

East-Westbound Interface APIs
Version 7.0
17 July 2026

Security Classification: Non-Confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2026 GSM Association

Disclaimer

The GSMA makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSMA Antitrust Compliance Policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

1 Introduction 5
1.1 Overview 5
1.2 Scope 5
1.3 Definitions 5
1.4 Abbreviations 5
1.5 References 7
1.6 Conventions 8
2 Procedures over OP East/West Bound Interface 8
2.1 General 8
2.1.1 Federation 8
2.1.2 Directed Federation 8
2.1.3 Federation Context Identifier 8
2.1.4 Originating OP 9
2.1.5 Partner OP 9
2.1.6 Offered Availability Zones 9
2.1.7 Accepted Availability Zones 9
2.1.8 Mobile Country Codes 9
2.1.9 Mobile Network Codes 9
2.1.10 Availability Zone Meta-information 10
2.1.11 Edge Discovery Service 10
2.1.12 Mobility Strategy 10
2.1.13 Latency Constraints 10
2.1.14 Application Identifier 10
2.1.15 Artefact Identifier 11
2.1.16 Edge Node 11
2.1.17 QoS Performance Profiles 11
2.1.18 QoS Reference 11
2.1.19 Operator Platform Capabilties 11
2.1.20 Service Level Objectives 11
2.1.21 Service Level Indicators 11
2.1.22 Service APIs 11
2.1.23 Service API Federation 12
2.1.24 Notifications Management 12
2.1.25 Federation Timers 12
2.1.26 OP Policies 12
2.1.27 E/WBI APIs Transport Networks 12
2.2 Generic E/WBI Procedures 12
2.2.1 Procedures for Federation establishment between Partner OPs 13
2.2.2 Procedures for Availability Zone information synchronization 25
2.2.3 Federation Operator Identity Management 28
2.3 Application Services Procedures 30
2.3.1 Edge Service Procedures 30
2.3.2 Service API Usage on E/WBI 46

2.4 Resource Monitoring Procedures 54
2.4.1 Edge Resource Monitoring Procedures 55
2.4.2 Applications Resource Monitoring Procedures 57
2.5 Operational Visibility Procedures 58
2.5.1 Fault Management 58
2.5.2 Event Management 62
2.6 Notification Management 66
2.6.1 Network Events Notification Management Procedures 66
2.6.2 Applications Events Report Notification Procedures 70
2.7 Resource Control Policies Management Procedures 72
2.7.1 Application-level Policies Management Procedures 72
2.7.2 Operation-level Policies Management Procedures 74
3 OP East/West Bound APIs 75
3.1 Generic East/West Bound Service APIs 75
3.1.1 East/West Bound Interface Management - API 75
3.1.2 Availability Zone Information Synchronization Service – API 98
4 Application Service APIs 109
4.1 Edge Service APIs 109
4.1.1 Application Artefacts Management - APIs 109
4.1.2 Application Provider Resource Management - APIs 124
4.1.3 Application Onboarding Management - API 132
4.1.4 Application Instance Lifecycle Management - API 144
4.1.5 Edge Node Sharing - API 152
4.2 Service APIs Federation 155
4.2.1 Service APIs Forwarding Methods 155
5 Resource Monitoring APIs 163
5.1 Edge Resource Monitoring APIs 163
5.1.1 Availability Zone Resources Utilization – API 163
5.2 Applications Resource Monitoring APIs 171
5.2.1 Application Resource Utilization Management – API 171
6 Operational Visibility APIs 176
6.1 Fault Management APIs 176
6.1.1 Alarms Management – API 176
6.2 Event Management APIs 183
6.2.1 Events Management – API 183
7 Notification Management APIs 190
7.1 Network Events Notification context Management APIs 190
7.1.1 Network Events Management – API 191
7.2 Application-level Events Notification Management API 197
7.2.1 Application-level Events Management – API 197
8 Resource Control Policies Management API 208
8.1 Application-Level Policy Management 208
8.1.1 Application-Level Policy Methods 208
8.2 Operation-Level Policy Management 218

8.2.1 Operation-Level Policy Methods 218
9 Security 227
9.1 Authorization Service for E/WBI APIs 228
9.2 oAuth 2.0 Roles for E/WBI 228
9.3 Usage of signed JWT for oAuth2.0 228
9.4 Authentication and TLS Certificate Validations 228
9.5 Notifications and Callbacks Security 228
10 Usage of HTTP Versions on E/WBI 229
11 Federation Template 229
Annex A OpenAPI Specification Sample 231
Annex B E/WBI APIs Implementation 371
B.1 Kubernetes operator pattern 371
Annex C Document Management 374
C.1 Document History 374
C.2 Other Information 374

1 Introduction

1.1 Overview

This document specifies RESTful Application Programming Interface (APIs) that allow an Operator Platform (OP) to share the Edge Resources and capabilities securely to other Partner OPs over the East/West Bound Interface (E/WBI).

1.2 Scope

The present specification describes the APIs, sequence flows and the representation of the API and parameters in REpresentational State Transfer (REST) for the E/WBI between the two OPs. The E/WBI related stage 1 functional requirements are defined in the GSMA PRD OPG.02 [1].

1.3 Definitions

For the purposes of the present document, the term definitions given in OPG.02 [1] and the following apply.

Term	Description
API Consumer	An entity that makes an API request to, or within, the OP
Availability Zone	the lowest level of abstraction exposed to an Application Provider who wants to deploy an Application on Edge Cloud (see PRD OPG.11 [21])
Directed Federation	A Federation between two OP instances A and B, in which Edge Resources are shared by B to A, but not from A to B.
Edge Cloud	Refers to cloud-like capabilities located at the network edge including, from the Application Provider's perspective, access to elastically allocated compute, data storage and Network Resources as defined in the GSMA PRD OPG.11 [21]
Edge Discovery Service	OP service responsible to discover the most optimal edge for edge application instantiation
LCM Service	Lifecycle Management (LCM) Service to enable UCs for requesting dynamic application instantiation or termination
Originating OP	The OP instance initiating the Federation creation request to selected federated Operator Platforms/Operators. Both Leading OP and Home OP will be acting as Originating OP while creating the Federation with Partner OP.
Zone	Zone refers to an Availability Zone as defined in GSMA PRD OPG.11 [21]

1.4 Abbreviations

Term	Description
API	Application Programming Interface
CPU	Central Processing Unit
CRD	Custom Resource Definition
DNS	Domain Name System
DPDK	Data Plane Development Kit
E/WBI	East/West Bound Interface

Term	Description
FPGA	Field Programmable Gate Array
FQDN	Fully Qualified Domain Name
GPU	Graphical Processing Unit
GPRS	General Packet Radio Service
GRX	GPRS Roaming Exchange
HTTP	HyperText Transfer Protocol
IMSI	International Mobile Subscriber Identity
IPX	Internet Protocol Exchange
ISA	Instruction Set Architecture
ISV	Independent Software Vendor
KPI	Key Performance Indicator
LCM	LifeCycle Management
MCC	Mobile Country Code
MNC	Mobile Network Code
NBI	NorthBound Interface
NIC	Network Interface Card
OP	Operator Platform
OPG	Operator Platform Group
OS	Operating System
PLMN	Public Land Mobile Network
PRD	Permanent Reference Document
QoS	Quality of Service
RAM	Random Access Memory
REST	REpresentational State Transfer
SRIOV	Single Root Input Output Virtualisation
TLS	Transport Level Security
UC	User Client
UNI	User to Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
SDK	Software Development Kit
SLI	Service Level Indicator
SLO	Service Level Objective
TLS	Transport Layer Security
vCPU	Virtual CPU
VM	Virtual Machine
VPU	Visual Processing Unit
YAML	YAML Ain't Markup Language

1.5 References

Ref	Doc Number	Title
[1]	GSMA PRD OPG.02	Operator Platform Telco Edge Requirements", Version 12.0, 17 July 2026
[2]	RFC 2119	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt
[3]	Telco Edge Cloud	Telco Edge Cloud: Edge Service Description & Commercial Principles Whitepaper, version 1.0, 27 October 2020 https://www.gsma.com/futurenetworks/resources/telco-edge-cloud-october-2020-download/
[4]	RFC 6749	“The OAuth 2.0 Authorization Framework”, D. Hardt, Ed., October 2012. Available at http://www.ietf.org/rfc/rfc6749.txt
[5]	TMF642	Alarm Management API REST Specification, version 4.0.0, Available at https://github.com/tmforum-apis/TMF642_AlarmManagement
[6]	RFC 7515	JSON Web Signature (JWS), Available at https://datatracker.ietf.org/doc/html/rfc7515
[7]	RFC 7519	JSON Web Token (JWT), Available at https://datatracker.ietf.org/doc/html/rfc7519
[8]	RFC 9114	HTTP/3, Available at https://datatracker.ietf.org/doc/html/rfc9114
[9]	RFC 9000	QUIC: A UDP-Based Multiplexed and Secure Transport, Available at https://datatracker.ietf.org/doc/html/rfc9000
[10]	RFC7540	Hypertext Transfer Protocol Version 2 (HTTP/2), Available at, https://datatracker.ietf.org/doc/html/rfc7540
[11]	RFC 8446	The Transport Layer Security (TLS) Protocol Version 1.3 https://datatracker.ietf.org/doc/html/rfc8446
[12]	ITU-T X.509	Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, Available at, https://www.itu.int/rec/T-REC-X.509-201910-I/en
[13]	ITU-T X.520	Information technology – Open Systems Interconnection – The Directory: Selected attribute types, Available at, https://www.itu.int/rec/T-REC-X.520-201910-I/en
[14]	RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Available at, https://www.rfc-editor.org/rfc/rfc5280
[15]	RFC 6125	Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) Available at, https://datatracker.ietf.org/doc/html/rfc6125
[16]	GSMA IR.67	DNS Guidelines for Service Providers and GRX and IPX Providers Version 23.0 Available at, https://www.gsma.com/newsroom/wp-content/uploads//IR.67-v23.0-4.pdf

Ref	Doc Number	Title
[17]	Kubernetes	“Kubernetes.” Available at https://kubernetes.io/
[18]	Kubernetes custom resources	“Kubernetes custom resources.” Available at: https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/
[19]	NeoNephos Katalis GitHub	NeoNephos Katalis project. EWBI Kubernetes Operator GitHub repository. Available at https://github.com/neonephos-katalis/opg-ewbi-operator
[20]	NeoNephos Katalis GitHub	NeoNephos Katalis project. EWBI Kubernetes Operator GitHub repository. CRDs directory. Available at https://github.com/neonephos-katalis/opg-ewbi-operator/tree/main/config/crd/bases
[21]	GSMA PRD OPG.11	Operator Platform: Requirements for Edge Services

1.6 Conventions

The key words “must”, “must not”, “required”, “shall”, “shall not”, “should”, “should not”, “recommended”, “may”, and “optional” in this document are to be interpreted as described in RFC2119 [2].

2 Procedures over OP East/West Bound Interface

2.1 General

This section describes some of the key concepts and terms which applies to E/WBI procedures.

2.1.1 Federation

A Federation between two OPs conceptually refers an agreement to allow exposure of Edge Resources and Network capabilities by the other OP. The procedures which enable the establishment or creation of a Federation between the OPs are referred as E/WBI procedures. These procedures can be initiated by an OP towards the Partner OP using the set of APIs corresponding to the E/WBI.

2.1.2 Directed Federation

A Federation relationship in context of OPs is a directional relationship wherein a Federation creation request initiated by an OP to a Partner OP results in the Partner OP exposing their Edge Resources and network capabilities to the requesting OP. Thus, if two OPs want to expose Edge Resources and network capabilities with each other, then both the OPs would need to initiate a Directed Federation creation request towards each other.

2.1.3 Federation Context Identifier

A Federation context identifier is a dynamically generated identifier created by the OP which receives the Federation creation request from its Partner OPs. Based on the prior information if the OP accepts the Federation creation request, then the Federation context identifier is generated and returned to the requesting OP to represent the successful creation of the Federation.

This Federation identifier shall be included in all the subsequent E/WBI APIs invocations having operations associated to this Federation.

2.1.4 Originating OP

The creation of a Directed Federation from an OP to a Partner OP may be initiated by an administrative action by the OP administrator. Procedures like E/WBI interconnect management as defined in the GSMA PRD OPG.02 [1] are independent of any Application management procedures and any OP can independently initiate such requests towards the Partner OP.

The OP initiating the Federation creation request towards the Partner OP is defined as the Originating OP. GSMA PRD OPG.02 [1] defines the term “Leading OP” which can be interpreted as a role an OP instance is playing when it is serving Application Providers on the NorthBound Interface (NBI).

As described, the OP when initiating Federation creation request without any dependency to the NBI, requires an additional identification which in this document is termed as “Originating OP”.

2.1.5 Partner OP

The Partner OP, also defined as Operator Platform which offers exposure of its Edge Cloud and network capabilities to other Operator Platforms via E/WBI. In this document the E/WBI procedure considers that the Partner OP on receiving a Federation creation request from an Originating OP may validate, authenticate (requirements have been described in section 9), and authorize the request and the initiating OP's identity and accepts the Federation request by generating and sharing the Federation identifier with the Originating OP.

2.1.6 Offered Availability Zones

The Partner OP may offer to expose one or more Availability Zone(s) and associated Edge Resources to the Originating OP based on the prior agreement and local configuration. These Availability Zone(s) are defined as “Offered Availability Zones” wherein the Applications from Originating OP (also Leading OP here) can be orchestrated on requests from the Application Providers of the Originating OP.

2.1.7 Accepted Availability Zones

Based on the offered Availability Zone(s) from a Partner OP, the Originating OP may accept one or more Availability Zone(s) from the Partner OP and subscribe the accepted Availability Zone(s) over E/WBI to the Partner OP by initiating the Availability Zone subscription procedures.

2.1.8 Mobile Country Codes

Mobile Country Code (MCC) represents the serving country of the OP when it is shared in Federation establishment procedures. For any of the E/WBI APIs, the MCC associated to an OP shall have a single value and it is a non-modifiable parameter.

2.1.9 Mobile Network Codes

Mobile Network Code (MNC) represents the serving network code(s) of the OP when it is shared in Federation establishment procedures. For any of the E/WBI APIs, there can one or

more instances of MNC and its E/WBI procedures consider the MNCs to be a modifiable parameter.

MNCs are having a significant role for determination of the roaming Subscribers in visited networks and in conjunction with MCC they can be used by Home OPs to determine the roaming in Partner OPs footprints.

2.1.10 Availability Zone Meta-information

Availability Zone meta-information refers to the attributes associated to a group of Edge Resources which an OP can define as Availability Zone with a unique Availability Zone identifier and other locality information e.g., city, latitude/longitude, country, locality etc.

Availability Zone related meta-information can be shared by an OP with a Partner OP for various purposes e.g., in an Availability Zone offer during Federation create procedure, in Application onboarding requests to indicate intended Availability Zone(s) for Application deployment etc.

2.1.11 Edge Discovery Service

The Edge Discovery Service is defined as a HyperText Transfer Protocol (HTTP)-based API endpoint identified by a well-defined FQDN or IP-address, Port pair to assist UCs to discover adequate Edge Cloud in the current location of the End-Users. Every OP may host a publicly accessible discovery service which can be reached by the UCs over the UNI to enquire about the nearby application instance(s).

2.1.12 Mobility Strategy

An Application Provider may be able to provide the mobility strategy over the NBI for their Applications and it may additionally include the Application sensitivity to a UC’s mobility events.

The mobility strategy may cause an OP to take Application session relocation decisions based on the End-Users’ mobility events and taking into account the mobility strategy provided by the Application Provider.

2.1.13 Latency Constraints

The latency constraints refer to the limits on end-to-end latency between the UC and an edge Application which if exceeded may result in degradation of user experience or quality of experience as requested by the Application Provider. An OP may provide information about different latency profiles for the Availability Zone(s) to Application Providers and such information can be used to define the latency constraints for an Application on the NBI.

2.1.14 Application Identifier

While communicating with a Partner OP, the Leading OP uses Application Identifiers to refer uniquely to an Application from the Leading OP in the context of a Federation relationship with the Partner OP. The Application Identifier can be used to ensure uniqueness among the Applications, application instances, Application monitoring information etc.

2.1.15 Artefact Identifier

While communicating with a Partner OP, the Leading OP uses an artefact identifier to refer uniquely to an artefact from the Leading OP in the context of a Federation relationship with the Partner OP. The artefact identifier can be used to distinguish artefacts for all the Application Providers of the Leading OP on the E/WBI. Artefacts of an Application Provider can be reused by other Applications of the same Application Provider.

2.1.16 Edge Node

A resource in a physical data centre. The term Edge Node used in context with the Edge Node Sharing refers to the Edge Resources offered by the Partner OP to the Leading OP. The Leading OP may use such resources to serve its own Subscribers in scenarios such as not having the Edge Cloud footprint in locations where the End-Users requesting access to edge services, but a Partner OP is offering Edge Resources in those locations.

2.1.17 QoS Performance Profiles

Quality of Service (QoS) Performance Profile refers to a set of network characteristics e.g., end-to-end latency, packet loss, bandwidth etc. and the associated values between UCs and the edge Applications that a mobile network can provide.

2.1.18 QoS Reference

It is an identifier which refers to a pre-defined QoS Performance Profile configured in the mobile core network by the Operator and which can be requested by an Application function to request specified QoS for pplication sessions.

2.1.19 Operator Platform Capabilties

The set of Operator Platform Capabilities which an OP supports. These capabilities are optional and an can include functions e.g. support for home routing, Service API Federation, reporting of faults and/or events, Availability Zone resource consumption, network capability exposure for federated Applications etc. The E/WBI provides an OP with the ability to share the information about the Operator Platform capabilities which it can offer to the Partner OP.

2.1.20 Service Level Objectives

Service Level Objectives (SLOs) are specific measurable characteristics such as throughput, jitter, latency etc. SLOs provide a quantitative means to define the level of service a Leading OP can expect from the Partner OP.

2.1.21 Service Level Indicators

Service Level Indicators (SLIs) are the metrics used to measure the level of service provided against the SLOs as agreed between the OPs.

2.1.22 Service APIs

Service APIs in context of OP refers to set of REST APIs exposed by an OP on the NBI to expose mobile network capabilities in a secure and authorized manner to external Applications or enterprise customers of the OP.

2.1.23 Service API Federation

Service API Federation in context of OP refers to the process for forwarding Service API request from a Leading OP to the Partner OP who shall be executing the given service capability requested by the customers of the Leading OP via the Service APIs.

2.1.24 Notifications Management

Notifications management in context of the E/WBI refers to the capability which allows an OP to subscribe for receiving state change events or periodic reporting of aggregated SLIs etc for the API resources managed by the Partner OP.

2.1.25 Federation Timers

The Partner OP may share the renewal date and time and expiry date and time with the Originating OP to manage the Federation relationship identified with the Federation context identifier. The Originating OP shall start the renewal procedure starting at renewal date with the Partner OP. If the Originating OP does not renew the Federation before the expiry date, the Partner OP may terminate the federated Applications and free up the resources allocated to the Originating OP.

2.1.26 OP Policies

In context of the OP, a policy may refer to set of rules or conditions that may be applied by the OP to govern or control the Edge Resource assignment to the edge Applications. An OP NBI may expose Application-level policy to enable Application Providers to optionally select appropriate policies to control an Application’s behavior when deployed by the OP on Edge Resources.

2.1.27 E/WBI APIs Transport Networks

The E/WBI APIs can be exchanged between the federating OPs over public internet following the security requirements as mentioned in the current specification. Other potential alternatives e.g., the inter-PLMN IP backbone network (GRX/IPX) may also be considered as a viable future option among the federating Operators interconnected over GRX/IPX. However, the feasibility analysis for GRX/IPX for E/WBI usage is FFS in the current version of this document.

2.2 Generic E/WBI Procedures

These procedures generically cover the Federation interconnect and Availability Zone management functions to support Application deployment and lifecycle management capabilities across Partner OPs.

The E/WBI communications from security perspectives would require the OPs identification, authentication and authorisation which shall be in accordance with the mechanisms described in section 9 and the details of obtaining credentials are outside the scope of this document.

Note: This specification uses the term “HTTP POST” and “HTTP GET” as a generic reference to the action of using the POST or GET methods of HTTP. However, as specified in section9, the E/WBI APIs which may contain

sensitive information shall be sent over a secure connection and/or via HTTPS explicitly.

Note: In this specification any occurrence of HTTP keyword is used only to describe the behaviours of various E/WBI procedures using standard HTTP protocol defined methods. However, the actual E/WBI APIs must use the HTTPS for secure communication, as specified in section 9.4 .

2.2.1 Procedures for Federation establishment between Partner OPs

These procedures will provide key functionalities to establish Federation between two Partner OPs as described in the section 3.5.4.1 of the GSMA PRD OPG.02 [1].

Basic functionalities must cover:

- Create Federation between Partner OPs
- Update an already establish Federation between Partner OPs
- Remove a Federation establishment between Partner OPs.

2.2.1.1 Create Federation between Partner OPs

The Create Federation Operation is initiated by the Originating OP towards the Partner OP to establish a Directed Federation relationship between the two partners. By invoking this operation, the entity that originates the first message in the API sequences say ‘OP-A’ requests Partner OP-B permission to consume the OP-B resources and network capabilities on edge sites of 'OP-B'.

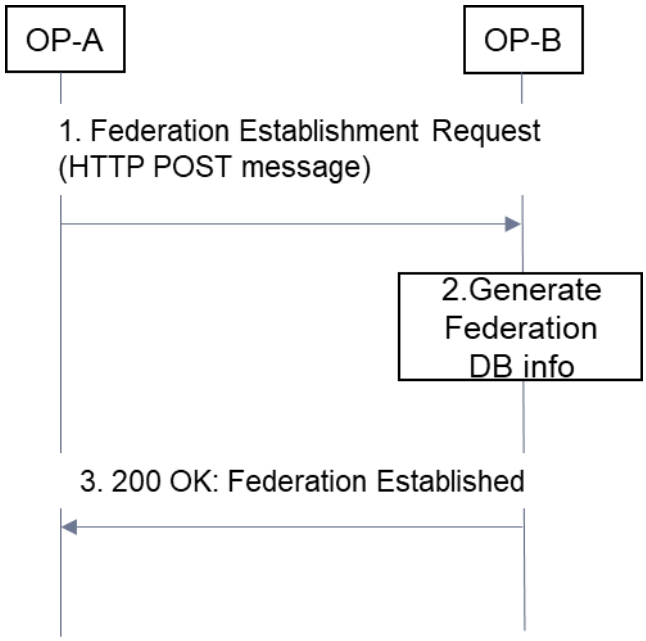


Figure 1: Create Federation

The message flow for creating a one-way (directed) Federation relationship is as follows:

1. A Federation create request (HTTP POST) is sent by the OP-A (Originating OP to the OP-B (Partner OP).

- The Originating OP provides all required identification, authentication, and authorisation information elements required to allow the Partner OP to decide if the request can be granted.
- The request may include also the Callback URLs to receive Federation status change notifications and resource monitoring information reports, time periodicity for receiving resource monitoring information reports from the Partner OP.

2. After authentication and authorization of OP-A, the Partner OP i.e., OP-B validates the Create Federation request from OP-A and stores the Federation information at OP-B.
3. The Partner OP sends a HTTP POST response to the Originating OP to inform about the result of the operation.

   - On success, a 200 OK message is sent along with a message body containing the Partner OP's Edge Discovery Service FQDN, list of Availability Zone(s) meta-information (e.g., Availability Zone Id, geolocation details etc.), Supported OP capabilities which may include Applications and network events reporting, faults notifications, Service APIs Federation, Application mobility, Application-level policies etc that the Partner OP can offer to the Originating OP

     The Partner OP can also include in the 200 OK response the Federation expiry time period, Federation guard time period (renew Federation) and default resource monitoring subscriptions created by the Partner OP.

   - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned. In this case the Originating OP shall remove any Federation context information created for handling the response from the Partner OP.
   - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

4. The Originating OP updates the response received from the Partner OP to local DB and may start timers for Federation guard and expiry periods.

Note: Two OPs in a Federation relationship are partners to each other, but in the context of this document, the Partner OP is referring to the OP responding to the Federation Establishment request from the Originating OP.

Note: The Service API capabilities that the Partner OP shares with the Originating OP are assumed to be available to the Originating OP for Service API invocation and it can forward the associated Service API requests to the Partner OP

2.2.1.2 Update Federation between Partner OPs – By Originating OP

To make an update of a Federation partnership the request initiator i.e., the Originating OP sends an HTTP PATCH message to Partner OP to update modifiable parameters which were earlier exchanged during the create Federation request flow (e.g., MNC, MCC or Edge Discovery Service Uniform Resource Locator (URL)).

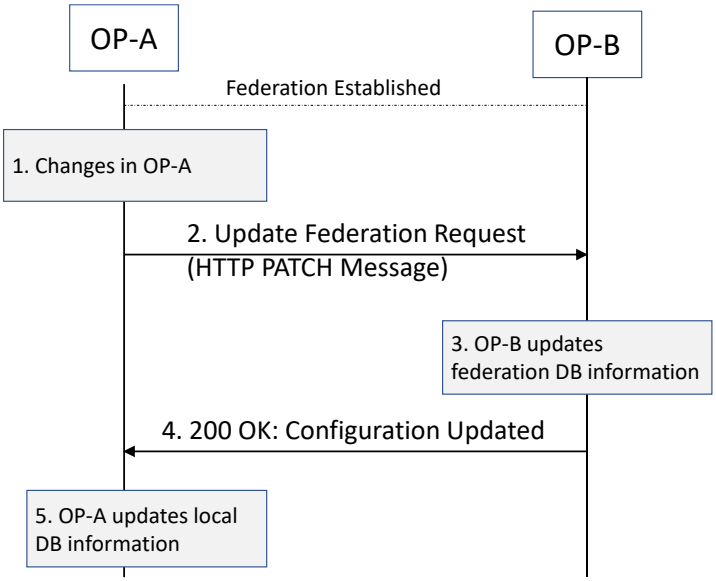


Figure 2: Update Federation

The message flow for updating a one-way (directed) Federation relationship is as follows:

1. The OP Administrator at OP-A updates parameters e.g., MNC etc. associated to the existing Federation between OP-A and OP-B.
2. An Update Federation request (HTTP PATCH) is sent by the Originating OP to the Partner OP.
    - The Originating OP provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
3. After authentication and authorization of OP-A, the Partner OP i.e., OP-B validates the Update Federation request from OP-A and updates the Federation information stored at OP-B.
4. The Partner OP sends a HTTP PATCH response to the Originating OP to inform about the result of the operation.
    - On success, a 200 OK message is sent to indicate that the Partner OP has updated the information as requested by the Originating OP for the existing Federation.
    - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned. In this case the Originating OP shall remove any Federation context information created for handling the response from the Partner OP.
    - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

Note: The Originating OP provides a Callback URL as part of the Create Federation request. The Partner OP shall use this Callback URL to share any updates on existing Federation relationship.

2.2.1.3 Update Federation between Partner OPs – By Partner OP

The Partner OP sends an HTTP POST request on the Callback URL of the Originating OP to update modifiable parameters which were earlier exchanged during the create Federation request flow e.g., MNCs, newly added Availability Zones, list of supported Service APIs etc.

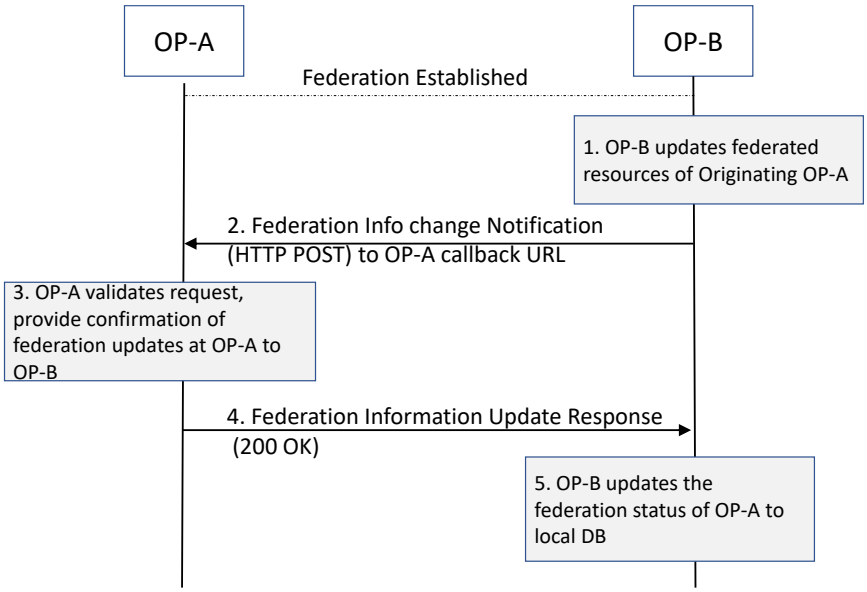


Figure 3: Update Federation by Partner OP

The message flow for updating a one-way (directed) Federation relationship by the Partner OP is as follows:

1. The OP administrator at OP-B updates parameters e.g., MNC, supported Service API capabilities etc. associated to the existing Federation between OP-B and OP-A.
2. An Update Federation request (HTTP POST) is sent by the Partner OP-B to the Originating OP-A Callback URL.
    - The Partner OP-B provides all required identification, authentication, and authorization information elements required to allow the Originating OP-A to decide if the request can be granted.
    - Message body contain parameters e.g., a list with the Service APIs name identifiers, UE public IP address ranges at the Partner OP-B etc.
3. After authentication and authorization of OP-B, the Originating OP-A validates the POST request from OP-B and updates the Federation information stored at OP-A.
4. The Originating OP-A sends a HTTP POST response to the Partner OP-B to inform about the result of the operation.
    - On success, a 200 OK message is sent to indicate that the Originating OP-A has updated the information as requested by the Partner OP-B for the existing Federation.
    - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned.

- The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Originating OP-A and should be retried at a later point of time.

2.2.1.4 Remove Federation between Partner OPs

This procedure is intended to remove existing Federation information within a Partner OP. By Remove Federation Operation, the OP of the entity that originates the first message in the API sequences say 'OP-A' sends an HTTP DELETE request to the Partner OP say 'OP-B' to terminate the existing Federation configuration from OP-A to OP-B (earlier created by OP-A via create Federation procedures).

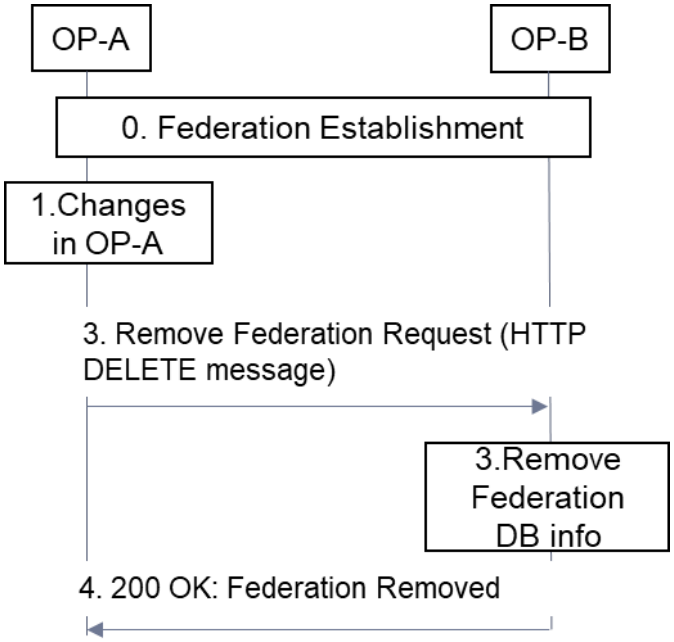


Figure 4: Remove Federation

The Partner OP can also terminate the existing Federation with the Originating OP for reasons e.g., administrative and business decisions, expiry of the Federation lifetime timer etc. The Partner OP say 'OP-B' sends an HTTP POST request to the Originating OP say 'OP-A' to terminate the existing Directed Federation earlier created on request from the Originating OP "OP-A". The Partner OP uses the HTTP notification destination provided by the Originating OP as part of the Create Federation Operation API where the Originating OP shall be receiving any HTTP notifications from the Partner OP.

2.2.1.5 Retrieve partner federated Availability Zone meta-information

The Originating OP may use this procedure towards federated Partner OPs to retrieve Availability Zone(s) meta-information e.g., Availability Zone identifier(s), Availability Zone(s) location etc. which the Partner OP may offer to the Originating OP. This operation can be invoked on existing Federation between the two OPs.

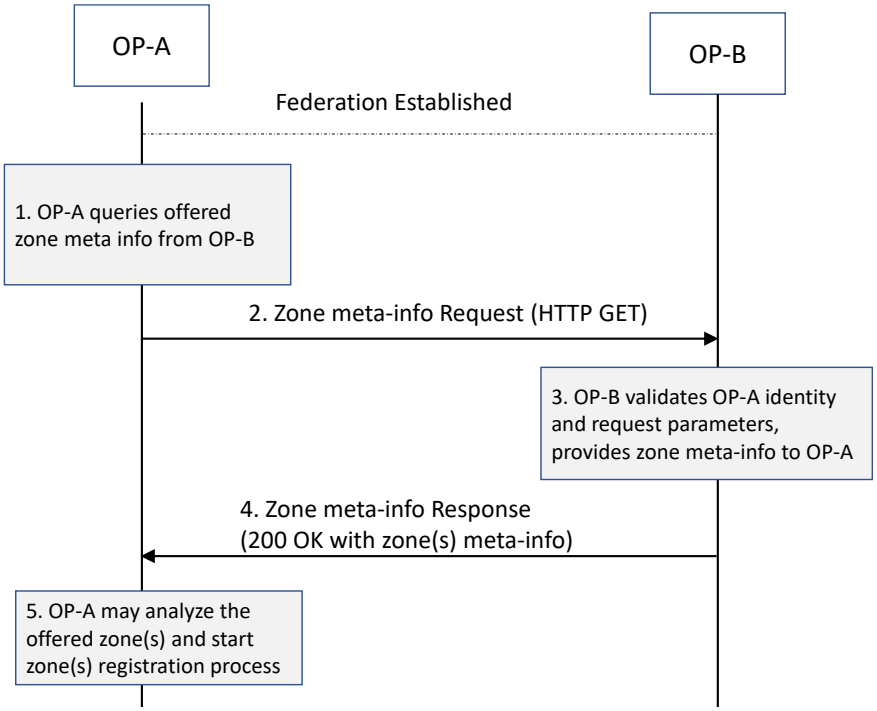


Figure 5: Retrieve partner federated Availability Zone meta-information

The message flow for retrieving the Partner OP Availability Zone(s) meta-information by the Originating OP on an existing Federation relationship is as follows:

1. A partner federated Availability Zone meta-information get request (HTTP GET) is sent by the Originating OP to the Partner OP.
    - The Originating OP provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
2. The Partner OP sends a HTTP GET response to the Originating OP to inform about the result of the operation.
    - On success, a 200 OK message is sent along with a message body containing a list of Availability Zones and their geolocation details that the Partner OP has available to share with the Operator.
    - On failure, an appropriate error codes (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

2.2.1.6 Retrieve Partner OP Capabilities

The Originating OP should be able to query the Partner OP using the GET method to retrieve the details of the Operator Platform capabilities which the Partner OP can offer to the Originating OP. The Partner OP shares the names of the offered Operator Platform Capabilities to the Originating OP at the time of Federation establishment phase.

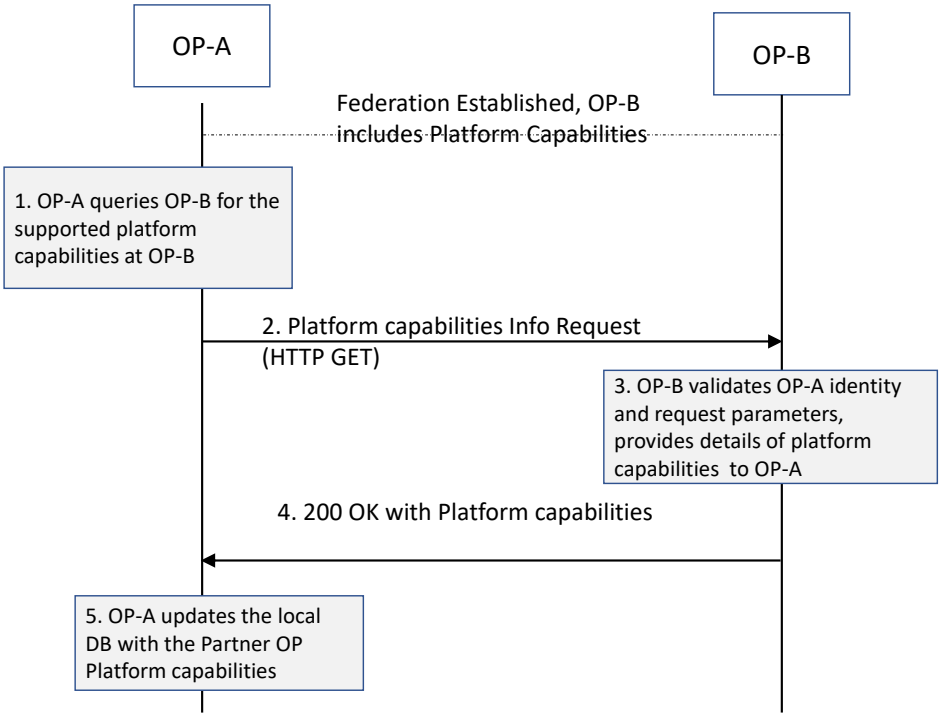


Figure 6: Retrieve Partner OP Capabilities

The message flow for retrieving the Operator Platform Capabilities supported by the Partner OP on an existing Federation relationship is as follows:

1. The Originating OP may decide to retrieve supported Operator Platform Capabilities by the federated Partner OP.
2. A HTTP GET request with query parameter is sent by the Originating OP-A to the federated Partner OP-B to retrieve the supported Operator Platform Capabilities information.

