4.5. High-Level Flow

Site Retrieval is 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 Site Retrieval's 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.

- **Trouble Ticketing:**
  - Allows the Buyer to create, retrieve, and update Trouble Tickets as well as receive notifications about Incidents' and Trouble Tickets' updates. This allows managing issues and situations that are not part of normal operations of the Product provided by the Seller.
5. API Description

This section discusses the API structure and design patterns. It starts with the high-level use cases diagram and then it describes the REST endpoints with use case mapping.

5.1. High-level use cases

Figure 4 presents a high-level use case diagram as specified in MEF 79 [MEF79] for Site Retrieval in section 7.1. This picture aims to help understand the endpoint mapping. Use cases are described extensively in chapter 6.

*Note:* Use Cases keep the original numbering coming from MEF 79.

![Figure 4: High-level use cases](image)

5.2. Resource/endpoint Description

5.2.1. Seller Side Endpoints

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

{{?/seller_prefix}}/mefApi/cantata/geographicSiteManagement/v1/

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

{{?/seller_prefix}}/mefApi/sonata/geographicSiteManagement/v7/

*Note:* All examples will include only the Sonata version of the Base Path.

Following endpoints are exposed by the Seller and allow the Buyer to:

- perform a query for a Geographic Site list
- get a single Geographic Site by id

The endpoints and corresponding data model are defined in

productApi/serviceability/site/geographicSiteManagement.api.yaml.
## 5.2.2. 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 `geographicSiteManagement.api.yaml`.

**[R2]** 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].
[R3] If the Requesting Entity represents precisely one Buyer with the Responding Entity, the request **MUST NOT** specify the `buyerId` [MEF79 R81].

[R4] 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].

[R5] If the Responding Entity represents precisely one Seller to this Buyer, the request **MUST NOT** specify the `sellerId` [MEF79 R83].

5.3. API Resource Schema summary

This subchapter describes the resource model used by the API.

Each entity is a simple or composed type (with the use of `allOf` keyword for data types composition). A simple type defines a set of properties that might be of an object, primitive, or reference type.

Section 6 provides examples of data model and API usage. For a detailed description of the data model, please refer to API Details.

Figure 5 presents the whole data model of the API.
Figure 5: Data Model

Note: While showing the extends relation, for clarity, the extending type lists only own attributes without the inherited ones.

[R6] If an entity is used in the payload, all properties marked as required (*) MUST be provided.

The root entity is the GeographicSite. It is used by both endpoints. Apart from a few simple attributes it defines referential attributes:

- relatedContactInformation - to provide the details of the person or organization at the specific site location that is the local contact. This contact will primarily be used for gaining access to the site.
• place - a set of location descriptions, each of which describes where this Geographic Site is located.

**Note:** place is a set because a particular Geographic Site might be described with multiple locations. For example, one Geographic Site might have two Fielded Addresses (for a building on the corner of two streets). It also can store several Address format representations of the same single Address.

The place can be specified with the use of `GeoAddressRefOrValue`. This means that a value representation of any of possible Address Types can be provided, or a reference to already known (by id) Address.

**Note:** In MEF approach a Site cannot be a part of Site nor describes it’s place. That is why `GeoAddressRef` is used instead of standard `PlaceRef` - to ensure that a Site will not be used accidentally.

There are four possible types of address representation:

- FieldedAddress
- FormattedAddress
- GeographicAddressLabel
- MEFGeographicPoint

[R7] A Buyer MUST support at least one of Fielded Addresses or Formatted Addresses to describe locations. [MEF79 R84]

[R8] A Seller MUST support at least one of Fielded Addresses or Formatted Addresses to specify a location. [MEF79 R85]

The mandatory `@type` attribute of `GeoAddressRefOrValue` is used as a discriminator. It is possible for the Buyer and the Seller to go beyond those four specified type. This can be done with the use of the `@schemaLocation` attribute that will point to the schema defining the model of the new agreed address representation. Using additional address schema must be bilaterally agreed during the onboarding process.

