Mplify Standard
Mplify 143

LSO Allegro, LSO Interlude and LSO Legato
Performance Monitoring API
Developer Guide

February 2026

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 Mplify Alliance (Mplify) is not responsible for any errors. Mplify does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by Mplify concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by Mplify 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. Mplify 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 Mplify member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

b) any warranty or representation that any Mplify members 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 Mplify member and the recipient or user of this document.

Implementation or use of specific Mplify standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in Mplify Alliance. Mplify is a global alliance of network, cloud, cybersecurity, and enterprise organizations working together to accelerate the AI-powered digital economy through standardization, automation, certification, and collaboration. Mplify does not, expressly or otherwise, endorse or promote any specific products or services.

Table of Contents

List of Contributing Members

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

Member
Amartus

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 Mplify 133.1. 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

commit id: e1e192cb7dd9949feaeed17f93bdaaaef97c62e9

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

MEF-LSO-Interlude-SDK

commit id: 6cf196bd5acd0eacca24090c3b7b2f339c1660a8

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

MEF-LSO-Legato-SDK

commit id: 11bb297954521081a0fbcc1b07c6ce806f3d7cec

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

The Performance Monitoring API is defined using Open API 3.0

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 Mplify 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:

- Mplify 133.1 *Allegro, Interlude and Legato Fault Management and Performance Monitoring BR&UC* July 2025 [Mplify 133.1]
- 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 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 an 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.	Mplify 133.1
On-Demand	Performance Monitoring Job actions that are initiated for a limited time to carry out the performance measurements.	Mplify 133.1
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 Server and Client, potentially involving multiple back-and-forth transactions.	This document
Passive	Performance Monitoring Job actions 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, bridging/Ethernet, IP, TCP, and UDP layers.	Mplify 133.1
PM Metric	A metric that is measured or calculated as a part of Performance Monitoring.	MEF 105
Proactive	Performance Monitoring Job actions that are carried on continuously to permit timely reporting of performance status.	Mplify 133.1

Term	Definition	Source
Representational State Transfer	Representational State Transfer 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
SchemaObject	The construct that allows the definition of input and output data types. These types can represent object classes, as well as primitives and array specifications.	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	MEF 55.1
OAS	OpenAPI Specification	openapis.org
PM	Performance Monitoring	Mplify 133.1
REST	Representational State Transfer	REST
SLA	Service Level Agreement	MEF 10.3
SLS	Service Level Specification	MEF 10.3
SOF	Service Orchestration Functionality	MEF 55.1
SP	Service Provider	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, RFC 8174) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

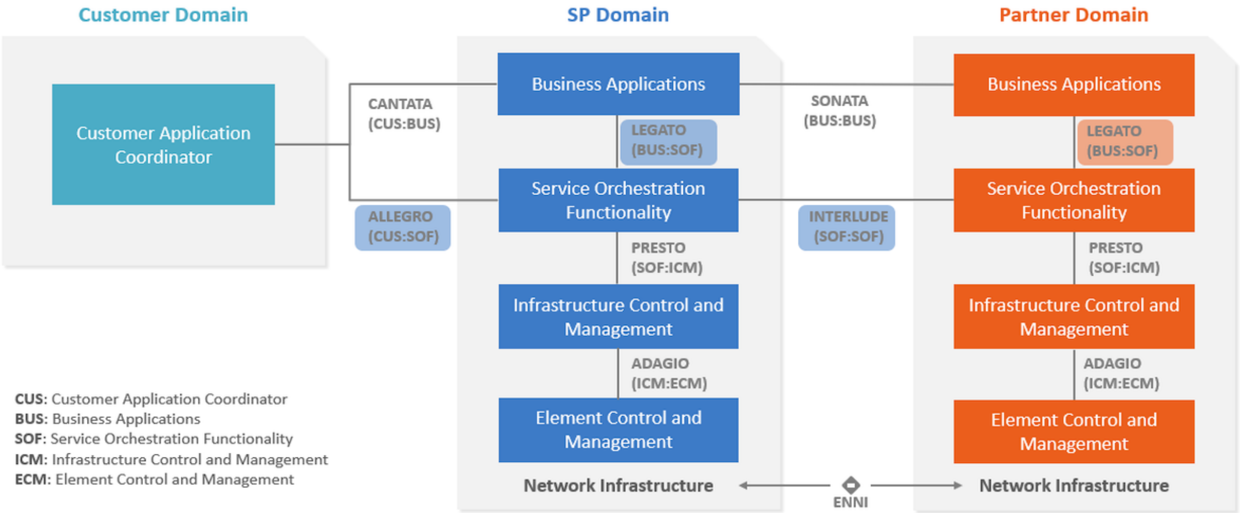
Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words MAY or OPTIONAL) are labeled 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 (SLS) 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 Service Provided (SP) or Operator providing the service if the objectives are not met. The Performance Monitoring API allows managing Performance Profiles, Performance Jobs, and collecting Performance Reports, as well as receiving notifications related to these entities. This facilitates the management of performance objectives commonly associated with an SLS and enables on-demand troubleshooting when necessary.

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



Customer Domain
SP Domain
Partner Domain
Customer Application Coordinator
Business Applications
Business Applications
CANTATA (CUS:BUS)
SONATA (BUS:BUS)
LEGATO (BUS:SOF)
LEGATO (BUS:SOF)
Service Orchestration Functionality
Service Orchestration Functionality
ALLEGRO (CUS:SOF)
INTERLUDE (SOF:SOF)
PRESTO (SOF:ICM)
PRESTO (SOF:ICM)
Infrastructure Control and Management
Infrastructure Control and Management
ADAGIO (ICM:ECM)
ADAGIO (ICM:ECM)
Element Control and Management
Element Control and Management
Network Infrastructure
ENNI
Network Infrastructure
CUS: Customer Application Coordinator
BUS: Business Applications
SOF: Service Orchestration Functionality
ICM: Infrastructure Control and Management
ECM: Element Control and Management

Figure 1. The LSO Reference Architecture

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 relationships 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 between 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

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 the scope of this standard) across the Allegro IRP, Interlude IRP, and Legato IRP.

The Performance Monitoring API allows the Buyer (CUS/SOF) 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.

This standard is based on TMF Open API (v5.0) for Performance Management TMF 628.

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 Mplify 133.1. The API definition builds on *TMF 628 Performance Management API REST Specification v5.0* [TMF 628]. Performance Monitoring Use Cases must support the use of Mplify 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 (Mplify service specification data model)

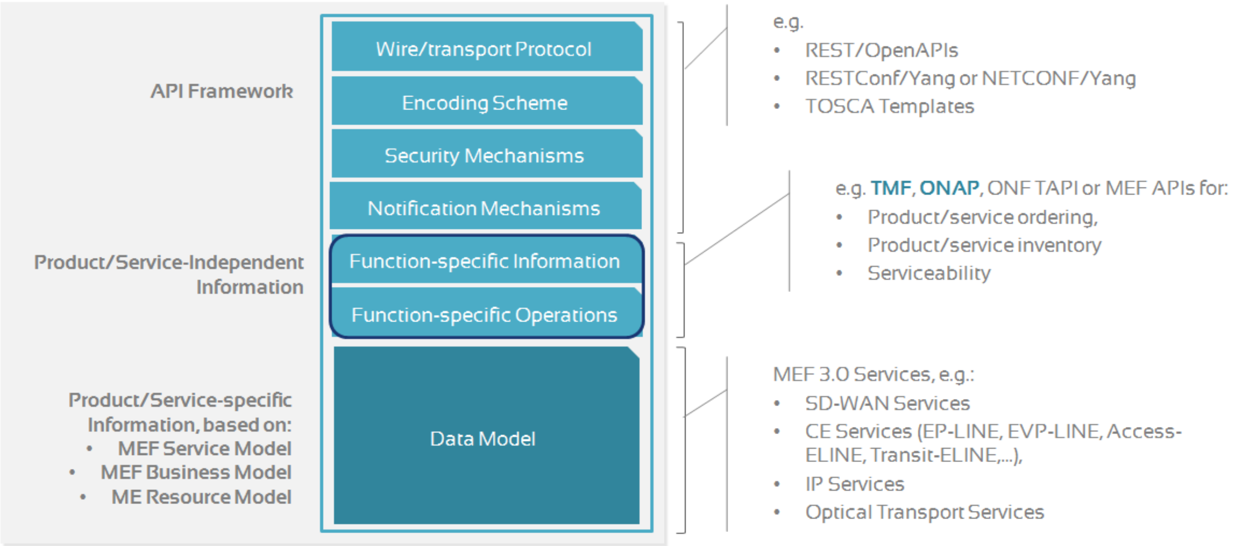

API Framework
Product/Service-Independent Information
Product/Service-specific Information, based on:
• MEF Service Model
• MEF Business Model
• ME Resource Model
Wire/transport Protocol
Encoding Scheme
Security Mechanisms
Notification Mechanisms
Function-specific Information
Function-specific Operations
Data Model
e.g.
• REST/OpenAPIs
• RESTConf/Yang or NETCONF/Yang
• TOSCA Templates
e.g. TMF, ONAP, ONF TAPI or MEF APIs for:
• Product/service ordering,
• Product/service inventory
• Serviceability
MEF 3.0 Services, e.g.:
• SD-WAN Services
• CE Services (EP-LINE, EVP-LINE, Access-ELINE, Transit-ELINE,...),
• IP Services
• Optical Transport Services

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, ordered pair, or entity. 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 does not define Mplify service performance specifications but can be used in combination with any performance specifications defined by or compliant with Mplify. Service Performance schemas are defined in other Mplify documents.

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

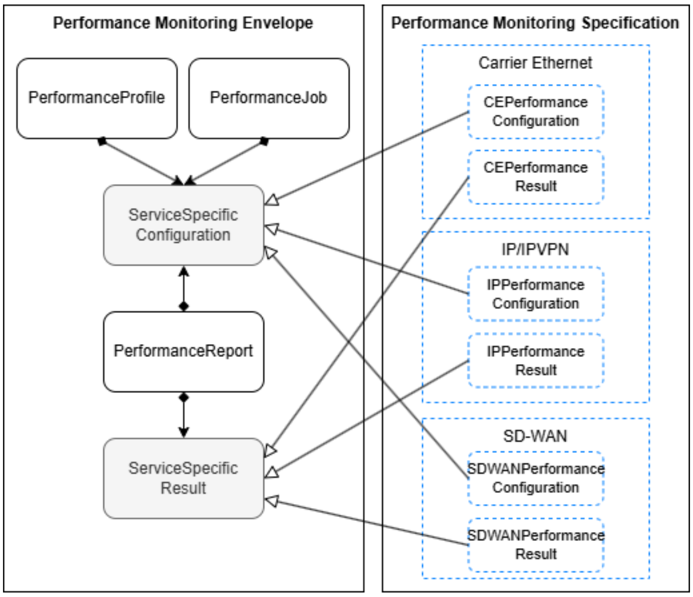

Performance Monitoring Envelope
PerformanceProfile
PerformanceJob
ServiceSpecific Configuration
PerformanceReport
ServiceSpecific Result
Performance Monitoring Specification
Carrier Ethernet
CEPerformance Configuration
CEPerformance Result
IP/IPVPN
IPPerformance Configuration
IPPerformance Result
SD-WAN
SDWANPerformance Configuration
SDWANPerformance Result

Figure 3. Performance specification for Allegro, Interlude, Legato

4.5. High-Level Flow

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

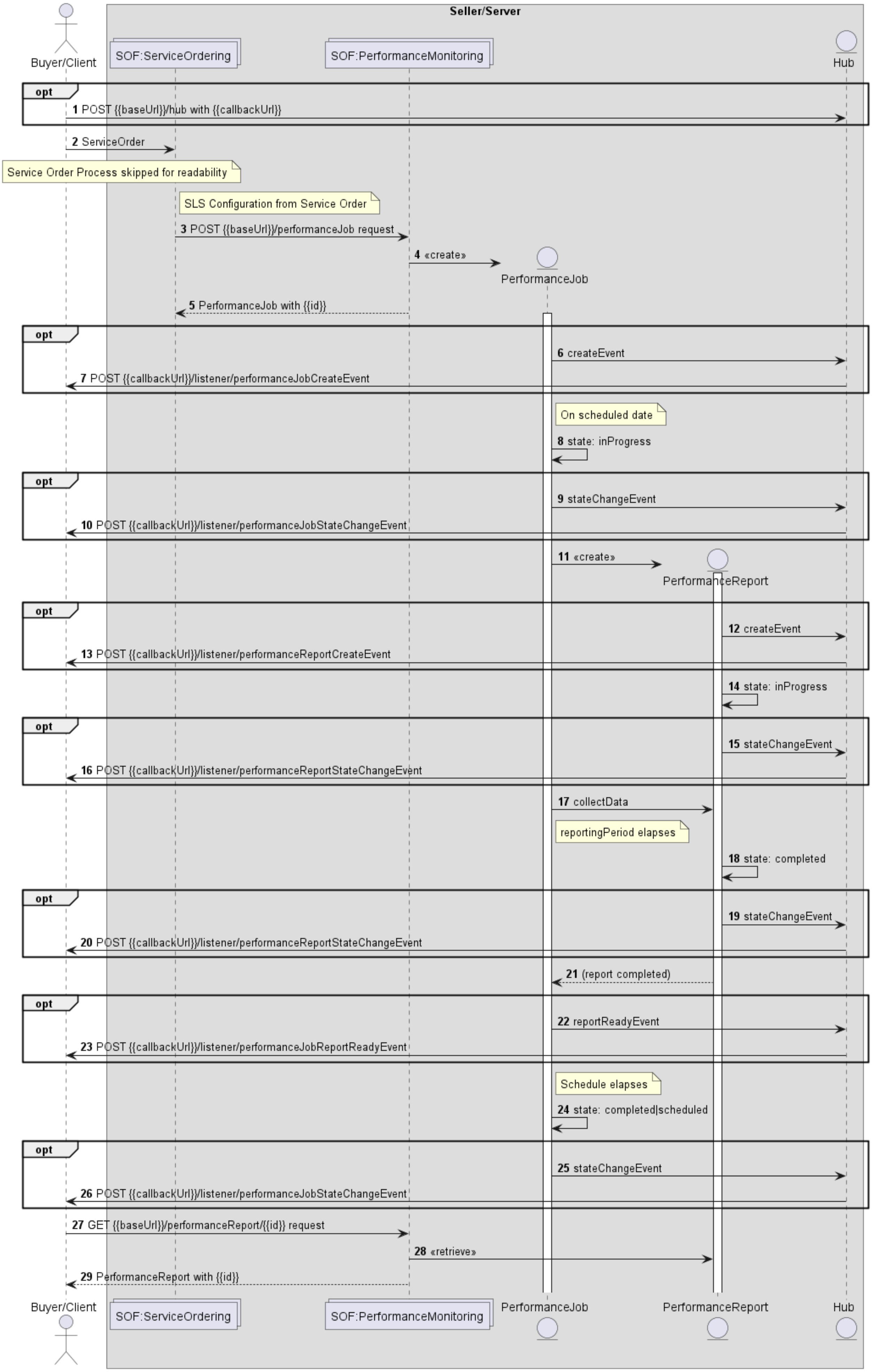

Seller/Server
Buyer/Client
SOF:ServiceOrdering
SOF:PerformanceMonitoring
Hub
opt
1 POST {{baseUrl}}/hub with {{callbackUrl}}
2 ServiceOrder
Service Order Process skipped for readability
SLS Configuration from Service Order
3 POST {{baseUrl}}/performanceJob request
4 «create»
PerformanceJob
5 PerformanceJob with {{id}}
opt
6 createEvent
7 POST {{callbackUrl}}/listener/performanceJobCreateEvent
On scheduled date
8 state: inProgress
opt
9 stateChangeEvent
10 POST {{callbackUrl}}/listener/performanceJobStateChangeEvent
11 «create»
PerformanceReport
opt
12 createEvent
13 POST {{callbackUrl}}/listener/performanceReportCreateEvent
14 state: inProgress
opt
15 stateChangeEvent
16 POST {{callbackUrl}}/listener/performanceReportStateChangeEvent
17 collectData
reportingPeriod elapses
18 state: completed
opt
19 stateChangeEvent
20 POST {{callbackUrl}}/listener/performanceReportStateChangeEvent
21 (report completed)
opt
22 reportReadyEvent
23 POST {{callbackUrl}}/listener/performanceJobReportReadyEvent
Schedule elapses
24 state: completed|scheduled
opt
25 stateChangeEvent
26 POST {{callbackUrl}}/listener/performanceJobStateChangeEvent
27 GET {{baseUrl}}/performanceReport/{{id}} request
28 «retrieve»
29 PerformanceReport with {{id}}
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 Buyer/Client system registers for notifications.
  - Performance Notifications are optional and do not impact end-to-end flow
- Service ordering flow in the diagram is simplified and is only supposed to show that in case of SLS included in the service order, a corresponding `PerformanceJob` is provisioned internally.
- During provisioning of performance monitoring, the SOF internally instantiates the 'PerformanceJob'
  - Process of identification of applicable service performance specification schema is out of scope for this standard.
  - `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).
- `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 a scheduled date according to schedule definition, a `PerformanceReport` entity is created and performance data generation is started.
- Performance data is collected in the `PerformanceReport`.
- `PerformanceReport` is processed as per the state transition rules described in 6.18.4.
- (optional) The SOF reports the `PerformanceJob` and `PerformanceReport` state changes.
- The Buyer/Client system can collect `PerformanceReport` through *Performance Monitoring API*

The same *Performance Monitoring API* is used by the Buyer/Client to create **new** `PerformanceJob` instances, as well as update **existing** ones or trigger state transitions (e.g. cancel **existing** `PerformanceJob` instance)

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

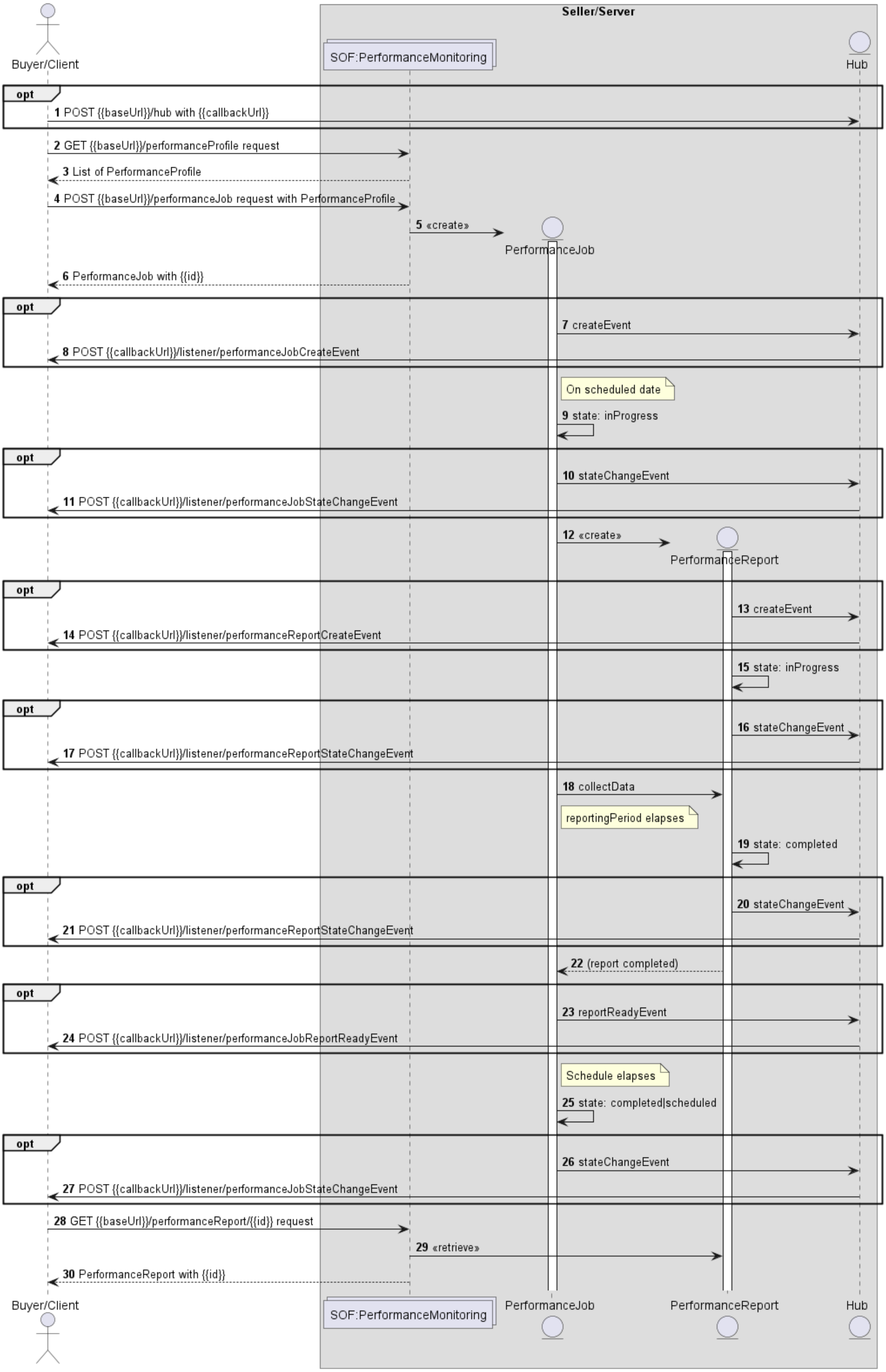

Seller/Server
Buyer/Client
SOF:PerformanceMonitoring
Hub
opt
1 POST {{baseUrl}}/hub with {{callbackUrl}}
2 GET {{baseUrl}}/performanceProfile request
3 List of PerformanceProfile
4 POST {{baseUrl}}/performanceJob request with PerformanceProfile
5 «create»
PerformanceJob
6 PerformanceJob with {{id}}
opt
7 createEvent
8 POST {{callbackUrl}}/listener/performanceJobCreateEvent
On scheduled date
9 state: inProgress
opt
10 stateChangeEvent
11 POST {{callbackUrl}}/listener/performanceJobStateChangeEvent
12 «create»
PerformanceReport
opt
13 createEvent
14 POST {{callbackUrl}}/listener/performanceReportCreateEvent
15 state: inProgress
opt
16 stateChangeEvent
17 POST {{callbackUrl}}/listener/performanceReportStateChangeEvent
18 collectData
reportingPeriod elapses
19 state: completed
opt
20 stateChangeEvent
21 POST {{callbackUrl}}/listener/performanceReportStateChangeEvent
22 (report completed)
opt
23 reportReadyEvent
24 POST {{callbackUrl}}/listener/performanceJobReportReadyEvent
Schedule elapses
25 state: completed|scheduled
opt
26 stateChangeEvent
27 POST {{callbackUrl}}/listener/performanceJobStateChangeEvent
28 GET {{baseUrl}}/performanceReport/{{id}} request
29 «retrieve»
30 PerformanceReport with {{id}}
Buyer/Client
SOF:PerformanceMonitoring
PerformanceJob
PerformanceReport
Hub

Figure 5. High-Level Flow for non-SLS case or on-demand monitoring

This flow applies when a Performance Job is created not as a result of the SLS attached to the service, or when a Performance Job is created alongside an existing Job associated with the SLS (for example, in on-demand monitoring scenarios). Provisioning of `PerformanceJob` by the Buyer/Client occurs separately from service ordering and only when a Performance Job is needed.

- The Buyer/Client 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 Buyer/Client system uses the *Performance Monitoring API*.
- The rest of the flow is the 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.

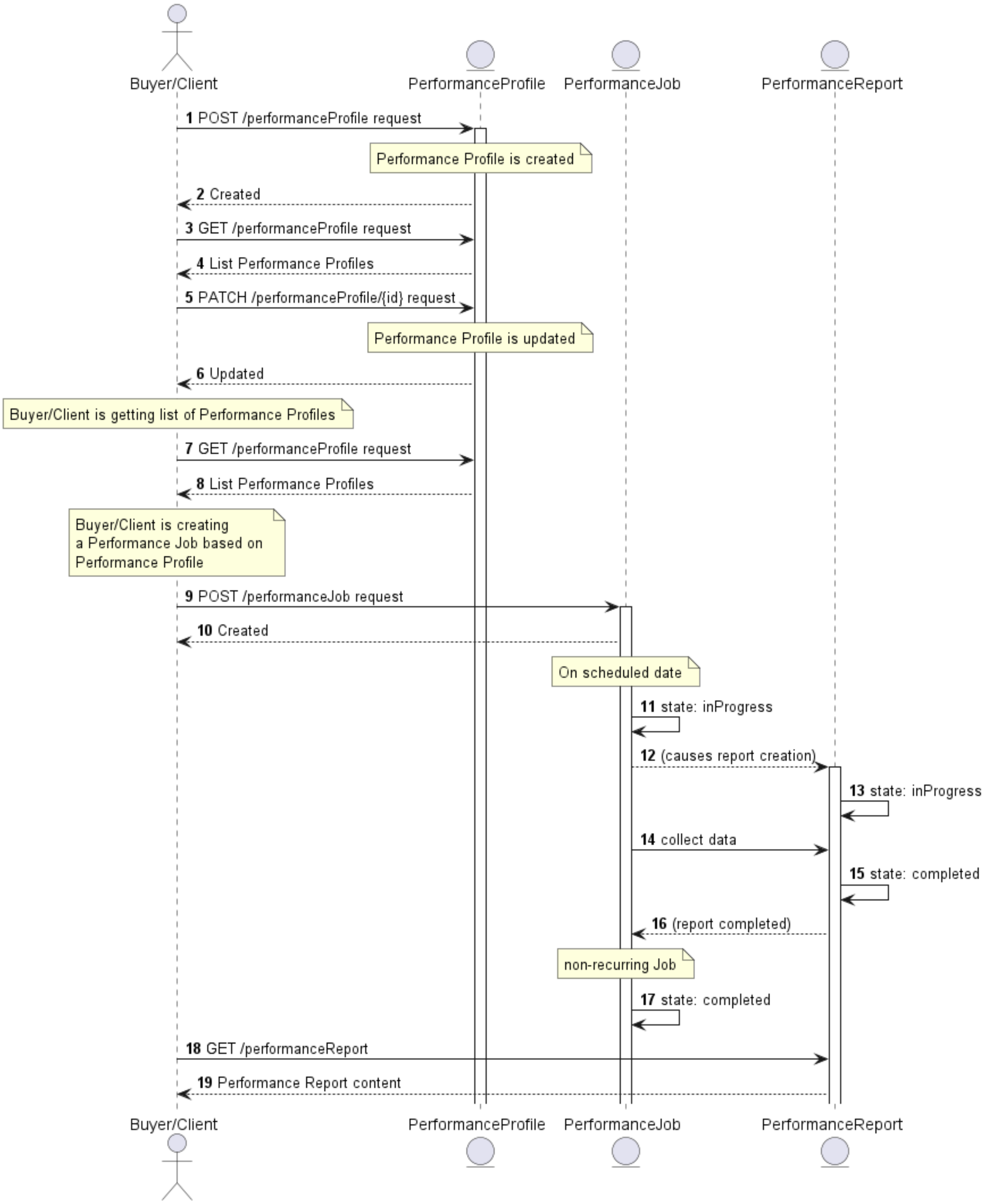

Buyer/Client
PerformanceProfile
PerformanceJob
PerformanceReport
1 POST /performanceProfile request
Performance Profile is created
2 Created
3 GET /performanceProfile request
4 List Performance Profiles
5 PATCH /performanceProfile/{id} request
Performance Profile is updated
6 Updated
Buyer/Client is getting list of Performance Profiles
7 GET /performanceProfile request
8 List Performance Profiles
Buyer/Client is creating
a Performance Job based on
Performance Profile
9 POST /performanceJob request
10 Created
On scheduled date
11 state: inProgress
12 (causes report creation)
13 state: inProgress
14 collect data
15 state: completed
16 (report completed)
non-recurring Job
17 state: completed
18 GET /performanceReport
19 Performance Report content
Buyer/Client
PerformanceProfile
PerformanceJob
PerformanceReport

Figure 6. The 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 explains 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

Figures 7-10 present high-level use case diagrams. They aim to help understand the endpoint mapping. Use cases are described extensively in chapter 6. The mandatory use cases are highlighted in bold.