    - The Originating OP-A provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
    - The Originating OP-A includes the query parameter in the HTTP GET request to indicate the specific capability that the Partner OP supports

3. The Partner OP-B validates the Originating OP-A request based on the identification and authorization information provided and prepares the response containing the information about the supported Service APIs.
4. The Partner OP sends a HTTP GET response to the Originating OP to inform about the result of the operation.

    a) On success, a 200 OK message is sent along with a message body containing a list with the Operator Platform Capabilities as indicated by the query parameter in HTTP GET request which may include e.g., Application events information reporting, network events reporting, faults reporting, Service APIs name identifiers, UE public IP address ranges at the Partner OP-B, network capability exposure information etc

b) On failure, an appropriate error codes (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
c) The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

5. The Originating OP updates the local DB with the supported Operator Platform Capabilities by the corresponding Partner OP.

2.2.1.7 Periodic Health Check

The Originating OP may use this procedure to check the health status of the communication link with the Partner OP over E/WBI once the successful Federation has been established between the two OPs. The Originating OP can initiate the Health Check with HTTP GET method if it detects a long inactivity period over E/WBI towards the Partner OP.

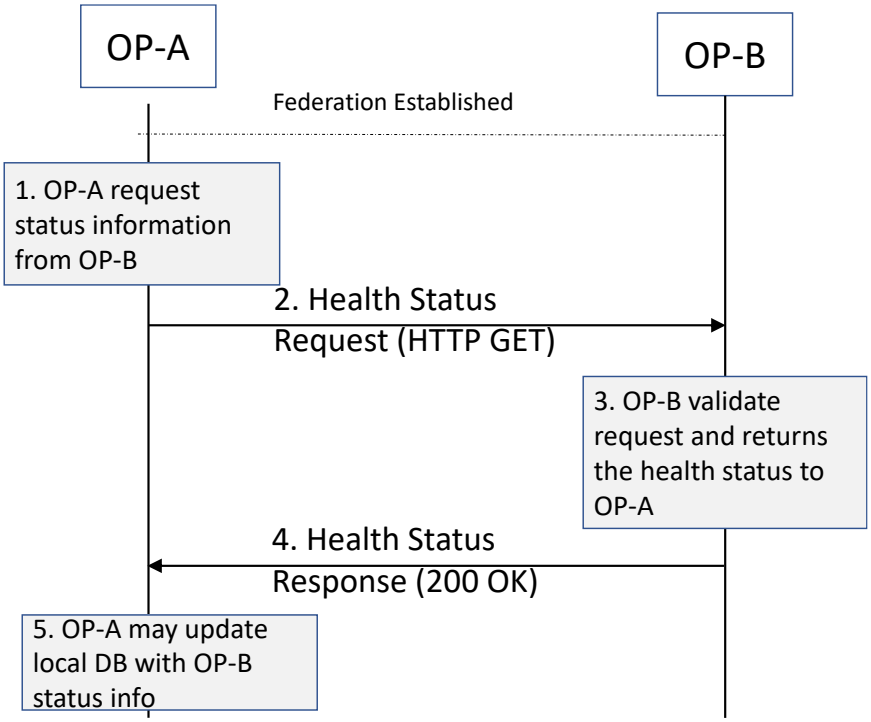


Figure 7: Retrieve Partner OP Health Status Information

The message flow initiated by the Originating OP for retrieving the health status of existing Federation with the Partner OP:

1. The Originating OP may decide to retrieve the health status information for an existing Federation relationship with the Partner OP.
2. A HTTP GET request is sent by the Originating OP-A to the federated Partner OP-B including the Federation context identifier.

    - The Originating OP-A provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.

3. Partner OP-B validates the Originating OP-A request based on the identification and authorization information provided and prepares the response containing the health status of the Federation at OP-B.
4. The Partner OP sends a HTTP GET response to the Originating OP to inform about the result of the operation.

    a) On success, a 200 OK message is sent along with a message body containing the health status of the Federation at OP-B.
    b) On failure, an appropriate error codes (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    c) The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

The Originating OP updates the local DB with the Federation health status corresponding to the Partner OP i.e., OP-B.

Note: The procedures for handling failed health checks are for further study.

Note: The Partner OP may use the E/WBI fault management procedures to inform of the situation to the Originating OP.

2.2.1.8 Retrieve Federation context information

The Originating OP can use this procedure to retrieve the Federation context identifier which has been already created by the Partner OP for the Federation create request initiated by the Originating OP earlier.

The Originating OP uses the HTTP GET method to request the Partner OP to share the Federation context identifier by including the authorization token in the request.

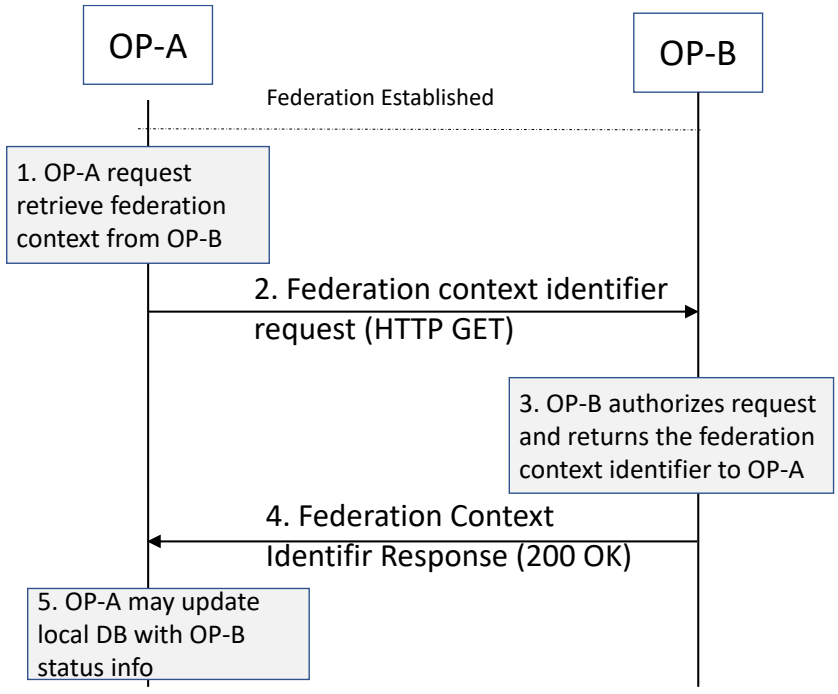


Figure 8: Retrieve Federation Context Identifier Information

The message flow initiated by the Originating OP for retrieving the Federation context identifier earlier generated by the Partner OP for the Originating OP:

1. The Originating OP may decide to retrieve the Federation context identifier for an existing Federation relationship at the Partner OP.
2. A HTTP GET request is sent by the Originating OP-A to the federated Partner OP-B including the authorization token to retrieve the Federation context identifier information.

    - The Originating OP-A provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.

3. The Partner OP-B validates the Originating OP-A request based on the identification and authorization information provided and prepares the response containing the Federation context identifier of OP-A.
4. The Partner OP sends a HTTP GET response to the Originating OP to inform about the result of the operation.

    a) On success, a 200 OK message is sent along with a message body containing the Federation context identifier at the Partner OP-B etc
    b) On failure, an appropriate error codes (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    c) The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

The Originating OP updates the local DB with the Federation context identifier corresponding to the Partner OP i.e., OP-B.

2.2.1.9 Renew Federation relationship

The Originating OP can use this procedure to renew the existing Federation context which has been already created by the Partner OP. The renewal request shall be initiated by the Originating OP before the guard timer expiry period that the Partner OP shared during the Federation create procedure.

The Originating OP uses the HTTP POST method to request the Partner OP to renew the Federation by including the Federation context identifier and renewed authorization token in the request.

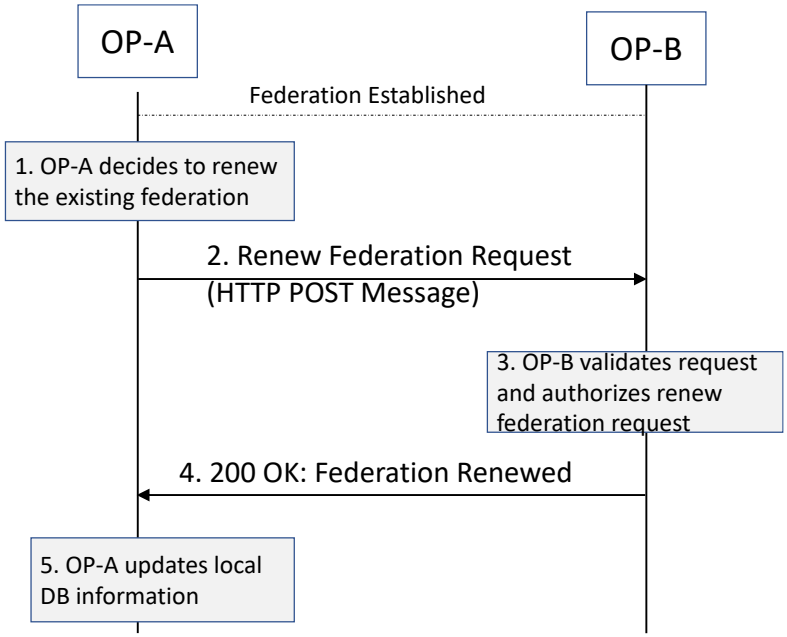


Figure 9: Renewal of Federation Relationship

The message flow initiated by the Originating OP for renewing the existing Federation relationship between the Partner OP and the Originating OP:

1. The Originating OP may decide to renew the existing Federation relationship with the Partner OP.
2. A HTTP POST request is sent by the Originating OP-A to the federated Partner OP-B including the authorization token and the Federation context identifier.
    - The Originating OP-A provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
3. The Partner OP-B validates the Originating OP-A request based on the identification and authorization information provided and prepares the response containing the Federation context identifier of OP-A.
4. The Partner OP sends a HTTP POST response to the Originating OP to inform about the result of the operation.
    a) On success, a 200 OK message is sent along with a message body containing the Federation context identifier at the Partner OP-B
    b) On failure, an appropriate error codes (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    c) The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

The Originating OP updates the local DB with the Federation context identifier corresponding to the Partner OP i.e., OP-B.

2.2.1.10 API handling for terminated Federation

A Partner OP may decide to terminate any existing Federation relationship with any Originating OP. In such a situation the Partner OP shall inform the Originating OP about the termination status of the Federation relationship in response to any E/WBI API request initiated by the Originating OP. The Originating OP, based on the error response from the Partner OP can determine if it should retry any pending requests at a later point of time or not.

The Originating OP on receiving the error response with an indication of a terminated Federation relationship, should not try to renew or reuse the existing Federation context identifier provided earlier by the Partner OP.

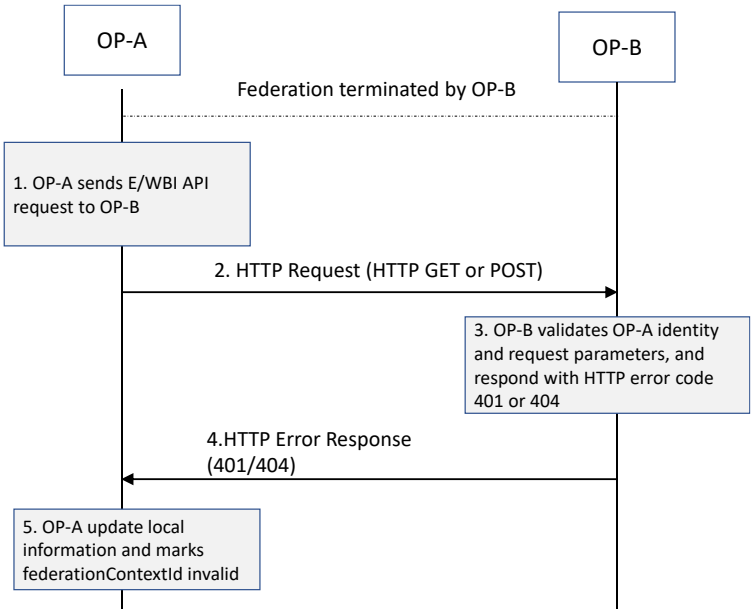


Figure 10: Request handling for Terminated Federation

The behaviour for E/WBI requests by the Originating OP on a terminated Federation relationship identified by the Federation context identifier is described below:

1. The Originating OP may decide to initiate a HTTP request assuming a valid Federation relationship with the Partner OP.
2. A HTTP request is sent by the Originating OP-A to the federated Partner OP-B including the authorization token and the Federation context identifier.
    - The Originating OP-A provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
3. The Partner OP-B validates the Originating OP-A request.
4. The Partner OP sends a HTTP response to the Originating OP to inform about the Federation termination status.
    a) An appropriate error codes (e.g., 401, 404 etc.) along with an Application-level error message shall be returned.

b) The response may also include an indication about the freeing up of resources that the Partner OP allocated to the Originating OP

Note: As an implementation choice, the Partner OP can decide to remove the Federation context identifier at any point of time on the termination of a Federation by administrative actions.

It is also recommended that a new Federation context identifier shall be used if the Federation with the same Originating OP is created in future.

2.2.2 Procedures for Availability Zone information synchronization

As described in the section 3.5.4.2 of the GSMA PRD OPG.02 [1] these procedures will enable sharing of pre-provisioned Availability Zone information and updating the resource information, notifying partners if there are new Availability Zones available etc.

When the Partner OP accepts the create Federation request from the Originating OP, the Partner OP also provides the Availability Zones meta information which it can offer to the Originating OP and their Application Providers.

This request can be sent only after a successful creation of the Federation relationship between the Originating OP and the Partner OP. The entity that originates the first message in the API sequences subscribes one or more Availability Zones from the list of Availability Zones being offered by the Partner OP to the Originating OP. The ISVs of the Originating OP can deploy their Applications on the Availability Zone(s) being subscribed by this API.

2.2.2.1 Availability Zone Subscription

On receiving the Availability Zones meta information (e.g., Availability Zone id, serving location etc.), the Originating OP may send an HTTP POST message that contain the accepted Availability Zone(s) subscription request for one or more Availability Zones offered by the Partner OP.

The Partner OP may reserve the resources for the Originating OP on indicated Availability Zone(s) and provides the details of resources configuration, QoS Performance Profiles, supported network capabilities with SLOs etc in the Availability Zones information to the Originating OP in the response to Availability Zone subscription request.

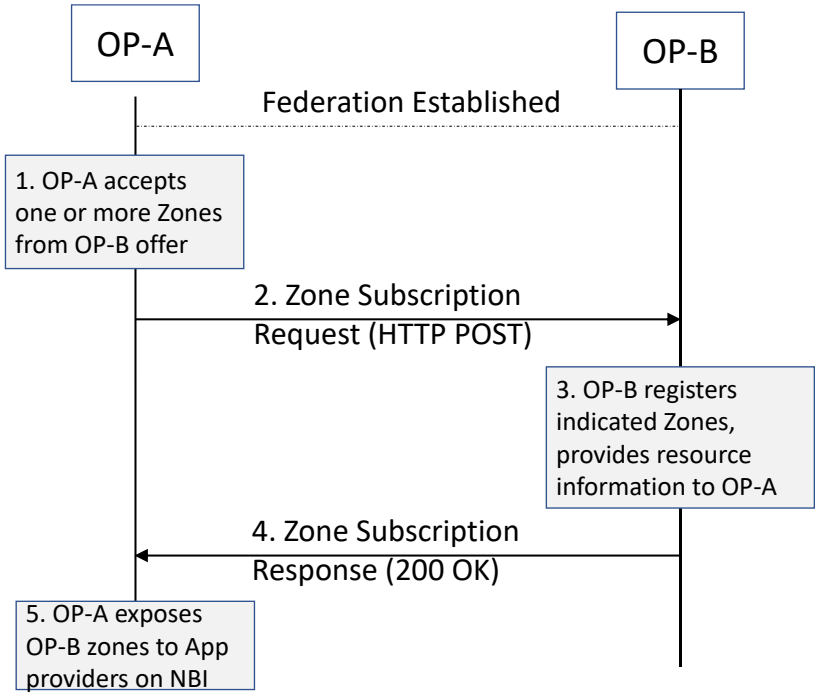


Figure 11: Availability Zone subscription

2.2.2.2 View Availability Zone information

The Originating OP at any moment can query the Partner OP for the Availability Zone(s) status information (e.g., resource availability, serving location etc.). The Originating OP may send an HTTP GET request that contain the Availability Zone(s) identifier for one or more Availability Zones offered by the Partner OP.

The HTTP GET method supports the query parameters to allow the Originating OP to retrieve specific information in context of an existing Federation identified by the Federation context identifier.

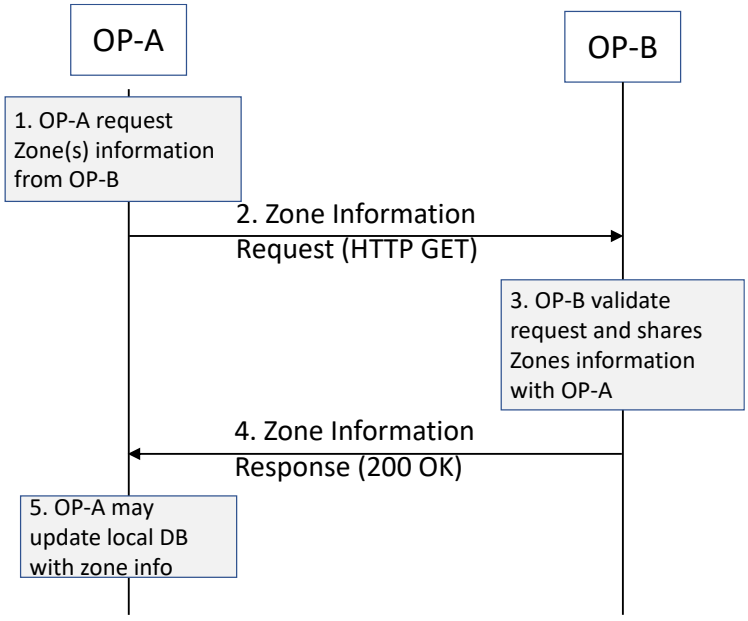


Figure 12: View Availability Zone information

2.2.2.3 Update Availability Zone Information By Partner OP

A Partner OP shall update the Originating OP of any changes to the Edge Resources or network capabilities subscribed by the Leading OP. For this purpose, the Leading OP provides a Callback URL to the Partner OP while sending the Availability Zone subscription request. The Partner OP can use the Callback URL to provide any changes to earlier resource subscription e.g., additional or deletion of Edge Resources, new compute flavours availability etc.

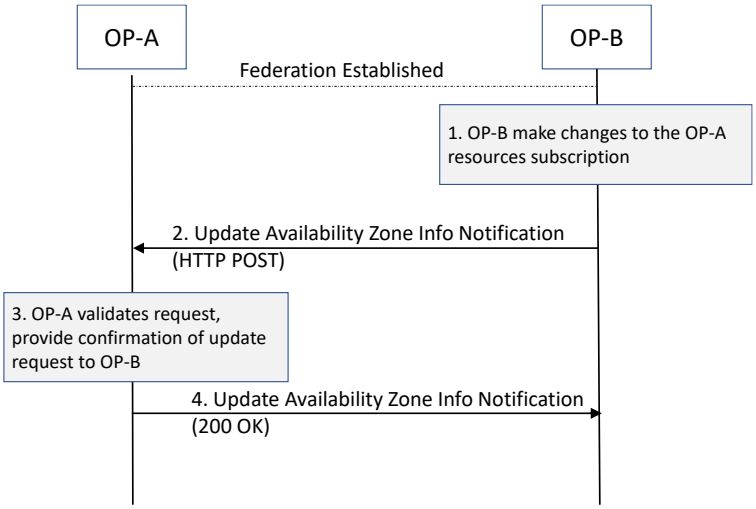


Figure 13: Update Availability Zones Information

The message flow for updating Availability Zone information is as follows:

1. The administrator at OP-B performs updates to resource configuration associated to OP-A e.g., add GPU resources in an Availability Zone shared with OP-A.
2. An Update Availability Zone request (HTTP POST) is sent by the Partner OP to the Leading OP.
    - The Partner OP provides all required identification, authentication, and authorization information elements required to allow the Leading OP to decide if the request can be granted.
3. After authentication and authorization of OP-B, the Leading OP i.e., OP-A validates the Update Availability Zone request from OP-B and updates the given information stored at OP-A.
4. The Leading OP sends a HTTP POST response to the Partner OP i.e., OP-B to inform about the result of the operation.
    - On success, a 200 OK message is sent to indicate that the Leading OP has updated the information as requested by the Partner OP for the existing Federation.
    - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Leading OP and should be retried at a later point of time.

2.2.2.4 Update Availability Zone Information By Originating OP

The Originating OP can update the Availability Zone subscription information for already subscribed Availability Zones e.g., assigning infrastructure policies to the Edge Resources offered by the Partner OP.

The message flow for updating Availability Zone information is as follows:

1. The OP administrator at OP-A performs updates to Availability Zone subscription information associated to already subscribed Availability Zones from OP-B e.g., attaching Edge Resource policies supported and shared by OP-B.
2. An Update Availability Zone request (HTTP PATCH) is sent by the Leading OP to the Partner OP.
    - The Leading OP provides all required identification, authentication, and authorization information elements required to allow the Partner OP to decide if the request can be granted.
3. After authentication and authorization of OP-A, the Partner OP i.e., OP-B validates the Update Availability Zone request from OP-A and updates the given information stored at OP-B.
4. The Partner OP sends a HTTP PATCH response to the Leading OP i.e., OP-A to inform about the result of the operation.
    - On success, a 200 OK message is sent to indicate that the Partner OP has updated the information as requested by the Leading OP for the existing Federation.
    - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.

2.2.3 Federation Operator Identity Management

A Federation member must have some type of identity credential whereby their actions within the Federation can be authorized by the resource provider OP or the Partner OP.

For any communication between the federating OPs, they must be able to determine each other’s identity to ensure that the information between them can be shared in a secure and reliable manner. In the current model, the Partner OP assigns the authorization token to the Originating OP which then uses it to authorize with the Partner OP.

For the authorisation purpose, the Partner OP manages the authorisation credentials for the OPs requesting for the Edge and API resources. The Partner OP will act as the central authorization service manager for those OPs that request for the resources and provides the authorization credentials to each of the federating partners to ascertain their identity as seen by the Partner OP.

However, to determine the identity of the Operator, it is recommended that the digital certificate used for mutual TLS on E/WBI can be used as the identity of the Operator in a given Federation instance between the two OP.

The identity determination from the digital certificate (X.509 certificate) may rely on following key attributes from a valid certificate (validation rules defined as per ITU-T X.509 [12], X.520[13]):

- CN (“cn”, “commonName”) - It is a name by which the object is commonly known in some limited scope (such as an organization)
    - The federating partner Operators may agree to use unique FQDNs for the commonName field as an indicative value of the corresponding E/WBI API server host name
    - A representative example could be CN= <operator-domain.com> where the “operator-domain.com” refers to the well-known domain name or common name of a given Operator
- subject (Subject Name) - Identifies the entity associated with the public key stored in the subject public key field of the certificate. Where it is non-empty, the subject field shall contain an X.500 distinguished name (DN).)
    - The Partner OPs may use standard Distinguished Name (DN) attributes like C (Country), O (Organization - e.g., Telecom Operator Name), OU (Organizational Unit - e.g., Platform Services)
    - A representative example could be C=US, O=Example Teleops Inc., OU=Platform Operations, CN= example-teleops.com
- subjectAltName (Subject Alternative Name) - The subject name may be carried in the subject field and/or the subjectAltName extension
    - The domain name (e.g., example-teleops.com, ewbi-api.operator-domain.com etc.) may be used in the Subject Alternative Name (SAN) extension (DNS Name type)

Taking guidance from RFC 6125 [15] which recommends and prefers the subjectAltName over the subject, the subjectAltName is recommended as the first preference for determining the identity of the federating Operator followed by subject and CN in order.

The federating Operators may agree to use the specific identity attribute from X.509 certificate as per the recommendations given above via offline communication outside the scope of this document.

It is also recommended that before allowing access to E/WBI API resources, the federating Operators should check and verify the partner Operator identity using identity determination fields from the X.509 certificate as recommended above.

The recommendation on X.509 CAs and certificate validation rules are described in section 9.4.

Note: Any recommendations for the FQDN fields values or format in relation to commonName, subject Name, subjectAltName etc in X.509 certificates which may use telco specific network identifiers e.g., MCC, MNC etc. for determination of Federation Operator identity is FFS in the current version of this document.

Note: As specified in GSMA IR.67 [16], an Operator may have delegated ownership of sub-domains of ".3gppnetwork.org" e.g., ".mnc<MNC>.mcc<MCC>.3gppnetwork.org" where MNC and MCC are owned by the Operator. However, a new service identity "op-ewbi" for E/WBI as in "op-ewbi.mnc<MNC>.mcc<MCC>.3gppnetwork.org" needs to be added to GSMA IR.67 [16]. The same applies to sub-domains of "pub.3gppnetwork.org" for E/WBI usage purpose e.g., "op-ewbi.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org" when used on internet DNS. The feasibility analysis for adding new service identity in GSMA IR.67 [16] is FFS in the current version of this document.

Note: The RFC 5280 [14] provide different options for representing the identities that can be bound to the subject other than the DNS name and federating Operators may mutually agree to consider those options too.

2.3 Application Services Procedures

The OP services as defined in GSMA PRD OPG.02 [1] can be provided to the Application Providers via the NBI to manage, deploy and monitor Applications with the Leading OP and its federated Partner OPs.

This section provides the coverage to the edge services which requires E/WBI APIs to enable the OP services with those federated Partner OPs based on the NBI operations invoked by the Application Providers.

2.3.1 Edge Service Procedures

The following section describes the OP supported edge service procedures over E/WBI to provide Application Providers access to federated Partner OP's services.

2.3.1.1 Procedures for Application Artefacts Management Service

According to section 3.5.4.3 of the GSMA PRD OPG.02 [1] an OP shall be capable of onboarding and managing Application artefacts towards a Partner OP, considering that a Federation has been established between Partner OPs previously (see section 2.2).

The following procedures need to be supported:

- Transfer Application images (container or Virtual Machines (VMs) per section 3.6 and 3.7 of the GSMA PRD OPG.02 [1])

2.3.1.1.1 Application Artefacts Upload

This is intended for an OP to upload Application images e.g., Docker container image file(s) and associated Application component descriptors i.e., artefacts such as Helm charts, Terraform scripts etc. to a Partner OP.

The same artefact(s) can be reused by multiple Applications within an Application Provider account. An Application Provider specifies the Partner OPs that an Application should be deployed to. As an artefact can be associated to one or more Applications, they are delivered to the set of Partner OPs that are associated with the Applications.

For this operation message flows should be as follows:

1. An artefact upload request i.e., HTTP POST message with the Application artefacts provided by the Application Provider over the NBI from the Originating OP is sent to a Partner OP.
2. The Partner OP authenticates the Originating OP and validates the requested operation and the parameters e.g., country code, Federation keys and the indicated Availability Zone(s) status.
3. Once the artefact push is finished

    a) If the procedure is completed successfully, a response message HTTP POST response with “202: Artefact Accepted” shall be sent from the Partner OP. (onboarded artefact information can be included as well).
    b) The Originating OP may send GET request at a later point of time to enquire about the actual upload status with Partner OP. The Partner OP may return a successful response to HTTP GET operation with status code “200 OK” containing the onboarded artefact information.

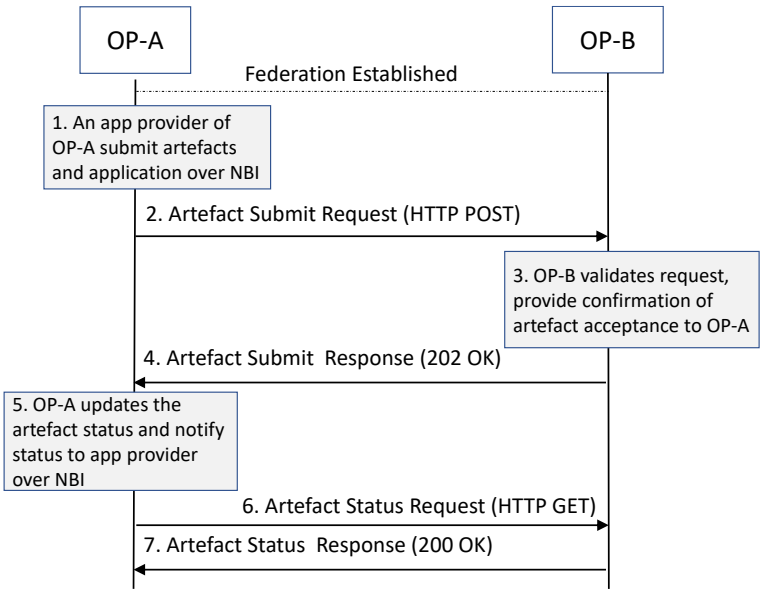


Figure 14: Artefact upload request

2.3.1.1.2 Application Artefacts Update

This is used by an OP to update the already submitted artefacts e.g., Docker container image file(s) and scripts to a Partner OP. Artefacts are onboarded by the Leading OP and stored in local repositories. There artefacts are linked to the Applications by the Application Provider and are delivered to a Partner OP based on the Application Availability Zones indicated by the Application Provider.

The updated information e.g., Application images, helm charts etc. shall be used by the Partner OP when requested by the Application Provider for Applications deployed in the Partner OP footprint. It is to be noted that the already running application instances using the artefact are not affected by this operation.

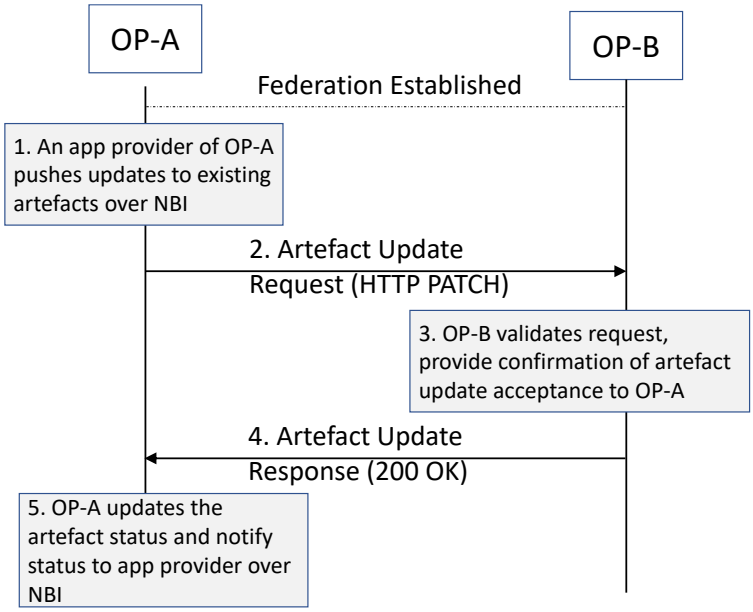


Figure 15: Artefact update request

2.3.1.1.3 Application Artefacts Delete

An OP must support to delete the already submitted artefacts e.g., Docker container image file(s), Application components descriptor scripts e.g., Helm charts etc. to a Partner OP.

An OP may initiate the Application artefact deletion process on receiving the request from Application Provider over the NBI.

The Partner OP shall remove the artefacts e.g., Application images, helm charts etc. using the artefact information present in the HTTP DELETE request.

Note: Aspects like audits of complete removal of artefacts in the Partner OP environment is beyond the scope of this document and are not covered here.

2.3.1.2 Application Provider Resource Management Service

According to section 5.2 of the Telco Edge Cloud whitepaper [3], the capacity reservation model is described as using a preselected combination of service units (computing, storage, and networking) that is permanently allocated to the Customer. Usually chosen for longer time periods in which the Customer has a permanent demand to attend.

Following procedures needs to be supported:

- To reserve Edge Resources with Partner OP based on the request from Application Provider on the NBI
- To update or modify the already reserved resource pool e.g., to add or remove resources in existing reservation
- Delete already reserved resource pool created for an Application Provider

2.3.1.2.1 Resource Reservation

This is intended for an OP to reserve resources for an Application Provider e.g., Edge Resource flavours when the Application Provider initiates the reservation using NBI.

The Application Provider shall be able to request reservation of resources with a Partner OP on per Availability Zone basis. The Partner OP shall be able to reserve resources for a given Application Provider from the allocated quota for the Leading OP. Once the resource reservation request is approved by the Partner OP, a resource pool identifier is provided to the Leading OP to refer to the specific resource pool for the Application Provider. The Application Provider can use the identifier while instantiating the Application to indicate from which resource pool resources are to be used when deploying Applications in the Partner OP Availability Zones.

For this operation message flows should be as follows:

1. A resource reservation request i.e., HTTP POST message describing the resources to be reserved along with the Availability Zone where they should be located as provided by the Application Provider over NBI, is sent to the Partner OP by the Originating OP.
2. The Partner OP authenticates the Originating OP and validates the requested operation and the parameters e.g., Federation keys, Application Provider identifier, resource identifiers and the indicated Availability Zone.
3. Once the request is validated

    a) If the procedure is completed successfully, a response message HTTP POST response with “200: Resource reservation request accepted” shall be sent from the Partner OP.
    b) The Originating OP may send a GET request at later point of time to retrieve the reservation details with the Partner OP (see section 2.3.1.2.2)

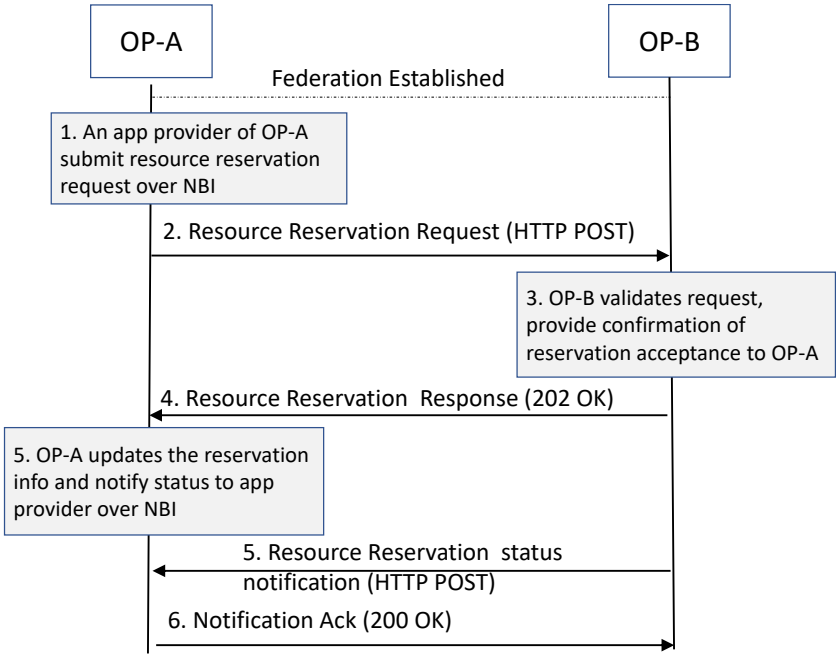


Figure 16: Resource Reservation request

2.3.1.2.2 View Resource Reservation

This is used by the Leading OP to retrieve the status of the already created resource pool with the Partner OP. The Leading OP uses the HTTP GET method to fetch the details of the resource pool as indicated by the Application Provider in a given Availability Zone.

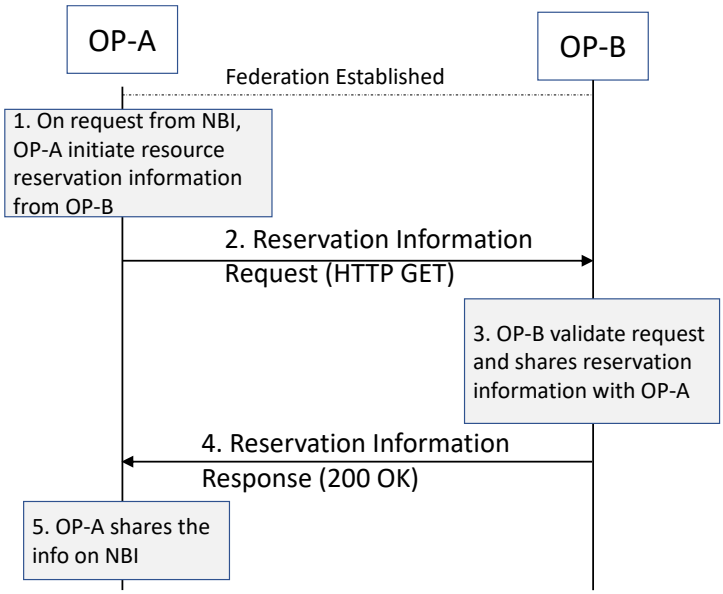


Figure 17: View Resource Reservation request

2.3.1.2.3 Update Resource Reservation

This procedure is used by an OP to update the existing resource reservation to a Partner OP. The Leading OP provides the Application Provider identifier, Availability Zone and operation to be performed e.g., add or remove the resources from a given resource pool etc.

