Working Draft
MEF W143 v0.1

LSO Allegro, LSO Interlude and LSO Legato Performance Monitoring Profiles, Jobs, Passive Statistics, Notifications and Collections API - Developer Guide

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

June 2023

Disclaimer

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

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

Table 1. Contributing Members

1. Abstract

This standard is intended to assist the implementation of the Application Programming Interfaces (APIs) for the Performance Monitoring functionality of the Service Orchestration Function at the LSO Allegro, LSO Interlude and LSO Legato Interface Reference Points (IRPs), for which requirements and use cases are defined in MEF W133.1 [MEF133.1]. The requirements and use cases are the same for all IRPs. This standard consists of this document and complementary API definitions for Performance Monitoring and Performance Notification.

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

MEF-LSO-Allegro-SDK

- `serviceApi/pm/performanceMonitoring.api.yaml`
- `serviceApi/pm/performanceNotification.api.yaml`

MEF-LSO-Interlude-SDK

- `serviceApi/pm/performanceMonitoring.api.yaml`
- `serviceApi/pm/performanceNotification.api.yaml`

MEF-LSO-Legato-SDK

- `serviceApi/pm/performanceMonitoring.api.yaml`
- `serviceApi/pm/performanceNotification.api.yaml`

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

2. Terminology and Abbreviations

This section aims to clarify the terminology used throughout this document. In many cases, the authoritative definitions of terms can be found in separate documents. To ensure accuracy and consistency, the third column of this document serves to provide the appropriate references from MEF or external sources that govern these definitions.

In addition, terms defined in the standards referenced below are included in this document by reference and are not repeated in the table below:

- MEF W133.1 *Allegro, Interlude and Legato Fault Management and Performance Monitoring BR&UC* February 2023 [MEF 55.1]
- MEF 55.1, *Lifecycle Service Orchestration (LSO): Reference Architecture and Framework* February 2021 [MEF 55.1]

Term	Definition	Source
API Endpoint	The endpoint of a communication channel (the complete URL of an API Resource) to which the HTTP-REST requests are addressed in order to operate on the *API Resource*.	rapidapi.com This document
API Resource	A REST Resource. In REST, the primary data representation is called Resource. In this document, *API Resource* is defined as a OAS *SchemaObject* with specified *API Endpoints*.	restfulapi.net This document
Notification	A notification is a representation of an event that is exchanged between interested parties. An event is a significant occurrence or change in system state that is important from the perspective of system administration.	MEF W133.1
On-Demand	Performance Monitoring Job actions that are initiated for a limited time to carry out the Performance Monitoring Job or measurements.	MEF W133.1
OpenAPI	The OpenAPI 3.0 Specification, formerly known as the Swagger specification is an API description format for REST APIs.	spec.openapis.org
Operation	An interaction between the BUS and SOF, potentially involving multiple back and forth transactions.	This document

Term	Definition	Source
Passive	Performance Monitoring Job action to support the collection and reporting of network and service statistics. The statistics collections include but are not limited to telemetry associated with an interface, (Net/Application) Flow, VLAN, bridg-ing/Ethernet, IP, TCP, UDP layers.	MEF W133.1
PM Metric	A metric that is measured or calculated as a part of Per-formance Monitoring.	MEF W105
Proactive	Performance Monitoring Job actions that are carried on continuously to permit timely reporting of fault and/or performance status.	MEF W133.1
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 API
SchemaObject	The construct that allows the definition of input and output data types. These types can represent object classes, as well as primitives and arrays specification.	spec.openapis.org

Table 2. Terminology

Term	Definition	Source
API	Application Programming Interface. In this document, API is used synonymously with REST API.	This document
BUS	Business Applications	MEF 55.1
CUS	Customer Application Coordinator	MEF 55.1
IRP	Interface Reference Point	This document
OAS	OpenAPI Specification	openapis.org
PM	Performance Monitoring	MEF W133.1
SOF	Service Orchestration Functionality	MEF 55.1

Table 3. Abbreviations

3. Compliance Levels

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

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words MAY or OPTIONAL) are labeled as **[Ox]** for optional.

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

4. Introduction

The Service Level Specification describes the performance objectives for the performance of conforming traffic (i.e., frames, packets) that flow over a VC (i.e., EVC, IPVC, etc.). For example, objectives in the SLS might be specified for frame or packet delay (latency). The performance objectives specified in the SLS often form part of a Service Level Agreement (SLA), which can also specify penalties for the SP or Operator providing the service if the objectives are not met. The Performance Monitoring API allows managing Performance Profiles, Performance Jobs and collect Performance Reports, as well as receive notifications related to these entities. This allows managing the performance objectives that are typically associated with an SLS.

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

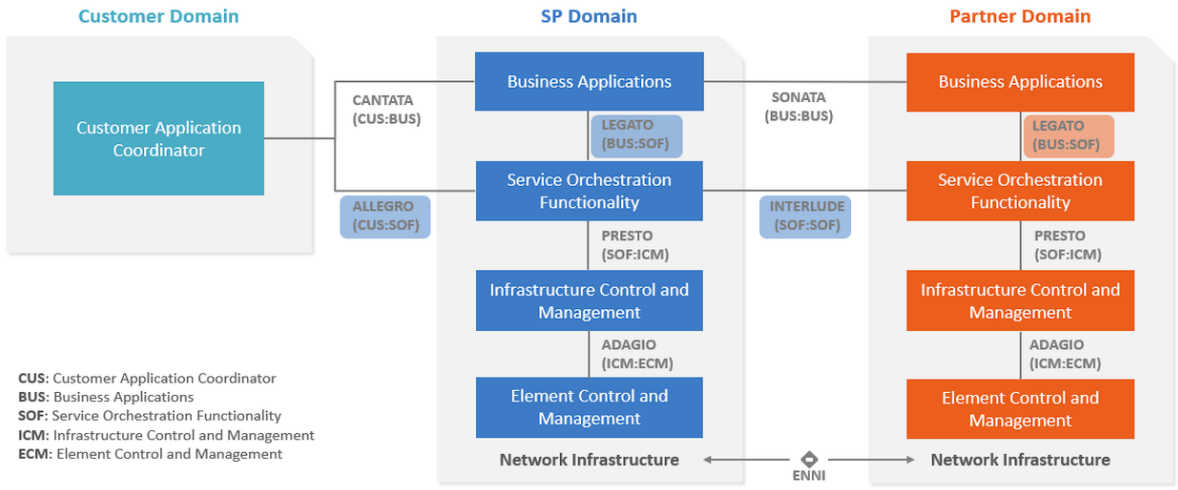


Figure 1. The LSO Reference Architecture

Note: The use cases and business requirements in this document assume a two-actor relationship based on the set of actors in the LSO architecture. The names of the relationship are specific to the Interface Reference Point. For both Allegro and Interlude there is a Buyer and Seller. For Allegro the Buyer is the Customer and the Seller is the Service Provider. In Interlude the Buyer is the Service Provider and the Seller is the Partner. In the case of the Legato IRP, given this is within a single Service Provider or Partner, the relationship is Client and Server, where the Business Application (BA) is the Client, and the Service Orchestration Functionality (SOF) is the Server. Considering this duality, actors in the document are referred to as Buyer/Client and Seller/Server.

4.1. Description

This standard is scoped to cover APIs for following Service Orchestration Functionalities:

- Performance Monitoring
  - Includes management of Performance Profiles, Performance Jobs and collecting Performance Reports
- Performance Notification
  - Includes Event Subscription/Hub and Listener notification functions

This document supports interactions over the Legato interface within a single operator as well as interaction with Partner Domain and Customer Domain through Interlude and Allegro interfaces respectively.

Business Applications (BUS), Customer Application Coordinator (CUS) and Service Orchestration Functionality (SOF) systems use the information contained within this document.

This standard is intended to support the design of API implementations that enable interoperable SOF operations (in scope of this standard) across the Allegro IRP, Interlude IRP and Legato IRP.

This standard is based on TMF Open API (v14.5.1) for Performance Management TMF 638.

The Performance Monitoring API allows the Buyer (CUS) or Client (BUS) to provision performance objectives in the Server (intra-operator SOF) or in the Seller (inter-operator SOF) and collect performance data from Server/Seller.

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. `PerformanceJob`).
- In UML diagrams the default cardinality of associations is `0..1`. Other cardinality markers are compliant with the UML standard.
- In the API details tables and UML diagrams required attributes are marked with a `*` next to their names.
- In UML sequence diagrams `{{variable}}` notation is used to indicate a variable to be substituted with a correct value.

4.3. Relation to Other Documents

This API implements the Performance Monitoring related requirements and use cases that are defined in MEF W133.1 [MEF133.1]. The API definition builds on *TMF 628 Performance Management API REST Specification R14.5.1*[TMF621]. Performance Monitoring Use Cases must support the use of MEF service performance specifications as payload.

4.4. Approach

As presented in Figure 2. the Allegro, Interlude and Legato API frameworks consist of three structural components:

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

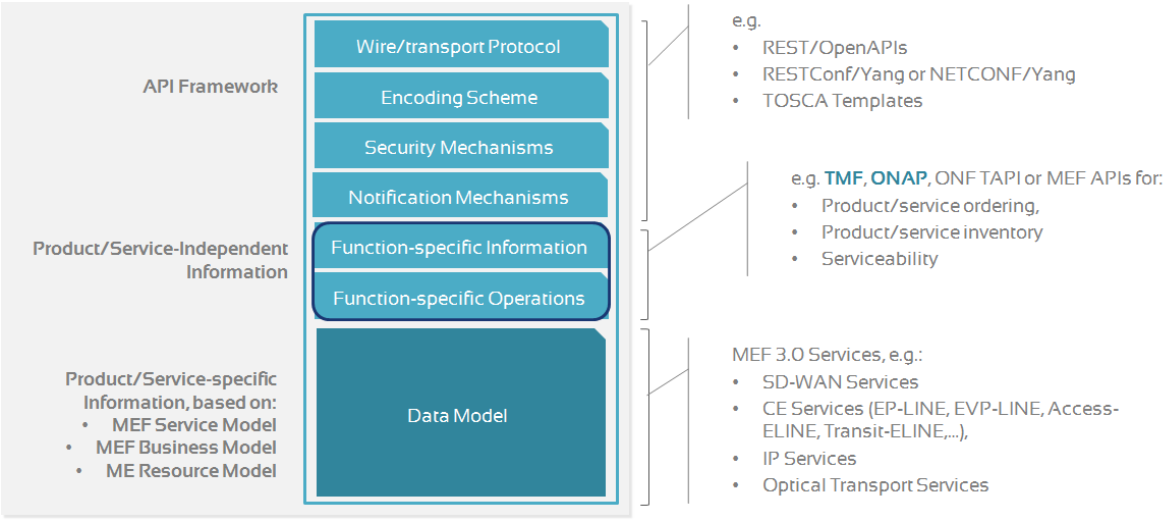


Figure 2. Allegro, Interlude and Legato API Structure

The essential concept behind the framework is to decouple the common structure, information, and operations from the specific service information content. Firstly, the Generic API Framework defines a set of design rules and patterns that are applied across all Allegro, Interlude and Legato APIs. Secondly, the service-independent information of the framework focuses on a model of a particular Allegro, Interlude or Legato functionality and is agnostic to any of the service specifications. For example, this standard is describing the Performance Monitoring model and operations that allow provisioning of the performance objectives of any service. Finally, the service-specific information part of the framework focuses on performance-related attributes and requirements for provisioning intra-provider or inter-provider performance objectives.

This Developer Guide is not defining MEF service performance specifications but can be used in combination with any performance specifications defined by or compliant with MEF. MEF Service Performance schemas are defined by:

- MEF 152: Carrier Ethernet Payload Schema/Guide for SOAM [MEF152]
- MEF 153: IP/IPVPN Schema/Guide for SOAM [MEF153]
- MEF 154: SD-WAN Schema/Guide for SOAM [MEF154]

Figure 3 presents the relations between the Performance Monitoring API entities and the service performance specification model. The `ServiceSpecificPayloadAttribute` serves as an extension point for configuring service-specific performance parameters. On the other hand, the `ResultPayload` acts as an extension point for capturing and representing the outcome of performance monitoring.

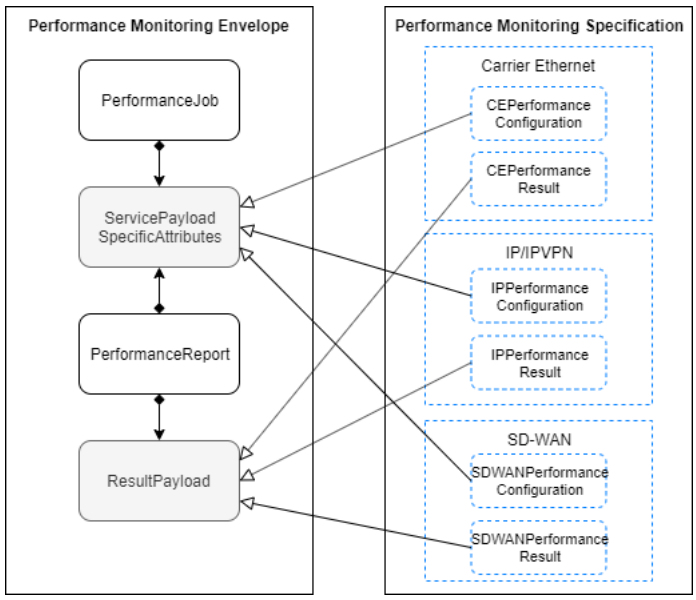


Figure 3. Performance specification for Allegro, Interlude, Legato

4.5. High-Level Flow

The Performance Monitoring API in essence allow the Buyer/Client to request SOF to provision measurement intervals, schedules, and performance objectives between one or more ordered pairs. An ordered pair is an association between two end points. Performance objectives are typically associated with an SLS but can be used for on-demand measurements in case SLS is not attached to service order. The Performance Notification API provides means to exchange information about significant changes in the system state between interested parties. Figure 4 presents a exemplary high-level flow of performance monitoring provisioning for SLS case.

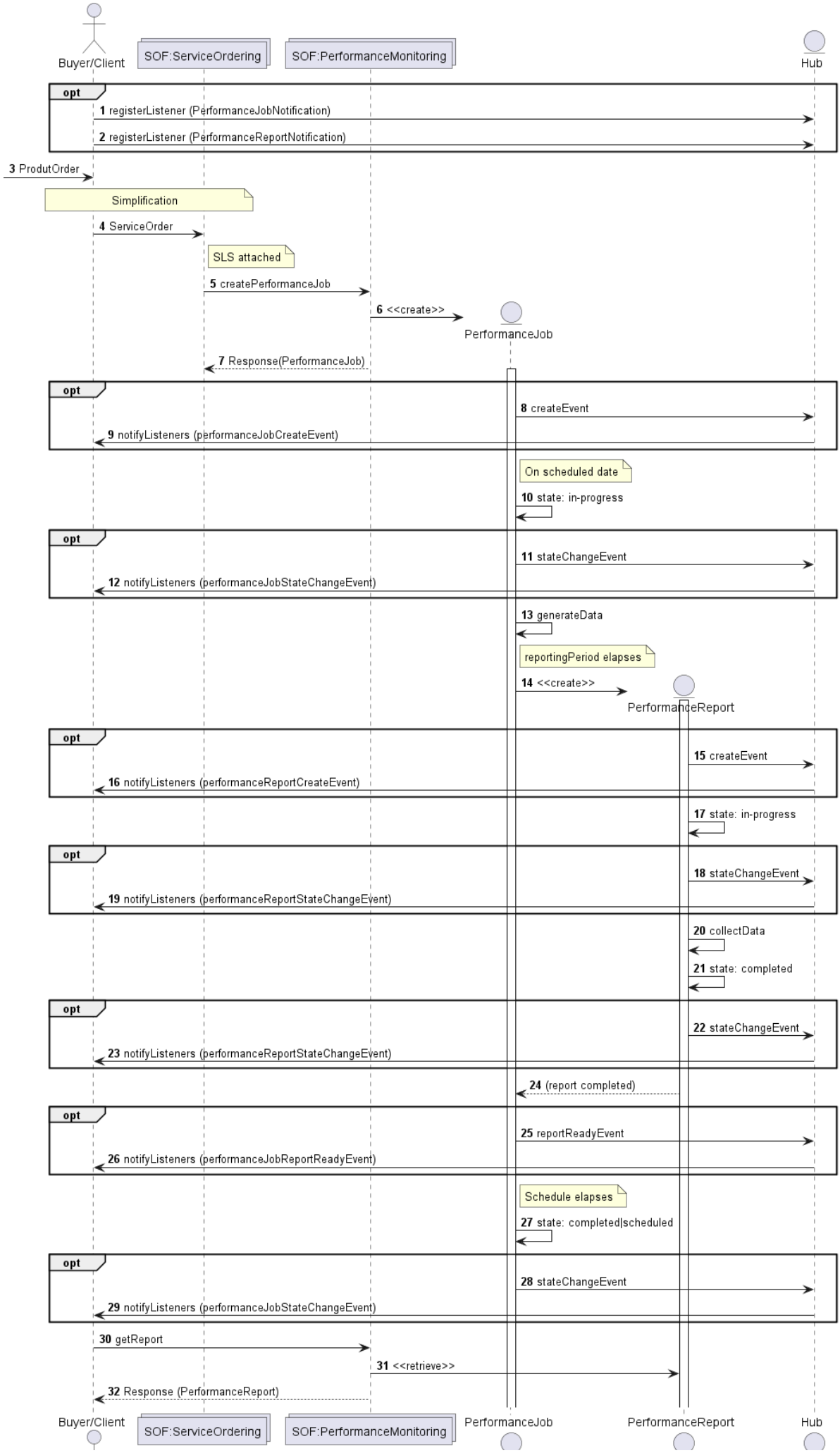

Buyer/Client
SOF:ServiceOrdering
SOF:PerformanceMonitoring
Hub
opt
1 registerListener (PerformanceJobNotification)
2 registerListener (PerformanceReportNotification)
3 ProdutOrder
Simplification
4 ServiceOrder
SLS attached
5 createPerformanceJob
6 <<create>>
PerformanceJob
7 Response(PerformanceJob)
opt
8 createEvent
9 notifyListeners (performanceJobCreateEvent)
On scheduled date
10 state: in-progress
opt
11 stateChangeEvent
12 notifyListeners (performanceJobStateChangeEvent)
13 generateData
reportingPeriod elapses
14 <<create>>
PerformanceReport
opt
15 createEvent
16 notifyListeners (performanceReportCreateEvent)
17 state: in-progress
opt
18 stateChangeEvent
19 notifyListeners (performanceReportStateChangeEvent)
20 collectData
21 state: completed
opt
22 stateChangeEvent
23 notifyListeners (performanceReportStateChangeEvent)
24 (report completed)
opt
25 reportReadyEvent
26 notifyListeners (performanceJobReportReadyEvent)
Schedule elapses
27 state: completed|scheduled
opt
28 stateChangeEvent
29 notifyListeners (performanceJobStateChangeEvent)
30 getReport
31 <<retrieve>>
32 Response (PerformanceReport)
Buyer/Client
SOF:ServiceOrdering
SOF:PerformanceMonitoring
PerformanceJob
PerformanceReport
Hub

Figure 4. High-Level Flow for SLS case

The following steps describe the high-level flow:

- (optional) The BUS system registers for notifications.
  Note1: Performance Notifications are optional and do not impact end-to-end flow
- As part of the ordering flow, the BUS system receives the product order (through Cantata or Sonata) which triggers the fulfillment processes in the BUS system.
- Service ordering flow in the diagram is simplified and is only supposed to show that in case of SLS attached to the service, a corresponding `PerformanceJob` is provisioned.
- During provisioning performance monitoring, the SOF internally uses the *Performance Monitoring API* to instantiate the 'PerformanceJob'
  Note2: *Process of identification of applicable service performance specification schema is out of scope for this standard.* ***Note3***: `PerformanceJob` can be provisioned using `PerformanceProfile`, but this is not depicted in the sequence diagram.
  - The SOF provisions performance monitoring by creating a `PerformanceJob` which contains the configuration of performance objectives and related subject (service or other type of entity).
  - `PerformanceJob` also carries a configuration including granularity, reporting period, schedule definition and output format.
  - The `PerformanceJob` is processed by the SOF as per the state transition rules described in 6.6.4.
  - (optional) The SOF reports the `PerformanceJob` state changes.
  - On scheduled date according to schedule definition, performance data generation is started.
  - When the configured reporting period elapses, a `PerformanceReport` entity is created to collect the performance data.
  - `PerformanceReport` is processed as per the state transition rules described in 6.22.4.
  - (optional) The SOF reports the `PerformanceJob` state change.
  - The BUS system can collect `PerformanceReport` through *Performance Monitoring API*
- The same *Performance Monitoring API* is used by the BUS to create **new** `PerformanceJob` instances, as well as update **existing** ones or trigger state transitions (e.g. delete **existing** `PerformanceJob` instance)

Figure 5 presents a high-level examplary flow of performance monitoring provisioning for non-SLS use case.


BUS
SOF:PerformanceMonitoring
Hub
opt
1 registerListener (PerformanceJobNotification)
2 registerListener (PerformanceReportNotification)
3 listPerformanceProfile
4 Response (List.PerformanceProfile)
5 createPerformanceJob
6 <<create>>
PerformanceJob
7 Response (PerformanceJob)
opt
8 createEvent
9 notifyListeners (performanceJobCreateEvent)
On scheduled date
10 state: in-progress
opt
11 stateChangeEvent
12 notifyListeners (performanceJobStateChangeEvent)
13 generateData
reportingPeriod elapses
14 <<create>>
PerformanceReport
opt
15 createEvent
16 notifyListeners (performanceReportCreateEvent)
17 state: in-progress
opt
18 stateChangeEvent
19 notifyListeners (performanceReportStateChangeEvent)
20 collectData
21 state: completed
opt
22 stateChangeEvent
23 notifyListeners (performanceReportStateChangeEvent)
24 (report completed)
opt
25 reportReadyEvent
26 notifyListeners (performanceJobReportReadyEvent)
Schedule elapses
27 state: completed|scheduled
opt
28 stateChangeEvent
29 notifyListeners (performanceJobStateChangeEvent)
30 getReport
31 <<retrieve>>

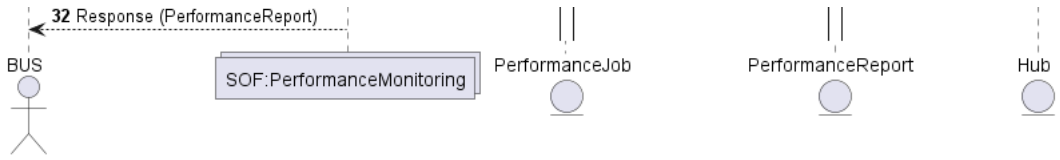


Figure 5. High-Level Flow for non-SLS case

Difference from the previous flow is due to the fact that in this case service does not define attached SLS. This requires the BUS to provision `PerformanceJob` as a step separate from service ordering.

- The BUS can provision performance monitoring by selecting a `PerformanceProfile`which is a template containing common configuration shared by multiple `PerformanceJob` entities.
- When querying `PerformanceProfile` instances the BUS system uses the *Performance Monitoring API*.
- Rest of the flow is same as described previously.

Figure 6 presents relations between entities that are managed through *Performance Monitoring API*. The diagram is simplified and does not contain all types of objects.

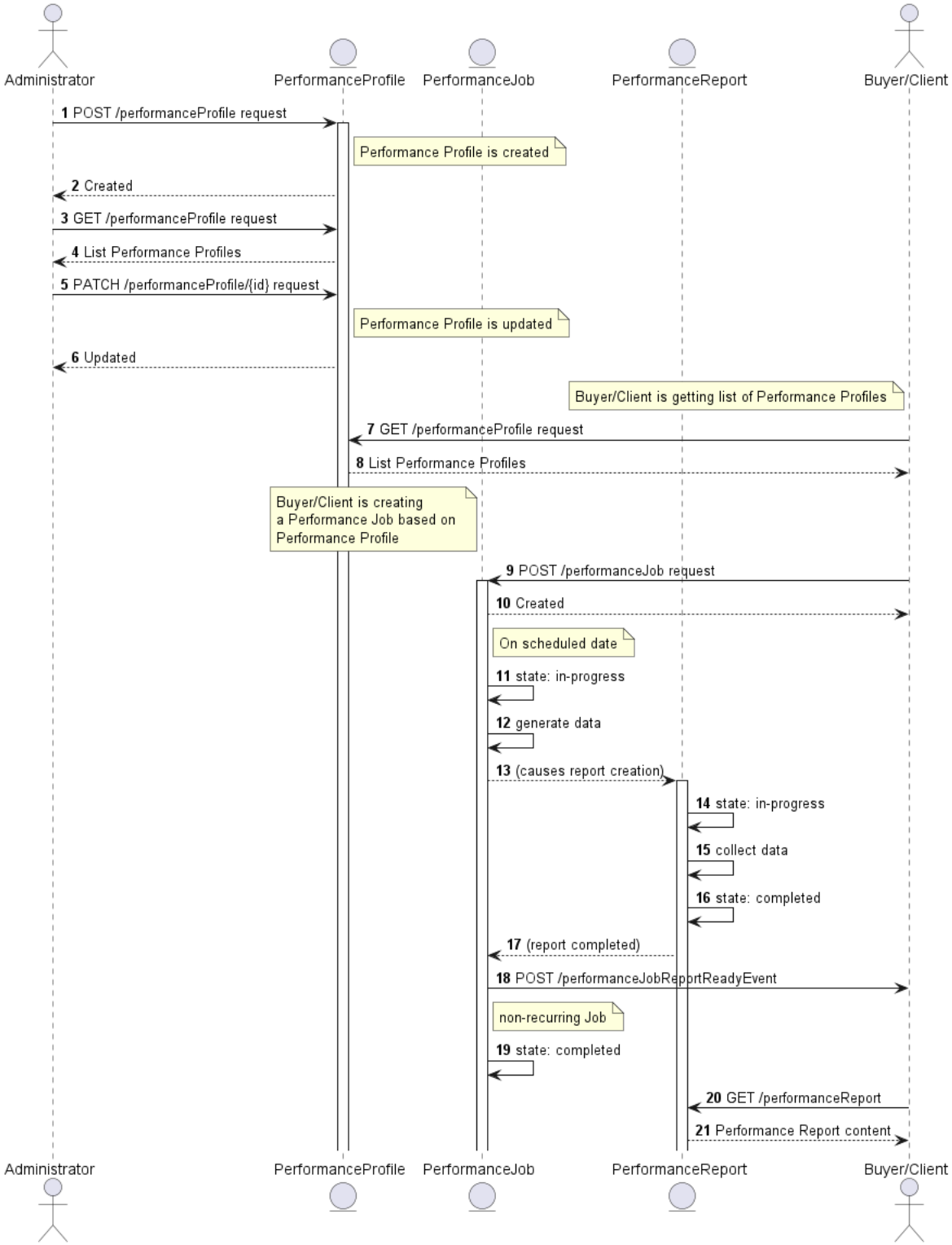


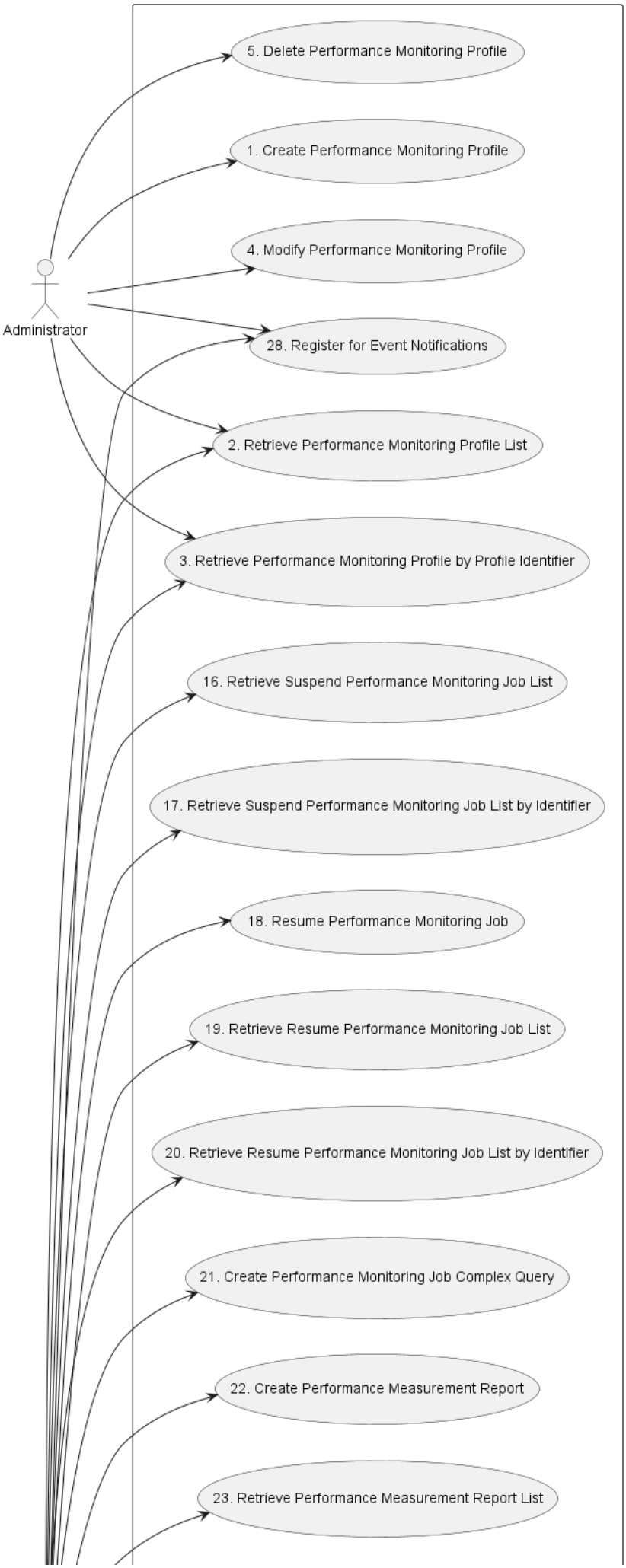
Figure 6. Flow between API endpoints

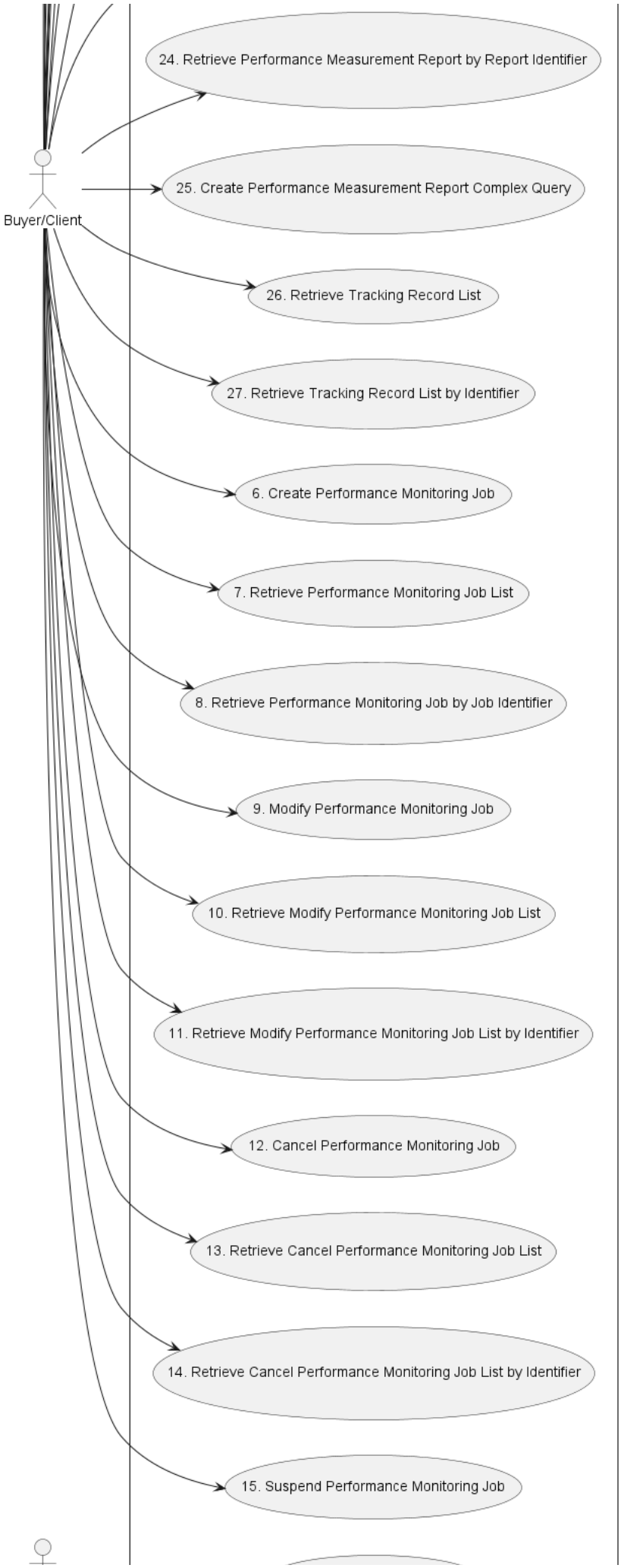
5. API Description

This section presents the API structure and design patterns. It starts with the high-level use cases diagram. Then it describes the REST endpoints with use case mapping. Next, it gives an explanation of the design pattern that is used to combine service-agnostic and service-specific parts of API payloads. Finally, payload validation and API security aspects are discussed.