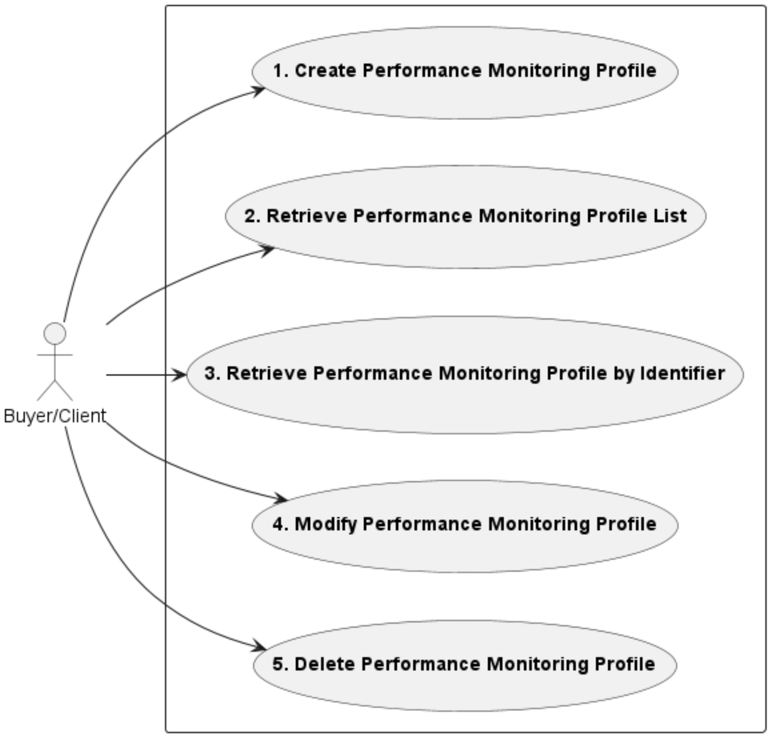

1. Create Performance Monitoring Profile
2. Retrieve Performance Monitoring Profile List
3. Retrieve Performance Monitoring Profile by Identifier
Buyer/Client
4. Modify Performance Monitoring Profile
5. Delete Performance Monitoring Profile

Figure 7. Use cases - Performance Profile

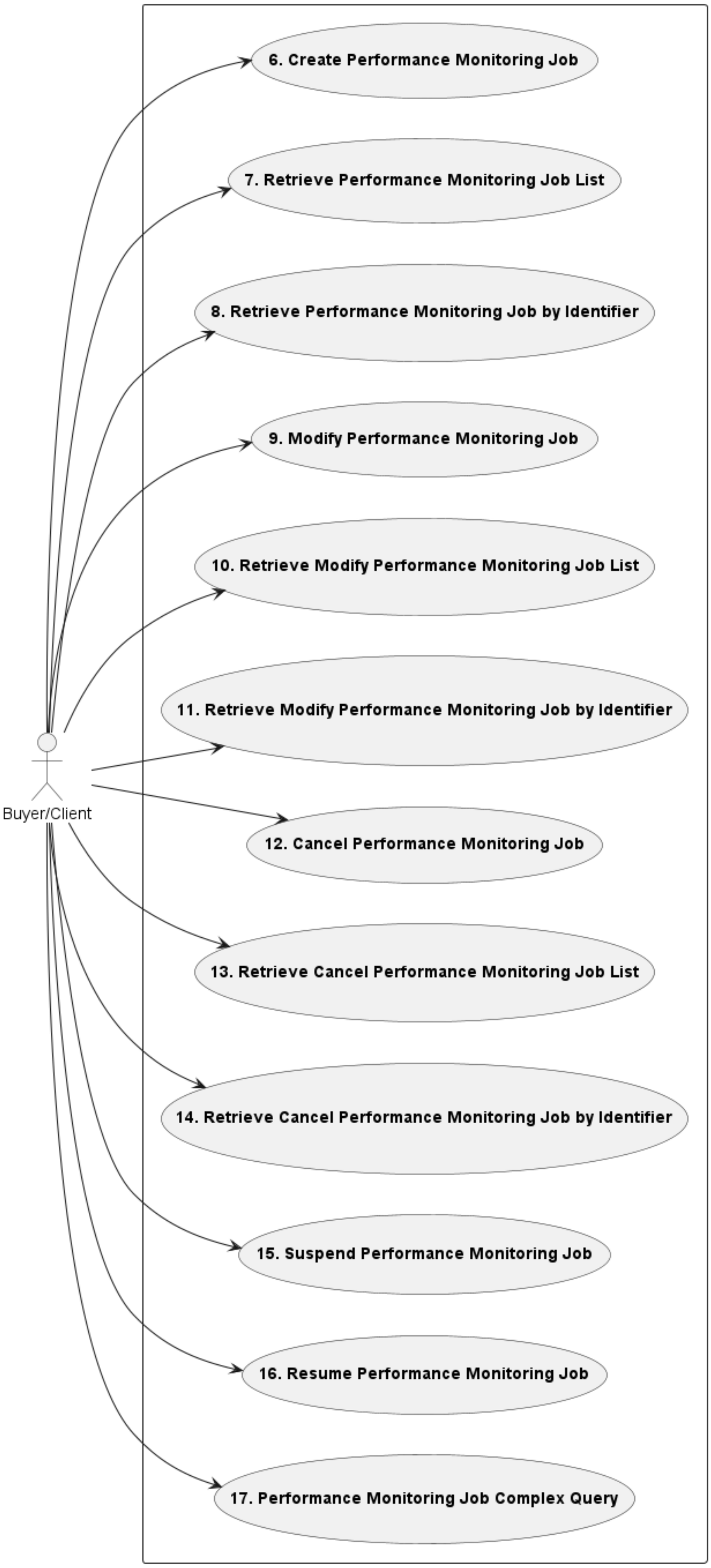

6. Create Performance Monitoring Job
7. Retrieve Performance Monitoring Job List
8. Retrieve Performance Monitoring Job by Identifier
9. Modify Performance Monitoring Job
10. Retrieve Modify Performance Monitoring Job List
11. Retrieve Modify Performance Monitoring Job by Identifier
Buyer/Client
12. Cancel Performance Monitoring Job
13. Retrieve Cancel Performance Monitoring Job List
14. Retrieve Cancel Performance Monitoring Job by Identifier
15. Suspend Performance Monitoring Job
16. Resume Performance Monitoring Job
17. Performance Monitoring Job Complex Query

Figure 8. Use cases - Performance Job

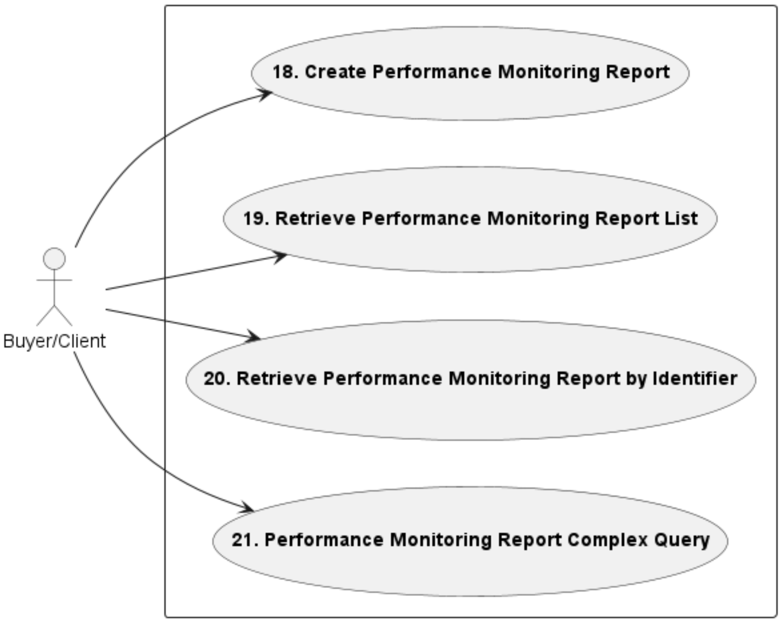

18. Create Performance Monitoring Report
19. Retrieve Performance Monitoring Report List
20. Retrieve Performance Monitoring Report by Identifier
21. Performance Monitoring Report Complex Query
Buyer/Client

Figure 9. Use cases - Performance Report

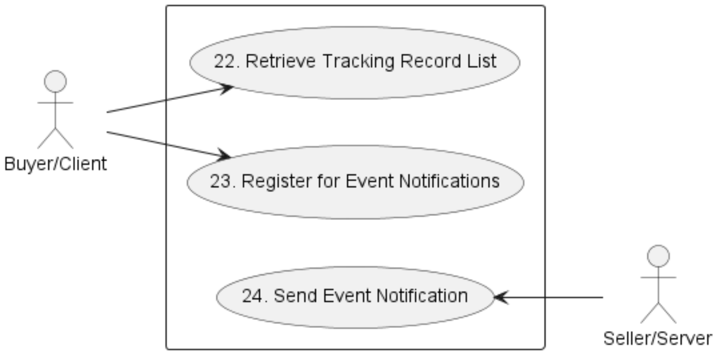

22. Retrieve Tracking Record List
23. Register for Event Notifications
24. Send Event Notification
Buyer/Client
Seller/Server

Figure 10. Use cases - Common

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/v5/`

Base URL for Interlude:

`https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/interlude/performanceMonitoring/v5/`

Base URL for Legato:

`https://{{serverBase}}:{{port}}{{?/sof_prefix}}/mefApi/legato/performanceMonitoring/v5/`

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	Mplify 133.1 Use Case Mapping
POST /performanceProfile	A request initiated by the Buyer/Client to create a Performance Monitoring Profile in the Seller/Server system.	11
GET /performanceProfile	The Buyer/Client requests a list of Performance Monitoring Profiles based on a set of filter criteria.	12
GET /performanceProfile/{id}	The Buyer/Client requests detailed information about a single Performance Monitoring Profile.	13
PATCH /performanceProfile/{id}	A request initiated by the Buyer/Client to modify a Performance Monitoring Profile in the Seller/Server system based on a Performance Monitoring Identifier.	14
DELETE /performanceProfile/{id}	The Buyer/Client requests deletion of Performance Monitoring Profile by specifying Performance Monitoring Identifier.	15
POST /performanceJob	A request initiated by the Buyer/Client to create a Performance Monitoring Job in the Seller/Server system.	19,36
GET /performanceJob	The Buyer/Client requests a list of Performance Monitoring Jobs based on a set of filter criteria.	24
GET /performanceJob/{id}	The Buyer/Client requests detailed information about a single Performance Monitoring Job.	25
POST /modifyPerformanceJob	A request initiated by the Buyer/Client to modify a Performance Monitoring Job in the Seller/Server system.	20,37
GET /modifyPerformanceJob	The Buyer/Client requests a list of Modify Performance Monitoring Job based on a set of filter criteria.	20,37
GET /modifyPerformanceJob/{id}	The Buyer/Client requests detailed information about a single Modify Performance Monitoring Job.	20,37
POST /cancelPerformanceJob	A request initiated by the Buyer/Client to cancel a Performance Monitoring Job in the Seller/Server system.	21,38
GET /cancelPerformanceJob	The Buyer/Client requests a list of Cancel Performance Monitoring Job based on a set of filter criteria.	21,38
GET /cancelPerformanceJob/{id}	The Buyer/Client requests detailed information about a single Cancel Performance Monitoring Job.	21,38

API Endpoint	Description	Mplify 133.1 Use Case Mapping
POST /performanceJob/{id}/suspend	A request initiated by the Buyer/Client to suspend a Performance Monitoring Job in the Seller/Server system.	22
POST /performanceJob/{id}/resume	A request initiated by the Buyer/Client to resume a Performance Monitoring Job in the Seller/Server system.	23
POST /performanceJobComplexQuery	A request initiated by the Buyer/Client to create a Performance Monitoring Job Complex Query in the Seller/Server system.	24
POST /performanceReport	A request initiated by the Buyer/Client to collect data points existing in the Seller/Server system in the form of a Performance Monitoring Report.	55
GET /performanceReport	The Buyer/Client requests a list of Performance Monitoring Reports based on a set of filter criteria.	29,39
GET /performanceReport/{id}	The Buyer/Client requests detailed information about a single Performance Monitoring Report, including the content of the report.	30,40
POST /performanceReportComplexQuery	A request initiated by the Buyer/Client to create a Performance Monitoring Report Complex Query in the Seller/Server system.	29,39

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

[R1] Seller/Server (SOF) **MUST** support all API endpoints listed in Table 4. [Mplify133.1 R45, R49, R52, R53, R63, R85, R97, R115]

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

API Endpoint	Description	Mplify 133.1 Use Case Mapping
GET /trackingRecord	The Buyer/Client requests a list of Tracking Records based on a set of filter criteria.	56
POST /hub	The Buyer/Client requests to subscribe to the Performance Monitoring Profile, Performance Monitoring Job, and/or Performance Monitoring Report Notifications.	16,26
GET /hub/{id}	The Buyer/Client retrieves a specific `EventSubscription` from the SOF, that matches the *`id`* value provided as *`path`* parameter.	16,26
DELETE /hub/{id}	The Buyer/Client requests to unsubscribe from the Performance Monitoring Profile, Performance Monitoring Job, and/or Performance Monitoring Report Notifications.	18,27

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

[O1] The implementation **MAY** support API endpoints listed in Table 5. [Mplify133.1 O5, O7, O9]

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

Base URL for Allegro:

```
https://{{serverBase}}:{{port}}
{{?/bus_cus_sof_prefix}}/mefApi/allegro/performanceNotification/v5/
```

Base URL for Interlude:

```
https://{{serverBase}}:{{port}}
{{?/bus_cus_sof_prefix}}/mefApi/interlude/performanceNotification/v5/
```

Base URL for Legato:

```
https://{{serverBase}}:{{port}}
{{?/bus_cus_sof_prefix}}/mefApi/legato/performanceNotification/v5/
```

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 `serviceApi/pm/performanceNotification.api.yaml`

API Endpoint	Description	Mplify 133.1 Use Case Mapping
POST /listener/performanceJobCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceJob` instance creation.	28
POST /listener/performanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `PerformanceJob` instance state change.	28
POST /listener/performanceJobAttributeValueChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `PerformanceJob` instance attribute value change.	28
POST /listener/performanceJobReportReadyEvent	A request initiated by the Seller/Server to notify Buyer/Client that `PerformanceReport` was generated for the `PerformanceJob` instance.	28

API Endpoint	Description	Mplify 133.1 Use Case Mapping
POST /listener/performanceJobReportPreparationErrorEvent	A request initiated by the Seller/Server to notify Buyer/Client that `PerformanceReport` was not generated for the `PerformanceJob` instance due to an error.	28
POST /listener/cancelPerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `CancelPerformanceJob` instance state change.	28
POST /listener/modifyPerformanceJobStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `ModifyPerformanceJob` instance state change.	28
POST /listener/performanceProfileCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance creation.	17
POST /listener/performanceProfileAttributeValueChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `PerformanceProfile` instance attribute value change.	17
POST /listener/performanceProfileDeleteEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceProfile` instance deletion.	17
POST /listener/performanceReportCreateEvent	A request initiated by the Seller/Server to notify Buyer/Client on `PerformanceReport` instance creation.	28
POST /listener/performanceReportStateChangeEvent	A request initiated by the Seller/Server to notify Buyer/Client on the `PerformanceReport` instance state change.	28

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. [Mplify133.1 O8]

[R2] The Seller/Server **MUST** support sending notifications to API endpoints listed in Table 6 to the registered Buyer/Client.

5.3. Integration of Service Monitoring Specification into Performance Monitoring API

The Performance Monitoring API is a generic envelope that allows for the lifecycle management of relevant performance monitoring objects. The API itself does not provide particular technology-specific 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 TM Forum extension pattern.

The extension hosting types in the API data model are:

- `ServiceSpecificConfiguration` - this type is extended with monitoring configuration schema
- `ServiceSpecificResult` - this type is extended with 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 Mplify standard service schemas is in URN format and is assigned by Mplify. This identifier is provided as root schema `$id`. Use of non-Mplify 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
```

Payload for configuring service performance monitoring is used in multiple PM API entities through a `serviceSpecificConfiguration` attribute of type `ServiceSpecificConfiguration`. It is an extension point for configuration attributes.

In terms of monitoring results, the appropriate payload is introduced via `ReportContent`. This entity has a `measurementData` array of items of type `ServiceSpecificResult` which is used as an extension point for service-specific output content.

Implementations might choose to integrate selected performance monitoring specifications into the data model during development. In such a case an integrated data model contains monitoring specifications which are in an inheritance relationship accordingly with either `ServiceSpecificConfiguration` or `ServiceSpecificResult` 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 Mplify monitoring specifications, for both extension points.

```
ServiceSpecificConfiguration:
  type: object
  description:
    ServiceSpecificConfiguration is used as an extension point for
    Mplify-specific service performance monitoring configuration. 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:
        Uniquely identifies the type of performance monitoring configuration
        that specifies PM objectives. In the case of Mplify services, this is
        the URN provided in the performance monitoring configuration
        specification. The named type must be a subclass of
        ServiceSpecificConfiguration.
```

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

```
ServiceSpecificResult:
  type: object
  description:
    ServiceSpecificResult is used as an extension point for Mplify 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 Mplify
        services this is the URN provided in performance monitoring results
        specification. The named type must be a subclass of
        ServiceSpecificResult.
```

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

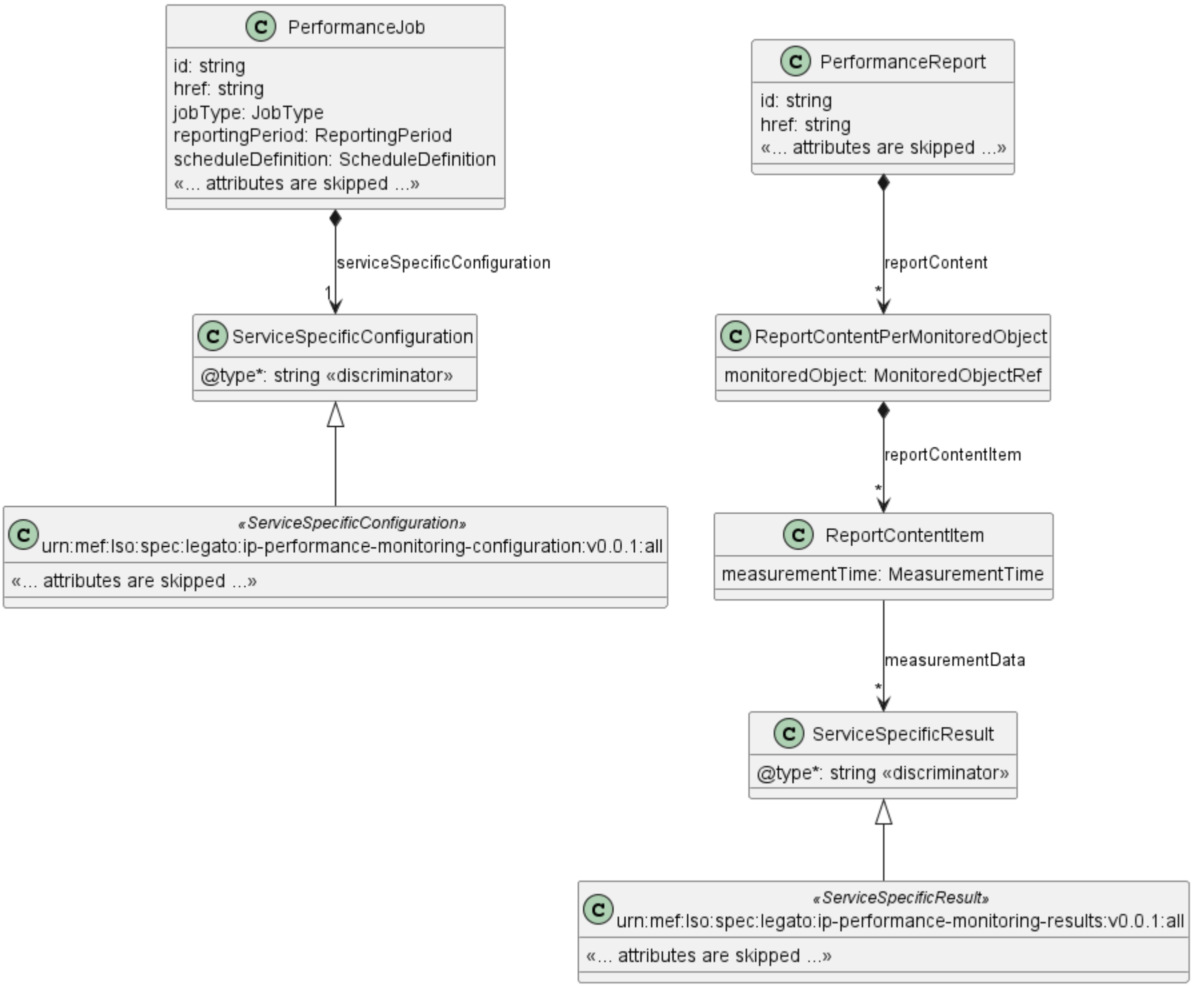
Alternatively, implementations might choose not to build an integrated model and choose a different mechanism allowing runtime validation of service-specific fragments of the payload. In this case systems can validate a given monitoring configuration against a new schema without redeployment. This pattern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

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

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

[R4] The @type property of ServiceSpecificConfiguration and ServiceSpecificResult **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.



PerformanceJob
id: string
href: string
jobType: JobType
reportingPeriod: ReportingPeriod
scheduleDefinition: ScheduleDefinition
«... attributes are skipped ...»
serviceSpecificConfiguration
1
ServiceSpecificConfiguration
@type*: string «discriminator»
«ServiceSpecificConfiguration»
urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all
«... attributes are skipped ...»
PerformanceReport
id: string
href: string
«... attributes are skipped ...»
reportContent
*
ReportContentPerMonitoredObject
monitoredObject: MonitoredObjectRef
reportContentItem
*
ReportContentItem
measurementTime: MeasurementTime
measurementData
*
ServiceSpecificResult
@type*: string «discriminator»
«ServiceSpecificResult»
urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all
«... attributes are skipped ...»

Figure 11. The Extension Pattern with Sample Service-Specific Extension

Figure 11 presents two Mplify performance monitoring schemas that represent configuration and result for IP services. When these schemas are used, the @type of ServiceSpecificConfiguration takes value of "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all" to indicate which performance specification should be used to interpret a set of service-specific attributes included in the payload. Similarly, for ServiceSpecificResult, the @type attribute takes "urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all" value which indicates how the output 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.

[R7] The Buyer/Client and the Seller/Server **MUST NOT** use any operation, entity or attribute that is not explicitly defined or allowed by this standard.

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 128.1 *LSO API Security Profiles* [MEF 128.1].

6. API Interactions and Flows

This section provides a detailed insight into the API functionality, use cases, and flows. It starts with Table 7 showing a list and short description of all business use cases, then presents the variants of end-to-end interaction flows, and in the subchapters describes the API usage flows 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 Buyer/Client to create a Performance Monitoring Profile in the Seller/Server system.
2	Retrieve Performance Monitoring Profile List	The 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 Identifier	The Buyer/Client requests detailed information about a single Performance Monitoring Profile based on the Performance Monitoring Identifier.
4	Modify Performance Monitoring Profile	A request initiated by the Buyer/Client to modify a Performance Monitoring Profile in the Seller/Server system based on a Performance Monitoring Identifier.
5	Delete Performance Monitoring Profile	The Buyer/Client requests deletion of the Performance Monitoring Profile by specifying the Performance Monitoring 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 Job based on a set of filter criteria. The Seller/Server returns a summarized list of PM Jobs.
8	Retrieve Performance Monitoring Job by Identifier	The Buyer/Client requests detailed information about a single Performance Monitoring Job based on the Performance Monitoring Job Identifier.
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 Job based on a set of filter criteria.
11	Retrieve Modify Performance Monitoring Job 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.

Use Case #	Use Case Name	Use Case Description
13	Retrieve Cancel Performance Monitoring Job List	The Buyer/Client requests a list of Cancel Performance Monitoring Job based on a set of filter criteria.
14	Retrieve Cancel Performance Monitoring Job 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	Resume Performance Monitoring Job	A request initiated by the Buyer/Client to resume a Performance Monitoring Job in the Seller/Server system.
17	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.
18	Create Performance Monitoring Report	A request initiated by the Buyer/Client to collect data points existing in the Seller/Server system in the form of a Performance Monitoring Report.
19	Retrieve Performance Monitoring Report List	The Buyer/Client requests a list of Performance Monitoring Reports based on a set of filter criteria. The Seller/Server returns a summarized list of PM Reports .
20	Retrieve Performance Monitoring Report by Identifier	The Buyer/Client requests detailed information, including generated content, about a single Performance Monitoring Report based on the Performance Monitoring Report Identifier.
21	Create Performance Monitoring Report Complex Query	A request initiated by the Buyer/Client to create a Performance Monitoring Report Complex Query in the Seller/Server system.
22	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 list of Tracking Records.
23	Register for Performance Monitoring Notifications	The Buyer/Client requests to subscribe to Performance Monitoring Profile, Performance Monitoring Job, and/or Performance Monitoring Report Notifications.
24	Send Event Notification	A request initiated by the Seller/Server to notify the Buyer/Client.

Table 7. Use cases description

6.1. Use case 1: Create a Performance Monitoring Profile

Performance Monitoring Profile is a template that 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. A Performance

Monitoring Profile can be created by the Buyer/Client according to their needs, or it can be centrally managed by the Seller to facilitate the creation of Performance Jobs.

6.1.1. Interaction flow

The flow of this use case is described in Figure 12.

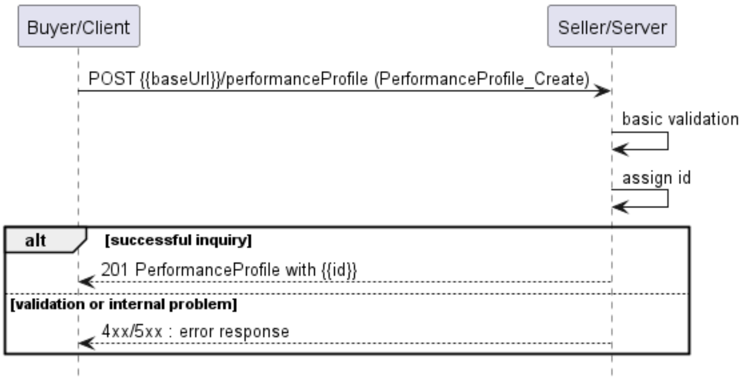

Buyer/Client
Seller/Server
POST {{baseUrl}}/performanceProfile (PerformanceProfile_Create)
basic validation
assign id
alt
[successful inquiry]
201 PerformanceProfile with {{id}}
[validation or internal problem]
4xx/5xx : error response

Figure 12. Use Case 1 - Performance Monitoring Profile create request flow

The Buyer/Client 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.

6.1.2. Create Performance Monitoring Profile Request

Figure 13 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 requestor. The Seller/Server then enriches the entity in the response with additional information.

A `PerformanceProfile_Create` defines configuration details of the `PerformanceJob` that will use the profile as a template. This allows to setup 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.

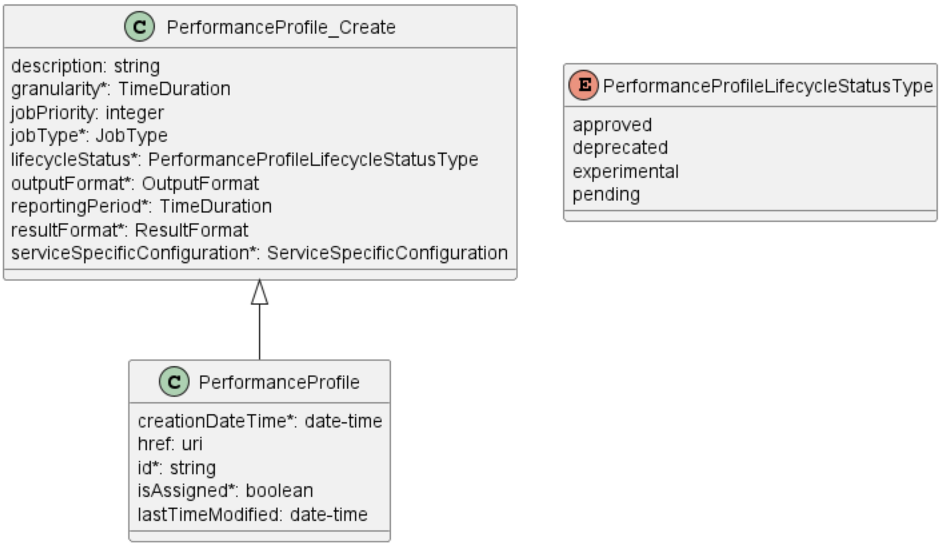

C PerformanceProfile_Create
description: string
granularity*: TimeDuration
jobPriority: integer
jobType*: JobType
lifecycleStatus*: PerformanceProfileLifecycleStatusType
outputFormat*: OutputFormat
reportingPeriod*: TimeDuration
resultFormat*: ResultFormat
serviceSpecificConfiguration*: ServiceSpecificConfiguration
E PerformanceProfileLifecycleStatusType
approved
deprecated
experimental
pending
C PerformanceProfile
creationDateTime*: date-time
href: uri
id*: string
isAssigned*: boolean
lastTimeModified: date-time

Figure 13. 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 a Create Performance Profile request:

`Performance Profile` **Create Request**

```
{
  "description": "Exemplary Create Performance Profile request",
  "granularity": {
    "timeDurationValue": 10,
    "timeDurationUnits": "SEC"
  },
  "jobPriority": 5,
  "jobType": "proactive",
  "lifecycleStatus": "approved",
  "outputFormat": "json",
  "reportingPeriod": {
    "timeDurationValue": 1,
    "timeDurationUnits": "HOUR"
  },
  "resultFormat": "payload",
  "serviceSpecificConfiguration": {
    "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
    "packetsIn": true,
    "charsIn": true,
    "packetsOut": true,
    "charsOut": true
  }
}
```

[R8] The Buyer/Client's Create Performance Profile request **MUST** provide the following attributes: [Mplify133.1 R44]

- `jobType`
- `granularity`
- `reportingPeriod`
- `outputFormat`
- `resultFormat`
- `lifecycleStatus`
- `serviceSpecificConfiguration`

[O3] The Buyer/Client's Create Performance Profile request **MAY** contain the following attributes: [Mplify133.1 O4]

- `description`
- `jobPriority`

6.1.3. Create Performance Monitoring Profile Response

Entities used for providing a response to the Create Performance Profile request are presented in Figure 13. The Seller/Server responds with a `PerformanceProfile` type, which adds some attributes to the `PerformanceProfile_Create` that was used in the Buyer/Client request.

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**

```
{
  "description": "Exemplary Create Performance Profile request",
  "granularity": {
    "timeDurationValue": 10,
    "timeDurationUnits": "SEC"
  },
  "jobPriority": 5,
  "jobType": "proactive",
  "lifecycleStatus": "approved",
  "outputFormat": "json",
  "reportingPeriod": {
    "timeDurationValue": 1,
    "timeDurationUnits": "HOUR"
  },
  "resultFormat": "payload",
  "serviceSpecificConfiguration": {
    "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
    "packetsIn": true,
    "charsIn": true,
    "packetsOut": true,
    "charsOut": true
  },
  "creationDateTimeTimeTime": "2023-06-12T17:47:50.399Z", << added by SOF >>
   "href": "{{baseUrl}}/performanceMonitoring/v5/performanceProfile/8df0981a-0949-11ee-be56-0242ac120002", << added
by SOF >>
  "isAssigned": false, << added by SOF >>
  "id": "8df0981a-0949-11ee-be56-0242ac120002", << added by SOF >>
  "lastTimeModified": "2023-06-12T17:47:50.399Z" << added by SOF >>
}
```

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

[R9] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by the Buyer/Client in the request.

[R10] The Seller/Server **MUST** specify the following attributes in a response:

- `creationDateTimeTime`
- `id`
- `isAssigned`

[R11] The `id` **MUST** remain the same value for the life of the Performance Profile.

6.2. Use Case 2: Retrieve List of Performance Monitoring Profile

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

[O4] The Buyer/Client Retrieve List of Performance Profiles request **MAY** contain none or more of the following attributes as filter criteria:

- `creationDateTimeTime.gt`
- `creationDateTimeTime.lt`
- `jobType`
- `jobPriority`
- `lifecycleStatus`

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 error message that indicates the result set is too large (HTTP status 422 with `tooManyRecords` error code) or

- A response that indicates the result is too large and includes a subset of the matching PM Profiles.

The Buyer may also ask for pagination of the response when the number of results is too big. The following query attributes related to pagination can be provided:

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

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/performanceProfile?limit=20&offset=0
```

The example above shows a Buyer's request to get the first twenty `PerformanceProfile`. The correct response (HTTP code `200`) contains a list of `PerformanceProfile` objects matching the criteria in the response body.

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 then the smaller list of results 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`

[D1] The Seller **SHOULD** support the pagination mechanism.

[CR1]<[D1] Seller **MUST** use either `X-Total-Count` or `X-Pagination-Throttled` to indicate that the page was truncated and additional results are available.

[R12] The Seller/Server's response to the Buyer's/Client's retrieve List of PM Profiles **MUST** include the following attribute as applicable: [Mplify133.1 R47]

- `creationDateTime`
- `description`
- `granularity`
- `id`
- `isAssigned`
- `jobPriority`
- `jobType`
- `lastTimeModified`
- `lifecycleStatus`
- `outputFormat`
- `reportingPeriod`
- `resultFormat`
- `serviceSpecificConfiguration`

[R13] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list. [Mplify133.1 R48]

Get List returns full representation of matched Performance Profiles which includes all attributes as shown in the Create Performance Monitoring Profile Response in section 6.1.3.

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

The Buyer/Client can get detailed information about single Performance Monitoring Profile from the Seller/Server by using a `GET /performanceProfile/{id}` operation. The payload returned in the response is a full representation of the 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.

[R14] In case `id` does not match a `PerformanceProfile` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R15] The Seller/Server's response **MUST** include all attributes of Performance Monitoring Profile.

6.4. Use Case 4: Modify Performance Monitoring Profile

The update operation is realized with the use of the REST PATCH operation (`PATCH /performanceProfile/{id}`). For that purpose, a specialized type `PerformanceProfile_Update` is provided. It consists of attributes limited to a subset that includes only the updateable attributes. The Performance Profile cannot be in use by any Performance Job, otherwise Performance Profile will not be modified.

The PATCH usage recommendation follows RFC 7386 json/merge (https://tools.ietf.org/html/rfc7386).

Figure 14 presents the model used in the PATCH request. The Seller/Server responds with a `PerformanceProfile`.