The Leading OP uses the HTTP PATCH method to inform the Partner OP about the Application Provider identifier, Availability Zone identifier and resources to be updated.

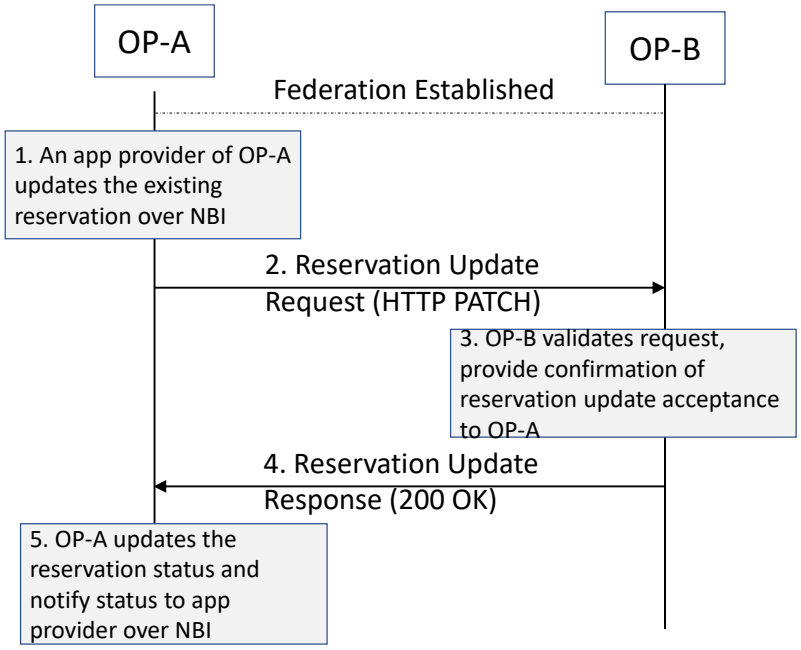


Figure 18: Resource reservation update request

2.3.1.2.4 Delete Resource Reservation

The Leading OP shall support the delete procedure to remove an existing resource reservation with a Partner OP when requested by the Application Provider over NBI.

An OP uses the HTTP DELETE method to inform the Partner OP of a request to remove a resource pool providing the Application Provider identifier, Availability Zone and existing reservation identifier earlier generated by the Partner OP during the creation of the pool.

After receiving the delete procedure response from the Partner OP, the Leading OP shall inform the Application Provider of the outcome of the remove operation on NBI.

2.3.1.3 Procedures for Application Onboarding Management Service

According to section 3.5.4.3 of the GSMA PRD OPG.02 [1] an OP shall be capable to onboard and manage Applications towards a Partner OP, assuming that a Federation has been established between Partner OPs previously.

Following procedures needs to be supported:

- Transfer Application Provider Criteria towards a Partner OP. The procedure may also request the launch of application instance(s) in a Partner OP’s Edge Resources as a follow-up action after onboarding.
- Transfer of other Application-specific files, e.g., Application manifest, specifying the workload information like mobility capabilities of the Application, Quality of Service (QoS) Performance Profiles and privacy policies etc., and other optional characteristics indicating the Application's request for, network capabilities, alternate QoS Performance Profiles etc
- Removal of Applications (Application images and metadata).

2.3.1.3.1 General

Application onboarding process on E/WBI is initiated by the Leading OP towards the Partner OP. An Application as described above comprises of Application components and meta-information which requires to be transferred over E/WBI to Partner OP and this process may take some time and the outcome or result of this operation can be notified by the Partner OP at a later point of time asynchronously to the Leading OP.

An Application may have one or more components having reference to the artefacts containing the component descriptors e.g., Helm charts, Container Specs etc. Also, the Application may be deployed on already reserved resources or from the available shared resources offered by the Partner OP in various Availability Zones. An Application meta-data may include references to reserved resources on Availability Zones to indicate if application instances should be deployed on resources already reserved.

2.3.1.3.2 Application Organization

An Application is logical group of related components that can be managed as a single unit by the OP. A component represent a runnable unit which is described using component descriptors. Application components descriptors e.g., Helm Charts, Container Specs, Terraform scripts etc. are provided by the Application Providers along with other Application characteristics e.g., QoS Performance Profile, Availability Zone info with Leading and federated OP, resource requirements etc. which may be considered for Application orchestration decisions by the OP.

Application components require reference to component image(s) which can be retrieved from public repositories, private repositories or may also be provided by the Application Providers to the OP by using OP supported image management capabilities. The Leading OP on behalf of Application Provider is responsible to transfer Applications and corresponding component descriptors and images to the Partner OP over E/WBI.

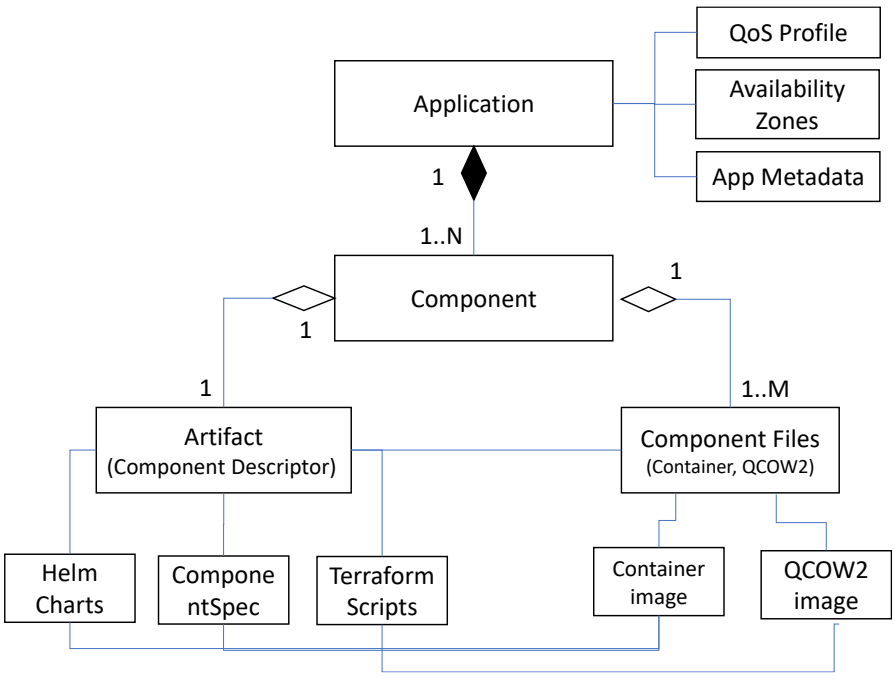


Figure 19: Application Schema

2.3.1.3.3 Onboard Application

An Application Provider uses the OP NBI to manage edge Applications via the Application management capabilities provided by the Leading OP. Using these capabilities an Application Provider can also request the Leading OP to share and deploy Applications in the federated partner Availability Zones.

These events on the NBI may also result into the Leading OP to initiate Application management procedures towards the Partner OP(s) over E/WBI and share the Application images, artefacts, and other meta information as provided by the Application Provider over NBI.

An OP can use Application onboarding APIs to submit an Application to a Partner OP Availability Zone(s). Submitting Applications may include Application images, Application type, Application Provider criteria, target Availability Zones etc. towards a Partner OP.

1. An onboard Application request is sent to a Partner OP.

    a) HTTP POST message contains Application details e.g., Application name, Application Identifier, Application Provider identifier, Artefact identifier, Availability Zone(s), QoS Performance Profile etc.

2. The Partner OP validates the OP identity and authorization info, Federation keys and Availability Zone onboarding status.

    a) If Application is already onboarded or is ongoing a correspondent failure response will be sent.
    b) Otherwise, after validations, the Partner OP shall respond to the Originating OP to indicate the progress of the request handling.

3. Once the Application push is finished

    a) If the procedure is completed successfully, a response message HTTP POST response with “201: Application onboarded” shall be sent from the Partner OP. (appId and requestId information can be included as well).
    b) If the Application handling procedures at the Partner OP cannot be completed immediately, the Partner OP shall respond with the HTTP “202 Accepted” to indicate that the Application onboarding is in progress.
    c) In other case a correspondent failure message will be generated from the Partner OP.

Note: This version of the document covers the resource model for Application where the Applications get resources as they need them. Coverage for resource reservation model will be provided in the next releases of this document.

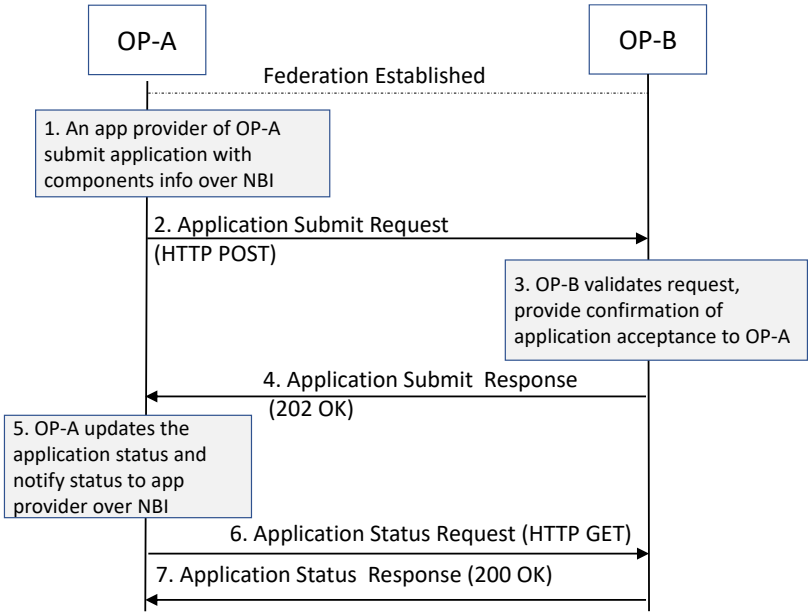


Figure 20: Onboard Application

2.3.1.3.4 Update Application Information

An OP must have alternatives to update parameters of an Application onboarded on a Partner OP. Update Application information towards a Partner OP (e.g., Application versions, Application Provider criteria, target Availability Zones).

Application update request can be initiated by the Leading OP due to the Application Provider initiated Application update action over NBI.

1. The Leading OP shall send the HTTP PATCH request message to the Partner OP(s) to start the Application update procedure.

    a) HTTP PATCH message may contain Application details e.g., Application name, Application Identifier, Application Provider identifier, Availability Zone(s), QoS Performance Profile etc.

2. The Partner OP validates the OP identity and authorization info, Federation keys etc.
3. On successful validation, if Application indicated by Application Identifier is already onboarded and parameters to be updated are valid then

    a) If the update procedure is completed successfully, a HTTP PATCH response with “200: Application updated successfully” shall be sent from the Partner OP. (Application name and Application Identifier information can be included as well).
    b) If the Application update procedures at the Partner OP cannot be completed immediately, the Partner OP shall respond with the HTTP “202 Accepted” to indicate that the Application onboarding is in progress.
    c) In other case a correspondent failure message will be generated by the Partner OP.

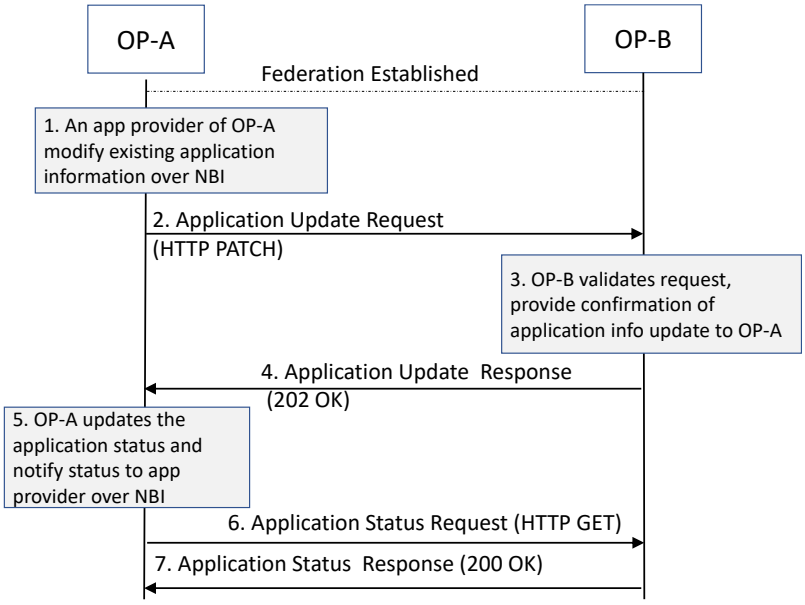


Figure 21: Update Onboarded Application

Note: The Leading OP shall make the Application update results available on the NBI interface as the Application Provider may have started the update procedure over NBI. It is to be noted that the Application information update does not results in updating the existing instances of the Application or create new instances with updated information.

Note: After successful update of the Application information, the Application Provider can request to instantiate the Application instances with updated information on one or more Availability Zones used earlier during the onboarding procedure.

2.3.1.3.5 Remove Application

This will be use by an OP to remove an Application from a Partner OP Availability Zone. Removal of Applications (Application images and metadata) from a Partner OP. The Leading OP shall make the Application de-boarding result available on the NBI interface.

After successful de-boarding of the Application, the Application and any of the associated information e.g., images, metadata etc. shall no longer be available in the indicated Availability Zones.

Note: Verification and compliance of the removal of Application information by an OP is beyond the scope of this document and such requirements and verification process shall be part of other specifications e.g., GSM PRD OPG.02 [1].

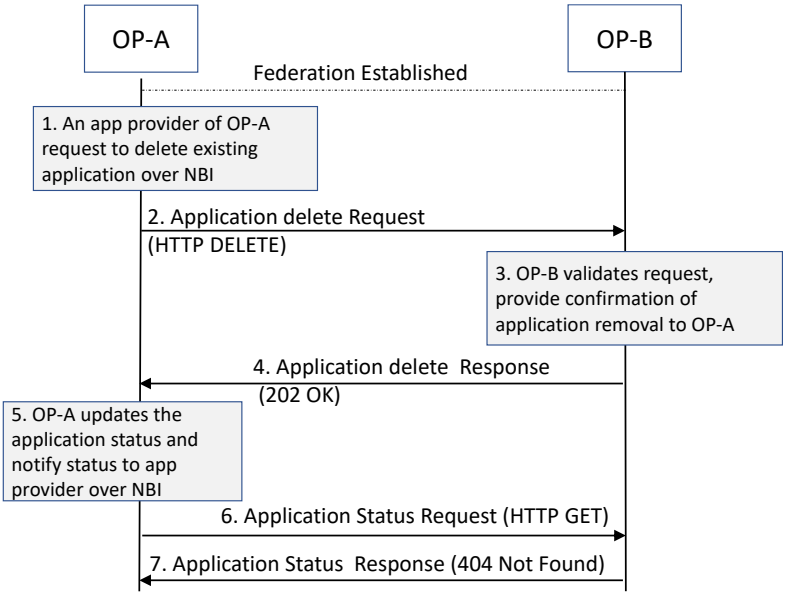


Figure 22: Delete Application

2.3.1.3.6 Onboard Application On New Availability Zones

This Originating OP can use this API to request the Partner OP to make already onboarded Applications available on additional Availability Zones as indicated in the HTTP POST request from the Leading OP.

1. An onboard Application to new Availability Zones request is sent to a Partner OP.
    a) HTTP POST message contains Application Identifier, Availability Zone(s) etc.
2. The Partner OP validates the OP identity and authorization info, Federation keys and Availability Zone onboarding status.
    a) If Application is already onboarded on the indicated Availability Zone a failure response will be sent.
    b) Otherwise, after OP validation the Partner OP proceed to update the local database to update the Application additional Availability Zone indicated in the request.
3. Once the request handling is finished by the Partner OP
    a) If the procedure is completed successfully, a response message HTTP POST response with “201: Application onboarded” shall be sent from the Partner OP.
    b) In other cases a correspondent failure message will be generated from the Partner OP as detailed in the API parameters description table in section 4.

2.3.1.3.7 Restrict Application On Specific Availability Zones

This Originating OP can use this API to request the Partner OP to either restrict or allow application instantiation of already onboarded Applications to a given Availability Zone.

1. The Leading OP sends a request to a Partner OP.

a) HTTP POST request message contains Application Identifier, Availability Zone(s), restriction condition (allow, restrict) etc.

2. The Partner OP validates the OP identity and authorization info, Federation keys and Availability Zone onboarding status.

   a) If Application on indicated Availability Zone is not already onboarded, a failure response will be sent.
   b) Otherwise, after OP validation, the Partner OP proceed to update the local database about the Application Availability Zone restriction status indicated in the request.

3. Once the request handling is finished by the Partner OP

   a) If the procedure is completed successfully, a response message HTTP POST response with “200: Application restriction updated successfully” shall be sent from the Partner OP.
   b) If the request handling at the Partner OP cannot be completed immediately, the Partner OP shall respond with the HTTP “202 Accepted” to indicate that the request handling is in progress.
   c) In other cases a correspondent failure message will be generated from the Partner OP as detailed in the API parameters description table in section 4.

2.3.1.3.8 Network Events Notifications

The Partner OP may send network events in context of the Application instances that it may have created based on the application instance provisioning requests from the Leading OP if the given Application has subscribed for network capabilities.

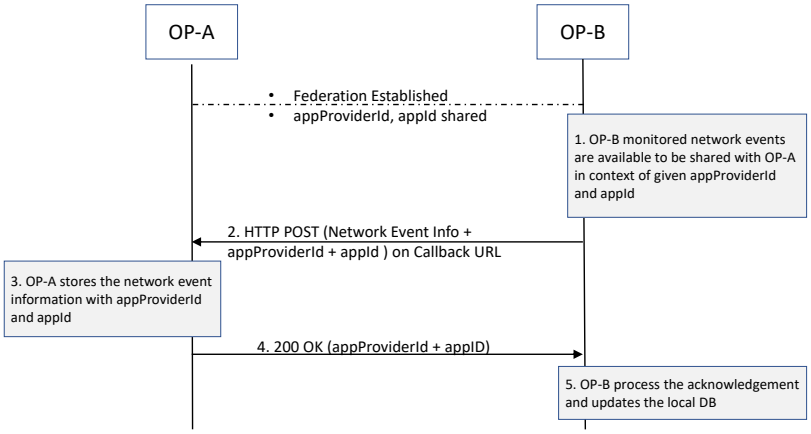


Figure 23: Network Event Notifications

The message flows for event notifications in context of the running instances of the edge Applications:

3. The Partner OP-B determines the availability of network event information (SLIs) that needs to be shared with OP-A and it maps the network events information with the Application onboarded by the Leading OP.

a) The Partner OP provides all required identification, authentication, and authorisation information elements required to allow the Leading OP to decide if the request can be granted.

4. The Partner OP sends the HTTP POST request to the Leading OP-A with appProviderId, appId, network event information on Callback URL which OP-A has earlier provided during the Application onboarding request.
5. After authentication and authorization of OP-B, the Leading OP i.e., OP-A validates the E/WBI Notification API from OP-B.
6. The Leading OP sends a HTTP response to the Partner OP-B to inform about the result of the operation.
7. The Partner OP-B updates the processing status of the given network event in its local DB for the given subscriptionID, Application ID.

2.3.1.4 Application Deployment Management Service

As defined in GSMA PRD OPG.02 [1], the Application Deployment Management Service on E/WBI shall control the launch and termination of Applications that have been onboarded on a Partner OP.

2.3.1.4.1 Instantiate Application

This API will be use by an OP to instantiate an Application to Edge Resources of the Partner OP and to a Partner OP Availability Zone(s) as requested by Application Provider over NBI.

The Partner OP shall also provide the application instance status over E/WBI to Leading OP which the Leading OP for example may expose to Application Providers on NBI on request from the Application Providers.

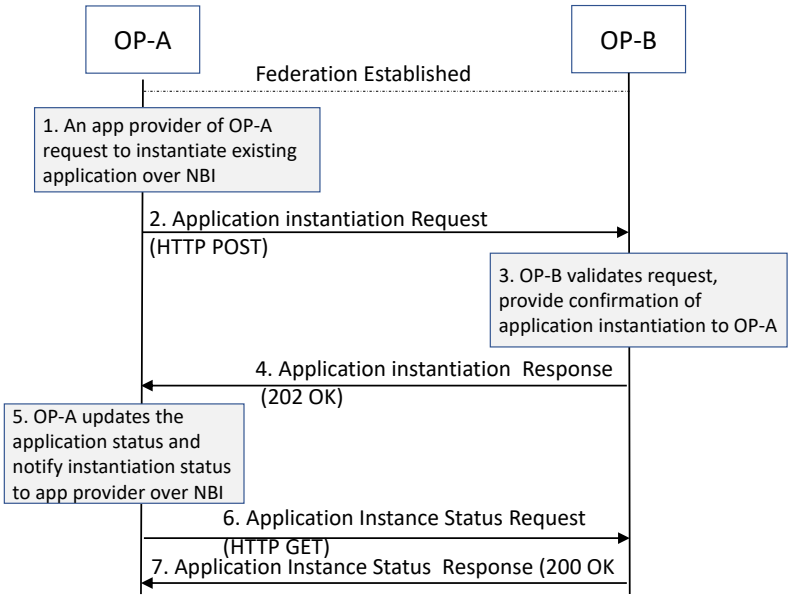


Figure 24: Application Instantiation

This Originating OP uses this procedure to request the Partner OP to instantiate an application already onboarded on various Availability Zones of the Partner OP.

1. The Originating OP sends a request to the Partner OP to create application instances on one or more Availability Zones.

    a) The HTTP POST message contains Application Identifier, Availability Zone(s) etc.
    b) The HTTP POST message also includes a transaction identifier for tracking future responses and retries if a response is not received. The Originating OP saves the transaction identifier in a local DB for future use

2. The Partner OP validates the OP identity and authorization info, Federation keys and Availability Zone onboarding status.

    a) If the Application is not available on indicated Availability Zones, a failure response will be sent.
    b) Otherwise, after OP validation the Partner OP shall proceed to launch application instances on indicated Availability Zones in the request.

3. Once the request handling is finished by the Partner OP

    a) If the request is accepted by the Partner OP, a response message (HTTP POST response) with “202: Application Instantiated” shall be sent from the Partner OP.
    b) In other cases a corresponding failure message will be generated by the Partner OP as detailed in the API parameters description table in section 4.

The Originating OP uses a transaction identifier with the application instantiation requests to track the future responses about the application instance status at the Partner OP. If no response is received from the Partner OP, the Originating OP can retry the request with the same transaction identifier indicating a retry of a past request to the Partner OP.

The Partner OP shall store the transaction identifiers in a local DB to handle any future retries of the application instantiation requests by the Leading OP. The Partner OP on receiving the application instantiation request shall analyze the transaction identifier from the request to confirm whether a past request is being retried by the Originating OP. If so, it can return the application instantiation status from its local DB including the same transaction identifier in the response.

Note: The number of retries and the retry interval is an implementation choice and not specified in this document.

2.3.1.4.2 Notify Application Instance Information

After successful instantiation of the application, the Application Provider should be able to view the application instance information on partner Availability Zone(s) e.g., Application Identifier, instance identifier, health status, network interfaces communication endpoints etc.

The application instantiation request may take time for Partner OP to create the application instance on the indicated Availability Zone(s). Based on the result of the instantiation the Partner OP sends the notification request (HTTP POST) to the Leading OP with the application instance information e.g., application instance identifier, Application Identifier, Availability Zone meta-information, application instance endpoints etc.

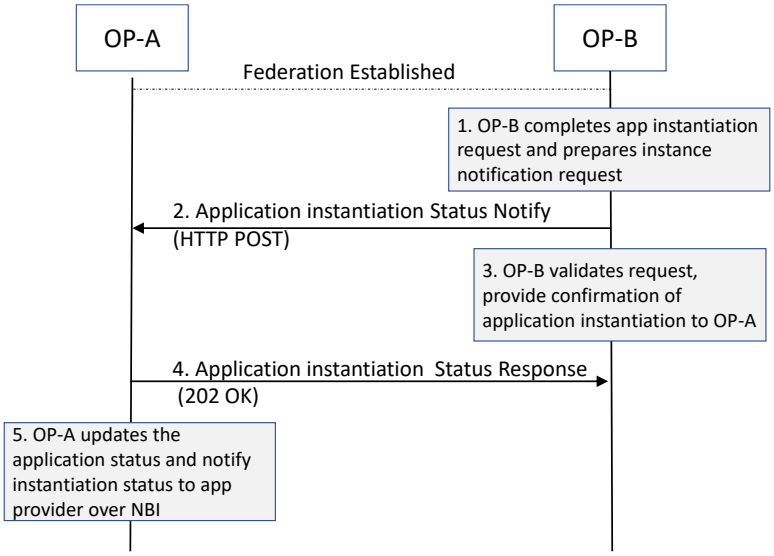


Figure 25: Application Instance Status Notification

2.3.1.4.3 Terminate Application Instance Information

After successful instantiation of the application, the Application Provider should be able to request the termination of application instance on one or more Availability Zone of Leading and/or Partner OP.

Application Providers request the application instance termination via NBI, and the Leading OP shall initiate HTTP DELETE request towards the Partner OP. The request may contain the information e.g., Application Identifier, instance identifier etc. to enable Partner OP to identify the application instance uniquely on his Edge Resources.

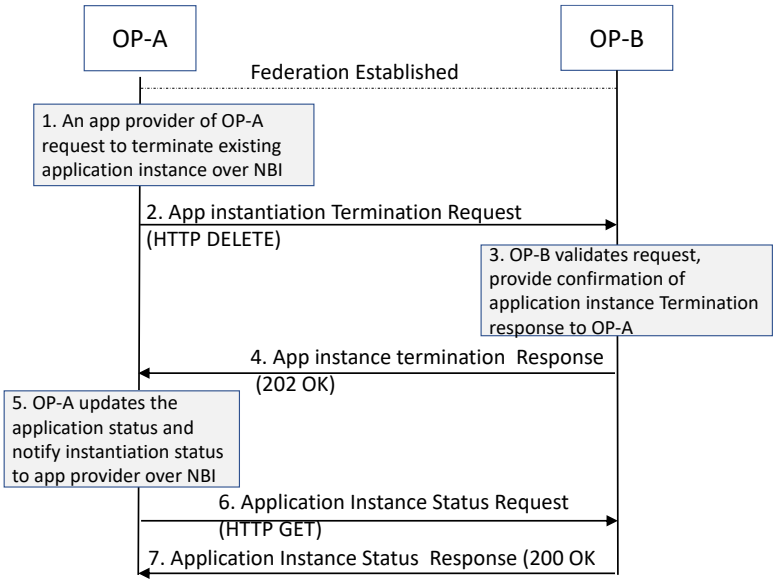


Figure 26: Application Instance Termination

2.3.1.5 Procedures for sharing Edge Resources between federated Partner OPs

As defined in the section 3.3.5 of the GSMA PRD OPG.02 [1], Edge node sharing is the concept for two Operators to share edge nodes (should be read as Edge Resources in the Partner OP Availability Zones) between their coverage area for example from a geographical point of view (south and north).

2.3.1.5.1 Edge node discovery procedure with Partner OP

A Subscriber of Operator A accesses its home network/Home OP and requests for the required Edge Application instantiation. When Operator A's OP identifies that the most suitable Edge Resources are in Partner B, Operator A's OP requests by an HTTP POST message over the E/WBI to Partner B's OP (see Figure below, steps 3, 3a and 3b) to provide the suitable Availability Zone(s) where Application can be hosted in the Partner OP Edge Resources. Alternatively, the Partner OP can also provide the communication endpoints of existing application instances to Home OP.

In this example, since the two OPs have a Federation agreement, they may have pre-established commercial agreements, security relationships and policy decisions (for instance, QoS-related). Operator B’s OP sends the response for the HTTP POST request to OP-A (assuming enough Edge Resources are available at OP-B). The OP-B response contains the Application endpoint (e.g., FQDN) on the Edge Resources of OP-B at which the Subscriber can connect to the Application.

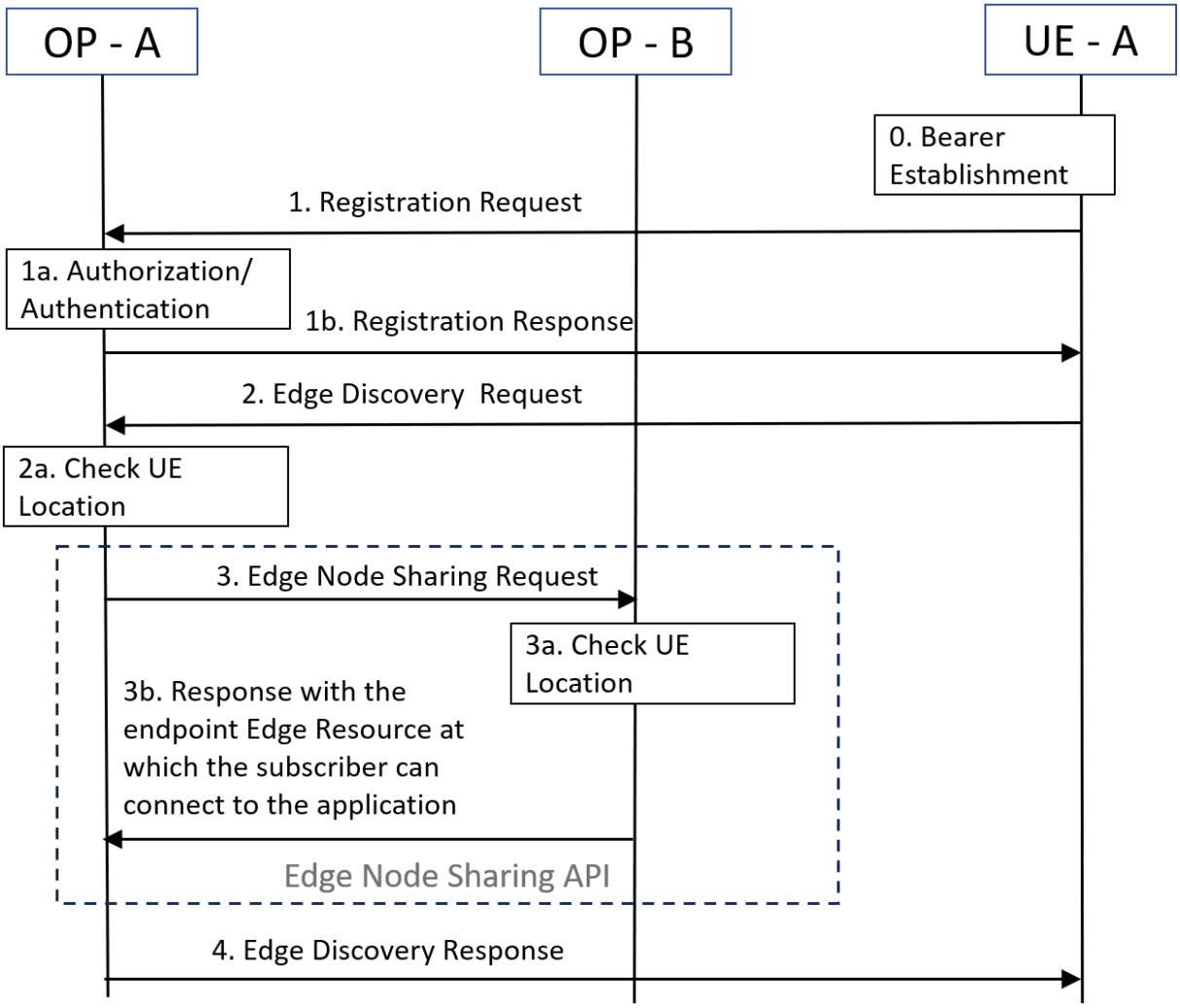


Figure 27: Sharing Edge Resources between federated Partner OPs

Based on the OP-B response to edge node (Edge Resources in the Partner OP Availability Zones) discovery request, the OP A sends an Edge discovery response to the UE, which includes information about the discovered Application endpoint (e.g., FQDN) from OP B.

If the OP-A includes the Edge node discovery filters in Edge discovery request, the OP-B response may include additional information regarding matched capabilities, e.g., service permissions levels, Key Performance Indicators (KPIs), Edge Application locations(s) that the Edge node can support.

The Edge discovery response from OP-B may contain a list of Edge Node endpoints. This list may be based on Edge discovery filters containing a Geographical or Topological Service Area, e.g., Latitude/Longitude of the UCs, Application Identifier etc. In case of failure, OP B should send an appropriate failure response including the cause to Leading OP.

2.3.2 Service API Usage on E/WBI

This section describes the OP services that may not strictly require edge capabilities with the Partner OPs. Such services may require E/WBI to support the capabilities they offer e.g., Service API Federation to enable the Leading OP to identify the Partner OP that should be handling the requested service.

The Leading OP shall be responsible to perform the authentication and authorization of the Application Provider using credentials provided with the Service API and the E/WBI API requests from the Leading OP will be considered as authenticated by the Partner OP when shared over an already established Federation relationship.

For the Service API Federation, the Leading OP E/WBI may not be sharing the details of the relationship or agreement details of its API Consumers with the Partner OP. The Leading OP may share an identifier with the Partner OP over E/WBI to represent such agreements at the Leading OP for the capability monitoring and consumption tracking purposes. These identifiers can then be used in subsequent API requests over the E/WBI shared in the context of a given Service API invocation and related events.

The Leading OP shall be able to determine the Partner OP i.e., “Service API Routing”, which shall execute the Service API. Also, the Leading OP depending upon the nature of the Service API may need to perform “Service API Context Management” to relate any future events to the API session between the Leading and the Partner OP.

2.3.2.1 Service API Routing on E/WBI

The Service API routing refers to the process at the Leading OP that helps the Leading OP to route the Service API to the federated Partner OP which should be serving the API request.

2.3.2.1.1 Subscriber Identifier Based Routing

The Service APIs may include Subscriber identifiers which the Leading OP can use to determine the federated Partner OP. The Subscriber identifiers e.g., UE IP address, MSISDN, GPSI etc. could be part of the Service APIs.

The Leading OP receiving the Service APIs can use these identifiers from the APIs and the prior information shared by the Partner OPs over the E/WBI to determine the Partner OP that should serve a given request.

2.3.2.1.1.1 UE IP Address

The UE IP address assigned to a UE (or PDU Session) could change over a period based on the Operator-specific policies and may get reassigned to other UEs at a later point of time.

Such an IP Address if included in the Service API can be used by the Leading OP to determine the federated Partner OP that the given UE IP address belongs to and route the Service API request over the E/WBI to that Partner OP.

The E/WBI shall provide the capabilities to the OPs to synchronize the IP address management events e.g., reassignment of IP addresses to other devices, changes in IP addresses range, expiry of the End-User mapping with the IP address etc.

2.3.2.1.1.2 MSISDN

The Service APIs may include static Subscriber identifiers e.g. MSISDN, GPSI etc. which doesn’t change over a much longer period. The E/WBI shall provide the capabilities to the OPs to exchange the set of such public identifiers which an OP is authorized to serve.

2.3.2.1.2 Availability Zone Based Routing

In some of the cases where the Service APIs does not include Subscriber identifiers or include the service to made available in specific locations e.g., Availability Zones as location indicator, the Leading OP may have option to determine the federated Partners providing services in those locations and route the request to the selected Partner OP.

2.3.2.2 Procedures for Service API Context Management

The Service API capabilities e.g., Quality on Demand (QoD) by nature may remain active in the network for a longer duration of time and may have associated events in its lifetime which the Partner OP shall share with the Leading OP e.g., change in level of QoS etc.

For handling such APIs, a session context associated to the API state needs to be managed which is defined as Service API Context Management and details are described in below sections. As described above the session context creation depends on the nature of the Service API that does not need to be created for all the Service APIs.

2.3.2.2.1 Service API Context Creation

The API Consumers of the Leading OP may invoke Service APIs that intend to request a specific network capability e.g., Quality on Demand (QoD) for an End-User's PDU session by providing their Public IP address.

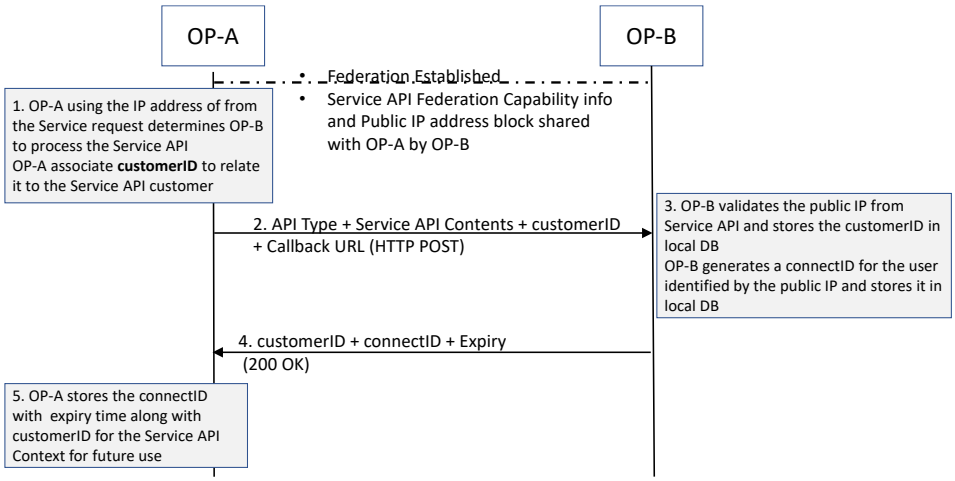


Figure 28: Service API Forwarding

The message flow for Service API Federation containing the UC Public IP address is as follows:

1. A Leading OP-A compares the public IP address from the Service API with public IP addresses provided by the Partner OPs and determines that API shall be served by OP-B.
    - The Leading OP provides all required identification, authentication, and authorisation information elements required to allow the Partner OP to decide if the request can be granted.
    - The Leading OP identifies the Application Provider and maps it to an identifier customerID to be used in the context of the Service API on the E/WBI.
    - The Leading OP stores the customerID to the Service API context in a local DB.