5.3.1. Fielded Address

```json
{
    "@type": "FieldedAddress",
    "streetType": "ul.",
    "streetName": "Edmunda Wasilewskiego",
    "streetNr": "20",
    "streetNrSuffix": "14",
    "city": "Kraków",
    "stateOrProvince": "Lesser Poland",
    "postcode": "30-305",
    "country": "Poland",
    "geographicSubAddress": {
        "levelType": "floor",
```
Example of a Geographic Address of type Fielded Address. The type discriminator has the value FieldedAddress. A subset of available attributes is used to describe the Geographic Address. The Fielded Address has an optional geographicSubAddress structure that defines several attributes that can be used in case precise address information has to be provided. In the example above, a floor in the building at the given address is specified using this structure.

5.3.2. Formatted Address

```json
{
  "@type": "FormattedAddress",
  "addrLine1": "ul. Edmund Wasilewskiego 28/14",
  "addrLine2": "Floor 4",
  "city": "Kraków",
  "stateOrProvince": "Lesser Poland",
  "postcode": "30-305",
  "country": "Poland"
}
```

Geographic Address of type Formatted Address. The type discriminator has the value FormattedAddress. This example contains the same information as the previous FieldedAddress example.

5.3.3. Geographic Point

```json
{
  "@type": "MEFGeographicPoint",
  "spatialRef": "EPSG:4326 WGS 84",
  "x": "50.048868",
  "y": "19.929523"
}
```

Place information in a form of geographic point. spatialRef determines the standard that has to be used to interpret coordinates provided in x (latitude), y (longitude), and z (elevation) values.

This type allows only providing a point. It cannot carry more detailed information like the floor number from previous examples.

[R17] The spatialRef value that can be used MUST be agreed upon between Buyer and Seller during the onboarding process.

5.3.4. Geographic Address Label
The Geographic Address Label represents a unique identifier controlled by a generally accepted independent Administrative Authority or standard (externalReferenceType) that specifies a fixed geographical location. The example above is a place that represents a CLLI (Common Language Location Identifier) identifier which is commonly used to refer locations in North America for network equipment installations.

5.4. Model Structural Validation

The structure of the HTTP payloads exchanged via Site Retrieval API endpoints is defined in the OpenAPI document which is integral part of this standard.

[R9] 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 Interaction & Flows

This section provides a detailed insight into the API functionality, use cases, and flows. First a list of business use cases is presented, followed by examples and an explanation of all usage aspects involved.

Table 2 keeps the original use case numbering and naming from MEF 79. The descriptions use API naming.

<table>
<thead>
<tr>
<th>Use Case #</th>
<th>Use Case Name</th>
<th>Use Case Description</th>
</tr>
</thead>
</table>
3  
Retrieve Service Site List  
The Buyer requests that the Seller provides a list of GeographicSites known to the Seller based on a set of Geographic Site/Address filter criteria. For each GeographicSite returned, the Seller also provides a GeographicSite.id, which uniquely identifies it within the Seller.

4  
Retrieve Service Site by Site Identifier  
The Buyer requests the full details for a single GeographicSite based on a GeographicSite.id that was previously provided by the Seller.

**Table 2. Use cases description**

*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.*

### 6.1. Use case 3: Retrieve Service Site List

The flow is a simple request - response pattern, as presented on Figure 6:

![Figure 6. Use case 3: Retrieve Service Site List flow](image)

The Buyer can retrieve a list of GeographicSites by using a GET /geographicSite operation with desired filtering criteria.

The Buyer may search for Geographic Sites with use of following attributes:

- companyName
- customerName
- description
- name
- serviceSiteContactName
- siteType
[R10] The Buyer **MUST** specify the Geographic Site Location either by identifier - geographicAddress.id OR by value with use of streetNr, streetName, streetType, city, postcode, country.

[MEF79 R17]

**Note:** MEF 79 requires that the streetName, city, postcode, and country are provided when the Unknown Address ID method is used [MEF79 R90]. However, for expected future use case enhancement the API does not mandates those attributes to be populated, allowing for more open query (this will be reflected in the future revisions of the Business Requirements Document). The Seller is expected to return an Error422 with tooManyRecords code if the search criteria are to wide.