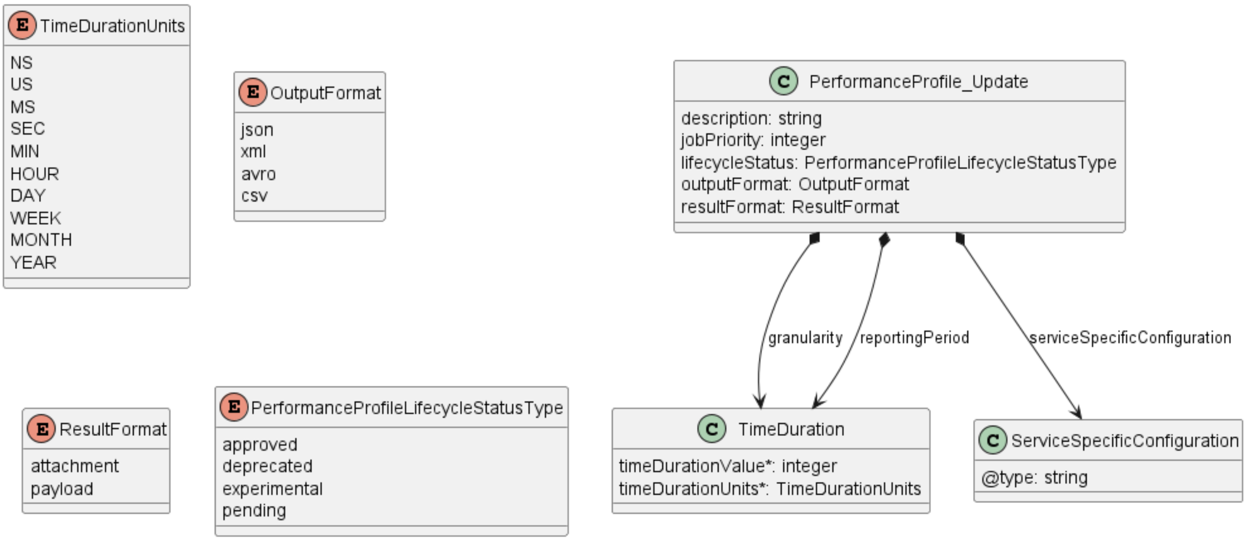

E TimeDurationUnits
NS
US
MS
SEC
MIN
HOUR
DAY
WEEK
MONTH
YEAR
E OutputFormat
json
xml
avro
csv
C PerformanceProfile_Update
description: string
jobPriority: integer
lifecycleStatus: PerformanceProfileLifecycleStatusType
outputFormat: OutputFormat
resultFormat: ResultFormat
granularity
reportingPeriod
serviceSpecificConfiguration
E ResultFormat
attachment
payload
E PerformanceProfileLifecycleStatusType
approved
deprecated
experimental
pending
C TimeDuration
timeDurationValue*: integer
timeDurationUnits*: TimeDurationUnits
C ServiceSpecificConfiguration
@type: string

Figure 14. Performance Profile Patch request Model

[R16] The Buyer/Client **MUST** be able to modify all Buyer/Client settable attributes except the PM Job Type. [Mplify133.1 R51]

[R17] In case `id` does not match a `PerformanceProfile` that is to be updated in Seller/Server's system, an error response `Error404` **MUST** be returned.

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 the `granularity` of the performance measurements collection
- change `reportingPeriod` which is the frequency of report generation

```
{
  "description": "updated description",
  "granularity": {
    "timeDurationValue": 5,
    "timeDurationUnits": "MIN"
  },
  "reportingPeriod": {
    "timeDurationValue": 30,
    "timeDurationUnits": "MIN"
  }
}
```

[R18] Modification of Performance Profile **MUST** be allowed only for profiles created by the Buyer.

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.

The sequence diagram below presents this use case in detail.

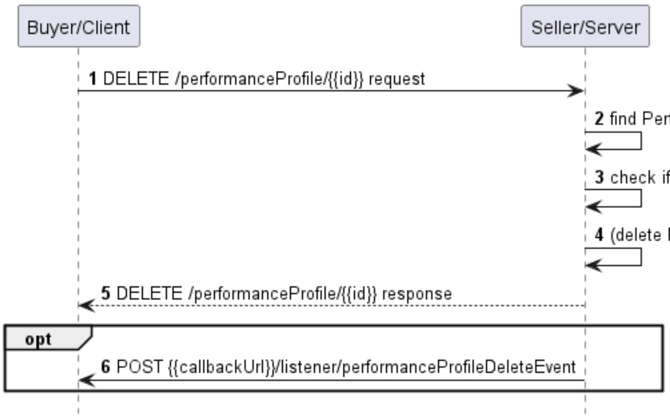

Buyer/Client
Seller/Server
1 DELETE /performanceProfile/{{id}} request
2 find Performance Profile by id
3 check if any Performance Job refers to Performance Profile
4 (delete Performance Profile resource)
5 DELETE /performanceProfile/{{id}} response
opt
6 POST {{callbackUrl}}/listener/performanceProfileDeleteEvent

Figure 15. Delete Performance Profile Flow

The Seller/Server verifies the request, then searches for a Performance Profile to be deleted by the given `id`. The Seller/Server checks if there are any active Performance Job objects that refer to the Performance Profile (active means the state of `PerformanceJob` is different from `rejected`, `completed`, `cancelled`, or `resourcesUnavailable`). If everything is verified correctly, the Seller deletes the Performance Profile, sends a successful response to a request followed by `performanceProfileDeleteEvent` in case the Buyer/Client subscribed for relevant notifications.

[R19] The Seller/Server **MUST** return an error (`Error422`) if the Performance Profile is referenced by an active `PerformanceJob` (active means a state of `PerformanceJob` is different from `rejected`, `completed`, `cancelled`, or `resourcesUnavailable`)

[R20] In case there is no `PerformanceProfile` with provided `id`, an error response `Error404` **MUST** be returned.

[R21] Deletion of Performance Profile **MUST** be allowed only for profiles created by the Buyer.

6.6. Use Case 6: Create a 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 a service, an ordered pair (an association between two endpoints, e.g. UNIs) or an entity (defined as an object other than a service that can be monitored and has 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 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 taking measurements as part of a troubleshooting procedure. The Performance Monitoring Job also provides 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 that are not related to objectives defined by SLS.

The Performance Monitoring Jobs produce Performance Monitoring Reports that provide the Buyer/Client with performance monitoring 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, and UDP layers.

In case of services with SLS attached, Proactive Performance Job is created by the Seller/Server as part of the provisioning process.

Proactive, On-Demand, and Passive Performance Jobs can use Performance Monitoring Profiles as templates for the provisioning. In case Performance Monitoring Job is created without relationship to Performance Profile, all necessary attributes have to be provided in Performance Job creation request. Create Performance Job request can refer to attributes of the Performance Profile by:

- reference - direct reference to existing Performance Profile by its id, or
- value - assigning characteristics typically associated with the Performance Profile model directly in the Create Performance Job request.

[O5] 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 16.

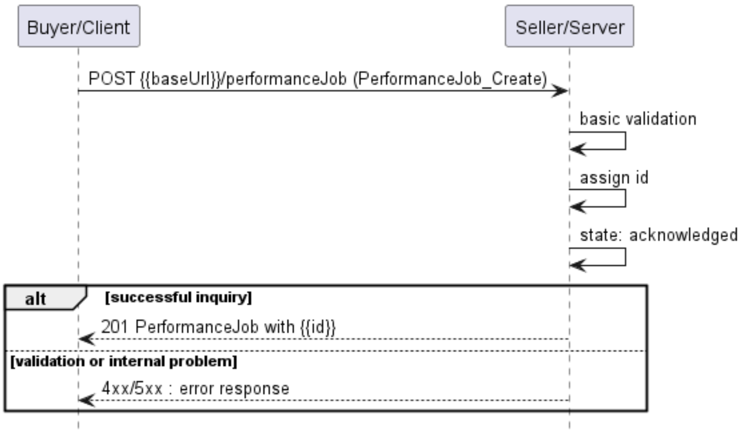

Buyer/Client
Seller/Server
POST {{baseUrl}}/performanceJob (PerformanceJob_Create)
basic validation
assign id
state: acknowledged
alt
[successful inquiry]
201 PerformanceJob with {{id}}
[validation or internal problem]
4xx/5xx : error response

Figure 16. 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 the `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.

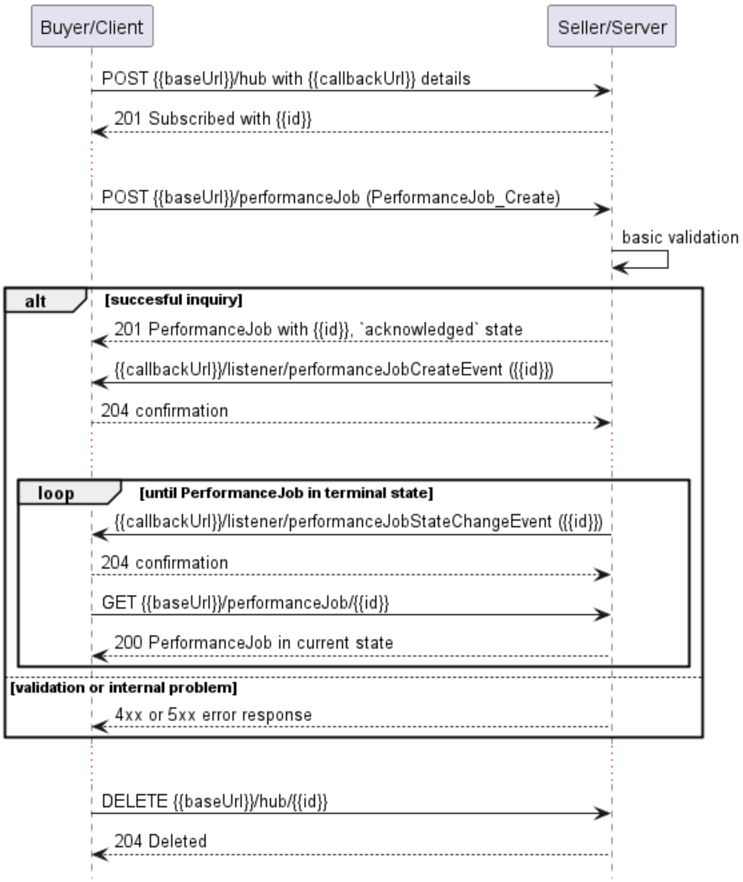

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}} details
201 Subscribed with {{id}}
POST {{baseUrl}}/performanceJob (PerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 PerformanceJob with {{id}}, `acknowledged` state
{{callbackUrl}}/listener/performanceJobCreateEvent ({{id}})
204 confirmation
loop
[until PerformanceJob in terminal state]
{{callbackUrl}}/listener/performanceJobStateChangeEvent ({{id}})
204 confirmation
GET {{baseUrl}}/performanceJob/{{id}}
200 PerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response
DELETE {{baseUrl}}/hub/{{id}}
204 Deleted

Figure 17. Performance Job progress tracking - Notifications

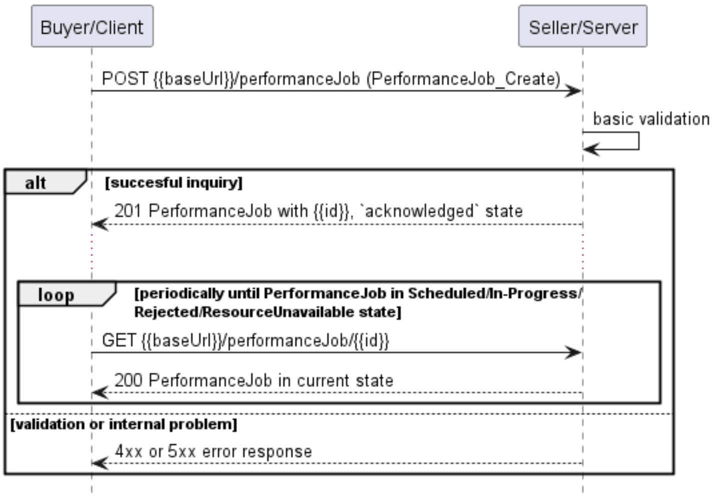

Buyer/Client
Seller/Server
POST {{baseUrl}}/performanceJob (PerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 PerformanceJob with {{id}}, `acknowledged` state
loop
[periodically until PerformanceJob in Scheduled/In-Progress/
Rejected/ResourceUnavailable state]
GET {{baseUrl}}/performanceJob/{{id}}
200 PerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response

Figure 18. Performance Job progress tracking - Polling

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 Monitoring Job Request

Figure 19 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.

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

Section `serviceSpecificConfiguration` of the Create Performance Job request allows for the introduction of service-specific properties of performance monitoring as the API payload, including performance objectives. 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.

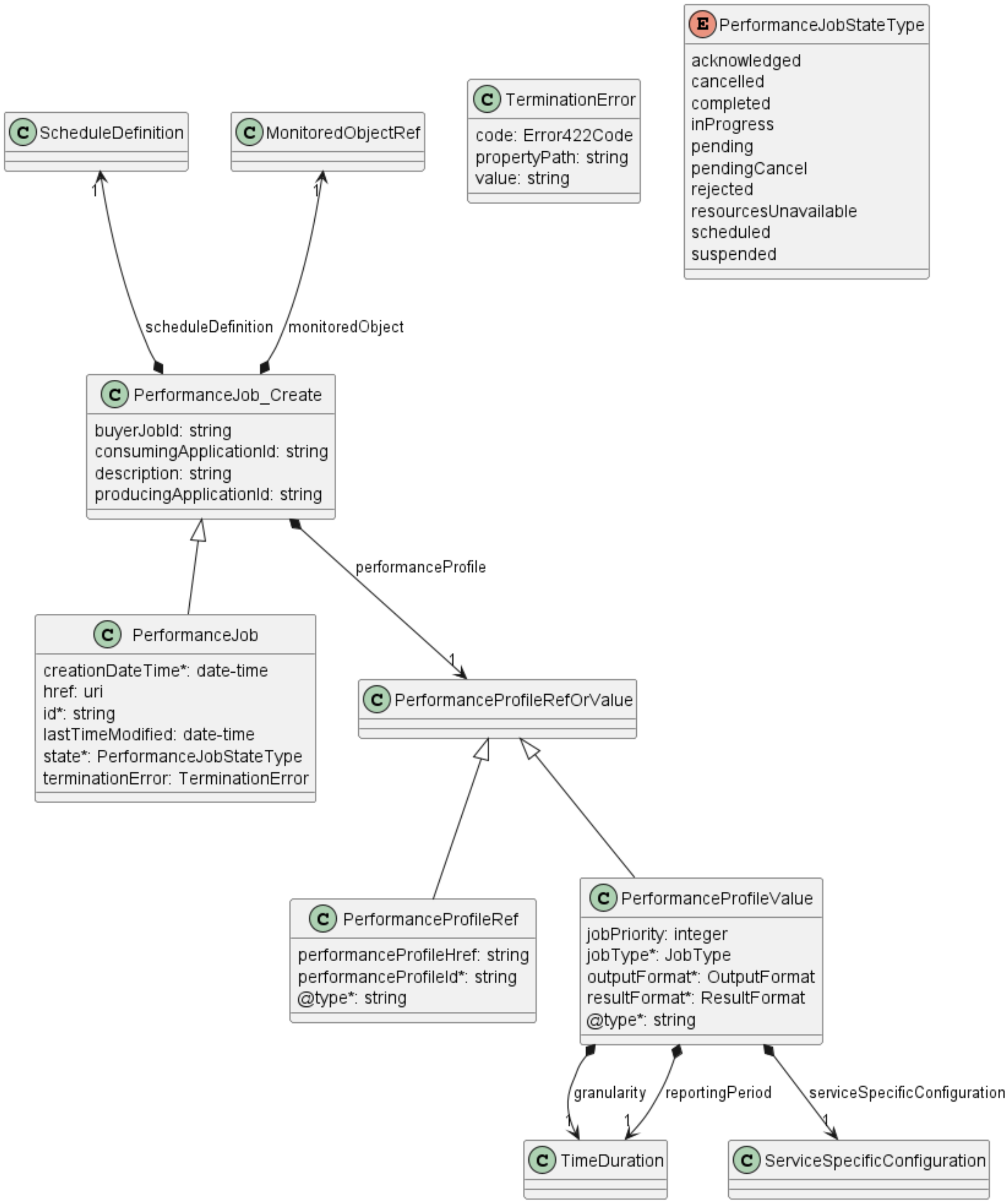


Figure 19. Performance Job Key Entities

To send a create Performance Job request the Buyer/Client uses the `createPerformanceJob` operation from the API: `POST /performanceJob`. For clarity, some of the 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",
  "monitoredObject": {
    "@type": "ServiceRef",
    "serviceId": "d6c998f0-c723-4754-81fe-ad3ef7c15683"
  },
  "performanceProfile": {
    "@type": "PerformanceProfileRef",
    "performanceProfileId": "8df0981a-0949-11ee-be56-0242ac120002"
  },
  "producingApplicationId": "SOF",
  "scheduleDefinition": {
    "scheduleDefinitionStartTime": "2024-12-01T08:02:01.370Z",
    "recurringSchedule": {
      "second": "0",
      "minute": "*/15",
      "hour": "*",
      "dayOfMonth": "*",
      "month": "*",
      "dayOfWeek": "*"
    }
  }
}
```

[R22] The Buyer's/Client's Create Performance Job request **MUST** provide the following attributes: [Mplify133.1 R55, R91]

- `performanceProfileId` (if used)
- `jobType` (only present when no `performanceProfileId` is referenced)
- `reportingPeriod` (only present when no `performanceProfileId` is referenced)
- `granularity` (only present when no `performanceProfileId` is referenced)
- `outputFormat` (only present when no `performanceProfileId` is referenced)
- `resultFormat` (only present when no `performanceProfileId` is referenced)
- `serviceSpecificConfiguration` (only present when no `performanceProfileId` is referenced)
- `monitoredObject`
- `scheduleDefinition`

[R23] If the Buyer/Client requests to provision a Performance Job for an ordered pair, they **MUST** provide reference to service endpoints by specifying the following: [Mplify133.1 R56]

- `serviceFromId`
- `serviceToId`

[R24] If the Buyer/Client requests to provision a Performance Job for a service, they **MUST** provide reference to a service by specifying `serviceId`. [Mplify133.1 R56]

[R25] If the Buyer/Client requests to provision a Performance Job for an entity other than a service, they **MUST** provide `entityId` [Mplify133.1 R57]

[O6] The Buyer's/Client's Create Performance Job request **MAY** contain the following attributes: [Mplify133.1 O11, O16]

- `buyerJobId`
- `consumingApplicationId`
- `description`
- `producingApplicationId`
- `jobPriority` (only present when no `performanceProfileId` is referenced)

[O7] A Performance Job **MAY** be scheduled as reoccurring. [Mplify133.1 O13]

6.6.3. Schedule Definition

Schedule (configured by `ScheduleDefinition` object) specifies when and how performance jobs are executed. It includes a start time (`scheduleDefinitionStartTime`), which if empty means the schedule starts immediately after job provisioning, and an end time (`scheduleDefinitionEndTime`), which if not provided means the job runs indefinitely. The `executionDuration` defines how long each job execution lasts, which may result in multiple reports per execution based on the reporting period. The `recurringSchedule` part defines the frequency of job's periodic execution using a Cron-like syntax, allowing jobs to run at specified seconds, minutes, hours, days, months, or weekdays. This scheduling supports special characters such as:

- "*" (asterisk) - represents "any value" for the given field.
  - For example, in the `minute` field, * means every minute.
  - ▪ ▪ ▪ ▪ ▪ — run a job every second, of every minute, of every hour, and so forth.
- "," (comma) - acts as a list separator to specify multiple discrete values.
  - For example, 0,30 in the `minute` field means "at minute 0 and 30".
- "-" (hyphen) - defines a range of values.
  - For example, 9-17 in the `hour` field includes all hours from 9 AM to 5 PM inclusive.
  - 1-5 in the `dayOfWeek` field means "Monday to Friday"
- "/" (slash) — specifies step values, meaning "every N units".
  - For example, */5 in the `second` field means "every 5 seconds".
  - 0 _/10 _ * * — every 10 minutes

This enables complex and precise job run times, for example:

- _/5 _ * * * * — runs a job every 5 seconds

  - */5 in the `second` field means: run every 5 seconds (0, 5, 10, 15, …)
  - ▪ in all other fields means every minute, hour, day, month, and weekday

- 0 _/10 _ * * * — runs a job every 10 minutes, at the start of the minute

  - 0 in seconds: run at the start of the minute
  - */10 in minutes: every 10 minutes (0, 10, 20, 30, 40, 50)
  - ▪ in other fields means every hour, day, month, and weekday

- 0 0 10 10 * * — runs a job at 10 AM on the 10th day of every month

  - 0 seconds, 0 minutes: at exactly the hour
  - 10 hour: 10 AM
  - 10 day-of-month: 10th day of the month
  - ▪ in month and weekday means every month and day of the week

- 0 0 22 * * 1-5 — runs a job at 10 PM every weekday (Monday to Friday)

  - 0 seconds, 0 minutes: at the start of the hour

  - 22 hour: 10 PM
  -
    - day-of-month and month: every day and month
  - 1-5 day-of-week: Monday to Friday

6.6.4. Create Performance Monitoring Job Response

Entities used for providing a response to Create Performance Job requests are presented in Figure 19. 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.

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",
  "monitoredObject": {
    "@type": "ServiceRef",
    "serviceId": "d6c998f0-c723-4754-81fe-ad3ef7c15683"
  },
  "performanceProfile": {
    "@type": "PerformanceProfileRef",
    "performanceProfileId": "8df0981a-0949-11ee-be56-0242ac120002"
  },
  "producingApplicationId": "SOF",
  "scheduleDefinition": {
    "scheduleDefinitionStartTime": "2024-12-01T08:02:01.370Z",
    "recurringSchedule": {
      "second": "0",
      "minute": "*/15",
      "hour": "*",
      "dayOfMonth": "*",
      "month": "*",
      "dayOfWeek": "*"
    }
  },
  "creationDateTimeTime": "2024-12-01T00:00:00.000Z", << added by SOF >>
   "href": "{{baseUrl}}/performanceMonitoring/v5/performanceJob/755e55e2-72b0-4e3b-af00-693e3beac691", << added by
SOF >>
  "id": "755e55e2-72b0-4e3b-af00-693e3beac691", << added by SOF >>
  "lastTimeModified": "2024-12-01T00:00:00.000Z", << 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.

[R26] The Seller/Server **MUST** assign an `id` to the Performance Job that is unique within the Seller's system. [Mplify133.1 R58, R59, R92, R93]

[R27] The Performance Job **MUST** apply the attributes included in the Buyer's/Client's Create Performance Job request. [Mplify133.1 R60, R94]

[R28] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by the Buyer/Client in the request.

[R29] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`

- creationDateTimeTime

[R30] The id **MUST** remain the same value for the life of the Performance Job.

6.6.5. Performance Monitoring Job State Machine

Figure 20 presents the Performance Job state machine:

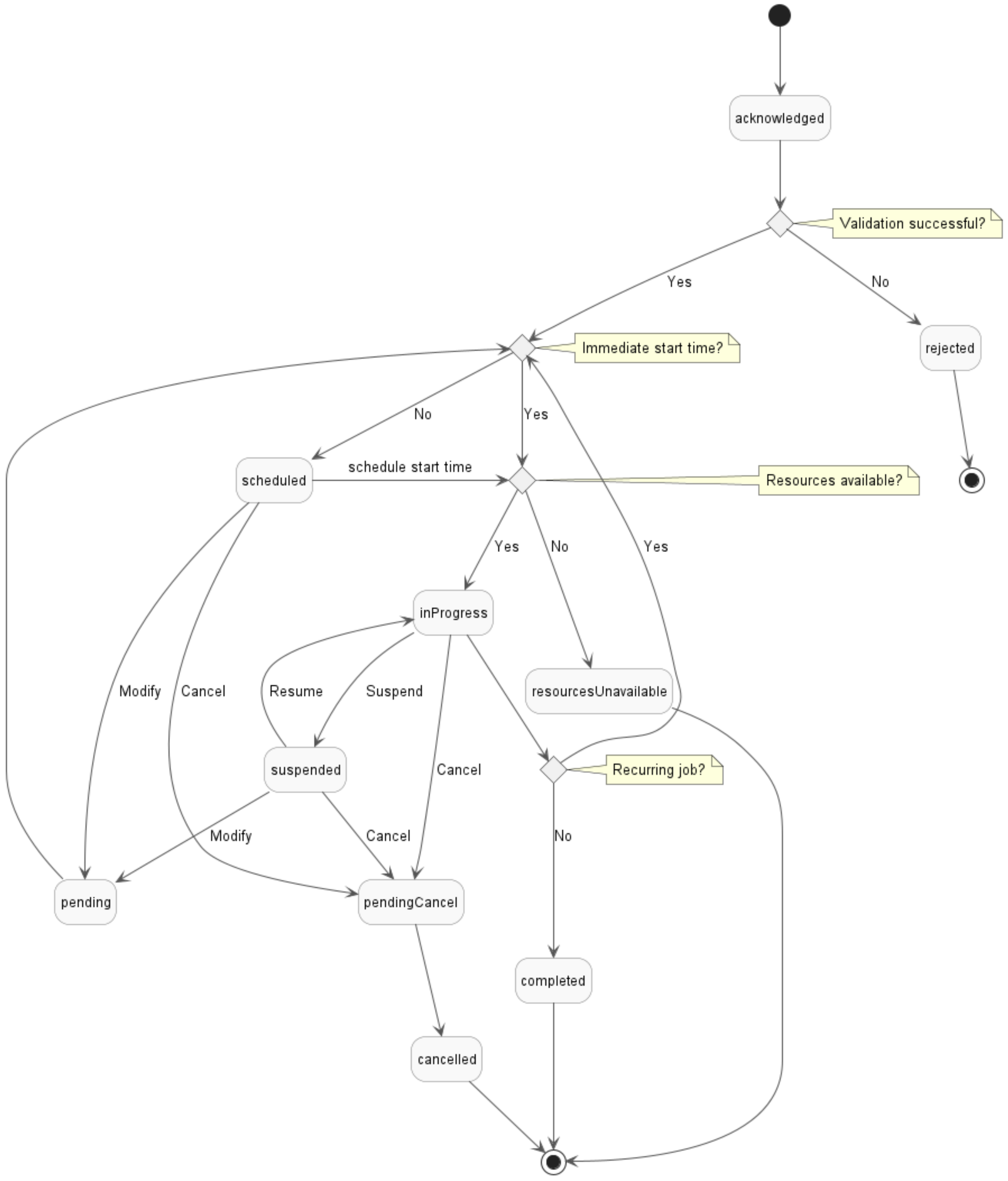


Figure 20. Performance Job State Machine

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.terminationError acts as a placeholder to provide a detailed description of what caused the problem. PerformanceJob moves to either the scheduled or inProgress state depending on the assigned schedule. PerformanceJob remains scheduled state until the scheduled start time is reached. PerformanceJob that is starting needs appropriate resources on Seller/Server side. If required resources cannot be assigned, PerformanceJob moves to resourcesUnavailable state. After measurements gathering is completed, the Seller/Server verifies if PerformanceJob is recurring. If yes, PerformanceJob moves to either scheduled or inProgress state depending on the schedule definition. Otherwise, it moves to a completed state. PerformanceJob can be cancelled when in scheduled, inProgress or suspended. When cancellation is successful, PerformanceJob moves to cancelled state. Cancellation includes an intermediary pendingCancel state. PerformanceJob can be modified only in the scheduled or suspended state. Modification includes an intermediary pending step.

Table 8 presents the mapping between the API status names and the Mplify 133.1 naming, together with the statuses' description.

state	Mplify 133.1 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's id 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 Seller/Server determines if the start time is immediate or scheduled. If immediate, the Performance Monitoring Job moves to the inProgress 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 was inProgress, suspended, or scheduled is cancelled.
completed	Completed	A non-recurring Performance Monitoring Job finished execution or recurring Performance Monitoring Job finished its schedule.
inProgress	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. If the PM job has an immediate start time it moves back to inProgress state, otherwise it moves to scheduled. 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	Mplify 133.1 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 `inProgress` status depending on the schedule definition.
`pendingCancel`	Pending Cancel	A Cancel Performance Monitoring Job request has been accepted by the Seller/Server. The Performance Monitoring Job remains `pendingCancel` while resources used by the Job are being released. Once updates are complete, the Job moves to the `cancelled` status.
`rejected`	Rejected	A Create Performance Monitoring Job request fails validation and is rejected with error indications by the Seller/Server.
`resourcesUnavailable`	Resource Unavailable	A Performance Monitoring Job cannot be allocated with necessary resources when moving to execution (`inProgress` 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 `inProgress`. If the Cancel Performance Monitoring Job request is accepted, the Job moves to the `cancelled` state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the `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 `inProgress` state. If the Cancel Performance Monitoring Job request is accepted, the Job moves to the `cancelled` state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the `pending` state.

Table 8. Performance Job State Machine states

[R31] The Seller/Server **MUST** support all Performance Job statuses and their associated transitions as described in Figure 20 and Table 8.

6.6.6. Relationship to Performance Monitoring Profile

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

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

- provide required attributes that are typically defined by `PerformanceProfile` model directly in the create Performance Job request.

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

`PerformanceJob_Create` and related entities that allow for referencing to Performance Profile or specifying corresponding attributes are presented in Figure 19.

6.7. Use Case 7: Retrieve List of Performance Monitoring Job

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

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/performanceJob?state=suspended&limit=10&offset=0
```

[O8] The Buyer/Client Retrieve List of Performance Jobs request **MAY** contain zero or more of the following attributes as filter attributes: [Mplify133.1 O18]

- `buyerJobId`
- `serviceId`
- `serviceFromId`
- `serviceToId`
- `entityId`
- `performanceProfileId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`
- `jobType`
- `jobPriority`
- `consumingApplicationId`
- `producingApplicationId`

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.

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. Section 6.2 provides details about the implementation of pagination mechanism.

[R32] The Seller/Server's response to the Buyer's/Client's Retrieve List of Performance Jobs **MUST** include the following attributes as applicable: [Mplify133.1 R111]

- `creationDateTime`
- `id`
- `monitoredObject`
- `performanceProfile`

- scheduleDefinition
- state

[O9] The Seller/Server's response to the Buyer's/Client's Retrieve List of Performance Jobs **MAY** include the following attributes as applicable: [Mplify133.1 O19]

- buyerJobId
- consumingApplicationId
- producingApplicationId
- description

[R33] 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. [Mplify133.1 R78, R112]

Get List returns full representation of matched Performance Jobs which includes all attributes as shown in the Create Performance Monitoring Job Response in section 6.6.3.

6.8. Use Case 8: Retrieve Performance Monitoring Job by 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 the 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 as presented in section 6.6.3.

[R34] In case `id` does not match a `PerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R35] The Seller/Server's response **MUST** contain all the PM Job attributes. [Mplify133.1 R80, R114]

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 the Performance Monitoring Job may involve a 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 21.

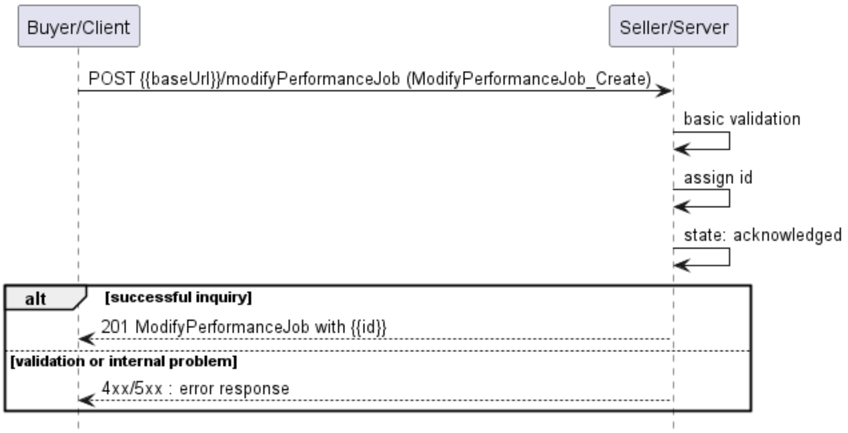

Buyer/Client
Seller/Server
POST {{baseUrl}}/modifyPerformanceJob (ModifyPerformanceJob_Create)
basic validation
assign id
state: acknowledged
alt
[successful inquiry]
201 ModifyPerformanceJob with {{id}}
[validation or internal problem]
4xx/5xx : error response

Figure 21. 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 the 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 the 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.