Note: The customerID identifier is independent of the API sessions and can also be used by the Partner OP to consult the End-User as identified in the Service API request for obtaining the Consent for sharing information about the End-User e.g., identity related information etc. in context of the given customerID.

2. The Leading OP sends the HTTP POST request to the Partner OP-B with the customerID, Service API type, Service API contents and a Callback URL that OP-B shall use to provide updates on future events in conjunction with the Service API context at OP-B.

Note: The OP-A shall be able to map the event information received on the OP-A Callback URL to the Callback URL received in the NBI Service API request if the Service API contains a Callback URL.

3. After authentication and authorisation of OP-A, the Partner OP i.e., OP-B validates the E/WBI Service API Federation request from OP-A and stores the API information at OP-B. The Partner OP on successful validation of the received public IP address assigns an identifier, connectID, for the End-User currently assigned the given public IP address.
4. The Partner OP stores the connectID for the received Service API context along with the customerID in local DB and sends a HTTP response to the Leading OP to inform about the result of the operation.
    - On success, a 200 OK message is sent along with a message body containing customerID, connectID with expiry time to the Originating OP.
    - On failure, an appropriate error code (e.g., 401, 404 etc.) along with Application-level error message shall be returned.
    - The server errors 500 (Internal Server Error), 503 (Service Unavailable) may also indicate that the request could not be processed by the Partner OP and should be retried at a later point of time.
5. The Leading OP stores the connectID and the expiry time to the Service API context earlier created in local DB along with the customerID and starts the connectID validity timer.

Note: The connectID generated at OP-B refers to an End-User subscription and the OP-B can confirm the identity of the End-User at the time of reception of Service API request on the E/WBI. In future the public IP address association may change but the connectID can still point to the same End-User subscription at OP-B.

Note: The OP-B should be including the connectID in subsequent requests to the Callback URL for the events related to the Service API context in which the given connectID was generated.

2.3.2.2.2 Service API Context Events Notifications

The Partner OP may send network events in context of the Service API session that it may have created for handling the Service API request.

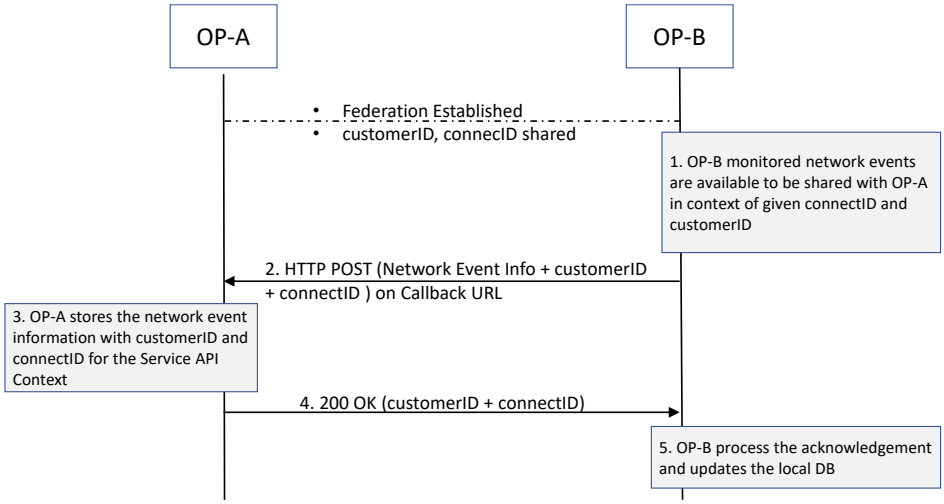


Figure 29: Service API Event Notifications

The message flows for event notifications in context of the Service API Federation:

1. The Partner OP-B determines the availability of network event information that needs to be shared with OP-A and it retrieves the connectID, customerID from the local DB in the context of the Service API request session.
    - The Partner OP provides all required identification, authentication, and authorisation information elements required to allow the Leading OP to decide if the request can be granted.
2. The Partner OP sends the HTTP POST request to the Leading OP-A with customerID, connectID, network event information on Callback URL that OP-A has earlier provided along with the Service API request.
3. After authentication and authorisation of OP-B, the Leading OP i.e., OP-A validates the E/WBI Notification API from OP-B.
4. The Leading OP sends a HTTP response to the Partner OP-B to inform about the result of the operation.
5. The Partner OP-B updates the processing status of the given network event in its local DB for the given connectID, customerID.

2.3.2.2.3 Service API Context Deletion By Leading OP

The Leading OP can initiate the process of deleting the Service API context identified by the connectID and customerID for the events e.g., connectID validity timer expiry, a request from the Application Provider to remove the ongoing session etc.

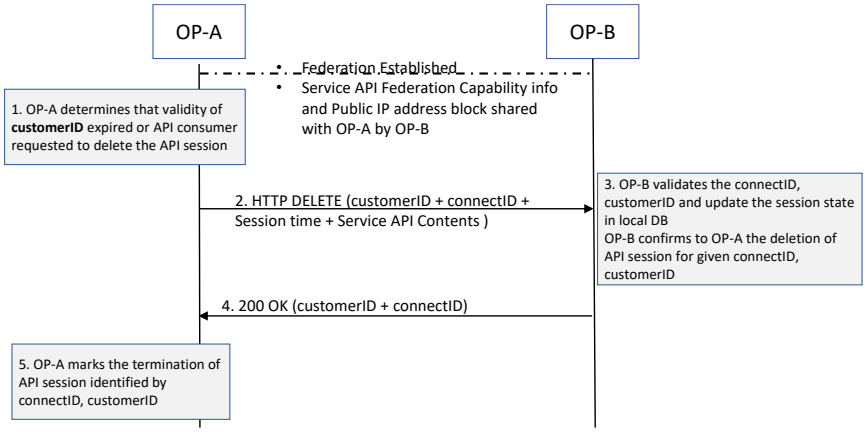


Figure 30: Service API Context Termination by Leading OP

The message flow for Service API context termination over E/WBI is as follows:

1. The Leading OP-A prepares the termination event e.g., due to the validity of the customerID having expired or a request from the Application Provider to delete the API session.
    - The Leading OP provides all required identification, authentication, and authorisation information elements required to allow the Partner OP to decide if the request can be granted.
2. The Leading OP sends the HTTP DELETE request to the Partner OP-B with the customerID, connectID, API session duration at OP-A and Service API contents.
3. After authentication and authorization of OP-A, the Partner OP i.e., OP-B validates the E/WBI Service API Federation request from OP-A and updates the session state identified by connectID and customerID.
4. The Partner OP includes connectID, customerID in the HTTP response to the Leading OP to inform about the result of the operation.
5. The Leading OP-A updates the API session state for the given connectID and customerID and based on local policy can archive or remove the session state.

2.3.2.2.4 Service API Context Retrieval By Leading OP

The Leading OP can initiate a GET request to the Partner OP to query for the Service API context information that is identified by the connectID and customerID.

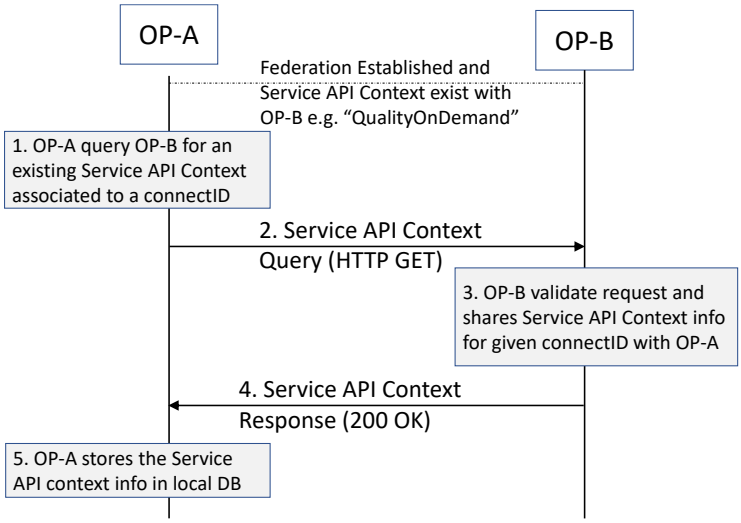


Figure 31: Service API Context Retrieval by Leading OP

The message flow for Service API context retrieval over E/WBI is as follows:

1. The Leading OP-A prepares the HTTP GET request to retrieve the Service API context information for an existing connectID due to an event e.g., the API Consumer requested to fetch API session information.
    - The Leading OP-A provides all required identification, authentication, and authorisation information elements required to allow the Partner OP-B to decide if the request can be granted.
2. The Leading OP-A sends the HTTP GET request to the Partner OP-B with the customerID, connectID, API session duration at OP-A and Service API contents.
3. After authentication and authorization of OP-A, the Partner OP-B validates the E/WBI Service API context request from OP-A and retrieves the session state information identified by connectID and customerID.
4. The Partner OP-B includes connectID, customerID and the API specific information as defined in Service API specifications in the HTTP response to the Leading OP-A to inform about the result of the operation.
5. The Leading OP-A may update the API session state in a local DB.

2.3.2.2.5 Service API Context Deletion By Partner OP

The Partner OP can also initiate the process of deleting the Service API context identified by the connectID and customerID for the events e.g., due to an Operator decision to terminate the services for a Subscriber, etc.

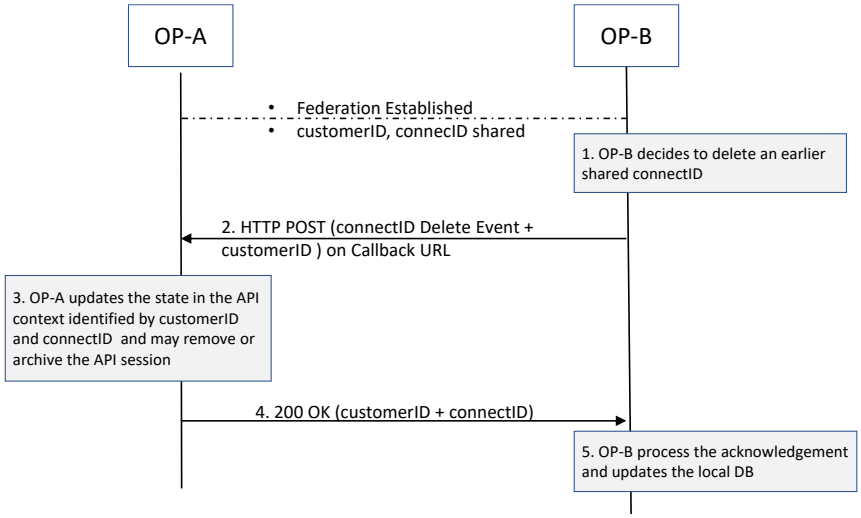


Figure 32: Service API Context Termination by Partner OP

The message flow for Service API context termination over E/WBI is as follows:

1. The Partner OP-B determines the need for a termination event due to OP-B's decision to terminate the API session for a given End-User, etc.
    - The Partner OP provides all required identification, authentication, and authorisation information elements required to allow the Leading OP to decide if the request can be granted.
2. The Partner OP sends the HTTP POST request to the Leading OP-A with the customerID, connectID, API session duration at OP-B on the Callback URL earlier shared by OP-A.
3. After authentication and authorisation of OP-B, the Leading OP-A validates the E/WBI POST method from OP-B with event type as “session deleted” and moves the session state identified by connectID, customerID to the terminated state.
4. The Leading OP includes the connectID, customerID in the HTTP response to the Partner OP to inform about the result of the operation.
5. The Partner OP-B changes the API session state to terminated for the given connectID and customerID and frees up the connectID.

2.3.2.2.6 ConnectID Validity Expiration Notification at the Partner OP

The Partner OP may send a connectID expiry event to the Leading OP for cases such as an earlier provided connectID validity expires or the association of the connectID changes to other End-Users etc. and which would also result into the removal of the existing API context that the Leading OP may have.

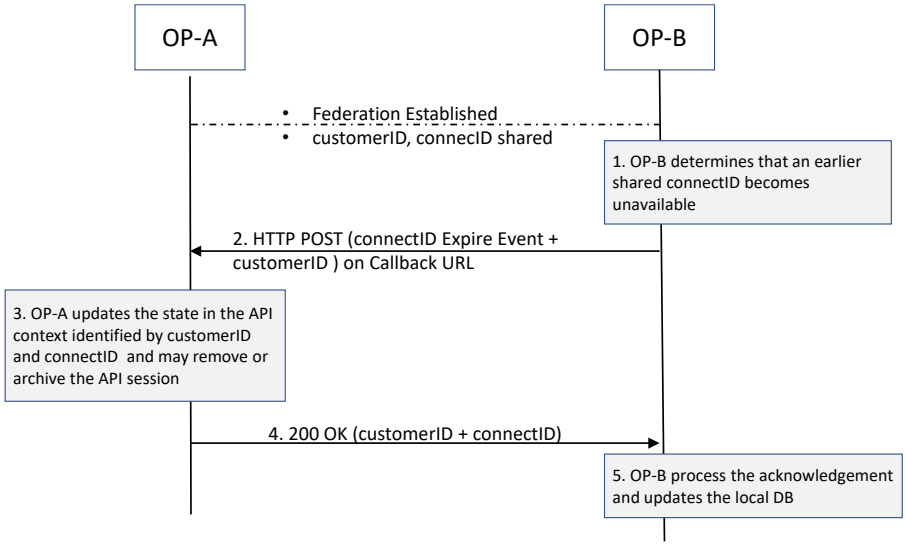


Figure 33: connectID Expiry Notification

The message flows for connectID event notification in context of the Service API Federation:

1. The Partner OP-B determines that the earlier shared connectID with the Leading OP is getting unavailable for the ongoing Service API session.
    - The Partner OP provides all required identification, authentication, and authorisation information elements required to allow the Leading OP to decide if the request can be granted.
2. The Partner OP sends the HTTP POST request as a notification to the Leading OP-A with customerID, connectID on Callback URL which OP-A has earlier provided along with the Service API request.
3. After authentication and authorisation of OP-B, the Leading OP i.e., OP-A validates the E/WBI Notification API from OP-B and stores the network event information in an OP-A local DB. OP-A may remove the given API session for the given connectID, customerID as per the local policy of the OP-A.
4. The Leading OP sends a HTTP response to the Partner OP-B to inform about the result of the operation.

Note: The Leading OP in future shall not send any request over the E/WBI in the context of the expired connectID irrespective of the result of the handling of connectID expiration request from OP-B.

5. The Partner OP-B updates the processing status of the given connectID expiry event in its local DB.

2.4 Resource Monitoring Procedures

Resource Monitoring in context of the Operator Platform may include usage of compute infrastructure, performance metrics, aggregate statistics of the various metrics, etc.

An OP should be able to collect the necessary resource monitoring information as per the requirements defined in GSMA PRD OPG.02 [1]. Some of the information collected should be reported to the Leading OP's by the Partner OP.

2.4.1 Edge Resource Monitoring Procedures

The Edge Resource monitoring information that a Partner OP can share comprises of e.g., virtual CPU, memory, storage and network utilization, Application registry utilization by the Leading OP apps, utilization of resources from reserved pools, compute flavours utilization etc.

Metering measures levels of resource utilization such as usage of CPU and memory, network bandwidth, data storage volume etc consumed by the Applications of the Application Providers (also referred to as tenants). The E/WBI should enable an OP to share such information with the federating Partner OPs to inform about the overall resource utilization against what they have been offered.

The resource monitoring information on E/WBI should be provided at Availability Zone level.

Note: It is expected that Usage data reporting is enabled by default.

2.4.1.1 Subscribe Edge Resource Monitoring

The Originating OP uses the subscription procedure to indicate the Partner OP to start the reporting of the Edge Resource information.

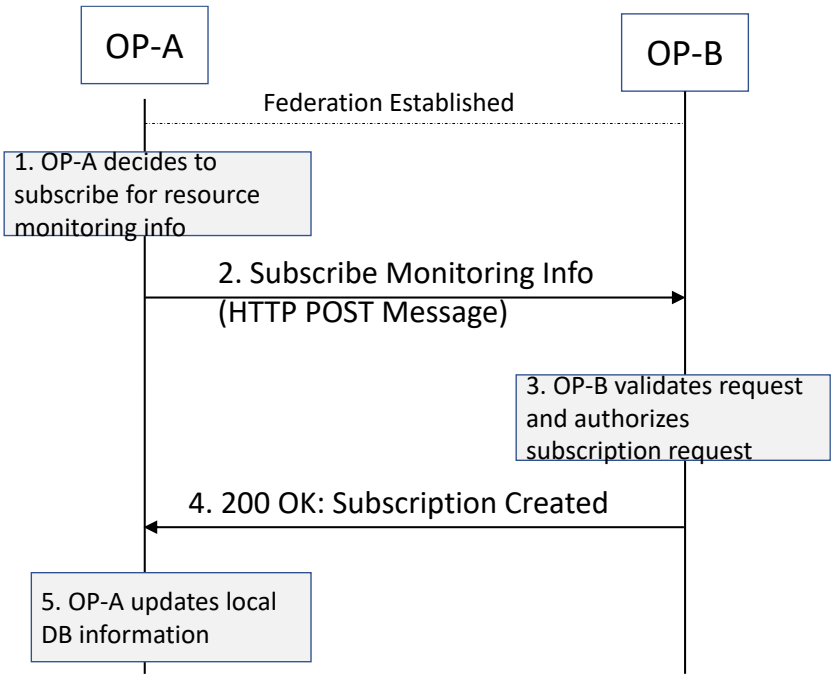


Figure 34: Subscribe Edge Resource Monitoring

2.4.1.2 Unsubscribe Edge Resource Monitoring

The Originating OP uses the HTTP DELETE method to indicate the Partner OP to stop the reporting of the Edge Resource information.

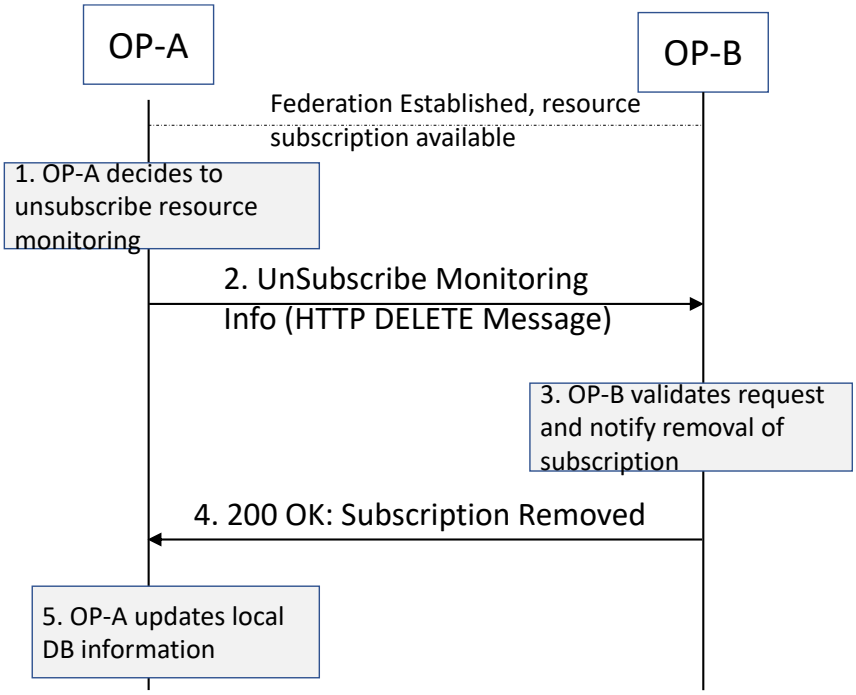


Figure 35: Unsubscribe Edge Resource Monitoring

2.4.1.3 Reporting of Edge Resource Monitoring Information

The Partner OP uses the Edge Resource monitoring procedure to report the Edge Resource monitoring information to the Originating OP.

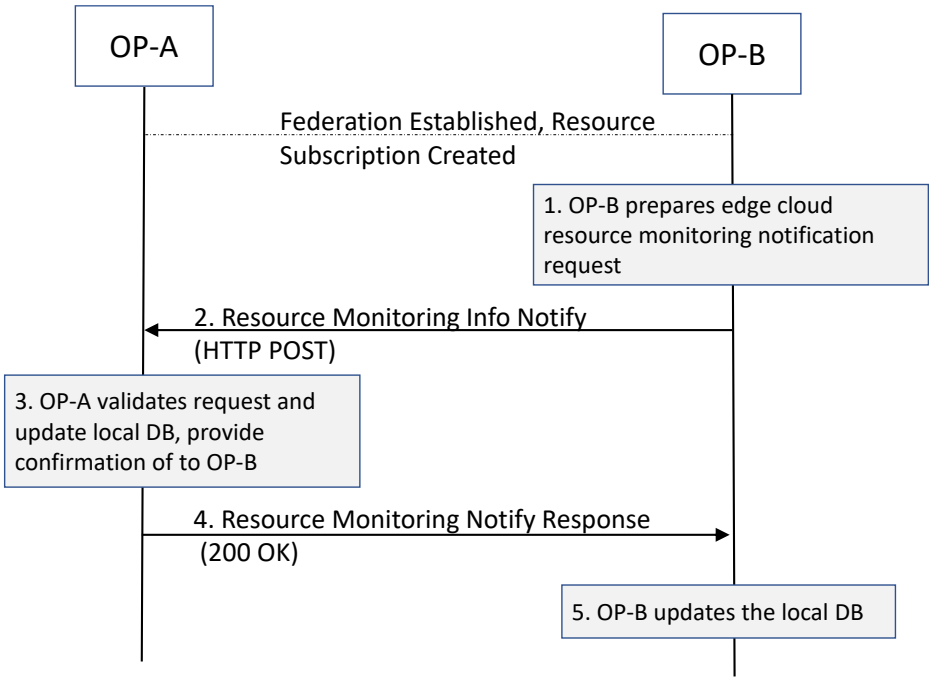


Figure 36: Reporting Edge Resource Monitoring

The Partner OP uses the periodicity interval provided by the Originating OP in the Federation create request to send the monitoring information reports periodically to the

Originating OP. Monitoring reports are sent by the Partner OP on the Callback URL provided by the Originating OP in the Federation create request.

2.4.1.4 Retrieve Edge Resource Monitoring Information

The Originating OP uses the HTTP GET method to retrieve Edge Resource monitoring information from the Partner OP.

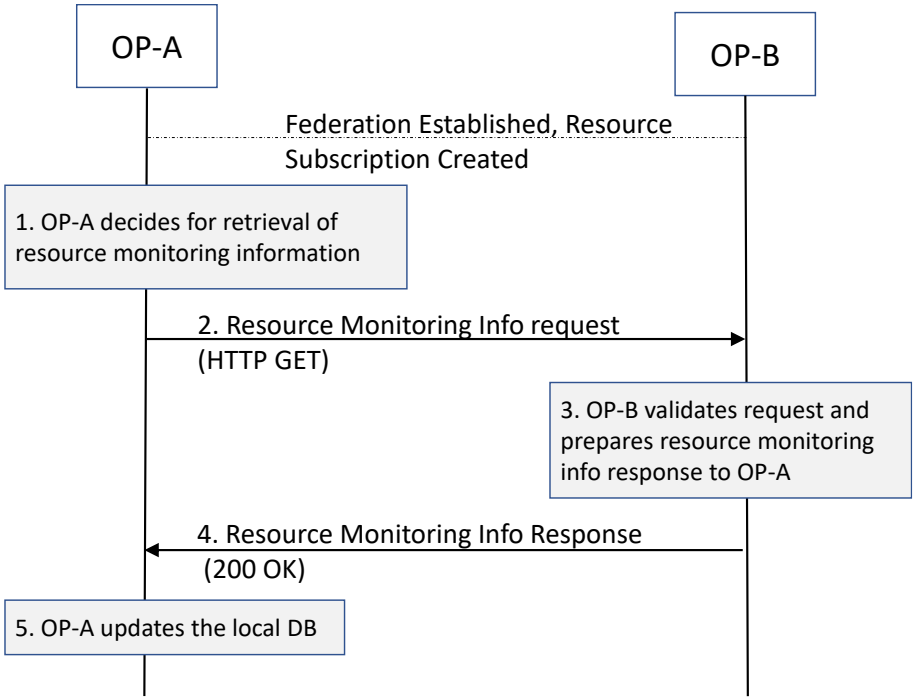


Figure 37: Retrieve Edge Resource Monitoring Information

2.4.2 Applications Resource Monitoring Procedures

The Applications resource monitoring information that a Partner OP can share comprises of resource utilization by Applications of an Application Provider (tenant) in one or more Availability Zones. The Applications may utilize resources e.g., virtual CPU,memory, storage and network, Application registry storage, compute flavors etc.

2.4.2.1 Subscribe Application Resource Monitoring

The Originating OP uses the subscription procedure (HTTP POST) to indicate the Partner OP to start the reporting of the resource information of an Application in one or more Availability Zones.

Procedure for creating subscription for Application resource monitoring information is similar to Edge Resource monitoring subscription procedure with the request carrying identification information for Applications and corresponding instances.

2.4.2.2 Unsubscribe Application Resource Monitoring

The Originating OP uses the unsubscribe procedure (HTTP DELETE) to indicate the Partner OP to stop the reporting of the resource information of an Application in one or more Availability Zones.

Procedure for removing subscription for Application resource monitoring information is similar to Edge Resource monitoring unsubscribe procedure with the request carrying corresponding subscription identifier.

2.4.2.3 Reporting of Applications Resource Monitoring Information

The Partner OP uses the Applications resource monitoring procedure (HTTP POST) to report to the Originating OP the Application resource monitoring information on per Availability Zone basis.

Procedure for sharing the Application resource monitoring information by the Partner OP is similar to Edge Resource monitoring reporting procedure with the request carrying corresponding subscription identifier and Application resource information.

2.4.2.4 Retrieve Applications Resource Monitoring Information

The Originating OP uses the HTTP GET method to retrieve Applications resource consumption information from the various Edge Resources that the Partner OP has assigned.

Procedure for retrieving the Application resource monitoring information by the Originating OP is similar to Edge Resource monitoring retrieval procedure with the request carrying corresponding subscription identifier and Application information.

2.5 Operational Visibility Procedures

An OP should be able to report the events that may be important for the other Federation partners to know about. These events could be the faults affecting the offered resource availability, consumption of resources reaches a pre-defined threshold, etc.

2.5.1 Fault Management

The fault management procedures on E/WBI enable the Partner OP to inform the Originating OP about the service impacting events that may affect the resource availability or degradation of the services that the Partner OP has offered to the Originating OP. These events are referred as “Alarm” in the context of the E/WBI.

The fault management operations in context of the OP are derived from TMForum TMF642 [5] specification. The E/WBI shall be defining the specific alarms and associated traits that covers the requirements defined in the GSMA PRD OPG.02 [1].

Below sections describe the subset of procedures to perform the alarm management over E/WBI in alignment with the TMForum TMF642 [5] specification.

2.5.1.1 Subscribe Alarm Reporting

The Originating OP uses the subscription procedure to indicate the Partner OP to enable the alarm reporting functionality for the existing Federation.

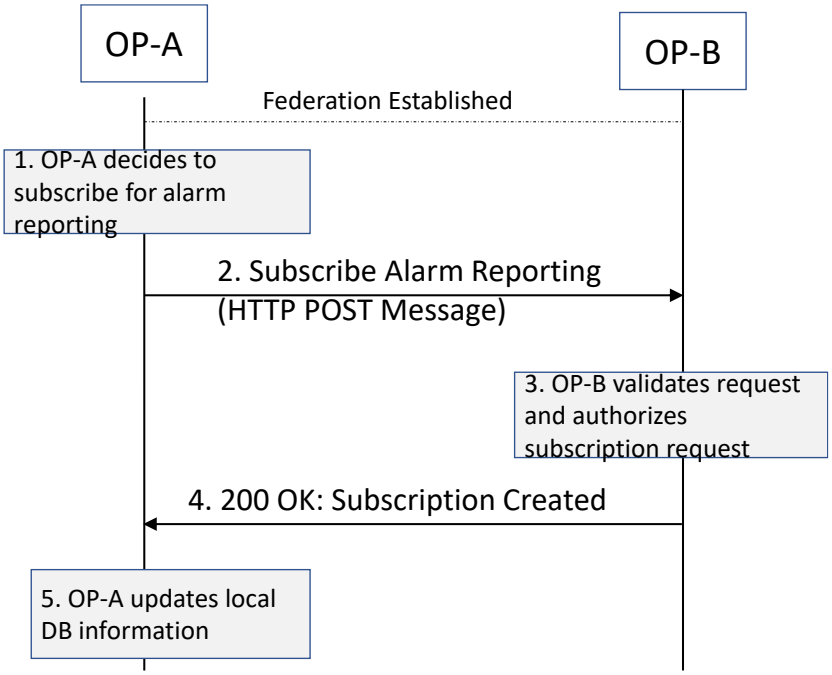


Figure 38: Subscribe Alarm Reporting

2.5.1.2 Unsubscribe Alarm Reporting

The Originating OP uses the HTTP DELETE method to indicate the Partner OP to stop the alarm reporting functions for the existing Federation.

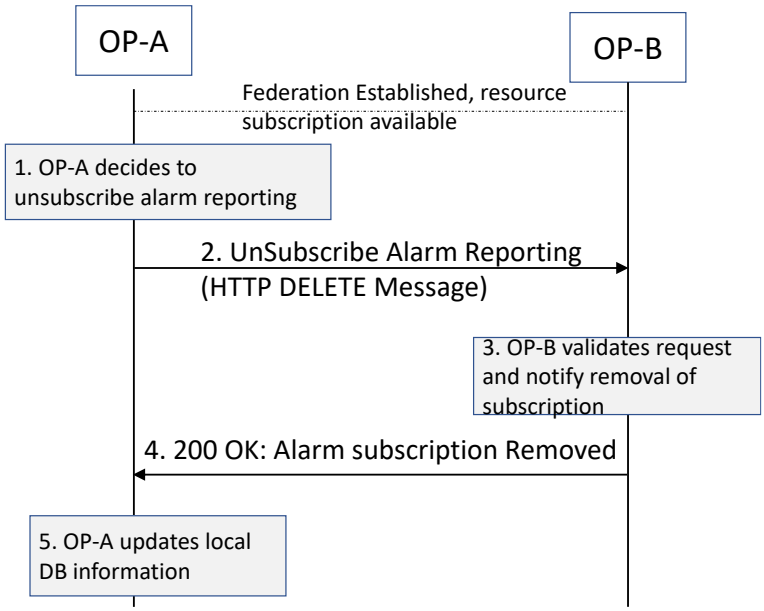


Figure 39: Unsubscribe Alarm reporting

2.5.1.3 Raise Alarm

The Partner OP uses the Raise Alarm procedure to notify an alarm on detecting any fault in the context of resources and services offered to the Originating OP.

The alarm raised by the Partner OP is delivered to the Originating OP via the HTTP POST method on E/WBI with the inclusion of an alarm identifier provided by the Partner OP. This identifier is used later for updating, deleting or querying the alarm information.

An alarm at a minimum should include the following information,

- An alarmType, to categorize alarms in broad categories of related problems
- A probableCause, is an agreed set of descriptions of the probable causes.
- A specificProblem can include specific descriptions particular to an OP implementation of an alarmType
- An alarmDetails, more detailed information about the fault may be provided in alarmDetails

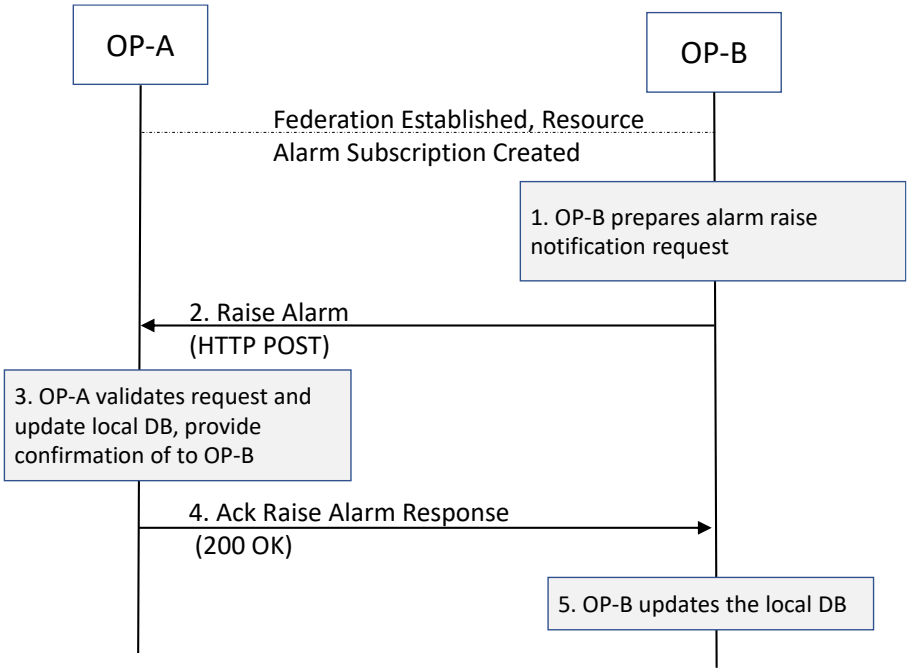


Figure 40: Raise Alarm Request

2.5.1.4 Update Alarm

The Partner OP uses this procedure to update the alarm information already reported to the Originating OP.

The update alarm procedure is performed by HTTP PATCH method by the Partner OP towards the Originating OP. PATCH method allows to partially update the alarm information of an existing alarm with an alarm identifier.

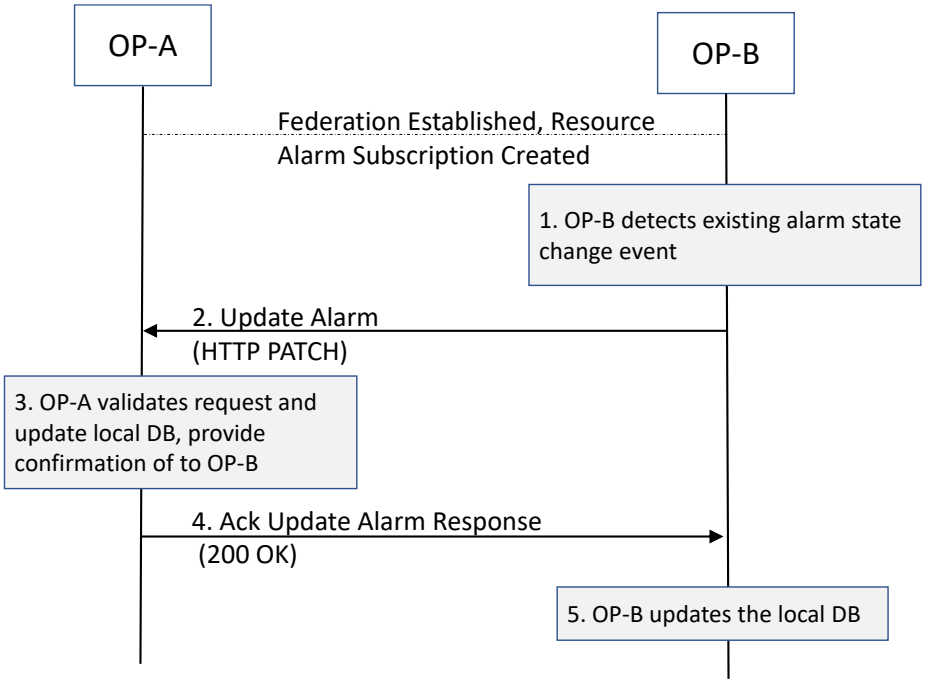


Figure 41: Update Alarm Request

2.5.1.5 Clear Alarm

The Partner OP uses this procedure to clear an existing alarm earlier raised towards the Originating OP.

The Partner OP uses the HTTP DELETE method to clear the existing alarm with an alarm identifier at the Originating OP.