5.1. High-level use cases

Figure 7 presents a high-level use case diagram. It aims to help understand the endpoint mapping. Use cases are described extensively in chapter 6.


Administrator
5. Delete Performance Monitoring Profile
1. Create Performance Monitoring Profile
4. Modify Performance Monitoring Profile
28. Register for Event Notifications
2. Retrieve Performance Monitoring Profile List
3. Retrieve Performance Monitoring Profile by Profile Identifier
16. Retrieve Suspend Performance Monitoring Job List
17. Retrieve Suspend Performance Monitoring Job List by Identifier
18. Resume Performance Monitoring Job
19. Retrieve Resume Performance Monitoring Job List
20. Retrieve Resume Performance Monitoring Job List by Identifier
21. Create Performance Monitoring Job Complex Query
22. Create Performance Measurement Report
23. Retrieve Performance Measurement Report List


24. Retrieve Performance Measurement Report by Report Identifier
25. Create Performance Measurement Report Complex Query
Buyer/Client
26. Retrieve Tracking Record List
27. Retrieve Tracking Record List by Identifier
6. Create Performance Monitoring Job
7. Retrieve Performance Monitoring Job List
8. Retrieve Performance Monitoring Job by Job Identifier
9. Modify Performance Monitoring Job
10. Retrieve Modify Performance Monitoring Job List
11. Retrieve Modify Performance Monitoring Job List by Identifier
12. Cancel Performance Monitoring Job
13. Retrieve Cancel Performance Monitoring Job List
14. Retrieve Cancel Performance Monitoring Job List by Identifier
15. Suspend Performance Monitoring Job

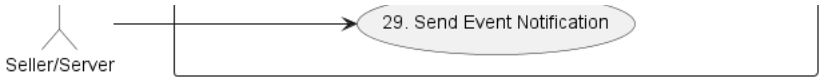


Figure 7. Use cases

5.2. API Endpoint and Operation Description

5.2.1. Seller/Server (SOF) side Performance Monitoring API Endpoints

Base URL for Allegro: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/allegro/performanceMonitoring/v1/`

Base URL for Interlude: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/interlude/performanceMonitoring/v1/`

Base URL for Legato: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/legato/performanceMonitoring/v1/`

The following API endpoints are implemented by the Seller/Server (SOF) and allow the Buyer/Client (SOF/CUS/BUS) to create, retrieve and modify `PerformanceJob`, `PerformanceProfile` and `PerformanceReport` instances. The endpoints and corresponding data model are defined in `serviceApi/pm/performanceMonitoring.api.yaml`.

API Endpoint	Description	MEF W133.1 Use Case Mapping
POST /performanceProfile	A request initiated by the Administrator to create a Performance Monitoring Profile in the Seller/Server system.	10
GET /performanceProfile	The Administrator or Buyer/Client requests a list of Performance Monitoring Profiles based on a set of filter criteria.	11
GET /performanceProfile/{{id}}	The Administrator or Buyer/Client requests detailed information about a single Performance Monitoring Profile.	12
POST /performanceJob	A request initiated by the Buyer/Client to create a Performance Monitoring Job in the Seller/Server system.	18,30
GET /performanceJob	The Buyer/Client requests a list of Performance Monitoring Jobs based on a set of filter criteria.	23

API Endpoint	Description	MEF W133.1 Use Case Mapping
GET /performanceJob/{{id}}	The Buyer/Client requests detailed information about a single Performance Monitoring Job.	24
POST /modifyPerformanceJob	A request initiated by the Buyer/Client to modify a Performance Monitoring Job in the Seller/Server system.	19,31
GET /modifyPerformanceJob	The Buyer/Client requests a list of Modify Performance Monitoring Jobs based on a set of filter criteria.	19,31
GET /modifyPerformanceJob/{{id}}	The Buyer/Client requests detailed information about a single Modify Performance Monitoring Job.	19,31
POST /cancelPerformanceJob	A request initiated by the Buyer/Client to cancel a Performance Monitoring Job in the Seller/Server system.	20,32
GET /cancelPerformanceJob	The Buyer/Client requests a list of Cancel Performance Monitoring Jobs based on a set of filter criteria.	20,32
GET /cancelPerformanceJob/{{id}}	The Buyer/Client requests detailed information about a single Cancel Performance Monitoring Job.	20,32
POST /suspendPerformanceJob	A request initiated by the Buyer/Client to suspend a Performance Monitoring Job in the Seller/Server system.	21
GET /suspendPerformanceJob	The Buyer/Client requests a list of Suspend Performance Monitoring Jobs based on a set of filter criteria.	21
GET /suspendPerformanceJob/{{id}}	The Buyer/Client requests detailed information about a single Suspend Performance Monitoring Job.	21
POST /resumePerformanceJob	A request initiated by the Buyer/Client to resume a Performance Monitoring Job in the Seller/Server system.	22

API Endpoint	Description	MEF W133.1 Use Case Mapping
GET /resumePerformanceJob	The Buyer/Client requests a list of Resume Performance Monitoring Jobs based on a set of filter criteria.	22
GET /resumePerformanceJob/{{id}}	The Buyer/Client requests detailed information about a single Resume Performance Monitoring Job.	22
POST /performanceJobComplexQuery	A request initiated by the Buyer/Client to create a Performance Monitoring Job Complex Query in the Seller/Server system.	23
POST /performanceReport	A request initiated by the Buyer/Client to create an ad-hoc Performance Measurement Report in the Seller/Server system.	29,34
GET /performanceReport	The Buyer/Client requests a list of Performance Measurement Reports based on a set of filter criteria.	28
GET /performanceReport/{{id}}	The Buyer/Client requests detailed information about a single Performance Measurement Report.	29,34
POST /performanceReportComplexQuery	A request initiated by the Buyer/Client to create a Performance Measurement Report Complex Query in the Seller/Server system.	28
GET /trackingRecord	The Buyer/Client requests a list of Tracking Records based on a set of filter criteria.	
GET /trackingRecord/{{id}}	The Buyer/Client requests detailed information about a single Tracking Record.	

Table 4. Seller/Server (SOF) Performance Monitoring mandatory API endpoints

[R1] Seller/Server (SOF) **MUST** support all API endpoints listed in Table 4.

API endpoints listed in table 5 are optional and may be exposed by the SOF.

API Endpoint	Description	MEF W133.1 Use Case Mapping
PATCH /performanceProfile/{{id}}	A request initiated by the Administrator to modify a Performance Monitoring Profile in the Seller/Server system based on a Performance Monitoring Profile Identifier.	13
DELETE /performanceProfile/{{id}}	The Administrator requests deletion of Performance Monitoring Profile by specifying Performance Monitoring Profile Identifier.	14
POST /hub	The Buyer/Client or Administrator requests to subscribe to Performance Monitoring Profile, Performance Monitoring Job and/or Performance Measurement Report Notifications.	15,25
GET /hub/{{id}}	The Buyer/Client or Administrator retrieves a specific `EventSubscription` from the SOF, that matches the *`id`* value provided as *`path`* parameter.	15,25
DELETE /hub/{{id}}	The Buyer/Client or Administrator requests to unsubscribe from Performance Monitoring Profile, Performance Monitoring Job and/or Performance Measurement Report Notifications.	17,26

Table 5. Seller/Server (SOF) Performance Monitoring optional API endpoints

[O1] The implementation **MAY** support API endpoints listed in Table 5. [W133 O4, O6, O8]

5.2.2. Buyer/Client (CUS, BUS, SOF) side Performance Monitoring API Endpoints

Base URL for Allegro: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/allegro/performanceNotification/v1/`

Base URL for Interlude: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/interlude/performanceNotification/v1/`

Base URL for Legato: `https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/legato/performanceNotification/v1/`

The following API Endpoints are used by SOF to post notifications to registered CUS, BUS or SOF listeners. The endpoints and corresponding data model are defined in

API Endpoint	Description	MEF W133.1 Use Case Mapping
POST /listener/performanceJobCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceJob` instance creation.	16,27
POST /listener/performanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceJob` instance state change.	16,27
POST /listener/performanceJobAttributeValueChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceJob` instance attribute value change.	16,27
POST /listener/performanceJobReportReadyEvent	A request initiated by the Seller/Server to notify Buyer/Client that `PerformanceReport` was generated for `PerformanceJob` instance.	16,27
POST /listener/performanceJobReportPreparationErrorEvent	A request initiated by the Seller/Server to notify Buyer/Client that `PerformanceReport` was not generated for `PerformanceJob` instance due to an error.	16,27

API Endpoint	Description	MEF W133.1 Use Case Mapping
POST /listener/cancelPerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `cancelPerformanceJob` instance state change.	16,27
POST /listener/modifyPerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `modifyPerformanceJob` instance state change.	16,27
POST /listener/resumePerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `resumePerformanceJob` instance state change.	16,27
POST /listener/suspendPerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `suspendPerformanceJob` instance state change.	16,27
POST /listener/performanceProfileCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance creation.	16,27

API Endpoint	Description	MEF W133.1 Use Case Mapping
POST /listener/performanceProfileStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance state change.	16,27
POST /listener/performanceProfileAttributeValueChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance attribute value change.	16,27
POST /listener/performanceProfileDeleteEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance deletion.	16,27
POST /listener/performanceReportCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceReport` instance creation.	16,27
POST /listener/performanceReportStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceReport` instance state change.	16,27

Table 6. Buyer/Client (CUS, BUS, SOF) Performance Monitoring API endpoints

[O2] The Buyer/Client (CUS, BUS, SOF) **MAY** support API endpoints listed in Table 6.

[O3] The Buyer/Client (CUS, BUS, SOF) **MAY** register to receive performance monitoring notifications.

[R2] The Seller/Server **MUST** support sending notification to API endpoints listed in Table 6 to registered Buyer/Client. [MEF133.1 R74]

5.3. Integration of Service Monitoring Specification into Performance Monitoring API

Performance Monitoring API discussed in this document is a generic envelope that allows for lifecycle management of relevant performance monitoring objects. The API itself does not provide explicit definitions for configuring performance monitoring or prescribing the structure of output data. However, it offers flexible extensibility to accommodate the configuration of service-specific performance objectives and results. This allows for customization and adaptation to various monitoring requirements and desired data formats. This monitoring configuration and result schemas are defined using JsonSchema (draft 7) format JSON Schema draft 7 and can be integrated into the `PerformanceJob` and `PerformanceReport` using the TMF extension pattern.

The extension hosting types in the API data model are:

- `ServicePayloadSpecificAttributes` - this type is extended with Service monitoring configuration schema
- `ResultPayload` - this type is extended with Service monitoring result schema The `@type` attribute of those extension hosting types must be set to a value that uniquely identifies the service monitoring configuration. A unique identifier for MEF standard service schemas is in URN format and is assigned by MEF. This identifier is provided as root schema `$id`. Use of non-MEF standard service monitoring configuration is allowed. In such a case the schema identifier must be agreed upon between the Buyer/Client and the Seller/Server.

The example below shows a header of a schema, which describes the IP service performance monitoring configuration, where `"$id": urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all` is the above-mentioned URN:

```
'$schema': http://json-schema.org/draft-07/schema#
'$id': urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all
title: MEF LSO Legato - IP Performance Monitoring Configuration
```

Monitoring configuration payload is introduced in multiple API entities through a `servicePayloadSpecificAttributes` attribute of type `ServicePayloadSpecificAttributes` which is used as an extension point for service-specific attributes.

In terms of monitoring results, appropriate payload is introduced via `ReportContent`. This entity has a `measurementDataPoints` array of items of type `ResultPayload` which is used as an extension point for service-specific attributes.

Implementations might choose to integrate selected performance monitoring specifications to data model during development. In such a case an integrated data model is built, and monitoring specifications are in an inheritance relationship accordingly with either `ServicePayloadSpecificAttributes` or `ResultPayload` as described in the OAS specification. This pattern is called **Static Binding**. The snippets below present an example of a static binding of the envelope API with exemplary MEF monitoring specifications, for both extension points.

```
ServicePayloadSpecificAttributes:
  type: object
  description: ServicePayloadSpecificAttributes is used as an extension point
    for MEF specific service performance monitoring configuration. It includes
    definition of service/entity and applicable performance monitoring objectives.
    The `@type` attribute is used as a discriminator
  discriminator:
    mapping:
      urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all:
'#/components/schemas/IpPerformanceMonitoringConfiguration'
    propertyName: '@type'
  properties:
    '@type':
      type: string
      description:
        The name that uniquely identifies type of performance monitoring configuration
        that specifies PM objectives. In case of MEF services this is the URN
        provided in performance monitoring configuration specification.
        The named type must be a subclass of ServicePayloadSpecificAttributes.
```

```
IpPerformanceMonitoringConfiguration:
  allOf:
    - $ref: '#/components/schemas/ServicePayloadSpecificAttributes'
    - type: object
      description: IP Performance Monitoring Configuration Schema.
```

```
ResultPayload:
  type: object
  description:
    ResultPayload is used as an extension point for MEF specific service
    performance monitoring results. The `@type` attribute is used as a discriminator
  discriminator:
    mapping:
      urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all:
'#/components/schemas/IpPerformanceMonitoringResults'
    propertyName: '@type'
  properties:
    '@type':
      type: string
      description:
        The name that uniquely identifies type of performance monitoring
        results that are returned by the Performance Report. In case of MEF services this
        is the URN provided in performance monitoring results specification.
        The named type must be a subclass of ResultPayload.
```

```
IpPerformanceMonitoringResults:
  allOf:
    - $ref: '#/components/schemas/ResultPayload'
    - type: object
      description: IP Performance Monitoring Results Schema.
```

Alternatively, implementations might choose not to build an integrated model and choose a different mechanism allowing runtime validation of service-specific fragments of the payload. The system can validate a given monitoring configuration against a new schema without redeployment. This pattern is called **Dynamic Binding.**

Regardless of chosen implementation pattern, the HTTP payload is exactly the same. Both implementation approaches must conform to the requirements specified below.

[R3] `ServicePayloadSpecificAttributes` and `ResultPayload` type are extension points that **MUST** be used to integrate service performance properties into a request/response payload.

[R4] The `@type` property of `ServicePayloadSpecificAttributes` and `ResultPayload` **MUST** be used to specify the type of the extending entity.

[R5] Attributes specified in the payload must conform to the performance definition specified in the `@type` property.

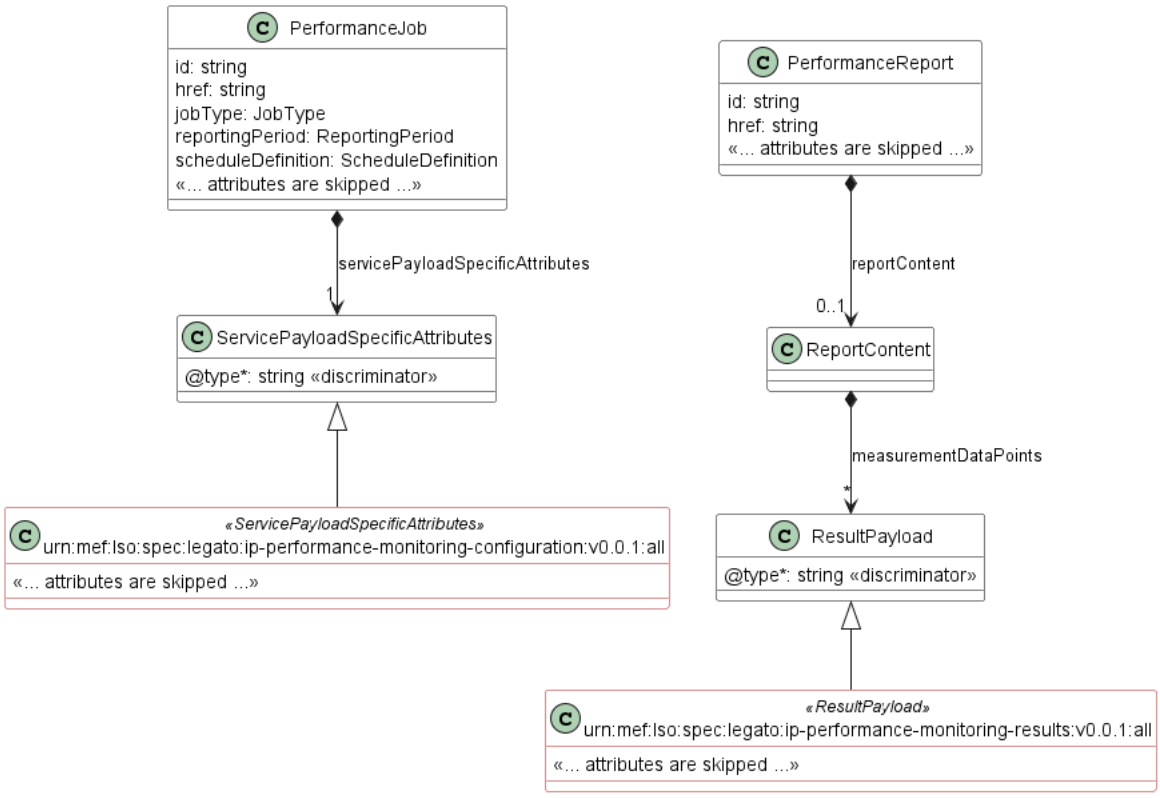


Figure 8. The Extension Pattern with Sample Service-Specific Extension

Figure 8 presents two MEF performance monitoring schemas that represent configuration and result classes for IP services. When these schemas are used, the `@type` of `ServicePayloadSpecificAttributes` takes `"urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all"` value to indicate which performance specification should be used to interpret a set of service-specific attributes included in the payload. Similarly, for `ResultPayload`, the `@type` attribute takes `"urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all"` value which indicates how the resulting performance collection should be interpreted.

5.4. Model structure and validation

The structure of the payloads exchanged via Allegro, Interlude and Legato Performance Monitoring API endpoints is defined using:

- OpenAPI version 3.0 for the service-agnostic part of the payload
- JsonSchema (draft 7) for the service-specific part of the payload

[R6] Implementations **MUST** use payloads that conform to these definitions.

5.5. Security Considerations

Although the Legato IRP is internal to a Service Provider/Operator business boundary, it is expected that some minimal security mechanisms are in place for any communication over this IRP. There must also be authorization mechanisms in place to control what a particular Buyer/Client or SOF is allowed to do and what information may be obtained. For Allegro and Interlude IRPs, security should follow rules for external communication. The definition of the exact security mechanism and configuration is outside the scope of this document. The LSO Security mechanisms are defined by MEF 128 *LSO API Security Profiles* [MEF128].

6. API Interactions and Flows

This section provides a detailed insight into the API functionality, use cases, and flows. It starts with Table 7 presenting a list and short description of all business use cases then presents the variants of end-to-end interaction flows, and in the following subchapters describes the API usage flow and examples for each of the use cases.

Use Case #	Use Case Name	Use Case Description
1	Create Performance Monitoring Profile	A request initiated by the Administrator to create a Performance Monitoring Profile in the Seller/Server system.
2	Retrieve Performance Monitoring Profile List	The Administrator or Buyer/Client requests a list of Performance Monitoring Profiles based on a set of filter criteria. The Seller/Server returns a summarized list of PM Profiles.
3	Retrieve Performance Monitoring Profile by Profile Identifier	The Administrator or Buyer/Client requests detailed information about a single Performance Monitoring Profile based on the Performance Monitoring Profile Identifier.
4	Modify Performance Monitoring Profile	A request initiated by the Administrator to modify a Performance Monitoring Profile in the Seller/Server system based on a Performance Monitoring Profile Identifier.
5	Delete Performance Monitoring Profile	The Administrator requests deletion of Performance Monitoring Profile by specifying Performance Monitoring Profile Identifier.
6	Create Performance Monitoring Job	A request initiated by the Buyer/Client to create a Performance Monitoring Job in the Seller/Server system to indicate performance monitoring objectives.
7	Retrieve Performance Monitoring Job List	The Buyer/Client requests a list of Performance Monitoring Jobs based on a set of filter criteria. The Seller/Server returns a summarized list of PM Jobs.
8	Retrieve Performance Monitoring Job by Job Identifier	The Buyer/Client requests detailed information about a single Performance Monitoring Job based on the Performance Monitoring Job Identifier.

Use Case #	Use Case Name	Use Case Description
9	Modify Performance Monitoring Job	A request initiated by the Buyer/Client to modify a Performance Monitoring Job in the Seller/Server system.
10	Retrieve Modify Performance Monitoring Job List	The Buyer/Client requests a list of Modify Performance Monitoring Jobs based on a set of filter criteria.
11	Retrieve Modify Performance Monitoring Job List by Identifier	The Buyer/Client requests detailed information about a single Modify Performance Monitoring Job based on the Modify Performance Monitoring Job Identifier.
12	Cancel Performance Monitoring Job	A request initiated by the Buyer/Client to cancel a Performance Monitoring Job in the Seller/Server system.
13	Retrieve Cancel Performance Monitoring Job List	The Buyer/Client requests a list of Cancel Performance Monitoring Jobs based on a set of filter criteria.
14	Retrieve Cancel Performance Monitoring Job List by Identifier	The Buyer/Client requests detailed information about a single Cancel Performance Monitoring Job based on the Cancel Performance Monitoring Job Identifier.
15	Suspend Performance Monitoring Job	A request initiated by the Buyer/Client to suspend a Performance Monitoring Job in the Seller/Server system.
16	Retrieve Suspend Performance Monitoring Job List	The Buyer/Client requests a list of Suspend Performance Monitoring Jobs based on a set of filter criteria.
17	Retrieve Suspend Performance Monitoring Job List by Identifier	The Buyer/Client requests detailed information about a single Suspend Performance Monitoring Job based on the Suspend Performance Monitoring Job Identifier.
18	Resume Performance Monitoring Job	A request initiated by the Buyer/Client to resume a Performance Monitoring Job in the Seller/Server system.
19	Retrieve Resume Performance Monitoring Job List	The Buyer/Client requests a list of Resume Performance Monitoring Jobs based on a set of filter criteria.

Use Case #	Use Case Name	Use Case Description
20	Retrieve Resume Performance Monitoring Job List by Identifier	The Buyer/Client requests detailed information about a single Resume Performance Monitoring Job based on the Resume Performance Monitoring Job Identifier.
21	Create Performance Monitoring Job Complex Query	A request initiated by the Buyer/Client to create a Performance Monitoring Job Complex Query in the Seller/Server system.
22	Create Performance Measurement Report	A request initiated by the Buyer/Client to create an ad-hoc Performance Measurement Report based on existing performance data in the Seller/Server system.
23	Retrieve Performance Measurement Report List	The Buyer/Client requests a list of Performance Measurement Reports based on a set of filter criteria. The Seller/Server returns a summarized list of PM Profiles.
24	Retrieve Performance Measurement Report by Report Identifier	The Buyer/Client requests detailed information about a single Performance Measurement Report based on the Performance Measurement Report Identifier.
25	Create Performance Measurement Report Complex Query	A request initiated by the Buyer/Client to create a Performance Measurement Report Complex Query in the Seller/Server system.
26	Retrieve Tracking Record List	The Buyer/Client requests a list of Tracking Records based on a set of filter criteria. The Seller/Server returns a summarized list of Tracking Records.
27	Retrieve Tracking Record List by Identifier	The Buyer/Client requests detailed information about a single Tracking Record based on the Tracking Record Identifier.
28	Register for Event Notifications	The Buyer/Client or Administrator requests to subscribe to Performance Monitoring Profile, Performance Monitoring Job, and/or Performance Measurement Report Notifications.
29	Send Event Notification	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceJob` instance creation.

Table 7. Use cases description

6.1. Use case 1: Create Performance Monitoring Profile

Performance Monitoring Profile is a template which is used to simplify the Performance Monitoring Job provisioning. Common attributes can be defined in the Performance Monitoring Profile which can be centralized and leveraged across multiple Performance Jobs.

6.1.1. Interaction flow

The flow of this use case is described in Figure 9.

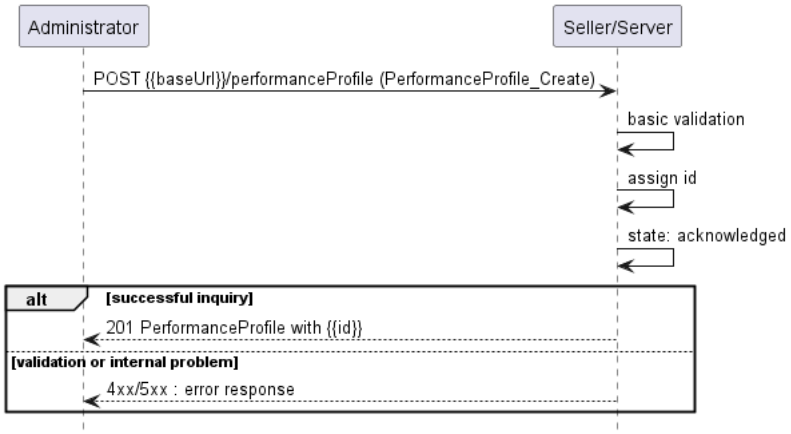


Figure 9. Use Case 1 - Performance Monitoring Profile create request flow

The only actor allowed executing create Performance Monitoring Profile is Administrator. Administrator is a special role that represent additional access rights not available to standard Buyer/Client roles.

[R7] - Only Administrator role **MUST** have access rights to create Performance Monitoring Profile.

The Administrator sends a request with a `PerformanceProfile_Create` type in the body. The SOF performs request validation, assigns an `id`, and returns `PerformanceProfile` type in the response body, with a `state` set to `acknowledged`. From this point, the Performance Profile will undergo further validations before it is ready to be used, and its state is set to `active`. The Administrator can track the progress of the process either by subscribing for notifications or by periodically polling the `PerformanceProfile`. The two patterns are presented in the following diagrams.

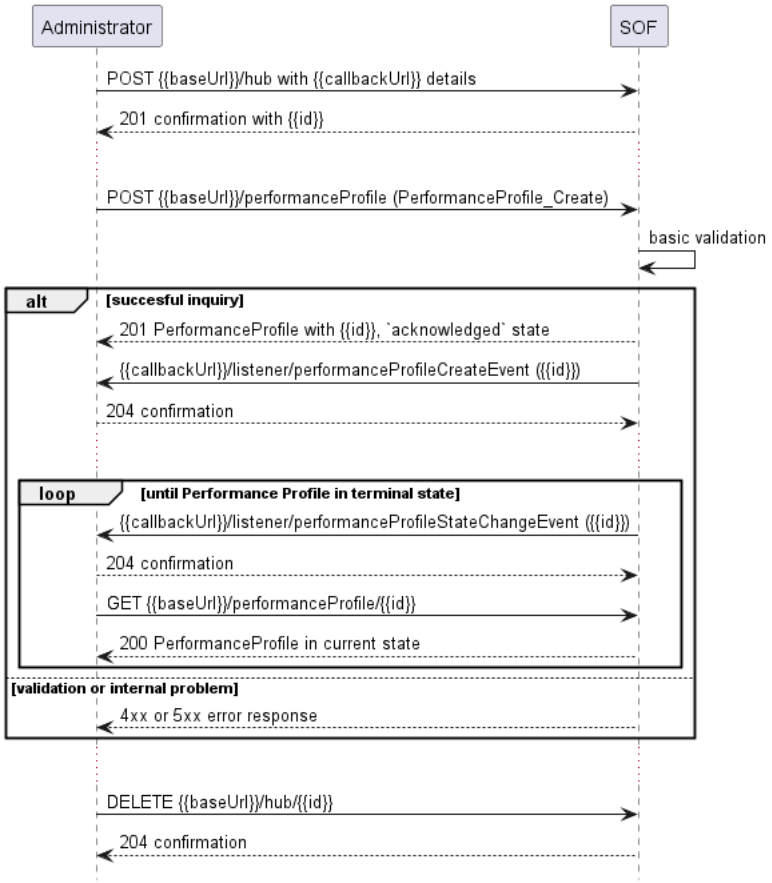


Figure 10. Performance Profile progress tracking - Notifications

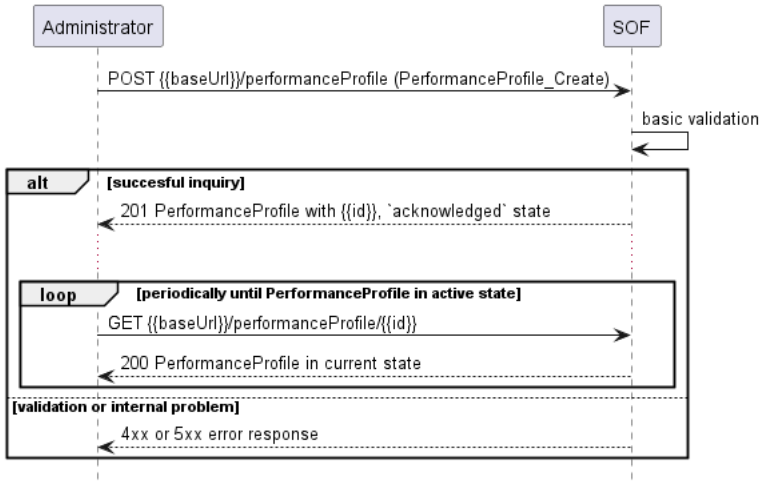


Figure 11. Performance Profile progress tracking - Polling

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

6.1.2. Create Performance Profile Request

Figure 12 presents the most important part of the data model used during the Create Performance Profile request (`POST /performanceProfile`) and response. The model of the request message - `PerformanceProfile_Create` is a subset of the `PerformanceProfile` model and contains only

attributes that can (or must) be set by the Buyer/Client. The Seller/Server then enriches the entity in the response with additional information.

Note: `PerformanceProfile_Create` is an entity used by the Buyer/Client to make a request. `PerformanceProfile` is an entity used by the Seller/Server to provide a response. The request entity have a subset of attributes of the response entity. Thus for visibility of these shared attributes `PerformanceProfile_Common` has been introduced. Though, this class is not to be used directly in the exchange.

A `PerformanceProfile_Create` defines details of execution of the `PerformanceJob` that will use the profile as a template. This includes parameters that can be shared by multiple Performance Monitoring Jobs.

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

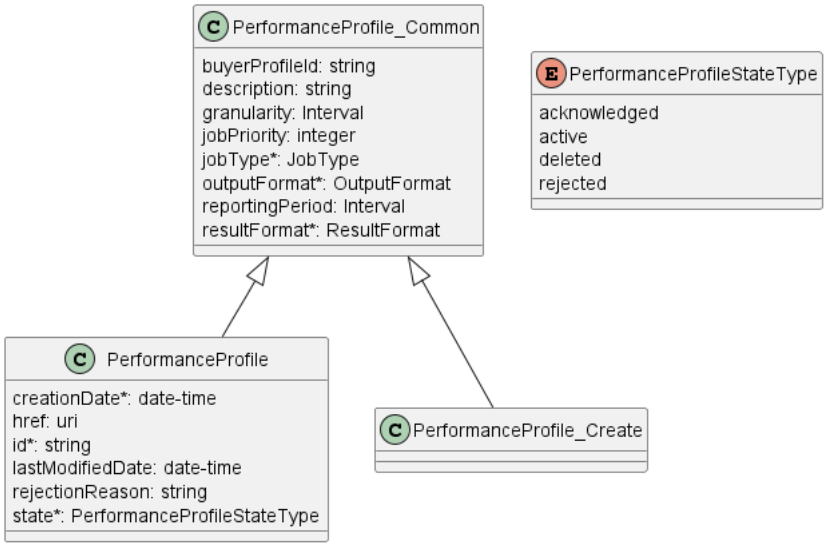


Figure 12. Performance Profile Key Entities

To send a request the Buyer/Client uses the `createPerformanceProfile` operation from the API. The snippet below presents an example of Create Performance Profile request:

`Performance Profile` **Create Request**

```
{
  "buyerProfileId": "a5240110-0945-11ee-be56-0242ac120002",
  "description": "Exemplary Create Performance Profile request",
  "granularity": "10 second",
  "jobPriority": 5,
  "jobType": "proactive",
  "outputFormat": "json",
  "reportingPeriod": "1 hour",
  "resultFormat": "payload"
}
```

[R8] The Administrator's Create Performance Profile **MUST** support the following attributes: [MEF133.1 R43]

- PM Profile ID
- Buyer PM Profile ID
- PM Job Type
- Granularity
- Reporting Period
- Schedule Definition

[O4] The Administrator's Create Performance Profile **MAY** contain the following attributes: [MEF133.1 O3]

- Description
- PM Job Priority

[R9] Administrator's Create Performance Profile request **MUST** include the following attributes:

- `jobType`
- `outputFormat`
- `resultFormat`

[R10] Performance Profile is unique on the envelope level within the Seller/Server's network.

6.1.3. Create Performance Profile Response

Entities used for providing a response to Create Performance Profile request are presented in Figure 12. The Seller/Server responds with a `PerformanceProfile` type, which adds some attributes to the `PerformanceProfile_Create` that was used in the Buyer/Client request.