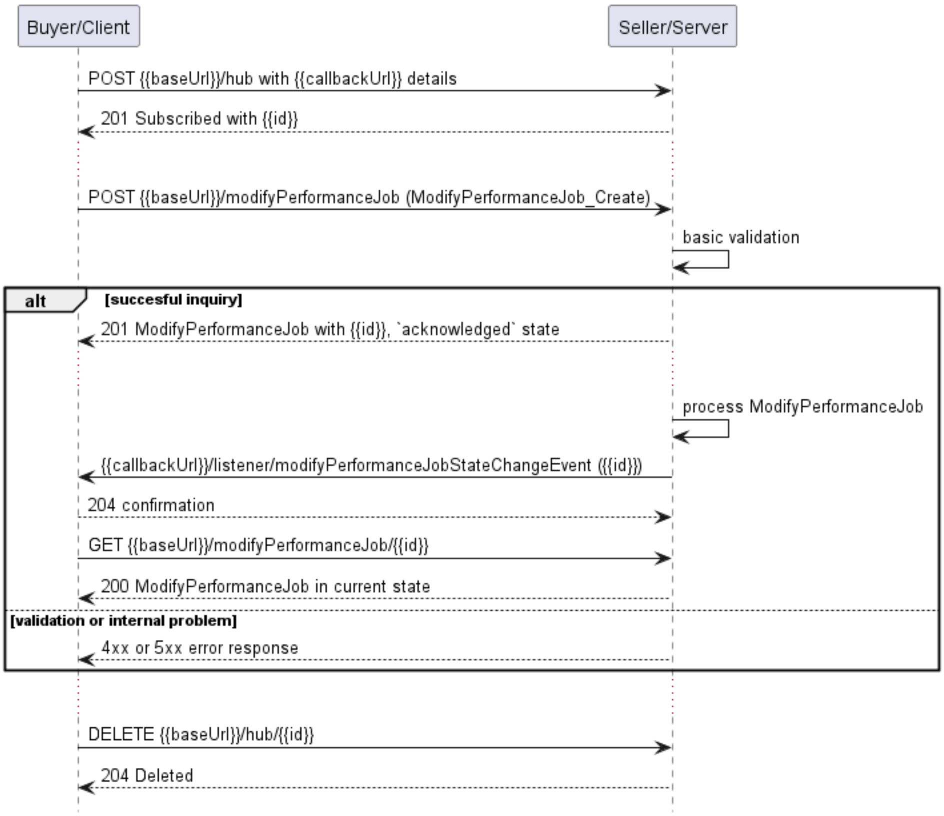

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}} details
201 Subscribed with {{id}}
POST {{baseUrl}}/modifyPerformanceJob (ModifyPerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 ModifyPerformanceJob with {{id}}, `acknowledged` state
process ModifyPerformanceJob
{{callbackUrl}}/listener/modifyPerformanceJobStateChangeEvent ({{id}})
204 confirmation
GET {{baseUrl}}/modifyPerformanceJob/{{id}}
200 ModifyPerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response
DELETE {{baseUrl}}/hub/{{id}}
204 Deleted

Figure 22. Modify Performance Job progress tracking - Notifications

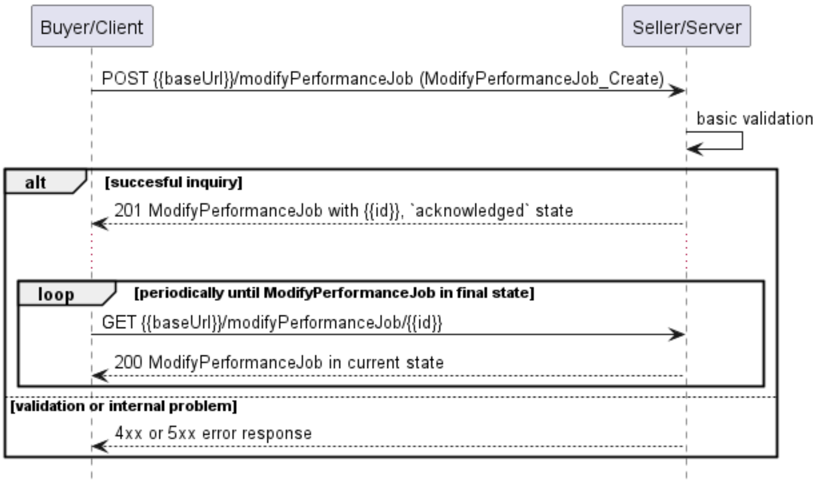

Buyer/Client
Seller/Server
POST {{baseUrl}}/modifyPerformanceJob (ModifyPerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 ModifyPerformanceJob with {{id}}, `acknowledged` state
loop
[periodically until ModifyPerformanceJob in final state]
GET {{baseUrl}}/modifyPerformanceJob/{{id}}
200 ModifyPerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response

Figure 23. Modify Performance Job progress tracking - Polling

The Modify Performance Job process is altering the state of the PM job itself. It is important to note that notifications resulting from changes in the state of the Performance Job are not represented in Figure 22.

6.9.2. Modify Performance Monitoring Job Request

Figure 24 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.

It is important to note that updating the reference to the Performance Profile must not be possible. To change this assignment, the existing Performance Job must be cancelled and replaced by a new job that relates to the relevant profile. The same restriction applies to changing from a reference to a Performance Profile (`PerformanceProfileRef`) to directly assigning the attribute values defined by the Performance Profile class (`PerformanceProfileValue`). In summary, modification of Performance Job allows for changing attributes defined directly by the `PerformanceJob` type and attributes from Performance Profile model that are defined by value in Performance Job. The latter attributes are contained in the `performanceProfile` attribute.

The `performanceJob` attribute of `ModifyPerformanceJob_Create` is used to specify which Performance Job object is a subject of the modification process.

Only attributes of the Performance Job that are supposed to be modified, should be included in the Modify Performance Job Request.

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

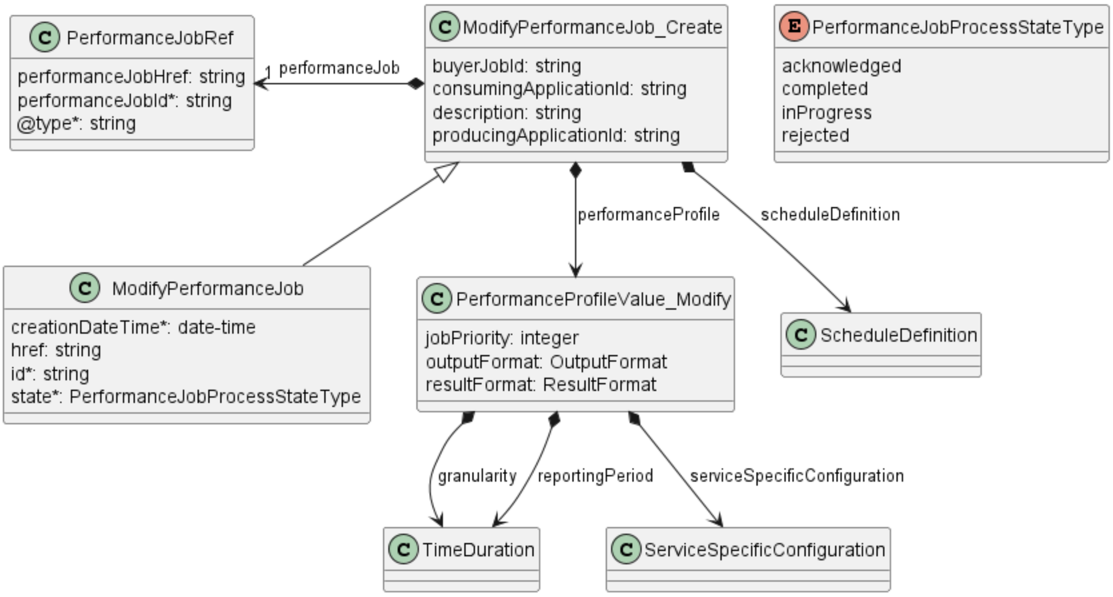

PerformanceJobRef
performanceJobHref: string
performanceJobId*: string
@type*: string
1
performanceJob
ModifyPerformanceJob_Create
buyerJobId: string
consumingApplicationId: string
description: string
producingApplicationId: string
PerformanceJobProcessStateType
acknowledged
completed
inProgress
rejected
performanceProfile
scheduleDefinition
ModifyPerformanceJob
creationDateTime*: date-time
href: string
id*: string
state*: PerformanceJobProcessStateType
PerformanceProfileValue_Modify
jobPriority: integer
outputFormat: OutputFormat
resultFormat: ResultFormat
ScheduleDefinition
granularity
reportingPeriod
serviceSpecificConfiguration
TimeDuration
ServiceSpecificConfiguration

Figure 24. 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`
- change `description` of the Performance Job

```
{
  "buyerJobId": "TestJob54321",
  "description": "Performance Job after modification",
  "performanceJob": {
    "performanceJobId": "755e55e2-72b0-4e3b-af00-693e3beac691"
  }
}
```

[R36] The Buyer's/Client's Modify PM Job request **MUST** include the Performance Job's `id`. [Mplify133.1 R62, R96]

[O10] The Buyer/Client **MAY** include one or more of the following attributes of `ModifyPerformanceJob_Create` in the request: [Mplify133.1 O14, O17]

- `buyerJobId`
- `consumingApplicationId`
- `description`
- `granularity`
- `jobPriority`
- `outputFormat`
- `producingApplicationId`
- `reportingPeriod`
- `resultFormat`
- `scheduleDefinition`
- `serviceSpecificConfiguration`

6.9.3. Modify Performance Monitoring Job Response

Entities used for providing a response to Modify Performance Job requests are presented in Figure 24. 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.

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",
  "performanceJob": {
    "performanceJobId": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "creationDateTime": "2023-06-19T12:58:17.088Z", << added by SOF >>
    "href":  "{{baseUrl}}/performanceMonitoring/v5/modifyPerformanceJob/9c51d971-185d-403e-952f-2110f33a9628",  <<
added by SOF >>
  "id": "9c51d971-185d-403e-952f-2110f33a9628", << added by SOF >>
  "state": "acknowledged" << added by SOF >>
}
```

[R37] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by the Buyer/Client in the request.

[R38] The Seller/Server **MUST** specify the following attributes in a response:

- `id`
- `state`
- `creationDateTime`

[R39] 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 the state to rejected. This includes situation when:

- id does not match a PerformanceJob that is to be updated in Seller/Server's system
- requested attributes cannot be modified
- Performance Job is in a state that does not allow for modification (any state other than suspended).

6.9.4. Modify Performance Monitoring Job State Machine

Figure 25 presents the Modify Performance Monitoring Job state machine:

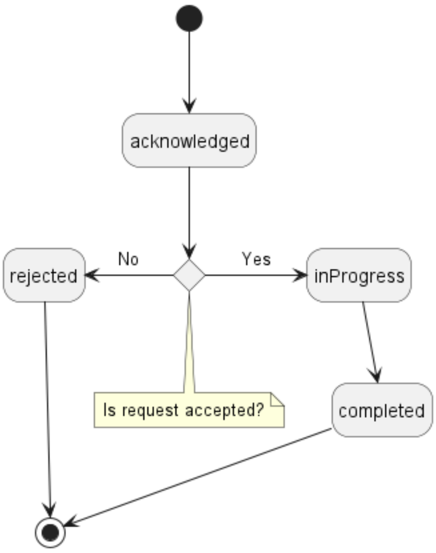


Figure 25. Modify Performance Job State Machine

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 process moves to a rejected state if some issues are found. If validation is successful, ModifyPerformanceJob moves to the inProgress state. At this point, the related PerformanceJob moves to a pending state, and the Seller/Server starts all necessary arrangements to provision modification request. PerformanceJob remains in the pending state until the Modify Performance Job process is finished and moved to the completed state. This causes the PerformanceJob state to change to scheduled or inProgress depending on the ScheduleDefinition.

Table 9 presents the mapping between the API status names and the Mplify 133.1 naming, together with the statuses' description. The list of statuses is the same for all processes related to Performance Job (cancel/modify).

state	Mplify 133.1 name	Description

state	Mplify 133.1 name	Description
acknowledged	Acknowledged	The Cancel/Modify 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 acknowledged until all validations as applicable are completed. If the attributes are validated, the request moves to the inProgress state. If not all attributes are validated, the request moves to the rejected state.
completed	Completed	The Cancel/Modify Performance Monitoring Job request has been completed by the Seller/Server.
inProgress	In-Progress	The Cancel/Modify Performance Monitoring Job request has been validated and accepted by the Seller/Server and is in-progress.
rejected	Rejected	The Cancel/Modify Performance Monitoring Job request has failed validation and has been declined by the Seller/Server.

Table 9. Performance Job Process State Machine states

[R40] The Seller/Server **MUST** support all Modify Performance Job statuses and their associated transitions as described in Figure 25 and Table 9.

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.

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/modifyPerformanceJob?state=acknowledged&limit=10&offset=0
```

[O11] The Buyer/Client Retrieve List of Modify Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. Section 6.2 provides details about the implementation of pagination mechanism.

[R41] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Get List returns full representation of matched Modify Performance Job objects which includes all attributes as shown in the Modify Performance Monitoring Job Response in section 6.9.3.

6.11. Use Case 11: Retrieve Modify Performance Monitoring Job 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 as presented in section 6.9.3.

[R42] In case id does match a ModifyPerformanceJob in Seller/Server's system, an error response Error404 **MUST** be returned.

[R43] The Seller/Server **MUST** include following attributes in the ModifyPerformanceJob object in the response:

- creationDateTime
- id
- performanceJob
- state

[R44] 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 the Performance Monitoring Job may involve a prolonged duration. Consequently, this operation is implemented through a separate lifecycle process.

[R45] Performance Monitoring Job must be in state inProgress, suspended, or scheduled in order for the Buyer/Client to be able to cancel it. [Mplify133.1 R65]

6.12.1. Interaction flow

The flow of this use case is shown in Figure 26.

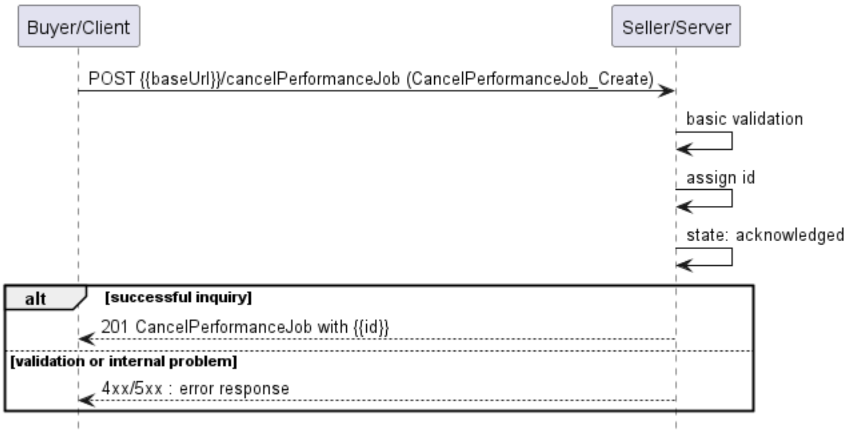

Buyer/Client
Seller/Server
POST {{baseUrl}}/cancelPerformanceJob (CancelPerformanceJob_Create)
basic validation
assign id
state: acknowledged
alt
[successful inquiry]
201 CancelPerformanceJob with {{id}}
[validation or internal problem]
4xx/5xx : error response

Figure 26. 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 the 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 cancel the 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.

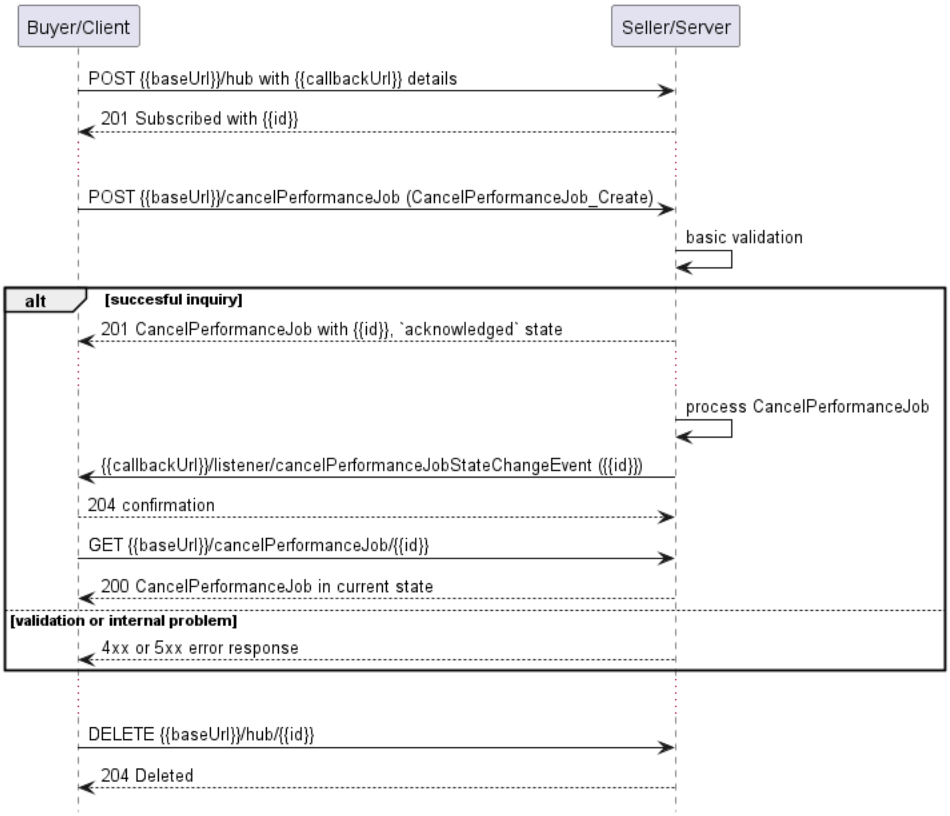

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}} details
201 Subscribed with {{id}}
POST {{baseUrl}}/cancelPerformanceJob (CancelPerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 CancelPerformanceJob with {{id}}, `acknowledged` state
process CancelPerformanceJob
{{callbackUrl}}/listener/cancelPerformanceJobStateChangeEvent ({{id}})
204 confirmation
GET {{baseUrl}}/cancelPerformanceJob/{{id}}
200 CancelPerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response
DELETE {{baseUrl}}/hub/{{id}}
204 Deleted

Figure 27. Cancel Performance Job progress tracking - Notifications

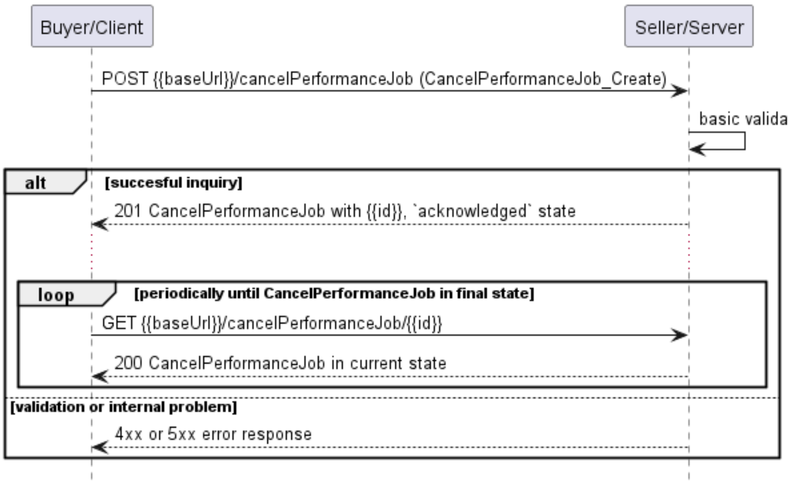

Buyer/Client
Seller/Server
POST {{baseUrl}}/cancelPerformanceJob (CancelPerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 CancelPerformanceJob with {{id}}, `acknowledged` state
loop
[periodically until CancelPerformanceJob in final state]
GET {{baseUrl}}/cancelPerformanceJob/{{id}}
200 CancelPerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response

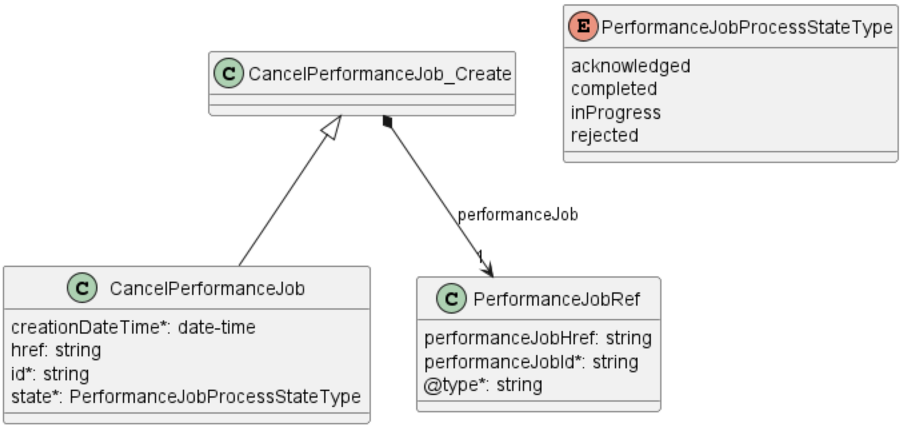
Figure 28. Cancel Performance Job progress tracking - Polling

6.12.2. Cancel Performance Monitoring Job Request

Figure 29 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.

The `performanceJob` attribute of `CancelPerformanceJob_Create` is used to specify which Performance Job object is a subject of the cancellation process.

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



PerformanceJobProcessStateType
acknowledged
completed
inProgress
rejected
CancelPerformanceJob_Create
performanceJob
1
CancelPerformanceJob
creationDateTime*: date-time
href: string
id*: string
state*: PerformanceJobProcessStateType
PerformanceJobRef
performanceJobHref: string
performanceJobId*: string
@type*: string

Figure 29. 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.

```
{
  "performanceJob": {
    "performanceJobId": "755e55e2-72b0-4e3b-af00-693e3beac691"
  }
}
```

[R46] The Buyer's/Client's Cancel Performance Monitoring Job request **MUST** include the Performance Job's `id`. [Mplify133.1 R64, R98]

If action arrives when Performance Job is running, it is recommended to run until the end and only afterward action should be applied. [Mplify133.1 O25]

6.12.3. Cancel Performance Monitoring Job Response

Entities used for providing a response to Cancel Performance Job requests are presented in Figure 29. 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.

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

```
{
  "performanceJob": {
    "performanceJobId": "755e55e2-72b0-4e3b-af00-693e3beac691"
  },
  "creationDateTime": "2023-06-19T12:58:17.088Z", << added by SOF >>
    "href":  "{{baseUrl}}/performanceMonitoring/v5/cancelPerformanceJob/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.

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

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

- `id`
- `state`
- `creationDateTime`

[R49] 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 the state to `rejected`. This includes situation when:

- `id` does not match a `PerformanceJob` that is to be cancelled in Seller/Server's system
- Performance Job is in a state that does not allow for cancellation (any state other than `inProgress`, `suspended`, or `scheduled`).

6.12.4. Cancel Performance Monitoring Job State Machine

Figure 30 presents the Cancel Performance Monitoring Job state machine:

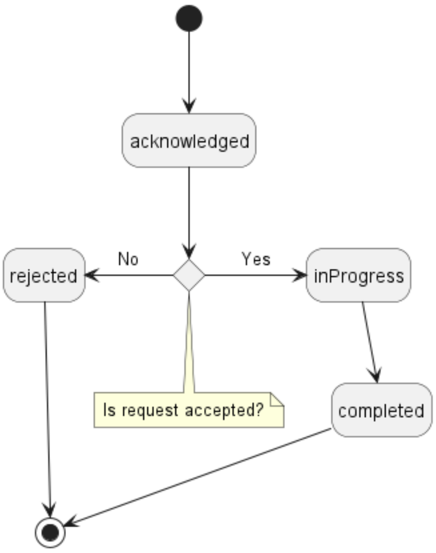


Figure 30. Cancel Performance Job State Machine

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 `rejected` state if some issues are found. If validation is successful, `CancelPerformanceJob` moves to the `inProgress` state. At this point, the related `PerformanceJob` moves to a `pendingCancel` state, and the Seller/Server starts all necessary arrangements to deprovision resources. `PerformanceJob` remains in the `pendingCancel` state until the Cancel Performance Job process is finished and moved to the `completed` state. This causes the `PerformanceJob` state to change to `cancelled`.

Description and mapping of the Cancel Performance Job States are the same as in table 9.

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.

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/cancelPerformanceJob?state=acknowledged&limit=10&offset=0
```

[O12] The Buyer/Client Retrieve List of Cancel Performance Jobs request **MAY** contain none or more of the following attributes:

- `performanceJobId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. Section 6.2 provides details about the implementation of pagination mechanism.

[R50] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list.

Get List returns full representation of matched Cancel Performance Job objects which includes all attributes as shown in the Cancel Performance Monitoring Job Response in section 6.12.3.

6.14. Use Case 14: Retrieve Cancel Performance Monitoring Job 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 the 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, as presented in section 6.12.3.

[R51] In case `id` does not match a `CancelPerformanceJob` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R52] The Seller/Server **MUST** include following attributes in the `CancelPerformanceJob` object in the response:

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

[R53] 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

The Buyer/Client may request to suspend a Performance Monitoring Job by using `POST /performanceJob/{id}/suspend` endpoint. This operation only requires providing the `id` of the Performance Job in the path and has an empty `204` confirmation response.

The sequence diagram in Figure 31 presents this use case in detail.

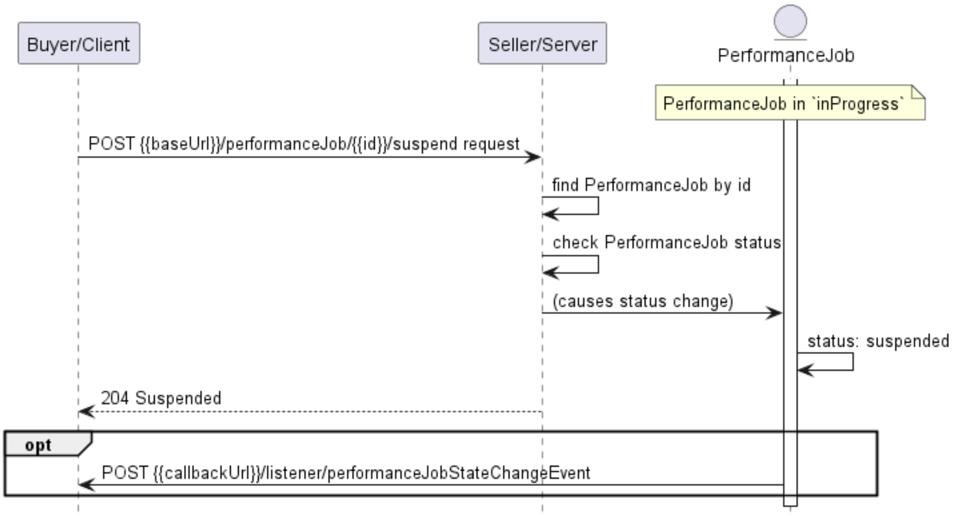

Buyer/Client
Seller/Server
PerformanceJob
PerformanceJob in `inProgress`
POST {{baseUrl}}/performanceJob/{{id}}/suspend request
find PerformanceJob by id
check PerformanceJob status
(causes status change)
status: suspended
204 Suspended
opt
POST {{callbackUrl}}/listener/performanceJobStateChangeEvent

Figure 31. Use Case 15 - Suspend Performance Monitoring Job Flow

The Buyer/Client sends a request specifying `id` of the Performance Monitoring Job to be suspended. The Seller/Server performs request validation, then searches for Performance Monitoring Job. If found, the status is verified (`inProgress`). If everything is verified correctly, the Seller/Server moves the Performance Monitoring Job to the `suspended` status, and sends a successful response to a suspension request followed by `performanceJobStateChangeEvent`.

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

[R54] In case `id` does not match a `PerformanceJob` that is to be suspended, an error response `Error404` **MUST** be returned.

[R55] The Performance Job **MUST** be in the `inProgress` state to be suspended. [Mplify133.1 R67, R100]

[R56] In case of a successful validation of the suspend request, the Seller **MUST** move the Performance Job to `suspended` status. [Mplify133.1 R69, R102]

[R57] If the Seller/Server declines the Buyer/Client's suspend Performance Job request, the Performance Job **MUST NOT** be suspended. [Mplify133.1 R70, R103]

[R58] If the Seller/Server declines the Buyer/Client's request to suspend a Performance Job, they **MUST** provide an Error with a meaningful explanation in `reason` field. [Mplify133.1 R71, R104]

[O13] In case the 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.

6.16. Use Case 16: Resume Performance Monitoring Job

The Buyer/Client may request to resume a Performance Monitoring Job by using `POST /performanceJob/{id}/resume` endpoint. This operation only requires providing the `id` of the Performance Job in the path and has an empty `204` confirmation response.

The sequence diagram in Figure 32 presents this use case in detail.

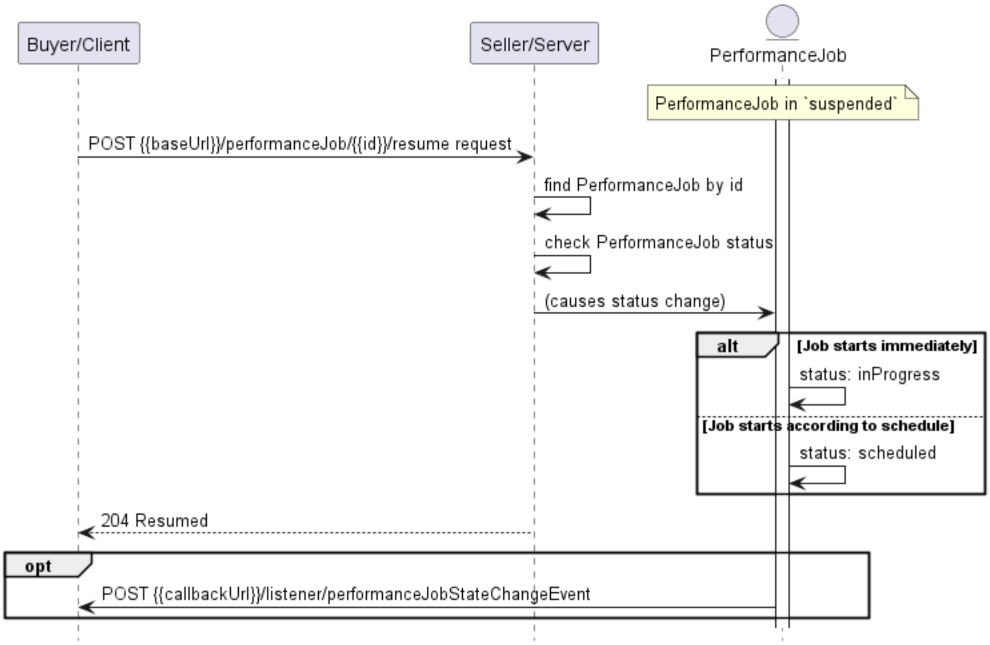

Buyer/Client
Seller/Server
PerformanceJob
PerformanceJob in `suspended`
POST {{baseUrl}}/performanceJob/{{id}}/resume request
find PerformanceJob by id
check PerformanceJob status
(causes status change)
alt
[Job starts immediately]
status: inProgress
[Job starts according to schedule]
status: scheduled
204 Resumed
opt
POST {{callbackUrl}}/listener/performanceJobStateChangeEvent

Figure 32. Use Case 16 - Resume Performance Monitoring Job Flow

The Buyer/Client sends a request specifying `id` of the Performance Monitoring Job to be resumed. The Seller/Server performs request validation, then searches for Performance Monitoring Job. If found, the status is verified (`suspended`). If everything is verified correctly, the Seller/Server moves the Performance Monitoring Job to `scheduled` or `inProgress` status depending on the schedule, and sends a successful response to a resumption request followed by `performanceJobStateChangeEvent`.