The example below shows a query that aims to find public (siteType=public) GeographicSites available at known location with geographicAddress.id=00000000-0000-0030-0305-873500002014.

https://seller.com/mefApi/sonata/geographicSiteManagement/v7/geographicSite?siteType=public&geographicAddress.id=00000000-0000-0030-0305-873500002014

After receiving the request, the Seller validates it. Then the Seller attempts to match the Buyer's provided criteria with own Site information. The determination of what is considered a match is at the Seller's discretion. If the request filter criteria match one or more Sites known to the Seller, a response is returned with the site information including an id for each Site returned.

[R11] In case of no matching records found, the Seller **MUST** return a valid 200 response with an empty list.

[R12] In case of too many matching records found (the definition of 'too many' is up to Seller's discretion), the Seller **MUST** return an Error422 with code equal to tooManyRecords.

[R13] For each GeographicSite returned, the Seller **MUST** specify the GeographicSite.id attribute. [MEF79 R23]

[R14] For each GeographicSite returned, the Seller **MUST** specify at least one place using any of available Address Types. (Based on the default agreed upon by this Buyer/Seller pair). [MEF79 R24]
Below you can find a response with 1 matching item:

```json
[
  {
    "id": "12340000-0000-0030-0305-8735000002014",
    "href": "{{baseUrl}}/geographicSite/12340000-0000-0030-0305-8735000002014",
    "description": "This is a description of the Site",
    "name": "Wasilewskiego Data Centre",
    "companyName": "Building Owner Co.",
    "customerName": "Data Centre Space Mgt Co.",
    "siteType": "public",
    "@type": "GeographicSite",
    "place": {
      "@type": "GeographicAddressRef",
      "id": "00000000-0000-0030-0305-8735000002014"
    }
  },
  "relatedContactInformation": {
    "emailAddress": "john.example@buildingowner.com",
    "name": "John Example",
    "number": "12-345-678-900",
    "organization": "Building Owner Co.",
    "role": "Building Owner Contact",
    "postalAddress": {
      "@type": "FioIidedAddress",
      "streetNr": "20",
      "streetNrSuffix": "14",
      "streetName": "Edmundas Wasilewskiego",
      "city": "Kraków",
      "stateOrProvince": "Lesser Poland",
      "postcode": "30-305",
      "country": "Poland"
    }
  }
]
```

**Note:** Not having any `GeographicSite` provided for given Address does not indicate if the Seller is able to serve any type of Product there. For the Buyer to proceed with latter steps, it is sufficient to use the `GeographicAddress`.

The business requirements of each of the use cases are described in sections 7.2, 8.2, and 8.10 of MEF 79 [MEF79] and their updates in MEF 79.0.2 [MEF79.0.2].

### 6.2. Use case 4: Retrieve Service Site by Identifier

To get detailed up to date information about the Site, the Buyer sends a Retrieve Site by Identifier Request using a `GET /geographicSite/{id}` operation.

The flow is a simple request - response pattern, as presented on Figure 7:
Figure 7. Use case 4: Retrieve Service Site by Identifier flow

Example request and response:

GET /meAPI/sonata/geographicSiteManagement/v7/geographicSite/0000000-0000-0305-0305-87350002014

```
[
  {
    "id": "56787654-0000-0305-0305-000000001234",
    "href": "{(baseUrls)}/geographicSite/56787654-0000-0305-0305-000000001234",
    "description": "This is a description of the Site",
    "name": "Main DC",
    "siteType": "public",
    "@type": "GeographicSite",
    "place": [
      {
        @type": "GeographicAddressRef",
        "id": "11223456-0000-0305-456789012345"
      }
    ]
  }
]
```

[R15] In case id does not find a GeographicSite in Seller's system, an error response 404 MUST be returned.