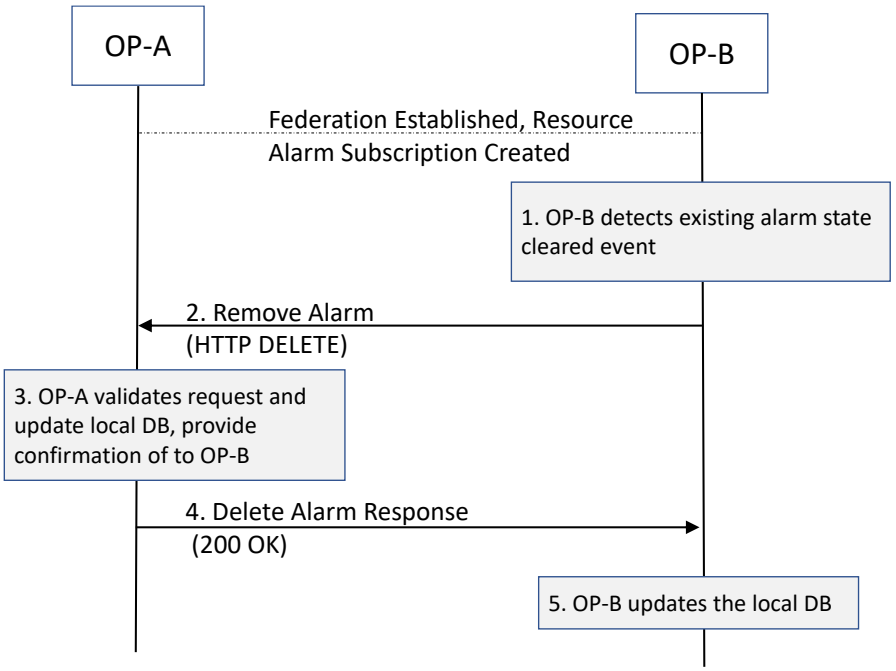


Figure 42: Clear Alarm Request

2.5.1.6 Retrieve Alarms Information

The Originating OP uses this procedure to retrieve the list of active alarms or for an alarm identifier. To fetch the list of active alarms, the Originating OP may use combination of one or more attributes of alarms e.g., alarm type, alarm ID etc.

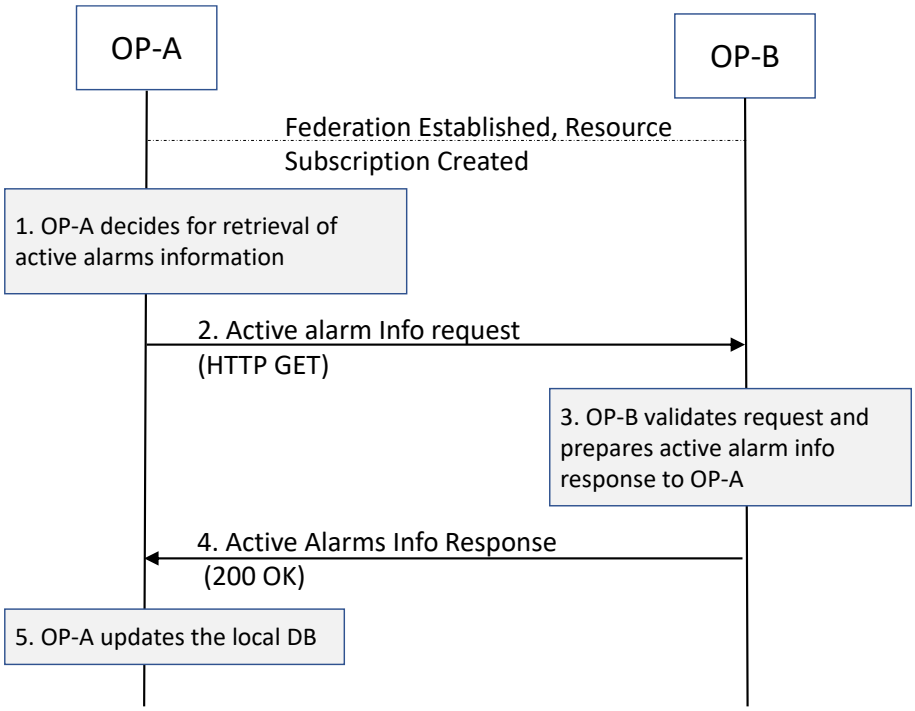


Figure 43: Retrieve Active Alarms Info Request

2.5.2 Event Management

The event management procedures on E/WBI to enable the Partner OP to inform the Originating OP about the specific events e.g., CPU or memory utilization in an Availability Zone reaches 80% and remains at that level for 15 minutes etc.

On detecting such conditions, the Parter OP shall notify the occurrence immediately to the Originating OP along with the event specific information.

2.5.2.1 Subscribe Event Monitoring

The Originating OP uses the subscription procedure to indicate the Partner OP to enable the event detection functionality for the existing Federation.

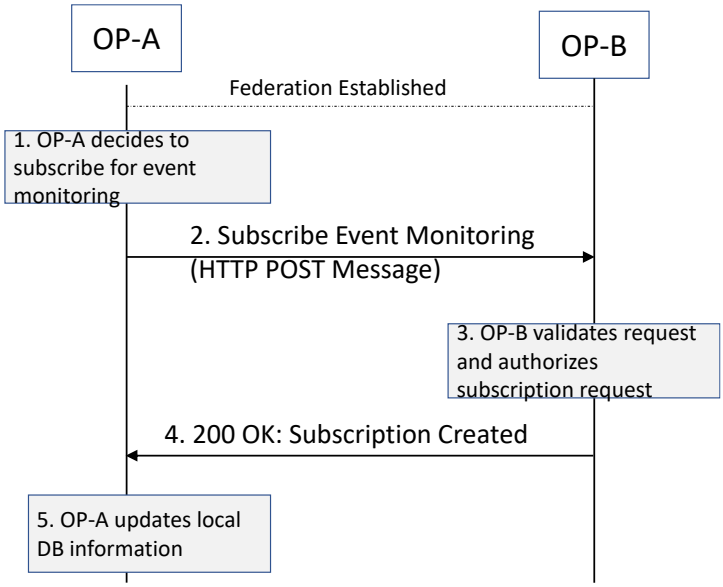


Figure 44: Subscribe Event Monitoring

2.5.2.2 Unsubscribe Event Monitoring

The Originating OP uses the HTTP DELETE method to indicate the Partner OP to stop the event detection functions for the existing Federation.

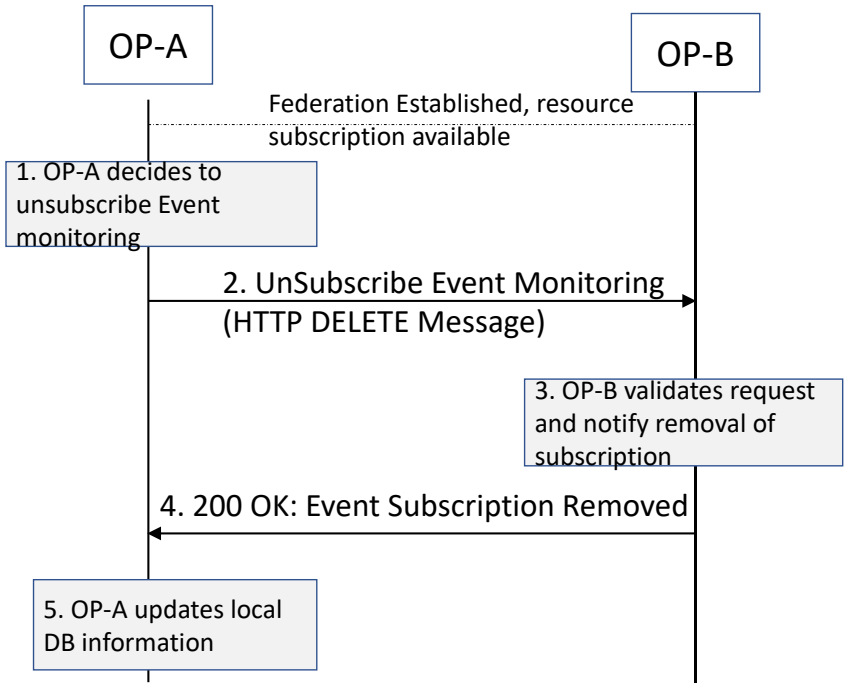


Figure 45: Unsubscribe Event Monitoring

2.5.2.3 Create Event Criteria

The Originating OP uses the subscription procedure (HTTP POST) to define the trigger conditions which on detection by the Partner OP result into a event notification towards the Originating OP.

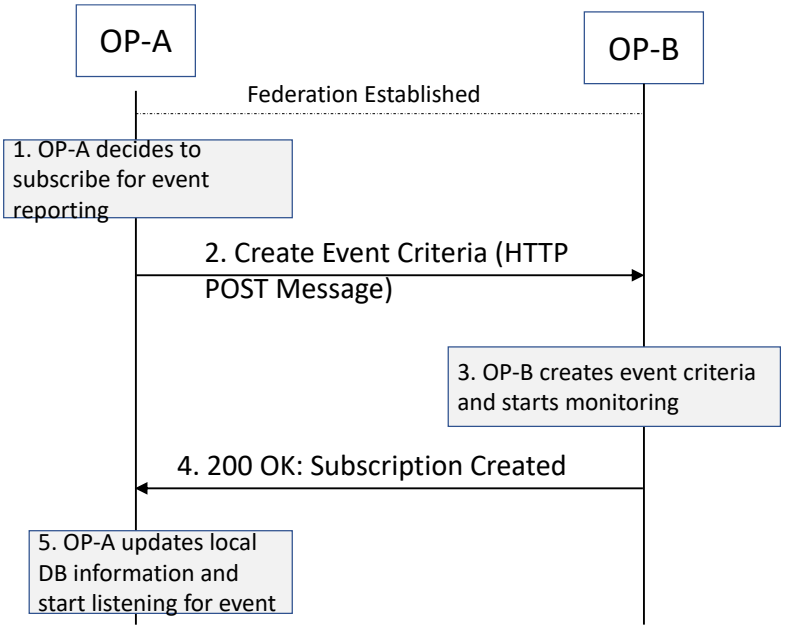


Figure 46: Create subscription for Event Information

2.5.2.4 Remove Event Criteria

The Originating OP uses the unsubscribe procedure (HTTP DELETE) to remove the event criteria created earlier at the Partner OP.

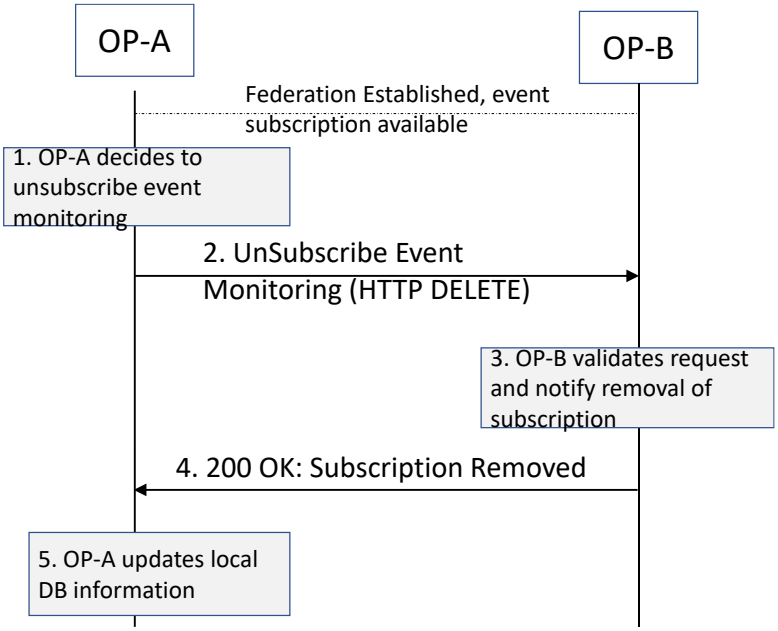


Figure 47: Remove subscription for Event Information

2.5.2.5 Event Reporting

The Partner OP uses the event criterions created on request by the Originating OP to monitor the occurenaces of events associated to the given criterions and notify the Originating OP via HTTP POST method to provide information about such events.

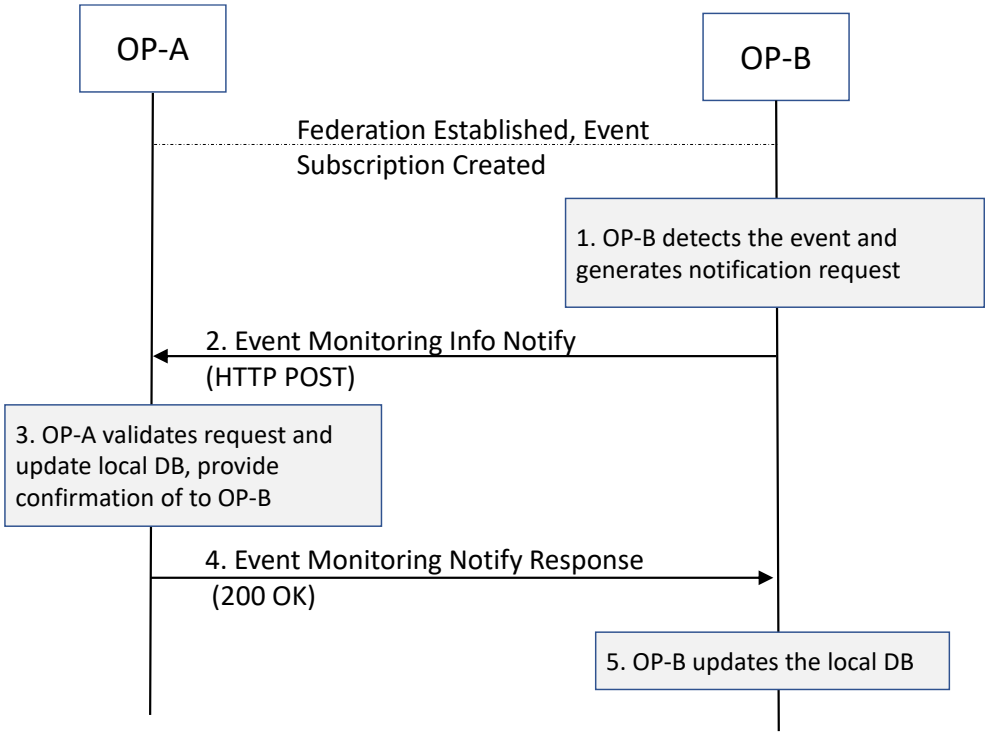


Figure 48: Notification of Event Information

2.5.2.6 Retrieve Event Criterions

The Originating OP uses the HTTP GET method to retrieve set of event criterions created at the Partner OP.

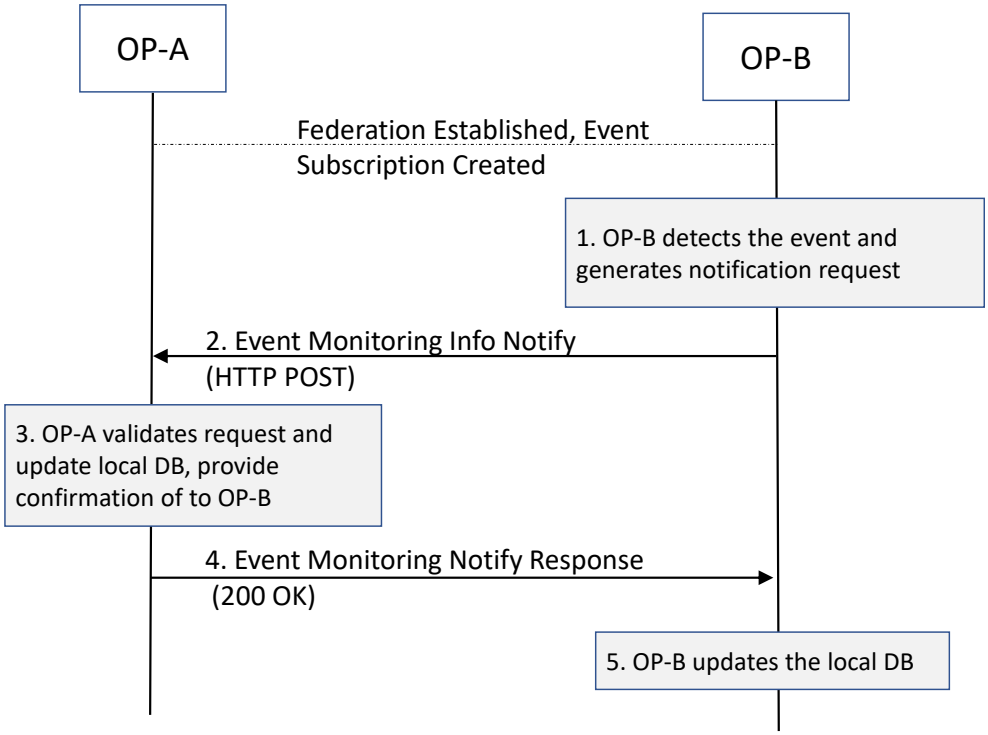


Figure 49: Retrieve Event Information

2.6 Notification Management

Subscription management procedures allow an OP to register its request to get notified for various events like Application relocations, critical alarms, periodic reporting of the monitoring information etc. The procedure gives control to an OP to subscribe or unsubscribe from receiving reports of such events.

2.6.1 Network Events Notification Management Procedures

The notification management process is initiated by the Originating OP towards the Partner OP over already established Federation i.e., Notification Creation request for reporting of events in context of OP services which could affect the agreed SLAs between the Partner OPs.

The Originating OP provides the Callback URI or HTTP listner along with the events of interests to receive the future notifications from the Partner OP on one or more Federation resources. The Partner OP stores the Callback URL and indicated resources on successful acceptance of the notification creation request from the Originating OP.

2.6.1.1 Notification Context Creation Request

The Originating OP uses the HTTP POST method to subscribe for the reporting of events associated to network capabilities managed by the Partner OP.

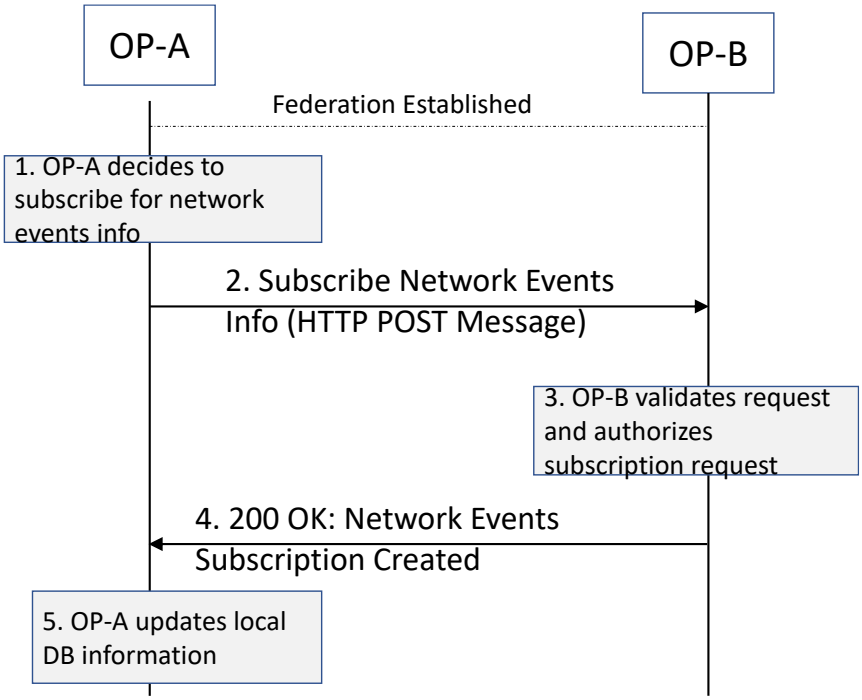


Figure 50: Subscribe for Network Events Notifications

2.6.1.2 Remove Existing Notification Context

The Originating OP uses the HTTP DELETE method to request the Partner OP to remove the existing network events notification subscription and stop sending any further notifications for the the given network capabilities (API resources).

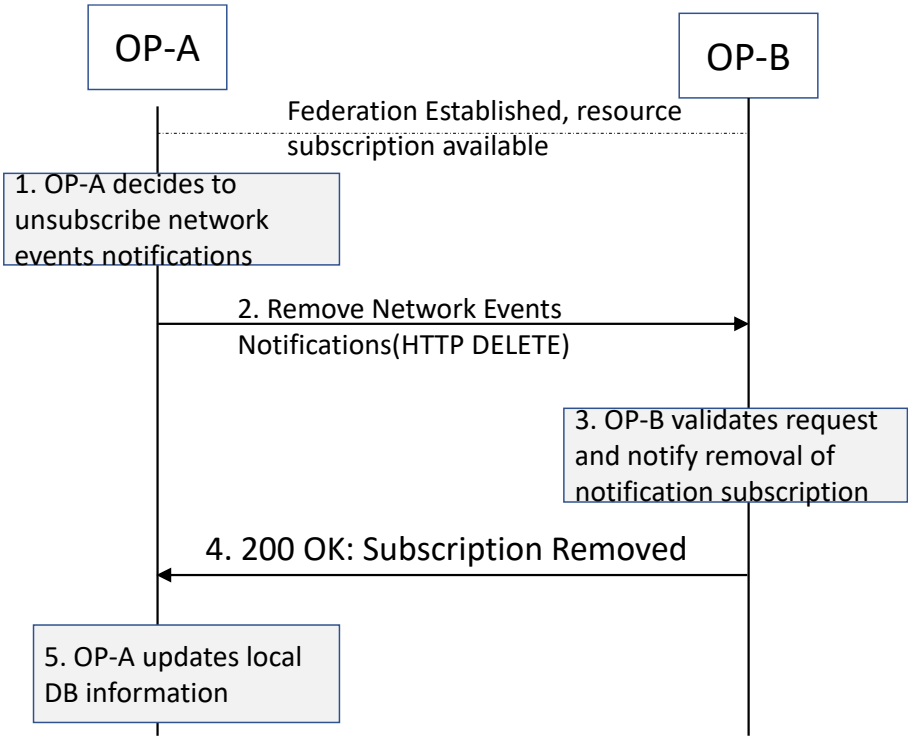


Figure 51: Unsubscribe Existing Notification Context

2.6.1.3 Add Network Capability Monitoring To Notification Context

The Originating OP uses this procedure (HTTP POST) to add the network capabilities in context of an Application which the Partner OP should monitor for the given network capabilities and notify the Originating OP at agreed periodicity.

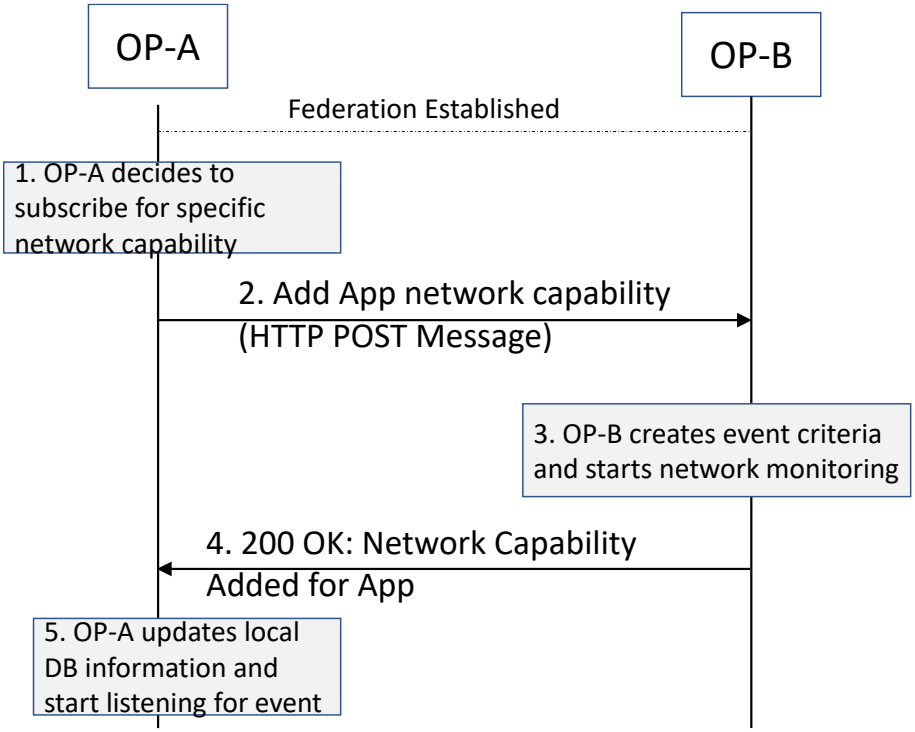


Figure 52: Create subscription for Network Event Information

2.6.1.4 Remove Network Capability Monitoring From Notification Context

The Originating OP uses this procedure (HTTP DELETE) to remove the monitored network capabilities in context of an Application and stop reporting network events to the Originating OP.

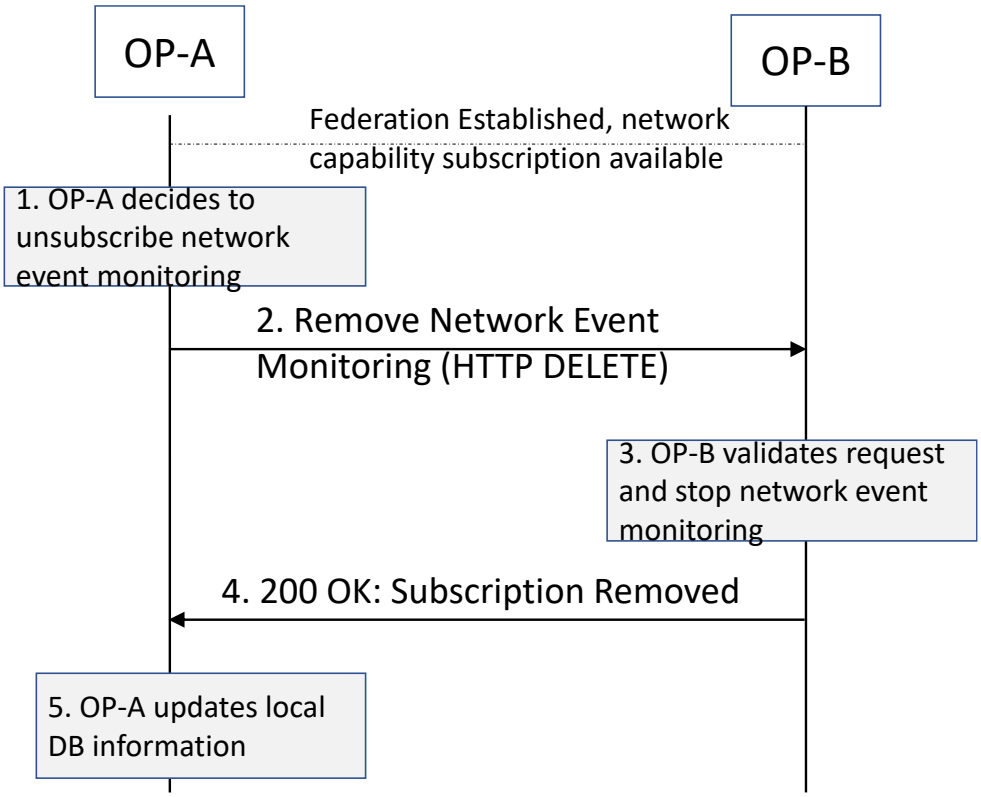


Figure 53: Remove subscription for Event Information

2.6.1.5 Network Event Reporting

The Partner OP applies and monitors the network capabilities submitted by the Originating OP for the set of Applications. Partner OP collects the network events for the given Applications and notify the Originating OP via HTTP POST method to report the collected information about network events.

The Partner OP may report the network events monitoring information at periodicity agreed at the time of subscribing for the notification context creation process by the Originating OP. The reporting of collected network event information may not be performed in real-time by the Partner OP.

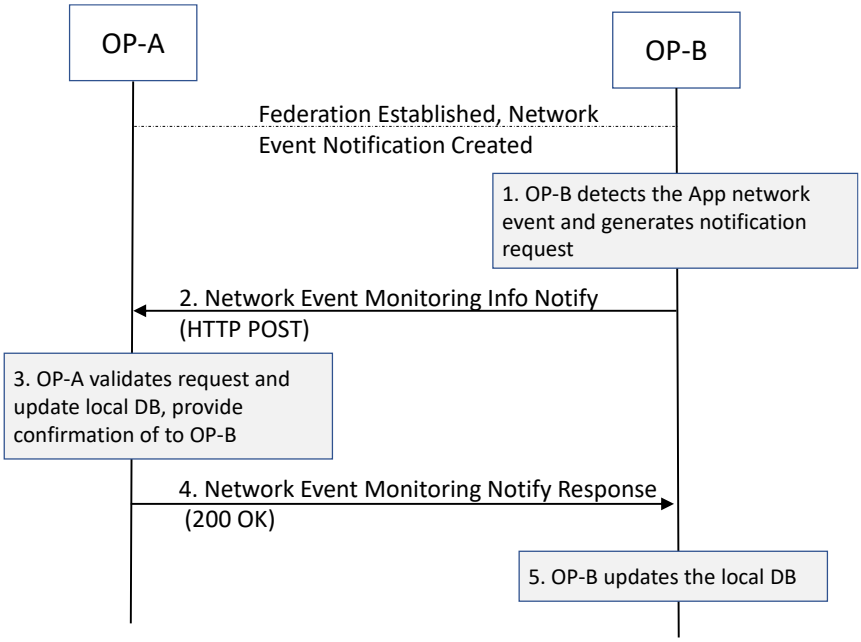


Figure 54: Notification of Network Event Information

2.6.1.6 Retrieve Network Event Information

The Originating OP uses the HTTP GET method to retrieve network event information available at the Partner OP.

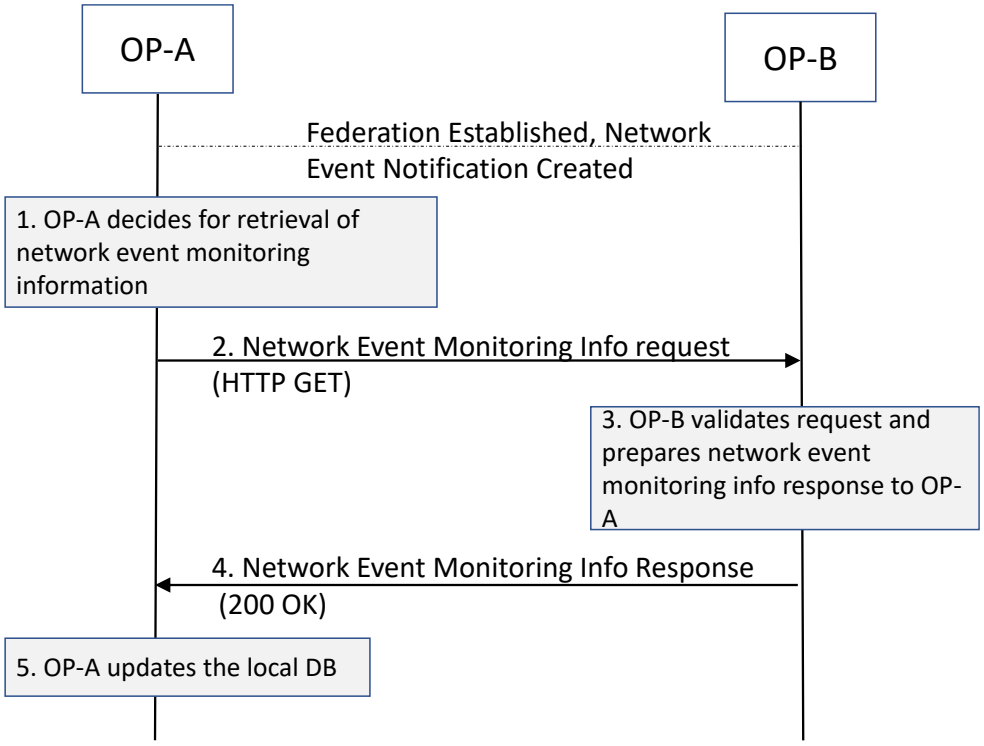


Figure 55: Retrieve Network Event Information

2.6.2 Applications Events Report Notification Procedures

The Applications events notifications procedures on E/WBI provides capabilities to report Application related events by the Partner OP to the Originating OP in the context of federated Applications.

The Application-level events may include information about the events in the context of the applications instances that the Partner OP may generate or detect e.g., instantiation and termination events, Application relocations, restarts, scaling up or down etc,

Using the HTTP POST request the Originating OP requests the Partner OP to enable the reporting of the Application's events periodically towards the Originating OP over an already established Federation.

The Originating OP provides the Callback URI or HTTP listner along with the events of interests to receive the future notifications from the Partner OP. The Partner OP stores the Callback URL and indicated resources on successful acceptance of the periodic reporting of Application's events request from the Originating OP. The Originating OP can also use a query-based HTTP POST method to retrieve the Application-level events information.

2.6.2.1 Create Subscription for Applications Events Notification

The Originating OP uses the HTTP POST method to subscribe for the reporting of Application-level events associated with the federated edge Applications managed by the Partner OP.

The Partner OP on creating a successful subscription for the Application-level events notifications should respond with a 200 OK while for any failures it shall respond with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.2 Modify Subscription Information for Applications Events

The Originating OP uses the HTTP PATCH method to modify Application-level events subscription information for the Application-level events reporting associated to federated edge Applications managed by the Partner OP. The modification may include requests to enable or disable the notifications, change the notifications frequency etc.

The Partner OP on successfully updating the subscription information should respond with a 200 OK while for any failures it respond with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.3 Retrieve Subscription Information for Applications Events

The Originating OP uses the HTTP POST method to retrieve subscription information for the reporting of Application-level events reporting associated to federated edge Applications managed by the Partner OP.

The Partner OP on a successful retrieval of subscription information for the Application-level events should respond with a 200 OK while for any failures it responds with appropriate HTTP error codes as described for other procedures described in previous sections.

2.6.2.4 Remove Subscription for Applications Events Notifications

The Originating OP uses the HTTP DELETE method to request the Partner OP to remove the existing Application-level events notifications subscription and stop sending any further notifications.

The Partner OP on successfully removing the subscription for the Application-level events notifications should respond with a 200 OK while for any failures it responds with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.5 Add Applications To Applications Notifications Subscription

The Originating OP uses this procedure (HTTP POST) to add the Application-level events in context of an Application which the Partner OP should monitor and report to the Originating OP at the periodicity indicated during Application events subscription process.

The Partner OP on successfully adding the Applications to the Application-level event subscription context should respond with a 200 OK while for any failures it responds with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.6 Remove Applications From Applications Notification Context

The Originating OP uses this procedure (HTTP DELETE) to remove the Application event types in context of an Application and stop reporting of the given events to the Originating OP.

The Partner OP on successfully removing the Applications from the Application-level event subscription context should respond with a 200 OK while for any failures it responds with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.7 Retrieve Subscribed Applications Information From Notification Context

The Originating OP uses this procedure (HTTP GET) to retrieve the list of Applications currently subscribed for Application-level events notifications.

The Partner OP on successful outcome should responds with a 200 OK with the Applications information currently subscribed for different Application-level events while for any failures it responds with appropriate HTTP error codes as described for other procedures in previous sections.

2.6.2.8 Applications Events Reporting Notification

The Partner OP monitors the Applications events submitted by the Originating OP for the set of Applications. The Partner OP collects the Applications events for the given set of Applications and notifies the Originating OP via a HTTP POST method (HTTP Callback) to report the collected information.

The Partner OP may submit the Applications events reports at the periodicity agreed at the time of subscribing for the notification context creation process by the Originating OP. The reporting of collected Application event information may not be performed in real-time by the Partner OP.

2.6.2.9 Retrieve Application-level Events Information

The Originating OP uses the HTTP POST method to retrieve Application-level events reports available at the Partner OP.

The Partner OP on successful outcome should respond with a 200 OK with the Application-level event information while for any failures it responds with appropriate HTTP error codes as described for other procedures in previous sections.

2.7 Resource Control Policies Management Procedures

As described in GSMA PRD OPG.02 [1] the E/WBI should have the capabilities to share Edge Cloud and Network resource sharing policies information.

As an extension, the policies like an Application's policies may help an OP for decision making in some of the scenario e.g., target Edge Resource selection for the Application deployment and an Application's relocation considering the Application's sensitivity to a change of Edge Resources, required notification before relocation etc.

The resource control policies may be created by the administrative actions supported by an OP while the E/WBI allows to share such policies with the other OPs (i.e., Originating OP) who can then in turn associate these policies to federated Applications or to the Edge Resource infrastructure at the Partner OP.

Such policies can be categorized in two classes i.e.,

1. Application-level Policies
    - These policies enables control over Applications deployment behavior in the Partner OP environment for the federated Applications
2. Operation-level Policies
    - These policies enable Operators to control the infrastructure usage assigned by the Partner OP to the Originating OP

2.7.1 Application-level Policies Management Procedures

Application-level policies can be associated with the edge Applications to control the deployment behavior in Edge Resources managed by the Partner OP.

The E/WBI enables the Partner OP to share policy templates with rules and actions for a given policy created in their network while the Originating OP may register the Application-level policies using such policy templates from the Partner OP specifying the Application's attributes for the parameters defined in the policy templates.

Once the Application policies have been registered at the Partner OP, the Originating OP may request assignment of these policies to federated Applications managed by the Partner OP.

2.7.1.1 Subscribe For Application-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to create the Application-level policies subscription to enable policy based treatment at the Partner OP for federated Applications.

The Partner OP responds with a 200 OK with the Application-level policy subscription identifier if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.2 Retrieve Application-level Policy Templates

The Originating OP may query the Partner OP by a HTTP GET method to retrieve the Application-level policy templates supported by the Partner OP.

The Partner OP responds with a 200 OK with the Application-level policy template details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.3 Register Application-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to register the Application-level policies supported by the Partner OP. The registration request includes Application's attribute along with the policy thresholds to trigger the policy rules execution.

The Partner OP responds with a 200 OK with the registered Application-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.4 Apply Application-level Policies

The Originating OP may request the Partner OP by aHTTP POST method to apply the registered Application-level policies to one or more Applications in one or more Availability Zones.