[R59] In case `id` does not match a `PerformanceJob` that is to be resumed, an error response `Error404` **MUST** be returned.

[R60] The Performance Job **MUST** be in the `suspended` state in order to be resumed. [Mplify133.1 R73, R106]

[R61] In case of a successful validation of the resume request, the Seller **MUST** move the Performance Job to `inProgress` or `scheduled` status depending on the schedule. [Mplify133.1 R75, R108]

[R62] If the Seller/Server declines the Buyer/Client's Resume Performance Job request, the Performance Job **MUST NOT** be resumed. [Mplify133.1 R76, R109]

[R63] If the Seller/Server declines the Buyer/Client's Resume Performance Job request, they **MUST** provide an Error with a meaningful explanation in `reason` field [Mplify133.1 R77, R110]

6.17. Use Case 17: 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`.

6.17.1. Performance Monitoring Job Complex Query Request

Figure 33 depicts the key components of the data model utilized in the Performance Job Complex Query request (`POST /performanceJobComplexQuery`) and its corresponding response. The request message model, `PerformanceJobComplexQuery_Create`, includes only attributes that can be specified by the Buyer/Client, representing filtering options. In response, the Seller/Server provides a list of PerformanceJob entities that contain the matched PM Job objects.

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

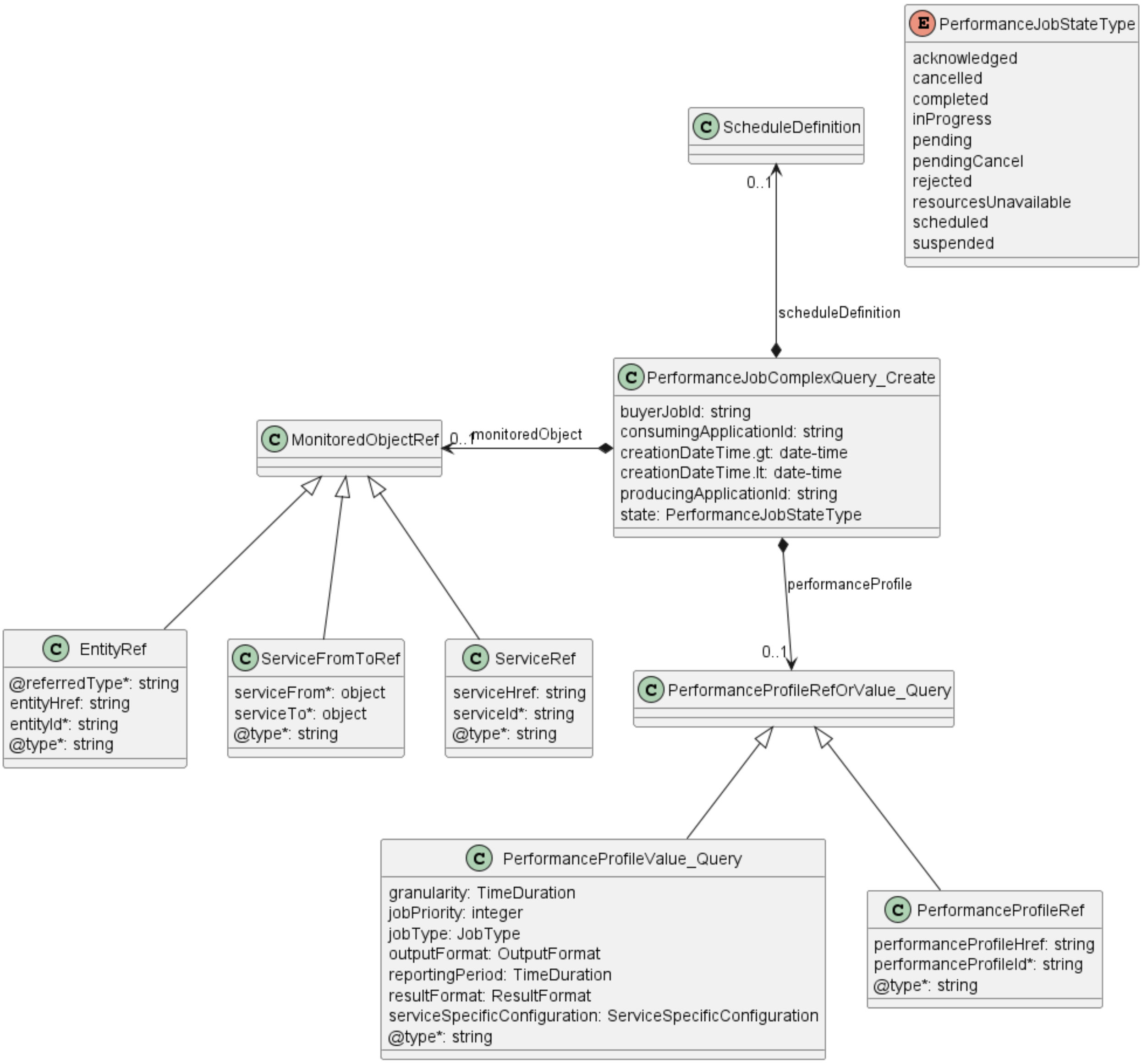


Figure 33. Performance Job Complex Query Key Entities

To send a request the Buyer/Client uses performanceJobComplexQuery operation from the API and provides PerformanceJobComplexQuery_Create as payload. The snippet below presents an example of a Performance Job Complex Query request. It lists 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 the recurring frequency set to 1 hour
- are in a scheduled state

Performance Job Complex Query Request

```
{
  "consumingApplicationId": "CUS",
  "performanceProfile": {
    "@type": "PerformanceProfileRef",
    "performanceProfileId": "8df0981a-0949-11ee-be56-0242ac120002"
  },
  "scheduleDefinition": {
    "recurringSchedule": {
      "second": "0",
      "minute": "0",
      "hour": "*/1",
      "dayOfMonth": "*",
      "month": "*",
      "dayOfWeek": "*"
    }
  },
  "state": "scheduled"
}
```

[O14] The Buyer/Client Performance Job Complex Query request **MAY** contain zero or more of the following attributes as filter attributes: [Mplify133.1 O18]

- `buyerJobId`
- `serviceId`
- `serviceFromId`
- `serviceToId`
- `entityId`
- `performanceProfileId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`
- `jobType`
- `jobPriority`
- `consumingApplicationId`
- `producingApplicationId`
- `scheduleDefinition`
- `granularity`
- `outputFormat`
- `reportingPeriod`
- `resultFormat`
- `serviceSpecificConfiguration`

6.17.2. Performance Monitoring Job Complex Query Response

The Seller/Server responds with a list of `PerformanceJob` objects, which are full representation of matched Performance Jobs.

The following snippet presents the Seller/Server response.

`Performance Job Complex Query` Response

```
  [
    {
      "buyerJobId": "TestJob12345",
      "consumingApplicationId": "CUS",
      "creationDateTime": "2024-12-01T00:00:00.000Z",
      "description": "Exemplary Create Performance Job request",
      "href": "{{baseUrl}}/performanceMonitoring/v5/performanceJobComplexQuery/755e55e2-72b0-4e3b-af00-693e3beac691",
      "id": "755e55e2-72b0-4e3b-af00-693e3beac691",
      "lastTimeModified": "2024-12-01T00:00:00.000Z",
      "monitoredObject": {
        "@type": "ServiceRef",
        "serviceId": "d6c998f0-c723-4754-81fe-ad3ef7c15683"
      },
      "performanceProfile": {
        "@type": "PerformanceProfileRef",
        "performanceProfileId": "8df0981a-0949-11ee-be56-0242ac120002"
      },
      "producingApplicationId": "SOF",
      "scheduleDefinition": {
        "scheduleDefinitionStartTime": "2024-12-01T08:02:01.370Z",
        "recurringSchedule": {
          "second": "0",
          "minute": "*/15",
          "hour": "*",
          "dayOfMonth": "*",
          "month": "*",
          "dayOfWeek": "*"
        }
      },
      "state": "acknowledged"
    }
  ]
```

[R64] The Seller/Server's response to the Buyer’s/Client’s Performance Monitoring Job Complex Query **MUST** include the following attributes as applicable: [Mplify133.1 R111]

- `creationDateTime`
- `id`
- `lastTimeModified`
- `monitoredObject`
- `performanceProfileId` or `granularity`, `jobPriority`, `jobType`, `outputFormat`, `reportingPeriod`, `resultFormat`, `serviceSpecificConfiguration`
- `scheduleDefinition`
- `state`
- **[O15]** The Seller/Server's response to the Buyer’s/Client’s Performance Monitoring Job Complex Query **MAY** include the following attributes as applicable: [Mplify133.1 O19]
- `buyerJobId`
- `consumingApplicationId`
- `producingApplicationId`
- `description`

6.18. Use Case 18: Create a Performance Monitoring Report

The execution of all types of Performance Monitoring Jobs results in the generation of Performance Monitoring Reports, which deliver comprehensive performance or statistics collections to the Buyer/Client. In certain scenarios, performance data can be collected by directly creating a Performance Report. This functionality can be useful for retrieving PM data from a dataset that is already available on the Server (e.g., historical statistics).

6.18.1. Interaction flow

The flow of this use case is illustrated in Figure 34. A Performance Report can be create either by SOF as an outcome of processing a Performance Job or by the Buyer/Client executing a Create Performance Report request. Both of these options are depicted in the figure. If a Performance Report is created internally by the SOF as a result of executing a Performance Job, the Buyer/Client can learn about the new report either by subscribing to events or by periodically polling Performance Reports using the `id` of the related Performance Job. The Buyer/Client can retrieve the report as described in section 6.20.

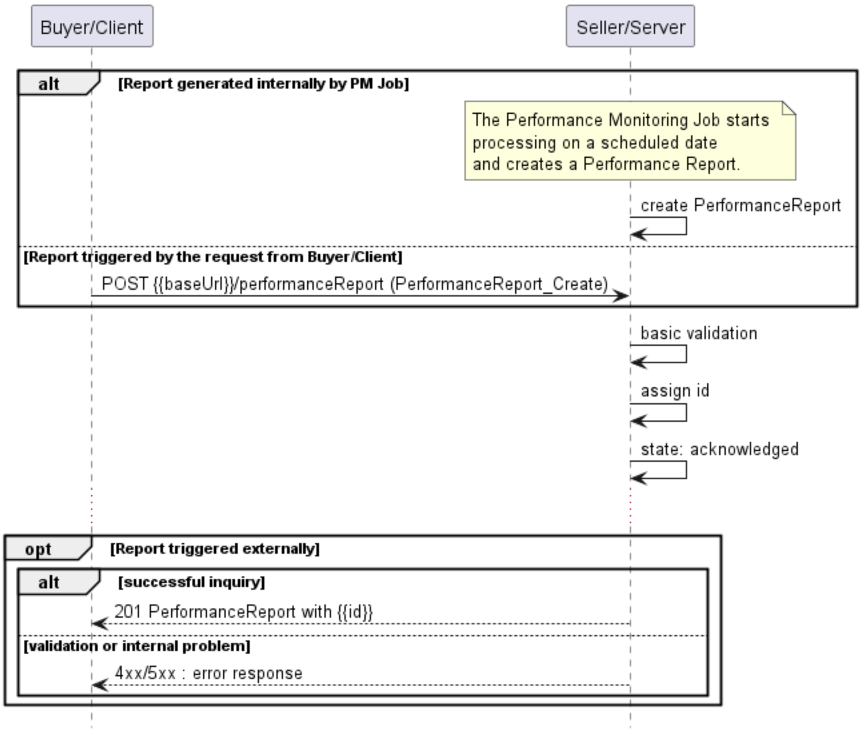

Buyer/Client
Seller/Server
alt
[Report generated internally by PM Job]
The Performance Monitoring Job starts processing on a scheduled date and creates a Performance Report.
create PerformanceReport
[Report triggered by the request from Buyer/Client]
POST {{baseUrl}}/performanceReport (PerformanceReport_Create)
basic validation
assign id
state: acknowledged
opt
[Report triggered externally]
alt
[successful inquiry]
201 PerformanceReport with {{id}}
[validation or internal problem]
4xx/5xx : error response

Figure 34. Use Case 18 - Create Performance Monitoring Report flow

In order to create a report, 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.

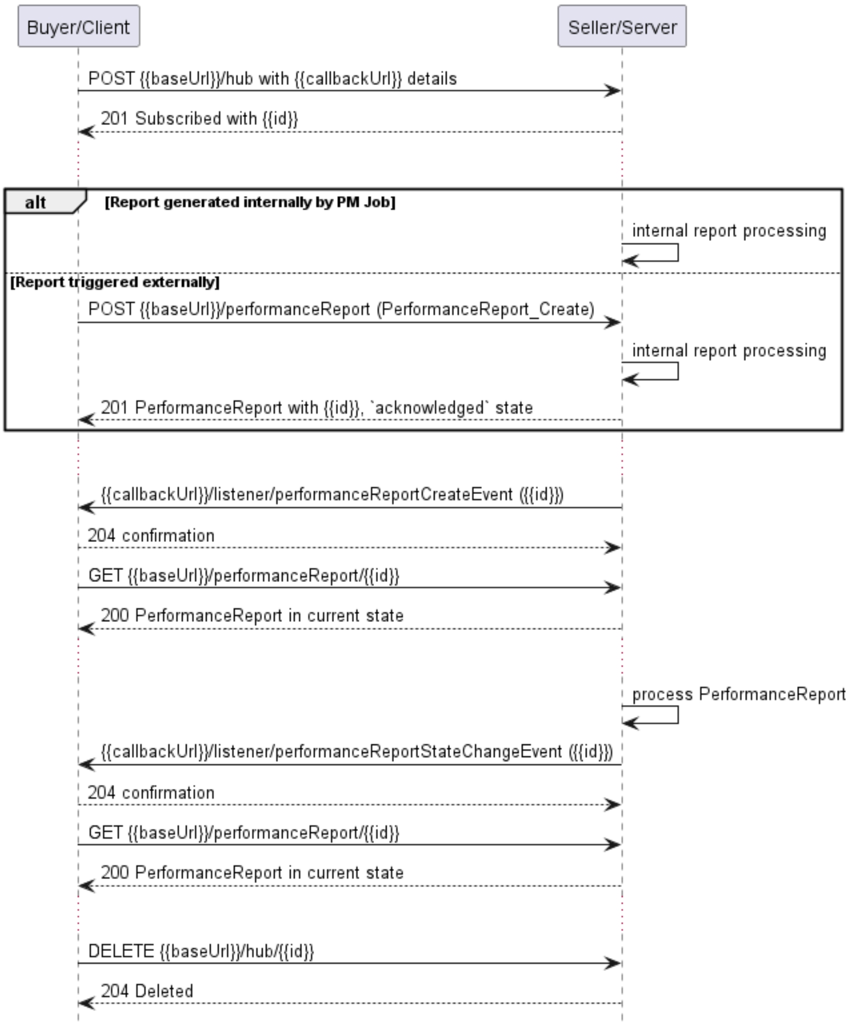

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}} details
201 Subscribed with {{id}}
alt
[Report generated internally by PM Job]
internal report processing
[Report triggered externally]
POST {{baseUrl}}/performanceReport (PerformanceReport_Create)
internal report processing
201 PerformanceReport with {{id}}, `acknowledged` state
{{callbackUrl}}/listener/performanceReportCreateEvent ({{id}})
204 confirmation
GET {{baseUrl}}/performanceReport/{{id}}
200 PerformanceReport in current state
process PerformanceReport
{{callbackUrl}}/listener/performanceReportStateChangeEvent ({{id}})
204 confirmation
GET {{baseUrl}}/performanceReport/{{id}}
200 PerformanceReport in current state
DELETE {{baseUrl}}/hub/{{id}}
204 Deleted

Figure 35. Performance Job progress tracking - Notifications

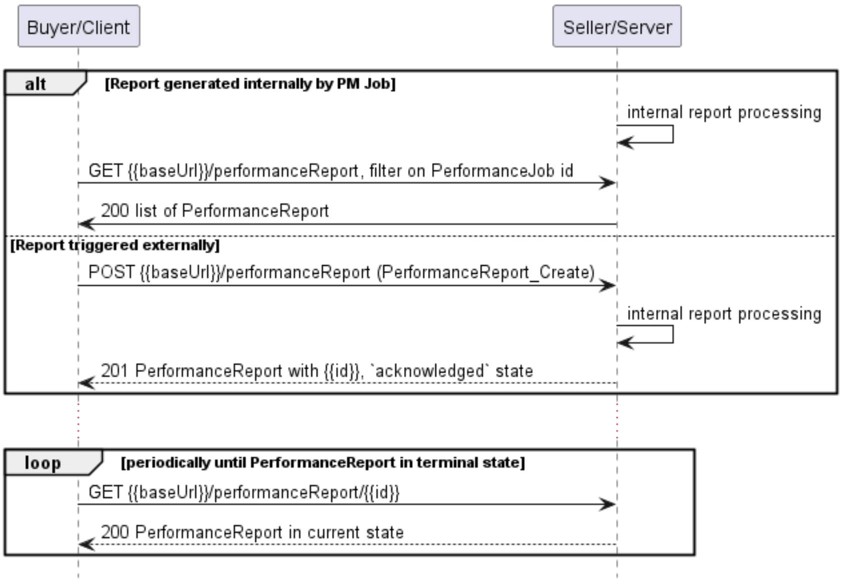

Buyer/Client
Seller/Server
alt
[Report generated internally by PM Job]
internal report processing
GET {{baseUrl}}/performanceReport, filter on PerformanceJob id
200 list of PerformanceReport
[Report triggered externally]
POST {{baseUrl}}/performanceReport (PerformanceReport_Create)
internal report processing
201 PerformanceReport with {{id}}, `acknowledged` state
loop
[periodically until PerformanceReport in terminal state]
GET {{baseUrl}}/performanceReport/{{id}}
200 PerformanceReport in current state

Figure 36. Performance Job progress tracking - Polling

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

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.

6.18.2. Create Performance Monitoring Report Request

Create Performance Report covers the use case Retrieve PM data from a PM database defined in Mplify133.1.

Figure 37 presents the most important part of the data model used for 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 (link to the generated file or content of the report).

Creation of Performance Report involves providing reporting timeframe, output format, subject of reporting (Service, Service Pair, or Entity), type of statistics to be taken, and other attributes.

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

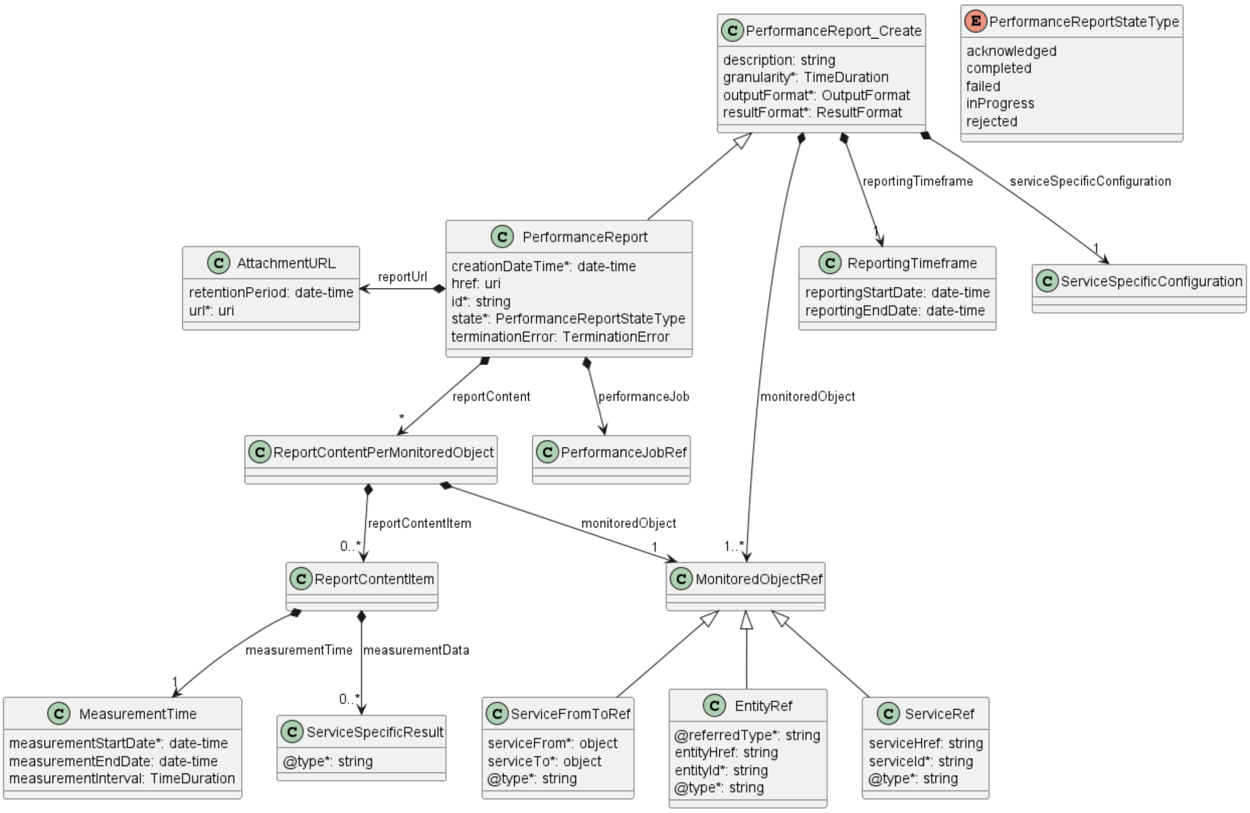


Figure 37. Performance Report Key Entities

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

`Performance Monitoring Report` Create Request

```
{
  "description": "Exemplary Create Performance Report request",
  "granularity": {
    "timeDurationValue": 15,
    "timeDurationUnits": "MIN"
  },
  "monitoredObject": [
    {
      "@type": "ServiceFromToRef",
      "serviceFromId": "d6c998f0-c723-4754-81fe-ad3ef7c15683",
      "serviceToId": "e02ee8c7-b6ec-4ab7-8307-b1c4cdba2891"
    }
  ],
  "outputFormat": "json",
  "reportingTimeframe": {
    "reportingStartDate": "2023-06-01T00:00:00.00",
    "reportingEndDate": "2023-06-02T00:00:00.00"
  },
  "resultFormat": "payload",
  "serviceSpecificConfiguration": {
    "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
    "packetsIn": true,
    "charsIn": true,
    "packetsOut": true,
    "charsOut": true
  }
}
```

[R65] If the Buyer/Client desires to retrieve PM data for a service they **MUST** include the `ServiceRef` in the Create Performance Monitoring Report request. [Mplify133.1 R159]

[R66] If the Buyer/Client desires to retrieve PM data for an ordered pair they **MUST** include the `ServiceFromToRef` in the Create Performance Monitoring Report request. [Mplify133.1 R159]

[R67] If the Buyer/Client desires to retrieve PM data for an entity they **MUST** include the `EntityRef` in the Create Performance Monitoring Report request. [Mplify133.1 R160]

[R68] The Buyer/Client **MUST** provide one and only one type of the following: [Mplify133.1 R161]

- `ServiceRef`
- `EntityRef`
- `ServiceFromToRef`

[R69] In addition to the `ServiceRef`, `ServiceFromToRef` or `EntityRef`, a Retrieve PM Data from a PM Database request **MUST** contain the following: [Mplify133.1 R162]

- `granularity`
- `outputFormat`
- `reportingTimeframe`
- `resultFormat`
- `serviceSpecificConfiguration`

6.18.3. Create Performance Monitoring Report Response

Figure 37 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 to the original `PerformanceReport_Create` object used in the Buyer/Client request. Additional attributes in the response include: `id`, `state`, `reportUrl` for accessing the file with generated report, or `reportContent` for including measurement data in the response payload, and other.

Depending on the `resultFormat` attribute, Seller/Server will provide a link to the generated report (`resultFormat=attachment`), or include captured values inside the response body (`resultFormat=payload`). This applies to reports that were successfully processed by the Seller/Server (state is `completed`).

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.

A Performance Report can be created by specifying multiple objects in the request, as the `monitoredObject` attribute is an array. As a result, the generated report will contain measurement data for each of these objects. The `reportContent` attribute addresses this by using the `ReportContentPerMonitoredObject` class, which pairs a monitored object reference with all performance data points collected for that object.

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

Create `PerformanceReport` Response

```
{
  "description": "Exemplary Create Performance Report request",
  "granularity": {
    "timeDurationValue": 15,
    "timeDurationUnits": "MIN"
  },
  "monitoredObject": [
    {
      "@type": "ServiceFromToRef",
      "serviceFromId": "d6c998f0-c723-4754-81fe-ad3ef7c15683",
      "serviceToId": "e02ee8c7-b6ec-4ab7-8307-b1c4cdba2891"
    }
  ],
  "outputFormat": "json",
  "reportingTimeframe": {
    "reportingStartDate": "2023-06-01T00:00:00.00",
    "reportingEndDate": "2023-06-02T00:00:00.00"
  },
  "resultFormat": "payload",
  "serviceSpecificConfiguration": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "packetsIn": true,
      "charsIn": true,
      "packetsOut": true,
      "charsOut": true
  },
  "reportContent": [
    {
      "monitoredObject":{
        "@type": "ServiceFromToRef",
        "serviceFromId": "d6c998f0-c723-4754-81fe-ad3ef7c15683",
        "serviceToId": "e02ee8c7-b6ec-4ab7-8307-b1c4cdba2891"
      },
      "reportContentItem": [
        {
          "measurementTime": {
            "measurementStartDate": "2023-06-01T00:00:00.00",
            "measurementEndDate": "2023-06-01T01:00:00.00"
          },
          "measurementData": [
            {
              "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-results:v0.0.1:all",
              "packetsIn": 100000,
              "charsIn": 8000000,
              "packetsOut": 95000,
              "charsOut": 7600000
            }
          ]
        }
      ]
    }
  ], << added by SOF >>
  "creationDateTime": "2023-06-01T08:02:01.370Z", << added by SOF >>
   "href": "{{baseUrl}}/performanceMonitoring/v5/performanceReport/8ae5f9f3-554f-4d93-8314-1630f171da54",  << added
by SOF >>
  "id": "8ae5f9f3-554f-4d93-8314-1630f171da54", << 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.

[R70] The Seller/Server's response **MUST** include all and unchanged attributes' values as provided by the Buyer/Client in the request.

[R71] The Seller/Server **MUST** specify the following attributes in a response:

- `creationDateTime`
- `id`
- `state`

[R72] The `id` **MUST** remain the same value for the life of the Performance Report.

6.18.4. Performance Monitoring Report State Machine

Figure 38 presents the Performance Report state machine:

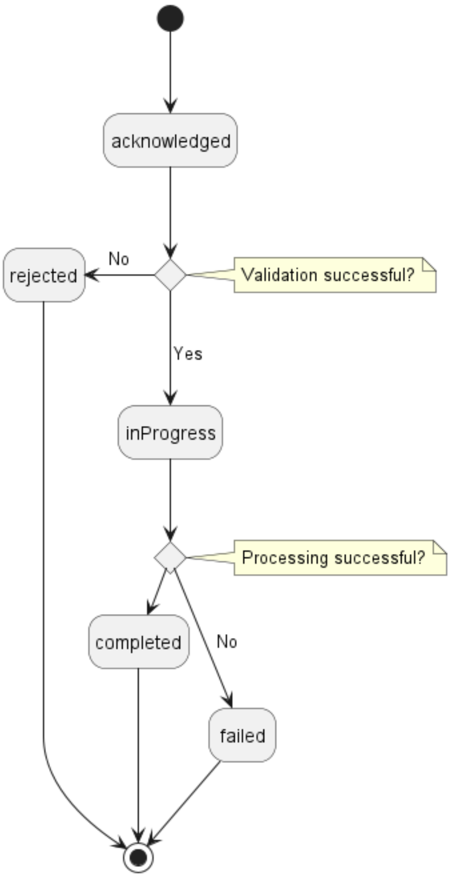


Figure 38. Performance Report State Machine

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.terminationError` acts as a placeholder to provide a detailed description of what caused the problem. `PerformanceReport` moves to `inProgress` state during which report content is collected into the report. Depending on the outcome of the processing, `PerformanceReport` moves to `completed` or `failed` state.

Table 10 presents the list of `status` names and their descriptions.

State	Description

State	Description
`acknowledged`	A Create Performance Report request has been received by the Seller/Server and has passed basic validations. Performance Report Identifier is assigned in the Acknowledged state. The report remains Acknowledged 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.
`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 10. Performance Report State Machine states

[R73] The Seller/Server **MUST** support all Performance Report statuses and their associated transitions as described in Figure 38 and Table 10.

6.19. Use Case 19: Retrieve Performance Monitoring Report List

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

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/performanceReport?state=completed&limit=10&offset=0
```

[O16] The Buyer's/Client's Retrieve List of Performance Reports request **MAY** contain none or more of the following attributes as filter criteria: [Mplify133.1 O15]

- `performanceJobId`
- `serviceFromId`
- `serviceToId`
- `serviceId`
- `entityId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`
- `reportingTimeframe.startDate.gt`
- `reportingTimeframe.startDate.lt`
- `reportingTimeframe.endDate.gt`
- `reportingTimeframe.endDate.lt`
- `outputFormat`
- `resultFormat`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. Section 6.2 provides details about the implementation of pagination mechanism.

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`.

Since a `PerformanceReport` can be created for more than one `monitoredObject`, and the Retrieve List of Performance Reports use case supports single values as a `monitoredObject` filter, the list of returned `PerformanceReport_Find` should include all reports that contain the provided filter value among their `monitoredObject` values.

[R74] The Seller/Server's response to the Buyer's/Client's Retrieve List of Performance Monitoring Reports **MUST** include the following attributes as applicable: [Mplify133.1 R87, R88, R117]

- `creationDateTime`
- `granularity`
- `id`
- `monitoredObject`
- `outputFormat`
- `performanceJob` (if Performance Report created as a result of Performance Job execution)
- `reportingTimeframe`
- `resultFormat`
- `serviceSpecificConfiguration`
- `state`

[R75] In case no items matching the criteria are found, the Seller/Server **MUST** return a valid response with an empty list. [Mplify133.1 R118]

Figure 39 presents entities related to the use case.

