Letter Ballot
MEF W122

LSO Cantata and LSO Sonata Site Management API - Developer Guide

January 2022

Disclaimer

Copyright

Table of Contents

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
Amartus
Colt
Lumen Technologies
NEC/Netcracker
Orange
Proximus
Spirent Communications

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.

Term	Description	Reference
Address	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.	[MEF79]
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]
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]
Geographic Address Label	An identifier that is unique within the Administrative Authority that controls assignment of the label and that specifies a fixed location on earth.	[MEF79.0.2]
Fielded Address	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.	[MEF79]
Formatted Address	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.	[MEF79]
OpenAPI	RESTful API Documentation Specification for machine-readable interface files for describing, producing, consuming, and visualizing RESTful web services.	[OAS-V3]
Requesting Entity	The business organization that is acting on behalf of one or more Buyers. In the most common case, the Requesting	[MEF79]

	Entity represents only one Buyer and these terms are then synonymous.	
Responding Entity	The business organization that is acting on behalf of one or more Sellers. In the most common case, the Responding Entity represents only one Seller and these terms are then synonymous.	[MEF79]
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]
Geographic Site	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.	[MEF79.0.2]

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.
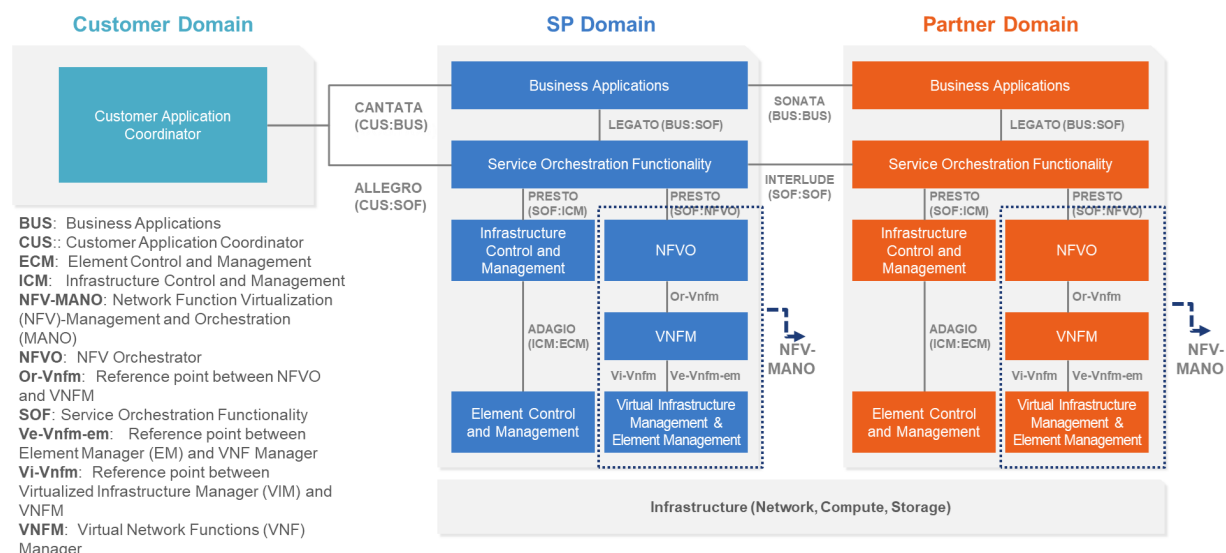
Figure 1. The LSO Reference Architecture