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

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

`Performance Profile` Create Response

```
{
  "buyerProfileId": "a5240110-0945-11ee-be56-0242ac120002",
  "description": "Exemplary Create Performance Profile request",
  "granularity": "10 second",
  "jobPriority": 5,
  "jobType": "proactive",
  "outputFormat": "json",
  "reportingPeriod": "1 hour",
  "resultFormat": "payload",
  "creationDate": "2023-06-12T17:47:50.399Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/8df0981a-0949-11ee-be56-0242ac120002", << added by SOF >>
  "id": "8df0981a-0949-11ee-be56-0242ac120002", << added by SOF >>
  "lastModifiedDate": "2023-06-12T17:47:50.399Z", << added by SOF >>
  "state": "active" << added by SOF >>
}
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R11] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R12] The Seller/Server **MUST** specify the following attributes in a response:

- `creationDate`
- `id`
- `state`

[R13] The `id` **MUST** remain the same value for the life of the Performance Profile.

6.1.4. Performance Profile State Machine

Figure 13 presents the Performance Profile state machine:

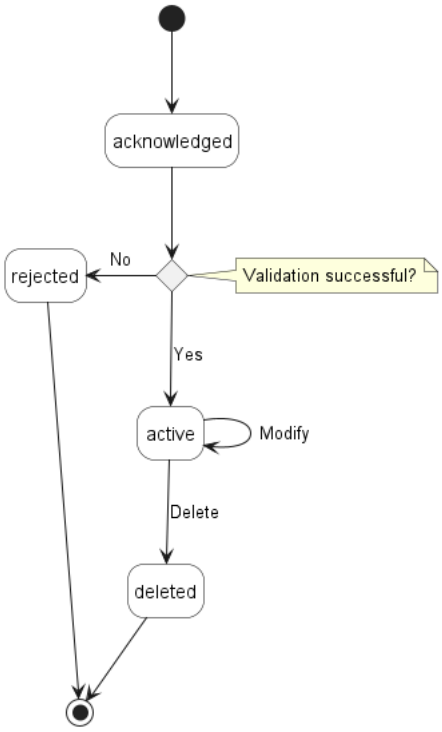


Figure 13. Performance Profile State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `PerformanceProfile` in `acknowledged` status. Before moving to the `active` state, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `rejected` state if some issues are found. The `performanceProfile.rejectionReason` acts as a placeholder to provide a detailed description of what caused the problem.

Table 8 presents the mapping between the API `status` names and the MEF W133.1 naming, together with statuses' description.

state	MEF W133.1 name	Description
`acknowledged`	Acknowledged	A Create Performance Monitoring Profile request has been received by the Server and has passed basic validation. Performance Monitoring Profile Identifier is assigned in the `Acknowledged` state. The request remains in the `Acknowledged` state until all validations as applicable are completed. If the attributes are validated the Performance Monitoring Profile moves to the `Active` state. If not all attributes are validated, the request moves to the Rejected state.
`active`	Active	A Performance Monitoring Profile is active and can be used as a template for Performance Monitoring Job creation.
`deleted`	Deleted	A Performance Monitoring Profile that does not have any Performance Monitoring Jobs attached is deleted.
`rejected`	Rejected	A Create Performance Monitoring Profile request fails validation and is rejected with error indications by the Server.

Table 8. Performance Profile states

[R14] The Seller/Server **MUST** support all Performance Profile statuses and their associated transitions as described in Figure 13 and Table 8.

6.2. Use Case 2: Retrieve List of Performance Profile

The Buyer/Client can retrieve a list of `PerformanceProfile` by using a `GET /performanceProfile` operation with desired filtering criteria.

[O5] The Buyer/Client Retrieve List of Performance Profiles request **MAY** contain none or more of the following attributes as filter criteria:

- `buyerProfileId`
- `state`
- `creationDate.gt`
- `creationDate.lt`
- `jobType`
- `granularity`
- `reportingPeriod`
- `jobPriority`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/performanceProfile?state=active&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Performance Profile objects that are in the `active` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `PerformanceProfile_Find` objects matching the criteria. To get all details, the Buyer/Client has to query a specific `PerformanceProfile` by its `id`. Details related to pagination are described in section 7.1.2

If the quantity of the records requested to be returned exceeds a Seller/Server policy, the Seller/Server must choose to respond with either:

- An empty list and message that indicates the result set is too large or
- A response that indicates the result is too large and includes a subset of the matching PM Profiles.

[R15] The Seller/Server **MUST** support the retrieval of a Performance Profile List Use Case. [MEF133.1 R44]

[R16] The Administrator or Buyer/Client **MUST** support the retrieval of a Performance Profile List Use Case. [MEF133.1 R45]

[R17] The Seller **MUST** include following attributes (if set) in the `PerformanceProfile_Find` object in the response: [MEF133.1 R46]

- `description`
- `id`
- `state`

[R18] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list. [MEF133.1 R47]

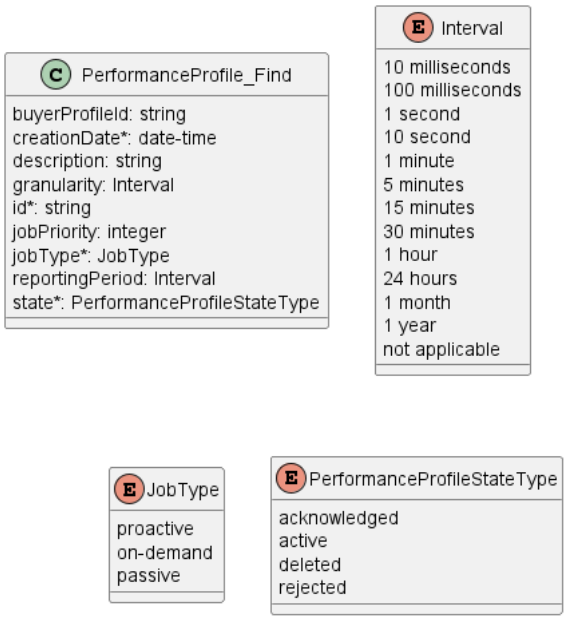


Figure 14. Use Case 2: Retrieve Performance Profile List - Model

6.3. Use Case 3: Retrieve Performance Monitoring Profile by Profile Identifier

The Buyer/Client can get detailed information about the Performance Profile from the Seller/Server by using a `GET /performanceProfile/{{id}}` operation. The payload returned in the response is a full representation of Performance Profile and includes all attributes the Buyer/Client has provided while sending a Performance Profile create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Performance Profile. Get List returns `PerformanceProfile_Find` object which is a subset of `PerformanceProfile` returned by Get by Identifier operation. A response to a get by id for a `PerformanceProfile` with `id=8df0981a-0949-11ee-be56-0242ac120002` would return exactly same response as presented in section 6.1.3.

[R19] The Seller/Server **MUST** support the retrieval of a Performance Profile Use Case. [MEF133.1 R48]

[R20] The Administrator or Buyer/Client **MUST** support the retrieval of a Performance Profile Use Case. [MEF133.1 R49]

[R21] In case `id` does not allow finding a `PerformanceProfile` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R22] The Seller/Server **MUST** include following attributes in the `PerformanceProfile` object in the response:

- `id`
- `description`

[R23] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

6.4. Use Case 4: Modify Performance Monitoring Profile

The update operation is realized with the use of the REST PATCH operation. For that purpose, a specialized type `PerformanceProfile_Update` is provided. It consists of attributes limited to a subset that includes only the updateable attributes. Modify Performance Profile operation is allowed only for API client with Administrator access rights. The Performance Profile cannot be used by a Performance Job, otherwise Performance Profile cannot be modified.

[R24] - Modify Performance Monitoring Profile **MUST** be available only to Administrator role

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

Figure 15 presents the model used in the PATCH request. The Seller/Server responds with a `PerformanceProfile` type which is a full representation of Performance Profile instance.

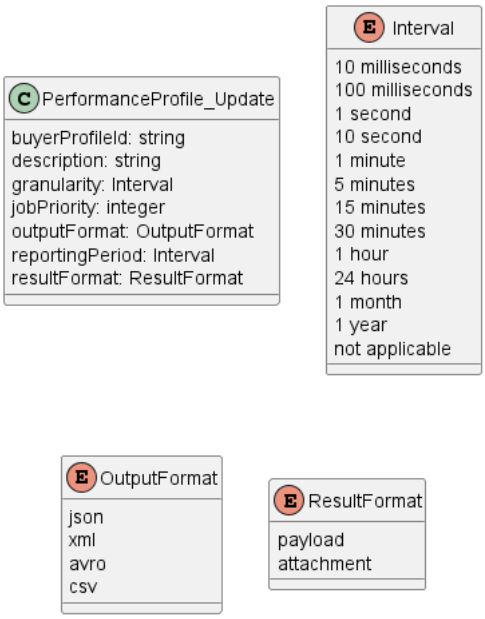


Figure 15. Patch request Model

[O6] The Seller/Server **MAY** support the modification of a Performance Profile Use Case. [MEF133.1 O4]

[O7] The Administrator **MAY** support the modification of a Performance Profile Use Case. [MEF133.1 O5]

[R25] In case `id` does not allow to find a `PerformanceProfile` that is to be updated in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R26] The Seller/Server **MUST** return an error (`Error422`) if the Performance Profile `state` is not `active`.

The example below shows a request to patch a `PerformanceProfile` that was created in section 6.1.2.

The request below aims to:

- update `description`
- modify `granularity` of the performance measurements collection
- change `reportingPeriod` which is a frequency of reports generation

```
{
  "description": "string",
  "granularity": "5 minute",
  "reportingPeriod": "1 hour",
}
```

6.5. Use Case 5: Delete Performance Monitoring Profile

The Buyer/Client may request to delete a Performance Profile by using `DELETE /performanceProfile/{{id}}` endpoint. This operation only requires providing the `id` in the path and has an empty `204` confirmation response.

Delete Performance Profile operation is allowed only for API client with Administrator access rights.

[R27] Delete Performance Monitoring Profile **MUST** be available only to Administrator role

The sequence diagram below presents this use case in detail.

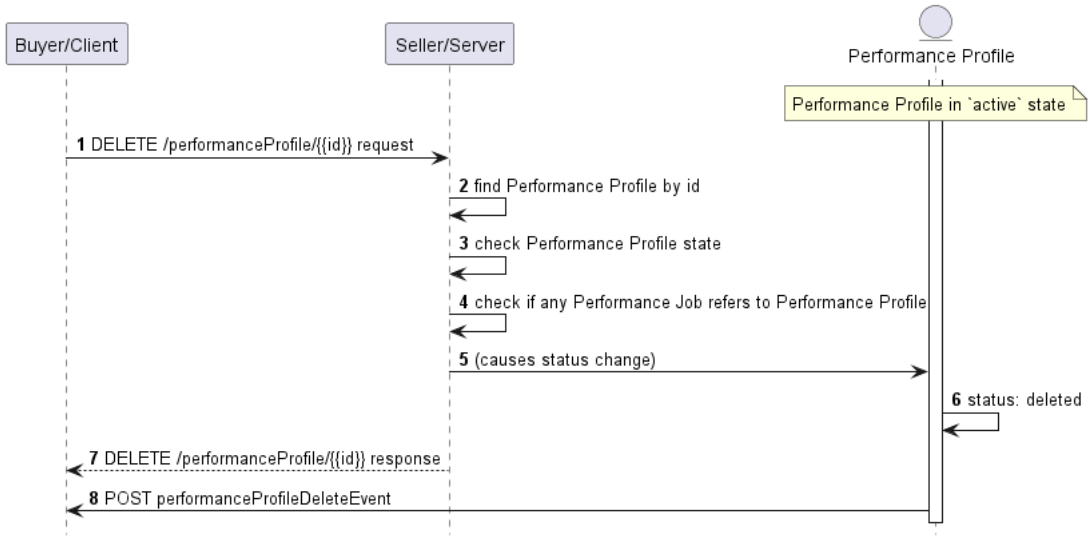


Figure 16. Delete Performance Profile Flow

The Seller/Server verifies the request, then searches for a Performance Profile to be deleted by given `id`. If found, the status is verified (`active`). The Seller/Server checks also if there are any active Performance Job objects that refer to the Performance Profile (active means state of `PerformanceJob` is different from `rejected`, `completed`, `cancelled`, or `resource-unavailable`). If everything is verified correctly, the Seller moves the ticket to the `deleted` status, sends a successful response to a request followed by `performanceProfileDeleteEvent`

[O8] The Seller/Server **MAY** support the deletion of a Performance Profile Use Case. [MEF133.1 O6]

[O9] The Administrator **MAY** support the deletion of a Performance Profile Use Case. [MEF133.1 O7]

[R28] The Seller/Server **MUST** return an error (`Error422`) if the Performance Profile is referenced to by an active `PerformanceJob` (active means state of `PerformanceJob` is different from `rejected`, `completed`, `cancelled`, or `resource-unavailable`)

[R29] In case there is no `PerformanceProfile` with provided `id`, an error response `Error404` **MUST** be returned.

6.6. Use case 6: Create Performance Monitoring Job

A Performance Monitoring Job is used by the client to specify the performance monitoring objectives specific to each measurement point which could be an ordered pair (an association between two end points, e.g. UNIs) or an entity (defined as an object other than a service that can be monitored and have associated telemetry, e.g. port). Examples of performance objectives encompass various metrics such as frame/packet delay, frame/packet loss ratio, inter-frame/packet delay variation, and more. These objectives serve as measurable criteria for assessing the performance characteristics of a service. Performance Jobs are responsible for provisioning these measurement points, performance objectives, together with measurement intervals and schedules. Performance objectives are typically associated with an SLS but can be used for an On-Demand Job for making measurements as part of a troubleshooting procedure.

The Performance Monitoring Job provides also the capability to provision and collect passive statistics. These statistics encompass various telemetry data associated with interfaces, (Net/Application) Flows, VLANs, bridging/Ethernet, IP, TCP, and UDP layers. It is important to note that these measured statistics fall outside the scope of measuring and responding to performance objectives. Nevertheless, the same set of APIs is employed to manage both types of data. In some cases, these statistics may not require a Performance Job to be instantiated prior to the collection, but are enabled and ready for collection on an interface, VLAN, etc.

The Performance Monitoring Jobs should result in Performance Measurement Collections (Reports) that will provide the Buyer/Client with performance objective results.

There are three types of Performance Job:

- Proactive - carried on continuously to permit timely reporting of performance status and to support SLS measurement. Typically, it runs indefinitely.
- On-Demand - initiated for a limited time, typically a single run or non-continual run, to carry out the performance measurement tests and support troubleshooting during service assurance.
- Passive - supports the collection and reporting of network and service statistics. The statistics collections include but are not limited to telemetry associated with an interface, (Net/Application) Flow, VLAN, bridging/Ethernet, IP, TCP, UDP layers.

Proactive, On-Demand and Passive Performance Jobs can use Performance Monitoring Profiles for the provisioning. In case Performance Monitoring Job is created without relationship to Performance Profile, all necessary attributes have to be associated with a Performance Job object. Create Performance Job request can refer to Performance Profile by:

- reference - direct reference by using Performance Profile id, or
- value - assigning characteristics defined by Performance Profile directly in the Performance Job.

[O10] Performance Job **MAY** use Performance Monitoring Profile as a template.

6.6.1. Interaction flow

The flow of this use case is shown in Figure 17.

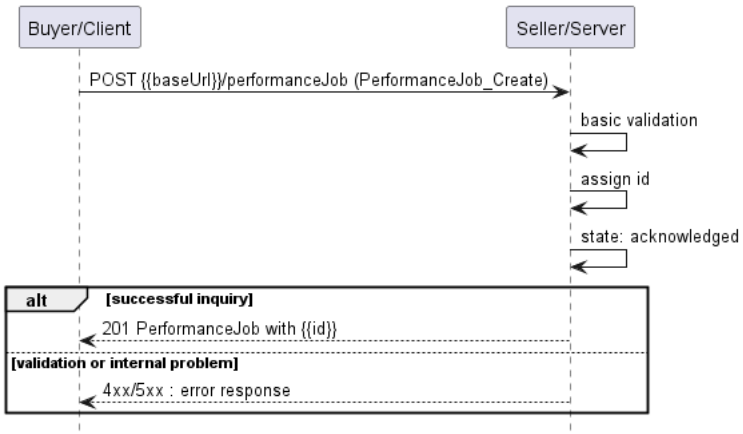


Figure 17. Use Case 6 - Performance Monitoring Job create request flow

The Buyer/Client sends a request with a `PerformanceJob_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `PerformanceJob` type in the response body, with a `state` set to `acknowledged`. From this point, the Performance Job is ready for further processing. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `PerformanceJob`. The two patterns are presented in the following diagrams.

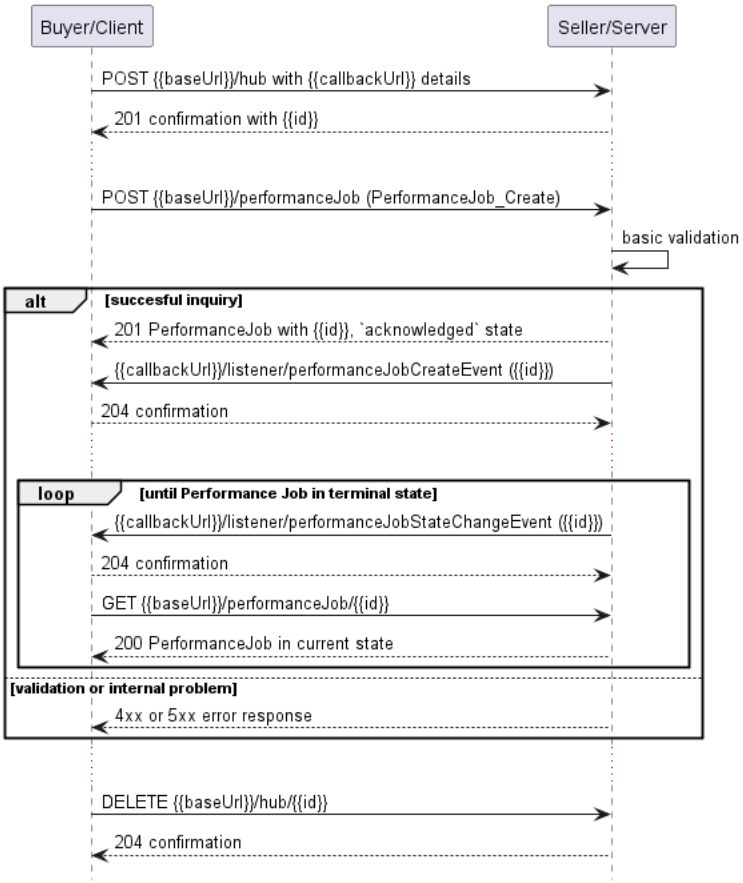


Figure 18. Performance Job progress tracking - Notifications

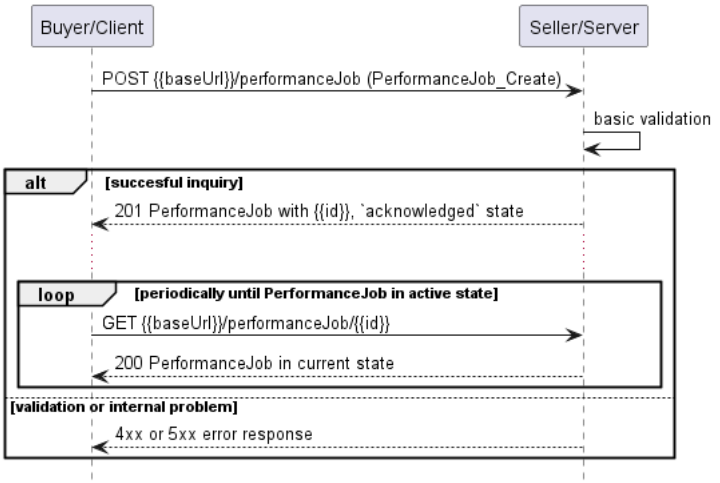


Figure 19. Performance Job progress tracking - Polling

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

6.6.2. Create Performance Job Request

Figure 20 presents the most important part of the data model used during the Create Performance Job request (`POST /performanceJob`) and response. The model of the request message - `PerformanceJob_Create` is a subset of the `PerformanceJob` model and contains only attributes that can

(or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information.

Note: `PerformanceJob_Create` is an entity used by the Buyer/Client to make a request. `PerformanceJob` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `PerformanceJob_Common` has been introduced (this class is not supposed to be used directly in the exchange).

A `PerformanceJob_Create` defines measurement intervals, schedules, and objectives of performance monitoring (in `servicePayloadSpecificAttributes` section). It also refers to existing `PerformanceProfile` by its `id` or directly provides values of attributes defined by `PerformanceProfile` type. See chapter section 6.6.5 for more details.

Section `servicePayloadSpecificAttributes` of the create Performance Job request allows for the introduction of service-specific properties of performance monitoring as the API payload. The extension mechanism is described in detail in Section 5.3.

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

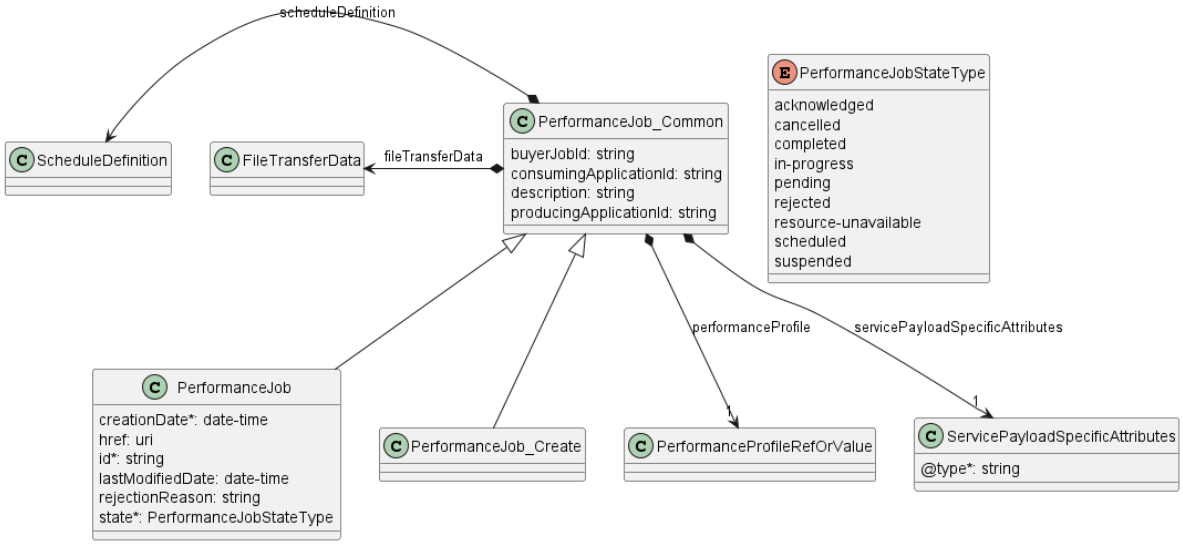


Figure 20. Performance Job Key Entities

To send a create Performance Job request the Buyer/Client uses the `createPerformanceJob` operation from the API: `POST /performanceJob`. For clarity, some of create Performance Job payload's attributes might be omitted to improve examples' readability.

`Performance Job` Create Request

```
{
  "buyerJobId": "TestJob12345",
  "consumingApplicationId": "CUS",
  "description": "Exemplary Create Performance Job request",
  "fileTransferData": {
    "fileFormat": "JSON",
```

```
      "fileLocation": "ftp://cus.com/",
      "transportProtocol": "ftp",
      "compressionType": "NO_PACKING"
    },
    "performanceProfile": {
      "@type": "PerformanceProfileRef",
      "id": "8df0981a-0949-11ee-be56-0242ac120002"
    },
    "producingApplicationId": "SOF",
    "scheduleDefinition": {
      "recurringFrequency": {
        "recurringFrequencyValue": 1,
        "recurringFrequencyUnits": "HOURS"
      },
      "scheduleDefinitionStartTime": "2023-06-01T08:02:01.370Z"
    },
    "servicePayloadSpecificAttributes": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "interface": {
        "ipvcEndpoint": [
          "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
          "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
        ],
        "name": "slsRpPairTest1",
        "description": "Exemplary performance monitoring service pair",
        "cloudService": true
      }
    }
  }
```


[R30] The Buyer's/Client's Create Performance Job **MUST** support the following attributes: [MEF133.1 R50, R85]

- Buyer Profile ID
- Consumer Application Indicator
- Granularity
- Job Priority
- Job Type
- Output
- PM Profile ID (if used)
- Reporting Period
- Result Format
- Schedule Definition
- Service Specific Payload

[O11] The Buyer's/Client's Create Performance Job **MAY** contain the following attributes: [MEF133.1 O14, O19]

- Description
- PM Job Priority

[O12] A Performance Job **CAN** be scheduled as reoccurring. [MEF133.1 O15]

6.6.3. Create Performance Job Response

Entities used for providing a response to Create Performance Job request are presented in Figure 20. The Seller/Server responds with a `PerformanceJob` type, which adds some attributes

(like `id` or `state`) to the `PerformanceJob_Create` that was used in the Buyer/Client request.

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

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

`Performance Job` Create Response


```
{
  "buyerJobId": "TestJob12345",
  "consumingApplicationId": "CUS",
  "description": "Exemplary Create Performance Job request",
  "fileTransferData": {
    "fileFormat": "JSON",
    "fileLocation": "ftp://cus.com/",
    "transportProtocol": "ftp",
    "compressionType": "NO_PACKING"
  },
  "performanceProfile": {
    "@type": "PerformanceProfileRef",
    "id": "8df0981a-0949-11ee-be56-0242ac120002"
  },
  "producingApplicationId": "SOF",
  "scheduleDefinition": {
    "recurringFrequency": {
      "recurringFrequencyValue": 1,
      "recurringFrequencyUnits": "HOURS"
    },
    "scheduleDefinitionStartTime": "2023-06-01T08:02:01.370Z"
  },
  "servicePayloadSpecificAttributes": {
    "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
    "interface": {
      "ipvcEndpoint": [
        "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
        "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
      ],
      "name": "slsRpPairTest1",
      "description": "Exemplary performance monitoring service pair",
      "cloudService": true
    }
  },
  "creationDate": "2023-06-01T08:02:01.370Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691", << added by SOF >>
  "id": "755e55e2-72b0-4e3b-af00-693e3beac691", << added by SOF >>
  "lastModifiedDate": "2023-06-01T08:02:01.370Z", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```


Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R31] The Seller/Server **MUST** assign a Job Identifier to the Performance Job that is unique within the network. [MEF133.1 R51, R86]

[R32] The Performance Job Identifier supplied by the Seller/Server **MUST** be unique within the Seller/Server's network. [MEF133.1 R52, R87]

[R33] The Performance Job **MUST** use the attributes included in the Buyer's/Client's Create Performance Job request. [MEF133.1 R53, R88]

[R34] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R35] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDate`

[R36] The `id` **MUST** remain the same value for the life of the Performance Job.

6.6.4. Performance Job State Machine

Figure 21 presents the Performance Job state machine:

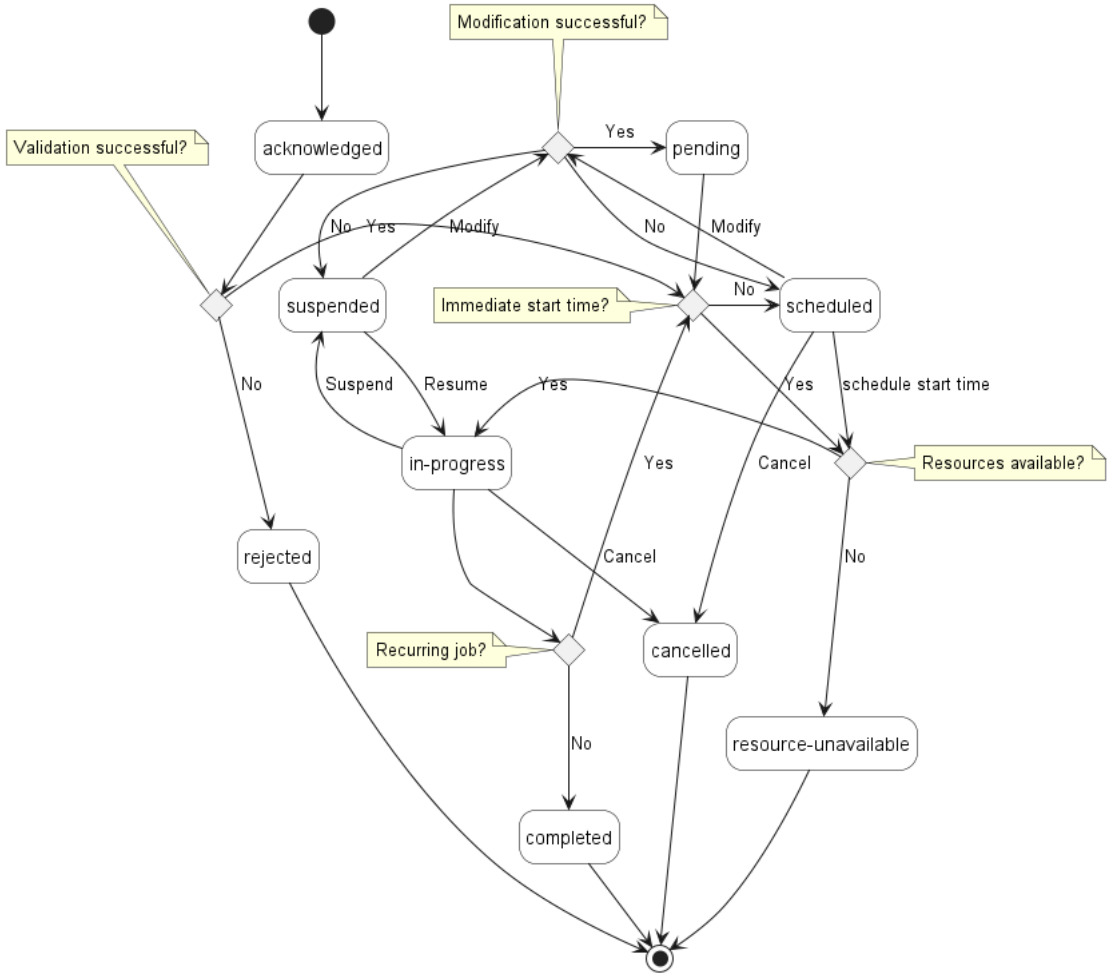


Figure 21. Performance Job State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `PerformanceJob` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `rejected` state if some issues are found. The `performanceJob.rejectionReason` acts as a placeholder to provide a detailed description of what

caused the problem. `PerformanceJob` moves to either `scheduled` or `in-progress` state depending on the assigned schedule. `PerformanceJob` remains `scheduled` state until schedule start time is reached. `PerformanceJob` that is starting needs appropriate resources on Seller/Server side. If required resources cannot be assigned, `PerformanceJob` moves to `resource-unavailable` state. After completion, Seller/Server verifies if `PerformanceJob` is recurring. If yes, `PerformanceJob` moves to either `scheduled` or `in-progress` state. Otherwise, it moves to `completed` state. `PerformanceJob` can be cancelled when in `scheduled` or `in-progress`. When cancellation is successful, `PerformanceJob` moves to `cancelled` state. `PerformanceJob` can be modified only in `scheduled` or `suspended` state. Modification includes an intermediary `pending` step.

Table 9 presents the mapping between the API `status` names and the MEF E133.1 naming, together with statuses' description.

state	MEF W133.1 name	Description
`acknowledged`	Acknowledged	A Create Performance Monitoring Job request has been received by the Seller/Server and has passed basic validation. Performance Monitoring Job Identifier is assigned in the Acknowledged state. The request remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated the request determines if the start time is immediate or scheduled. If immediate, the Performance Monitoring Job moves to the In-progress state. Otherwise, the Performance Monitoring Job moves to the Scheduled state. If not all attributes are validated, the request moves to the Rejected state.
`cancelled`	Cancelled	A Performance Monitoring Job that is In-Progress, Suspended or Scheduled is cancelled.
`completed`	Completed	A non-recurring Performance Monitoring Job finished execution.
`in-progress`	In-Progress	A Performance Monitoring Job is running. Upon completion of the Job, a determination if the Performance Monitoring Job is a one-time Job or is recurring is performed. If the Performance Monitoring Job is a one-time Job, the state of the Performance Monitoring Job moves to the Completed state. If the Performance Monitoring Job is recurring, the Performance Monitoring Job circles back to determine if it has an immediate start time or a scheduled start time. In case a Suspend Performance Monitoring Job request is accepted, the Job moves to the Suspended state. If a Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state.

state	MEF W133.1 name	Description
`pending`	Pending	A Modify Performance Monitoring Job request has been accepted by the Seller/Server. The Performance Monitoring Job remains in the Pending state while updates to the Job are completed. Once updates are complete, the Job returns to the Scheduled or In-Progress status depending on the schedule definition.
`rejected`	Rejected	A Create Performance Monitoring Job request fails validation and is rejected with error indications by the Seller/Server.
`resource-unavailable`	Resource Unavailable	A Performance Monitoring Job cannot be allocated necessary resources when moving to execution (In-Progress state).
`scheduled`	Scheduled	A Performance Monitoring Job is created that does not have an immediate start time. The Performance Monitoring Job stays in the Scheduled state until the start time is reached. The Performance Monitoring Job then moves to In-Progress. If Cancel Performance Monitoring Job request is accepted, Job moves to Cancelled state. If modify Performance Monitoring Job request is accepted, Job moves to Pending state.
`suspended`	Suspended	A Suspend Performance Monitoring Job request is accepted by the Seller/Server. The Job remains in the Suspended state until a Resume Performance Monitoring Job request is accepted by the Seller/Server at which time the Job returns to the In-Progress state. If Cancel Performance Monitoring Job request is accepted, Job moves to Cancelled state. If modify Performance Monitoring Job request is accepted, Job moves to Pending state.