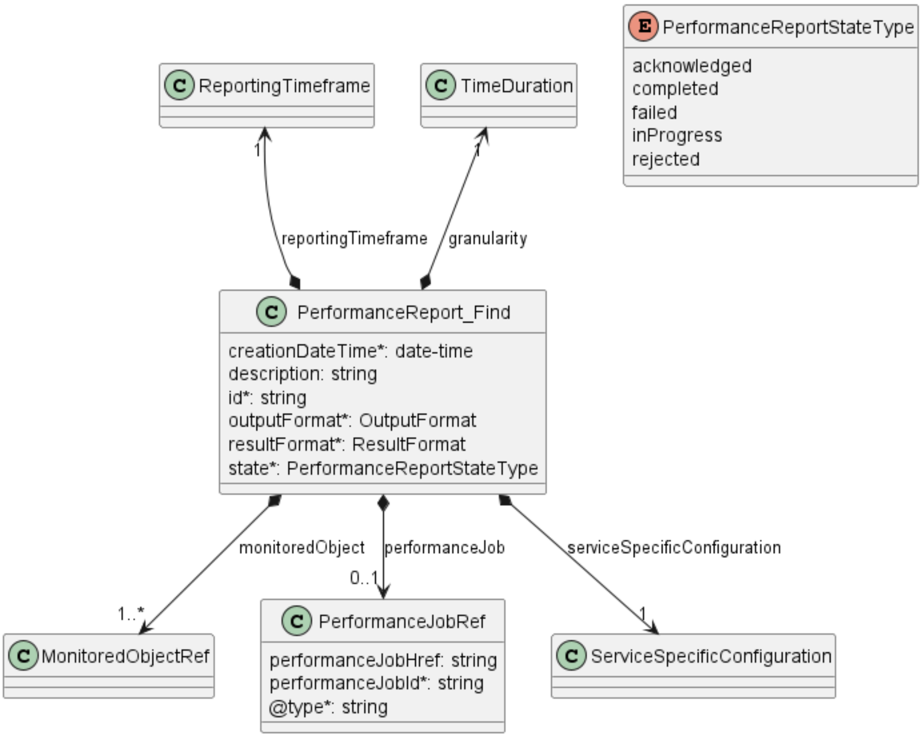


Figure 39. Use Case 19: Retrieve Performance Report List - Model

6.20. Use Case 20: Retrieve Performance Monitoring Report by Identifier

The Buyer/Client can get detailed information about a Performance Report from the Seller/Server by using a `GET /performanceReport/{id}` operation. The response payload provides a comprehensive representation of the Performance Report and encompass all attributes that the

Buyer/Client has provided when submitting a Create Performance Report request (or attributes of the Performance Job that generated the report), together with any attributes added by the Seller/Server, including the results of performance measurements collection. In case the Performance Report was created by Performance Job, it contains a reference to the Performance Job.

Get List and Get by Identifier operations return different representations of Performance Report. Get List returns the `PerformanceReport_Find` object which is a subset of `PerformanceReport` returned by the Get by Identifier operation. A response to a Get by Identifier for a `PerformanceReport` with `id=8ae5f9f3-554f-4d93-8314-1630f171da54` would return exactly the same response as presented in section 6.18.3. Specifically, the object returned by the Get by Identifier operation contains a collection of measurement results, either in the form of reference to a file, or directly within the returned `PerformanceReport` object. Measurement results are not returned by the Get List operation.

[R76] The Seller/Server **MUST** support at least one of the two methods of retrieving results: [Mplify133.1 R119]

- `attachment`
- `payload`

[O17] The Seller/Server **MAY** support multiple methods of retrieving results. [Mplify133.1 O20]

[R77] In order to retrieve Performance Monitoring Report with results in Service Payload, Performance Report **MUST** be created with:

- `reportFormat` set to `payload`
- `outputFormat` set to `json`

[R78] In order to retrieve Performance Monitoring Report with results in Attachment, Performance Report **MUST** be created with:

- `reportFormat` set to `attachment`
- `outputFormat` set to `json` or `xml`, or `avro`, or `csv`

[R79] The Seller/Server **MUST** include following attributes in the `PerformanceReport` object in the response:

- `creationDateTime`
- `granularity`
- `id`
- `monitoredObject`
- `outputFormat`
- `reportingTimeframe`
- `resultFormat`
- `serviceSpecificConfiguration`
- `state`

[R80] The Seller/Server **MUST** provide all remaining attributes if they were previously set by the Buyer or the Seller.

[R81] In case `id` does not match a `PerformanceReport` in Seller/Server's system, an error response `Error404` **MUST** be returned.

[R82] If the `resultFormat` is set to `payload` the Seller/Server **MUST** provide the specified result in the API payload. [Mplify133.1 R120]

[R83] If the resultFormat is set to attachment the Seller/Server **MUST** provide the specified results as an attachment. [Mplify133.1 R121]

[R84] The results regardless of the format **MUST** contain the PM results as specified by Performance Job or Performance Report create request using the outputFormat attribute. [Mplify133.1 R90]

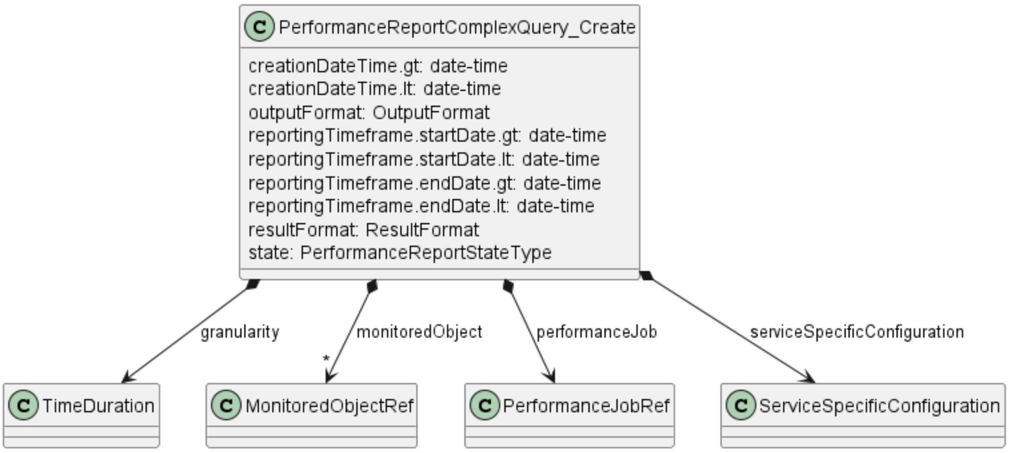
6.21. Use Case 21: Performance Monitoring Report Complex Query

The PerformanceReport defines complex structures with multiple levels of nesting, such as serviceSpecificConfiguration. To facilitate filtering based on these structures, the API provides an additional endpoint POST /performanceReportComplexQuery.

6.21.1. Performance Monitoring Report Complex Query Request

Figure 40 depicts the key components of the data model utilized in the Performance Report Complex Query request (POST /performanceReportComplexQuery). The request message model, PerformanceReportComplexQuery_Create, includes only attributes that can be specified by the Buyer/Client, representing filtering options. In response, the Seller/Server provides a list of PerformanceReport_Find entities that contain the matched Performance Report objects.

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



C PerformanceReportComplexQuery_Create
creationDateTime.gt: date-time
creationDateTime.lt: date-time
outputFormat: OutputFormat
reportingTimeframe.startDate.gt: date-time
reportingTimeframe.startDate.lt: date-time
reportingTimeframe.endDate.gt: date-time
reportingTimeframe.endDate.lt: date-time
resultFormat: ResultFormat
state: PerformanceReportStateType
granularity
monitoredObject
*
performanceJob
serviceSpecificConfiguration
C TimeDuration
C MonitoredObjectRef
C PerformanceJobRef
C ServiceSpecificConfiguration

Figure 40. Performance Report Complex Query Key Entities

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

- were created between 2023-06-01 08:00:00 and 2023-06-01 09:00:00
- outputFormat is JSON
- relate to specific Ordered Pair (Service Id From/To)
- are in completed state

Performance Report Complex Query Request

```
{
  "creationDateTime.gt": "2023-06-01T08:00:00.000Z",
  "creationDateTime.lt": "2023-06-01T09:00:00.000Z",
  "outputFormat": "json",
  "monitoredObject": [
    {
      "@type": "ServiceFromToRef",
      "serviceFromId": "d6c998f0-c723-4754-81fe-ad3ef7c15683",
      "serviceToId": "e02ee8c7-b6ec-4ab7-8307-b1c4cdba2891"
    }
  ],
  "state": "completed"
}
```

[O18] The Buyer's/Client's Performance Monitoring Report Complex Query request **MAY** contain none or more of the following attributes as filter criteria: [Mplify133.1 O15]

- `performanceJobId`
- `serviceFromId`
- `serviceToId`
- `serviceId`
- `entityId`
- `state`
- `creationDateTime.gt`
- `creationDateTime.lt`
- `reportingTimeframe.startDate.gt`
- `reportingTimeframe.startDate.lt`
- `reportingTimeframe.endDate.gt`
- `reportingTimeframe.endDate.lt`
- `outputFormat`
- `resultFormat`

6.21.2. Performance Monitoring Report Complex Query Response

The Seller/Server responds with a list of `PerformanceReport_Find` objects, which represent matched Performance Reports.

Since a `PerformanceReport` can be created for more than one `monitoredObject`, and the Performance Monitoring Report Complex Query use case supports an array as a filter on `monitoredObject`, the list of returned `PerformanceReport_Find` objects should include reports that have the provided filter values as a subset of their `monitoredObject` values.

The following snippet presents the Seller/Server response.

`Performance Report Complex Query` Response

```
[
  {
    "creationDateTime": "2023-06-01T08:02:01.370Z",
    "description": "Exemplary Create Performance Report request",
    "granularity": {
      "timeDurationValue": 15,
      "timeDurationUnits": "MIN"
    },
    "id": "8ae5f9f3-554f-4d93-8314-1630f171da54",
    "monitoredObject": [
      {
        "@type": "ServiceFromToRef",
        "serviceFromId": "d6c998f0-c723-4754-81fe-ad3ef7c15683",
        "serviceToId": "e02ee8c7-b6ec-4ab7-8307-b1c4cdba2891"
      }
    ],
    "outputFormat": "json",
    "reportingTimeframe": {
      "reportingStartDate": "2023-06-01T00:00:00.00",
      "reportingEndDate": "2023-06-02T00:00:00.00"
    },
    "resultFormat": "payload",
    "serviceSpecificConfiguration": {
      "@type": "urn:mef:lso:spec:legato:ip-performance-monitoring-configuration:v0.0.1:all",
      "packetsIn": true,
      "charsIn": true,
      "packetsOut": true,
      "charsOut": true
    },
    "state": "completed"
  }
]
```

[R85] The Seller/Server's response to the Buyer's/Client's Performance Report Complex Query **MUST** include the following attributes as applicable: [Mplify133.1 R87, R88, R117]

- `creationDateTime`
- `granularity`
- `id`
- `monitoredObject`
- `outputFormat`
- `performanceJob` (if Performance Report created as a result of Performance Job execution)
- `reportingTimeframe`
- `resultFormat`
- `serviceSpecificConfiguration`
- `state`

6.22. Use Case 22: Retrieve Tracking Record List

Tracking Records provide history 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 `PmTrackingRecord` type.

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

```
https://serverRoot/mefApi/legato/performanceMonitoring/v5/trackingRecord?relatedPerformanceJobId=755e55e2-72b0-4e3b-af00-693e3beac691&limit=10&offset=0
```

[R86] The Buyer/Client Retrieve List of Tracking Record request **MUST** contain none or more of the following attributes: [Mplify133.1 R163]

- `relatedPerformanceJobId`
- `relatedPerformanceProfileId`
- `relatedPerformanceReportId`
- `creationDateTimeTime.gt`
- `creationDateTimeTime.lt`

The Buyer may also ask for pagination with the use of the `offset` and `limit` parameters. Section6.2 provides details about the implementation of pagination mechanism.

[R87] The Seller/Server **MUST** include following attributes (if set) in the `PmTrackingRecord` object in the response: [Mplify133.1 R164]

- `creationDateTimeTime`
- `description`
- `id`
- `relatedObjectId`
- `request`
- `source`

[R88] In case no items matching the criteria are found, the Seller/Server 8 **MUST** return a valid response with an empty list.

Figure 41 presents the main Tracking Record entities.

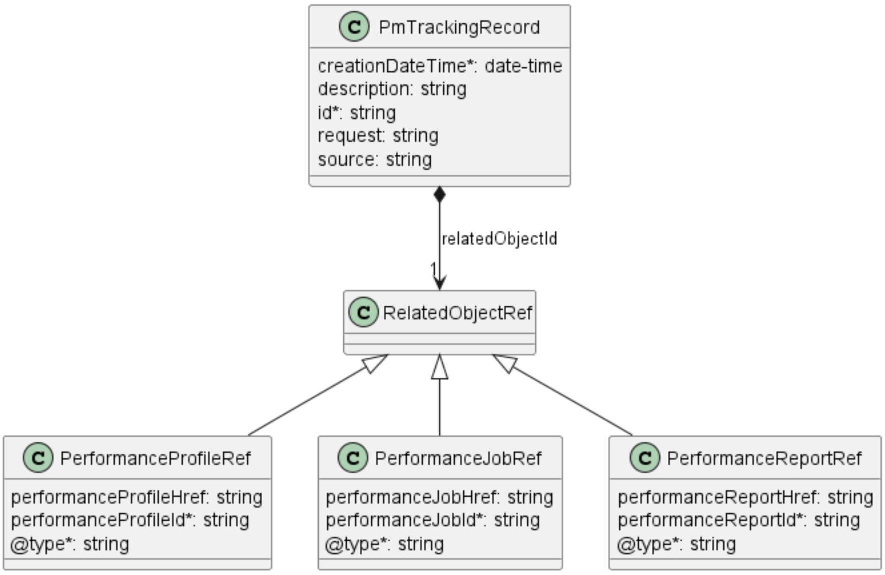


Figure 41. Tracking Record Model

6.23. Use Case 23: Register for Notifications

The Buyer/Client can track the lifecycle of the PM API entities by subscribing to notifications. An exemplary use case for exchanging notifications is presented in Figure 42.

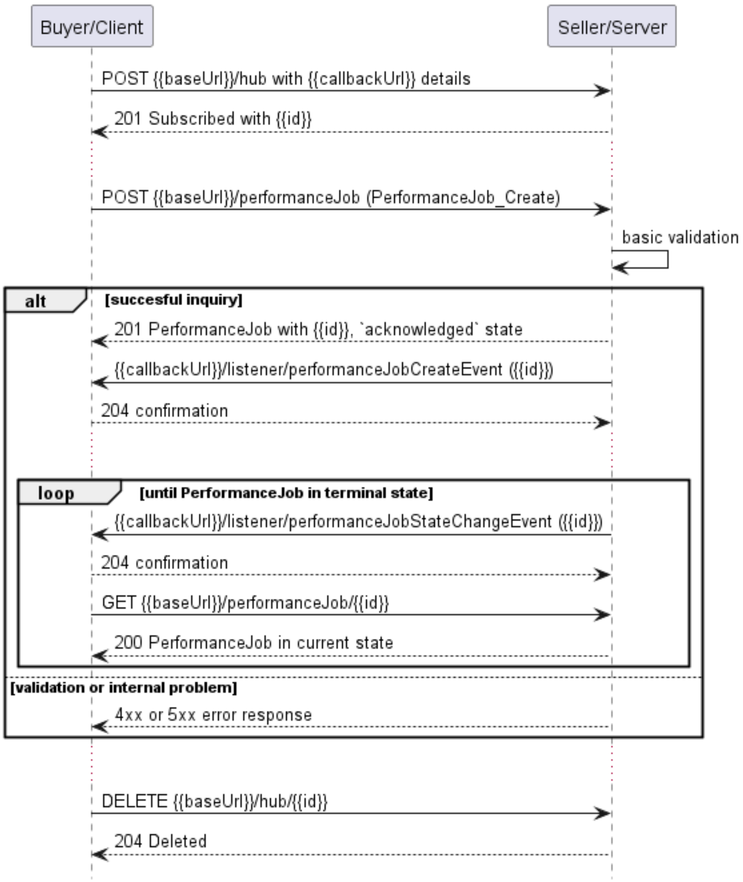

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}} details
201 Subscribed with {{id}}
POST {{baseUrl}}/performanceJob (PerformanceJob_Create)
basic validation
alt
[succesful inquiry]
201 PerformanceJob with {{id}}, `acknowledged` state
{{callbackUrl}}/listener/performanceJobCreateEvent ({{id}})
204 confirmation
loop
[until PerformanceJob in terminal state]
{{callbackUrl}}/listener/performanceJobStateChangeEvent ({{id}})
204 confirmation
GET {{baseUrl}}/performanceJob/{{id}}
200 PerformanceJob in current state
[validation or internal problem]
4xx or 5xx error response
DELETE {{baseUrl}}/hub/{{id}}
204 Deleted

Figure 42. 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 two attributes:

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

Figure 43 shows all entities involved in the Notification use cases.

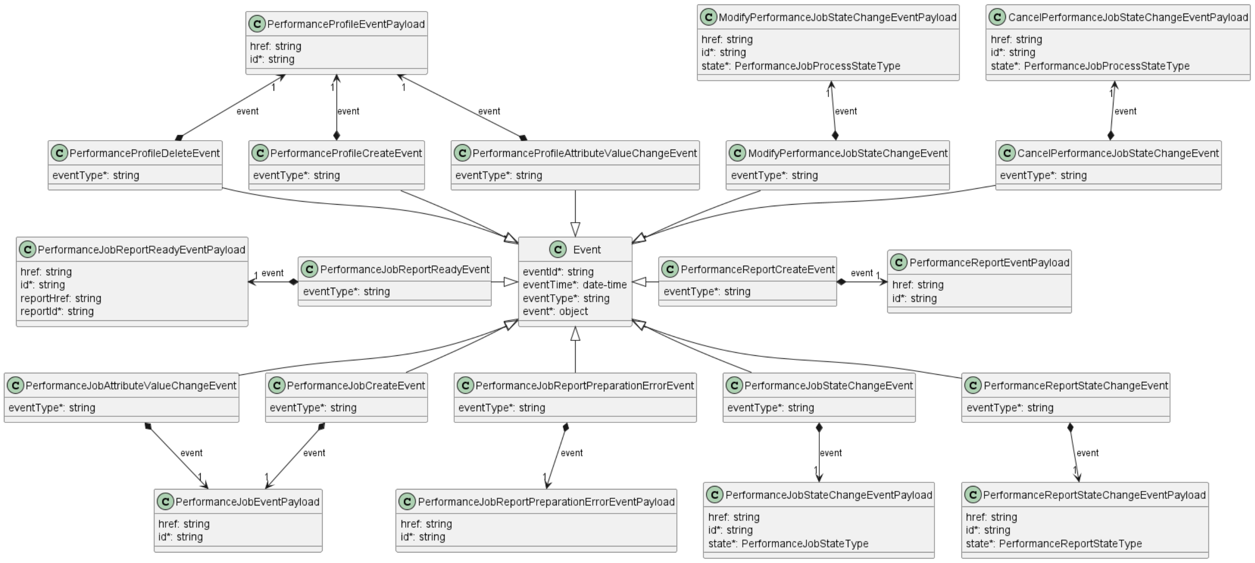

PerformanceProfileEventPayload
href: string
id*: string
ModifyPerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobProcessStateType
CancelPerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobProcessStateType
1
event
PerformanceProfileDeleteEvent
eventType*: string
PerformanceProfileCreateEvent
eventType*: string
PerformanceProfileAttributeValueChangeEvent
eventType*: string
ModifyPerformanceJobStateChangeEvent
eventType*: string
CancelPerformanceJobStateChangeEvent
eventType*: string
PerformanceJobReportReadyEventPayload
href: string
id*: string
reportHref: string
reportId*: string
PerformanceJobReportReadyEvent
eventType*: string
Event
eventId*: string
eventTime*: date-time
eventType*: string
event*: object
PerformanceReportCreateEvent
eventType*: string
PerformanceReportEventPayload
href: string
id*: string
PerformanceJobAttributeValueChangeEvent
eventType*: string
PerformanceJobCreateEvent
eventType*: string
PerformanceJobReportPreparationErrorEvent
eventType*: string
PerformanceJobStateChangeEvent
eventType*: string
PerformanceReportStateChangeEvent
eventType*: string
PerformanceJobEventPayload
href: string
id*: string
PerformanceJobReportPreparationErrorEventPayload
href: string
id*: string
PerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobStateType
PerformanceReportStateChangeEventPayload
href: string
id*: string
state*: PerformanceReportStateType

Figure 43. 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`
- `performanceProfileCreateEvent`
- `performanceProfileAttributeValueChangeEvent`
- `performanceProfileDeleteEvent`
- `performanceReportCreateEvent`
- `performanceReportStateChangeEvent`

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

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

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

[R89] The Buyer/Client's Subscribe to Performance Job Notifications request **MUST** include: [Mplify133.1 R81]

- `callback`

If the Buyer/Client wishes to subscribe to two different types of events, there are two possible syntax variants [TMF 630]:

```
eventType=performanceJobStateChangeEvent,performanceJobReportReadyEvent
```

or

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

The `query` formatting complies with RFC 3986. 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.mplify.com/listenerEndpoint",
  "query": "eventType=performanceJobStateChangeEvent"
}
```

Example of a final address that the Notifications to which the Buyer/Client subscribed with request in previous snippets will be sent to (for `performanceJobStateChangeEvent`):

- `https://bus.mplify.com/listenerEndpoint/mefApi/legato/performanceNotification/v5/listener/performanceJobStateChangeEvent`

6.24. Use Case 24: Send Notification

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

Figure 44 presents notifications produced by Seller/Server during the lifecycle of `PerformanceJob` assuming that Buyer/Client subscribed to all event types.

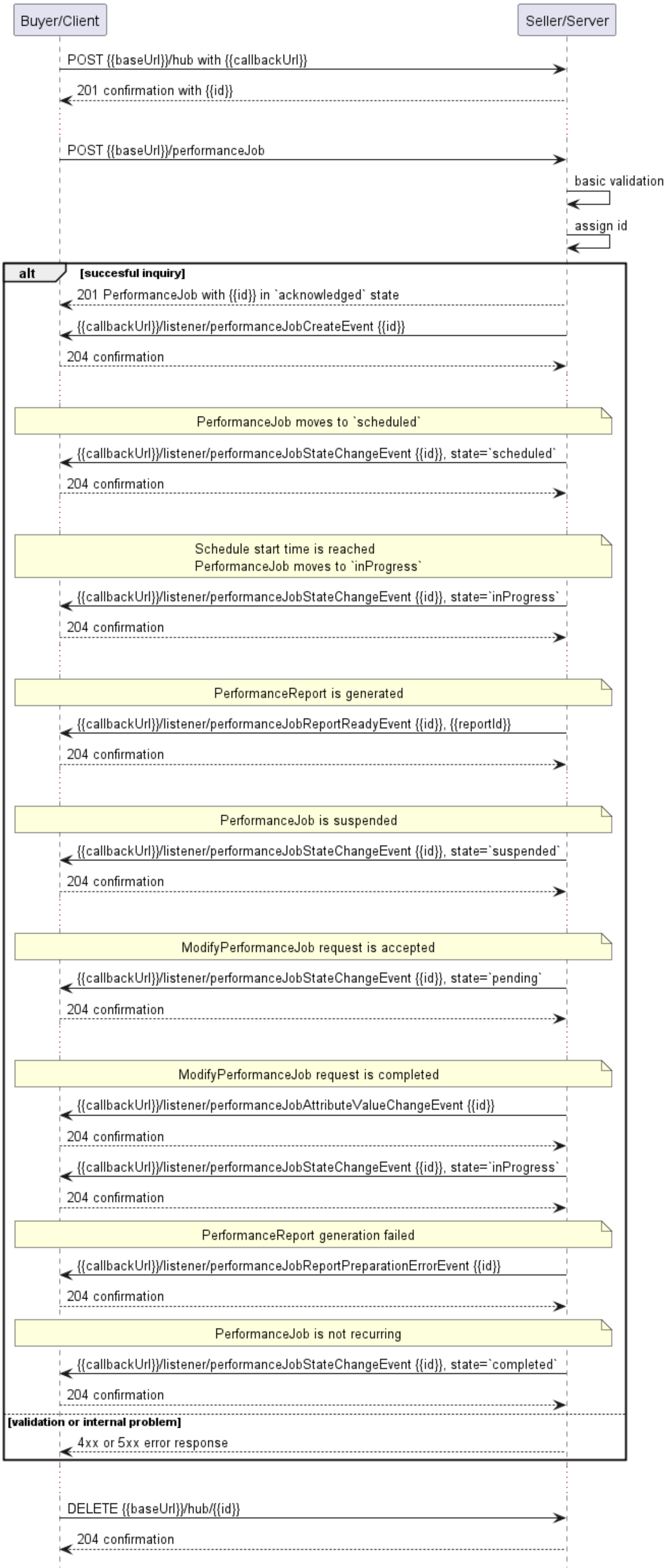

Buyer/Client
Seller/Server
POST {{baseUrl}}/hub with {{callbackUrl}}
201 confirmation with {{id}}
POST {{baseUrl}}/performanceJob
basic validation
assign id
alt
[succesful inquiry]
201 PerformanceJob with {{id}} in `acknowledged` state
{{callbackUrl}}/listener/performanceJobCreateEvent {{id}}
204 confirmation
PerformanceJob moves to `scheduled`
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`scheduled`
204 confirmation
Schedule start time is reached
PerformanceJob moves to `inProgress`
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`inProgress`
204 confirmation
PerformanceReport is generated
{{callbackUrl}}/listener/performanceJobReportReadyEvent {{id}}, {{reportId}}
204 confirmation
PerformanceJob is suspended
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`suspended`
204 confirmation
ModifyPerformanceJob request is accepted
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`pending`
204 confirmation
ModifyPerformanceJob request is completed
{{callbackUrl}}/listener/performanceJobAttributeValueChangeEvent {{id}}
204 confirmation
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`inProgress`
204 confirmation
PerformanceReport generation failed
{{callbackUrl}}/listener/performanceJobReportPreparationErrorEvent {{id}}
204 confirmation
PerformanceJob is not recurring
{{callbackUrl}}/listener/performanceJobStateChangeEvent {{id}}, state=`completed`
204 confirmation
[validation or internal problem]
4xx or 5xx error response
DELETE {{baseUrl}}/hub/{{id}}
204 confirmation

Figure 44. 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 an `acknowledged` state. Creation of `PerformanceJob` is notified with a `performanceJobCreateEvent`. When the validation is successful and the Performance Job is not immediate, it moves to `scheduled` and a `performanceJobStateChangeEvent` with state=`scheduled` is sent. When the scheduled start time is reached, `PerformanceJob` moves to `inProgress` status and the `performanceJobStateChangeEvent` with state=`inProgress` is sent. Performance Job periodically produces a Performance Report. This is when the `performanceJobReportReadyEvent` including id of generated report is sent. Further actions, like suspension or modification trigger `performanceJobStateChangeEvent`. In addition, in the case of `PerformanceJob` modification, Seller/Server produces `performanceJobAttributeValueChangeEvent` notification. When report generation fails, `performanceJobReportPreparationErrorEvent` is generated providing failure reason.

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

```
{
  "eventId": "event-001",
  "eventTime": "2021-06-03T15:56:08.559Z",
  "eventType": "performanceJobCreateEvent",
  "event": {
    "id": "e1c4565f-8678-47d3-80a5-496597a8abe4"
  }
}
```