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 Site retrieval API uses the error responses 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 case the error message body structure might be aligned with the Error.
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>message</code></td>
<td><code>string</code></td>
<td>Text that provides mode details and corrective actions related to the error. This can be shown to a client user.</td>
</tr>
<tr>
<td><code>reason</code></td>
<td><code>string</code></td>
<td>Text that explains the reason for the error. This can be shown to a client user.</td>
</tr>
<tr>
<td><code>referenceError</code></td>
<td><code>uri</code></td>
<td>URL pointing to documentation describing the error</td>
</tr>
</tbody>
</table>

7.1.1.2. Type Error400

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

Inherits from:

- **Error**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code*</td>
<td><code>string</code></td>
<td>One of the following error codes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- missingQueryParameter: The URI is missing a required query-string parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- missingParameterValue: The URI is missing a required query-string parameter value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- invalidQuery: The query section of the URI is invalid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- invalidBody: The request has an invalid body</td>
</tr>
</tbody>
</table>

7.1.1.3. Type Error401
**Description:** Unauthorized. (https://tools.ietf.org/html/rfc7235#section-3.1)

Inherits from:

- Error

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code*</td>
<td>string</td>
<td>One of the following error codes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- missingCredentials: No credentials provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- invalidCredentials: Provided credentials are invalid or expired</td>
</tr>
</tbody>
</table>

7.1.1.4. Type Error403

**Description:** Forbidden. (https://tools.ietf.org/html/rfc7231#section-6.5.3)

Inherits from:

- Error

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code*</td>
<td>string</td>
<td>This code indicates that the server understood the request but refuses to authorize it because of one of the following error codes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- accessDenied: Access denied</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- forbiddenRequester: Forbidden requester</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- tooManyUsers: Too many users</td>
</tr>
</tbody>
</table>

7.1.1.5. Type Error404

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

Inherits from:

- Error

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code*</td>
<td>string</td>
<td>The following error code:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- notFound: A current representation for the target resource not found</td>
</tr>
</tbody>
</table>

7.1.1.6. Type Error422

The response for HTTP status 422 is a list of elements that are structured using 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.  

Inherits from:

- **Error**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| code*    | string   | 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) |
| 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). |

7.1.1.7. **Type Error500**

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

Inherits from:

- **Error**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

24 / 39
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code*</td>
<td>string</td>
<td>- internalError: Internal server error - the server encountered an unexpected condition that prevented it from fulfilling the request.</td>
</tr>
</tbody>
</table>

7.2. API Data model

Figure 9 presents the Site Retrieval data model. The data types, requirements related to them, and mapping to MEF 79 and MEF 79.0.2 specifications are discussed later in this section.

This data model is used to construct requests and responses of the API endpoints described in Section 5.2.1
Figure 9. Site Retrieval Data Model

7.2.1. Geographic Site

7.2.1.1 Type GeographicSite

**Description:** A fixed physical location at which a Product can be installed. Its location can be described either with geographic point (Lat/Long information) or by association with an Address or Geographic Address Label. This association may include a Sub-address describing where within that Address or Geographic Address Label this particular Geographic Site is located.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>id*</td>
<td>string</td>
<td>Identifier of the Geographic Site unique within the Seller.</td>
<td>Service Site Identit</td>
</tr>
<tr>
<td>href</td>
<td>string</td>
<td>Unique reference of the Geographic Site unique within the Seller.</td>
<td>Not represented in 79</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>A textual description of the Geographic Site.</td>
<td>Service Site Descri</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>A user-friendly name for the place, such as [Paris Store], [London Store], [Main Home]</td>
<td>Service Site Name</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>MEF 79</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>companyName</td>
<td>string</td>
<td>The name of the company that is the administrative authority (e.g. controls access) for this Geographic Site. (For example, the building owner.)</td>
<td>Service Site Company Name</td>
</tr>
<tr>
<td>customerName</td>
<td>string</td>
<td>The name of the company that is the administrative authority for the space within this Geographic Site (For example, the company leasing space in a multi-tenant building).</td>
<td>Service Site Customer Name</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>place</td>
<td>GeographicAddressRefOrValue[]</td>
<td>A set of location descriptions, each of which describes where this Geographic Site is located. It is important to note that this is a set because a particular Geographic Site might be described with multiple locations. For example, one Geographic Site might have two Fielded Addresses (for a building on the corner of two streets), two Formatted Addresses, and a Geographic Point.</td>
<td>Service Site Locati</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>MEF 79</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>relatedContactInformation</td>
<td>RelatedContactInformation[]</td>
<td>An entity or organization that is involved to the geographical site, such as the Site Contact or Site Alternate Contact.</td>
<td>Service Site Contact (role=serviceSiteContact)</td>
</tr>
<tr>
<td>siteType</td>
<td>MEFSiteType</td>
<td>This defines whether a Geographic Site is public or private. public - means that the existence of this Geographic Site is public information. A meet-me-room in a hosted data center facility (where all interconnects between parties take place) is an example of a public Geographic Site. A shared facility in the basement of a multi-tenant business building where all</td>
<td>Service Site Restriction Type</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>MEF 79</td>
<td>interconnects between parties take place is another example of a public Geographic Site. private - means that the existence of this Geographic Site is on a need-to-know basis. A wiring closet set up inside a customer facility just to connect two parties is an example of a private Geographic Site. For private sites, the Seller does not return any information regarding the existence of this Geographic Site unless it has been established that this Buyer is authorized to</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 7.2.1.2. `enum` MEFSiteType