Table 9. Performance Job State Machine states

[R37] The Seller/Server **MUST** support all Performance Job statuses and their associated transitions as described in Figure 21 and Table 9.

6.6.5. Relationship to Performance Profile

Performance Profile is a template defining common attributes for multiple Performance Jobs. There are two options for creation of a Performance Job:

- specify relationship to `PerformanceProfile` by its `id`

- provide required attributes that are typically defined by `PerformanceProfile` type directly in the request. `PerformanceJob_Create` class used as a payload for `createPerformanceJob` operation supports both options in `performanceProfile` attribute which is of type `PerformanceProfileRefOrValue`. Depending on the value of `@type` attribute (discriminator) it is possible to refer to existing `PerformanceProfile` object (`@type=PerformanceProfileRef`) or specify attributes that describe `PerformanceProfile` (`@type=PerformanceProfileValue`). ***Note***: Defining attributes related to `PerformanceProfile` in Performance Job create request does not create new `PerformanceProfile` object.

Figure 22 presents `PerformanceJob_Create` and related entities that allow for referencing to Performance Profile or specifying corresponding attributes.

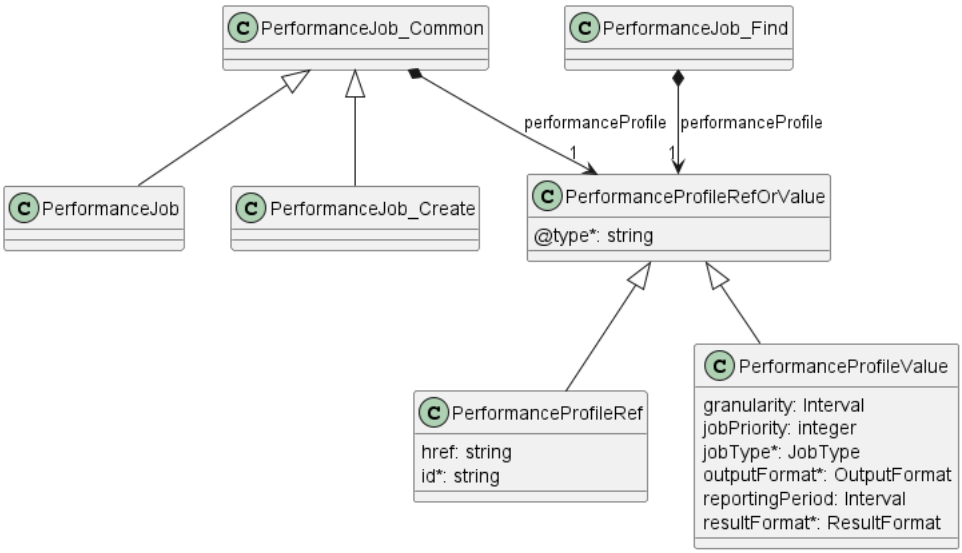


Figure 22. Relationship to Performance Profile

6.7. Use Case 7: Retrieve List of Performance Job

The Buyer/Client can retrieve a list of `PerformanceJob` by using a `GET /performanceJob` operation with desired filtering criteria.

[O13] The Buyer/Client Retrieve List of Performance Jobs request **MAY** contain none or more of the following attributes as filter criteria:

- `buyerJobId`
- `performanceProfileId`
- `state`
- `creationDate.gt`
- `creationDate.lt`
- `jobType`
- `granularity`
- `reportingPeriod`
- `consumingApplicationId`

- `producingApplicationId`
- `jobPriority`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/performanceJob?state=suspended&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Performance Job objects that are in the `suspended` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `PerformanceJob_Find` objects matching the criteria. To get all details, the Buyer/Client has to query a specific `PerformanceJob` by its `id`. Details related to pagination are described in section 7.1.2

If the quantity of the records requested to be returned exceeds a Seller/Server policy, the Seller/Server must choose to respond with either:

- An empty list and message that indicates the result set is too large or
- A response that indicates the result is too large and includes a subset of the matching PM Jobs.

[R38] The Seller/Server's response to the Buyer's/Client's Retrieve List of Performance Jobs **MUST** include the following attributes as applicable:

- `buyerJobId`
- `consumingApplicationId`
- `creationDate`
- `description`
- `id`
- `performanceProfile`
- `producingApplicationId`
- `scheduleDefinition`
- `state`

[R39] If the Seller/Server validates the Buyer's/Client's request but finds no matching Performance Jobs, the Seller/Server **MUST** return an empty list.

Figure 23 presents entities related to the use case.

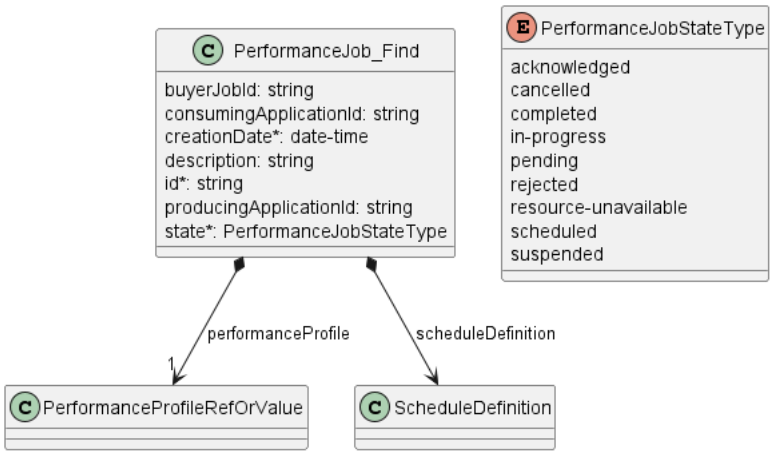


Figure 23. Use Case 7: Retrieve Performance Job List - Model

6.8. Use Case 8: Retrieve Performance Monitoring Job by Job Identifier

The Buyer/Client can get detailed information about the Performance Job from the Seller/Server by using a `GET /performanceJob/{{id}}` operation. The payload returned in the response is a full representation of Performance Job and includes all attributes the Buyer/Client has provided while sending a Performance Job create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Performance Job. Get List returns `PerformanceJob_Find` object which is a subset of `PerformanceJob` returned by Get by Identifier operation. A response to a get by id for a `PerformanceJob` with `id=755e55e2-72b0-4e3b-af00-693e3beac691` would return exactly same response as presented in section 6.6.3.

[R40] The Buyer/Client's Retrieve Performance Job by Job Identifier request **MUST** contain the Performance Job Identifier. [MEF133.1 R71]

[R41] In case `id` does not allow finding a `PerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R42] The Seller/Server **MUST** include following attributes in the `PerformanceJob` object in the response:

- `id`
- `description`

[R43] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller. [MEF133.1 R72]

6.9. Use Case 9: Modify Performance Monitoring Job

Due to the need for provisioning and resource reservation on the SOF side, the modification operation associated with Performance Monitoring Job may exhibit prolonged duration. Consequently, this operation is implemented through a separate lifecycle process.

6.9.1. Interaction flow

The flow of this use case is shown in Figure 24.

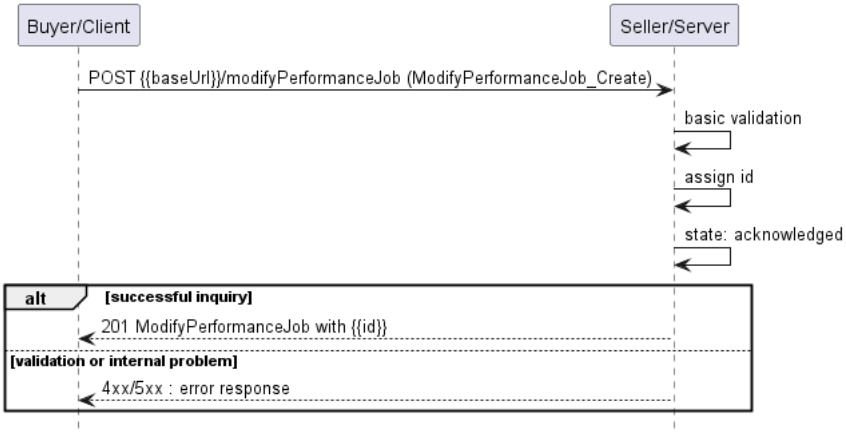


Figure 24. Use Case 9 - Modify Performance Monitoring Job create request flow

The Buyer/Client sends a request with a `ModifyPerformanceJob_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `ModifyPerformanceJob` type in the response body, with a `state` set to `acknowledged`. Further processing is performed by Seller/Server which will in case of success update Performance Monitoring Job. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `ModifyPerformanceJob`. The two patterns are presented in the following diagrams.

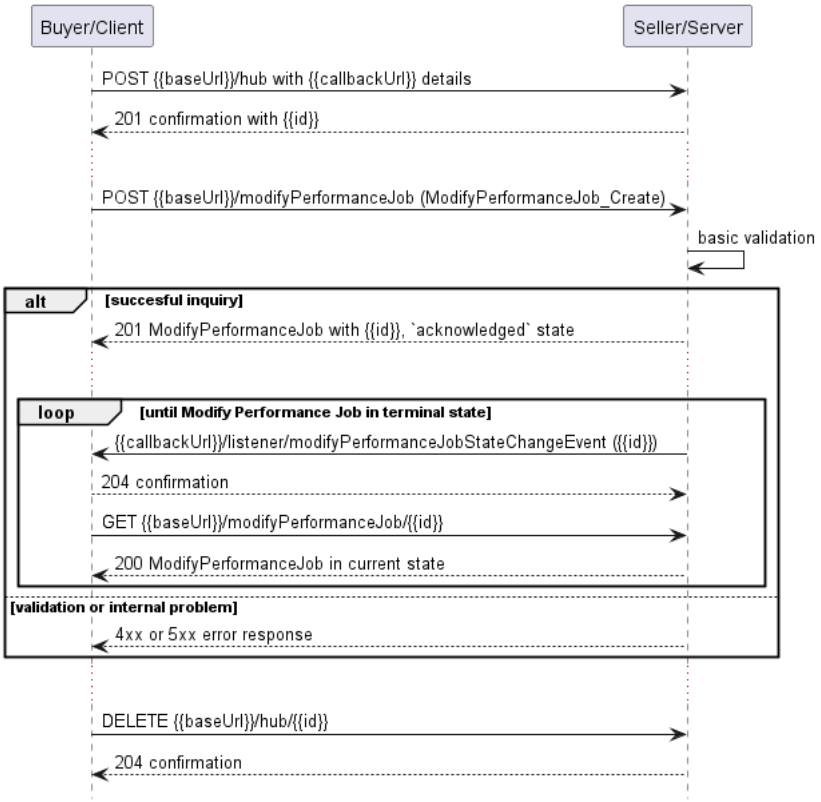


Figure 25. Modify Performance Job progress tracking - Notifications

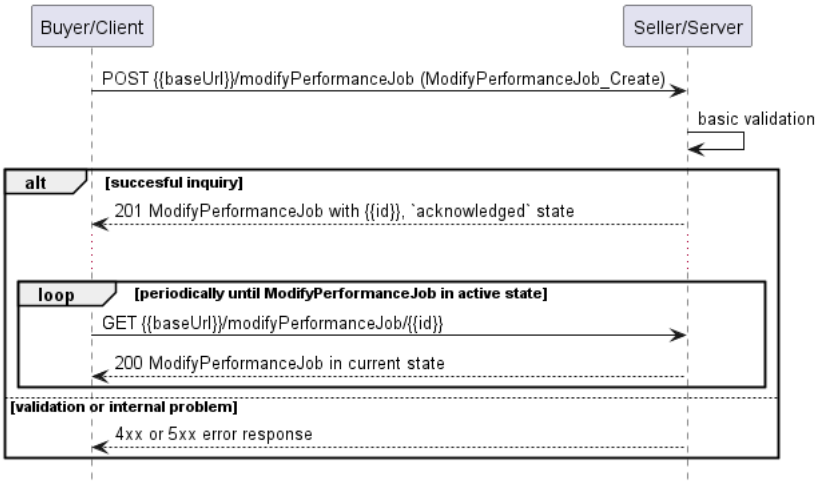


Figure 26. Modify Performance Job progress tracking - Polling

Note: The Modify Performance Job process is altering the state of the job itself. It is important to note that notifications resulting from changes in the state of the Performance Job are not represented in figures 25 and 26.

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

[R44] The Seller/Server **MUST** support Performance Job modifications. [MEF133.1 R56]

[R45] The Seller/Server **MUST** support Statistics Collection Job modifications. [MEF133.1 R91]

6.9.2. Modify Performance Monitoring Job Request

Figure 27 presents the most important part of the data model used during the Modify Performance Job request (`POST /modifyPerformanceJob`) and response. The model of the request message - `ModifyPerformanceJob_Create` is a subset of the `ModifyPerformanceJob` model and contains only attributes that can (or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information.

Note: `ModifyPerformanceJob_Create` is an entity used by the Buyer/Client to make a request. `ModifyPerformanceJob` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `ModifyPerformanceJob_Common` has been introduced (this class is not supposed to be used directly in the exchange).

A `ModifyPerformanceJob_Create` is a subset that includes only the updateable attributes. It is important to notice that updating the reference to Performance Profile must not be possible. In order to change this assignment, existing Performance Job must be cancelled and replaced by a new Job that relates to the relevant Profile. Modification of Performance Job allows for changing attributes defined directly by `PerformanceJob` type or Performance Profile attributes that are defined by value. These attributes are contained in `performanceProfile` group. The `performanceJobRef` section of `ModifyPerformanceJob_Create` is used to specify which Performance Job object is a subject of the modification process (relationship by reference using `id` of the Job).

Note: Modifying attributes defined by the Performance Profile type, when a Job uses a reference to a Performance Profile object, cannot modify the Performance Profile itself.

Note: Only attributes that should be modified on the Performance Job, should be included in the Modify Performance Job Request.

Section `servicePayloadSpecificAttributes` of the Modify Performance Job request allows for the introduction of service-specific properties of performance monitoring as the API payload. The extension mechanism is described in detail in Section 5.3.

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

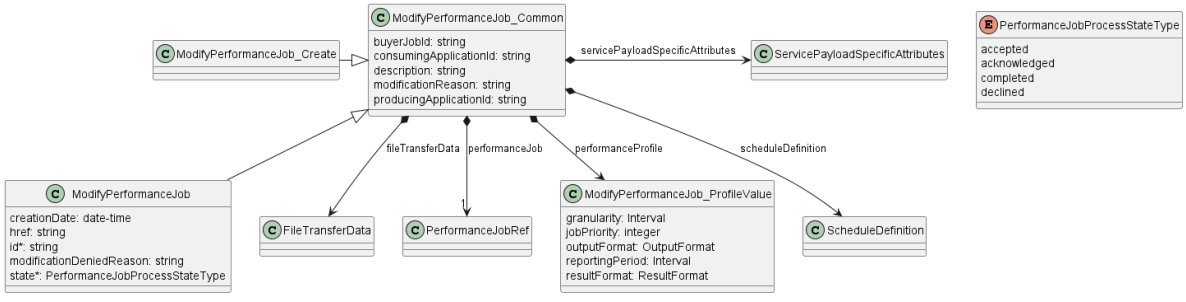


Figure 27. Modify Performance Job Key Entities

To send a Modify Performance Job request the Buyer/Client uses the `modifyPerformanceJob` operation from the API: `POST /modifyPerformanceJob`. Some of the payload's attributes might be omitted to improve examples' readability.

The example below shows a request to create a modification process for `PerformanceJob` that was created in section 6.6.2.

The request below aims to:

- update `buyerJobId`
- modify `fileTransferData`
- change `description` of the Performance Job

```
{
  "buyerJobId": "TestJob54321",
  "description": "Performance Job after modification",
  "fileTransferData": {
    "fileFormat": "JSON",
    "fileLocation": "ftp://cus.com/newLocation",
    "transportProtocol": "ftp",
    "compressionType": "NO_PACKING"
  },
  "modificationReason": "Modify Performance Job sample",
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  }
}
```

[R46] The Buyer/Client Modify Performance Job request **MUST** include the following attributes: [MEF133.1 R55, R90]

- `performanceJob`

[O14] The Buyer/Client **MAY** include one or more of the following attributes of `ModifyPerformanceJob_Create` in the request: [MEF133.1 O16, O20]

- buyerJobId
- consumingApplicationId
- description
- fileTransferData
- granularity
- jobPriority
- modificationReason
- performanceProfile
- producingApplicationId
- reportingPeriod
- scheduleDefinition
- servicePayloadSpecificAttributes

Note: In case Performance Job is running e.g., once a day for a short period of time, it may be difficult to change its state. If action arrives when Performance Job is running, it is recommended to run until the end and only afterwards action should be applied. [MEF133.1 O16, O26]

6.9.3. Modify Performance Monitoring Job Response

Entities used for providing a response to Modify Performance Job request are presented in Figure 27. The Seller/Server responds with a `ModifyPerformanceJob` type, which adds some attributes (like `id` or `state`) to the `ModifyPerformanceJob_Create` that was used in the Buyer/Client request.

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

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

```
{
  "buyerJobId": "TestJob54321",
  "description": "Performance Job after modification",
  "fileTransferData": {
    "fileFormat": "JSON",
    "fileLocation": "ftp://cus.com/newLocation",
    "transportProtocol": "ftp",
    "compressionType": "NO_PACKING"
  },
  "modificationReason": "Modify Performance Job sample",
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "creationDate": "2023-06-19T12:58:17.088Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/9c51d971-185d-403e-952f-2110f33a9628", << added by SOF >>
  "id": "9c51d971-185d-403e-952f-2110f33a9628", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R47] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R48] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDate`

[R49] The `id` **MUST** remain the same value for the life of the Modify Performance Job.

In case Seller/Server cannot successfully validate the request, Modify Performance Job process fails, which results in setting state to `declined` with a proper explanation in `modificationDeniedReason`. This includes situation when:

- `id` does not allow to find a `PerformanceJob` that is to be updated in Seller/Server's system
- requested attributes cannot be modified
- Performance Job is in the state that does not allow for modification.

6.9.4. Modify Performance Monitoring Job State Machine

Figure 28 presents the Modify Performance Monitoring Job state machine:

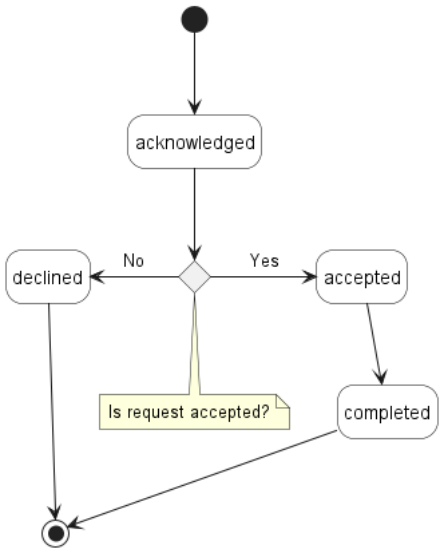


Figure 28. Modify Performance Job State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `ModifyPerformanceJob` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `declined` state if some issues are found. The `modifyPerformanceJob.modificationDeniedReason` acts as a placeholder to provide a detailed description of what caused the problem. If validation is successful, `ModifyPerformanceJob` moves to `accepted` state. At this point, related `PerformanceJob` moves to pending state and Seller/Server starts all necessary arrangements to provision modification request. `PerformanceJob` remains in `pending` state until Modify Performance Job process is finished and moved to `completed` state. This causes `PerformanceJob` state to change to `scheduled` or `in-progress` depending on the `ScheduleDefinition`.

Table 10 presents the mapping between the API `status` names and the MEF W133.1 naming, together with statuses' description. The list of statuses is the same for all processes related to Performance Job (cancel/modify/resume/suspend).

state	MEF W 133.1 name	Description

state	MEF W 133.1 name	Description
`accepted`	Accepted	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been validated and accepted by the Seller/Server.
`acknowledged`	Acknowledged	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been received by the Seller/Server and has passed basic validation. Performance Monitoring Job Process Identifier is assigned in the Acknowledged state. The request remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated, the request moves to the Accepted state. If not all attributes are validated, the request moves to the Declined state.
`completed`	Completed	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been completed by the Seller/Server.
`declined`	Declined	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has failed validation and been declined by the Seller/Server.

Table 10. Performance Job Process State Machine states

[R50] The Seller/Server **MUST** support all Modify Performance Job statuses and their associated transitions as described in Figure 28 and Table 10.

6.10. Use Case 10: Retrieve Modify Performance Monitoring Job List

The Buyer/Client can retrieve a list of Modify Performance Job objects by using a `GET /modifyPerformanceJob` operation with desired filtering criteria.

[O16] The Buyer/Client Retrieve List of Modify Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDate.gt`
- `creationDate.lt`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/modifyPerformanceJob?
state=acknowledged&limit=10&offset=0
```

The example above shows a Buyer's/Client's request to get all Modify Performance Job objects that are in the `acknowledged` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `ModifyPerformanceJob_Find` objects matching the criteria. Details related to pagination are described in section 7.1.2.

[R51] The Seller **MUST** include following attributes (if set) in the `ModifyPerformanceJob_Find` object in the response:

- `id`
- `performanceJobId`
- `state`

[R52] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 29 presents entities related to the use case.

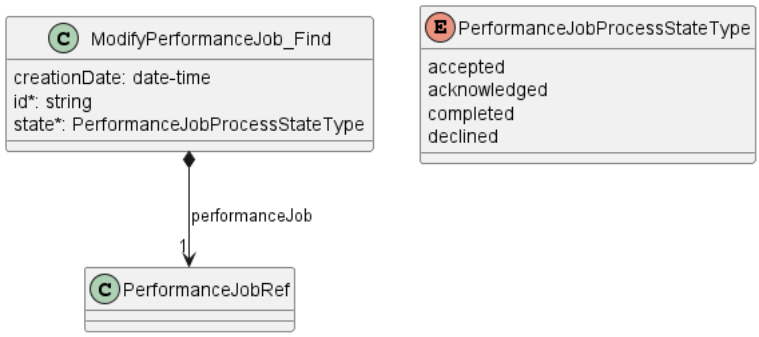


Figure 29. Use Case 10: Retrieve Modify Performance Job List - Model

6.11. Use Case 11: Retrieve Modify Performance Monitoring Job List by Identifier

The Buyer/Client can get detailed information about the Modify Performance Job from the Seller/Server by using a `GET /modifyPerformanceJob/{{id}}` operation. The payload returned in the response is a full representation of Modify Performance Job and includes all attributes the Buyer/Client has provided while sending a Modify Performance Job create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Modify Performance Job. Get List returns `ModifyPerformanceJob_Find` object which is a subset of `ModifyPerformanceJob` returned by Get by Identifier operation. A response to a Get by Identifier for a `ModifyPerformanceJob` with `id=9c51d971-185d-403e-952f-2110f33a9628` would return exactly same response as presented in section 6.9.3.

[R53] In case `id` does not allow finding a `ModifyPerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R54] The Seller/Server **MUST** include following attributes in the `ModifyPerformanceJob` object in the response:

- `id`
- `performanceJob`
- `state`

[R55] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

6.12. Use Case 12: Cancel Performance Monitoring Job

Due to the need for deprovisioning of the Performance Monitoring Job on the SOF side, the cancel operation associated with Performance Monitoring Job may exhibit prolonged duration. Consequently, this operation is implemented through a separate lifecycle process.

6.12.1. Interaction flow

The flow of this use case is shown in Figure 30.

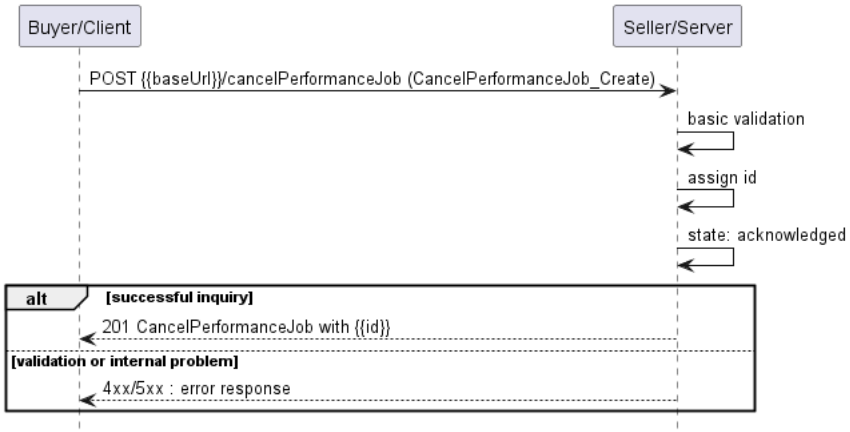


Figure 30. Use Case 12 - Cancel Performance Monitoring Job create request flow

The Buyer/Client sends a request with a `CancelPerformanceJob_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `CancelPerformanceJob` type in the response body, with a `state` set to `acknowledged`. Further processing is performed by Seller/Server which will in case of success update Performance Monitoring Job. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `CancelPerformanceJob`. The two patterns are presented in the following diagrams.

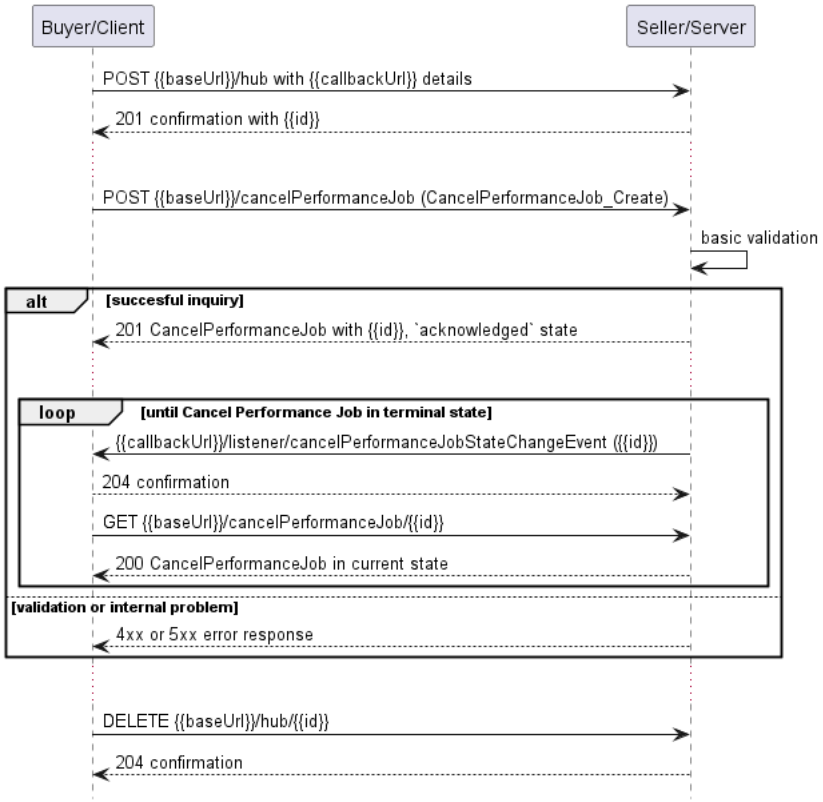


Figure 31. Cancel Performance Job progress tracking - Notifications

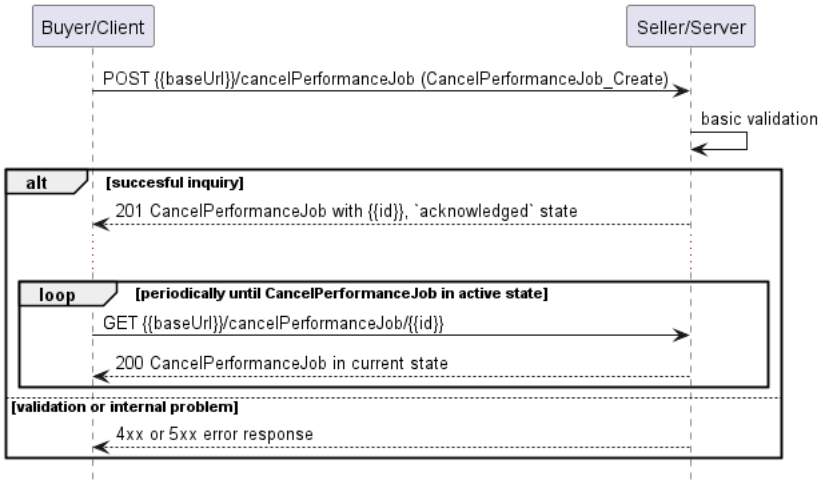


Figure 32. Cancel Performance Job progress tracking - Polling

Note: The Cancel Performance Job process is altering the state of the job itself. It is important to note that notifications resulting from changes in the state of the Performance Job are not represented in figures 31 and 32.

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

6.12.2. Cancel Performance Monitoring Job Request

Figure 33 presents the most important part of the data model used during the Cancel Performance Job request (`POST /cancelPerformanceJob`) and response. The model of the request message - `CancelPerformanceJob_Create` is a subset of the `CancelPerformanceJob` model and contains only attributes that can (or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information.

Note: `CancelPerformanceJob_Create` is an entity used by the Buyer/Client to make a request. `CancelPerformanceJob` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `CancelPerformanceJob_Common` has been introduced (this class is not supposed to be used directly in the exchange).

The `performanceJobRef` section of `CancelPerformanceJob_Create` is used to specify which Performance Job object is a subject of the cancellation process (relationship by reference using `id` of the Job).

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

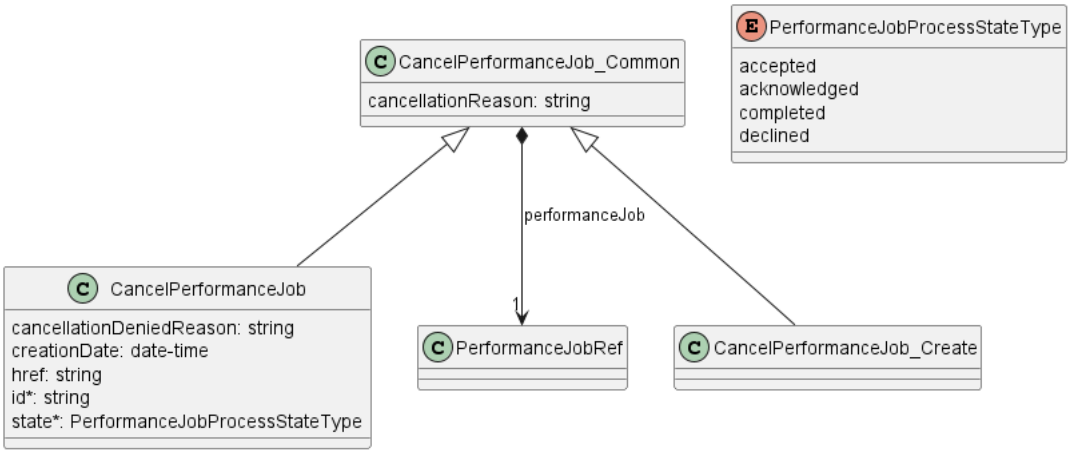


Figure 33. Cancel Performance Job Key Entities

To send a Cancel Performance Job request the Buyer/Client uses the `cancelPerformanceJob` operation from the API: `POST /cancelPerformanceJob`.

The example below shows a request to create a cancellation process for `PerformanceJob` that was created in section 6.6.2.

```
{
  "cancellationReason": "Cancel Performance Job sample",
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  }
}
```

[R56] The Buyer's/Client's Cancel Performance Job request **MUST** include the following attributes: [MEF133.1 R57]

- `performanceJob`

[R57] If the Performance Job is In-Progress or Suspended, the Seller/Server **MUST NOT** delete the Performance Job as requested by the Client. [MEF133.1 R58]

[R58] The Buyer's/Client's Cancel Statistics Collection Job request **MUST** include the following attributes: [MEF133.1 R92]

- `performanceJob`

[R59] If the Statistics Collection Job is In-Progress or Suspended, the Seller/Server **MUST NOT** delete the Performance Job as requested by the Client. [MEF133.1 R93]

Note: In case Performance Job is running e.g., once a day for a short period of time, it may be difficult to change its state. If action arrives when Performance Job is running, it is recommended to run until the end and only afterwards action should be applied. [MEF133.1 O16, O26]

6.12.3. Cancel Performance Monitoring Job Response

Entities used for providing a response to Cancel Performance Job request are presented in Figure 33. The Seller/Server responds with a `CancelPerformanceJob` type, which adds some attributes (like `id` or `state`) to the `CancelPerformanceJob_Create` that was used in the Buyer/Client request.