The Partner OP responds with a 200 OK with the applied Application-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.5 Modify Application-level Policies

The Originating OP may request the Partner OP by HTTP PATCH method to modify the Application-level policies associated earlier with the federated Applications.

The Partner OP responds with a 200 OK with the updated Application-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.6 Remove Application-level Policies

The Originating OP may request the Partner OP by HTTP POST method to remove the Application-level policies associated earlier with the federated Applications.

The Partner OP responds with a 200 OK with the removed Application-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.1.7 Retrieve Application-level Policies

The Originating OP may query the Partner OP by HTTP GET method to retrieve the Application-level policies supported by the Partner OP.

The Partner OP responds with a 200 OK with the Application-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2 Operation-level Policies Management Procedures

Operation-level policies may provide capability to control the usage of Edge Resources managed by the Partner OP for a given Originating OP. These policies may provide capabilities e.g., to control optimizing resource allocation, setting up thresholds on resource utilization etc.

The Originating OP may determine one or more policies that should be applied by the Partner OP to allocated resources while deploying a federated Application in the Partner OP managed Edge Resources.

2.7.2.1 Subscribe For Operation-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to create the operation-level policies subscription to enable policy based treatment at the Partner OP for resources dedicated to the Originating OP in various Availability Zones or Regions.

The Partner OP responds with a 200 OK with the operation-level policy subscription identifier if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.2 Retrieve Operation-level Policy Templates

The Originating OP may query the Partner OP by a HTTP GET method to retrieve the operation-level policy templates supported by the Partner OP.

The Partner OP responds with a 200 OK with the operation-level policy template details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.3 Register Operation-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to create the operation-level policies supported by the Partner OP. The registration request includes the Edge Resources limits for various policy rules to determine the applicability of the policy to Edge Resources.

The Partner OP responds with a 200 OK with the operation-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.4 Apply Operation-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to apply the operation-level policies supported by the Partner OP.

The Partner OP responds with a 200 OK with the applied operation-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.5 Modify Operation-level Policies

The Originating OP may request the Partner OP by a HTTP PATCH method to modify the operation-level policies associated earlier with the Edge Resources at the Partner OP environment.

The Partner OP responds with a 200 OK with the updated operation-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.6 Remove Operation-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to remove the operation-level policies associated earlier with the federated Applications.

The Partner OP responds with a 200 OK with the removed operation-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

2.7.2.7 Retrieve Operation-level Policies

The Originating OP may request the Partner OP by a HTTP POST method to retrieve the operation-level policies supported by the Partner OP to one or more Availability Zones.

The Partner OP responds with a 200 OK with the operation-level policies details if the request is successful or else with the errors including the appropriate HTTP Status codes as described earlier for various requests.

3 OP East/West Bound APIs

This section provides the information on various APIs and associated parameters and data models for the procedures mentioned in the previous sections.

3.1 Generic East/West Bound Service APIs

The interface management APIs provides the capabilities to perform the handshake between the two OPs and share the Availability Zone(s) and resource information with the Partner OPs.

3.1.1 East/West Bound Interface Management - API

The interface management APIs provides the capabilities to perform the handshake between the two OPs and share the Availability Zone(s) and resource information with the Partner OPs.

The following subsections specify the resource methods supported by the resource as described in below section.

3.1.1.1 Introduction

Following table describes the HTTP Methods for the Federation resource.

Operation	HTTP Methods	Resource URI	Qualifier
Create Federation	POST	/operatorplatform/federation/v1/partner	M
Notify Federation Updates	POST	{federationNotificationDest}	M
Remove Federation	DELETE	/operatorplatform/federation/v1/{federationContextId} partner	M
Get Federation Meta Info	GET	/operatorplatform/federation/v1/{federationContextId}/partner	M
Update Federation	PATCH	/operatorplatform/federation/v1/{federationContextId}/partner	M
Get Service Capabilities	GET	/operatorplatform/federation/v1/{federationContextId}/partner/{service_type}	O
Get Federation Context Identifier	GET	/operatorplatform/federation/v1/fed-context-id	O
HealthCheck	GET	/operatorplatform/federation/v1/{federationContextId}/health	O
Renew Federation	POST	/operatorplatform/federation/v1/{federationContextId}/renew	O
Get Operator Platform Capabilities	GET	/operatorplatform/federation/v1/{federationContextId}/Capabilities	O

Table 1: E/WBI Interface Management APIs

3.1.1.2 POST Method: Create Federation

The POST method creates a new Federation relationship resource for a given OP.

This method shall support the request data structures, response data structures and response codes as specified in data model section.

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
origOPFederationId	O	0..1	Operators in Federation shall be governing the namespace and Operator identifier assigned to it. This parameter is deprecated from now on.
origOPCountryCode	C	1	MCC of the Originating OP (i.e., the OP sending the Federation create request).
origOPMobileNetworkCodes	C	1..N	List of MNCs where an Operator may have one or more network codes assigned

Parameter Name	P	Cardinality	Description
origOPFixedNetworkCodes	C	1..N	Need the identifiers to refer to fixed network Operators
initialDate	M	1	Date and time, time zone info of the Federation initiated by the Originating OP
federationNotificationDest	M	1	Contains the API endpoint to receive the notifications from the Partner OP.for any updates done by the Partner OP on this Federation

Table 2: Request Parameter for Create Federation Request

The following table describes the data structures supported by the POST Response Body on this resource for 200 OK.

Parameter Name	P	Cardinality	Description
partnerOPFederationId	O	0..1	Operators in Federation shall be governing the namespace and Operator identifier assigned to it. This parameter is deprecated from now on.
partnerOPCountryCode	M	1	Mobile Country Code of Operator sending the response.
partnerOPMobileNetworkCodes	C	1..N	Mobile Network Codes of Operator sending the response to Federation create request.
origOPFixedNetworkCodes	C	0..N	Fixed line network identifier
federationContextId	M	1	This identifier shall be provided by the Partner OP on successful verification and validation of the Federation create request. The identifier is the indicator of a successful Federation establishment between the two OP. This identifier shall be used in subsequent requests by the Originating OP
edgeDiscoveyServiceEndPoint	M	1	IP and Port of Edge Discovery Service URL of the Partner OP. This can also be a FQDN
offeredAvailabilityZones	O	0..N	List of Availability Zones a Partner OP is willing to share. The Partner OP may configure such information using system management interface
platformCaps	M	1	List of extended Capabilities e.g., HomeRouting, Service

Parameter Name	P	Cardinality	Description
			APIs, Anchoring as supported by the Partner OP.
federationRenewalDate	C	1	Date and time indicating the renew the Federation. Shall be earlier than the federationExpiryDate
federationExpiryDate	C	1	Date and time indicating the expiry of the Federation unless renewed by the Originating OP.
defaultSubscriptions	C	1	Set of default subscriptions created at the Partner OP to report the monitoring information
subscriptionsPeriodicity	O	1	Periodic notifications duration which the Partner OP may override from the Originating OP to share resource Info and event notifications
opsInfoExposureEndpoint	O	1	REST API endpoint to provide monitoring info, events subscription management
NOTE: partnerOPAvailabilityZones is a data type which has the following attributes: zoneId, geolocation, city, state, locality, edgeCount.			

Table 3: Response Parameter for Create Federation Request

The following table describes the header supported by the POST Response Body on this resource.

Name	Data Type	P	Cardinality	Description
Location	String	M	1	Contains the URI of the newly created resource i.e., /operatorplatform/federation/v1/partner/{federationContextId}

Table 4: Header parameter for Create Federation Response

The following table describes the data structures supported by the POST Response Body on this resource for non-200 response codes.

Parameter Name	P	Cardinality	Response codes	Description
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized access
problemDetails	C	1	404	Content Not Found

Parameter Name	P	Cardinality	Response codes	Description
problemDetails	C	1	409	Conflict. Federation already exists or state mismatch
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 5: Failure Responses for Create Federation Request

3.1.1.3 POST Method: Notify Federation Updates

POST HTTP method is used by the Partner OP towards the Originating OP to update the parameters associated to the existing Federation. The Partner OP sends an update request on the URI defined by the parameter ‘federationNotificationDest‘.

The following table describes the POST request body for updating existing Federation.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
objectType	M	1	Refers to the resource being modified for e.g., Federation status, Availability Zone status, Edge Discovery Service URL, network codes, Service API support etc.
operationType	M	1	Type of update for e.g., Change in status, add network code, update edge Discovery Service URL, Change in Service APIs capabilities etc.
modificationDate	M	1	Date and time of the Federation modification by the Partner OP
edgeDiscoverySvcEndPoint	O	0..1	Edge Discovery Service URL for UNI interface.
lcmSvcEndPoint	O	0..1	LCM Service URL for UNI interface
addMobileNetworkIds	O	1..N	List of MNCs to be added
removeMobileNetworkIds	O	1..N	List of MNCs to be removed
addFixedNetworkIds	O	1..N	List of fixed network codes to be added

Parameter Name	P	Cardinality	Description
removeFixedNetworkIds	O	1..N	List of fixed network codes to be removed
addZones	O	1..N	New Availability Zones to be added. List of 'availabilityZone'.
removeZones	O	1..N	List of zoneIds to be removed
zoneStatus	O	1	Availability status of Availability Zones
serviceAPICaps	O	1..N	Service APIs capability information

Table 6: Notify Federation updates request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Completion status of the POST request handling procedure at Originating OP
problemDetails	C	400	Bad Request. Parameters in the request has conflicting values, content have semantic error.
problemDetails	C	401	Unauthorized
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict. Federation does not exist
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 7: Notify Federation updates response parameters

3.1.1.4 DELETE Method: Remove Federation Relationship

The Originating OP shall use the DELETE method towards the Partner OP to terminate the existing Federation between them. This method supports the query parameters.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.

Table 8: Remove Federation request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
status	C	200	Federation removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	400	Unauthorized Access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict. Federation already being terminated
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 9: Remove Federation response parameters

3.1.1.5 GET Method: Get Federation Meta Information

The GET method supports the path parameters. The Originating OP can send the HTTP GET method to retrieve the Federation information at the Partner OP identified by federationContextId

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.

Table 10: Federation meta info request parameters

The following table describes the data structures supported by the GET Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
edgeDiscoveryServiceEndPoint	O	0..1	IP and Port of Edge Discovery Service URL of the Partner OP. This can also be a FQDN. E.g., “discovery.operator1.com” or IPv4Addr:Port (in dotted decimal notation).
offeredAvailabilityZones	O	0..N	List of Availability Zones a Partner OP is willing to share. The Partner OP may configure such information using system management interface

Parameter Name	P	Cardinality	Description
allowedMobileNetworkIds	O	1..N	List of mobile network codes where an Operator may have one or more network codes assigned
allowedFixedNetworkIds	O	1..N	List of Fixed network codes
lcmServiceEndPoint	O	0..1	IP and Port of LCM Service URL of the Partner OP. This can also be a FQDN.
platformCaps	M	1	List of extended capabilities e.g., HomeRouting, Service APIs, Anchoring as supported by the Partner OP.

Table 11: Federation meta info response parameters

The following table describes the HTTP codes supported by the GET Response on this resource.

Parameter Name	P	Cardinality	Response Codes	Description
Status	C	1	200	Federation meta-information request accepted
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 12: Response codes for Availability Zone meta-information Request

3.1.1.6 PATCH Method: Update Federation by Originating OP

The PATCH HTTP method is used by the Originating OP towards the Partner OP to update the parameters associated to the existing Federation. The Table 13 below describes the PATCH request body for updating existing Federation.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
objectType	M	1	Refers to the resource being modified for e.g. network codes etc.
operationType	M	1	Type of update for e.g., add or remove mobile network codes or fixed network codes.
modificationDate	M	1	Date and time of the Federation modification by the Partner OP
addMobileNetworkIds	O	1..N	List of MNCs to be added
removeMobileNetworkIds	O	1..N	List of MNCs to be removed
addFixedNetworkIds	O	1..N	List of fixed network codes to be added
removeFixedNetworkIds	O	1..N	List of fixed network codes to be removed

Table 13: Update Federation request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Modification accepted
problemDetails	C	400	Bad Request. Parameters in the request has conflicting values, content have semantic error.
problemDetails	C	401	Unauthorized
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict. Federation does not exist
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 14: Update Federation response parameters

3.1.1.7 GET Method: Get Service APIs Capabilities

The GET method for the retrieval of Service APIs capabilities supports the query string parameter.

Parameter Name	P	Cardinality	Description
serviceType	M	1	The query string parameter serviceType can contain values as “api_federation”.

Table 15: Query parameter for retrieving Service APIs capability information

The table below describes the data structures supported by the GET Response Body on this resource for the response code 200 OK.

Parameter Name	P	Cardinality	Description
serviceType	M	1	The parameter serviceType can contain value as “api_federation”.
serviceAPICaps	M	1..N	List of strings with Service API identifier e.g., “QualityOnDemand”, “NumberVerification”, “DeviceStatus”, “DeviceIdentifier” etc. Public IP addresses block info, MSISDN block info etc. managed by the Partner OP for UEs

Table 16: Retrieve Service APIs capability response

The table below describes the HTTP non-200 response codes supported by the GET Response on this resource.

Parameter Name	P	Cardinality	Response Codes	Description
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 17: Response codes for Service APIs capability Request

3.1.1.8 GET Method: Get Federation Context Identifier

The Originating OP uses the HTTP GET request to retrieve the existing Federation context identifier on the basis of the authorization information. The authorization token is provided to the Partner OP as HTTP header to retrieve the Federation context identifier.

Name	Data Type	P	Cardinality	Description
Authorization Token	String	M	1	Contains the oAuth 2.0 authorization token received from the Partner OP

Table 18: Retrieve Federation Context Identifier

The following table describes the data structures supported by the GET Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	Federation Context Identifier allocated by the Partner OP to the Originating OP

Table 19: Federation Context Identifier response parameters

The following table describes the HTTP codes supported by the GET Response on this resource.

Parameter Name	P	Cardinality	Response Codes	Description
Status	C	1	200	Federation context identifier
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 20: Response codes for Federation context identifier Request

3.1.1.9 GET Method: Check Health Status

The HTTP GET method from the Originating OP is sent to the Partner OP to query the health status of the Federation identified by the Federation context identifier.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.

Table 21: Health Status request parameters

The following table describes the data structures supported by the GET Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
federationHealthInfo	M	1	The object describes the Federation status at the Partner OP

Table 22: Federation Health check response parameters

The following table describes the HTTP codes supported by the GET Response on this resource.

Parameter Name	P	Cardinality	Response codes	Description
federationHealthInfo	C	1	200	Federation health status Info at the Partner OP
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 23: Response codes for Health Check Request

3.1.1.10 POST Method: Renew Federation

The HTTP POST method from the Originating is sent to the Partner OP to renew the Federation identified by the Federation context identifier.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.

Table 24: Health Status request parameters

The following table describes the data structures supported by the POST Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	The object describes the Federation status at the Partner OP
federationRenewalDate	C	1	Date and time indicating the renew the Federation.
federationExpiryDate	C	1	Date and time indicating the expiry of the Federation unless renewed by the Originating OP.

Table 25: Federation Renewal response parameters

The following table describes the HTTP codes supported by the POST Response on this resource.

Parameter Name	P	Cardinality	Response codes	Description
federationContextId	C	1	200	Federation context identifier and date and time for next expiry at the Partner OP
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 26: Response codes for Federation Renewal Request

3.1.1.11 GET Method: Get Operator Platform Capabilities

The GET method for the retrieval of the Partner OP capabilities and this method supports the query string parameter to retrieve specific capability.

Parameter Name	P	Cardinality	Description
capabilityType	O	1	The query string parameter can contain values indicated in platformCaps returned by Partner OP in Federation create response.

Table 27: Query parameter for retrieving Operator Platform capabilities information

The table below describes the data structures supported by the GET Response Body on this resource for the response code 200 OK.

Parameter Name	P	Cardinality	Description
capabilityType	M	1	The parameter capabilityType contains values or enums from “platformCaps”.
capabilityInfo	M	1	Includes capabilityID, SLOs assosiated to specific capabilityID

Table 28: Retrieve Operator Platform Capabilities Response

The table below describes the HTTP non-200 response codes supported by the GET Response on this resource.

Parameter Name	P	Cardinality	Response Codes	Description
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 29: Response codes for Operator Platform capabilities Request

3.1.1.12 Data Model

3.1.1.12.1 General

This subclause specifies the application data model supported by the E/WBI interface management API.

3.1.1.12.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

3.1.1.12.2.1 offeredAvailabilityZones

Following table describes information about the Availability Zones which the Partner OP offers to the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
offeredAvailabilityZones	Array(availabilityZone)	M	1..N	List of Availability Zone Ids

Table 30: Availability Zones meta information

3.1.1.12.2.2 availabilityZone

Following table describes the data elements of an Availability Zone.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Unique Identity of an Availability Zone
geolocationInfo	String	M	1	Latitude/Longitude of Availability Zone
geographyDetails	String	O	1	Details about cities or state covered by the edge. Details about the type of locality for e.g., rural, urban, industrial etc. This information is defined in human readable form.

Table 31: Availability Zone location parameters

3.1.1.12.2.3 edgeDiscoveryServiceEndPoint

Attribute Name	Data Type	P	Cardinality	Description
serviceURL	String	M	1	FQDN or Public IP Address of the Edge Discovery Service
Port	Int	M	1	Port number of the Edge Discovery Service where UCs can send requests to over UNI

Table 32: Edge Discovery Service Endpoint

3.1.1.12.2.4 mncChangeInfo

Following table describes the network code update structure to notify change in supported network codes by the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
operationType	Enum	M	1	Whether the network code being added or removed
networkCodes	Array(String)	M	1..N	The list of network codes being added or removed

Table 33: Availability Zones meta information

3.1.1.12.2.5 platformCaps

The table below describes the capabilities that a Partner OP supports.

Attribute Name	Data Type	P	Cardinality	Description
platformCaps	Array(String)	M	1	The list of Operator Platform capabilities that a Partner OP supports e.g. homeRouting, Service API Federation, fault and event management, resource monitoring, network capability exposure, etc.

Table 34: Operator Platform Capabilities Information

3.1.1.12.2.6 ProblemDetails

Attribute Name	Data Type	P	Cardinality	Description
title	String	M	1	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem.
detail	String	O	0..1	A human-readable explanation specific to this occurrence of the problem.
cause	String	O	0..1	A machine-readable application error cause specific to this occurrence of the problem This IE should be present and provide application-related error information, if available.
invalidParams	array(InvalidParam)	O	1..N	Description of invalid parameters, for a request rejected due to invalid parameters.

Table 35: Response body for error responses

3.1.1.12.2.7 InvalidParam

Attribute Name	Data Type	P	Cardinality	Description
param	String	M	1	Parameter name
reason	String	O	0..1	A human-readable reason

Table 36: InvalidParam

3.1.1.12.2.8 zoneStatus

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone Identifier
Status	String	M	1	Availability Status for the Availability Zone.

Table 37: InvalidParam

3.1.1.12.2.9 apiRoutingInfo

The table below describes the Service API routing information that a Partner OP provides to the Leading OP.

Attribute Name	Data Type	P	Cardinality	Description
publicIPAddrRanges	Array(String)	C	1	List of Public IP addresses blocks/CIDRs ranges that a Partner OP manages for the its UE subscriptions
publicIdentifiersBlock	Array(String)	C	1	List of MSISDN/GPSI blocks etc.

Table 38: Partner OP API Routing Information

3.1.1.12.2.10 serviceAPICaps

Attribute Name	Data Type	P	Cardinality	Description
serviceAPINames	Array(String)	M	1	List of the Service APIs names that the Partner OP accepts to provide to the Leading OP e.g., “quality_on_demand”, “device_location” etc.

Attribute Name	Data Type	P	Cardinality	Description
apiRoutingInfo	Object	M	1	Partner OP provides the API routing info which the Leading OP can use to determine the Partner OP e.g., public IP address ranges, block of MSISDNs etc.

Table 39: serviceAPICaps

3.1.1.12.2.11 subscriptionsPeriodicity

Attribute Name	Data Type	P	Cardinality	Description
edgeResMonitorDuration	Object	M	1	Periodicity in hours and minutes for reporting Edge Resource monitoring information.
appResMonitorDuration	Object	M	1	Periodicity in hours and minutes for reporting applications resource monitoring information

Table 40: subscriptionsPeriodicity

3.1.1.12.2.12 federationMonitoringDest

Attribute Name	Data Type	P	Cardinality	Description
Status	enum	M	1	Federation Status
numActiveAlarms	String	C	1	If subscribed, number of active alarms at Partner OP
numApplications	String	C	1	Number of applications
numOfAcceptedZones	String	C	1	Number of Availability Zone accepted by the Originating OP
federationStartTime	String	M	1	Date-time at which Federation established

Table 41: federationHealthInfo

3.1.1.12.2.13 networkCapabilitiesInfo

Attribute Name	Data Type	P	Cardinality	Description
networkCapabilitiesInfo	Object	M	1..N	Details of one or more network capabilities offered

Table 42: networkCapabilitiesInfo

3.1.1.12.2.14 dynamicQoSCap

Attribute Name	Data Type	P	Cardinality	Description
capabilityID	String	M	1	Enum for describing Dynamic QoS application capability to influence QoS for application traffic towards edge workloads
qosProfileSet	Array(String)	M	1..N	SLOs applicable to a network capability. Set of one or more 5G QoS Identifier (5QI or 4G QCI)

Table 43: dynamicQoSCap

3.1.1.12.2.15 userPlaneMgmtEvtCap

Attribute Name	Data Type	P	Cardinality	Description
capabilityID	String	M	1	Enum for describing capability to detect user plane or UPF/PGW change event for UE and take actions like Application relocation
maxUserPlaneLatency	String	M	1	SLOs applicable to a network capability. Indicates the maximum user plane latency in units of milliseconds to decide whether edge relocation is needed to ascertain latency remain in this range

Table 44: userPlaneMgmtEvtCap

3.1.1.12.2.16 locationRetrievalCap

Attribute Name	Data Type	P	Cardinality	Description
capabilityID	String	M	1	Enum for describing an OP capability to Retrieve from mobile network the location of an UE
locationAccuracy	Array(string)	M	1..N	SLOs applicable to a network capability. It represents the capability of an OP to retrieve UE location with an accuracy from cell level to PLMN level
locationType	Array(string)	M	1..N	It reprsents the capability of an OP to retrieve Current Location, Initial Location or Last known Location of an UE from the mobile network

Table 45: locationRetrievalCap

3.1.1.12.2.17 deviceConnStatusChangeCap

Attribute Name	Data Type	P	Cardinality	Description
capabilityID	String	M	1	Enum for describing capbility of OP to detect data connectivity of an UE and take actions like shutdown applications based on application defined policies
maxiDetectionTime	string	M	1	Defines the maximum period of time after which the UE is considered to be unreachable

Table 46: deviceConnStatusChangeCap

3.1.1.12.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

3.1.1.12.3.1 Simple data types

Attribute Name	Data Type	Description
federationContextId	String	Federation relationship Identifier generated by the Partner OP
initialDate	String	date/time value as a string in ISO 8601 format., “ 2018-12-10T13:45:00.000Z”
partnerOPFederationId	String	Unique public identifier for the Partner OP This parameter is depericated nowonwards.
partnerOPCountryCode	String	Mobile Country Code (MCC) of the Partner OP
partnerOPMobileNetworkCodes	Array(String)	Mobile Network Codes (MNCs) of the Partner OP
partnerOPFixedNetworkCodes	Array(String)	Fixed Network Codes of the Partner OP
origOPFederationId	String	Unique public identifier for the Originating OP This parameter is deprecated from now on.
origOPCountryCode	String	Mobile Country Code (MCC) of the Originating OP
origOPMobileNetworkCodes	Array(String)	Mobile Network Codes (MNCs) of the Originating OP. MNCs are 2- or 3-digits codes with each digit is from the set {0,9}
origOPFixedNetworkCodes	Array(String)	Fixed Network Codes of the Originating OP
zoneId	String	Identifier for an Availability Zone
serviceType	String	Supported value “api_federation”
SubscribeServiceAPIs	String	Can be “QualityOnDemand”, “NumberVerification”, “DeviceStatus” etc.
initialDate	String	Date and Time zone Info as defined in RFC 3339, section 5.6, of the the Originating OP
federationExpiryDuration	String	Date and Time zone info as defined in RFC 3339, section 5.6, at which the Federation will be terminated by the Partner OP

Table 47: E/WBI Interface Management Simple Datatype table

3.1.1.12.3.2 Enumeration: objectType

The enumeration updateType represents the attribute being updated by the Partner OP on existing Federation.

Enumeration value	Description
FEDERATION	Change in status of Federation relationship

Enumeration value	Description
ZONES	Change in the availability status of an Availability Zone
EDGE_DISCOVERY_SERVICE	Edge Discovery Service endpoints are modified
LCM_SERVICE	LCM Service endpoints are modified
MOBILE_NETWORK_CODES	Addition or removal of mobile network code
FIXED_NETWORK_CODES	Addition or removal of fixed network codes
SERVICE_APIS	Addition or removal of Service API

Table 48: Federation Modification Parameter types

3.1.1.12.3.3 Enumeration: operationType

The enumeration operationType represents the if the network codes are being added or removed by the Partner OP on existing Federation.

Enumeration value	Description
STATUS	Status of the resource has changed
UPDATE	Update of an object type
ADD	Addition of resources of type indicated by objectType to the Originating OP
REMOVE	Removal of resources of type indicated by objectType by the Partner OP

Table 49: Operations type for network code change

3.1.1.12.3.4 Enumeration: Status

Enumeration value	Description
FAILED	Resource is in failure state
TEMPORARY_FAILURE	Temporary failure for the resource
AVAILABLE	Resource is available
LOCKED	Resource is locked and is no longer accessible

Table 50: Allowed status values

3.1.1.12.3.5 Enumeration: CapabilityID

Enumeration value	Description
NW_CAP_CONN_STATE_CHANGE	Capability to retrieve UE connectivity status from 5GC
NW_CAP_LOCATION_RETRIEVAL	Capability to retrieve UE network location from 5GC
NW_CAP_USERPLANE_MGMT_EVENTS	Capability to retrieve UE User plane change events for application relocation
NW_CAP_DYNAMIC_QOS	Capability to influence edge application traffic QoS dynamically via SBI-NR (NEF/SCEF)

Table 51: Allowed Network Capability IDs

3.1.1.12.3.6 Enumeration: platformCap

Enumeration value	Description
homeRouting	Capability to detect and route an edge applications traffic to the Home OP for non-roaming Subscribers
Anchoring	Capability to detect and route an edge applications traffic to the Home OP for roaming Subscribers
serviceAPIs	Capability to Service API Federation
faultMgmt	Capability to report fault management events to the Originating OP
eventMgmt	Capability to report events detected as per the event criterion configured by the Originating OP
resourceMonitor	Capability to report resource monitoring information to the Originating OP
networkEventMgmt	Capability to report network events in context of federated applications to the Originating OP
policyMgmt	Capability to manage and perform actions defined as resource control policies for Edge Resources

Table 52: Operator Platform Capabilities

3.1.2 Availability Zone Information Synchronization Service – API

The APIs for Availability Zone Information Synchronisation Service are used to share and update specific information on the Availability Zone corresponding to an OP's Edge Resources provided to another.

3.1.2.1 Introduction

Following table describes the APIs for Availability Zone resources synchronization services.

Operations	HTTP Method	Resource URI	Qualifier
Availability Zone Subscribe	POST	/operatorplatform/federation/v1/{federationContextId}/zones	M
Availability Zone Unsubscribe	DELETE	/operatorplatform/federation/v1/{federationContextId}/zones/{zoneId}	M
View Availability Zone Information	GET	/operatorplatform/federation/v1/{federationContextId}/zones/{zoneId}	M
Notify Availability Zone Information	POST	{ availZoneNotifLink}	M

Table 53: Availability Zone Synchronization APIs

3.1.2.2 POST Method: Availability Zone Subscribe

The Availability Zone subscribe POST request contains the following parameters towards the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
acceptedAvailabilityZones	M	1..N	Accepted list of one or more Availability Zones selected from the offered list of Availability Zones provided by the Partner OP which the Originating OP intends to use.
availZoneNotifLink	M	1	An Availability Zone info notification URL which shall be used by the Partner OP to inform the about any changes to Availability Zone information e.g., resource quota updates, addition of new Availability Zones etc. asynchronously

Table 54: Availability Zones subscription request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
acceptedZoneResourceInfo	M	1	Available Availability Zone Resource information provided by the Partner OP for accepted Availability Zone IDs by Originating OP. It includes zoneId, guaranteed Resources and upper Limit Quota (E.g., vCPU, Memory, Storage, GPU etc.)

Table 55: Availability Zones subscription response parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
Status	C	200	Zone subscribed
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 56: Availability Zones subscription response parameters

3.1.2.3 Availability Zone Unsubscribe : DELETE Method

Following table provides parameters which an Originating OP sends to the Partner OP in Availability Zone unsubscribe request to relinquish Availability Zone(s) and associated resources for indicated Availability Zones which the Originating OP may have been using in the Partner OP footprint.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to Partner OP to identify the existing Federation relationship.
zoneId	M	1	Availability Zone identifier of Partner OP. The Partner OP shall deregister the indicated Availability Zone and may reclaim the resources.

Table 57: Availability Zones Unsubscribe request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
Status	C	200	Zone Unsubscribed
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 58: Availability Zones Unsubscribe response parameters

3.1.2.4 GET Method: View Availability Zone Information

Following table provides parameters which an Originating OP sends to a Partner OP in a view Availability Zone information request for the indicated Availability Zones which the Originating OP may have been already using.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
zoneId	M	1	Availability Zone identifier of partner Operator Availability Zone. The Partner OP shall deregister the indicated Availability Zone.

Table 59: Availability Zones information request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
acceptedZoneResourceInfo	C	200	Available Availability Zone Resource information provided by Partner OP for accepted Availability Zone IDs by Originating OP. It includes zoneId, guaranteed Resources and upper Limit Quota (E.g., vCPU, Memory, Storage, GPU etc.)
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 60: Availability Zones information response parameters

3.1.2.5 POST Method: Notify Availability Zone Information

The Availability Zone notification request sent by the Partner OP contains the following parameters towards the Originating OP sent over a different HTTP session on the notification URL of the Originating OP to provide updates to existing resources or Availability Zone information. This can further be periodically sent to update the availability of resources.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
zoneId	M	1	Identifier of the Availability Zone
zoneResUpdInfo	M	1	Available Availability Zone Resource information provided by the Partner OP to Originating OP. It may include zoneId, guaranteed Resources and upper Limit Quota (E.g., vCPU, Memory, Storage, GPU etc.)

Table 61: Availability Zones notify request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Zone info notification acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 62: Availability Zones async response parameters

3.1.2.6 Data Model

3.1.2.6.1 General

This section provides the data types for the Availability Zone and resource management.

Parameter Name	Clause Defined	Description
acceptedZoneResourceInfo		Available Availability Zone Resource information provided by the Partner OP for accepted Availability Zone IDs to Originating OP. It includes zoneId, guaranteed Resources and upper Limit Quota (E.g., vCPU, Memory, Storage, GPU etc.)
partnerAvailabilityZones		List of Availability Zones a Partner OP is willing to share. Partner may configure such information using system management interface

Table 63: Data structures used in Availability Zones management services

3.1.2.6.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

3.1.2.6.2.1 acceptedZonesResourceInfo

Following table describes information about the Availability Zones which the Originating OP has accepted from the Partner OP offer.

Attribute Name	Data Type	P	Cardinality	Description
acceptedZoneResourceInfo	Array (ZoneRegisteredData)	M	1..N	Partner Edge Resource information available for applications consumptions

Table 64: List of Availability Zones with offered resources

3.1.2.6.2.2 ZoneRegisteredData

The Availability Zone resource information represents the computing resources which an OP can offer to the Application Providers of Partner OP from an Availability Zone.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone identifier to refer to an Availability Zone
reservedComputeResources	Array (computeResourceInfo)	M	1..N	Resources exclusively reserved for a Partner OP
computeResourceQuotaLimits	Array (computeResourceInfo)	M	1..N	Max quota on Resources that an OP may allow to Partner OP over reserved resources if available

Attribute Name	Data Type	P	Cardinality	Description
flavoursSupported	Array(compute Flavour)	M	1..N	Edge Resources flavours are set of OP defined Edge Resources combination which a Partner OP supports and offers to Application Providers to be link them to applications for runtime resource requirements
networkResources	Array(network ResourceInfo)	O	1..N	Type of networks supported by the partner Availability Zone
zoneServiceLevelObjsInfo	Object	O	1	Availability Zone specific Service Level Objectives and the supported values e.g., Latency (msec), Jitter (msec), badwidth (Mbps) etc.

Table 65: Availability Zones information data structure

3.1.2.6.2.3 computeResourceInfo

Edge Resources indicates the resource profile applicable for a particular Central Processing Unit (CPU) architecture.

Attribute Name	Data Type	P	Cardinality	Description
cpuArchType	Enum	M	1	CPU instruction set architecture (ISA). E.g., Intel, ARM etc.
numCPU	Integer	M	1	Total number of Virtual CPUs (vCPUs)
memory	Long	M	1	Total physical memory (Random Access Memory (RAM)) for given ISA type (in Mbytes)
diskStorage	Long	M	1	Total storage (RAM) for workloads for given ISA type (in GB)
gpuInfo	Array(gpuInfo)	O	0..1	Total Graphical Processing Unit (GPU) for workloads for given ISA type
FPGA	Int	O	0..1	Total FPGA for workloads for given ISA type
vpu	Int	O	0..1	Total VPUs (Visual Processing Units) for workloads for given ISA type
hugepages	Array(hugepageInfo)	O	1..N	Huge pages for workload for a given ISA type
cpuExclusivity	Boolean	O	0..1	Support for exclusive CPUs

Table 66: Availability Zone Edge Resource information

3.1.2.6.2.4 gpuInfo

GPU resources indicates the resource profile applicable for a particular CPU architecture.

Attribute Name	Data Type	P	Cardinality	Description
gpuVendorType	Enum	M	1	GPU vendor name e.g., NVIDIA, AMD etc.
gpuModeName	String	M	1	Model name corresponding to vendorType may include info e.g., for NVIDIA, model name could be “Tesla M60”, “Tesla V100”,
gpuMemory	Int	M	1	GPU memory in GB
numGPU	Int	M	1	Number of GPU of a given model

Table 67: GPU resources data model

3.1.2.6.2.5 computeFlavour

Compute flavours indicate templates associated to the computing capabilities of the application runtime environment in an OP’s Edge Resources.

Attribute Name	Data Type	P	Cardinality	Description
flavourId	Int	M	1	An identifier to refer to the combination of Edge Resource configuration as indicated by the other attributes in this table
cpuArchType	Enum	M	1	CPU Instruction Set Architecture (ISA) E.g., Intel, Arm etc.
supportedOSTypes	Array(operatingSystemInfo)	M	1..N	A list of operating systems which a flavour configuration can support e.g., RHEL Linux, Ubuntu 18.04 LTS, MS Windows 2012 R2, macOS
numCPU	Int	M	1	Number of CPU for a given flavour
memorySize	Int	M	1	RAM size for a given flavour
storageSize	Int	M	1	Amount of disk space (GB) to use for the root (/) partition.
gpuInfo	Array(gpuResourceInfo)	O	0..1	Total GPU for workloads for given ISA type
vpuInfo	Integer	O	0..1	Number of Intel VPUs available
hugepages	Array(hugePageInfo)	O	0..1	Hugepages supported on the Availability Zone
cpuExclusivity	Boolean	O	1	If the Availability Zone supports exclusive allocation of Intel CPUs.

Table 68: Compute flavour for Virtual Machines

3.1.2.6.2.6 operatingSystemInfo

The following table provides the information about the operating systems which may be supported by OP.

Attribute Name	Data Type	P	Cardinality	Description
osAddrSize	Enum	M	1	Provides machine architecture e.g., x86_64, x86
distroType	Enum	M	1	e.g., RHEL, Debian, Ubuntu etc.
versionInfo	Enum	M	1	Provides OS version information e.g., RHEL 8, Debian 11, Ubuntu 22.04 LTS etc.
licenseType	Enum	M	1	License type may include “on-Demand”, “Free” etc.

Table 69: Operating system information

3.1.2.6.2.7 networkResourceInfo

Attribute Name	Data Type	P	Cardinality	Description
egressBandWidth	Integer	M	1	Max dl throughput that this edge can offer. It is defined in Mbps.
dedicatedNIC	Integer	M	1	Number of Network Interface Cards (NICs) which can be dedicatedly assigned to application pods on isolated networks. This includes virtual as well physical NICs
supportSriov	Boolean	M	1	If the Availability Zone supports Single Root Input Output Virtualisation (SRIOV) based networking or not.
supportDPDK	Boolean	M	1	If the Availability Zone supports Data Plane Development Kit (DPDK)-enabled userspace networking or not.