**Description:** This defines whether a Geographic Site is public or private.

<table>
<thead>
<tr>
<th>Value</th>
<th>MEF 79</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>PUBLIC</td>
<td>The existence of this Geographic Site is public information. A meet-me-room in a hosted data center facility (where all interconnects between parties take place) is an example of a public Geographic Site. A shared facility in the basement of a multi-tenant business building where all interconnects between parties take place is another example of a public Geographic Site.</td>
</tr>
<tr>
<td>private</td>
<td>PRIVATE</td>
<td>The existence of this Service Site is on a need-to-know basis. A wiring closet set up inside a customer facility just to connect two parties is an example of a private Service Site. For private sites, the Seller does not return any information regarding the existence of this Geographic Site unless it has been established that this Buyer is authorized to obtain this information.</td>
</tr>
</tbody>
</table>

### 7.2.2. Geographic Address Representation

[R16] Buyer and Seller MUST support at least one of `FieldedAddress` or `FormattedAddress` place representations [MEF79 R84 & R85].

### 7.2.2.1 Type GeographicAddressRefOrValue

**Description:** Defines a GeographicAddress link by reference or by value. The polymorphic attributes @type and @schemaLocation are related to the GeographicAddress entity and not the GeographicAddressRefOrValue class itself.
### @schemaLocation

**Type:** uri  
**Description:** 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 the Buyer and the Seller.  
**MEF 79:** Not represented in MEF 79

- This field is used as a discriminator and is used between different geographicAddress representations. This type might discriminate for additional related GeographicAddress as defined in "@schemaLocation".

### 7.2.2.2. 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:

- GeographicAddressRefOrValue

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>city*</td>
<td>string</td>
<td>The city that the address is in</td>
<td>City</td>
</tr>
<tr>
<td>country*</td>
<td>string</td>
<td>The country that the address is in</td>
<td>Country</td>
</tr>
<tr>
<td>geographicSubAddress</td>
<td>GeographicSubAddress</td>
<td>Additional fields used to specify an address, as detailed as possible.</td>
<td>Not represented in MEF 79</td>
</tr>
<tr>
<td>locality</td>
<td>string</td>
<td>An area of defined or undefined Present boundaries within a local authority or other legislatively defined area, usually rural or semirural in nature.</td>
<td>Locality</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>postcode</td>
<td>string</td>
<td>A descriptor for a postal delivery area used to speed and simplify the delivery of mail (also known as zip code) MEF 79 defines it as required however as in certain countries it is not used we make it optional in API. Postal Code</td>
<td></td>
</tr>
<tr>
<td>postcodeExtension</td>
<td>string</td>
<td>An extension of a postal code. E.g. the part following the dash in a US urban property address Postal Code Extension</td>
<td></td>
</tr>
<tr>
<td>stateOrProvince</td>
<td>string</td>
<td>The State or Province that the address is in State Or Province</td>
<td></td>
</tr>
<tr>
<td>streetName*</td>
<td>string</td>
<td>Name of the street or other street type Street Name</td>
<td></td>
</tr>
<tr>
<td>streetNr</td>
<td>string</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>streetNrLast</td>
<td>string</td>
<td>Last number in a range of street numbers allocated to a property Street Number Last</td>
<td></td>
</tr>
<tr>
<td>streetNrLastSuffix</td>
<td>string</td>
<td>Last street number suffix for a ranged address Street Number Suffix Last</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>MEF 79</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>streetNrSuffix</td>
<td>string</td>
<td>The first street number suffix</td>
<td>Street Number Suffix</td>
</tr>
<tr>
<td>streetSuffix</td>
<td>string</td>
<td>A modifier denoting a relative direction</td>
<td>Street Suffix</td>
</tr>
<tr>
<td>streetType</td>
<td>string</td>
<td>The type of street (e.g., alley, avenue, boulevard, brae, crescent, drive, highway, lane, terrace, parade, place, tarn, way, wharf)</td>
<td>Street Type</td>
</tr>
</tbody>
</table>