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

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

```
{
  "cancellationReason": "Cancel Performance Job sample",
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "creationDate": "2023-06-19T12:58:17.088Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "id": "aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R60] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R61] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDate`

[R62] The `id` **MUST** remain the same value for the life of the Cancel Performance Job.

In case Seller/Server cannot successfully validate the request, Cancel Performance Job process fails, which results in setting state to `declined` with a proper explanation in `cancellationDeniedReason`. This includes situation when:

- `id` does not allow to find a `PerformanceJob` that is to be cancelled in Seller/Server's system
- Performance Job is in the state that does not allow for cancellation.

6.12.4. Cancel Performance Monitoring Job State Machine

Figure 34 presents the Cancel Performance Monitoring Job state machine:

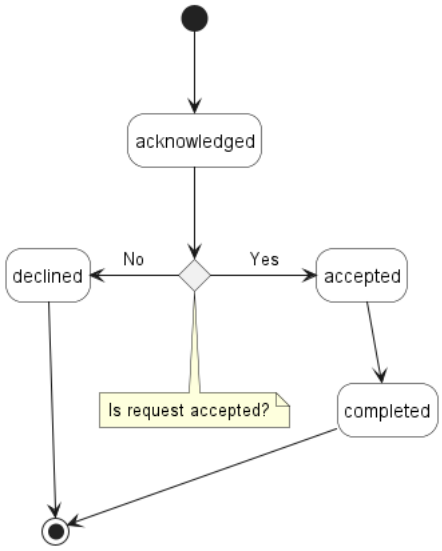


Figure 34. Cancel Performance Job State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `CancelPerformanceJob` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `declined` state if some issues are found. The `cancelPerformanceJob.cancellationDeniedReason` acts as a placeholder to provide a detailed description of what caused the problem. If validation is successful, `CancelPerformanceJob` moves to `accepted` state. When Cancel Performance Job process is finished, it moves to `completed` state. This causes `PerformanceJob` state to change to `cancelled`.

Description and mapping of the Cancel Performance Job States is the same as in table 10.

6.13. Use Case 13: Retrieve Cancel Performance Monitoring Job List

The Buyer/Client can retrieve a list of Cancel Performance Job objects by using a `GET /cancelPerformanceJob` operation with desired filtering criteria.

[O18] The Buyer/Client Retrieve List of Cancel Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDate.gt`
- `creationDate.lt`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/cancelPerformanceJob?
state=acknowledged&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Cancel Performance Job objects that are in the `acknowledged` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `CancelPerformanceJob_Find` objects matching the criteria. Details related to pagination are described in section 7.1.2.

[R63] The Seller **MUST** include following attributes (if set) in the `CancelPerformanceJob_Find` object in the response:

- `id`
- `performanceJobId`
- `state`

[R64] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 35 presents entities related to the use case.

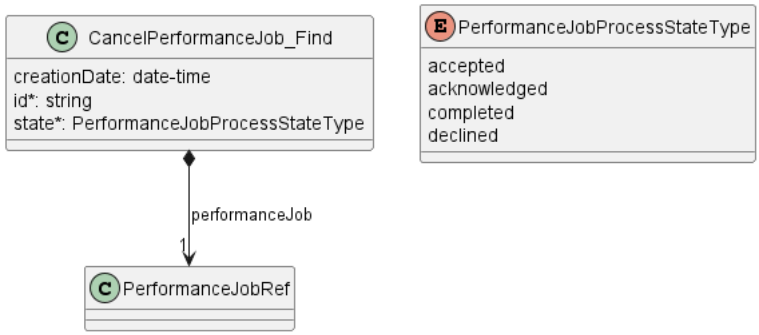


Figure 35. Use Case 13: Retrieve Cancel Performance Job List - Model

6.14. Use Case 14: Retrieve Cancel Performance Monitoring Job List by Identifier

The Buyer/Client can get detailed information about the Cancel Performance Job from the Seller/Server by using a `GET /cancelPerformanceJob/{{id}}` operation. The payload returned in the response is a full representation of Cancel Performance Job and includes all attributes the Buyer/Client has provided while sending a Cancel Performance Job create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Cancel Performance Job. Get List returns `CancelPerformanceJob_Find` object which is a subset of `CancelPerformanceJob` returned by Get by Identifier operation. A response to a get by id for a `CancelPerformanceJob` with `id=9c51d971-185d-403e-952f-2110f33a9628` would return exactly same response as presented in section 6.12.3.

[R65] In case `id` does not allow finding a `CancelPerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R66] The Seller/Server **MUST** include following attributes in the `CancelPerformanceJob` object in the response:

- `id`
- `performanceJob`
- `state`

[R67] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

6.15. Use Case 15: Suspend Performance Monitoring Job

Due to the need for releasing resources on the SOF side, the suspend operation associated with Performance Monitoring Job may exhibit prolonged duration. Consequently, this operation is implemented through a separate lifecycle process.

When the Performance Job is suspended, it does not generate Performance Reports.

6.15.1. Interaction flow

The flow of this use case is shown in Figure 36.

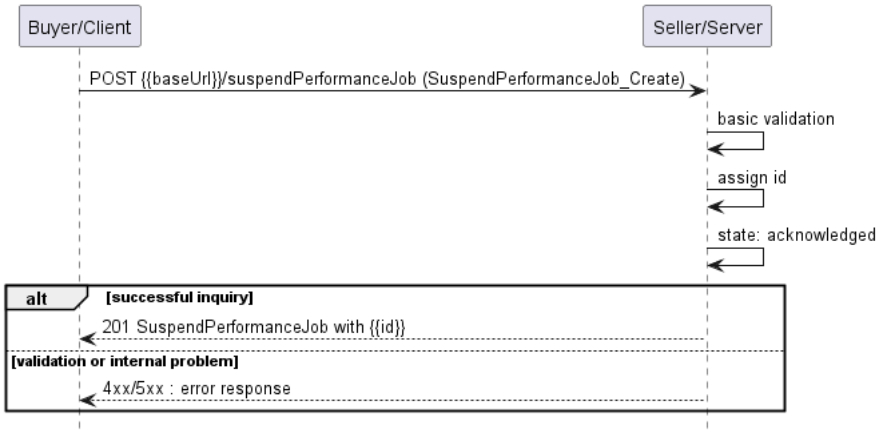


Figure 36. Use Case 15 - Suspend Performance Monitoring Job create request flow

The Buyer/Client sends a request with a `SuspendPerformanceJob_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `SuspendPerformanceJob` type in the response body, with a `state` set to `acknowledged`. Further processing is performed by Seller/Server which will in case of success update Performance Monitoring Job. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `SuspendPerformanceJob`. The two patterns are presented in the following diagrams.

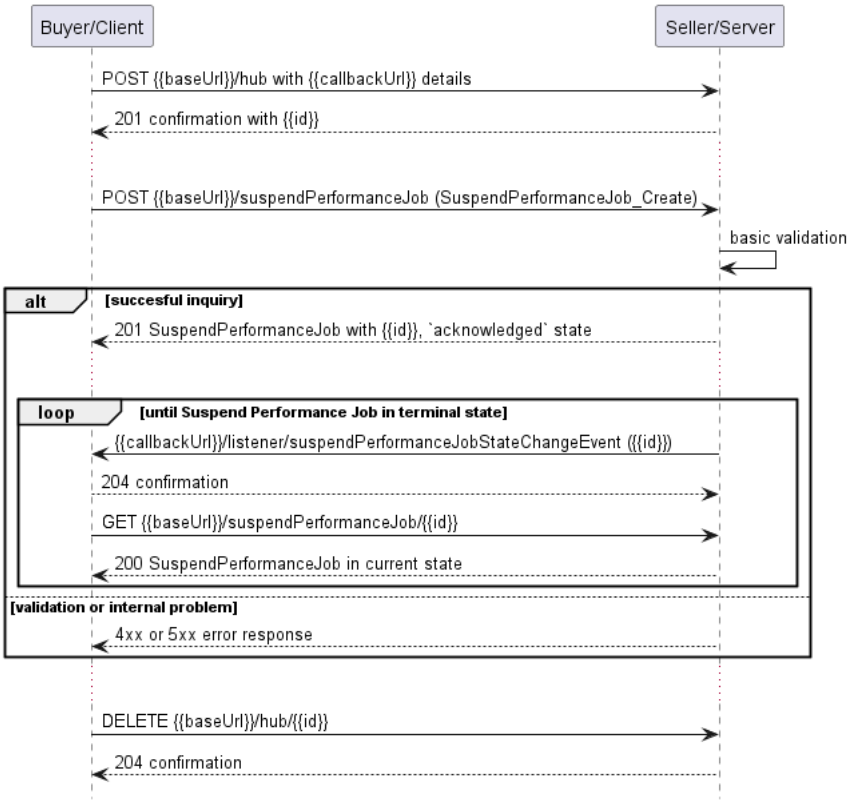


Figure 37. Suspend Performance Job progress tracking - Notifications

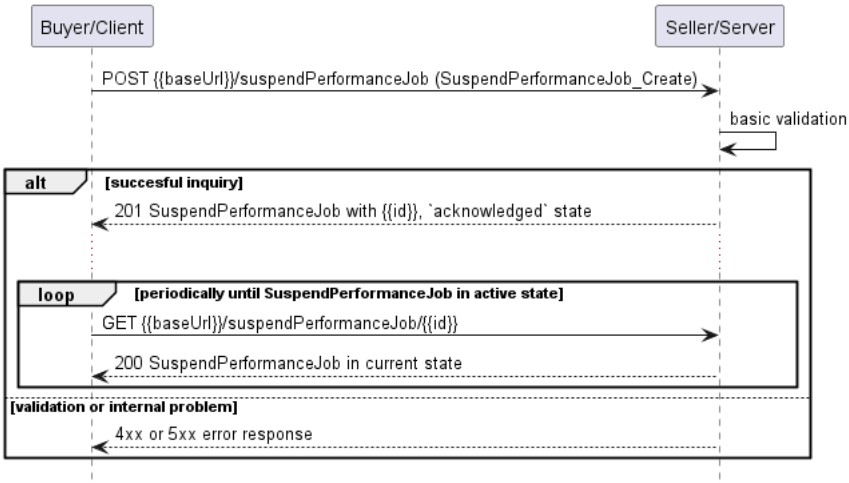


Figure 38. Suspend Performance Job progress tracking - Polling

Note: The Suspend Performance Job process is altering the state of the job itself. It is important to note that notifications resulting from changes in the state of the Performance Job are not represented in figures 37 and 38.

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

6.15.2. Suspend Performance Monitoring Job Request

Figure 39 presents the most important part of the data model used during the Suspend Performance Job request (`POST /suspendPerformanceJob`) and response. The model of the request message - `SuspendPerformanceJob_Create` is a subset of the `SuspendPerformanceJob` model and contains only attributes that can (or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information.

Note: `SuspendPerformanceJob_Create` is an entity used by the Buyer/Client to make a request. `SuspendPerformanceJob` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `SuspendPerformanceJob_Common` has been introduced (this class is not supposed to be used directly in the exchange).

The `performanceJobRef` section of `SuspendPerformanceJob_Create` is used to specify which Performance Job object is a subject of the suspension process (relationship by reference using `id` of the Job).

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

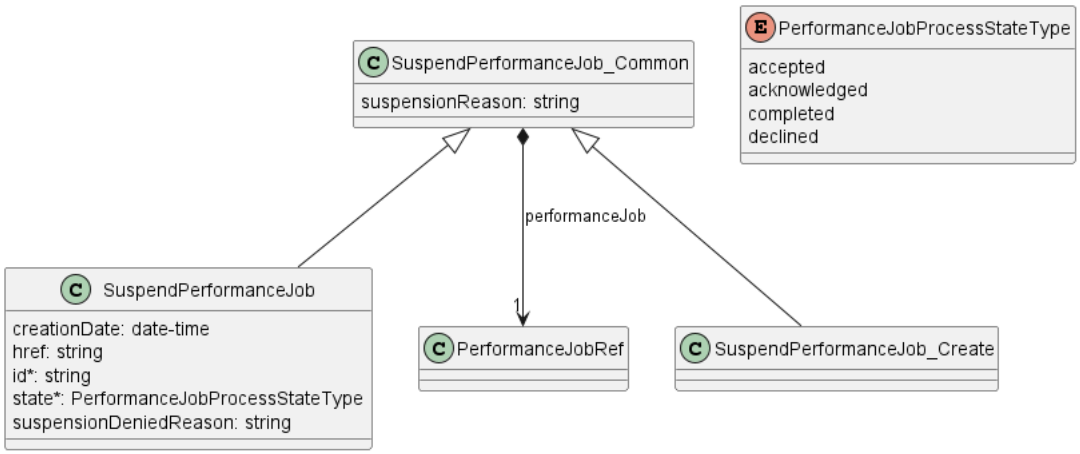


Figure 39. Suspend Performance Job Key Entities

To send a Suspend Performance Job request the Buyer/Client uses the `suspendPerformanceJob` operation from the API: `POST /suspendPerformanceJob`.

The example below shows a request to create a suspension process for `PerformanceJob` that was created in section 6.6.2.

```
{
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "suspensionReason": "Suspend Performance Job sample"
}
```

[R68] The Buyer/Client Suspend Performance Job request **MUST** include the following attributes: [MEF133.1 R59]

- `performanceJob`

[R69] The Performance Job **MUST** be in the In-Progress state to be suspended. [MEF133.1 R60]

[O19] In case Performance Job is running e.g., once a day for a short period of time, it may be difficult to change its state. If action arrives when Performance Job is running, it is recommended to run until the end and only afterwards action should be applied. [MEF133.1 O16, O26]

6.15.3. Suspend Performance Monitoring Job Response

Entities used for providing a response to Suspend Performance Job request are presented in Figure 39. The Seller/Server responds with a `SuspendPerformanceJob` type, which adds some attributes (like `id` or `state`) to the `SuspendPerformanceJob_Create` that was used in the Buyer/Client request.

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

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

```
{
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "suspensionReason": "Suspend Performance Job sample",
  "creationDate": "2023-06-19T12:58:17.088Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "id": "aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R70] The Seller/Server's response to the Buyer/Client's Suspend Performance Job request **MUST** indicate if the request is Accepted or Declined. [MEF133.1 R61]

[R71] If the Seller/Server accepts the Buyer/Client's Suspend Performance Job request, the Performance Job **MUST** be suspended and move to the Suspended state. [MEF133.1 R62]

[R72] If the Seller/Server declines the Buyer/Client's Suspend Performance Job request, the Performance Job **MUST NOT** be suspended. [MEF133.1 R63]

[R73] If the Seller/Server declines the Buyer/Client's Suspend Performance Job request, they **MUST** provide a reason why the request was declined. [MEF133.1 R64]

[R74] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R75] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDate`

[R76] The `id` **MUST** remain the same value for the life of the Performance Job.

In case Seller/Server cannot successfully validate the request, Suspend Performance Job process fails, which results in setting state to `declined` with a proper explanation in `suspensionDeniedReason`. This includes situation when:

- `id` does not allow to find a `PerformanceJob` that is to be suspended in Seller/Server's system
- Performance Job is in the state that does not allow for suspension.

6.15.4. Suspend Performance Monitoring Job State Machine

Figure 40 presents the Suspend Performance Monitoring Job state machine:

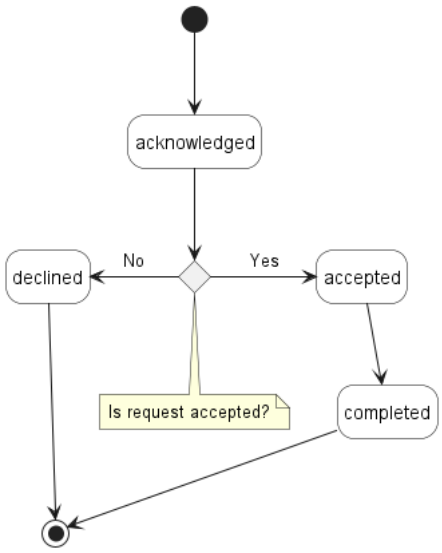


Figure 40. Suspend Performance Job State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `SuspendPerformanceJob` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `declined` state if some issues are found. The `suspendPerformanceJob.suspensionDeniedReason` acts as a placeholder to provide a detailed description of what caused the problem. If validation is successful, `SuspendPerformanceJob` moves to `accepted` state. When Suspend Performance Job process is finished, it moves to `completed` state. This causes `PerformanceJob` state to change to `suspended`.

Description and mapping of the Suspend Performance Job States is the same as in table 10.

6.16. Use Case 16: Retrieve Suspend Performance Monitoring Job List

The Buyer/Client can retrieve a list of Suspend Performance Job objects by using a `GET /suspendPerformanceJob` operation with desired filtering criteria.

[O20] The Buyer/Client Retrieve List of Suspend Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDate.gt`
- `creationDate.lt`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/suspendPerformanceJob?
state=acknowledged&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Suspend Performance Job objects that are in the `acknowledged` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `SuspendPerformanceJob_Find` objects matching the criteria. Details related to pagination are described in section 7.1.2.

[R77] The Seller **MUST** include following attributes (if set) in the `SuspendPerformanceJob_Find` object in the response:

- `id`
- `performanceJobId`
- `state`

[R78] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 41 presents entities related to the use case.

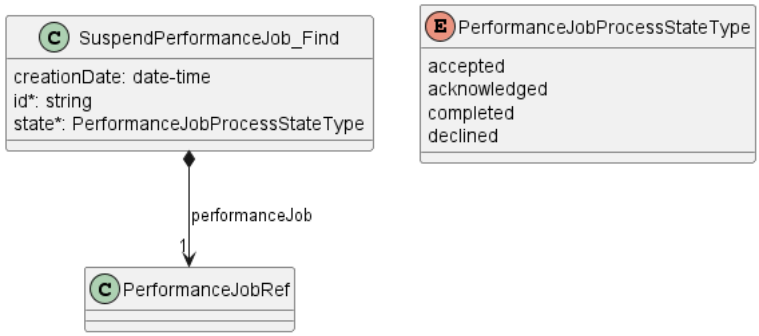


Figure 41. Use Case 16: Retrieve Suspend Performance Job List - Model

6.17. Use Case 17: Retrieve Suspend Performance Monitoring Job List by Identifier

The Buyer/Client can get detailed information about the Suspend Performance Job from the Seller/Server by using a `GET /suspendPerformanceJob/{{id}}` operation. The payload returned in the response is a full representation of Suspend Performance Job and includes all attributes the Buyer/Client has provided while sending a Suspend Performance Job create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Suspend Performance Job. Get List returns `SuspendPerformanceJob_Find` object which is a subset of `SuspendPerformanceJob` returned by Get by Identifier operation. A response to a get by identifier

for a `SuspendPerformanceJob` with `id=9c51d971-185d-403e-952f-2110f33a9628` would return exactly same response as presented in section 6.15.3.

[R79] In case `id` does not allow finding a `SuspendPerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R80] The Seller/Server **MUST** include following attributes in the `SuspendPerformanceJob` object in the response:

- `id`
- `performanceJob`
- `state`

[R81] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

6.18. Use Case 18: Resume Performance Monitoring Job

Due to the need for reserving resources on the SOF side, the resume operation associated with Performance Monitoring Job may exhibit prolonged duration. Consequently, this operation is implemented through a separate lifecycle process.

6.18.1. Interaction flow

The flow of this use case is shown in Figure 42.

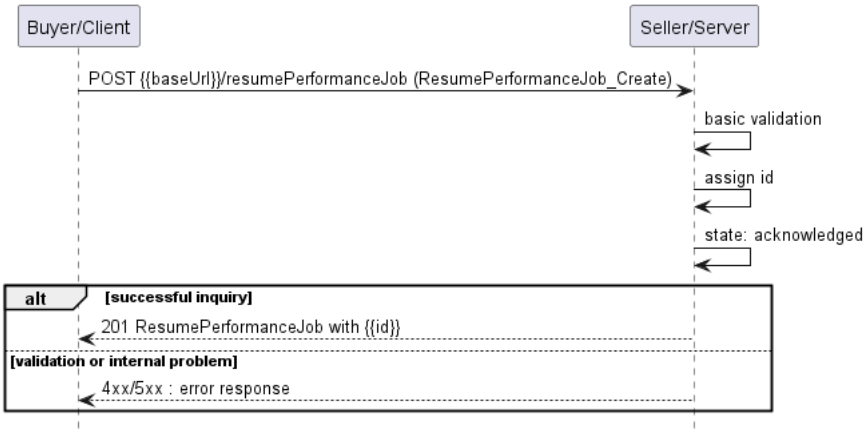


Figure 42. Use Case 18 - Resume Performance Monitoring Job create request flow

The Buyer/Client sends a request with a `ResumePerformanceJob_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `ResumePerformanceJob` type in the response body, with a `state` set to `acknowledged`. Further processing is performed by Seller/Server which will in case of success update Performance Monitoring Job. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `ResumePerformanceJob`. The two patterns are presented in the following diagrams.

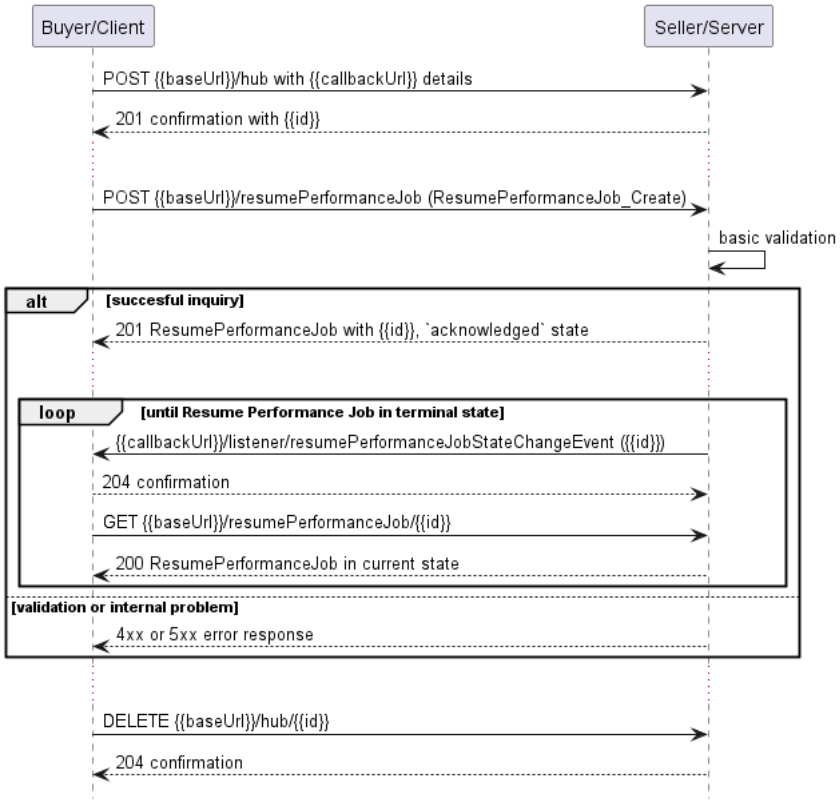


Figure 43. Resume Performance Job progress tracking - Notifications

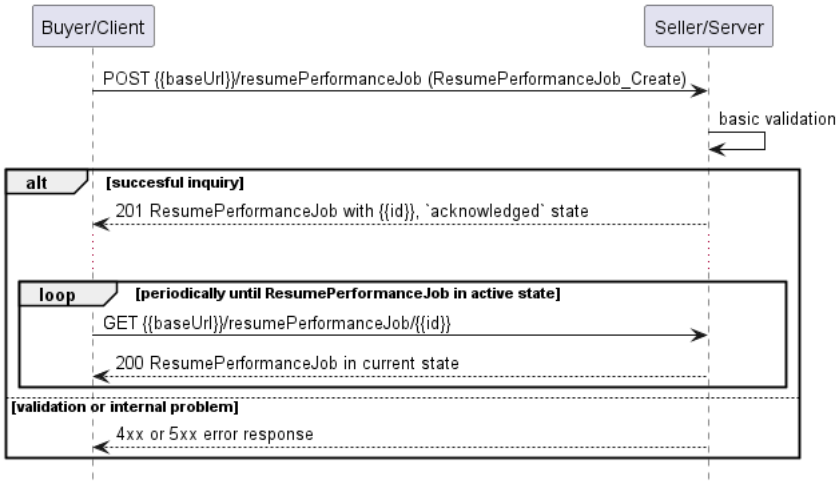


Figure 44. Resume Performance Job progress tracking - Polling

Note: The Resume Performance Job process is altering the state of the job itself. It is important to note that notifications resulting from changes in the state of the Performance Job are not represented in figures 43 and 44.

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

6.18.2. Resume Performance Monitoring Job Request

Figure 45 presents the most important part of the data model used during the Resume Performance Job request (`POST /resumePerformanceJob`) and response. The model of the request message - `ResumePerformanceJob_Create` is a subset of the `ResumePerformanceJob` model and contains only attributes that can (or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information.

Note: `ResumePerformanceJob_Create` is an entity used by the Buyer/Client to make a request. `ResumePerformanceJob` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `ResumePerformanceJob_Common` has been introduced (this class is not supposed to be used directly in the exchange).

The `performanceJob` section of `ResumePerformanceJob_Common` is used to specify which Performance Job object is a subject of the resume process (relationship by reference using `id` of the Job).

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

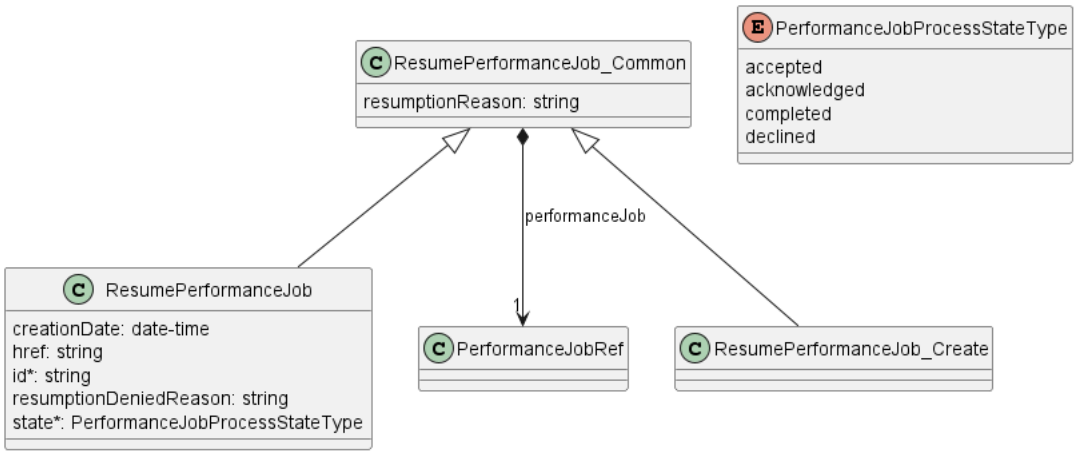


Figure 45. Resume Performance Job Key Entities

To send a Resume Performance Job request the Buyer/Client uses the `resumePerformanceJob` operation from the API: `POST /resumePerformanceJob`.

The example below shows a request to create a resumption process for `PerformanceJob` that was created in section 6.6.2.

```
{
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "resumptionReason": "Resume Performance Job sample"
}
```

[R82] The Buyer/Client Resume Performance Job request **MUST** include the following attributes: [MEF133.1 R65]

- `performanceJob`

[R83] The Performance Job **MUST** be in the Suspended state in order to be resumed. [MEF133.1 R66]

6.18.3. Resume Performance Monitoring Job Response

Entities used for providing a response to Resume Performance Job request are presented in Figure 45. The Seller/Server responds with a `ResumePerformanceJob` type, which adds some attributes (like `id` or `state`) to the `ResumePerformanceJob_Create` that was used in the Buyer/Client request.

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

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

```
{
  "performanceJob": {
    "@type": "PerformanceJobRef",
    "href": "{{baseUrl}}/performanceMonitoring/v1/755e55e2-72b0-4e3b-af00-693e3beac691",
    "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "resumptionReason": "Resume Performance Job sample",
  "creationDate": "2023-06-19T12:58:17.088Z", << added by SOF >>
  "href": "{{baseUrl}}/performanceMonitoring/v1/aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "id": "aea2769a-23f3-4ddc-b095-542a63b12481", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R84] The Seller/Server's response to the Buyer/Client's Resume Performance Job request **MUST** indicate if the request is Accepted or Declined. [MEF133.1 R67]

[R85] If the Seller/Server accepts the Buyer/Client's Resume Performance Job request, the Performance Job **MUST** be resumed and return to the In-Progress state. [MEF133.1 R68]

[R86] If the Seller/Server declines the Buyer/Client's Resume Performance Job request, the Performance Job **MUST NOT** be resumed. [MEF133.1 R69]

[R87] If the Seller/Server declines the Buyer/Client's Resume Performance Job request, they **MUST** provide a reason why the request was declined. [MEF133.1 R70]

[R88] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R89] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDate`

[R90] The `id` **MUST** remain the same value for the life of the Performance Job.

In case Seller/Server cannot successfully validate the request, Resume Performance Job process fails, which results in setting state to `declined` with a proper explanation in `resumptionDeniedReason`. This includes situation when:

- `id` does not allow to find a `PerformanceJob` that is to be resumed in Seller/Server's system
- Performance Job is in the state that does not allow for resumption.

6.18.4. Resume Performance Monitoring Job State Machine

Figure 46 presents the Resume Performance Monitoring Job state machine:

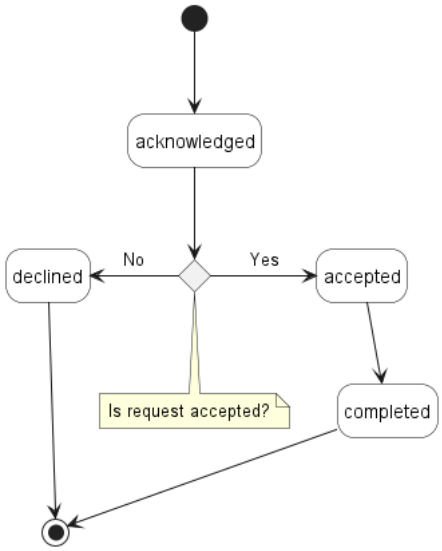


Figure 46. Resume Performance Job State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `ResumePerformanceJob` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `declined` state if some issues are found. The `resumePerformanceJob.resumptionDeniedReason` acts as a placeholder to provide a detailed description of what caused the problem. If validation is successful, `ResumePerformanceJob` moves to `accepted` state. When Resume Performance Job process is finished, it moves to `completed` state. This causes `PerformanceJob` state to change to `in-progress`.

Description and mapping of the Resume Performance Job States is the same as in table 10.

6.19. Use Case 19: Retrieve Resume Performance Monitoring Job List

The Buyer/Client can retrieve a list of Resume Performance Job objects by using a `GET /resumePerformanceJob` operation with desired filtering criteria.

[O21] The Buyer/Client Retrieve List of Resume Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDate.gt`
- `creationDate.lt`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/resumePerformanceJob?
state=acknowledged&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Resume Performance Job objects that are in the `acknowledged` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `ResumePerformanceJob_Find` objects matching the criteria. Details related to pagination are described in section 7.1.2.

[R91] The Seller **MUST** include following attributes (if set) in the `ResumePerformanceJob_Find` object in the response:

- `id`
- `performanceJobId`
- `state`

[R92] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 47 presents entities related to the use case.

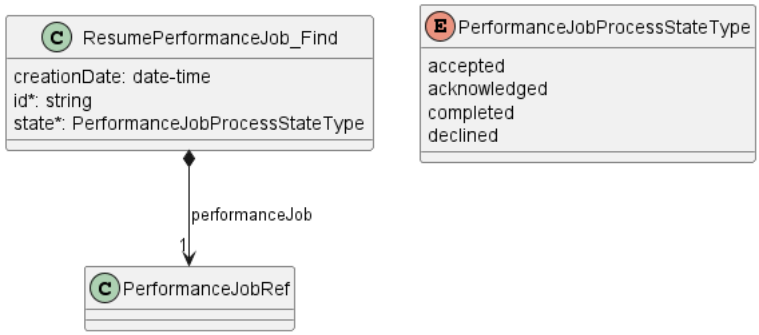


Figure 47. Use Case 16: Retrieve Resume Performance Job List - Model

6.20. Use Case 20: Retrieve Resume Performance Monitoring Job List by Identifier

The Buyer/Client can get detailed information about the Resume Performance Job from the Seller/Server by using a `GET /resumePerformanceJob/{{id}}` operation. The payload returned in the response is a full representation of Resume Performance Job and includes all attributes the Buyer/Client has provided while sending a Resume Performance Job create request, together with additional attributes set by Seller/Server.

Get List and Get by Identifier operations return different representations of Resume Performance Job. Get List returns `ResumePerformanceJob_Find` object which is a subset of `ResumePerformanceJob` returned by Get by Identifier operation. A response to a get by identifier for a `ResumePerformanceJob` with `id=9c51d971-185d-403e-952f-2110f33a9628` would return exactly same response as presented in section 6.15.3.

[R93] In case `id` does not allow finding a `ResumePerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R94] The Seller/Server **MUST** include following attributes in the `ResumePerformanceJob` object in the response:

- `id`
- `performanceJob`
- `state`

[R95] The Seller **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

6.21. Use Case 21: Create Performance Monitoring Job Complex Query

The `PerformanceJob` defines complex structures with multiple levels of nesting, such as `scheduleDefinition`. To facilitate filtering based on these structures, the API provides an additional endpoint `POST /performanceJobComplexQuery`. This endpoint allows filtering by values defined by the `PerformanceJob` and `PerformanceProfile` types and returns a list of `PerformanceJob` objects that match the specified filters.

6.21.1. Create Performance Monitoring Job Complex Query Request

Figure 48 depicts the key components of the data model utilized in the Create Performance Job Complex Query request (`POST /performanceJobComplexQuery`) and its corresponding response. The request message model, `PerformanceJobComplexQuery_Create`, is a subset of the `PerformanceJobComplexQuery` model and includes only attributes that can or must be specified by the Buyer/Client,

representing filtering options. In response, the Seller/Server provides a list of `PerformanceJobComplexQuery` entities that contain the matched `PerformanceJob` objects.

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

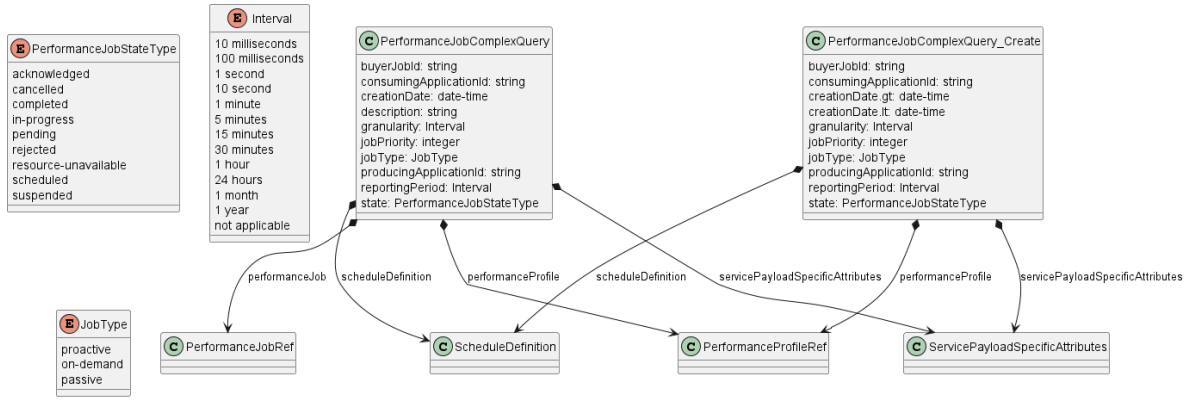


Figure 48. Performance Job Complex Query Key Entities

To send a request the Buyer/Client uses the `createPerformanceJobComplexQuery` operation from the API. The snippet below presents an example of Create Performance Job Complex Query request. It filters for `PerformanceJob` objects that:

- have `consumingApplicationId` set to `CUS`
- are based on the `performanceProfile` with `id=8df0981a-0949-11ee-be56-0242ac120002`
- run on a schedule with recurring frequency set to 1 hour
- are in `scheduled` state

`Performance Job Complex Query` **Create Request**