Table 70: Operating system information

3.1.2.6.2.8 hugePageInfo

Attribute Name	Data Type	P	Cardinality	Description
pageSize	Enum	M	1	Size of hugepage
number	Integer	M	1	Total number of hugepages

Table 71: GPU resources data model

3.1.2.6.2.9 zoneResUpdInfo

Attribute Name	Data Type	P	Cardinality	Description
availableCompResources	Array (computeResourceInfo)	M	1..N	Resources exclusively reserved for a Partner OP
availableNetResources	Array(networkResourceInfo)	O	1..N	Type of networks supported by the partner Availability Zone

Table 72: Definition of zoneResUpdInfo

3.1.2.6.2.10 zoneServiceLevelObjsInfo

Attribute Name	Data Type	P	Cardinality	Description
latencyRanges	String	O	1	The time for data/packet to reach from UC to edge application. It represent minimum latency that may exist between UCs and edge apps in this Availability Zone but it can be higher in actual
JitterRanges	String	O	1	The packet delay variation between UC and edge application. It indicates minimum jitter that apps in this Availability Zone may observe but can be higher in actual
throughput	String	O	1	It is a measure of the actual amount of data that is being sent over a network per unit of time and indicates maximum supported value for an Availability Zone

Table 73: Definition of zoneServiceLevelObjsInfo

3.1.2.6.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

3.1.2.6.3.1 Enumeration: computeAccel

The enumeration computeAccel represents the hardware acceleration supported.

Enumeration value	Description
HW_ACCEL_ GPU	GPU as accelerator
HW_ACCEL_ FPGA	FPGA as accelerator

Table 74: Instruction Set Architecture types

3.1.2.6.3.2 Enumeration: cpuArchType

The enumeration cpuArchType represents the Instruction Set Architecture (ISA) for CPU.

Enumeration value	Description
ISA_X86_64	Intel x86 ISA (CISC)
ISA_ARM_64	ARMv8 ISA (RISC)

Table 75: Instruction Set Architecture types

3.1.2.6.3.3 Enumeration: gpuVendorType

The enumeration gpuVendorType represents the GPU providers.

Enumeration value	Description
GPU_PROVIDER_NVIDIA	Nvidia GPUs for applications
GPU_PROVIDER_AMD	AMD GPUs for applications

Table 76 : GPU Providers types

3.1.2.6.3.4 Enumeration: versionInfo

The enumeration versionInfo represents the Operating System (OS) which may be supported by OP.

Enumeration value	Description
OS_VERSION_UBUNTU_2204_LTS	Refers to Ubuntu 22.04 LTS Linux operating system
OS_VERSION_RHEL_8	Refers to Red Hat Enterprise Linux 8 operating system
OS_VERSION_RHEL_7	Refers to Red Hat Enterprise Linux 7 operating system
OS_VERSION_ DEBIAN_11	Refers to Debian Linux 11 operating system
OS_VERSION_COREOS_STABLE	Refers to Fedora CoreOS Linux Stable operating system

Table 77 : Operating system version info

3.1.2.6.3.5 Enumeration: licenseType

The enumeration licenseType represents the license model which may be supported by OP and can be exposed over NBI.

Enumeration value	Description
OS_LICENSE_TYPE_FREE	Refer to free license and is the default option
OS_LICENSE_TYPE_ON_DEMAND	Refer to on-demand license which may be required with certain OS(s) which require mandatory license to deploy the operating system in virtual environment

Table 78 : Operating system version info

3.1.2.6.3.6 Enumeration: hugePageSize

Enumeration value	Description
HUGE_PAGE_2MB	Refer to a hugepage of 2 Megabytes
HUGE_PAGE_4MB	Refer to a hugepage of 4 Megabytes
HUGE_PAGE_1GB	Refer to a hugepage of 1 Gigabyte

Table 79 : Operating system version info

4 Application Service APIs

The interface management APIs provides the capabilities to perform the edge application management functions and other GSMA PRD OPG.02 [1] specified services e.g., network slicing etc. with the Partner OPs.

4.1 Edge Service APIs

This section provides the details of the edge centric services as part of the Operator Platform.

4.1.1 Application Artefacts Management - APIs

Application artefact management APIs enables an OP to share application component descriptors information with the Partner OP. The Application Providers via NBI interface submits artefacts information and link artefacts with their edge applications. Leading OP based on Application Provider intent can share the artefacts with the Partner OP over E/WBI interface.

4.1.1.1 Introduction

Following table describes the supported operations and resource URIs for artefacts and file upload management.

Operations	HTTP Method	Resource URI	Qualifier
Onboard Artefact	POST	/operatorplatform/federation/v1/{federationContextId}/artefact	M
Remove Artefact	DELETE	/operatorplatform/federation/v1/{federationContextId}/artefact/{artefactId}	M
View Artefact	GET	/operatorplatform/federation/v1/{federationContextId}/artefact/{artefactId}	M
Upload File	POST	/operatorplatform/federation/v1/{federationContextId}/files	M
Remove File	DELETE	/operatorplatform/federation/v1/{federationContextId}/files/{fileId}	M
View File Info	GET	/operatorplatform/federation/v1/{federationContextId}/files/{fileId}	M

Table 80: Artefacts Management APIs

4.1.1.2 POST Method: Onboard Artefact

The following table describes the data structures supported by the POST Request Body on this resource. This method is used for submitting the artefacts as provided by the Application Providers over NBI and contains the application component descriptors which lays out the component images, connectivity to UEs, resource requirements etc. The application component descriptors also contain references to the image files submitted by the Application Providers over NBI to be used with the components.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
artefactId	M	1	Identifier unique within a Federation context to distinguish different artefacts
appProviderId	M	1	A unique Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider on Leading OP NBI
artefactName	M	1	Name of the artefact
artefactDescription	O	1	Brief description of the artefact by the Application Provider
artefactVersionInfo	M	1	Artefact version information
artefactVirtType	M	1	Indicate if the artefact refers to a containerized or VM type workload descriptor
artefactDescriptorType	M	1	Descriptor type associated with the artefactType refers to a descriptor e.g., Helm, Terraform, ComponentSpec etc. Helm Charts or Terraform scripts files can be uploaded to OP managed repo or can be pulled from external repo e.g., Github, Helm.sh etc. ContainerSpec schema is proposed as part of this document to deploy containerized workloads to OP managed Edge Resources.

Parameter Name	P	Cardinality	Description
artefactRepoLocation	C	1	Artefact image repository location URL and access credentials e.g., Github, local OP repo, bitnami etc. from which given artefacts like charts, Terraform scripts etc. can be retrieved. Artefacts can also be uploaded to OP managed local repo and can be associated to application components. Application Providers may be able to upload artefacts over NBI which can be referenced from artefactDescriptorFile(s), and an OP shall also submit them to the Partner OP over E/WBI if requested by Application Provider
artefactDescriptorFileFormat	C	1	Artefacts like Helm charts or Terraform scripts may need compressed format while ContainerSpec can be plane text file (YAML Ain't Markup Language (YAML) format)
componentSpec	O	1..N	A component specification to define the image, meta info and resource requirements
artefactFile	O	1	Actual file embedded in the request.

Table 81: Onboard Artefact request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Artefacts uploaded successfully at Partner OP
N/A	C	202	Artefacts Onboarding request accepted by Partner OP
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error

Parameter Name	P	Response Codes	Description
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 82: Onboard Artefact response parameters

4.1.1.3 DELETE Method: Remove Artefact

The following table describes the data structures supported by the DELETE Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
artefactId	M	1	Identifier unique within an appProviderId to distinguish different artefacts

Table 83: Remove Artefact request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Code	Description
Status	C	200	Artefact deleted successful
Status	C	202	Artefact deletion request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 84: Remove Artefact response parameters

4.1.1.4 GET Method: View Artefact Information

The following table describes the data structures supported by the GET Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
artefactId	M	1	Identifier unique within an appProviderId to distinguish different artefacts

Table 85: View Artefact request parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
artefactId	M	1	Identifier unique within an appProviderId to distinguish different artefacts
appProviderId	M	1	Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider
artefactVersionInfo	M	1	Artefact version information
artefactName	M	1	Name of the artefact
artefactDescription	O	1	Brief description of the artefact by the Application Provider
artefactVersionInfo	M	1	Artefact version information
artefactVirtType	M	1	Indicate if the artefact refers to a containerized or VM type workload descriptor
artefactDescriptorType	M	1	Descriptor type associated with the artefactType refers to a descriptor e.g., Helm, Terraform, ContainerSpec etc. Helm Charts or Terraform scripts files can be uploaded to OP managed repo or can be pulled from external repo e.g., Github, Helm.sh etc. ContainerSpec schema is proposed as part of this document to deploy containerized workloads to OP managed Edge Resources.

Parameter Name	P	Cardinality	Description
artefactRepoLocation	C	1	Artefact image repository location URL and access credentials e.g., Github, local OP repo, bitnami etc. from which given artefacts like charts, Terraform scripts etc. can be retrieved. To refer to OP local repo, Application Provider can provide artefacts over NBI contained in file associated to artefactDescriptor to submit the a
artefactDescriptorFileFormat	C	1	Artefacts like Helm charts or Terraform scripts may need compressed format while ContainerSpec can be plane text file (YAML format)

Table 86: View Artefact response parameters

The following table describes the data structures supported by the GET Request Body on this resource for non-200 HTTP codes.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 87: Non-200 Response Codes for View Artefact Response

4.1.1.5 POST Method: Upload File

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
fileId	M	1	Identifier unique within a Federation context to distinguish different artefacts

Parameter Name	P	Cardinality	Description
appProviderId	M	1	A unique Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider
fileName	M	1	Name of the file provided by the Application Provider on NBI. The NBI may provide capabilities to upload files from local filesystems from where NBI is accessed
fileDescription	O	1	Brief description of the file by the Application Provider
fileVersionInfo	M	1	File version information
fileType	M	1	Indicate if the file is Container image or VM image (QCOW2)
imgOSType	M	1	Base OS for the image. Currently only “Linux” is supported
imgInsSetArch	M	1	“x86_64”, “arm64”
file	C	1	Binary Images of application components (e.g., container images) which can be referenced from the files indicated by artefactDescriptor (E.g., Helm charts)
repoLocation	C	1	File Repository location information and same as artefactRepoLocation parameter as defined in artefact onboarding API

Table 88: Upload File request Parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
N/A	C	200	File uploaded successfully
N/A	C	202	File upload request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	415	Unsupported Media Type
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 89: Upload File response Parameters

4.1.1.6 DELETE Method: Remove Upload File

The following table describes the data structures supported by the DELETE Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
fileId	M	1	Identifier unique within an appProviderId to distinguish different fileId

Table 90: Remove File request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	File deleted successfully
N/A	C	202	File delete request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 91: Remove File response parameters

4.1.1.7 GET Method: View File Information

The following table describes the data structures supported by the GET Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
fileId	M	1	Identifier unique within an appProviderId to distinguish different fileId

Table 92: View File request parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
fileDetails	C	200	File Details
problemDetails		400	Bad request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails		409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
N/A problemDetails	C	503	Service Unavailable.
problemDetails		520	Server Returned an Unknown error

Table 93: View file error response

4.1.1.8 Data Model

4.1.1.8.1 General

This subclause specifies the application data model supported by the Artefacts Management API.

4.1.1.8.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.1.1.8.2.1 artefactDescriptor

Following table describes the artefactDescriptor which defines schema of an application component. Application component may refer to an artefactDescriptor in form of e.g., Helm Chart, Terraform Script, ContainerSpec etc. The descriptors to be supported by the two OP can be extended based on supported capabilities.

Attribute Name	Data Type	P	Cardinality	Description
helmChartRootDir	Compressed File	C	1	Zip file containing the Helm Chart directories and files
terraformScript	CompressedFile	C	1	Zip file containing terraform scripts
componentSpec	Object	C	1	A containerized component specification to define the image, meta info and resource requirements

Table 94: artefactDescriptor

4.1.1.8.2.2 componentSpec

Attribute Name	Data Type	P	Cardinality	Description
componentName	String	M	1	Application Provider defined name of the container
OSType	Enum	M	1	Base OS for the container. Currently only “Linux” is supported
cpuInstSetArch	Enum	M	1	A list of OP supported ISAs e.g., “x86_64”, “arm64” etc.
imagePath	String	M	1	File identifier as used in upload file API
numOfInstances	Int	M	1	Number of container instances to be launched
restartPolicy	Enum	O	1	Container restart policy “Always” or “Never” defines the action to be taken on container failure
commandLineParams	Object	O	1	Any input parameters to passed to component instance during instantiation
exposedInterfaces	Array (ExposedInterface)	M	1..N	List of interfaces having public visibility exposed by the application component. It could be combination of container port and IP protocol (TCP, UDP) and/or upstream HTTP root URL etc.
computeResourceProfile	Object	M		Refers to the Edge Resources required for the container e.g., CPU, RAM, GPU etc.
compEnvParams	Array (compEnvParameter)	O	0..N	Environ variable are key value pairs to provide Application Provider input parameters to be passed to container process during container process creation
persistentVolumes	Array (persistentVolume)	O	0..1	The ephemeral volume a container process may need to temporary store internal data

Table 95: componentSpec

4.1.1.8.2.3 commandLineParams

Attribute Name	Data Type	P	Cardinality	Description
command	Array(String)	M	1	This overrides the command operation of the container file while running container inside a pod
commandArgs	Array(String)	O	1	These arguments will be added while running containers

Table 96: Command line parameters for application component

4.1.1.8.2.4 exposedInterface

Attribute Name	Data Type	P	Cardinality	Description
interfaceId	String	M	1	defines the unique identifier/name of the component's API endpoint. It is a logical API endpoint and will be used to provide a session handle by an Software Development Kit (SDK).
commPort	Integer	M	1	Defines the internal port value for the application component to exposed to UCs. OP may generate a dynamic port towards the UCs corresponding to this internal port and forward the client traffic from dynamic port to containerPort.
commProtocol	Enum	M	1	Defines the IP transport communication protocol i.e., TCP, UDP
visibilityType	Enum	M	1	defines whether the interface is exposed to outer world or not i.e., external, or internal. If this is set to "external", then it is exposed to external applications otherwise it is exposed internally to edge application components within an Edge Cloud. When exposed to external world, an external dynamic port is assigned for UC traffic and mapped to the internal containerPort

Attribute Name	Data Type	P	Cardinality	Description
network	String	O	1	Name of the network. In case the application must be assoisated with more than 1 network then the Application Provider must define the name of the network on which this interface has to be exposed. This parameter is required only if the port must be exposed on a specific network other than default.
interfaceName	String	O	1	Interface Name. Required only if application must be attached to a network other than default.

Table 97: Component interface exposure information

4.1.1.8.2.5 computeResourceProfile

Attribute Name	Data Type	P	Cardinality	Description
cpuArchType	Enum	M	1	CPU instruction set architecture (ISA). e.g. Intel, ARM etc.
numCPU	Integer	M	1	Total number of vCPUs
memory	Long	M	1	Total physical memory (RAM) for given ISA type (in Mbytes)
diskStorage	Long	M	1	Total storage (RAM) for workloads for given ISA type (in GB)
gpuInfo	Array(gpuResourceInfo)	O	0..1	Total GPU for workloads for given ISA type
FPGA	Int	O	0..1	Total FPGA for workloads for given ISA type
vpu	Int	O	0..1	Total VPUs for workloads for given ISA type
hugepages	Array(haugepageInfo)	O	1..N	Huge pages for workload for a given ISA type
cpuExclusivity	Boolean	O	0..1	Support for exclusive CPUs

Table 98: Compute Resource model for application components

4.1.1.8.2.6 compEnvParams

Attribute Name	Data Type	P	Cardinality	Description
envVarName	String	M	1	Environment variable name
envValueType	enum	M	1	Defines the content present in envVarValue. Possible value could be “network”, “constant”, “ewbi-dns”, “pri-dns”.

Attribute Name	Data Type	P	Cardinality	Description
				Based on envValueType, an OP may either assign the constant value to the environment variable and pass it to the application component. Or, the value to the be assigned to "envVarValue" will be generated by the application runtime environment and passed on to the component instance during instantiation. If set to "network", then the dynamic port assigned
envVarValue	String	M	1	Value assigned to the envVarName attribute and passed to the container instance during instantiation phase
envValSrc	String	C	1	Network interface Id defined by the Application Provider in ContainerSpec. Based on the given network interface Id, OP will assign the value of dynamic port it generates for the containerPort and assign to the envVarValue.

Table 99: Component Environment Variables

4.1.1.8.2.7 persistentVolume

Attribute Name	Data Type	P	Cardinality	Description
volumeName	String	M	1	Human readable name for the volume
volumeSize	Integer	M	1	size of the volume given by user (10GB, 20GB, 50 GB or 100GB)
volumeMountPath	string	M	1	defines the mount path of the volume where the volume will be available to containers
ephemeralType	Enum	M	1	It indicates the ephemeral storage on the node and contents are not preserved if containers restart
accessMode	String	M	1	Values are RW (read/write) and RO (read-only)
sharingPolicy	Enum	M	1	Exclusive or Shared. If shared, then in case of multiple containers same volume will be shared across the containers.

Table 100: Persistent Volume

4.1.1.8.2.8 artefactRepoLocation

Following table describes the artefactRepoLocation which could be an external repository from where application component images e.g., artefacts can be pulled.

Attribute Name	Data Type	P	Cardinality	Description
repoType	String	M	1	Github, Helm, localRepo. For ContainerSpec valid value is “localRepo”
repoURL	Link	M	1	defines the path/URL of the source artefact
userName	String	M	1	defines the container repo username in case external repository is used to provide component images
Password	String	M	1	defines the container repo password in case external repository is used to provide component images
Token	String	O	1	Authorization Token

Table 101: artefactRepoLocation

4.1.1.8.2.9 fileDetails

Parameter Name	P	Cardinality	Description
fileId	M	1	Identifier unique within a Federation context to distinguish different artefacts
appProviderId	M	1	A unique Application Provider identifier managed at the Leading OP representing the association of a given artefact with an Application Provider
fileName	M	1	Name of the file provided by the Application Provider on NBI. The NBI may provide capabilities to upload files from local filesystems from where NBI is accessed
fileDescription	O	1	Brief description of the file by the Application Provider
fileVersionInfo	M	1	File version information
fileType	M	1	Indicate if the file is Container image or VM image (QCOW2)
imgOSType	M	1	Base OS for the image. Currently only “Linux” is supported
imgInsSetArch	M	1	“x86_64”, “arm64”

4.1.1.8.3 File Details Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

4.1.1.8.3.1 Simple data types

Type Name	Type Definition	Description
artfactId	String	Identifier unique within an appProviderId to distinguish different artefacts
appProviderId	String	A unique Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider
artefactName	String	Name of the artefact
artefactDescription	String	Brief description of the artefact by the Application Provider
artefactVersionInfo	String	Artefact version information
artefactImageFileName	String	Artefact image file name
artefactDescriptorFileName	String	File Name of the artefact descriptor e.g. Helm File Name

Table 102: Artefacts simple datatype table

4.1.1.8.3.2 Enumeration: artefactVirtType

The enumeration cpuArchType represents the Instruction Set Architecture (ISA) for CPU.

Enumeration value	Description
VM_TYPE	Indicates VM images
CONTAINER_TYPE	Indicate containers images

Table 103: artefactVirtType table

4.1.1.8.3.3 Enumeration: artefactDescriptorType

The enumeration artefactDescriptorType represents the artefact descriptor which could be a helm chart for containers deployment, Terraform script for virtual machine deployment etc.

Enumeration value	Description
CONTAINERSPEC_TYPE	Indicates Container-as-a-service deployment specification
HELM_TYPE	Indicate Helm charts
TERRAFORM_TYPE	Indicates Terraform script for VM deployment

Table 104: artefactDescriptorType table

4.1.1.8.3.4 Enumeration: containerOSType

The enumeration containerOSType represents the operating system for which a container image is built for.

Enumeration value	Description
CONTAINER_OS_LINUX	Indicates Linux OS based container
CONTAINER_OS_WINDOWS	Indicate Windows OS based container

Table 105: artefactDescriptorType table

4.1.1.8.3.5 Enumeration: restartPolicy

The enumeration restartPolicy represents the action to be taken if a container instance fails.

Enumeration value	Description
RESTART_POLICY_ALWAYS	Indicates always restart the failed container
RESTART_POLICY_NEVER	Indicate never restart the failed container

Table 106: restartPolicy table

4.1.1.8.3.6 Enumeration: commProtocol

The enumeration commProtocol represents the IP network protocol i.e., TCP or UDP.

Enumeration value	Description
IP_PROTO_TCP	Indicates TCP protocol
IP_PROTO_UDP	Indicate UDP protocol

Table 107: commProtocol table

4.1.1.8.3.7 Enumeration: visibilityType

The enumeration visibilityType represents the if a given interface of application component to be exposed to external clients or to internal components only.

Enumeration value	Description
VISIBILITY_EXTERNAL	Indicates container interface is exposed externally to clients
VISIBILITY_INTERNAL	Indicate container interface is only internally accessible to other components of the application

Table 108: commProtocol table

4.1.2 Application Provider Resource Management - APIs

The REST APIs mentioned in this section provides the capabilities to reserve and manage Edge Resources for an Application Provider within the Availability Zones of a Partner OP.

4.1.2.1 Introduction

Following table describe the applicable HTTP methods for managing resource reservation with the Partner OP. Resources can be reserved per Availability Zone for an Application Provider and once reserved, the Application Provider can associate an application to consume the reserved resources.

Operations	HTTP Method	Resource URI	Qualifier
Reserve Compute Resources	POST	/operatorplatform/federation/v1/{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}	M
Update Compute Resource Reservation	PATCH	/operatorplatform/federation/v1/{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}/pool/{poolId}	M
View Reserved Resources	GET	/operatorplatform/federation/v1/{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}	M
Remove Reserved Resources	DELETE	/operatorplatform/federation/v1/{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}/pool/{poolId}	M
Resource Reservation Notification	POST	{ resourceReservationCallbackLink }	M

Table 109: Compute Resource Reservation Management Methods

4.1.2.2 POST Method: Reserve Compute Resources

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship
zoneId	M	1	Identifier of partner Availability Zone where resources are to be reserved.
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Leading OP
poolName	M	1	Application Provider defines a name to identify the resources reserved on the Availability Zone
resRequest	M	1	Compute flavours to be reserved and their counts
resourceReservationCallbackLink	M	1	Callback URI for the Partner OP to provide status update to the resource reservation request initiated by the Originating OP

Table 110: Reserve Compute Resources request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
reservedPoolId	C	200	ISV Resource reservation request accepted
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 111: Reserve Compute Resource response parameters

4.1.2.3 PATCH Method: Update Compute Resource Reservation

The following table describes the data structures supported by the PATCH Request Body on this resource to modify already reserved resources.

Data Type	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
zoneId	M	1	Availability Zone where resources are reserved.
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Leading OP
poolId	M	1	Identifier of the resource pool
UpdResInfo	M	1	List of modification to be done

Table 112: Update Compute Resource Reservation request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource.

Parameter Name	P	Response Codes	Description
NA	C	200	Resource pool updated
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 113: Update Compute Resource Reservation response parameters

4.1.2.4 GET Method: View Reserved Resources

The following table describes the data structures supported by the GET Request Body on this resource.

Data Type	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
zoneId	M	1	Availability Zone where resources are reserved.
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Originating OP

Table 114: View Reserved Resource request parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
reservedPools	C	200	Reserved Resources Details
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found

Parameter Name	P	Response Codes	Description
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 115: Notify resource reservation status response parameters

4.1.2.5 DELETE Method: Remove Reserved Resources

The following table describes the data structures supported by the DELETE Request Body on this resource.

Data Type	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
zoneId	M	1	Availability Zone where resources are reserved.
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Leading OP
poolId	M	1	Identifier of the resource pool

Table 116: Remove Reserved Resource request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
NA	C	200	Resource pool deleted
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available

Parameter Name	P	Response Codes	Description
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 117: Remove Reserved Resources response parameters

4.1.2.6 POST Method: Notify Resource Reservation Status

Parameter Name	P	Cardinality	Description
federationContextId	M	1	Federation context identifier
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Leading OP
zoneId	M	1	Identifier of partner Availability Zone where resources are to be reserved.
poolId	M	1	Identifier of resource pool
grantedFlavours	M	0..N	List of flavourResvInfo indicating the allocated resources against the requested resources by the Partner OP

Table 118: ISV resource reservation status notification parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	204	Resource reservation status updated
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 119: Application Onboarding response parameters

4.1.2.7 Data Model

4.1.2.7.1 General

This subclause specifies the application data model supported by the Application Provider resource Management APIs.

4.1.2.7.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.1.2.7.2.1 resRequest

The below table describes the flavours and their respective duration for which they may be reserved.

Attribute Name	Data Type	P	Cardinality	Description
flavours	Array(flavourResvInfo)	M	1	An identifier to refer to the combination of Edge Resource configuration as indicated by the other attributes in this table
reserveDuration	Object	M	1	Time period for which resources are to be reserved starting from now

Table 120: resRequest

4.1.2.7.2.2 flavourResvInto

The following table provides flavours and the corresponding amount to be reserved.

Attribute Name	Data Type	P	Cardinality	Description
flavourId	Int	M	1	Flavour Identifier
numFlavour	Int	M	1	Number of flavour to be reserved
minNumOfFlavours	Int	O	1	If specified, indicate the minimum numbers of flavours to be reserved up to maximum as given in “count” member. If the Partner OP cannot reserve the minimum number of flavours, then the request shall be failed.

Table 121: flavourResvInfo

4.1.2.7.2.3 reservedPoolId

The following table provides the information on the resource pool identifier and resource pool name which can be used to refer to an existing pool of resources reserved earlier on request from Application Providers.

Attribute Name	Data Type	P	Cardinality	Description
poolName	String	M	1	Name of the pool
poolId	String	M	1	Identifier generated by the OP to identify these reserved resources

Table 122: reservedPoolId

4.1.2.7.2.4 UpdResInfo

The data structure in the below Table 123 provides the information to modify existing resource pool created earlier on request from Application Providers towards the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
updateType	String	M	1	Enumerations – Add, Remove, Duration
flavourId	String	M	1	Flavour identifier
count	Int	M		Final count of flavour that should be reserved. Value 0 means remove all such flavour
reserveDuration	Object	C	1	New time period for which resources are to be reserved from initial reservation time

Table 123: updResInfo

4.1.2.7.2.5 reservedPools

The following table defines the relationship between the resource pool identifier and the resource flavours associated to it.

Attribute Name	Data Type	P	Cardinality	Description
reservedPoolName	String	M	1	Name of the pool
reservedPoolId	Object	M	1	Application Provider defined name of the pool
reservedFlavours	Array(flavourId)	M	1	List of flavours and their count reserved for this poolId
reserveDuration	Object	O	1	Time period for which resources are to be reserved starting from now
reservationTime	Date-Time	O	1	Date and time of resource reservation by the Application Provider

Table 124: Reserved Pool Info

4.1.2.7.2.6 reserveDuration

The following table defines the time duration for which resource reservation is being requested.

Attribute Name	Data Type	P	Cardinality	Description
numOfDays	Int	C	1	Number of days to be reserved
numOfMonths	Int	C	1	Number of months to be reserved
numOfYears	Int	C	1	Number of years to be reserved

Table 125: reservationDuration

4.1.2.7.2.7 grantedFlavours

The following table defines the structure of granted resources for a resource reservation request.

Attribute Name	Data Type	P	Cardinality	Description
grantedFlavours	Array(flavourResvInfo)	M	1..N	Number of flavours reserved

Table 126: grantedFlavours

4.1.3 Application Onboarding Management - API

Application onboarding management APIs are used to provide the application information to the Partner OP by the Leading OP.

4.1.3.1 Introduction

Following table describes the HTTP methods to the resources defined in the table.

Operation	HTTP Method	Resource URI	Qualifier
Onboard Application	POST	/operatorplatform/federation/v1/{federationContextId} /application/onboarding	M
Update Application	PATCH	/operatorplatform/federation/v1/{federationContextId} /application/onboarding/app/{appid}	M
Remove Application	DELETE	/operatorplatform/federation/v1/{federationContextId} / application/onboarding/app/{appid}/zone/{zoneId}	M
View Application	GET	/operatorplatform/federation/v1/{federationContextId} /application/onboarding/app/{appid}	M
Notify Application State Info	POST	{ appStatusCallbackLink }	M
Application Onboard at new Availability Zones	POST	/operatorplatform/federation/v1/{federationContextId} /application/onboarding/app/{appId}/additionalZones	M
Restrict Application	POST	/operatorplatform/federation/v1/{federationContextId} /application/onboarding/app/{appId}/zoneForbid	M

Table 127: Application Onboarding Management APIs

4.1.3.2 POST Method: Onboard Applications

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
applInformation	M	1	Application Edge Resource, component images, QoS, Availability Zone information

Table 128: Application Onboarding request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application onboarding request accepted
N/A	C	200	Application onboarded successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 129: Application Onboarding response parameters

4.1.3.3 PATCH Method: Update Application Information

The following table describes the data structures supported by the PATCH Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
appId	M	1	Application Identifier for a given appProviderId.
appUpdQoSProfile	O	1	Application resource requirement or deployment attributes that needs to be updated

Parameter Name	P	Cardinality	Description
appComponentSpecs	O	1	Application components and their assosiated artefacts or Domain Name System (DNS).

Table 130: Modify application information request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Application Updated successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 131: Modify application information response parameters

4.1.3.4 DELETE Method: Remove Application Information

The following table describes the data structures supported by the DELETE Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
appId	M	1	Application Identifier for a given appProviderId.
zoneId	M	1	Availability Zone identifiers from where application must be deboarded.

Table 132: Remove application request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application Update request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 133: Remove application response parameters

4.1.3.5 POST Method: Notify resource reservation Status Information

The following table describes the POST request which the Partner OP initiate towards the Leading OP to provide status update or completion of an earlier resource reservation request.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Partner OP to the Leading OP to identify the existing Federation relationship
appId	M	1	Application Identifier for a given appProviderId.
statusInfo	M	1	Status of an application on Availability Zone.

Table 134: Resource reservation notification parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	204	Resource reservation status updated
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed.

Parameter Name	P	Response Codes	Description
			Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 135: Resource reservation notification response parameters

4.1.3.6 POST Method: Application Onboarding At New Availability Zones

The Originating OP requests the Partner OP to make an already onboarded application available on additional Availability Zones specified in the request.

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
appId	M	1	Application Identifier
zones	M	1	List of Availability Zone identifiers where application shall be made available.

Table 136: Application Onboarding on new Availability Zones request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application onboard request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 137: Application Onboarding response parameters

4.1.3.7 POST Method: Restrict Application

The Originating OP request to the Partner OP to restrict or allow instantiation of the application on specified Availability Zones.

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
appId	M	1	Application Identifier
appInstantiationCtrlList	M	1	List of Availability Zone identifier and access info

Table 138: Application Onboarding request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application forbid/permit request accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 139: Application Onboarding response parameters

4.1.3.8 GET Method: View Application Information

The Originating OP can use the HTTP GET method to retrieve the details of application earlier onboarded to the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
appProviderId	M	1	Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider
appId	M	1	Application Identifier for a given appProviderId.

Table 140: View Artefact request parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
appInformation	M	1	Application Provider identifier managed at Leading OP representing the association of a given artefact with an Application Provider

Table 141: View Artefact response parameters

The following table describes the data structures supported by the GET Request Body on this resource for non-200 HTTP codes.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 142: Non-200 Response Codes for View Application Response

4.1.3.9 Data Model

4.1.3.9.1 General

This subclause specifies the application data model supported by the Application Onboarding Management API.

4.1.3.9.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.1.3.9.2.1 appInformation

Following table describes the information elements defining an edge application.

Attribute Name	Data Type	P	Cardinality	Description
appId	String	M	1	Identifier of the application
appProviderId	String	M	1	Unique Identifier to identify the Application Providers of the Leading OP
appDeploymentZones	Array(regionInfo)	M	1..N	Geographical location where application should be made available
appMetaData	Object	M	1	Application metadata details
appQoSProfile	Object	O	1	Parameters corresponding to the performance constraints, tenancy details etc.
appProvisioning	Bool	O	1	Define if application can be instantiated or not
appComponentSpecs	Array(appComponentSpec)	M	1..N	Details about application components, associated component images and descriptors, Edge Resources etc.
appStatusCallbackLink	uri	O	1	An Application Callback URL which shall be used by the Partner OP to inform Home OP about change in application status or changes in status or an application instance.
edgeAppFQDN	String	O	1	DNS FQDN assigned to application instances in an Availability Zone. User Clients can resolve the FQDN to communicate with the edge instances of the application

Table 143: appInformation

4.1.3.9.2.2 regionInfo

Attribute Name	Data Type	P	Cardinality	Description
countryCode	Object	M	1	ISO 3166-1 Country Code where application is to be deployed
zoneInfo	Object	M	1	Availability Zone identifiers for given targerOPId

Table 144: regionInfo

4.1.3.9.2.3 appMetaData

Attribute Name	Data Type	P	Cardinality	Description
version	String	M	1	Application version
appName	String	M	1	Name of the application
appDescription	String	O	1	Brief application description provided by Application Provider
accessToken	String	M	1	An application Access key to be used with UNI interface to authorize UCs Access to a given application
mobilitySupport	String	O	1	Indicates if an application is sensitive to User Mobility and can be relocated. Default is “NO”
requiresEdgeFQDN	String	O	1	Indicates if the OP needs to assign Edge Resource-specific FQDN for instances of the application in an Availability Zone

Table 145: Application meta data

4.1.3.9.2.4 appQoSProfile

Attribute Name	Data Type	P	Cardinality	Description
latencyConstraints	String	M	1	Latency requirements for the application. Allowed values (non-standardized) are none, low and very low. Very Low may corresponds to range 15 - 30 msec, Low correspond to range 30 - 50 msec. None means 51 and above
bandwidthRequired	String	O	1	Data transfer bandwidth requirement (minimum limit) for the application. It should in Mbits/sec

Table 146: Application QoS Performance Profile

4.1.3.9.2.5 appComponentsSpec

An application may consist of one or more components where a component represents a runnable unit of the application. A component tie together one or more artefacts i.e., an artefact associated to an image type and/or an artefact which refers to a component descriptor e.g., Helm chart, Terraform file etc.

Attribute Name	Data Type	P	Cardinality	Description
appComponentsSpec	Array(appComponentDetail)	M	1	

Table 147: Application Components

4.1.3.9.2.6 appComponentDetail

Attribute Name	Data Type	P	Cardinality	Description
serviceNameNB	String	M	1	Must be a valid RFC 1035 label name not more than 64 characters. This defines the DNS name via which the component can be accessed over NBI. Access via serviceNameNB is restricted on specific ports. Operator Platform shall expose component access externally via this DNS name
serviceNameEW	String	O	1	Must be a valid RFC 1035 label name not more than 64 characters. This defines the DNS name via which the component can be accessed via peer components. Access via serviceNameEW is open on all ports. Operator Platform shall not expose serviceNameEW externally outside edge.
componentName	String	M	1	Must be a valid RFC 1035 label name. Component name must be unique with an application. It should be atleast 8 characters in length and not more than 64 characters
artefactId	String	M	1	Identifier of the already onboarded artefact to be used for instantiating the component of the associated application. It refers to artefactDescriptors e.g., Helm chart, Container Spec, Terraform script etc.

Table 148: Application Component Details

4.1.3.9.2.7 countryCode

ISO 3166-1 country code to uniquely provide the country information where OP services have been deployed by an Operator.

Attribute Name	Data Type	P	Cardinality	Description
countryName	String	M	1	Name of the country
countryCode	String	M	1	Two digit ISO 3166-1-alpha-2 country code e.g., “ES” for Spain

Table 149: Country Code

4.1.3.9.2.8 zoneInfo

Following table describes Availability Zone identifier where an application shall be onboarded.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone identifier

Table 150: Availability Zone identifier info for application onboarding

4.1.3.9.2.9 appUpdQoSProfile

Update request shall contain at least one of the optional parameters defined inthe following table.

Attribute Name	Data Type	P	Cardinality	Description
latencyConstraints	String	O	1	Latency requirements for the application. Allowed values (non-standardized) are none, low and ultra-low. Ultra-Low may corresponds to range 15 - 30 msec, Low correspond to range 30 - 50 msec. None means 51 and above
bandwidthRequired	String	O	1	Data transfer bandwidth requirement (minimum limit) for the application. It should in Mbits/sec
mobilitySupport	String	O	1	Indicates if an application is sensitive to User Mobility and can be relocated. Default is “NO”
multiUserClients	Enum	O	1	Define if Application supports single user or multiple user clients (UCs)
noOfUsersPerAppInst	Integer	C	1	For multi user client’s Application, how many UCs an application instance can support

Attribute Name	Data Type	P	Cardinality	Description
appProvisioning	Bool	O	1	Define if application can be instantiated or not

Table 151: Application QoS Performance Profile Update Parameters

4.1.3.9.2.10 statusInfo

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	O	1	Availability Zone Identifier
onboardStatusInfo	Enum	O	1	Application onboarding status

Table 152: StatusInfo

4.1.3.9.2.11 appInstantiationCtrlList

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone Identifier
forbid	boolean	M	1	Value 'true' will forbid application instantiation on this Availability Zone. No new instance of the application can be created on this Availability Zone

Table 153: appInstantiationCtrlList

4.1.3.9.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

4.1.3.9.3.1 Simple data types

The following table provides the information about the single element data types that are used in application management HTTP methods.

Attribute Name	Data Type	Description
edgeAppFQDN	String	Based on the Partner OP policy the FQDN could be generated indicating the nature that the FQDN is Edge Resource, Availability Zone specific e.g. <appId>.<OPId>.<AZ>.<SubDomain>

Table 154: Simple Data Types for Application Onboarding

4.1.3.9.3.2 Enumeration: multiUserClients

The following table defines the attribute of an application to indicate if it can support single or multiple UCs.