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)
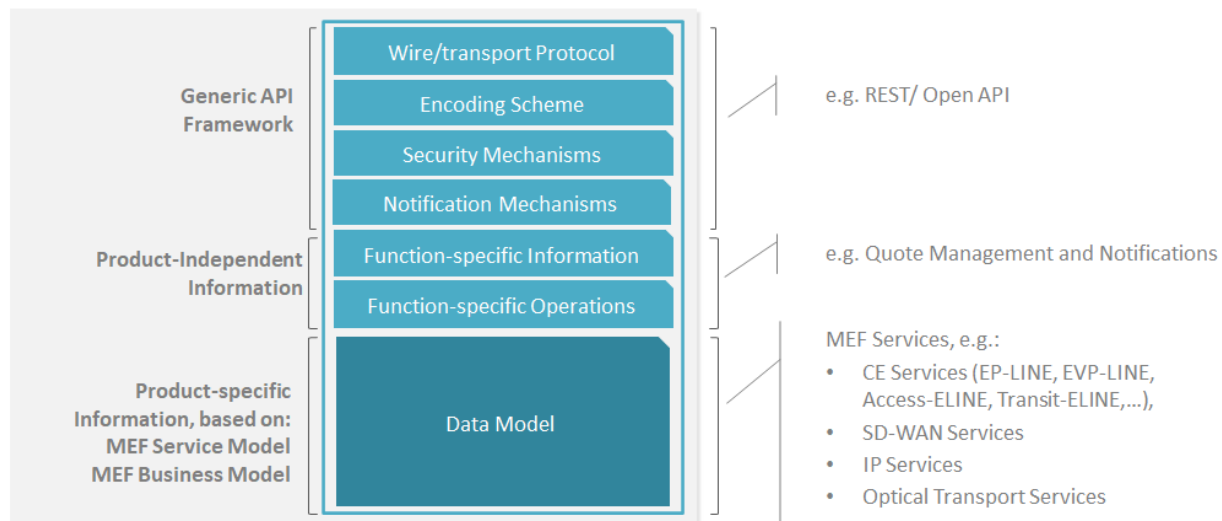
Figure 2. Cantata and Sonata API framework

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 it's 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.
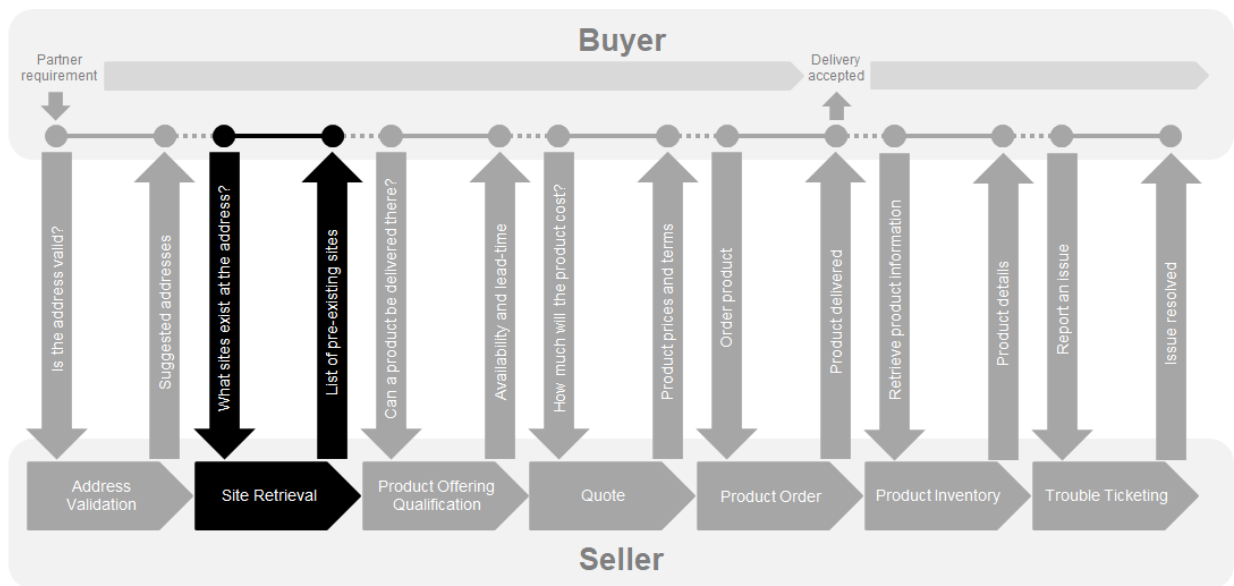
Figure 3. Cantata and Sonata End-to-End Function Flow

- 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.
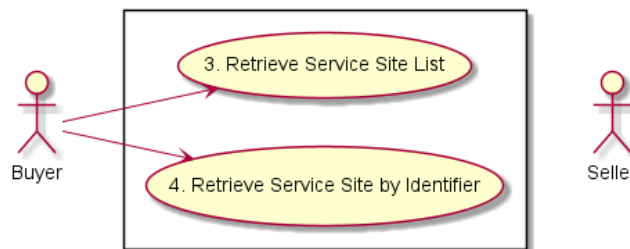
Figure 4: High-level use cases

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`.

API endpoint	Description	MEF 79 Use case Mapping
GET /geographicSite	A request initiated by the Buyer to retrieve a list of GeographicSites from the Seller based on filter criteria provided as query	UC 3: Retrieve Service Site List
GET /geographicSite/{id}	A request initiated by the Buyer to retrieve full details of a single GeographicSite based on on a Geographic Site identifier previously provided by the Seller.	UC 4: Retrieve Service Site by Identifier

[R1] The Buyer implementation **MUST** be able to use both of the above listed endpoints [MEF79 R1].