### 7.2.2.3. Type FormattedAddress

**Description:** A type of Address that has discrete fields for each type of boundary or identifier with the exception of the 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:

- **GeographicAddressRefOrValue**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>addrLine1*</td>
<td>string</td>
<td>The first address line in a formatted address</td>
<td>Address Line 1</td>
</tr>
<tr>
<td>addrLine2</td>
<td>string</td>
<td>The second address line in a formatted address</td>
<td>Address Line 2</td>
</tr>
<tr>
<td>city*</td>
<td>string</td>
<td>The city that the address is in</td>
<td>City</td>
</tr>
<tr>
<td>country*</td>
<td>string</td>
<td>The country that the address is in</td>
<td>Country</td>
</tr>
<tr>
<td>locality</td>
<td>string</td>
<td>An area of defined or undefined boundaries within a local authority or other legislatively defined area, usually rural or semi-rural in nature</td>
<td>Locality</td>
</tr>
<tr>
<td>postcode</td>
<td>string</td>
<td>A descriptor for a postal delivery area used to speed and simplify the delivery of mail (also known as ZIP code)</td>
<td>Postal Code</td>
</tr>
</tbody>
</table>
### 7.2.2.4. Type MEFGeographicPoint

**Description:** A MEFGeographicPoint defines a geographic point through coordinates.

Reference: MEF 79 (Sn 8.9.5), MEF 79.0.2 (Sn 8.9.5)

Inherits from:

- GeographicAddressRefOrValue

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>spatialRef*</td>
<td>string</td>
<td>The spatial reference system used to determine the coordinates (e.g. &quot;WGS84&quot;). The system used and the value of this field are to be agreed during the onboarding process.</td>
<td>Spatial Reference</td>
</tr>
<tr>
<td>x*</td>
<td>string</td>
<td>The latitude expressed in the format specified by the <code>spacialRef</code></td>
<td>Latitude</td>
</tr>
<tr>
<td>y*</td>
<td>string</td>
<td>The longitude expressed in the format specified by the <code>spacialRef</code></td>
<td>Longitude</td>
</tr>
<tr>
<td>z</td>
<td>string</td>
<td>The elevation expressed in the format specified by the <code>spacialRef</code></td>
<td>Elevation</td>
</tr>
</tbody>
</table>

### 7.2.2.5. Type GeographicSubAddress

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>buildingName</td>
<td>string</td>
<td>The well-known name of a building that is located at this Geographic Address (e.g. where there is one Address for a campus).</td>
<td>Building Name</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>MEF 79</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>levelNumber</td>
<td>string</td>
<td>Used where a level type may be repeated e.g. BASEMENT 1, BASEMENT 2</td>
<td>Level Number</td>
</tr>
<tr>
<td>levelType</td>
<td>string</td>
<td>Describes level types within a building</td>
<td>Level Type</td>
</tr>
<tr>
<td>privateStreetName</td>
<td>string</td>
<td>Private streets internal to a property (e.g. a university) may have internal names that are not recorded by the land title office</td>
<td>Private Street Name</td>
</tr>
<tr>
<td>privateStreetNumber</td>
<td>string</td>
<td>Private streets numbers internal to a private street</td>
<td>Private Street Number</td>
</tr>
<tr>
<td>subUnit</td>
<td>MEFSUBUNIT[]</td>
<td>Representation of a MEFSUBUNIT It is used for describing subunit within a subAddress e.g. BERTH, FLAT, PIER, SUITE, SHOP, TOWER, UNIT, WHARF.</td>
<td>Sub Unit List</td>
</tr>
</tbody>
</table>