```
{
  "consumingApplicationId": "CUS",
  "performanceProfile": {
    "@type": "PerformanceProfileRef",
    "id": "8df0981a-0949-11ee-be56-0242ac120002"
  },
  "scheduleDefinition": {
    "recurringFrequency": {
      "recurringFrequencyValue": 1,
      "recurringFrequencyUnits": "HOURS"
    }
  },
  "state": "scheduled"
}
```

6.21.2. Create Performance Monitoring Job Complex Query Response

Entities used for providing a response to Create Performance Job Complex Query request are presented in Figure 48. The Seller/Server responds with a list of `PerformanceJobComplexQuery` objects, which represent matched Performance Jobs.

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

The following snippet presents the Seller/Server response.

`Performance Job Complex Query` **Create Response**


```
[
  {
    "buyerJobId": "TestJob12345",
    "consumingApplicationId": "CUS",
    "creationDate": "2023-06-01T08:02:01.370Z",
    "description": "Exemplary Create Performance Job request",
    "performanceJob": {
      "@type": "PerformanceJobRef",
      "id": "755e55e2-72b0-4e3b-af00-693e3beac691"
    },
    "performanceProfile": {
      "@type": "PerformanceProfileRef",
      "id": "8df0981a-0949-11ee-be56-0242ac120002"
    },
    "producingApplicationId": "SOF",
    "scheduleDefinition": {
      "recurringFrequency": {
        "recurringFrequencyValue": 1,
        "recurringFrequencyUnits": "HOURS"
      },
      "scheduleDefinitionStartTime": "2023-06-01T08:02:01.370Z"
    },
    "servicePayloadSpecificAttributes": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "interface": {
        "ipvcEndpoint": [
          "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
          "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
        ],
        "name": "slsRpPairTest1",
        "description": "Exemplary performance monitoring service pair",
        "cloudService": true
      }
    },
    "state": "scheduled"
  }
]
```


6.22. Use Case 22: Create Performance Measurement Report

The execution of all types of Performance Monitoring Jobs results in the generation of Performance Measurement Reports, which deliver comprehensive performance collections to the Buyer/Client. In certain scenarios, performance data can be collected without the need for prior provisioning of a Performance Job. This occurs under the following conditions:

- When the Service Level Specification (SLS) is included in the Service Order request.
- When passive statistics are automatically generated by the server.
- When the client retrieves historical data that is already available on the server.

6.22.1. Interaction flow

The flow of this use case is illustrated in Figure 49. A Performance Report can be generated either as an outcome of processing a Performance Job or by executing a Create Performance

Report request. The latter option is particularly useful for generating ad-hoc reports based on existing data. Both of these options are depicted in the figure.

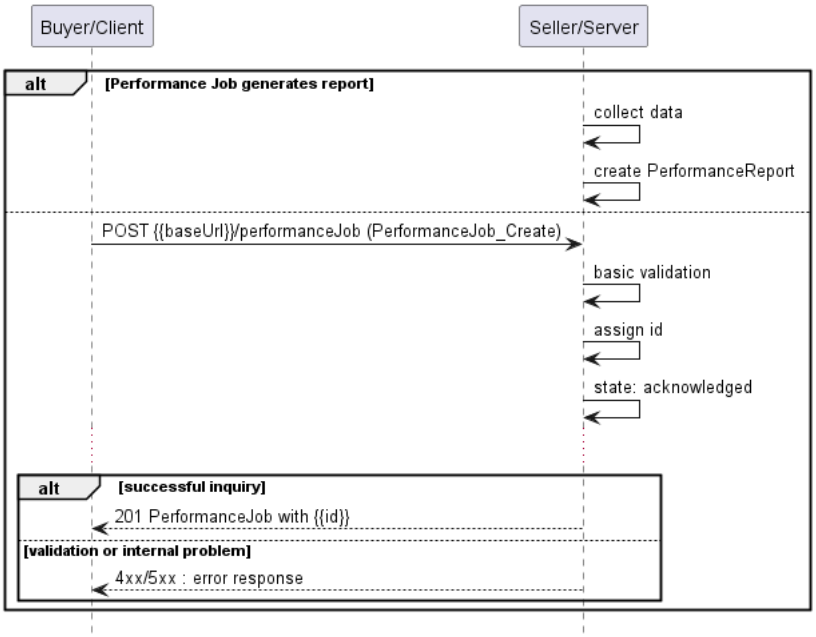


Figure 49. Use Case 22 - Performance Monitoring Report create request flow

In case of ad-hoc report creation, the Buyer/Client sends a request with a `PerformanceReport_Create` type in the body. The Seller/Server performs request validation, assigns an `id`, and returns `PerformanceReport` type in the response body, with a `state` set to `acknowledged`. From this point, the Performance Report is ready for further processing. The Buyer/Client can track the progress of the process either by subscribing for notifications or by periodically polling the `PerformanceReport`. The two patterns are presented in the following diagrams.

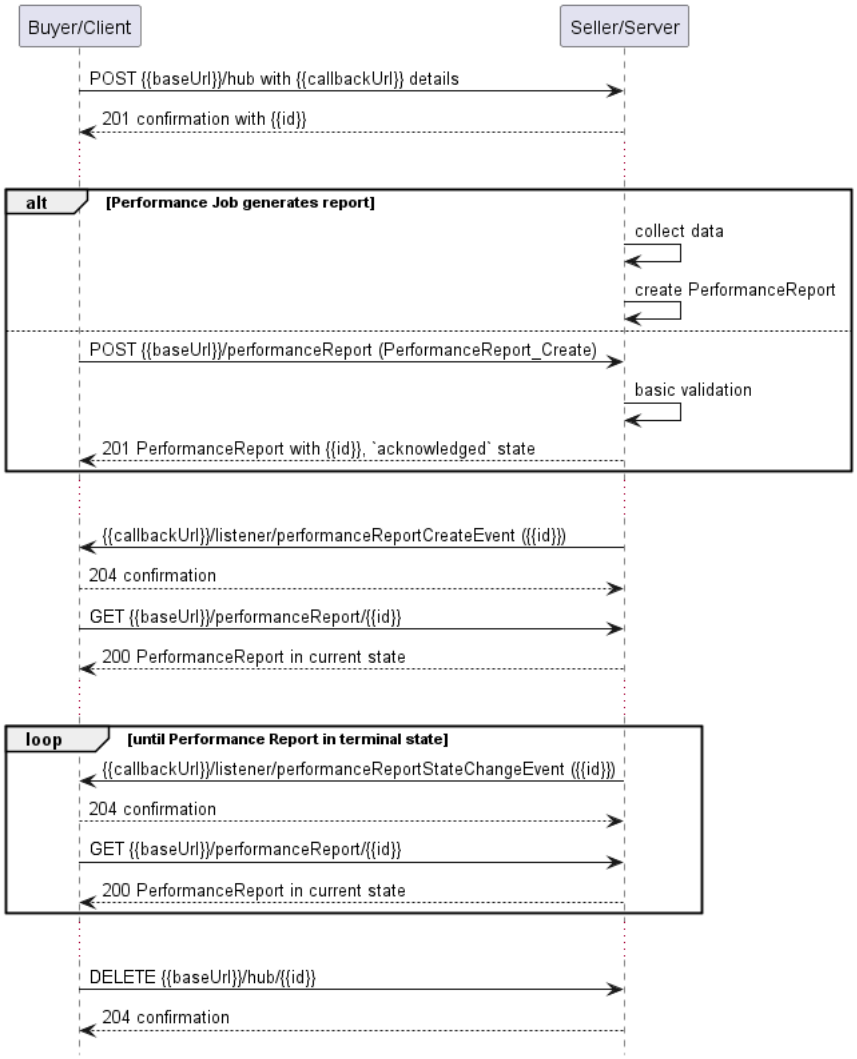


Figure 50. Performance Job progress tracking - Notifications

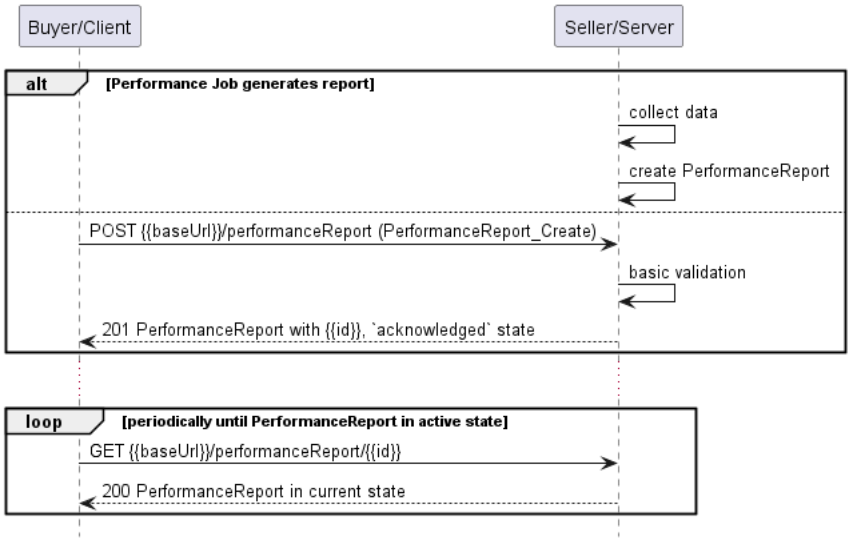


Figure 51. Performance Job progress tracking - Polling

Note: To provide clarity, the figures illustrate only successful scenarios, omitting any error or failure conditions.

Note: In the case of a Performance Report created by a Performance Job, the Buyer/Client can obtain the `id` of the `PerformanceReport` object either through a notification or by utilizing the

Retrieve List operation with the `performanceJobId` filter. It is important to note that neither of these options are represented in the figure.

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

6.22.2. Create Performance Measurement Report Request

Figure 52 presents the most important part of the data model used during the Create Performance Report request (`POST /performanceReport`) and response. The model of the request message - `PerformanceReport_Create` is a subset of the `PerformanceReport` model and contains only attributes that can (or must) be set by the Buyer/Client. The Seller/Server (SOF) then enriches the entity in the response with additional information including collected measurements (content of the report).

Note: `PerformanceReport_Create` is an entity used by the Buyer/Client to make a request. `PerformanceReport` is an entity used by the Seller/Server to provide a response. The request entity has a subset of attributes of the response entity. Thus for visibility of these shared attributes `PerformanceReport_Common` has been introduced (this class is not supposed to be used directly in the exchange).

A `PerformanceReport_Create` defines reporting timeframe, measurement intervals, output format, and objectives of performance monitoring (in `servicePayloadSpecificAttributes` section). Part of the attributes are defined by `PerformanceJob` type. See chapter section 6.22.5 for more details.

Section `servicePayloadSpecificAttributes` of the create Performance Report request allows for the introduction of service-specific properties of performance monitoring as the API payload. The extension mechanism is described in detail in Section 5.3.

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

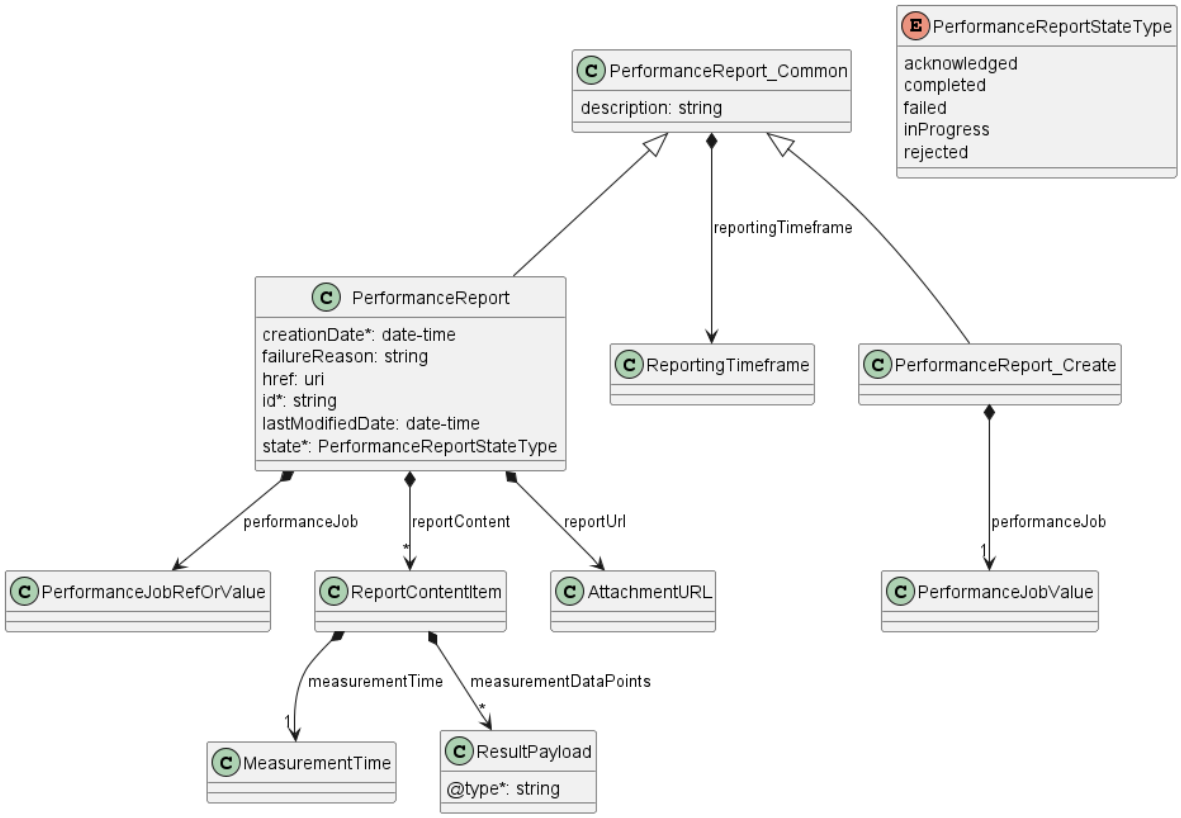


Figure 52. Performance Report Key Entities

To send a create Performance Report request the Buyer/Client uses the `createPerformanceReport` operation from the API: `POST /performancReport`. For clarity, some of create Performance Report payload's attributes might be omitted to improve examples' readability.

`Performance Measurement Report` Create Request

```
{
  "description": "Exemplary Create Performance Report request",
  "reportingTimeframe": {
    "reportingStartDate": "2023-06-01T00:00:00.00",
    "reportingEndDate": "2023-06-02T00:00:00.00"
  },
  "performanceJob": {
    "@type": "PerformanceJobValue",
    "consumingApplicationId": "CUS",
    "granularity": "1 hour",
    "outputFormat": "json",
    "producingApplicationId": "SOF",
    "resultFormat": "payload",
    "servicePayloadSpecificAttributes": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "interface": {
        "ipvcEndpoint": [
          "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
          "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
        ],
        "name": "slsRpPairTest1",
        "description": "Exemplary performance monitoring service pair",
        "cloudService": true
      }
    }
  }
}
```

[R96] The Buyer/Client Create Performance Report request **MUST** include the following attributes:

- `performanceJob`
- `performanceJob.@type`
- `performanceJob.outputFormat`
- `performanceJob.resultFormat`
- `performanceJob.servicePayloadSpecificAttributes`

6.22.3. Create Performance Measurement Report Response

Figure 52 showcases the entities involved in delivering a response to the Create Performance Report request. The Seller/Server provides a response of the `PerformanceReport` type, which introduces additional attributes (such as id, state, reportUrl for accessing the generated report, or reportContent for including measurement data in the response payload) to the original `PerformanceReport_Create` object used in the Buyer/Client request.

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

Depending on the `resultFormat` attribute,

Section `reportContent` of the Performance Report response allows for the introduction of service-specific results of performance monitoring as the API payload. The extension mechanism is described in detail in Section 5.3.

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

`Performance Measurement Report` Create Response

```
{
  "description": "Exemplary Create Performance Report request",
  "reportingTimeframe": {
    "reportingStartDate": "2023-06-01T00:00:00.00",
    "reportingEndDate": "2023-06-01T01:00:00.00"
  },
  "performanceJob": {
    "@type": "PerformanceJobValue",
    "consumingApplicationId": "CUS",
    "granularity": "1 hour",
    "outputFormat": "json",
    "producingApplicationId": "SOF",
    "resultFormat": "payload",
    "servicePayloadSpecificAttributes": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "interface": {
        "ipvcEndpoint": [
          "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
          "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
        ],
        "name": "slsRpPairTest1",
        "description": "Exemplary performance monitoring service pair",
        "cloudService": true
      }
    }
```

```
    },
    "reportContent": [
      {
        "measurementTime": {
          "measurementStartDate": "2023-06-01T00:00:00.00",
          "measurementEndDate": "2023-06-01T01:00:00.00"
        },
        "measurementDataPoints": [
          {
            "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all",
            "interface": {
              "ipvcEndpoint": [
                "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
                "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
              ],
              "name": "slsRpPairTest1",
              "description": "Exemplary performance monitoring service pair",
              "cloudService": true
            },
            "vlan": 100,
            "protocol": "IPV4",
            "packetsIn": 300,
            "charsIn": 30000,
            "packetsOut": 400,
            "charsOut": 40000,
            "utilizationIn": 60,
            "utilizationOut": 70,
            "peakUtilizationIn": 80,
            "peakUtilizationOut": 90
          }
        ]
      }
    ], << added by SOF >>
    "creationDate": "2023-06-01T08:02:01.370Z", << added by SOF >>
    "href": "{{baseUrl}}/performanceMonitoring/v1/8ae5f9f3-554f-4d93-8314-1630f171da54", << added by SOF >>
    "id": "8ae5f9f3-554f-4d93-8314-1630f171da54", << added by SOF >>
    "lastModifiedDate": "2023-06-01T08:02:01.370Z", << added by SOF >>
    "state": "completed" << added by SOF >>
  }
```

Attributes that are set by the Seller/Server in the response are marked with the `<< added by SOF >>` tag.

[R97] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by Buyer/Client in the request.

[R98] The Seller/Server **MUST** specify the following attributes in a response:

- `creationDate`
- `id`
- `state`

[R99] The `id` **MUST** remain the same value for the life of the Performance Report.

6.22.4. Performance Measurement Report State Machine

Figure 53 presents the Performance Report state machine:

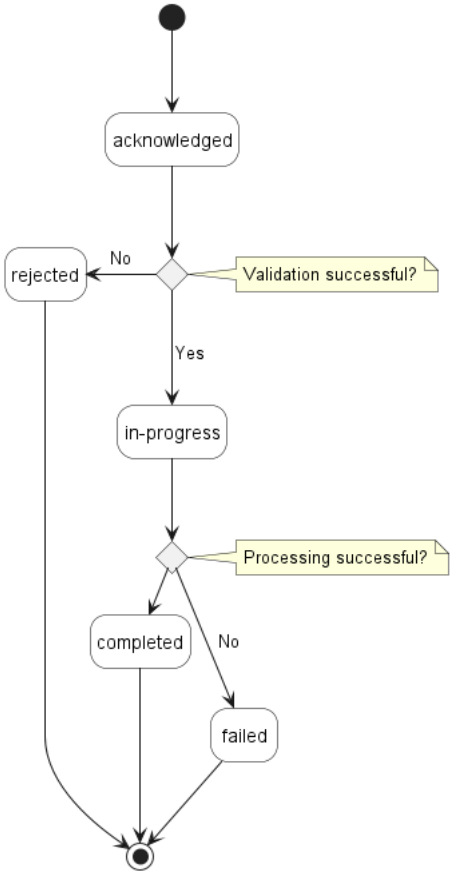


Figure 53. Performance Report State Machine

After receiving the request, the Seller/Server (SOF) performs basic checks of the message. If any problem is found an Error response is provided. If the validation passes a response is provided with `PerformanceReport` in `acknowledged` status. Next, the Seller/Server performs all the remaining business and time-consuming validations. At this point, an Error response cannot be provided anymore, so the profile moves to a `rejected` state if some issues are found. The `performanceReport.failureReason` acts as a placeholder to provide a detailed description of what caused the problem. `PerformanceReport` moves to `in-progress` state during which report content is prepared. Depending on the outcome of the processing, `PerformanceReport` moves to `completed` or `failed` state.

Table 11 presents the mapping between the API `status` names and the MEF W133.1 naming, together with statuses' description.

State	Description
acknowledged	A Performance Report request has been received by Seller/Server and has passed basic validations. Performance Report Identifier is assigned in the Acknowledged state. The report remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated, the Performance Report moves to the In-Progress state. If not all attributes are validated, the report moves to the Rejected state.
completed	A Performance Report is completed and results are available.

State	Description
failed	A Performance Report processing has failed.
inProgress	A Performance Report has successfully passed the validations checks and the report processing has started.
rejected	This state indicates that: - Invalid information is provided through the `PerformanceReport` request - The request fails to meet validation rules for `PerformanceReport` delivery (processing).

Table 11. Performance Report State Machine states

[R100] The Seller/Server **MUST** support all Performance Report statuses and their associated transitions as described in Figure 53 and Table 11.

6.22.5. Relationship to Performance Job

`PerformanceReport_Create` class used as a payload for `createPerformanceReport` operation refers to attributes defined by `PerformanceJob` type by directly assigning their values. These attributes are contained in `performanceJob` section. For this "by value" assignment, `@type` discriminator has to be set to `PerformanceJobValue`.

The `PerformanceReport` class, which represents the outcome of report processing, also includes a `performanceJob` section. However, this time it is defined as a `PerformanceJobRefOrValue`, enabling either a reference to a PerformanceJob object (when the Performance Report is generated by a Performance Job) or the listing of attribute values defined by the `PerformanceJob` type. Those two options are indicated by setting the @type (discriminator) attribute to either `PerformanceJobRef` or `PerformanceJobValue`.

Note: Defining attributes related to `PerformanceJob` in Create Performance Report request does not create new `PerformanceJob` object.

Figure 54 presents `PerformanceReport` and related entities that allow for referencing to Performance Job or providing corresponding attributes.

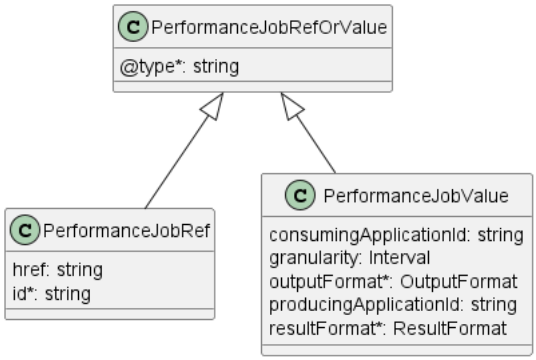


Figure 54. Relationship to Performance Job

6.23. Use Case 23: Retrieve Performance Measurement Report List

The Buyer/Client can retrieve a list of `PerformanceReport` by using a `GET /performanceReport` operation with desired filtering criteria.

[O22] The Buyer's/Client's Retrieve List of Performance Reports request **MAY** contain none or more of the following attributes as filter criteria: [MEF133.1 O17]

- `performanceJobId`
- `state`
- `creationDate.gt`
- `creationDate.lt`
- `reportingTimeframe.startDate.gt`
- `reportingTimeframe.startDate.lt`
- `reportingTimeframe.endDate.gt`
- `reportingTimeframe.endDate.lt`
- `granularity`
- `outputFormat`
- `resultFormat`
- `consumingApplicationId`
- `producingApplicationId`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/performanceReport?
state=completed&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Performance Report objects that are in the `completed` state. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `PerformanceReport_Find` objects matching the criteria. `PerformanceReport_Find` object is a subset of all Performance Report attributes. In particular, it does not contain the collected measurements. To get all details, the Buyer/Client has to query a specific `PerformanceReport` by its `id`. Details related to pagination are described in section 7.1.2

[R101] The Seller/Server **MUST** support the retrieval of a List of Performance Measurement Reports Use Case. [MEF133.1 R77, R94]

[R102] The Buyer/Client **MUST** support the retrieval of a List of Performance Measurement Reports Use Case. [MEF133.1 R78, R95]

[R103] The Seller/Server's response to the Buyer's/Client's retrieve List of Performance Measurement Reports **MUST** include the following attributes as applicable: [MEF133.1 R79, R96]

- `creationDate`
- `description`
- `id`
- `state`

[R104] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 55 presents entities related to the use case.

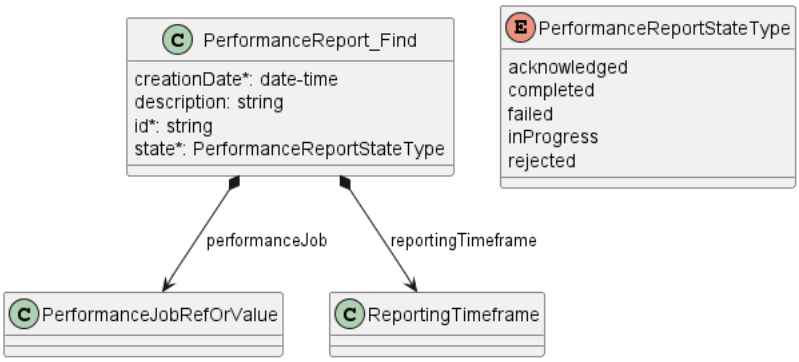


Figure 55. Use Case 23: Retrieve Performance Report List - Model

6.24. Use Case 24: Retrieve Performance Measurement Report by Report Identifier

The Buyer/Client can get detailed information about one or multiple Performance Reports from the Seller/Server by using a `GET /performanceReport/{{id}}` operation. The `{{id}}` parameter accepts an array of identifiers to support retrieving many reports with one request. The response payload provides a comprehensive representation of the Performance Report(s) and encompasses all attributes that the Buyer/Client has provided when submitting a Create Performance Report request. Alternatively, it includes attributes of the Performance Job that triggered the generation of the report, along with any additional attributes set by the Seller/Server.

Get List and Get by Identifier operations return different representations of Performance Report. Get List returns `PerformanceReport_Find` object which is a subset of `PerformanceReport` returned by Get by Identifier operation. A response to a get by identifier for a `PerformanceReport` with `id=8ae5f9f3-554f-4d93-8314-1630f171da54` would return exactly same response as presented in section 6.22.3. Specifically, the object returned by the Get by Identifier operation contains a collection of measurement results, either in the form of a URI of a generated file or directly within the returned `PerformanceReport` object. Measurement results are not returned by Get List operation.

[R105] The Seller/Server MUST support at least one of methods of retrieving results: [MEF133.1 R80, R97]:

- payload
- attachment

[O23] The Seller/Server **MAY** support multiple methods of retrieving results. [MEF133.1 O18, O21]

[R106] The Retrieve Results request **MUST** include the following attributes: [MEF133.1 R81, R82, R98, R99]

- list of `id`
- `fileTransferData` in case of retrieving results in attachment
- `outputFormat`

[R107] The Seller/Server **MUST** include following attributes in the `PerformanceReport` object in the response:

- `creationDate`
- `id`

[R108] The Seller/Server **MUST** provide all remaining optional attributes if they were previously set by the Buyer or the Seller.

[R109] The results regardless of the format MUST contain the Performance Metric results as specified with Performance Job request. [MEF133.1 R84]

[R110] In case `id` does not allow finding a `PerformanceReport` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R111] The Seller/Server **MUST** provide the specified result in the API payload. [MEF133.1 R101]

[R112] The Seller/Server **MUST** provide the specified results as an attachment. [MEF133.1 R102]

[R113] The Seller/Server **MUST** provide the specified results as an FTP'd file in JSON, AVRO, CSV, XML format. [MEF133.1 R103]

6.25. Use Case 25: Create Performance Measurement Report Complex Query

The `PerformanceReport` defines complex structures with multiple levels of nesting, such as `servicePayloadSpecificAttributes`. To facilitate filtering based on these structures, the API provides an additional endpoint `POST /performanceReportComplexQuery`. This endpoint allows filtering by values defined by the `PerformanceReport` and `PerformanceJob` types and returns a list of `PerformanceReport` objects that match the specified filters.

6.25.1. Create Performance Measurement Report Complex Query Request

Figure 56 depicts the key components of the data model utilized in the Create Performance Report Complex Query request (`POST /performanceReportComplexQuery`) and its corresponding response. The request message model, `PerformanceReportComplexQuery_Create`, is a subset of the `PerformanceReportComplexQuery` model and includes only attributes that can or must be specified by the Buyer/Client, representing filtering options. In response, the Seller/Server provides a list of `PerformanceReportComplexQuery` entities that contain the matched `PerformanceReport` objects.

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

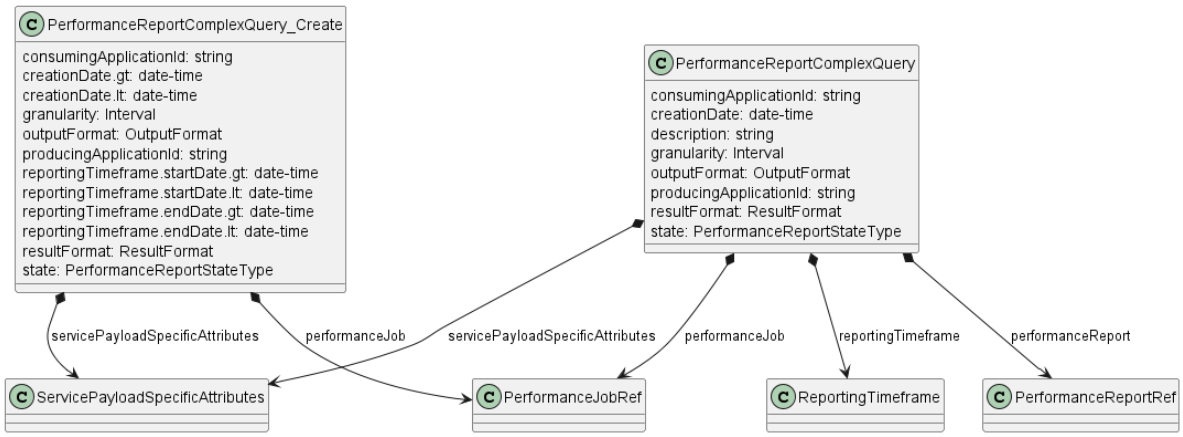


Figure 56. Performance Report Complex Query Key Entities

To send a request the Buyer/Client uses the `createPerformanceReportComplexQuery` operation from the API. The snippet below presents an example of Create Performance Report Complex Query request. It filters for `PerformanceReport` objects that:

- have `consumingApplicationId` set to `CUS`
- were created between 2023-06-01 08:00:00 and 2023-06-01 09:00:00
- `outputFormat` is JSON
- relate to specified IPVC endpoint

`Performance Report Complex Query` **Create Request**