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.
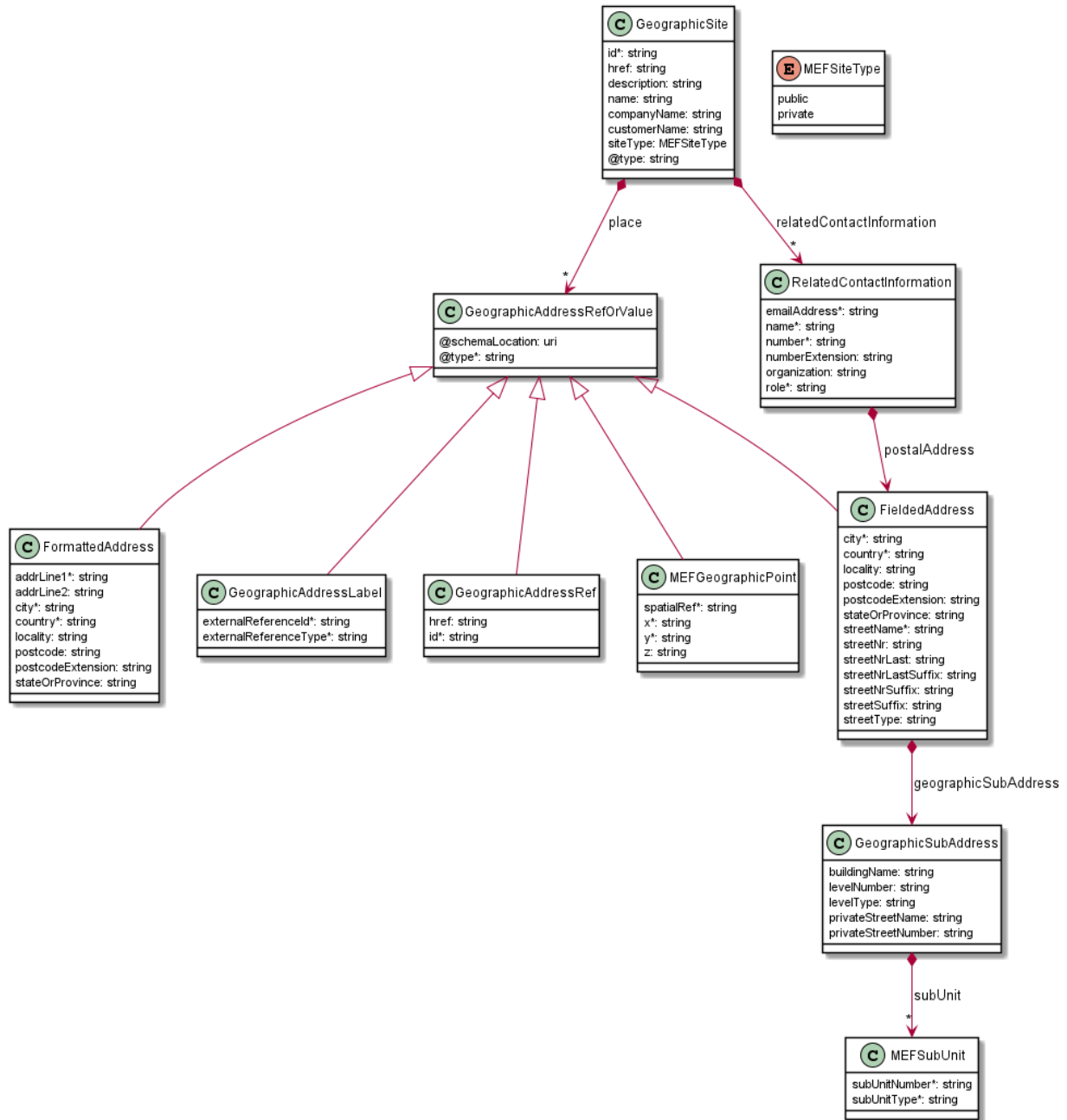
Figure 5: Data Model

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 `GeographicAddressRefOrValue`. 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 `GeographicAddressRef` 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 `GeographicAddressRefOrValue` 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

```
{
  "@type": "FieldedAddress",
  "streetType": "ul.",
  "streetName": "Edmunda Wasilewskiego",
  "streetNr": "20",
  "streetNrSuffix": "14",
  "city": "Kraków",
  "stateOrProvince": "Lesser Poland",
  "postcode": "30-305",
  "country": "Poland",
  "geographicSubAddress": {
    "levelType": "floor",
```

```
        "levelNumber": "4"
    }
  }
}
```

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

```
{
  "@type": "FormattedAddress",
  "addrLine1": "ul. Edmunda Wasilewskiego 20/14",
  "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

```
{
  "@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

```
{
  "@type": "GeographicAddressLabel",
  "externalReferenceType": "CLLI",
  "externalReferenceId": "PLTXCL01"
}
```

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.