```
{
  "eventId": "event-002",
  "eventTime": "2023-01-15T20:45:24.796Z",
  "eventType": "performanceJobReportReadyEvent",
  "event": {
    "id": "e1c4565f-8678-47d3-80a5-496597a8abe4",
    "reportId": "b54e7020-0bca-11ee-be56-0242ac120002"
  }
}
```

The body of the event doesn't contain all details of the object. The Buyer/Client needs to query it later by `id` to get details.

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

[R90] The Seller/Server **MUST** send Performance Profile, Performance Job, Performance Report notifications to the Buyer/Client that have registered for them. [Mplify133.1 R82]

[R91] The Seller/Server **MUST NOT** send Peformance Profile, Performance Job, Performance Report notifications to Buyer/Client that have not registered for them. [Mplify133.1 R83]

[R92] If the Buyer/Client registered for Performance Job notifications, the Seller/Server **MUST** notify the Buyer/Client when Performance Job results are available. [Mplify133.1 R61, R95]

[R93] An event triggered by the Performance Report creation (`performanceJobReportReadyEvent`) **MUST** additionally contain the identifier of the Report. [Mplify133.1 R84]

[R94] The Seller/Server **MUST** include the following attributes in the Performance Job State Change Notification: [Mplify133.1 R84]

- `id` of the Performance Job
- `state` of the Performance Job

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 Performance Monitoring 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 RFC7231. In such a case, the error message body structure might be aligned with the `Error`.

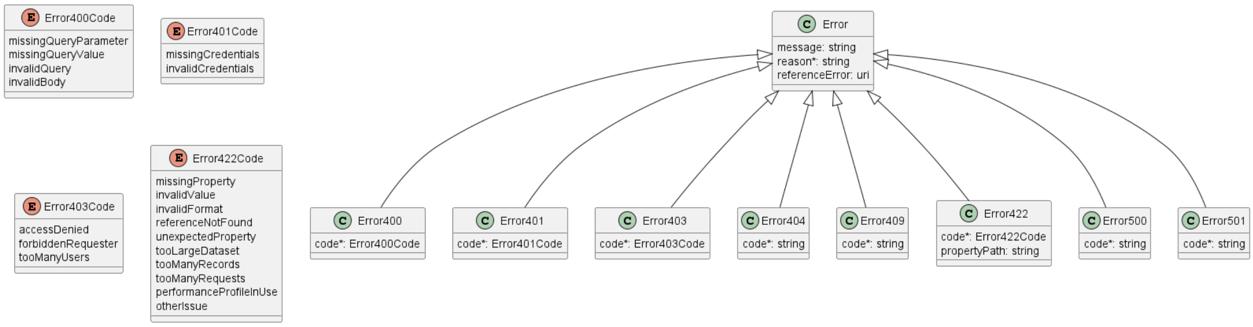


Figure 45. Data model types to represent an erroneous response

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.

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.

7.1.1.6. Type Error403

Description: Forbidden. This code indicates that the server understood the request but refused 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.

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 of the target resource not found.

7.1.1.9. Type Error409

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

Inherits from:

- Error

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

7.1.1.10. Type Error422

Description: Unprocessable entity due to a business validation problem. (https://datatracker.ietf.org/doc/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).

7.1.1.11. `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: The requested entity will produce too much 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).

7.1.1.12. Type Error500

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

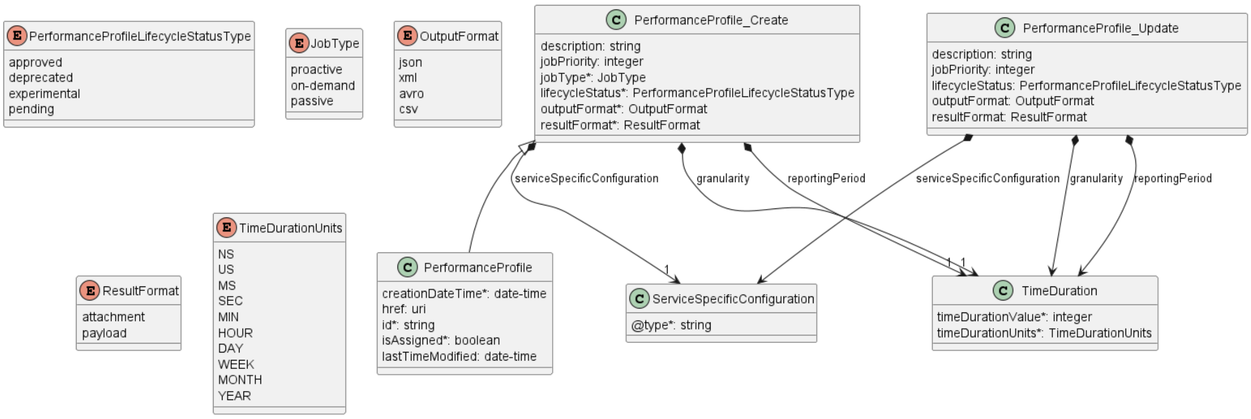
Inherits from:

- Error

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

7.2. Management API Data model

Figures 46-53 present the full Performance Monitoring data model. The data types, requirements related to them, and mapping to Mplify 133.1 specification are discussed later in this section.



E PerformanceProfileLifecycleStatusType
approved
deprecated
experimental
pending
E JobType
proactive
on-demand
passive
E OutputFormat
json
xml
avro
csv
C PerformanceProfile_Create
description: string
jobPriority: integer
jobType*: JobType
lifecycleStatus*: PerformanceProfileLifecycleStatusType
outputFormat*: OutputFormat
resultFormat*: ResultFormat
C PerformanceProfile_Update
description: string
jobPriority: integer
lifecycleStatus: PerformanceProfileLifecycleStatusType
outputFormat: OutputFormat
resultFormat: ResultFormat
serviceSpecificConfiguration
granularity
reportingPeriod
serviceSpecificConfiguration
granularity
reportingPeriod
1
1
1
E TimeDurationUnits
NS
US
MS
SEC
MIN
HOUR
DAY
WEEK
MONTH
YEAR
E ResultFormat
attachment
payload
C PerformanceProfile
creationDateTime*: date-time
href: uri
id*: string
isAssigned*: boolean
lastTimeModified: date-time
C ServiceSpecificConfiguration
@type*: string
C TimeDuration
timeDurationValue*: integer
timeDurationUnits*: TimeDurationUnits

Figure 46. Performance Monitoring Data Model - Performance Profile Entities

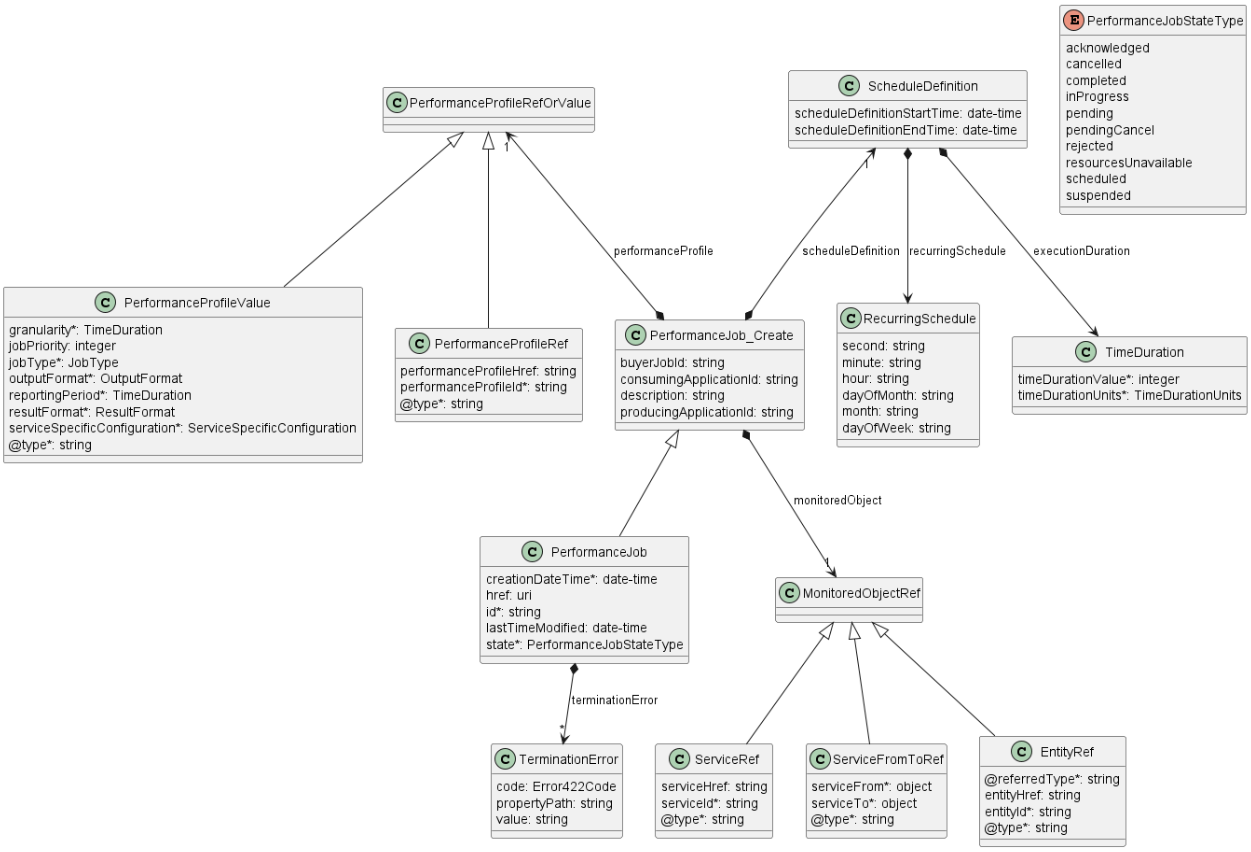


Figure 47. Performance Monitoring Data Model - Performance Job Entities

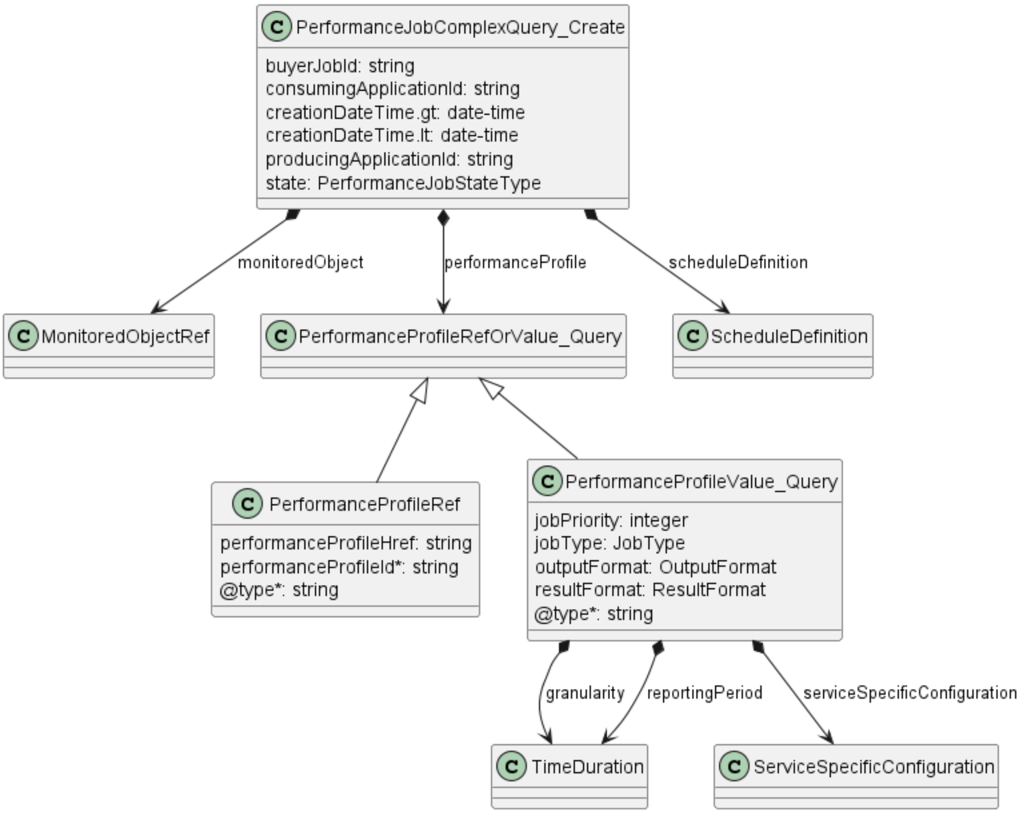

PerformanceJobComplexQuery_Create
buyerJobId: string
consumingApplicationId: string
creationDateTime.gt: date-time
creationDateTime.lt: date-time
producingApplicationId: string
state: PerformanceJobStateType
monitoredObject
performanceProfile
scheduleDefinition
MonitoredObjectRef
PerformanceProfileRefOrValue_Query
ScheduleDefinition
PerformanceProfileRef
performanceProfileHref: string
performanceProfileId*: string
@type*: string
PerformanceProfileValue_Query
jobPriority: integer
jobType: JobType
outputFormat: OutputFormat
resultFormat: ResultFormat
@type*: string
granularity
reportingPeriod
serviceSpecificConfiguration
TimeDuration
ServiceSpecificConfiguration

Figure 48. Performance Monitoring Data Model - Query Performance Job Entities

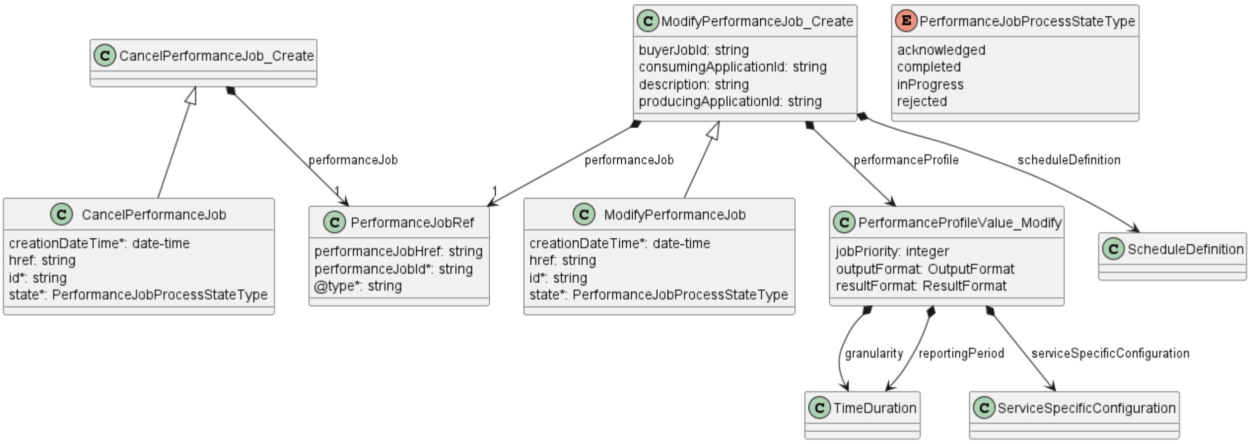

CancelPerformanceJob_Create
ModifyPerformanceJob_Create
buyerJobId: string
consumingApplicationId: string
description: string
producingApplicationId: string
PerformanceJobProcessStateType
acknowledged
completed
inProgress
rejected
performanceJob
1
performanceJob
1
performanceProfile
scheduleDefinition
CancelPerformanceJob
creationDateTime*: date-time
href: string
id*: string
state*: PerformanceJobProcessStateType
PerformanceJobRef
performanceJobHref: string
performanceJobId*: string
@type*: string
ModifyPerformanceJob
creationDateTime*: date-time
href: string
id*: string
state*: PerformanceJobProcessStateType
PerformanceProfileValue_Modify
jobPriority: integer
outputFormat: OutputFormat
resultFormat: ResultFormat
ScheduleDefinition
granularity
reportingPeriod
serviceSpecificConfiguration
TimeDuration
ServiceSpecificConfiguration

Figure 49. Performance Monitoring Data Model - Performance Job Process Entities

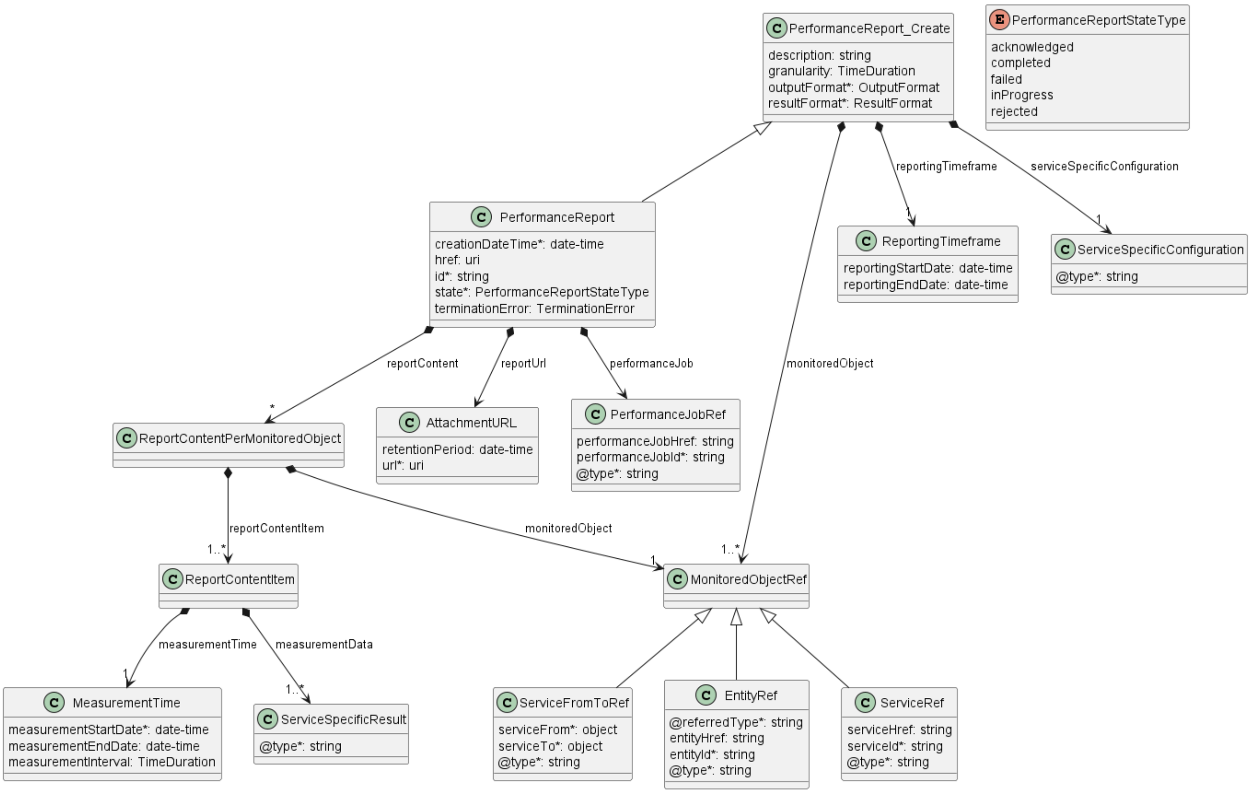


Figure 50. Performance Monitoring Data Model - Performance Report Entities

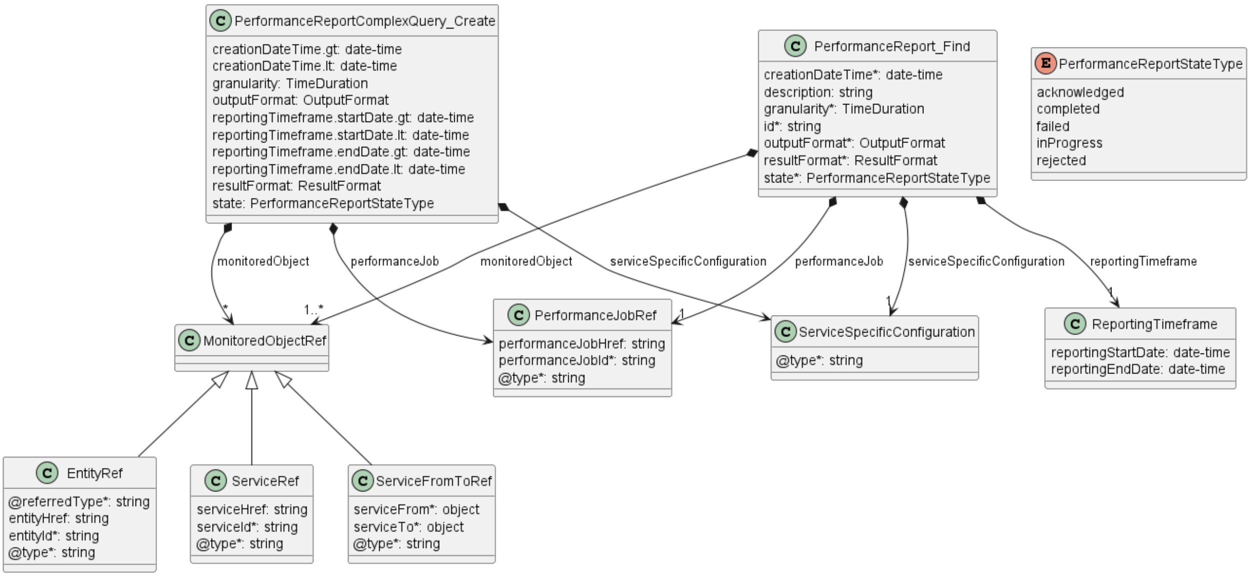

PerformanceReportComplexQuery_Create
creationDateTime.gt: date-time
creationDateTime.lt: date-time
granularity: TimeDuration
outputFormat: OutputFormat
reportingTimeframe.startDate.gt: date-time
reportingTimeframe.startDate.lt: date-time
reportingTimeframe.endDate.gt: date-time
reportingTimeframe.endDate.lt: date-time
resultFormat: ResultFormat
state: PerformanceReportStateType
PerformanceReport_Find
creationDateTime*: date-time
description: string
granularity*: TimeDuration
id*: string
outputFormat*: OutputFormat
resultFormat*: ResultFormat
state*: PerformanceReportStateType
PerformanceReportStateType
acknowledged
completed
failed
inProgress
rejected
monitoredObject
performanceJob
monitoredObject
serviceSpecificConfiguration
performanceJob
serviceSpecificConfiguration
reportingTimeframe
*
1..*
1
1
1
MonitoredObjectRef
PerformanceJobRef
performanceJobHref: string
performanceJobId*: string
@type*: string
ServiceSpecificConfiguration
@type*: string
ReportingTimeframe
reportingStartDate: date-time
reportingEndDate: date-time
EntityRef
@referredType*: string
entityHref: string
entityId*: string
@type*: string
ServiceRef
serviceHref: string
serviceId*: string
@type*: string
ServiceFromToRef
serviceFrom*: object
serviceTo*: object
@type*: string

Figure 51. Performance Monitoring Data Model - Query Performance Report Entities

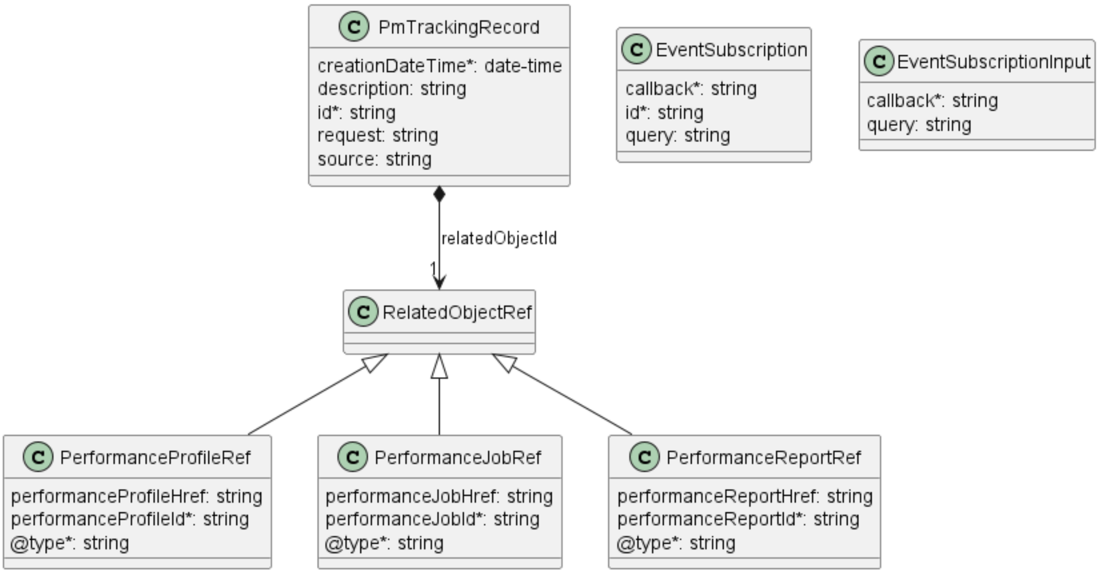


Figure 52. Performance Monitoring Data Model - Tracking Record and Event Subscription Entities

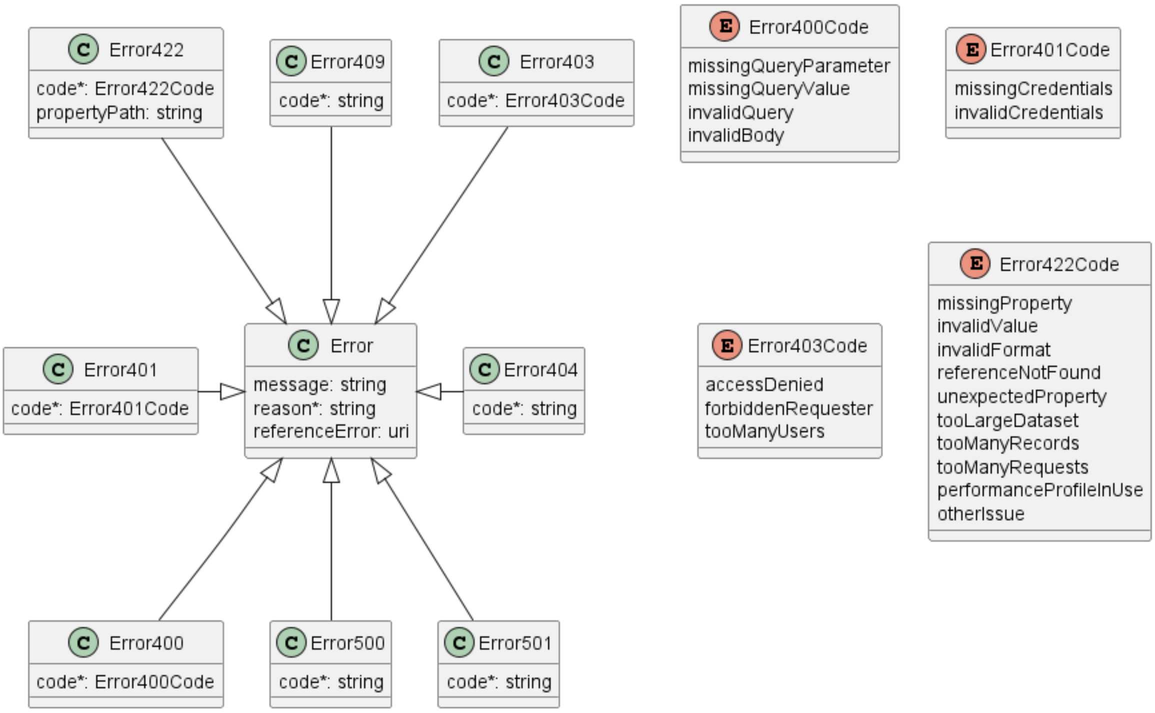


Figure 53. Performance Monitoring Data Model - Error Entities

7.2.1. PerformanceProfile

7.2.1.1. Type PerformanceProfile_Create

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

Name	Type	Description	Mplify 133.1
description	string	A free-text description of the Performance Profile.	Description
granularity*	TimeDuration	Sampling rate of the collection or production of performance indicators.	Granularity

Name	Type	Description	Mplify 133.1
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.	PM Job Priority
jobType*	JobType	The type of PM Job.	PM Job Type
lifecycleStatus*	PerformanceProfile LifecycleStatusType	Lifecycle Status is used to indicate the lifecycle status of the Performance Profile.	Lifecycle Status
outputFormat*	OutputFormat	List of possible output formats for the Performance Report.	Output Format
reportingPeriod*	T_TimeDuration	Defines the interval for the report generation.	Reporting Period
resultFormat*	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecific Configuration*	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration

7.2.1.2. Type PerformanceProfile

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

Inherits from:

- PerformanceProfile_Create

Name	Type	Description	Mplify 133.1
creationDateTime*	date-time	Date when Performance Profile was created.	Creation Time
href	uri	Hyperlink reference	Not present
id*	string	Unique identifier	PM Profile ID
isAssigned*	boolean	Indicates if Performance Profile is assigned to a Performance Job.	PM Profile Assigned
lastTimeModified	date-time	Date when the Performance Profile was last modified.	Last Time Modified

7.2.1.3. enum PerformanceProfileLifecycleStatusType

Description: Possible values for the Performance Profile Lifecycle Status.

State	Mplify 133.1 name	Description

State	Mplify 133.1 name	Description
`approved`	Approved	Performance Monitoring Profile has been approved for general use.
`deprecated`	In-Deprecated	Performance Monitoring Profile has been replaced by another profile.
`experimental`	Experimental	Performance Monitoring Profile use may be limited to a small number of users.
`pending`	Pending	Performance Monitoring Profile is waiting to be approved.

7.2.1.4. Type PerformanceProfile_Update

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

Name	Type	Description	Mplify 133.1
description	string	A free-text description of the Performance Profile	Description
granularity	TimeDuration	Sampling rate of the collection or production of performance indicators	Granularity
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.	PM Job Priority
lifecycleStatus	PerformanceProfile LifecycleStatusType	Lifecycle Status is used to indicate the lifecycle status of the Performance Profile.	Lifecycle Status
outputFormat	OutputFormat	List of possible output formats for the Performance Report.	Output Format
reportingPeriod	TimeDuration	Defines the interval for the report generation.	Reporting Period
resultFormat	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecific Configuration	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration

7.2.1.5. Type PerformanceProfileRef

Description: A reference to a Performance Profile resource

Name	Type	Description	Mplify 133.1

Name	Type	Description	Mplify 133.1
performanceProfileHref	string	Hyperlink to the referenced Performance Profile	Not present
performanceProfileId*	string	Identifier of the referenced Performance Profile	PM Profile ID
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.1.6. Type PerformanceProfileRefOrValue

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

7.2.1.7. Type PerformanceProfileRefOrValue_Query

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

7.2.1.8. Type PerformanceProfileValue

Description: The setting of the attributes defined by the PerformanceProfile class when the Performance Job does not have a reference to the Performance Profile.

Name	Type	Description	Mplify 133.1
granularity*	TimeDuration	Sampling rate of the collection or production of performance indicators	Granularity
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.	PM Job Priority
jobType*	JobType	The type of PM Job.	PM Job Type
outputFormat*	OutputFormat	List of possible output formats for the Performance Report.	Output Format
reportingPeriod*	TimeDuration	Defines the interval for the report generation.	Reporting Period
resultFormat*	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecificConfiguration*	ServiceSpecificConfiguration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.1.9. Type PerformanceProfileValue_Modify

Description: The setting of the attributes defined by the PerformanceProfile class when the Performance Job does not have a reference to the Performance Profile. This class is used to modify these Performance Job attributes.

Name	Type	Description	Mplify 133.1
granularity	TimeDuration	Sampling rate of the collection or production of performance indicators.	Granularity
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.	Job Priority
outputFormat	OutputFormat	List of possible output formats for the Performance Report.	Output Format
reportingPeriod	TimeDuration	Defines the interval for the report generation.	Reporting Period
resultFormat	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecific Configuration	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration

7.2.1.10. Type PerformanceProfileValue_Query

Description: The setting of the attributes defined by the PerformanceProfile class when the Performance Job does not have a reference to the Performance Profile.

Name	Type	Description	Mplify 133.1
granularity	TimeDuration	Sampling rate of the collection or production of performance indicators.	Granularity
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.	Job Priority
jobType	JobType	The type of PM Job.	Job Type
outputFormat	OutputFormat	List of possible output formats for the Performance Report.	Output Format
reportingPeriod	TimeDuration	Defines the Duration for the report generation.	Reporting Period
resultFormat	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecific Configuration	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.2. PerformanceJob

7.2.2.1. Type PerformanceJob_Create

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

Name	Type	Description	Mplify 133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	Buyer Job ID
consumingApplicationId	string	Identifier of consuming application	Consuming Application Indicator
description	string	A free-text description of the Performance Job	Description
monitoredObject*	MonitoredObjectRef	Defines the reference to object which is a subject of performance monitoring.	Service ID/Service Pair/Entity ID
performanceProfile*	PerformanceProfileRefOrValue	Defines the reference to Performance Monitoring Profile or defines values from PerformanceProfile type.	PM Profile ID
producingApplicationId	string	Identifier of producing application	Producing Application Identifier
scheduleDefinition*	ScheduleDefinition	The schedule definition for running jobs.	Schedule Definition

7.2.2.2. Type PerformanceJob

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

Inherits from:

- PerformanceJob_Create

Name	Type	Description	Mplify 133.1

Name	Type	Description	Mplify 133.1
creationDateTime*	date-time	Date when Performance Job was created.	Creation Date
href	uri	Hyperlink reference	Href
id*	string	Unique identifier	PM Job Identifier
lastTimeModified	date-time	Date when the Performance Job was last modified.	Last Time Modified
state*	PerformanceJobStateType	The state of the Performance Monitoring Job.	State
terminationError	TerminationError[]	When the Seller/Server cannot process the request, the Seller/Server returns a text-based list of reasons here.	Termination Error

7.2.2.3. Type CancelPerformanceJob_Create

Description: Request for cancellation of an existing Performance Job

Name	Type	Description	Mplify 133.1
performanceJob*	PerformanceJobRef	A reference to a Performance Job resource.	PM Job Identifier

7.2.2.4. Type CancelPerformanceJob

Description: Request for cancellation of an existing Performance Job

Inherits from:

- CancelPerformanceJob_Create

Name	Type	Description	Mplify 133.1
creationDateTime*	date-time	Date when Cancel Performance Job was created.	Creation Time
href	string	Hyperlink to the Cancel Performance Job entity	Not present
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`.	PM Job Cancellation Process Identifier
state*	PerformanceJob ProcessStateType	The state of the process related to the Performance Job.	PM Job Cancellation Process State

7.2.2.5. Type ModifyPerformanceJob_Create

Description: Request for modification of an existing Performance Job

Name	Type	Description	Mplify 133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	Buyer Job ID
consumingApplicationId	string	Identifier of consuming application	Consuming Application Indicator
description	string	A free-text description of the Performance Job	Description
performanceJob*	PerformanceJobRef	A reference to a Performance Job resource.	PM Job Identifier
performanceProfile	PerformanceProfileValue_Modify	Direct assignment of values defined by PerformanceProfile type to PerformanceJob object. Necessary when PerformanceJob is created without reference to PerformanceProfile.	PM Profile ID
producingApplicationId	string	Identifier of producing application	Producing Application Identifier
scheduleDefinition	ScheduleDefinition	The schedule definition for running jobs.	Schedule Definition

7.2.2.6. Type ModifyPerformanceJob

Description: Request for modification of an existing Performance Job

Inherits from:

- ModifyPerformanceJob_Create

Name	Type	Description	Mplify 133.1
creationDateTime*	date-time	Date when Modify Performance Job was created.	Creation Time
href	string	Hyperlink to the Modify Performance Job entity	Not present
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`	PM Job Modification Process Identifier

Name	Type	Description	Mplify 133.1
state*	PerformanceJob ProcessStateType	The state of the process related to the Performance Job.	PM Job Modification Process State

7.2.2.7. Type PerformanceJobComplexQuery_Create

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

Name	Type	Description	Mplify 133.1
buyerJobId	string	Identifier of the job understood and assigned by the Buyer/Client.	Buyer Job ID
consumingApplicationId	string	Identifier of consuming application	Consuming Application Indicator
creationDateTime.gt	date-time	Date when Performance Job was created - greater than.	Creation Date
creationDateTime.lt	date-time	Date when Performance Job was created - lower than.	Creation Date
monitoredObject	MonitoredObjectRef	Defines the reference to object which is a subject of performance monitoring.	Service ID/Service Pair/Entity ID
performanceProfile	PerformanceProfile RefOrValue_Query	Defines the reference to Performance Monitoring Profile or defines values from PerformanceProfile type.	PM Profile Reference
producingApplicationId	string	Identifier of producing application	Producing Application Identifier
scheduleDefinition	ScheduleDefinition	The schedule definition for running jobs.	Schedule Definition
state	PerformanceJobStateType	The state of the Performance Monitoring Job.	State

7.2.2.8. enum PerformanceJobProcessStateType

Description: The state of the process related to the Performance Job.

state	Mplify 133.1 name	Description
acknowledged	Acknowledged	The Cancel/Modify 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 Acknowledged until all validations as applicable are completed. If the attributes are validated, the request moves to the In-Progress state. If not all attributes are validated, the request moves to the Rejected state.
completed	Completed	The Cancel/Modify Performance Monitoring Job request has been completed by the Seller/Server.
inProgress	In-Progress	The Cancel/Modify Performance Monitoring Job request has been validated and accepted by the Seller/Server and is in-progress.
rejected	Rejected	The Cancel/Modify Performance Monitoring Job request has failed validation and has been declined by the Seller/Server.

7.2.2.9. Type PerformanceJobRef

Description: A reference to a Performance Job resource.

Name	Type	Description	Mplify 133.1
performanceJobHref	string	Hyperlink to the referenced Performance Job	Href
performanceJobId*	string	Identifier of the referenced Performance Job	PM Job Identifier
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.2.10. enum PerformanceJobStateType

Description: The state of the Performance Monitoring Job.

state	Mplify 133.1 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 Acknowledged 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. If scheduled, the Performance Monitoring Job moves to the Scheduled state. If not all attributes are validated, the request moves to the Rejected state.

state	Mplify 133.1 name	Description
`cancelled`	Cancelled	A Performance Monitoring Job that was In-Progress, Suspended, or Scheduled is cancelled.
`completed`	Completed	A non-recurring Performance Monitoring Job finished execution.
`inProgress`	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. If 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 Pending 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.
`pendingCancel`	Pending Cancel	A Cancel Performance Monitoring Job request has been accepted by the Seller/Server. The Performance Monitoring Job remains Pending Cancel while resources used by the Job are being released. Once updates are complete, the Job moves to the Cancelled status.
`rejected`	Rejected	A create Performance Monitoring Job request fails validation and is rejected with error indications by the Seller/Server.
`resourcesUnavailable`	Resources 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 Scheduled until the start time is reached. The Performance Monitoring Job then moves to In-Progress. If the Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the Pending state.

state	Mplify 133.1 name	Description
suspended	Suspended	A Suspend Performance Monitoring Job request is accepted by the Seller/Server. The Job remains Suspended 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 the Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the Pending state.

7.2.3. PerformanceReport

7.2.3.1. Type PerformanceReport_Create

Description: The creation of the PM Report enables the collection of PM results, which are stored in the PM database within the Seller/Server system.

Name	Type	Description	Mplify 133.1
description	string	A free-text description of the performance report	Description
granularity*	TimeDuration	Sampling rate of the collection or production of performance indicators.	Granularity
monitoredObject*	MonitoredObjectRef[]	Defines the reference to object which is a subject of performance monitoring. Constraint: All items in the array must be of the same type - either all EntityRef, all ServiceFromToRef, or all ServiceRef. Mixing different types within the same array is not allowed and will be rejected by the server.	Service ID/Service Pair/Entity ID
outputFormat*	OutputFormat	List of possible output formats for the Performance Report.	Output format
reportingTimeframe*	ReportingTimeframe	Specifies the date range between which data points will be included in the report.	Reporting Timeframe
resultFormat*	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format

Name	Type	Description	Mplify 133.1
serviceSpecificConfiguration*	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration

7.2.3.2. Type PerformanceReport

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

Inherits from:

- PerformanceReport_Create

Name	Type	Description	Mplify 133.1
creationDateTimeTime*	date-time	Date when Performance Report was created.	Creation Time
href	uri	Hyperlink reference	Not present
id*	string	Unique identifier	Report ID
performanceJob	PerformanceJobRef	Defines the reference to Performance Monitoring Job or defines values from PerformanceJob type.	PM Job Identifier
reportContent	ReportContentPerMonitoredObject[]		PM Report Content
reportUrl	AttachmentURL	The URL pointing to an Attachment for download.	PM Report URL
state*	PerformanceReportStateType	Possible values for the state of a Performance Report.	PM Report State

Name	Type	Description	Mplify 133.1
terminationError	TerminationError[]	When the Seller/Server cannot process the request, the Seller/Server returns a text-based list of reasons here.	Termination Error

7.2.3.3. Type PerformanceReport_Find

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

Name	Type	Description	Mplify 133.1
creationDateTime*	date-time	Date when the report was created.	Creation Time
description	string	A free-text description of the Performance Report	Description
granularity*	TimeDuration	Sampling rate of the collection or production of performance indicators.	Granularity
id*	string	Unique identifier	Report ID
monitoredObject*	MonitoredObjectRef[]	Defines the reference to object which is a subject of performance monitoring. Constraint: All items in the array must be of the same type - either all EntityRef, all ServiceFromToRef, or all ServiceRef. Mixing different types within the same array is not allowed and will be rejected by the server.	Service ID/Service Pair/Entity ID
outputFormat*	OutputFormat	List of possible output formats for the Performance Report.	Output Format
performanceJob*	PerformanceJobRef	Defines the reference to Performance Monitoring Job.	PM Job Identifier
reportingTimeframe*	ReportingTimeframe	Specifies the date range between which data points will be included in the report.	Reporting timeframe
resultFormat*	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format

Name	Type	Description	Mplify 133.1
serviceSpecific Configuration*	ServiceSpecific Configuration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Specific Configuration
state*	PerformanceReport StateType	Possible values for the state of a Performance Report.	State

7.2.3.4. Type PerformanceReportComplexQuery_Create

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

Name	Type	Description	Mplify 133.1
creationDateTime.gt	date-time	Date when Performance Report was created - greater than.	Creation Time greater than
creationDateTime.lt	date-time	Date when Performance Report was created - lower than.	Creation Time less than
granularity	TimeDuration	Sampling rate of the collection or production of performance indicators	Granularity
monitoredObject	MonitoredObjectRef[]	Defines the reference to object which is a subject of performance monitoring. Constraint: All items in the array must be of the same type - either all EntityRef, all ServiceFromToRef, or all ServiceRef. Mixing different types within the same array is not allowed and will be rejected by the server.	Service ID/Service Pair/Entity ID
outputFormat	OutputFormat	List of possible output formats for the Performance Report.	Output Format
performanceJob	PerformanceJobRef	Defines the reference to Performance Monitoring Job.	PM Job Identifier
reportingTimeframe. startDate.gt	date-time	Start date of reporting timeframe - greater than.	Reporting Timeframe start date greater than
reportingTimeframe. startDate.lt	date-time	Start date of reporting timeframe - lower than.	Reporting Timeframe start date less than

Name	Type	Description	Mplify 133.1
reportingTimeframe.endDate.gt	date-time	End date of reporting timeframe - greater than.	Reporting Timeframe end date greater than