7.2.2.6. 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), MEF 79.0.2 (Sn 8.9.4)

Inherits from:

- GeographicAddressRefOrValue

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>externalReferenceId*</td>
<td>string</td>
<td>The unique reference to an Address as provided by the Administrative Authority.</td>
<td>Administrative Authority Address Label</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td>MEF 79</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>externalReferenceType*</td>
<td>string</td>
<td>The organization or standard from the organization that administers this Geographic Address Label ensuring it is unique within the Administrative Authority. 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.</td>
<td>Administrative Authority</td>
</tr>
</tbody>
</table>

7.2.2.7. Type MEFSubUnit

**Description:** Allows for sub unit identification

Reference: MEF 79 (Sn 8.9.2)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>subUnitNumber*</td>
<td>string</td>
<td>The discriminator used for the subunit, often just a simple number but may also be a range.</td>
<td>Sub Unit Name</td>
</tr>
<tr>
<td>subUnitType*</td>
<td>string</td>
<td>The type of subunit e.g.BERTH, FLAT, PIER, SUITE, SHOP, TOWER, UNIT, WHARF.</td>
<td>Sub Unit Type</td>
</tr>
</tbody>
</table>

7.2.2.8. Type GeographicAddressRef

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

Inherits from:

- [GeographicAddressRefOrValue](#)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>href</td>
<td>string</td>
<td>Hyperlink to the referenced place Hyperlink <strong>MAY</strong> be used by the Seller in responses Hyperlink <strong>MUST</strong> be ignored by the Seller in case it is provided by the Buyer in a request</td>
<td>Not represented in MEF 79</td>
</tr>
</tbody>
</table>
### 7.3. Common

#### 7.3.1. Type RelatedContactInformation

**Description:** Contact data for a person or organization that is involved in the product offering qualification. In a given context it is always specified by the Seller (e.g. Seller Contact Information) or by the Buyer.

Reference: MEF 79 (Sn 8.11)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>MEF 79</th>
</tr>
</thead>
<tbody>
<tr>
<td>emailAddress*</td>
<td>string</td>
<td>Email address</td>
<td>Contact email Address</td>
</tr>
<tr>
<td>name*</td>
<td>string</td>
<td>Name of the contact</td>
<td>Contact Name</td>
</tr>
<tr>
<td>number*</td>
<td>string</td>
<td>Phone number</td>
<td>Contract Phone Number</td>
</tr>
<tr>
<td>numberExtension</td>
<td>string</td>
<td>Phone number extension</td>
<td>Contract Phone Number Extension</td>
</tr>
<tr>
<td>organization</td>
<td>string</td>
<td>The organization or company that the contact belongs to</td>
<td>Not represented in MEF 79</td>
</tr>
<tr>
<td>role*</td>
<td>string</td>
<td>The role of the particular contact in a given context.</td>
<td>Not represented in MEF 79</td>
</tr>
<tr>
<td>postalAddress</td>
<td>FieldedAddress</td>
<td>Identifies the postal address of the person or office to be contacted.</td>
<td>Not represented in MEF 79</td>
</tr>
</tbody>
</table>

The **role** attribute is used to provide a reason the particular party information is used. It can result from MEF 79 requirements (e.g. Service Site Contact) or from the Product Specification requirements.

The rule for mapping a represented attribute value to a **role** is to use the *lowerCamelCase* pattern e.g.
• business attribute name: "Service Site Contact": role must be serviceSiteContact

Note: MEF 79 does not define the organization and postalAddress attributes. These are added for consistency with Quote and Order APIs.

Note: MEF 79 defines numberExtension as a required attribute, yet for consistency with the rest of APIs it is defined as not required. This will be addressed in anticipated revision of MEF 79.

8. References

• [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
• [MEF79.0.2] MEF 79.0.2, Amendment to MEF 79: Address Validation, July 2021
• [MEF80] MEF 80, Quote Management Requirements and Use Cases, July 2021
• [TMF674] TMF 674 TMF674 Geographic Site Management API User Guide v4.0.1