```
{
  "consumingApplicationId": "CUS",
  "creationDate.gt": "2023-06-01T08:00:00.000Z",
  "creationDate.lt": "2023-06-01T09:00:00.000Z",
  "outputFormat": "json",
  "servicePayloadSpecificAttributes": {
    "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
    "interface": {
      "ipvcEndpoint": [
        "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
        "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
      ],
      "name": "slsRpPairTest1",
      "description": "Exemplary performance monitoring service pair",
      "cloudService": true
    }
  },
```

```
    "state": "completed"
  }
```

6.25.2. Create Performance Monitoring Report Complex Query Response

Entities used for providing a response to Create Performance Report Complex Query request are presented in Figure 56. The Seller/Server responds with a list of `PerformanceReportComplexQuery` objects, which represent matched Performance Reports.

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

The following snippet presents the Seller/Server response.

`Performance Report Complex Query` Create Response

```
[
  {
    "consumingApplicationId": "CUS",
    "creationDate": "2023-06-01T08:02:01.370Z",
    "description": "Exemplary Create Performance Report request",,
    "granularity": "1 hour",
    "outputFormat": "json",
    "performanceReport": {
      "id": "8ae5f9f3-554f-4d93-8314-1630f171da54"
    },
    "producingApplicationId": "SOF",
    "reportingTimeframe": {
      "reportingStartDate": "2023-06-01T00:00:00.00",
      "reportingEndDate": "2023-06-01T01:00:00.00"
    },
    "resultFormat": "payload",
    "servicePayloadSpecificAttributes": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "interface": {
        "ipvcEndpoint": [
          "6e4e338a-8105-481e-8bf6-b3ca768a4b89",
          "38bfa4c6-48a3-46e9-8746-bcba59f3cbc4"
        ],
        "name": "slsRpPairTest1",
        "description": "Exemplary performance monitoring service pair",
        "cloudService": true
      }
    },
    "state": "completed"
  }
]
```

6.26. Use Case 26: Retrieve Tracking Record List

Tracking Records allow the tracking of actions performed on main entities described in this document:

- Performance Monitoring Profile
- Performance Monitoring Job
- Performance Monitoring Report

Tracking Records store information regarding the timing and nature of actions performed on a specific object. The association with Performance Monitoring entities can be established through the `relatedObjectId` attribute of the `TrackingRecord` type.

The Buyer/Client can retrieve a list of `TrackingRecord` by using a `GET /trackingRecord` operation with desired filtering criteria.

[O24] The Buyer/Client Retrieve List of Tracking Record request **MAY** contain none or more of the following attributes:

- `relatedObjectId`
- `creationDate.gt`
- `creationDate.lt`
- `user`

```
https://serverRoot/mefApi/legato/performanceMonitoring/v1/trackingRecord?relatedObjectId=755e55e2-72b0-4e3b-af00-693e3beac691&limit=10&offset=0
```

The example above shows a Buyer/Client's request to get all Tracking Record objects that are related to object with `id=755e55e2-72b0-4e3b-af00-693e3beac691`. Additionally, the Buyer/Client asks only for a first (`offset=0`) pack of 10 results (`limit=10`) to be returned. The correct response (HTTP code `200`) in the response body contains a list of `TrackingRecord_Find` objects matching the criteria. To get all details, the Buyer/Client has to query a specific `TrackingRecord` by its `id`. Details related to pagination are described in section 7.1.2

[R114] The Seller/Server **MUST** include following attributes (if set) in the `TrackingRecord_Find` object in the response:

- `creationDate`
- `relatedObjectId`

[R115] Optionally The Seller/Server **MAY** return :

- `description`
- `user`

[R116] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Figure 57 presents main Tracking Record entities.

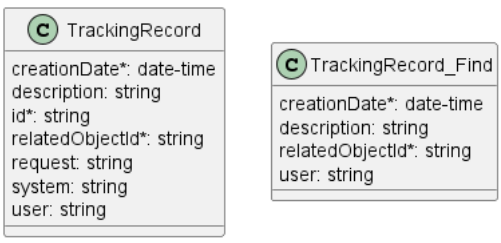


Figure 57. Tracking Record Model

6.27. Use Case 27: Retrieve Tracking Record List by Identifier

The Buyer/Client can get detailed information about the Tracking Record from the Seller/Server by using a `GET /trackingRecord/{{id}}` operation. The payload returned in the response is a full representation of Tracking Record.

Get List and Get by Identifier operations return different representations of Tracking Record. Get List returns `TrackingRecord_Find` object which is a subset of `TrackingRecord` returned by Get by Identifier operation.

[R117] In case `id` does not allow finding a `TrackingRecord` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R118] The Seller/Server **MUST** include following attributes in the `TrackingRecord` object in the response:

- `creationDate`
- `id`
- `relatedObjectId`

The full list of attributes of Tracking Record is available in Section 7 and in the API specification which is an integral part of this standard.

6.28. Use case 28: Register for Notifications

The Buyer/Client can track the lifecycle of the Performance Monitoring objects by subscribing for notifications. Exemplary use case for exchanging notifications is presented in Figure 58.

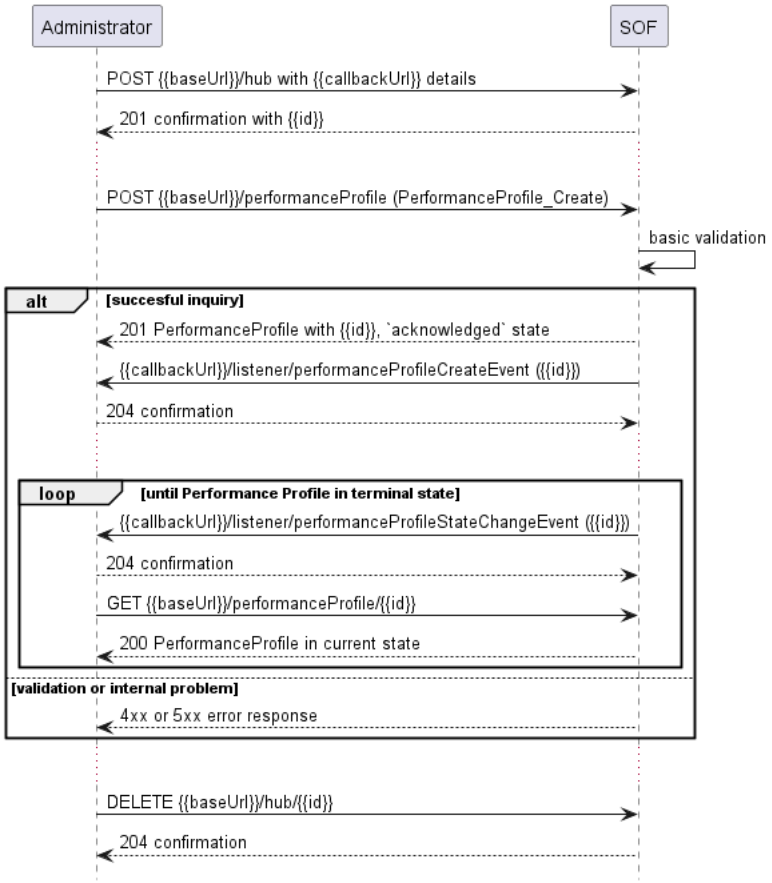


Figure 58. Performance Monitoring Notification Example

The Seller/Server communicates with the Buyer/Client with Notifications provided that:

- Buyer/Client supports a notification mechanism
- Buyer/Client has registered to receive notifications from the Seller/Server

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

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

Figure 59 shows all entities involved in the Notification use cases.

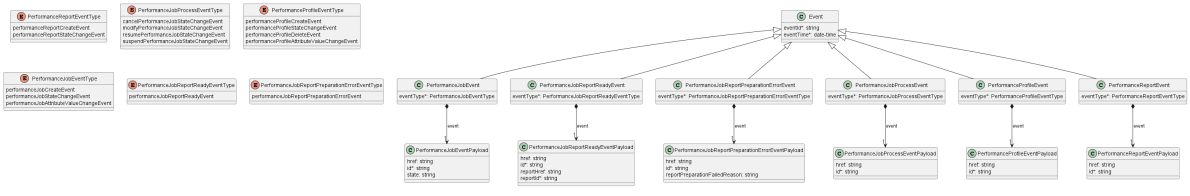


Figure 59. Performance Monitoring Notification Data Model

By using a request in the following snippet, the Buyer/Client subscribes for notification of all types of events. Those are:

- `performanceJobCreateEvent`

- `performanceJobStateChangeEvent`
- `performanceJobAttributeValueChangeEvent`
- `performanceJobReportReadyEvent`
- `performanceJobReportPreparationErrorEvent`
- `cancelPerformanceJobStateChangeEvent`
- `modifyPerformanceJobStateChangeEvent`
- `resumePerformanceJobStateChangeEvent`
- `suspendPerformanceJobStateChangeEvent`
- `performanceProfileCreateEvent`
- `performanceProfileStateChangeEvent`
- `performanceProfileAttributeValueChangeEvent`
- `performanceProfileDeleteEvent`
- `performanceReportCreateEvent`
- `performanceReportStateChangeEvent`

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

[O25] The Seller/Server **MAY** support subscription to Performance Profile Notifications Use Case. [MEF133.1 O8]

[O26] The Buyer/Client **MAY** support subscription to Performance Profile Notifications Use Case. [MEF133.1 O9]

[O27] The Seller/Server **MAY** support unsubscribing from Performance Profile Notifications Use Case. [MEF133.1 O12]

[O28] The Buyer/Client **MAY** support unsubscribing from Performance Profile Notifications Use Case. [MEF133.1 O13]

If the Buyer/Client wishes to receive only notifications of a certain type, a `query` must be added:

```
{
  "callback": "https://bus.com/listenerEndpoint",
  "query": "eventType=performanceJobStateChangeEvent"
}
```

[R119] The Buyer/Client's Subscribe to Performance Job Notifications request **MUST** include: [MEF133.1 R73]

- Notification target information
- List of notification types

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

```
eventType=performanceJobStateChangeEvent,performanceJobReportReadyEvent
```

or

```
eventType=performanceJobStateChangeEvent&eventType=performanceJobReportReadyEvent
```

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

The Seller/Server responds to the subscription request by adding the `id` of the subscription to the message that must be further used for unsubscribing.

```
{
  "id": "00000000-0000-0000-0000-000000000678",
  "callback": "https://bus.com/listenerEndpoint",
  "query": "eventType=performanceJobStateChangeEvent"
}
```

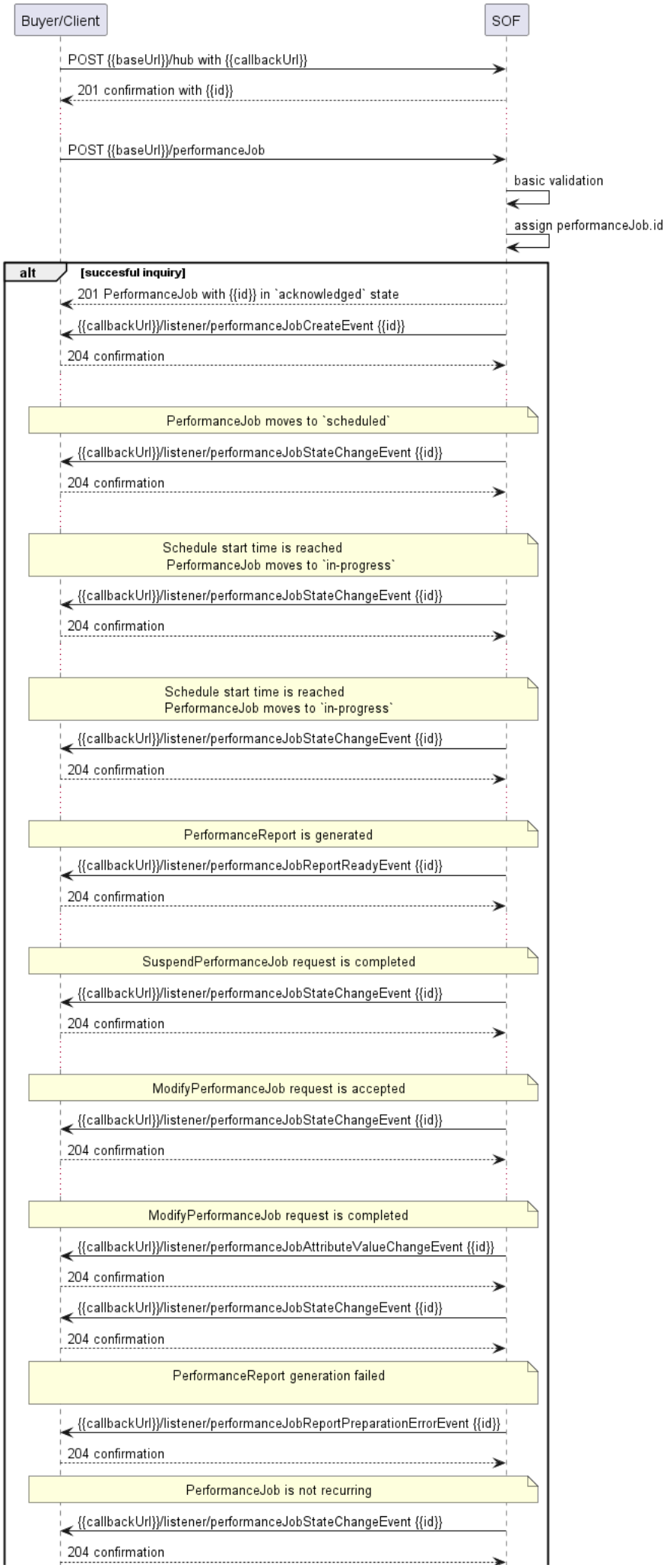
Example of a final address that the Notifications will be sent to (for `performanceJobStateChangeEvent`):

- `https://bus.com/listenerEndpoint/mefApi/legato/performanceNotification/v1/listener/performanceJobStateChangeEvent`

6.29. Use case 29: Send Notification

Notifications are used to asynchronously inform the Buyer/Client about the respective objects and attributes changes.

Figure 60 presents notifications produced by Seller/Server for whole lifecycle of `PerformanceJob` assuming that Buyer/Client subscribed to all event types.


Buyer/Client
SOF
POST {{baseUrl}}/hub with {{callbackUrl}}
201 confirmation with {{id}}
POST {{baseUrl}}/performanceJob
basic validation
assign performanceJob.id
alt
[succesful inquiry]
201 PerformanceJob with {{id}} in `acknowledged` state
{{callbackUrl}}/listener/performanceJobCreateEvent {{id}}
204 confirmation
PerformanceJob moves to `scheduled`
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
Schedule start time is reached
PerformanceJob moves to `in-progress`
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
Schedule start time is reached
PerformanceJob moves to `in-progress`
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
PerformanceReport is generated
{{callbackUrl}}/listener/performanceJobReportReadyEvent {{id}}
204 confirmation
SuspendPerformanceJob request is completed
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
ModifyPerformanceJob request is accepted
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
ModifyPerformanceJob request is completed
{{callbackUrl}}/listener/performanceJobAttributeValueChangeEvent {{id}}
204 confirmation
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation
PerformanceReport generation failed
{{callbackUrl}}/listener/performanceJobReportPreparationErrorEvent {{id}}
204 confirmation
PerformanceJob is not recurring
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}
204 confirmation

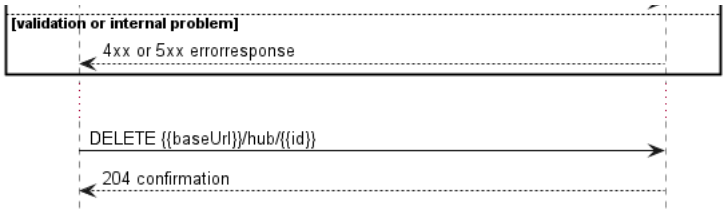


Figure 60. Performance Job lifecycle with all Notifications

After a successful Notification subscription, the Seller/Server sends a `PerformanceJob` create request. The SOF responds with `PerformanceJob` in `acknowledged` state. Creation of `PerformanceJob` is notified with a `performanceJobCreateEvent`. When the validation is successful and Performance Job is not immediate, it moves to `scheduled` and a `performanceJobStateChangeEvent` is sent. When the schedule start time is reached, `PerformanceJob` moves to `in-progress` status and the `performanceJobStateChangeEvent` is sent. Performance Job periodically produces a Performance Report. This is when the `performanceJobReportReadyEvent` is sent. Additional actions, like suspension or modification trigger `performanceJobStateChangeEvent`. In addition, in case of `PerformanceJob` modification, Seller/Server produces `performanceJobAttributeValueChangeEvent` notification. When report generation fails, `performanceJobReportPreparationErrorEvent` is generated.

The following snippets present an example of `performanceJobCreateEvent` and `performanceJobReportReadyEvent`.

```
{
  "eventId": "event-001",
  "eventTime": "2021-06-03T15:56:08.559Z",
  "eventType": "performanceJobCreateEvent",
  "event": {
    "id": "00000000-4444-5555-6666-000000000987"
  }
}
```