reportingTimeframe.endDate.lt	date-time	End date of reporting timeframe - lower than.	Reporting Timeframe end date less than
resultFormat	ResultFormat	List of possible result formats that define how Seller/Server will deliver Performance Report to the Buyer/Client.	Result Format
serviceSpecificConfiguration	ServiceSpecificConfiguration	ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration.	Service Payload Specific Attributes
state	PerformanceReportStateType	Possible values for the state of a Performance Report.	State

7.2.3.5. Type PerformanceReportRef

Description: A reference to a Performance Report resource.

Name	Type	Description	Mplify 133.1
performanceReportHref	string	Hyperlink to the referenced Performance Report.	Href
performanceReportId*	string	Identifier of the referenced Performance Report.	Report ID
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.3.6. `enum` PerformanceReportStateType

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

State	Mplify 133.1 name	Description
`acknowledged`	Acknowledged	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 Acknowledged 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`	Completed	A Performance Report is completed and results are available.

State	Mplify 133.1 name	Description
`failed`	Failed	A Performance Report processing has failed.
`inProgress`	In-Progress	A Performance Report has successfully passed the validations checks and the report processing has started.
`rejected`	Rejected	This state indicates that: - Invalid information is provided in Create Performance Report request - The request fails to meet validation rules for Performance Report delivery (processing).

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 URL pointing to an Attachment for download.

Name	Type	Description	Mplify 133.1
retentionPeriod	date-time	A date until which the file will be retained.	Retention Period
url	uri	The URL pointing to an Attachment for download.	File Location

7.2.4.2. Type EntityId

Description: Identifier of an Entity.

7.2.4.3. Type EntityRef

Description: Reference to Entity.

Name	Type	Description	Mplify 133.1
@referredType*	string	The type of the referred object. This is used to distinguish between different types of objects that can be referred to by the EntityId.	Not present
entityHref	string	Hyperlink to the Entity resource. This is used to provide a link to the Entity resource in the Seller/Server system.	Not present
entityId*	string	Identifier of an Entity.	Entity ID
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not present

7.2.4.4. `enum` JobType

Description: The type of a Job. Proactive jobs are carried on continuously to permit timely reporting of fault or performance status. On-demand jobs are actions that are initiated for a limited time to carry out the measurements. Passive jobs support the collection and reporting of network and service statistics/faults.

Value	Mplify 133.1
proactive	PROACTIVE
on-demand	ON-DEMAND
passive	PASSIVE

7.2.4.5. Type MeasurementTime

Description: Timeframe boundary for collected data. Provide measurementStartDate and measurementEndDate or measurementStartDate and measurementInterval.

Name	Type	Description	Mplify 133.1
measurementStartDate*	date-time	Start date of the period to which collected data points belong.	Start Date
measurementEndDate	date-time	End date of the period to which collected data points belong.	End Date
measurementInterval	TimeDuration	Length of the measurement interval.	Measurement Interval

7.2.4.6. Type MonitoredObjectRef

Description: Defines the reference to object which is a subject of performance monitoring.

7.2.4.7. `enum` OutputFormat

Description: List of possible output formats for the Performance Report

Value	Mplify 133.1
json	JSON
xml	XML
avro	AVRO
csv	CSV

7.2.4.8. Type PmTrackingRecord

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

Name	Type	Description	Mplify 133.1
creationDateTimeTime*	date-time	Date when the record was created.	Creation Date

Name	Type	Description	Mplify 133.1
description	string	Free-text field describing the action that created the Tracking Record and its details.	Description
id*	string	Identifier of the Tracking Record	Identifier
relatedObjectId*	RelatedObjectRef	Reference to the Performance Job, Profile, or Report for which the Tracking Record was created.	Related Object Identifier
request	string	Contains the REST request used in the action that resulted in the creation of the Tracking Record.	Request
source	string	Describes the source (Buyer/Client or Seller/Server) from which the action was done.	Source

7.2.4.9. Type RecurringSchedule

Description: A definition of recurring schedule to run a job based on the Cron utility in Linux-like systems. It defines how the job should periodically run at specified times, dates, or intervals. Special characters can be used in the definition of individual fields, such as:

- `*` -> any value
- `,` -> value list separator
- `-` -> range of values
- '/' -> step values

Name	Type	Description	Mplify 133.1
second	string	A definition of time (seconds) to run a job. Allowed values: 0-59, and special characters. For example: - _/5 _ * * * * -> run a job at every 5th second. - _/30 _/1 * * * _ -> run a job at every 30 seconds past every minute.	Second
minute	string	A definition of time (minutes) to run a job. Allowed values: 0-59, and special characters: (,-_/). For example: - 0 _/10 _ * * * -> run a job at every 10th minute. - 0 _/30 0 _ * * -> run a job at every 30th minute past midnight. - 0 _/30 8 _ _ 1 -> run a job at every 30th minute past hour 8 on Monday.	Minute
hour	string	A definition of time (hour) to run a job. Allowed values: 0-23, and special characters: (,-_/). For example: - 0 0 10 10 * * -> run a job at 10 am on 10th day of every month. - 0 _ 1,2 _ * * -> run a job at every minute past hour 1 and 2. - 0 0 _/2 _ * _ -> run a job at every 2nd hour.	Hour

Name	Type	Description	Mplify 133.1
dayOfMonth	string	A definition of time (day of month) to run a job. Allowed values: 1-31, and special characters: (,-_/). For example: - 0 0 0 1,5,10,15 * * -> run a job at midnight on every 1st, 5th, 10th, 15th day of month. - 0 0 _/1 1-10 1-3 _ -> run a job at every full hour between 1st and 10th day of month in January, February, and March.	Day of Month
month	string	A definition of time (month) to run a job. Allowed values: 1-12 or JAN-DEC, and special characters: (,-*/). For example: - 0 5 0 _ 8 _ -> run a job at 00:05 on every day in August. - 0 0 0,12 1 _/2 _ -> run a job at midnight and noon on every 1st day of every 2nd month.	Month
dayOfWeek	string	A definition of time (day of week) to run a job. Allowed values: 0-6 or SUN-SAT, and special characters: (,-*/). For example: - 0 0 22 * * 1-5 -> run a job at 22:00 on every day between Monday and Friday. - 0 5 4 * * sun -> run a job at 04:05 on Sunday.	Day of Week

7.2.4.10. Type RelatedObjectRef

Description: Reference to the Performance Job, Profile, or Report for which the Tracking Record was created.

7.2.4.11. Type ReportContentItem

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

Name	Type	Description	Mplify 133.1
measurementTime*	MeasurementTime	Timeframe boundary for collected data.	Measurement Time
measurementData*	ServiceSpecificResult[]	List of performance monitoring values measured in the related timeframe.	Measurement Data

7.2.4.12. Type ReportContentPerMonitoredObject

Description: Result of the performance monitoring for a specific monitored object.

Name	Type	Description	Mplify 133.1
monitoredObject*	MonitoredObjectRef	Defines the reference to object which is a subject of performance monitoring.	Service ID/Service Pair/Entity ID
reportContentItem*	ReportContentItem[]	List of result items for the monitored object.	PM Report Content

7.2.4.13. Type ReportingTimeframe

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

Name	Type	Description	Mplify 133.1
reportingStartDate	date-time	Start date of reporting timeframe.	Reporting Timeframe
reportingEndDate	date-time	End date of reporting timeframe.	Reporting Timeframe

7.2.4.14. `enum` ResultFormat

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

Value	Mplify 133.1
attachment	attachment
payload	payload

7.2.4.15. Type ScheduleDefinition

Description: The schedule definition for running jobs.

Name	Type	Description	Mplify 133.1
scheduleDefinitionStartTime	date-time	The start time of the Schedule Definition. If the attribute is empty the Schedule starts immediately after provisioning of the Job.	Schedule Definition Start Time
scheduleDefinitionEndTime	date-time	The end time of the Schedule Definition. If the attribute is empty the Schedule runs forever, not having a time constraint.	Schedule Definition End Time
recurringSchedule	RecurringSchedule	A recurring frequency to run a job within a timeframe defined by schedule definition, for example, every 5 minutes, 15 minutes, 1 hour, 1 day. If the attribute is empty, job runs non-stop.	Recurring Schedule
executionDuration	Duration	Total time for running one execution of a schedule. Depending on the reportingPeriod attribute, one execution of a schedule might produce multiple reports (e.g., when reporting period is 15 minutes and executionDuration is 1 hour, every execution of a schedule will produce 4 reports).	Execution Duration

7.2.4.16. Type ServiceFromToRef

Description: Reference to Service From and Service To endpoints.

Name	Type	Description	Mplify 133.1
serviceFrom*	object	A reference to a Service From endpoint resource.	Service Id From
serviceTo*	object	A reference to a Service To endpoint resource.	Service Id To
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not Present

7.2.4.17. Type ServiceRef

Description: Reference to a Service.

Name	Type	Description	Mplify 133.1
serviceHref	string	Hyperlink to the Service resource.	Not Present
serviceId*	string	Identifier of the Service.	Service Id
@type*	string	Used to unambiguously designate the class type when using `oneOf`	Not Present

7.2.4.18. Type ServiceSpecificConfiguration

Description: ServiceSpecificConfiguration is used as an extension point for Mplify-specific service performance monitoring configuration. The `@type` attribute is used as a discriminator.

Name	Type	Description	Mplify 133.1
@type*	string	Uniquely identifies the type of performance monitoring configuration that specifies PM objectives. In the case of Mplify services, this is the URN provided in the performance monitoring configuration specification. The named type must be a subclass of ServiceSpecificConfiguration.	Not present

7.2.4.19. Type ServiceSpecificResult

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

Name	Type	Description	Mplify 133.1
@type*	string	The name that uniquely identifies the type of performance monitoring results that are returned by the Performance Report. In the case of Mplify services, this is the URN provided in the performance monitoring results specification. The named type must be a subclass of ServiceSpecificResult.	Not present

7.2.4.20. Type TerminationError

Description: This indicates an error that caused a PM Job or PM Report to be terminated. The code and propertyPath should be used like in Error422.

Name	Type	Description
code	string	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: The requested entity will produce too much 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).
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).
value	string	Text to describe the reason of the termination.

7.2.4.21. Type TimeDuration

Description: This class is used to describe durations expressed as a 2-tuple, (value, units). The units from nanoseconds to years.

Name	Type	Description	Mplify 133.1
timeDurationValue*	integer	The value of the duration. For example, if the duration is 20 ms, this element is 20.	Duration Value
timeDurationUnits*	TimeDurationUnits	The unit of measure in the duration. For example, if an interval is 2ms, this element is MS.	Duration Unit

7.2.4.22. `enum` TimeDurationUnits

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

Value
NS

Value
US
MS
SEC
MIN
HOUR
DAY
WEEK
MONTH
YEAR

7.2.5. Notification Registration

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

7.2.5.1. Type EventSubscriptionInput

Description: This class is used to register for Notifications.

Name	Type	Description	Mplify 133.1
callback*	string	This callback value must be set to the *host* property from Performance Notification API (performanceNotification.api.yaml). This property is appended with the base path and notification resource path specified in that API to construct a URL to which notification is sent. E.g. for 'callback': "https://bus.mplify.com/listenerEndpoint", the performance job state change event notification will be sent to `https://bus.mplify.com/listenerEndpoint /mefApi/legato/performanceMonitoring/v5 /listener/performanceJobStateChangeEvent`	Notification Target Information
query	string	This attribute is used to define which type of events to register to. Example: 'query':'eventType = performanceReportStateChangeEvent'. To subscribe for more than one event type, put the values separated by a comma: `eventType=performanceReportStateChangeEvent, performanceJobCreateEvent`. The possible values are enumerated by Event type enums in performanceNotification.api.yaml. An empty query is treated as specifying no filters - ending in subscription for all event types.	List of Notification Types

7.2.5.2. Type EventSubscription

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

Name	Type	Description	Mplify 133.1
callback*	string	The value provided by the `EventSubscriptionInput` during notification registration.	Notification Target Information
id*	string	An identifier of this Event Subscription assigned when a resource is created.	Not present
query	string	The value provided by the `EventSubscriptionInput` during notification registration.	List of Notification Types

7.3. Notification API Data model

Figure 54 presents the Performance Monitoring Notification data model.

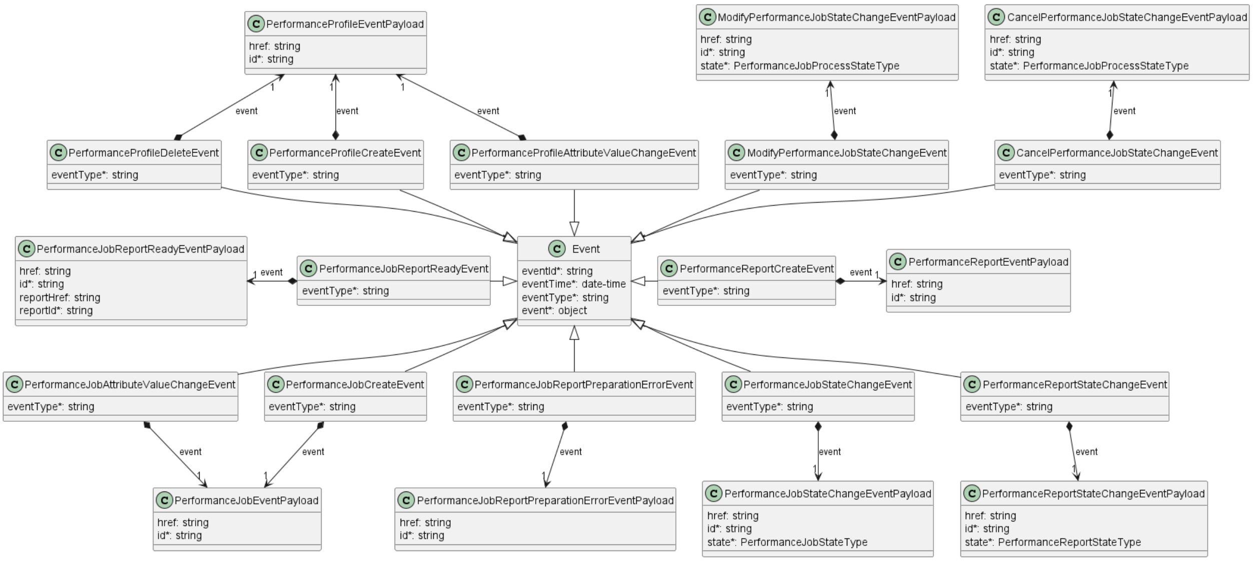

PerformanceProfileEventPayload
href: string
id*: string
ModifyPerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobProcessStateType
CancelPerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobProcessStateType
1
event
PerformanceProfileDeleteEvent
eventType*: string
PerformanceProfileCreateEvent
eventType*: string
PerformanceProfileAttributeValueChangeEvent
eventType*: string
ModifyPerformanceJobStateChangeEvent
eventType*: string
CancelPerformanceJobStateChangeEvent
eventType*: string
PerformanceJobReportReadyEventPayload
href: string
id*: string
reportHref: string
reportId*: string
PerformanceJobReportReadyEvent
eventType*: string
Event
eventId*: string
eventTime*: date-time
eventType*: string
event*: object
PerformanceReportCreateEvent
eventType*: string
PerformanceReportEventPayload
href: string
id*: string
PerformanceJobAttributeValueChangeEvent
eventType*: string
PerformanceJobCreateEvent
eventType*: string
PerformanceJobReportPreparationErrorEvent
eventType*: string
PerformanceJobStateChangeEvent
eventType*: string
PerformanceReportStateChangeEvent
eventType*: string
PerformanceJobEventPayload
href: string
id*: string
PerformanceJobReportPreparationErrorEventPayload
href: string
id*: string
PerformanceJobStateChangeEventPayload
href: string
id*: string
state*: PerformanceJobStateType
PerformanceReportStateChangeEventPayload
href: string
id*: string
state*: PerformanceReportStateType

Figure 54. 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 the information structure used for notification.

Name	Type	Description	Mplify 133.1
eventId*	string	Id of the event	PM Notification Identifier
eventTime*	date-time	Date-time when the event occurred	Event Time
eventType*	string	The type of the notification.	PM Notification Type
event*	object	The event linked to the involved resource object	Event

7.3.2. Type PerformanceProfileCreateEvent

Description: performanceProfileCreateEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Profile event.	PM Notification Type

Name	Type	Description	Mplify 133.1
event*	PerformanceProfile EventPayload	A reference to the Performance Monitoring Profile that is source of the notification.	Event

7.3.3. Type PerformanceProfileEventPayload

Description: A reference to the Performance Monitoring Profile that is the source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Profile.	Not present
id*	string	ID of the Performance Profile.	Performance Profile ID

7.3.4. Type PerformanceProfileAttributeValueChangeEvent

Description: performanceProfileCreateEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Profile event.	PM Notification Type
event*	PerformanceProfile EventPayload	A reference to the Performance Monitoring Profile that is source of the notification.	Event

7.3.5. Type PerformanceProfileDeleteEvent

Description: PerformanceProfileDeleteEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Profile event.	PM Notification Type
event*	PerformanceProfile EventPayload	A reference to the Performance Monitoring Profile that is source of the notification.	Event

7.3.6. Type PerformanceJobCreateEvent

Description: PerformanceJobCreateEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Job event.	PM Notification Type
event*	PerformanceJob EventPayload	A reference to the Performance Monitoring Job that is source of the notification.	Event

7.3.7. Type PerformanceJobEventPayload

Description: A reference to the Performance Monitoring Job that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Job.	Not present
id*	string	ID of the Performance Job.	PM Job ID

7.3.8. `enum` PerformanceJobProcessStateType

Description: The state of the process related to the Performance Job.

state	Mplify 133.1 name	Description
`acknowledged`	Acknowledged	The Cancel/Modify 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 Acknowledged until all validations as applicable are completed. If the attributes are validated, the request moves to the In-Progress state. If not all attributes are validated, the request moves to the Rejected state.
`completed`	Completed	The Cancel/Modify Performance Monitoring Job request has been completed by the Seller/Server.
`inProgress`	In-Progress	The Cancel/Modify Performance Monitoring Job request has been validated and accepted by the Seller/Server and is in-progress.
`rejected`	Rejected	The Cancel/Modify Performance Monitoring Job request has failed validation and has been declined by the Seller/Server.

7.3.9. Type PerformanceJobStateChangeEvent

Description: PerformanceJobStateChangeEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Job event.	PM Notification Type
event*	PerformanceJob StateChangeEventPayload	A reference to the Performance Monitoring Job that is source of the notification.	Event

7.3.10. Type PerformanceJobStateChangeEventPayload

Description: A reference to the Performance Monitoring Job that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Job.	Not present
id*	string	ID of the Performance Job.	PM Job ID
state*	PerformanceJob StateType	The state of the Performance Monitoring Job.	State

7.3.11. Type PerformanceJobAttributeValueChangeEvent

Description: PerformanceJobAttributeValueChangeEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Job event.	PM Notification Type
event*	PerformanceJob EventPayload	A reference to the Performance Monitoring Job that is source of the notification.	Event

7.3.12. Type CancelPerformanceJobStateChangeEvent

Description: CancelPerformanceJobStateChangeEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Cancel Performance Monitoring Job event.	PM Notification Type

Name	Type	Description	Mplify 133.1
event*	CancelPerformanceJob StateChangeEventPayload	A reference to the Cancel Performance Monitoring Job that is source of the notification.	Event

7.3.13. Type CancelPerformanceJobStateChangeEventPayload

Description: A reference to the Cancel Performance Monitoring Job that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Cancel Performance Monitoring Job.	Not present
id*	string	ID of the Cancel Performance Monitoring Job.	PM Job Cancellation Process Identifier
state*	PerformanceJob ProcessStateType	The state of the Cancel Performance Monitoring Job.	PM Job Cancellation Process State

7.3.14. Type ModifyPerformanceJobStateChangeEvent

Description: ModifyPerformanceJobStateChangeEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Modify Performance Monitoring Job event.	PM Notification Type
event*	ModifyPerformanceJob StateChangeEventPayload	A reference to the Modify Performance Monitoring Job that is source of the notification.	Event

7.3.15. Type ModifyPerformanceJobStateChangeEventPayload

Description: A reference to the Modify Performance Monitoring Job that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Modify Performance Monitoring Job.	Not present
id*	string	ID of the Modify Performance Monitoring Job.	PM Job Modification Process Identifier
state*	PerformanceJob ProcessStateType	The state of the Modify Performance Monitoring Job.	PM Job Modification Process State

7.3.16. Type PerformanceJobReportReadyEvent

Description: PerformanceJobReportReadyEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Job event.	PM Notification Type
event*	PerformanceJob ReportReadyEventPayload	A reference to the Performance Monitoring Job that is source of the notification.	Event

7.3.17. Type PerformanceJobReportReadyEventPayload

Description: A reference to the Performance Monitoring Job that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Job.	Not present
id*	string	ID of the Performance Job.	PM Job ID
reportHref	string	Hyperlink to access the Performance Report.	Not present
reportId*	string	ID of generated Performance Report.	Report ID

7.3.18. Type PerformanceJobReportPreparationErrorEvent

Description: PerformanceJobReportPreparationErrorEvent structure

Inherits from:

- Event

Name	Type	Description	
eventType*	string	Indicates the type of Performance Monitoring Job event.	PM Notification Type
event*	PerformanceJob ReportPreparationErrorEventPayload	A reference to the Performance Monitoring Job that is source of the notification.	Event

7.3.19. Type PerformanceJobReportPreparationErrorEventPayload

Description: A reference to the Performance Monitoring Job that is source of the notification.

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.

7.3.20. enum PerformanceJobStateType

Description: The state of the Performance Monitoring Job.

state	Mplify 133.1 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 Acknowledged 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. If scheduled, 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 was In-Progress, Suspended, or Scheduled is cancelled.
completed	Completed	A non-recurring Performance Monitoring Job finished execution.
inProgress	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. If 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 Pending 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.
pendingCancel	Pending Cancel	A Cancel Performance Monitoring Job request has been accepted by the Seller/Server. The Performance Monitoring Job remains Pending Cancel while resources used by the Job are being released. Once updates are complete, the Job moves to the Cancelled status.
rejected	Rejected	A create Performance Monitoring Job request fails validation and is rejected with error indications by the Seller/Server.

state	Mplify 133.1 name	Description
resourcesUnavailable	Resources 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 Scheduled until the start time is reached. The Performance Monitoring Job then moves to In-Progress. If the Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the Pending state.
suspended	Suspended	A Suspend Performance Monitoring Job request is accepted by the Seller/Server. The Job remains Suspended 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 the Cancel Performance Monitoring Job request is accepted, the Job moves to the Cancelled state. If the Modify Performance Monitoring Job request is accepted, the Job moves to the Pending state.

7.3.21. Type PerformanceReportCreateEvent

Description: PerformanceReportCreateEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Report event.	PM Notification Type
event*	PerformanceReportEventPayload	A reference to the Performance Monitoring Report that is source of the notification.	Event

7.3.22. Type PerformanceReportEventPayload

Description: A reference to the Performance Monitoring Report that is source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Report.	Not present
id*	string	IID of the Performance Report.	Report ID

7.3.23. Type PerformanceReportStateChangeEvent

Description: PerformanceReportStateChangeEvent structure

Inherits from:

- Event

Name	Type	Description	Mplify 133.1
eventType*	string	Indicates the type of Performance Monitoring Report event.	PM Notification Type
event*	PerformanceReport StateChangeEventPayload	A reference to the Performance Monitoring Report that is source of the notification.	Event

7.3.24. Type PerformanceReportStateChangeEventPayload

Description: A reference to the Performance Monitoring Report that is the source of the notification.

Name	Type	Description	Mplify 133.1
href	string	Hyperlink to access the Performance Report.	Not present
id*	string	ID of the Performance Report.	Report ID
state*	PerformanceReport StateType	Possible values for the state of a Performance Report.	State

7.3.25. `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 Acknowledged 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.
`rejected`	This state indicates that: - Invalid information is provided in Create Performance Report request - The request fails to meet validation rules for Performance Report delivery (processing).

8. References

- JSON Schema draft 7, JSON Schema: A Media Type for Describing JSON Documents and associated documents, by Austin Wright and Henry Andrews, March 2018. Copyright © 2018 IETF Trust and the persons identified as the document authors. All rights reserved.
- MEF 55.1, Lifecycle Service Orchestration (LSO): Reference Architecture and Framework, February 2021
- MEF 105, Performance Monitoring and Service Readiness Testing for SD-WAN, April 2024
- MEF 128.1, LSO API Security Profile, April 2024
- Mplify 133.1 Allegro, Interlude and Legato Fault Management and Alarm API BR&UC, October 2025
- Open API 3.0, February 2020
- Chapter 5: Representational State Transfer (REST) Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures (Ph.D.).
- RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, by S. Bradner, March 1997
- RFC 3986 Uniform Resource Identifier (URI): Generic Syntax, January 2005
- 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.
- TMF 628, Performance Management API REST Specification v5.0, June 2024
- TMF 630 TMF630 API Design Guidelines 4.2.0

Appendix A Acknowledgments

Mike **BENCHECK**

Michał **ŁĄCZYŃSKI**

Dominik **OGRODNIK**

Jack **PUGACZEWSKI**

Boris **TRINAJSTIC**