Use Case #	Use Case Name	Use Case Description

Use Case #	Use Case Name	Use Case Description
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 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:
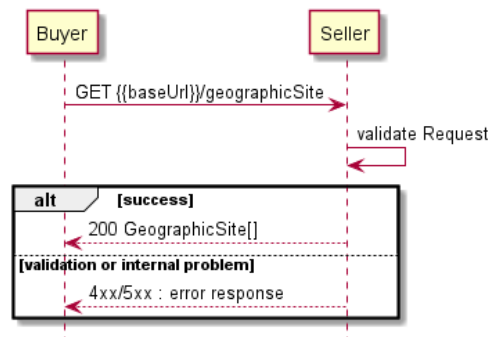

Figure 6. Use case 3: Retrieve Service Site List flow

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`

- `geographicAddress.id`
- `streetNr`
- `streetName`
- `streetType`
- `city`
- `postcode`
- `country`

[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": "12340000-0000-0030-0305-873500002014",
    "href": "{{baseUrl}}/geographicSite/12340000-0000-0030-0305-873500002014",
    "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": "00000000-0000-0030-0305-873500002014"
      }
    ],
    "relatedContactInformation": [
      {
        "emailAddress": "john.example@buildingowner.com",
        "name": "John Example",
        "number": "12-345-678-900",
        "organization": "Building Owner Co.",
        "role": "Building Owner Contact",
        "postalAddress": {
          "@type": "FieldedAddress",
          "streetNr": "20",
          "streetNrSuffix": "14",
          "streetName": "Edmunda 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:
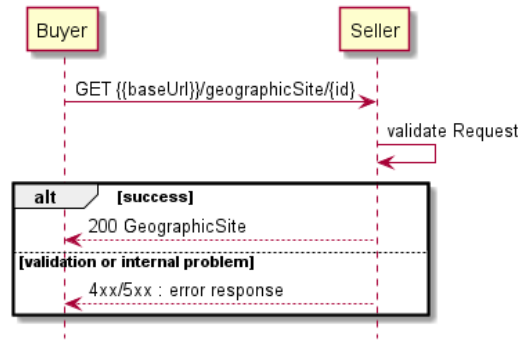
Figure 7. Use case 4: Retrieve Service Site by Identifier flow

Example request and response:

```
GET /mefApi/sonata/geographicSiteManagement/v7/geographicSite/00000000-0000-0030-0305-873500002014
```

```json
[
  {
    "id": "56787654-0000-0030-0305-000000001234",
    "href": "{{baseUrl}}/geographicSite/56787654-0000-0030-0305-000000001234",
    "description": "This is a description of the Site",
    "name": "Main DC",
    "siteType": "public",
    "@type": "GeographicSite",
    "place": [
      {
        "@type": "GeographicAddressRef",
        "id": "11221122-0000-0030-0305-463795472461"
      }
    ]
  }
]
```

[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`.
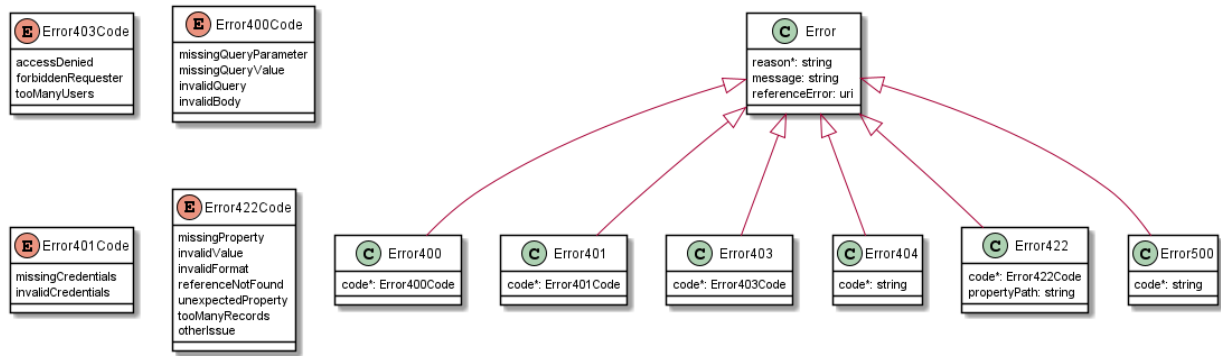
Figure 8. Data model types to represent an erroneous response