```
{
  "eventId": "event-002",
  "eventType": "performanceJobReportReadyEvent",
  "eventTime": "2023-01-15T20:45:24.796Z",
  "event": {
    "id": "00000000-3333-4444-5555-000000004567",
    "reportId": "b54e7020-0bca-11ee-be56-0242ac120002"
  }
}
```

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

Note: The state change notification are sent only when the state attribute actually changes its value. There are no status change notifications sent upon Performance Job creation.

[O29] The Seller/Server **MAY** support Performance Profile Notifications Use Case. [MEF133.1 O10]

[O30] The Buyer/Client **MAY** support Performance Profile Notifications Use Case. [MEF133.1 O11]

[R120] If the Buyer/Client registered for Performance Notifications, the Seller/Server **MUST** notify the Buyer/Client when Performance Job results are available. [MEF133.1 R54, R89]

[R121] The Seller/Server **MUST NOT** send Notifications to Buyer/Client that have not registered for them. [MEF133.1 R75]

[R122] The Seller/Server **MUST** send Notifications to Buyer/Client that have registered for them. [MEF133.1 R74]

[R123] An event triggered by the Performance Report creation (`performanceJobReportReadyEvent`) **MUST** additionally contain the identifier of the Report. [MEF133.1 R76]

[R124] The Seller/Server **MUST** include the following attributes in the Performance Job State Change Notification: [MEF133.1 R76]

- Job Identifier
- Performance Job State

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

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 the appropriate response payload. The Product Order API uses the error responses as depicted and described below.

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

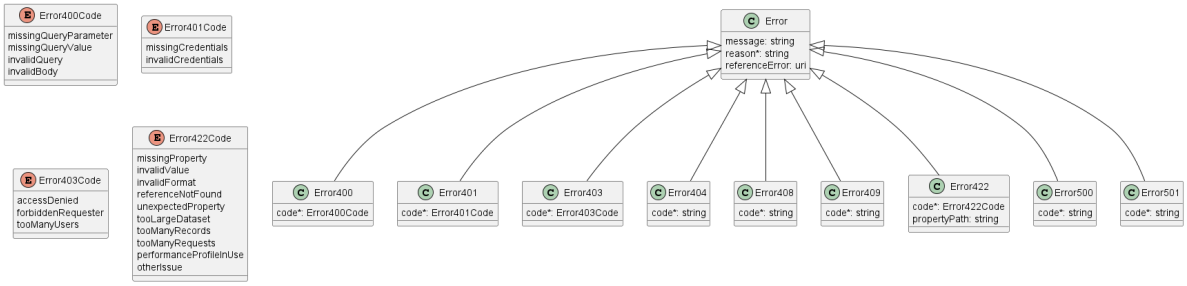


Figure 61. Data model types to represent an erroneous response

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*	Error400Code	

7.1.1.3. enum Error400Code

Description: One of the following error codes:

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

Value	MEF W133.1
missingQueryParameter	MISSING_QUERY_PARAMETER
missingQueryValue	MISSING_QUERY_VALUE
invalidQuery	INVALID_QUERY
invalidBody	INVALID_BODY

7.1.1.4. Type Error401

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

Inherits from:

- Error

Name	Type	Description
code*	Error401Code	

7.1.1.5. enum Error401Code

Description: One of the following error codes:

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

Value	MEF W133.1
missingCredentials	MISSING_CREDENTIALS
invalidCredentials	INVALID_CREDENTIALS

7.1.1.6. Type Error403

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

Inherits from:

- Error

Name	Type	Description
code*	Error403Code	

7.1.1.7. enum Error403Code

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

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

Value	MEF W133.1
accessDenied	ACCESS_DENIED
forbiddenRequester	FORBIDDEN_REQUESTER
tooManyUsers	TOO_MANY_USERS

7.1.1.8. Type Error404

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

Inherits from:

- Error

Name	Type	Description
code*	string	The following error code: - notFound: A current representation for the target resource not found.

7.1.1.9. Type Error408

Description: Request Time-out (https://tools.ietf.org/html/rfc7231#section-6.5.7)

Inherits from:

- Error

Name	Type	Description
code*	string	List of supported error codes: - timeOut: Request Time-out - indicates that the server did not receive a complete request message within the time that it was prepared to wait.

7.1.1.10. Type Error409

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

Inherits from:

- Error

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

7.1.1.11. Type Error422

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*	Error422Code	
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).

7.1.1.12. enum Error422Code

Description: One of the following error codes:

- missingProperty: The property that was 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 target system
- unexpectedProperty: Additional, not expected property has been provided
- tooLargeDataset: Requested entity will produce too many data
- tooManyRecords: The number of records to be provided in the response exceeds the threshold
- tooManyRequests: The number of simultaneous requests from one API client exceeds the threshold
- performanceProfileInUse: Requested Performance Profile is being used by a Performance Job
- otherIssue: Other problem was identified (detailed information provided in a reason).

Value	MEF W133.1
missingProperty	MISSING_PROPERTY
invalidValue	INVALID_VALUE
invalidFormat	INVALID_FORMAT
referenceNotFound	REFERENCE_NOT_FOUND
unexpectedProperty	UNEXPECTED_PROPERTY
tooLargeDataset	TOO_LARGE_DATASET
tooManyRecords	TOO_MANY_RECORDS
tooManyRequests	TOO_MANY_REQUESTS
performanceProfileInUse	PERFORMANCE_PROFILE_IN_USE
otherIssue	OTHER_ISSUE

7.1.1.13. Type Error500

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

Inherits from:

- Error

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.

7.1.1.14. Type Error501

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

Inherits from:

- Error

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

7.1.2. Response pagination

A response to retrieve a list of results (e.g. `GET /performanceJob`) can be paginated. The Buyer/Client can specify following query attributes related to pagination:

- `limit` - number of expected list items
- `offset` - offset of the first element in the result list

The filtering and pagination attributes must be specified in URI query format RFC3986.The Seller/Server returns a list of elements that comply with the requested `limit`. If the requested `limit` is higher than the supported list size the smaller list result is returned. In that case, the size of the result is returned in the header attribute `X-Result-Count`. The Seller can indicate that there are additional results available using:

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

[R125] Seller **MUST** use either `X-Total-Count` or `X-Pagination-Throttled` to indicate that the page was truncated, and additional results are available.

7.2. Management API Data model

Figure 62 presents the whole Performance Monitoring data model. The data types, requirements related to them, and mapping to MEF W133.1 specification are discussed later in this section.

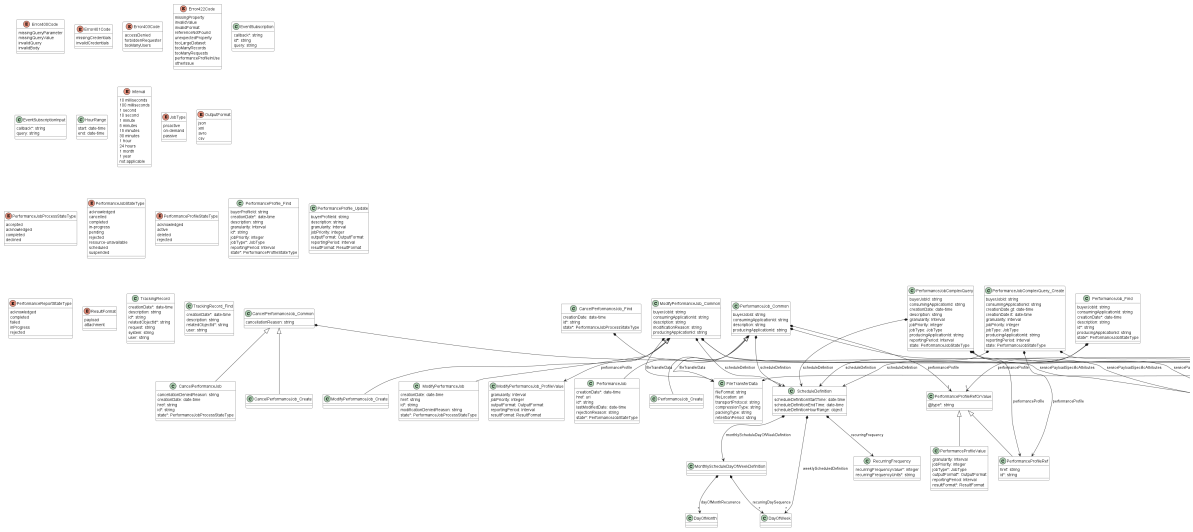


Figure 62. Performance Monitoring Data Model

7.2.1. PerformanceProfile

7.2.1.1. Type PerformanceProfile_Common

Description: A Performance Monitoring Profile specifies the common performance configuration that can be re-used by multiple Performance Jobs.

Name	Type	Description	MEF W133.1
buyerProfileId	string	Identifier of the profile understood and assigned by the Buyer/Client.	
description	string	A free-text description of the Performance Profile	
granularity	Interval	Sampling rate of the collection or production of performance indicators	
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
jobType*	JobType		
outputFormat*	OutputFormat		
reportingPeriod	Interval	Defines the interval for the report generation	
resultFormat*	ResultFormat		

7.2.1.2. Type PerformanceProfile_Create

Description: A Performance Monitoring Profile specifies the common performance configuration that can be re-used by multiple Performance Jobs.

Inherits from:

- PerformanceProfile_Common

7.2.1.3. Type PerformanceProfile

Description: A Performance Monitoring Profile specifies the common performance configuration that can be re-used by multiple Performance Jobs.

Inherits from:

- PerformanceProfile_Common

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when Performance Profile was created.	
href	uri	Hyperlink reference	
id*	string	Unique identifier	
lastModifiedDate	date-time	Date when profile was last modified.	
rejectionReason	string	Reason in case creation request was rejected.	
state*	PerformanceProfileStateType		

7.2.1.4. Type PerformanceProfile_Find

Description: This class represents a single list item for the response of `listPerformanceProfile` operation.

Name	Type	Description	MEF W133.1
buyerProfileId	string	Identifier of the profile understood and assigned by the Buyer/Client.	
creationDate*	date-time	Date when profile was created.	
description	string	A free-text description of the Performance Profile	

Name	Type	Description	MEF W133.1
granularity	Interval	Sampling rate of the collection or production of performance indicators	
id*	string	Unique identifier	
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
jobType*	JobType		
reportingPeriod	Interval	Defines the interval for the report generation.	
state*	PerformanceJobStateType		

7.2.1.5. Type PerformanceProfile_Update

Description: A Performance Monitoring Profile specifies the common performance configuration that can be re-used by multiple Performance Jobs.

Name	Type	Description	MEF W133.1
buyerProfileId	string	Identifier of the profile understood and assigned by the Buyer/Client.	
description	string	A free-text description of the Performance Profile	
granularity	Interval	Sampling rate of the collection or production of performance indicators	
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
outputFormat	OutputFormat		
reportingPeriod	Interval	Defines the interval for the report generation.	
resultFormat	ResultFormat		

7.2.1.6. Type PerformanceProfileRef

Description: A reference to a Performance Profile resource

Inherits from:

- PerformanceProfileRefOrValue

Name	Type	Description	MEF W133.1
href	string	Hyperlink to the referenced Performance Profile	
id*	string	Identifier of the referenced Performance Profile	

7.2.1.7. Type PerformanceProfileRefOrValue

Description: Defines the reference to Performance Monitoring Profile or defines values from PerformanceProfile type.

Name	Type	Description	MEF W133.1
@type*	string	This field is used as a discriminator to differentiate if object relates directly to Performance Profile entity or defines values from PerformanceProfile type.	

7.2.1.8. enum PerformanceProfileStateType

Description: The state of the Performance Monitoring Profile.

state	MEF W133.1 name	Description
acknowledged	Acknowledged	A Create Performance Monitoring Profile request has been received by the Server and has passed basic validation. Performance Monitoring Profile Identifier is assigned in the Acknowledged state. The request remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated the Performance Monitoring Profile moves to the Active state. If not all attributes are validated, the request moves to the Rejected state.
active	Active	A Performance Monitoring Profile is active and can be used as a template for Performance Monitoring Job creation.
deleted	Deleted	A Performance Monitoring Profile that does not have any Performance Monitoring Jobs attached is deleted.

state	MEF W133.1 name	Description
`rejected`	Rejected	A Create Performance Monitoring Profile request fails validation and is rejected with error indications by the Server.

Value	MEF W133.1
acknowledged	ACKNOWLEDGED
active	ACTIVE
deleted	DELETED
rejected	REJECTED

7.2.1.9. Type PerformanceProfileValue

Description: Direct assignment of values defined by PerformanceProfile type to PerformanceJob object. Necessary when PerformanceJob is created without reference to PerformanceProfile.

Inherits from:

- PerformanceProfileRefOrValue

Name	Type	Description	MEF W133.1
granularity	Interval	Sampling rate of the collection or production of performance indicators	
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
jobType*	JobType		
outputFormat*	OutputFormat		
reportingPeriod	Interval	Defines the interval for the report generation.	
resultFormat*	ResultFormat		

7.2.2. PerformanceJob

7.2.2.1. Type PerformanceJob_Common

Description: A Performance Monitoring Job specifies the performance monitoring objectives specific to each subject of monitoring which could be an ordered pair (i.e., two

UNIs) or an entity (i.e., port).

Name	Type	Description	MEF W133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	
consumingApplicationId	string	Identifier of consuming application	
description	string	A free-text description of the Performance Job	
fileTransferData	FileTransferData		
performanceProfile*	PerformanceProfileRefOrValue		
producingApplicationId	string	Identifier of producing application	
scheduleDefinition	ScheduleDefinition		
servicePayloadSpecificAttributes*	ServicePayloadSpecificAttributes		

7.2.2.2. Type PerformanceJob_Create

Description: A Performance Monitoring Job specifies the performance monitoring objectives specific to each subject of monitoring which could be an ordered pair (i.e., two UNIs) or an entity (i.e., port).

Inherits from:

- PerformanceJob_Common

7.2.2.3. Type PerformanceJob

Description: A Performance Monitoring Job specifies the performance monitoring objectives specific to each subject of monitoring which could be an ordered pair (i.e., two UNIs) or an entity (i.e., port).

Inherits from:

- PerformanceJob_Common

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when Performance Job was created.	
href	uri	Hyperlink reference	
id*	string	Unique identifier	
lastModifiedDate	date-time	Date when job was last modified.	
rejectionReason	string	Reason in case creation request was rejected.	
state*	PerformanceJobStateType		

7.2.2.4. Type PerformanceJob_Find

Description: This class represents a single list item for the response of `listPerformanceJob` operation.

Name	Type	Description	MEF W133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	
consumingApplicationId	string	Identifier of consuming application	
creationDate*	date-time	Date when job was created.	
description	string	A free-text description of the Performance Job	
id*	string	Unique identifier	
performanceProfile*	PerformanceProfileRefOrValue		

Name	Type	Description	MEF W133.1
producingApplicationId	string	Identifier of producing application	
scheduleDefinition	ScheduleDefinition		
state*	PerformanceJobStateType		

7.2.2.5. Type CancelPerformanceJob_Common

Description: Request for cancellation of an existing Performance Job

Name	Type	Description	MEF W133.1
cancellationReason	string	An optional attribute that allows the Buyer/Client to provide additional detail to the Seller/Server on the reason for cancelling Performance Job.	
performanceJob*	PerformanceJobRef		

7.2.2.6. Type CancelPerformanceJob_Create

Description: Request for cancellation of an existing Performance Job

Inherits from:

- CancelPerformanceJob_Common

7.2.2.7. Type CancelPerformanceJob

Description: Request for cancellation of an existing Performance job

Inherits from:

- CancelPerformanceJob_Common

Name	Type	Description	MEF W133.1

Name	Type	Description	MEF W133.1
cancellationDeniedReason	string	If the Cancel Performance Job request is denied by the Seller/Server, the Seller/Server provides a reason to the Buyer/Client using this attribute.	
creationDate	date-time	Date when Cancel Performance Job was created.	
href	string	Hyperlink to the Cancel Performance Job entity	
id*	string	Unique identifier for the Cancel Performance Job that is generated by the Seller/Server when the Cancel Performance Job request `state` is set to `acknowledged`.	
state*	PerformanceJobProcessStateType		

7.2.2.8. Type CancelPerformanceJob_Find

Description: This class represents a single list item for the response of `listCancelPerformanceJob`

Name	Type	Description	MEF W133.1

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Cancel Performance Job was created.	
id*	string	Unique identifier for the Cancel Performance Job that is generated by the Seller/Server when the Cancel Performance Job request `state` is set to `acknowledged`.	
performanceJob*	PerformanceJobRef		
state*	PerformanceJobProcessStateType		

7.2.2.9. Type ModifyPerformanceJob_Common

Description: Request for modification of an existing Performance Job

Name	Type	Description	MEF W133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	
consumingApplicationId	string	Identifier of consuming application	
description	string	A free-text description of the Performance Job	
fileTransferData	FileTransferData		

Name	Type	Description	MEF W133.1
modificationReason	string	An optional attribute that allows the Buyer/Client to provide additional detail to the Seller/Server on the reason for modifying Performance Job.	
performanceJob*	PerformanceJobRef		
performanceProfile	ModifyPerformanceJob_ProfileValue		
producingApplicationId	string	Identifier of producing application	
scheduleDefinition	ScheduleDefinition		
servicePayloadSpecificAttributes	ServicePayloadSpecificAttributes		

7.2.2.10. Type ModifyPerformanceJob_Create

Description: Request for modification of an existing Performance Job

Inherits from:

- ModifyPerformanceJob_Common

7.2.2.11. Type ModifyPerformanceJob

Description: Request for modification of an existing Performance Job

Inherits from:

- ModifyPerformanceJob_Common

Name	Type	Description	MEF W133.1

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Modify Performance Job was created.	
href	string	Hyperlink to the Modify Performance Job entity	
id*	string	Unique identifier for the Modify Performance Job that is generated by the Seller/Server when the Modify Performance Job request `state` is set to `acknowledged`	
modificationDeniedReason	string	If the Modify Performance Job request is denied by the Seller/Server, the Seller/Server provides a reason to the Buyer/Client using this attribute.	
state*	PerformanceJobProcessStateType		

7.2.2.12. Type ModifyPerformanceJob_Find

Description: This class represents a single list item for the response of `listModifyPerformanceJob`

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Modify Performance Job was created.	
id*	string	Unique identifier for the Modify Performance Job that is generated by the Seller/Server when the Modify Performance Job request `state` is set to `acknowledged`.	
performanceJob*	PerformanceJobRef		
state*	PerformanceJobProcessStateType		

7.2.2.12. Type ModifyPerformanceJob_ProfileValue

Description: Direct assignment of values defined by PerformanceProfile type to PerformanceJob object. Necessary when PerformanceJob is created without reference to PerformanceProfile.

Name	Type	Description	MEF W133.1
granularity	Interval	Sampling rate of the collection or production of performance indicators	
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
outputFormat	OutputFormat		
reportingPeriod	Interval	Defines the interval for the report generation	
resultFormat	ResultFormat		

7.2.2.13. Type PerformanceJobComplexQuery_Create

Description: Performance Job Complex Query entity is used to perform searches on Performance Job entities, including clauses based on ScheduleDefinition and ServicePayloadSpecificAttributes.

Name	Type	Description	MEF W133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	
consumingApplicationId	string	Identifier of consuming application	
creationDate.gt	date-time	Date when Performance Job was created - greater than.	
creationDate.lt	date-time	Date when Performance Job was created - lower than.	
granularity	Interval	Sampling rate of the collection or production of performance indicators	

Name	Type	Description	MEF W133.1
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
jobType	JobType		
performanceProfile	PerformanceProfileRef		
producingApplicationId	string	Identifier of producing application	
reportingPeriod	Interval	Defines the interval for the report generation.	
scheduleDefinition	ScheduleDefinition		
servicePayloadSpecificAttributes	ServicePayloadSpecificAttributes		
state	PerformanceJobStateType		

7.2.2.14. Type PerformanceJobComplexQuery

Description: Performance Job Complex Query entity is used to perform searches on Performance Job entities, including clauses based on ScheduleDefinition and ServicePayloadSpecificAttributes.

Name	Type	Description	MEF W133.1

Name	Type	Description	MEF W133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	
consumingApplicationId	string	Identifier of consuming application	
creationDate	date-time	Date when Performance Job was created.	
description	string	A free-text description of the Performance Job	
granularity	Interval	Sampling rate of the collection or production of performance indicators	

Name	Type	Description	MEF W133.1
jobPriority	integer	The priority of the Performance Job. The way the management application will use the Job priority to schedule Job execution is application specific and out the scope.	
jobType	JobType		
performanceJob	PerformanceJobRef		
performanceProfile	PerformanceProfileRef		
producingApplicationId	string	Identifier of producing application	
reportingPeriod	Interval	Defines the interval for the report generation.	
scheduleDefinition	ScheduleDefinition		
servicePayloadSpecificAttributes	ServicePayloadSpecificAttributes		
state	PerformanceJobStateType		

7.2.2.15. enum PerformanceJobProcessStateType

Description: The state of process related to Performance Job

state	MEF W133 name	Description

state	MEF W133 name	Description
accepted	Accepted	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been validated and accepted by the Seller/Server.
acknowledged	Acknowledged	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been received by the Seller/Server and has passed basic validation. Performance Monitoring Job Process Identifier is assigned in the Acknowledged state. The request remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated, the request moves to the Accepted state. If not all attributes are validated, the request moves to the Declined state.
completed	Completed	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has been completed by the Seller/Server.
declined	Declined	The Cancel/Modify/Resume/Suspend Performance Monitoring Job request has failed validation and been declined by the Seller/Server.

Value	MEF W133.1
accepted	ACCEPTED
acknowledged	ACKNOWLEDGED
completed	COMPLETED
declined	DECLINED

7.2.2.16. Type PerformanceJobRef

Description: A reference to a Performance Job resource

Inherits from:

- PerformanceJobRefOrValue

Name	Type	Description	MEF W133.1
href	string	Hyperlink to the referenced Performance Job	
id*	string	Identifier of the referenced Performance Job	

7.2.2.17. Type PerformanceJobRefOrValue

Description: Defines the reference to Performance Monitoring Job or defines values from PerformanceJob type.

Name	Type	Description	MEF W133.1
@type*	string	This field is used as a discriminator to differentiate if object relates directly to Performance Job entity or defines values from PerformanceJob type.	

7.2.2.18. enum PerformanceJobStateType

Description: The state of the Performance Monitoring Job.

state	MEF W133 name	Description
acknowledged	Acknowledged	A Create Performance Monitoring Job request has been received by the Seller/Server and has passed basic validation. Performance Monitoring Job Identifier is assigned in the Acknowledged state. The request remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated the request determines if the start time is immediate or scheduled. If immediate, the Performance Monitoring Job moves to the In-progress state. Otherwise, the Performance Monitoring Job moves to the Scheduled state. If not all attributes are validated, the request moves to the Rejected state.
cancelled	Cancelled	A Performance Monitoring Job that is In-Progress, Suspended or Scheduled is cancelled.
completed	Completed	A non-recurring Performance Monitoring Job finished execution.

state	MEF W133 name	Description
in-progress	In-Progress	A Performance Monitoring Job is running. Upon completion of the Job, a determination if the Performance Monitoring Job is a one-time Job or is recurring is performed. If the Performance Monitoring Job is a one-time Job, the state of the Performance Monitoring Job moves to the Completed state. If the Performance Monitoring Job is recurring, the Performance Monitoring Job circles back to determine if it has an immediate start time or a scheduled start time. In case a Suspend Performance Monitoring Job request is accepted, the Job moves to the Suspended state. If a Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state.
pending	Pending	A Modify Performance Monitoring Job request has been accepted by the Seller/Server. The Performance Monitoring Job remains in the Pending state while updates to the Job are completed. Once updates are complete, the Job returns to the Scheduled or In-Progress status depending on the schedule definition.
rejected	Rejected	A Create Performance Monitoring Job request fails validation and is rejected with error indications by the Seller/Server.
resource-unavailable	Resource Unavailable	A Performance Monitoring Job cannot be allocated necessary resources when moving to execution (In-Progress state).
scheduled	Scheduled	A Performance Monitoring Job is created that does not have an immediate start time. The Performance Monitoring Job stays in the Scheduled state until the start time is reached. The Performance Monitoring Job then moves to In-Progress. If Cancel Performance Monitoring Job request is accepted, Job moves to Cancelled state. If modify Performance Monitoring Job request is accepted, Job moves to Pending state.

state	MEF W133 name	Description
suspended	Suspended	A Suspend Performance Monitoring Job request is accepted by the Seller/Server. The Job remains in the Suspended state until a Resume Performance Monitoring Job request is accepted by the Seller/Server at which time the Job returns to the In-Progress state. If Cancel Performance Monitoring Job request is accepted, Job moves to Cancelled state. If modify Performance Monitoring Job request is accepted, Job moves to Pending state.

Value	MEF W133.1
acknowledged	ACKNOWLEDGED
cancelled	CANCELLED
completed	COMPLETED
in-progress	IN-PROGRESS
pending	PENDING
rejected	REJECTED
resource-unavailable	RESOURCE-UNAVAILABLE
scheduled	SCHEDULED
suspended	SUSPENDED

7.2.2.19. Type PerformanceJobValue

Description: Direct assignment of values defined by PerformanceJob type to PerformanceReport object. Necessary when PerformanceReport is not created by PerformanceJob and without relation to PerformanceJob.

Inherits from:

- PerformanceJobRefOrValue

Name	Type	Description	MEF W133.1
consumingApplicationId	string	Identifier of consuming application	
fileTransferData	FileTransferData		

Name	Type	Description	MEF W133.1
granularity	Interval	Sampling rate of the collection or production of performance indicators	
outputFormat*	OutputFormat		
producingApplicationId	string	Identifier of producing application	
resultFormat*	ResultFormat		
servicePayloadSpecificAttributes*	ServicePayloadSpecificAttributes		

7.2.2.20. Type ResumePerformanceJob_Common

Description: Request for resumption of an existing Performance Job

Name	Type	Description	MEF W133.1
performanceJob*	PerformanceJobRef		
resumptionReason	string	An optional attribute that allows the Buyer/Client to provide additional detail to the Seller/Server on the reason for resuming Performance Job.	

7.2.2.21. Type ResumePerformanceJob_Create

Description: Request for resumption of an existing Performance Job

Inherits from:

- ResumePerformanceJob_Common

7.2.2.22. Type ResumePerformanceJob

Description: Request for resumption of an existing Performance job

Inherits from:

- ResumePerformanceJob_Common

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Suspend Performance Job was created.	
href	string	Hyperlink to the Resume Performance Job entity	
id*	string	Unique identifier for the Resume Performance Job that is generated by the Seller/Server when the Resume Performance Job request `state` is set to `acknowledged`.	
resumptionDeniedReason	string	If the Resume Performance Job request is denied by the Seller/Server, the Seller/Server provides a reason to the Buyer/Client using this attribute.	
state*	PerformanceJobProcessStateType		

7.2.2.23. Type ResumePerformanceJob_Find

Description: This class represents a single list item for the response of `listResumePerformanceJob`

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Suspend Performance Job was created.	
id*	string	Unique identifier for the Resume Performance Job that is generated by the Seller/Server when the Resume Performance Job request `state` is set to `acknowledged`.	
performanceJob*	PerformanceJobRef		
state*	PerformanceJobProcessStateType		

7.2.2.24. Type SuspendPerformanceJob_Common

Description: Request for suspension of an existing Performance Job

Name	Type	Description	MEF W133.1
performanceJob*	PerformanceJobRef		
suspensionReason	string	An optional attribute that allows the Buyer/Client to provide additional detail to the Seller/Server on the reason for suspending Performance Job.	

7.2.2.25. Type SuspendPerformanceJob_Create

Description: Request for suspension of an existing Performance Job

Inherits from:

- SuspendPerformanceJob_Common

7.2.2.26. Type SuspendPerformanceJob

Description: Request for suspension of an existing Performance Job

Inherits from:

- SuspendPerformanceJob_Common

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Suspend Performance Job was created.	
href	string	Hyperlink to the Suspend Performance Job entity	
id*	string	Unique identifier for the Suspend Performance Job that is generated by the Seller/Server when the Suspend Performance Job request `state` is set to `acknowledged`.	
state*	PerformanceJobProcessStateType		
suspensionDeniedReason	string	If the Suspend Performance Job request is denied by the Seller/Server, the Seller/Server provides a reason to the Buyer/Client using this attribute.	

7.2.2.27. Type SuspendPerformanceJob_Find

Description: This class represents a single list item for the response of `listSuspendPerformanceJob`

Name	Type	Description	MEF W133.1
creationDate	date-time	Date when Suspend Performance Job was created.	
id*	string	Unique identifier for the Suspend Performance Job that is generated by the Seller/Server when the Suspend Performance Job request `state` is set to `acknowledged`.	
performanceJob*	PerformanceJobRef		
state*	PerformanceJobProcessStateType		

7.2.3. PerformanceReport

7.2.3.1. Type PerformanceReport_Common

Description: The execution of PM Job results in Performance Measurement collections that provide Buyer/Client with performance objectives results.

Name	Type	Description	MEF W133.1
description	string	A free-text description of the performance report	
reportingTimeframe	ReportingTimeframe		

7.2.3.2. Type PerformanceReport_Create

Description: In some cases, performance statistics are generated without provisioning a PM Job. These statistics can be collected with an ad-hoc Performance Report creation.

Inherits from:

- PerformanceReport_Common

Name	Type	Description	MEF W133.1
performanceJob*	PerformanceJobValue		

7.2.3.3. Type PerformanceReport

Description: The execution of PM Job results in Performance Measurement collections that provide Buyer/Client with performance objective results.

Inherits from:

- PerformanceReport_Common

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when Performance Report was created.	
failureReason	string	Reason in case report generation failed.	
href	uri	Hyperlink reference	
id*	string	Unique identifier	
lastModifiedDate	date-time	Date when report was last modified.	
performanceJob	PerformanceJobRefOrValue		
reportContent	ReportContentItem[]		
reportUrl	AttachmentURL		
state*	PerformanceReportStateType		

7.2.3.4. Type PerformanceReport_Find

Description: This class represents a single list item for the response of `listPerformanceReport` operation.

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when report was created.	
description	string	A free-text description of the Performance Report	
id*	string	Unique identifier	
performanceJob	PerformanceJobRefOrValue		
reportingTimeframe	ReportingTimeframe		
state*	PerformanceReportStateType		

7.2.3.5. Type PerformanceReportComplexQuery_Create

Description: Performance Report Complex Query entity is used to perform searches on Performance Report entities, including clauses based on ServicePayloadSpecificAttributes.

Name	Type	Description	MEF W133.1
consumingApplicationId	string	Identifier of consuming application	
creationDate.gt	date-time	Date when Performance Report was created - greater than.	
creationDate.lt	date-time	Date when Performance Report was created - lower than.	
granularity	Interval	Sampling rate of the collection or production of performance indicators	
outputFormat	OutputFormat		
performanceJob	PerformanceJobRef		
producingApplicationId	string	Identifier of producing application	
reportingTimeframe.startDate.gt	date-time	Start date of reporting timeframe - greater than.	
reportingTimeframe.startDate.lt	date-time	Start date of reporting timeframe - lower than.	

Name	Type	Description	MEF W133.1
reportingTimeframe.endDate.gt	date-time	End date of reporting timeframe - greater than.	
reportingTimeframe.endDate.lt	date-time	End date of reporting timeframe - lower than.	
resultFormat	ResultFormat		
servicePayloadSpecificAttributes	ServicePayloadSpecificAttributes		
state	PerformanceReportStateType		

7.2.3.6. Type PerformanceReportComplexQuery

Description: Performance Report Complex Query entity is used to perform searches on Performance Report entities, including clauses based on ServicePayloadSpecificAttributes.

Name	Type	Description	MEF W133.1
consumingApplicationId	string	Identifier of consuming application	
creationDate	date-time	Date when Performance Report was created.	
description	string	A free-text description of the performance report	
granularity	Interval	Sampling rate of the collection or production of performance indicators	

Name	Type	Description	MEF W133.1
outputFormat	OutputFormat		
performanceJob	PerformanceJobRef		
performanceReport	PerformanceReportRef		
producingApplicationId	string	Identifier of producing application	
reportingTimeframe	ReportingTimeframe		
resultFormat	ResultFormat		
servicePayloadSpecificAttributes	ServicePayloadSpecificAttributes		
state	PerformanceReportStateType		

7.2.3.7. Type PerformanceReportRef

Description: A reference to a Performance Report resource

Name	Type	Description	MEF W133.1
href	string	Hyperlink to the referenced Performance Report	
id*	string	Identifier of the referenced Performance Report	

7.2.3.8. enum PerformanceReportStateType

Description: Possible values for the state of a Performance Report.

State	Description
acknowledged	A Performance Report request has been received by Seller/Server and has passed basic validations. Performance Report Identifier is assigned in the Acknowledged state. The report remains in the Acknowledged state until all validations as applicable are completed. If the attributes are validated, the Performance Report moves to the In-Progress state. If not all attributes are validated, the report moves to the Rejected state.
completed	A Performance Report is completed and results are available.
failed	A Performance Report processing has failed.
inProgress	A Performance Report has successfully passed the validations checks and the report processing has started.

State	Description
rejected	This state indicates that: - Invalid information is provided through the `PerformanceReport` request - The request fails to meet validation rules for `PerformanceReport` delivery (processing).

Value	MEF W133.1
acknowledged	ACKNOWLEDGED
completed	COMPLETED
failed	FAILED
inProgress	IN_PROGRESS
rejected	REJECTED

7.2.4. Common

Types described in this subsection are shared among two or more LSO APIs.

7.2.4.1. Type AttachmentURL

Description: The AttachmentURL is used to get the PM report.

Name	Type	Description	MEF W133.1
url*	string	'Uniform Resource Locator, is a web page address (a subset of URI).'	

7.2.4.2. Type DayOfMonth

Description: Day of the month for recurrence

Type	Description
integer	Minimum: 1, maximum: 31

7.2.4.3. Type DayOfWeek

Description: Day of the week for recurrence. 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday.

Type	Description
integer	Minimum: 1, maximum: 7

7.2.4.4. Type FileTransferData

Description: Defines place where the report content should be stored.

Name	Type	Description	MEF W133.1
fileFormat	string	Format of the file containing collected data.	
fileLocation	uri	Location of the file containing collected data.	
transportProtocol	string	Transport protocol to use for file transfer.	
compressionType	string	Compression types used for the collected data file.	
packingType	string	Specify if the data file is to be packed.	
retentionPeriod	string	A time interval to retain the file.	

7.2.4.4. Type HourRange

Description:

Name	Type	Description	MEF W133.1
start	date-time		
end	date-time		

7.2.4.5. enum Interval

Description:

Value	MEF W133.1
10 milliseconds	10 MILLISECONDS
100 milliseconds	100 MILLISECONDS
1 second	1 SECOND
10 second	10 SECOND
1 minute	1 MINUTE
5 minutes	5 MINUTES
15 minutes	15 MINUTES
30 minutes	30 MINUTES
1 hour	1 HOUR
24 hours	24 HOURS

Value	MEF W133.1
1 month	1 MONTH
1 year	1 YEAR
not applicable	NOT APPLICABLE

7.2.4.6. enum JobType

Description: The type of PM Job

Value	MEF W133.1
proactive	PROACTIVE
on-demand	ON-DEMAND
passive	PASSIVE

7.2.4.6. Type MeasurementTime

Description: Timeframe boundary for collected data

Name	Type	Description	MEF W133.1
measurementStartDate*	date-time	Start date of the time period to which collected data points belong.	
measurementEndDate*	date-time	Start date of the time period to which collected data points belong.	
measurementInterval*	Interval	Length of the measurement interval	

7.2.4.7. Type MonthlyScheduleDayOfWeekDefinition

Description: Monthly scheduled day of week.

Name	Type	Description	MEF W133.1
recurringDaySequence	DayOfWeek[]		
dayOfMonthRecurrence	DayOfMonth[]		

7.2.4.8. enum OutputFormat

Description: List of possible output formats for the Performance Report

Value	MEF W133.1
json	JSON

Value	MEF W133.1
xml	XML
avro	AVRO
csv	CSV

7.2.4.9. Type RecurringFrequency

Description: A recurring frequency to run a job within timeframe defined by schedule definition, for example: every 5 minutes, 15 minutes, 1 hour, 1 day

Name	Type	Description	MEF W133.1
recurringFrequencyValue*	integer	The value of the recurrence as an integer. For example, if the recurring frequency is 2 weeks this value is 2.	
recurringFrequencyUnits*	string	The unit of measure in recurring frequency. For example, if a recurring frequency is 2 weeks this value is WEEKS.	

7.2.4.10. Type ReportContentItem

Description: Single item of the performance monitoring results in case result format was set to payload. Each item contains timeframe of the collected data and list of values measured in that timeframe.

Name	Type	Description	MEF W133.1
measurementTime*	MeasurementTime		
measurementDataPoints	ResultPayload[]	List of performance monitoring values measured in the related timeframe.	

7.2.4.11. Type ReportingTimeframe

Description: Specifies the date range between which data points will be included in the report.

Name	Type	Description	MEF W133.1
reportingStartDate	date-time		

Name	Type	Description	MEF W133.1
reportingEndDate	date-time		

7.2.4.12. `enum` ResultFormat

Description: List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.

Value	MEF W133.1
payload	PAYLOAD
attachment	ATTACHMENT

7.2.4.13. Type ResultPayload

Description: ResultPayload is used as an extension point for MEF specific service performance monitoring results. The `@type` attribute is used as a discriminator.

Name	Type	Description	MEF W133.1
@type*	string	The name that uniquely identifies type of performance monitoring results that are returned by the Performance Report. In case of MEF services this is the URN provided in performance monitoring results specification. The named type must be a subclass of ResultPayload.	

7.2.4.14. Type ScheduleDefinition

Description: The schedule definition for running jobs.

Name	Type	Description	MEI W13

Name	Type	Description	MEI W13
scheduleDefinitionStartTime	date-time	The Start time of the Schedule Definition. If the attribute is empty the Schedule starts immediately after provisioning of the Job.	
scheduleDefinitionEndTime	date-time	The Endtime of the Schedule Definition. If the attribute is empty the Schedule runs forever, not having a time constraint.	
recurringFrequency	RecurringFrequency		

Name	Type	Description	MEF W13
scheduleDefinitionHourRange	object[]	A list of time ranges within a specific day that the schedule will be active on, for example 08:00-12:00, 16:00-19:00.	
monthlyScheduleDayOfWeekDefinition	MonthlyScheduleDayOfWeekDefinition		
weeklyScheduledDefinition	DayOfWeek[]	The weekly schedule is used to define a schedule that is based on the days of the week, e.g. a schedule that will be active only on Monday and Tuesday.	

7.2.4.15. Type ServicePayloadSpecificAttributes

Description: ServicePayloadSpecificAttributes is used as an extension point for MEF specific service performance monitoring configuration. It includes definition of service/entity and applicable performance monitoring objectives. The `@type` attribute is used as a discriminator.

Name	Type	Description	MEF W133.1

Name	Type	Description	MEF W133.1
@type*	string	The name that uniquely identifies type of performance monitoring configuration that specifies PM objectives. In case of MEF services this is the URN provided in performance monitoring configuration specification. The named type must be a subclass of ServicePayloadSpecificAttributes.	

7.2.4.16. Type TrackingRecord

Description: Tracking Records allow the tracking of modifications of Performance Job, Profile or Report.

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when record was created.	
description	string	Free-text field describing the action that created the Tracking Record and its details.	
id*	string	Identifier of the Tracking Record	
relatedObjectId*	string	Identifier of Performance Job, Profile or Report	
request	string	Request that created the Tracking Record.	
system	string	Describes the system from which the action was done.	
user	string	Describes the user doing the action.	

7.2.4.17. Type TrackingRecord_Find

Description: This class represents a single list item for the response of `listTrackingRecord` operation.

Name	Type	Description	MEF W133.1
creationDate*	date-time	Date when record was created.	
description	string	Describes the action that created the Tracking Record, such as: create, update.	
relatedObjectId*	string	Identifier of Performance Job, Profile or Report.	

Name	Type	Description	MEF W133.1
user	string	User that executed the action which created a Tracking Record.	

7.2.5. Notification registration

Notification registration and management are done through `/hub` API endpoint. The below sections describe data models related to this endpoint.

7.2.4.1. Type EventSubscriptionInput

Description: This class is used to register for Notifications.

Name	Type	Description
callback*	string	This callback value must be set to *host* property from Performance Notification AI (performanceNotification.api.yaml). This property is appended with the base path an specified in that API to construct an URL to which notification is sent. E.g. for 'callb "https://buyer.co/listenerEndpoint", the performance job state change event notificati `https://buyer.co/listenerEndpoint/mefApi/legato/performanceMonitoring/v1/listener/
query	string	This attribute is used to define to which type of events to register to. Example: 'query performanceReportStateChangeEvent'. To subscribe for more than one event type, pu `eventType=performanceReportStateChangeEvent,performanceJobCreateEvent`. The 'PerformanceEventType' in performanceNotification.api.yaml. An empty query is trea ending in subscription for all event types.

7.2.4.2. Type EventSubscription

Description: This resource is used to respond to notification subscriptions.

Name	Type	Description	MEF W133.1
callback*	string	The value provided by the `EventSubscriptionInput` during notification registration.	
id*	string	An identifier of this Event Subscription assigned when a resource is created.	
query	string	The value provided by the `EventSubscriptionInput` during notification registration.	

7.3. Notification API Data model

Figure 63 presents the Performance Monitoring Notification data model.

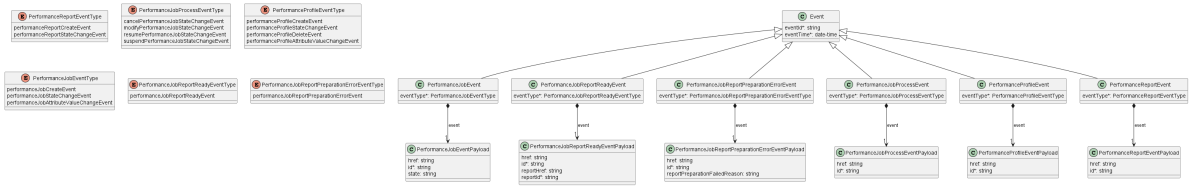


Figure 63. Performance Monitoring Notification Data Model

This data model is used to construct requests and responses of the API endpoints described in 5.2.2. Buyer/Client (CUS, BUS, SOF) side Performance Monitoring API Endpoints.

7.3.1. Type Event

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

Name	Type	Description	MEF W133.1
eventId*	string	Id of the event	
eventTime*	date-time	Date-time when the event occurred	

7.3.2. Type PerformanceProfileEvent

Description:

Inherits from:

- Event

Name	Type	Description	MEF W133.1
eventType*	PerformanceProfileEventType		
event*	PerformanceProfileEventPayload		

7.3.3. enum PerformanceProfileEventType

Description: Indicates the type of Performance Profile event.

Value	MEF W133.1
performanceProfileCreateEvent	PERFORMANCE_PROFILE_CREATE_EVENT
performanceProfileStateChangeEvent	PERFORMANCE_PROFILE_STATE_CHANGE_EV
performanceProfileDeleteEvent	PERFORMANCE_PROFILE_DELETE_EVENT

Value	MEF W133.1
performanceProfileAttributeValueChangeEvent	PERFORMANCE_PROFILE_ATTRIBUTE_VALUE_

7.3.4. Type PerformanceProfileEventPayload

Description: The identifier of the Performance Profile being subject of this event.

Name	Type	Description	MEF W133.1
href	string	Hyperlink to access the Performance Profile	
id*	string	ID of the Performance Profile	

7.3.5. Type PerformanceJobEvent

Description:

Inherits from:

- Event

Name	Type	Description	MEF W133.1
eventType*	PerformanceJobEventType		
event*	PerformanceJobEventPayload		

7.3.6. enum PerformanceJobEventType

Description: Indicates the type of Performance Job event.

Value	MEF W133.1
performanceJobCreateEvent	PERFORMANCE_JOB_CREATE_EVENT
performanceJobStateChangeEvent	PERFORMANCE_JOB_STATE_CHANGE_EVENT
performanceJobAttributeValueChangeEvent	PERFORMANCE_JOB_ATTRIBUTE_VALUE_CHANC

7.3.7. Type PerformanceJobEventPayload

Description: The identifier of the Performance Job being subject of this event and its state.

Name	Type	Description	MEF 133.1
href	string	Hyperlink to access the Performance Job	
id*	string	ID of the Performance Job	
state	string	State of the Performance Job	

7.3.8. Type PerformanceJobProcessEvent

Description:

Inherits from:

- Event

Name	Type	Description	MEF W133.1
eventType*	PerformanceJobProcessEventType		
event*	PerformanceJobProcessEventPayload		

7.3.9. enum PerformanceJobProcessEventType

Description: Indicates the type of Performance Job Process event.

Value	MEF W133.1
cancelPerformanceJobStateChangeEvent	CANCEL_PERFORMANCE_JOB_STATE_CHANGE_EV
modifyPerformanceJobStateChangeEvent	MODIFY_PERFORMANCE_JOB_STATE_CHANGE_EV
resumePerformanceJobStateChangeEvent	RESUME_PERFORMANCE_JOB_STATE_CHANGE_EV
suspendPerformanceJobStateChangeEvent	SUSPEND_PERFORMANCE_JOB_STATE_CHANGE_E

7.3.10. Type PerformanceJobProcessEventPayload

Description: The identifier of the Performance Job Process including:

- Modify Performance Monitoring Job
- Cancel Performance Monitoring Job
- Suspend Performance Monitoring Job
- Resume Performance Monitoring Job being subject of this event.

Name	Type	Description	MEF W133.1
href	string	Hyperlink to access the Performance Job Process	
id*	string	ID of the Performance Job Process	

7.3.11. Type PerformanceJobReportPreparationErrorEvent

Description:

Inherits from:

- Event

Name	Type	Description
eventType*	PerformanceJobReportPreparationErrorEventType	
event*	PerformanceJobReportPreparationErrorEventPayload	

7.3.12. enum PerformanceJobReportPreparationErrorEventType

Description: Indicates the type of Performance Job event.

Value	MEF W133.1
performanceJobReportPreparationErrorEvent	PERFORMANCE_JOB_REPORT_PREPARATION_ER

7.3.13. Type PerformanceJobReportPreparationErrorEventPayload

Description: The identifier of the Performance Job being subject of this event and reason for report preparation failure.

Name	Type	Description
href	string	Hyperlink to access the Performance Job
id*	string	ID of the Performance Job
reportPreparationFailedReason	string	Reason for Report preparation failure

7.3.14. Type PerformanceJobReportReadyEvent

Description:

Inherits from:

- Event

Name	Type	Description	MEF W.133.1
eventType*	PerformanceJobReportReadyEventType		
event*	PerformanceJobReportReadyEventPayload		

7.3.15. enum PerformanceJobReportReadyEventType

Description: Indicates the type of Performance Job event.

Value	MEF W133.1
performanceJobReportReadyEvent	PERFORMANCE_JOB_REPORT_READY_EVENT

7.3.16. Type PerformanceJobReportReadyEventPayload

Description: The identifier of the Performance Job and Report ID being subjects of this event.

Name	Type	Description	MEF W133.1
href	string	Hyperlink to access the Performance Job	
id*	string	ID of the Performance Job	
reportHref	string	Hyperlink to access the Performance Report	
reportId*	string	ID of generated Performance Report	

7.3.17. Type PerformanceReportEvent

Description:

Inherits from:

- Event

Name	Type	Description	MEF W133.1
eventType*	PerformanceReportEventType		
event*	PerformanceReportEventPayload		

7.3.18. enum PerformanceReportEventType

Description: Indicates the type of Performance Report event.

Value	MEF W133.1
performanceReportCreateEvent	PERFORMANCE_REPORT_CREATE_EVENT
performanceReportStateChangeEvent	PERFORMANCE_REPORT_STATE_CHANGE_EVENT

7.3.19. Type PerformanceReportEventPayload

Description: The identifier of the Performance Report being subject of this event.

Name	Type	Description	MEF W133.1
href	string	Hyperlink to access the Performance Report	
id*	string	ID of the Performance Report	

8. References

- [OAS-v3] Open API 3.0, February 2020
- [MEF55.1] MEF 55.1, Lifecycle Service Orchestration (LSO): Reference Architecture and Framework, February 2021
- [MEF128] MEF 128, LSO API Security Profile, July 2022
- [MEF133.1] Allegro, Interlude and Legato Fault Management and Performance Monitoring BR&UC, June 2023
- [MEF152] [MEF 152] Carrier Ethernet Payload Schema/Guide for SOAM
- [MEF153] [MEF 153] IP/IPVPN Schema/Guide for SOAM
- [MEF154] [MEF 154] SD-WAN Schema/Guide for SOAM
- [REST] Chapter 5: Representational State Transfer (REST) Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures (Ph.D.).
- [RFC2119] RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, by S. Bradner, March 1997
- [RFC3986] RFC 3986 Uniform Resource Identifier (URI): Generic Syntax, January 2005
- [RFC8174] RFC 8174, Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, by B. Leiba, May 2017, Copyright © 2017 IETF Trust and the persons identified as the document authors. All rights reserved.
- [TMF628] TMF 628, Performance Management API REST Specification R14.5.1, June 2015
- [TMF630] TMF 630 TMF630 API Design Guidelines 4.2.0

Appendix A Acknowledgments