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
code*	string	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

7.1.1.3. Type Error401

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

Inherits from:

- Error

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

7.1.1.4. Type Error403

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

Inherits from:

- Error

Name	Type	Description
code*	string	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

7.1.1.5. 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	The following error code: - notFound: A current representation for the target resource not found

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. (https://tools.ietf.org/html/rfc4918#section-11.2)

Inherits from:

- Error

Name	Type	Description
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

Name	Type	Description

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

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
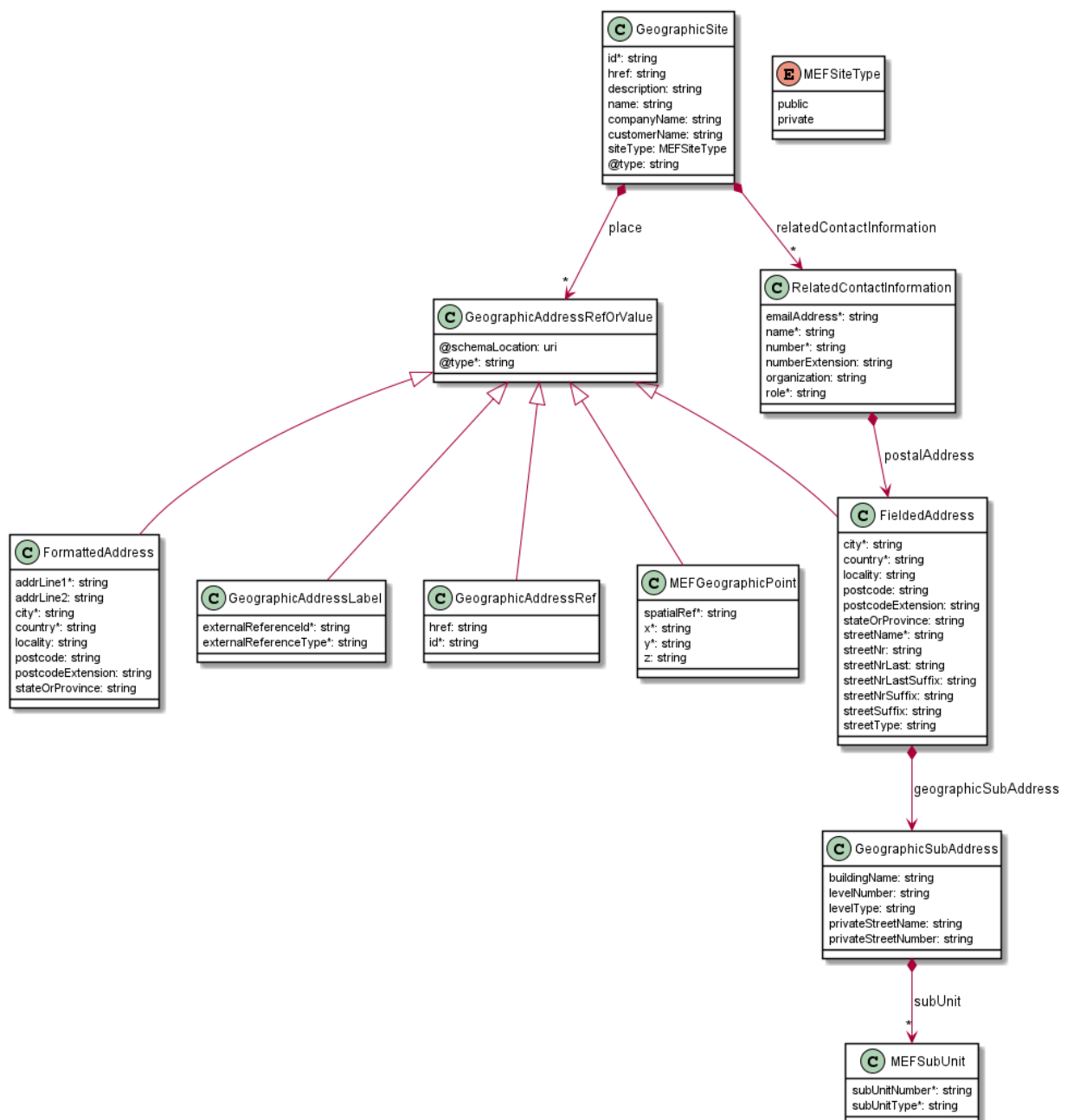
Figure 9. Site Retrieval Data Model

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.

Name	Type	Description	MEF 79
id*	string	Identifier of the Geographic Site unique within the Seller.	Service Site Identif
href	string	Unique reference of the Geographic Site unique within the Seller.	Not represented in 79
description	string	A textual description of the Geographic Site.	Service Site Descri
name	string	A user-friendly name for the place, such as [Paris Store], [London Store], [Main Home]	Service Site Name

Name	Type	Description	MEF 79
companyName	string	The name of the company that is the administrative authority (e.g. controls access) for this Geographic Site. (For example, the building owner.)	Service Site Company Name
customerName	string	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).	Service Site Customer Name

Name	Type	Description	MEF 79
place	GeographicAddressRefOrValue[]	A set of location descriptions, each of which describes where this GeographicSite 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.	Service Site Locati

Name	Type	Description	MEF 79
relatedContactInformation	RelatedContactInformation[]	An entity or organization that is involved to the geographical site, such as the Site Contact or Site Alternate Contact.	Service Site Contact (role=serviceSiteC
siteType	MEFSiteType	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	Service Site Restric Type

Name	Type	Description	MEF 79
		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	

Name	Type	Description	MEF 79
		obtain this information.	
@type	string	When sub-classing, this defines the sub-class entity name	Not represented in 79

7.2.1.2. enum MEFSiteType

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

Value	MEF 79	Description
public	PUBLIC	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.
private	PRIVATE	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.

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

Name	Type	Description	MEF 79

Name	Type	Description	MEF 79
@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 the Buyer and the Seller.	Not represented in MEF 79
@type*	string	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".	Not represented in MEF 79

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

Name	Type	Description	MEF 79
city*	string	The city that the address is in	City
country*	string	The country that the address is in	Country
geographicSubAddress	GeographicSubAddress	Additional fields used to specify an address, as detailed as possible.	Not represented in MEF 79
locality	string	An area of defined or undefined Present boundaries within a local authority or other legislatively defined area, usually rural or semirural in nature.	Locality

Name	Type	Description	MEF 79
postcode	string	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
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
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

Name	Type	Description	MEF 79
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

7.2.2.3. 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

Name	Type	Description	MEF 79
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	The 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	A descriptor for a postal delivery area used to speed and simplify the delivery of mail (also known as ZIP code)	Postal Code

Name	Type	Description	MEF 79
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

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

Name	Type	Description	MEF 79
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 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

7.2.2.5. Type GeographicSubAddress

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

Name	Type	Description	MEF 79
buildingName	string	The well-known name of a building that is located at this Geographic Address (e.g. where there is one Address for a campus).	Building Name

Name	Type	Description	MEF 79
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.	Sub Unit List

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

Name	Type	Description	MEF 79
externalReferenceId*	string	The unique reference to an Address as provided by the Administrative Authority.	Administrative Authority Address Label

Name	Type	Description	MEF 79
externalReferenceType*	string	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.	Administrative Authority

7.2.2.7. Type MEFSubUnit

Description: Allows for sub unit identification

Reference: MEF 79 (Sn 8.9.2)

Name	Type	Description	MEF 79
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

7.2.2.8. Type GeographicAddressRef

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

Inherits from:

- GeographicAddressRefOrValue

Name	Type	Description	MEF 79
href	string	Hyperlink to the referenced place 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 79

Name	Type	Description	MEF 79
id*	string	Identifier of the referenced Geographic Address. This identifier is assigned during a successful address validation request (Geographic Address Management API)	Fielded | Formatted | Geographic Address Label | Geographic Point Identifier

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)

Name	Type	Description	MEF 79
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
organization	string	The organization or company that the contact belongs to	Not represented in MEF 79
role*	string	The role of the particular contact in a given context.	Not represented in MEF 79
postalAddress	FieldedAddress	Identifies the postal address of the person or office to be contacted.	Not represented in MEF 79

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
- [REST] Chapter 5: Representational State Transfer (REST) Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures (Ph.D.).
- [RFC7231] RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, June 2014 https://tools.ietf.org/html/rfc7231
- [TMF674] TMF 674 TMF674 Geographic Site Management API User Guide v4.0.1