Enumeration value	Description
APP_TYPE_SINGLE_USER	A single user client (UC) can connect to an instance of the application
APP_TYPE_MULTI_USER	Multi user client (UCs) can connect to an instance of the application

Table 155: multiUserClients

4.1.3.9.3.3 Enumeration: onboardingStatusInfo

The following table defines the application onboarding status information.

Enumeration value	Description
PENDING	Application onboarding in progress
ONBOARDED	Application onboarded successfully
DEBOARDING	Application deboarding in progress
FAILED	Application onboarding failed

Table 156: Onboarding status info

4.1.3.9.3.4 Enumeration: resourceConsumption

The following table defines if an application instance shall use the resources from the reserved resource pool.

Enumeration value	Description
RESERVED_RES_ONLY	Instruct OP to use only the reserved resources
RESERVED_RES_PREFER	Instruct OP to first give preference to already reserved resource, If none available OP may use non reserved resources
RESERVED_RES_FORBID	instruct OP not to use pre-reserved resources

Table 157: Resource reservation indication table

4.1.4 Application Instance Lifecycle Management - API

The API mentioned in this section provides the capabilities for managing the edge applications instantiation and terminating the running instance, inquire the status of the application instance etc for applications with the Partner OPs.

4.1.4.1 Introduction

Following table describes the applicable HTTP methods for applications lifecycle management.

Operations	HTTP Method	Resource URI	Qualifier
Instantiate Application	POST	/operatorplatform/federation/v1/{federationContextId}/application/lcm	M

Operations	HTTP Method	Resource URI	Qualifier
Remove Application Instance	DELETE	/operatorplatform/federation/v1/ /{federationContextId}/application/lcm/app /{appId}/instance/{appInstanceId}/zone/{zoneId}	M
View Application Instance	GET	/operatorplatform/federation/v1 /{federationContextId}/application/lcm/app /{appId}/instance/{appInstance}/zone/{zoneId}	M
List Application Instances	GET	/operatorplatform/federation/v1/{federationContextId}/application/lcm/app/{appId}/appProvider/{appProviderId}	M
Notify Application Instance state information	POST	{appInstCallbackLink}	M

Table 158: Application Instance Management Methods

4.1.4.2 POST Method: Instantiate Applications

The following table describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
transactionId	M	1	Transaction identifier for supporting retries
appId	M	1	Application Identifier for a given Federation context
appProviderId	M	1	A unique Application Provider Identifier referring an Application Provider account with Leading OP
appVersion	M	1	Application Version of the Application provided by the Leading OP Application Provider
zoneInfo	M	1	Availability Zone where an already onboarded application can be instantiated. It also includes details about the resources to be used for application instantiation
appInstCallbackLink	M	1	An application instance Callback URL which shall be used by the Partner OP to inform the application instance information asynchronously

Table 159: Application instantiation request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
transId	C	202	Application provisioning accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 160: Application instantiation response parameters

4.1.4.3 DELETE Method: Terminate Application Instance

The tables below describe the data structures supported by the DELETE Request Body on this resource.

Data Type	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship
appId	M	1	Application Identifier for a given appProviderId .
zoneId	M	1	Availability Zone Identifier where application instance is running
appInstIdentifier	M	1	Application instance identifier to refer to a running instance of an application denoted by appId

Table 161: Application instance termination request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
appInstanceId	C	202	Application instance termination request Accepted
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found

Parameter Name	P	Response Codes	Description
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 162: Application instance termination response parameters

4.1.4.4 POST Method: Notify Application Instance Information

Partner OP uses this API to inform Originating OP about the results of application instantiation request. This API also includes details about endpoints (IP and Ports) that can be used to reach application instance.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
appId	M	1	Application Identifier unique per application in an appProviderId
appInstIdentifier	M	1	Application instance identifier sent by the Partner OP in response to application instantiation request
zoneId	M	1	Availability Zone identifier of the Application referred by appId
appInstanceInfo	M	1	Application instance information e.g., communication end points of various components of the Application, Availability Zone where it is deployed denoted by appId.

Table 163: Application instance async request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application provisioning notification acknowledged
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found

Parameter Name	P	Response Codes	Description
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 164: Application instance async response parameters

4.1.4.5 GET Method: View Application Instance Details

View application instance details GET request contains the following parameters towards the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
appInstanceId	M	1	Application instance identifier sent by the Partner OP in response to application instantiation request
zoneId	M	1	Identifier of partner Availability Zone where application instance is created.

Table 165: Application instance async request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
appInstanceInfo	C	200	Application instance details
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Server Returned an Unknown Error

Table 166: View application instance details response parameters

4.1.4.6 GET Method: List Application Instances

View application instance GET request contains the following parameters towards the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
appId	M	1	Application Identifier for a given appProviderId
zoneId	M	1	Availability Zone identifier where the Application referred by appId is deployed

Table 167: List application instance parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
appInstanceList	C	200	Application instance list
problemDetails	C	400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails	C	409	Conflict
problemDetails	C	412	Pre-condition failed. Application not onboarded or resources not available
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Server Returned an Unknown Error

Table 168: List application instance response parameters

4.1.4.7 Data Model

4.1.4.7.1 General

This subclause specifies the application data model supported by the Application Onboarding Management API.

4.1.4.7.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.1.4.7.2.1 appInstanceInfo

Following table describes the information elements associated to an instance of the edge application.

Attribute Name	Data Type	P	Cardinality	Description
appInstanceState	enum	M	1	Pending, Running, Failed etc.
accessPointInfo	Array(Object)	M	1	Information on external connectivity parameters where clients can connect to the application instance over UNI

Table 169: appInstanceInfo

4.1.4.7.2.2 accessPointInfo

Following table describes the connectivity information of an edge application instance.

Attribute Name	Data Type	P	Cardinality	Description
interfaceId	String	M	1	Developer/Independent Software Vendor (ISV) defined logical name for TCP/UDP endpoint exposed by the application as part of the Application component structure
accessPoints	Object	M	1	Details of IP address, port, FQDN etc.

Table 170: accessPointInfo

4.1.4.7.2.3 accessPoints

Following table describes the protocol level details of the connectivity information of an edge application instance.

Attribute Name	Data Type	P	Cardinality	Description
ipv4Addresses	IPv4 Address	C	1	IPv4 address of the application component instance on requested Availability Zone where UC can connect with application instance on edge
ipv6Addresses	IPv6 Address	C	1	IPv6 address of the application component instance on requested Availability Zone where UC can connect with application instance on edge

Attribute Name	Data Type	P	Cardinality	Description
port	string	M	1	Port of the application component instance on requested Availability Zone where UC can connect with application instance on edge
edgeAppFQDN	String	O	1	FQDN at which the application instance can be accessed

Table 171: accessPoints

4.1.4.7.2.4 appInstanceList

Following table describes the application instance list containing the details of the application running instances created for an application.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	fqdn of the application component instance on requested Availability Zone where UC can connect with application instance on edge
appInstanceInfo	Array	M	1..N	List for application instance Identifier and instance state

Table 172: application Instance list

4.1.4.7.2.5 InstanceIdentifiers

List of zoneIds and application instances created on that Availability Zone

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Partner Availability Zone identifier
appInstIdentifier	String	M	1	Application instance identifier. This identifier the instance created on the Availability Zone.

Table 173: Application Instance Identifiers

4.1.4.7.2.6 zoneInfo

The following table defines the Availability Zone and resource pool details where application instance shall be created and the resource pool to be used by the application instance.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone identifier
flavourId	String	M	1	Flavour that should be used for the application on an Availability Zone

Attribute Name	Data Type	P	Cardinality	Description
resPool	String	C	1	Id of resource pool that was reserved by the Application Provider and that shall be used to instantiate the application.
resourceConsumption	enum	C	1	Specifies if the application can be instantiated using pre-reserved resource or not. Application Provider can pre-reserve a pool of Edge Resources on each Availability Zone.

Table 174: Availability Zone and flavour info for application instantiation

4.1.5 Edge Node Sharing - API

4.1.5.1 Introduction

Following table describes the operations, applicable HTTP methods and the resource URI for edge node sharing API.

Operations	HTTP Method	Resource URI	Qualifier
Edge Node Discovery	POST	/operatorplatform/federation/v1/{federationContextId}/edgenodesharing/edgeDiscovery	M

Table 175: Edge Node Sharing Operations and Resource URI

4.1.5.2 POST Method: Edge Node Discovery

This operation is intended for OP A to fetch the edge nodes discovery information to be shared from a Partner OP B.

Parameter name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
appProviderId	M	1	Application Provider identifier
appId	M	1	Application Identifier
edgeDiscoveryFilters	O	1	Edge node discovery filters to help Partner OP to select adequate edge(s)

Table 176: Edge Node Discovery Request Parameters

The following table contains the HTTP Response body parameters for 200 OK response.

Parameter name	P	Response Codes	Description
easDiscoveryResp	M	1	Indicates the application access information in locations indicated in edge node share request

Table 177: Edge node discovery response Parameters

The following table describes the data structures supported by the POST Response Body on this resource for non-200 OK responses.

Parameter Name	P	Response Codes	Description
problemDetails		400	Bad Request
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Content not found
problemDetails		409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 178: Edge node discovery failure responses

4.1.5.3 Data Model

4.1.5.3.1 General

Parameter name	P	Cardinality	Description
easDiscoveryResp	C	1	Edge node share response parameter
edgeDiscoveryFilters	O	1	Edge node discovery filters to help the Partner OP to select adequate edge(s)

Table 179: Data structures for edge node discovery API

4.1.5.3.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.1.5.3.2.1 nodeDiscoveryResponse

The following table provides the information about the response parameters that may be returned by the Partner OP for edge node discovery request.

Attribute Name	Data Type	P	Cardinality	Description
candidateZones	Array (discoveredEdgeNodes)	O	1..N	List of Edge discovery information (e.g. URI, FQDN, IP address)
candidateAppInsts	discoveredAppInsts	O	0..N	Details of already running instances of the given Application identified by appId, appProviderId

Table 180: nodeDiscoveryResponse

4.1.5.3.2.2 discoveredEdgeNodes

The following table provides the information about the response parameter discoveredEdgeNodes that may be returned by the Partner OP in response to edge node discovery request.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone identifier of Partner OP
latencyServiceEndPoints	Object	M	1	FQDN, IP and Port information about the probe responder service that can be further used by the UE to determine traffic latency.

Table 181: discoveredEdgeNodes

4.1.5.3.2.3 edgeDiscoveryFilters

The following table provides the information about the edge discovery filters which Originating OP may include as additional qualifying information to Partner OP for filtering the available edge node(s) using this information.

Attribute Name	Data Type	P	Cardinality	Description
locationInfo	String	O	0..1	Information obtained from the Home OP regarding UE location.to help Partner OP locate the adequate Availability Zones in UE location. It could be Latitude/Longitude or zoneId of the UE

Table 182: edgeDiscoveryFilters

4.1.5.3.2.4 discoveredAppInsts

The following table provides the information about the Availability Zones and instances of a given application in one or more Availability Zones.

Attribute Name	Data Type	P	Cardinality	Description
appId	String	M	1	Application Identifier to indicate Edge Resources that are assosiated to this Application
appProviderId	String	M	1	The Application Provider ID to indicate the association between the application and the Application Provider at the Leading OP
appInstances	Array(instancesInfo)	O	0..N	Application Instances already running in one or more Edge Resources of the edge sharing Partner OP

Table 183: discoveredAppInsts

4.1.5.3.2.5 instancesInfo

The following table provides the information about the application instances already running on one or more Edge Resources in different Availability Zones.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone Identifier where the instance of an Application identified by appId is already running
instanceDetails	Array(appInstanceInfo)	O	1..N	Detailed information about the application instance on Edge Resources belonging to edge node sharing Partner OP

Table 184: instancesInfo

4.2 Service APIs Federation

This section provides the details of the Service APIs Federation methods and parameters over the E/WBI.

4.2.1 Service APIs Forwarding Methods

This section provides the details of various HTTP methods along with the associated resource URIs, API parameters in request and response, error codes etc. for the Service API Federation management over the E/WBI.

4.2.1.1 Introduction

The below table describes the supported operations and resource URIs for Service APIs Federation management.

Operations	HTTP Method	Resource URI	Qualifier
Service API Forwarding	POST	/operatorplatform/federation/v1/{federationContextId}/apiservice/{apiServiceId}	M
Service API Event Notification	POST	{svcNotificationDest}	M
Leading OP Remove API Context	DELETE	/operatorplatform/federation/v1/{federationContextId}/apiservice/{apiServiceId}/connid/{connectID}	M

Table 185:Service APIs Management Methods

4.2.1.2 POST Method: Service API Forwarding

The table below describes the data structures supported by the POST Request Body on this resource. The POST method on this resource provides the capability for the Leading OP to forward the Service API after determining the Partner OP indicated by the Federation context identifier for this service.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
apiTxnId	M	1	API request transaction identifier
apiServiceId	M	1	The name identifier of the Service API
customerInfo	M	1	The sharable identification information that a Leading OP can expose with the Partner OP to enable mechanisms like obtaining Consent of the End-User indicated in the Service API request
customerID	M	1	A unique static identifier at the Leading OP representing the Enterprise the Service API is received from and which may be used by the Partner OP to obtain Consent of the End-User whose identity is embedded in the ServiceAPIContent
ServiceAPIContent	M	1	Service API Body contents as received by the Leading OP over NBI

Parameter Name	P	Cardinality	Description
eventNotificationDest	O	0..1	A URL link for the Partner OP to provide event notifications for long duration contextful APIs e.g., QoD for a given API session

Table 186: Service API Forwarding parameters

The table below describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
serviceAPIResp	C	200	Artefacts uploaded successfully at Partner OP
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 187: Service API Forwarding response parameters

The table below describes the “Location” header supported by the POST Response Body on this resource if the Service API is associated to a long duration API session.

Name	Data Type	P	Cardinality	Description
Location	String	M	1	Contains the URI of the newly created resource i.e., /operatorplatform/federation/v1/partner/{federationContextId}/apiservice/customerid/{customerID}/connid/{connectID}

Table 188: Service API forwarding Response with Location header

4.2.1.3 POST Method: Service API Event Notification

The table below provides the details of the events which a Partner OP may send to the Leading OP based on the nature of the Service API.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to

Parameter Name	P	Cardinality	Description
			identify the existing Federation relationship.
apiTxnId	M	1	Notification request transaction identifier
apiServiceId	M	1	The name identifier of the Service API
eventType	M	1	If the event is connectID timer expiry or the mobile network event
customerID	M	1	A unique static identifier at the Leading OP representing the Enterprise the Service API is received from
connectID	C	1	A temporary identifier generated by the Partner OP representing the End-User whose identity is contained in the ServiceAPIContent. It is needed for long running API sessions e.g., QualityOnDemand API
ServiceAPIEventDef	M	1	Event Schema as defined by Service API specification e.g., for the QoD API

Table 189: Service API Event notification parameters

The table below describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Event Notification successful
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 190: Service API Event notification response parameters

4.2.1.4 GET Method: Retrieve Service API Context Information

The GET method supports the path parameters.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
customerID	M	1	A unique static identifier at the Leading OP representing the Enterprise the Service API is received from
connectID	C	1	A temporary identifier generated by the Partner OP representing the End-User whose identity is contained in the ServiceAPIContent. It is needed for long running API sessions e.g., QualityOnDemand API

Table 191: Service API context retrieval request parameters

The table below describes the data structures supported by the GET Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
ServiceAPIRespDef	M	1	Response Schema for GET request as defined by Service API specification e.g., for QualityOnDemand API
customerID	M	1	A unique static identifier at the Leading OP representing the Enterprise the Service API is received from
connectID	M	1	A temporary identifier generated by the Partner OP representing the End-User whose identity is contained in the ServiceAPIContent. It is needed for long running API sessions e.g., QualityOnDemand API

Table 192: Service API context retrieval response parameters

The table below describes the HTTP codes supported by the GET Response on this resource for non-200 codes.

Parameter Name	P	Cardinality	Response codes	Description
problemDetails	C	1	400	Bad Request. Parameters in the request has conflicting values.
problemDetails	C	1	401	Unauthorized Access
problemDetails	C	1	404	Content Not Found
problemDetails	C	1	409	Conflict.

Parameter Name	P	Cardinality	Response codes	Description
problemDetails	C	1	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	1	500	Internal Server Error
problemDetails	C	1	503	Service Unavailable.
problemDetails	C	1	520	Web Server Returned an Unknown Error

Table 193: Response codes for Availability Zone meta-information Request

4.2.1.5 DELETE Method: Remove API Context by Leading OP

The below table describes the data structures for the DELETE Request initiated by the Leading OP to the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
connectID	M	1	A temporary identifier generated by the Partner OP representing the End-User

Table 194: Remove API Context by Leading OP

The below table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Code	Description
Status	C	200	API context deleted successful
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Federation not found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 195: Remove API Context by Leading OP status codes

4.2.1.6 Data Model

4.2.1.6.1 General

This subclause specifies the application data model supported by the Service APIs Federation Management.

4.2.1.6.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

4.2.1.6.2.1 ServiceAPIContent

The below table describes the data structure the Leading OP uses to forward the Service API to the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
mediaType	String	M	1	The media type indicates the Service API body schema type e.g., “application/json”. The value shall be received in the NBI Service API
serviceAPIPayload	Object	M	1	The Service API body content as received over NBI with the schema indicated in mediaType

Table 196: ServiceAPIContent

4.2.1.6.2.2 serviceAPIResp

The below table describes the data structure that the Partner OP share in response to the Service API forwarding request.

Attribute Name	Data Type	P	Cardinality	Description
customerID	String	M	1	customerID is shared by the Leading OP with the Partner OP during Service API forwarding
targetUserContext	Object	C	1	A temporary End-User context object generated by the Partner OP representing the End-User whose identity is contained in the ServiceAPIContent request. It is needed for long running API sessions e.g., QualityOnDemand API
apiResponse	Object	C	1	This object for sessionless APIs represent the final response of the Service API processing generated by the Partner OP

Table 197: serviceAPIResp

4.2.1.6.2.3 targetUserContext

The table below describes the data structure that a Partner OP generates in context of the End-User whose identity is contained in the Service API forwarding request by the Leading OP.

Attribute Name	Data Type	P	Cardinality	Description
connectID	String	M	1	Temporary token to be generated at the Partner OP in context of session based APIs e.g., QualityOnDemand
expiryDuration	String	M	1	The timer value after which the given connectID expires

Table 198: targetUserConetxt

4.2.1.6.2.4 expiryDuration

The data structure in the following table provides the information to modify existing resource pool created earlier on request from Application Providers towards the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
numHours	Int	C	1	Hours (0-23)
numMins	Int	C	1	Minutes (0-59)
numSecs	Int	C	1	Seconds(0-59)

Table 199: ConnectID Expiry Timer

4.2.1.6.2.5 apiResponse

The below table describes the data structure which the Partner OP uses to provide the Service API processing result to the Leading OP.

Attribute Name	Data Type	P	Cardinality	Description
mediaType	String	M	1	May contain value e.g., “application/json”.
responseContent	Object	M	1	The result of the Service API processing response formatted according to scheme indicated in mediaType and typically defined in the Service API specification

Table 200: Service API Result Information

4.2.1.6.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

4.2.1.6.3.1 Simple data types

Attribute Name	Data Type	Description
apiTxnId	String	An transaction identifier created per request and shall be included in the all the responses associated to same API request
apiServiceId	String	Named identifier of the API service e.g., “QualityOnDemand”, “DeviceStatus”, “DeviceLocation” etc.
connectID	String	Temporary token to be generated at the Partner OP in context of session based APIs e.g., QualityOnDemand
customerID	String	A unique static identifier at the Leading OP representing the Enterprise the Service API is received from and which may be used by the Partner OP to obtain Consent of the End-User whose identity is embedded in the ServiceAPIContent
customerInfo	String	Name identification information associated to the Application Provider of the Leading OP
eventNotificationDest	Link	The URL to which the Partner OP can send the event notification in context of session based APIs e.g., QualityOnDemand

Table 201: Service API Federation simple datatype table

5 Resource Monitoring APIs

The resource monitoring APIs provide the Partner OP to expose Edge Resource consumption information by the applications shared over the E/WBI by the Leading OP. The information comprises

5.1 Edge Resource Monitoring APIs

The set of APIs provides the management of the flow of Edge Resource monitoring information between the OPs. The Partner OP based on the active subscription shares the resource consumption information with the Originating OP.

5.1.1 Availability Zone Resources Utilization – API

This section describes the reporting of Edge Resource measurements information by an OP. The Partner OP performs the measurement of utilization of Edge Resources for federated applications from the Originating OP and, depending upon the peridocity agreed, reports the measurements to Originating OP over the E/WBI.

5.1.1.1 Introduction

The following table describes the HTTP methods for the Availability Zone resources utilization service.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Resource Monitoring Info	POST	/operatorplatform/federation/v1/{federationContextId}/subscriptions	M
Unsubscribe Resource Monitoring	DELETE	operatorplatform/federation/v1/{federationContextId}/subscriptions/{subs_id}	M
Report Monitoring Info	POST	{monitorInfoNotificationDest}	M
Retrieve Monitoring Information	GET	operatorplatform/federation/v1/{federationContextId}/subscriptions/{subs_id}	O

Table 202: Resource Utilization HTTP Methods

5.1.1.2 POST Method: Resource Monitoring Subscription

The HTTP POST request from the Originating OP contains the following parameters towards the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
monitoringSubsType	M	1	Represent the subscription type for Edge Resource monitoring information

Table 203: Monitoring Information subscription parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
subscriptionId	C	200	Monitoring subscription created
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 204: Monitoring subscription response parameters

5.1.1.3 DELETE Method: Unsubscribe Resource Utilization

Following table provides HTTP DELETE request path parameters which an Originating OP sends to the Partner OP to unsubscribe resource monitoring information reporting.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
subscriptionId	M	1	Monitoring information subscription identifier at the Leading OP provided by the Partner OP

Table 205: Unsubscribe resource utilization request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
subscriptionId	C	200	Subscription removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 206: Unsubscribe resource monitoring response parameters

5.1.1.4 GET Method: Retrieve Resource Utilization Information

Following table provides HTTP GET request path and query parameters which an Originating OP sends to a Partner OP to retrieve Edge Resource utilization information.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship. It is a HTTP path parameter
subscriptionId	M	1	Subscription identifier earlier shared by the Partner OP with the Leading OP

Table 207: Retrieve resource utilization information request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
edgeResUtilizeInfo	C	200	Object representing Edge Resource utilization information
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 208: Retrieve resource utilization response parameters

5.1.1.5 POST Method: Reporting Resource Utilization Information

The HTTP POST request is sent by the Partner OP with the following parameters towards the Originating OP to provide Edge Resource utilization information.

Parameter Name	P	Cardinality	Description
subscriptionId	M	1	Subscription identifier earlier shared by the Partner OP with the Leading OP
edgeResUtilizeInfo	M	1	Object representing Edge Resource utilization information
sequenceNumber	M	1	The sequence number to track the delivery of usage reports reliably

Table 209: Resource Utilization information request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Edge Resource monitoring acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict

Parameter Name	P	Response Codes	Description
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 210: Resource Utilization response parameters

5.1.1.6 Data Model

5.1.1.6.1 General

This section provides the data types for the Edge Resource utilization information reporting.

5.1.1.6.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

5.1.1.6.2.1 edgeResUtilizeInfo

Following table describes information about the Edge Resource utilization of the offered resources which the Partner OP reports to the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
edgeResUtilizeInfo	Array(edgeResUtilizeMetrics)	M	1..N	Edge Resource utilization metrics collected by the Partner OP for each Availability Zone

Table 211: edgeResUtilizeInfo

5.1.1.6.2.2 edgeResUtilizeMetrics

Following table describes information about the edgeResUtilizeMetrics which defines the Availability Zone level resource utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone identifier
startTime	String	M	1	Date-time indicating the start of the measurement period
endTime	String	M	1	Date-time indicating the end of the measurement period offset by measurement periodicity from startTime
cpuUtil	cpuUtilization	M	1	CPU utilization metrics for a given Availability Zone

Attribute Name	Data Type	P	Cardinality	Description
memUtil	memUtilization	M	1	Memory utilization metrices for a given Availability Zone
diskUtil	diskUtilization	M	1	Disk utilization metrics for a given Availability Zone
networkUtil	networkUtilization	M	1	Network utilization metrics for a given Availability Zone
flavourUtil	flavourUtilization	M	1	Flavour utilization metrics per Flavour per Availability Zone

Table 212: edgeResUtilizeMetrics

5.1.1.6.2.3 cpuUtilization

Following table describes information about the cpuUtilization which defines the Availability Zone level CPU utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
cpuType	cpuArchType	M	1	CPU instruction set architecture
noOfSamples	String	M	1	Number of simples tha have been evaluated to determine the metric
averageUtilization	utilizationValue	M	1	Average CPU utilization in percent
maxUtilization	utilizationValue	M	1	Maximum CPU utilization in percent
minUtilization	utilizationValue	M	1	Minimum CPU utilization in percent
effectiveUtilization	utilizationValue	M	1	Actual time equivalent busy CPU in given time Interval (sum of busy period utilization)

Table 213: cpuUtilizeMetrics

5.1.1.6.2.4 memUtilization

Following table describes information about the parameters which defines the Availability Zone level memory utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
noOfSamples	String	M		Number of simples tha have been evaluated to determine the metric
averageUtilization	utilizationValue	M		Average memory utilization in percent
maxUtilization	utilizationValue	M	1	Maximum memory utilization in percent
minUtilization	utilizationValue	M		Minimum memory utilization in percent

Table 214: memUtilizeMetrics

5.1.1.6.2.5 diskUtilization

Following table describes information about the parameters which defines the Availability Zone level disk utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
noOfSamples	String	M		Number of simples tha have been evaluated to determine the metric
averageUtilization	utilizationValue	M		Average disk utilization in percent
maxUtilization	utilizationValue	M	1	Maximum disk utilization in percent
minUtilization	utilizationValue	M		Minimum disk utilization in percent

Table 215: diskUtilizeMetrics

5.1.1.6.2.6 networkUtilization

Following table describes information about the parameters which defines the Availability Zone level network utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
noOfSamples	String	M	1	Number of simples tha have been evaluated to determine the metric
ingressUsage	utilizationValue	M	1	Total bytes received
egressUsage	utilizationValue	M	1	Total bytes transmitted
averageThroughput	utilizationValue	M	1	Average throughput acheived
maxThroughput	utilizationValue	M	1	Maximum throughput acheived
minThroughput	utilizationValue	M	1	Minimum throughput acheived

Table 216: networkUtilizeMetrics

5.1.1.6.2.7 flavourUtilization

Following table describes information about the cpuUtilization which defines the Availability Zone level flavours utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
flavUtilization	Array(flavourMetrics)	M	1..N	Flavour utilization of flavours

Table 217: flavourUtilizeMetrics

5.1.1.6.2.8 flavourMetrics

Following table describes information about the flavour metrics which defines the Availability Zone level flavour utilization metrics.

Attribute Name	Data Type	P	Cardinality	Description
flavourId	String	M	1	Flavour Identifier
noOfSamples	String	M	1	Number of simples tha have been evaluated to determine the metric
averageUtilization	utilizationValue	M	1	Average Flavour utilization per flavour
maxUtilization	utilizationValue	M	1	Maximum Flavour utilization per flavour
minUtilization	utilizationValue	M	1	Minimum Flavour utilization per Flavour

Table 218: flavourMetrics

5.1.1.6.2.9 utilizationValue

Following table describes generic structure to report measured values for different usage types.

Attribute Name	Data Type	P	Cardinality	Description
resType	Enum	M	1	CPU, Memory, Disk, Network, Flavour etc
value	String	M	1	Whole number representing measured value of a given resource
Unit	Enum	M	1	Unit of the measured quantity e.g., Mbps, MB, GB, Percent, Cores etc

Table 219: utilizationValue

5.1.1.6.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

5.1.1.6.3.1 Simple data types

Attribute Name	Data Type	Description
resType	String	CPU, Memory, Disk, Network, Flavour etc

Table 220: Monitoring Resource Types table

5.2 Applications Resource Monitoring APIs

The application resource monitoring information APIs provides the management of the application resource consumption information between the federating OPs.

5.2.1 Application Resource Utilization Management – API

The Applications deployed by the Application Providers of the Leading OP consumes Edge Resources that the Partner OP has offered via its Availability Zones. The set of APIs provides the management of reporting of applications resource utilization information by the Partner OP to the Leading OP.

5.2.1.1 Introduction

Following table describes the HTTP Methods for applications resource utilization reporting services.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Apps Resource Monitoring Info	POST	/operatorplatform/federation/v1/{federationContextId}/subscriptions?res_type=app&mon_type=utilization	M
Unsubscribe Apps Resource Monitoring	DELETE	operatorplatform/federation/v1/{federationContextId}/subscriptions/{subs_id}	M
Report Apps Monitoring Info	POST	{monitorInfoNotificationDest}	M
Retrieve Apps Monitoring Information	GET	operatorplatform/federation/v1/{federationContextId}/subscriptions/{subs_id}	O

Table 221: Application resource utilization management HTTP Methods

5.2.1.2 POST Method: Subscribe Applications Resource Utilization

The HTTP POST request from the Originating OP contains the following parameters towards the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
monitoringSubsType	M	1	Represent the subscription type for edge applications resource monitoring information

Table 222: Monitoring Information subscription parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
subscriptionId	C	200	Monitoring subscription created
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 223: Subscribe Applications Resource Utilization response parameters

5.2.1.3 DELETE Method: Unsubscribe Applications Resource Utilization

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP DELETE request.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship.
subscriptionId	M	1	Application Monitoring information subscription identifier at the Leading OP provided by the Partner OP

Table 224: Unsubscribe Applications Resource Utilization request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
Status	C	200	Subscription removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 225: Unsubscribe Applications Resource Utilization response parameters

5.2.1.4 GET Method: Retrieve Applications Resource Utilization Information

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP GET request.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to the Partner OP to identify the existing Federation relationship. It is a HTTP path parameter
subscriptionId	M	1	Subscription identifier earlier shared by the Partner OP with the Leading OP

Table 226: Retrieve applications resource utilization request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
appsResUtilizeInfo	C	200	Object representing Apps resource utilization information
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 227: Applications resource utilization response parameters

5.2.1.5 POST Method: Reporting Applications Resource Utilization Information

The HTTP POST request sent by the Partner OP contains the following parameters towards the Originating OP to report applications resource utilization report.

Parameter Name	P	Cardinality	Description
subscriptionId	M	1	Subscription identifier earlier shared by the Partner OP with the Leading OP
appsResUtilizeInfo	M	1	Object representing Apps resource utilization information

Table 228: Applications resource utilization request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Application resource monitoring Info notification Acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 229: Applications resource utilization response parameters

5.2.1.6 Data Model

5.2.1.6.1 General

This section provides the data types for the managing applications resource utilization reports.

5.2.1.6.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

5.2.1.6.2.1 appsResUtilizeInfo

Following table describes information about the application resource utilization metrics collected by the Partner OP about the applications deployed by the Leading OP.

Attribute Name	Data Type	P	Cardinality	Description
appsResUtilizeInfo	Array(appsResUtilizeMetrics)	M	1	Contains the applications resource consumption information on various Availability Zones of the Partner OP

Table 230: appsResUtilizeInfo

5.2.1.6.2.2 appsResUtilizeMetrics

Following table describes information about the parameters which defines the Availability Zone level resource utilization metrics of the edge applications.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone Identifier
startTime	String	M	1	Date-time indicating the start of the measurement period
endTime	String	M	1	Date-time indicating the end of the measurement period offset by measurement periodicity from startTime
appMetrics	Array(appAggrResUtil)	M	1	

Table 231: appsResUtilizeMetrics

5.2.1.6.2.3 appAggrResUtil

Following table describes information about the parameters which defines the Availability Zone level resource utilization metrics of an edge application.

Attribute Name	Data Type	P	Cardinality	Description
appId	String	M	1	Application Identifier
appProviderId	String	M	1	Application Provider Identifier
noOfAppInstances	String	M	1	Number of application instances in this Availability Zone
appInstances	Array(appInstanceId)	C	1	List of instances identifier for a given appId
cpuUtil	cpuUtilization	M	1	CPU utilization metrics for a given Availability Zone for a given appId
memUtil	memUtilization	M	1	Memory utilization metrices for a given Availability Zone for a given appId

Attribute Name	Data Type	P	Cardinality	Description
diskUtil	diskUtilization	M	1	Disk utilization metrics for a given Availability Zone for a given appId
networkUtil	networkUtilization	M	1	Network utilization metrics for a given Availability Zone for a given appId
flavourUtil	flavourUtilization	M	1	Flavour utilization metrics per Flavour per Availability Zone for a given appId

Table 232: appMetrics

6 Operational Visibility APIs

The resource monitoring APIs provide the Partner OP to expose Edge Resource consumption information by the applications shared over E/WBI by the Leading OP. The information comprises

6.1 Fault Management APIs

This section provides the details of the HTTP methods for management of alarms generated at the Partner OP and reported over E/WBI to the Originating OP.

6.1.1 Alarms Management – API

This section details the alarm management requests and notifications between the Originating OP and the Partner OP to share the events that can affect the level of offered services by the Partner OP to the Originating OP Application Providers.

6.1.1.1 Introduction

Following table describes the various HTTP methods for alarm management service over E/WBI.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Alarm Reporting	POST	/operatorplatform/federation/v1/{federationContextId}/alarms	M
Unsubscribe Event Monitoring	DELETE	operatorplatform/federation/v1/{federationContextId}/alarms/{alarm_subs_id}	M
Raise Alarm	POST	{listnerAlarms}	O
Update Alarm	PATCH	{listnerAlarms}	O
Clear Alarm	DELETE	{listnerAlarms}	O
Retrieve Alarm	GET	/operatorplatform/federation/v1/{federationContextId}/alarm	C

Table 233: Alarm Management HTTP Methods

6.1.1.2 POST Method: Raise Alarm

The HTTP POST request contains the following parameters towards the Originating OP.

Parameter Name	P	Cardinality	Description
alarmType	M	1	The categories for alarm classification are defined in X.733 8.1.1 and 3GPP TS 32.111-2 Annex A. The importan ones from OP perspective e.g., processingErrorAlarm, qualityOfServiceAlarm, securityService etc.
perceivedSeverity	M	1	The values are consistent with ITU-T Recommendation X.733. E.g., Critical, Major, Minor, Warning etc
probableCause	M	1	Provides the probable cause of the alarm
alarmedObject	M	1	Identifies the managed object instance associated with the alarm
sourceSystemId	M	1	Source system identity
state	M	1	Defines the alarm state during its life cycle (raised | updated | cleared).
alarmRaisedTime	M	1	A date time (DateTime). Indicates the time (as a date + time) at which the alarm occurred at its source
affectedService	O	1	A list of affected services. For OP they could be Application Lifecycle Management, Edge Discovery etc
alarmDetails	O	1	Contains further information on the alarm
specificProblem	O	1	Provides more specific information about the alarm
serviceAffecting	M	1	Indicates whether the alarm affects service or not

Table 234: Raise Alarm request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
Id	M	1	Identifier of the alarm
alarmType	O	1	Categorize the alarm
State	O	1	Defines the alarm state during its life cycle

Table 235: Raise Alarm response parameters

The following table describes the data structures supported by the response codes on this resource.

Parameter Name	P	Response Codes	Description
alarmState	C	200	Alarm received successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 236: Raise Alarm response parameters

6.1.1.3 DELETE Method: Clear Alarm

Following table provides parameters which the Partner OP sends to the Originating OP in HTTP DELETE request to clear an existing alarm.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
id	M	1	Identifier of the alarm. It is a HTTP URI path parameter

Table 237: Clear Alarm request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
Status	C	200	Alarm clear Acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error

Parameter Name	P	Response Codes	Description
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 238: Clear Alarm response parameters

6.1.1.4 GET Method: Retrieve Alarms

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP GET request to retrieve one or more alarms.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
Id	O	1	Identifier of the Alarm

Table 239: Alarm retrieval request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
activeAlarms	C	200	Alarms information retrieved successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 240: Alarm information retrieval response parameters

6.1.1.5 PATCH Method: Update Alarm

The HTTP PATCH request sent by the Partner OP contains the following parameters towards the Originating OP to update patchable parameters of existing alarm.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
Id	M	1	Alarm identifier.
updateAlarmParams	M	1	Parameters of the alarm require Update

Table 241: Alarm update request parameters

The following table describes the response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Alarm Parameters updated successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 242: Update Alarm response parameters

6.1.1.6 Data Model

6.1.1.6.1 General

This section provides the data types for the Alarm Management API.

6.1.1.6.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

6.1.1.6.2.1 patchableParams

Following table describes parameters that can be updated for an existing alarm.

Attribute Name	Data Type	P	Cardinality	Description
perceivedSeverity	Note 1	O	1	The values are consistent with ITU-T Recommendation X.733. E.g., Critical, Major, Minor, Warning etc
probableCause	Note 1	O	1	Provides the probable cause of the alarm
alarmedObject	Note 1	O	1	Identifies the managed object instance associated with the alarm
state	Note 1	O	1	Defines the alarm state during its life cycle (raised | updated | cleared).
affectedService	Note 1	O	1	A list of affected services. For OP they could be Application Lifecycle Management, Edge Discovery etc
alarmDetails	Note 1	O	1	Contains further information on the alarm
specificProblem	Note 1	O	1	Provides more specific information about the alarm
serviceAffecting	Note 1	O	1	Indicates whether the alarm affects service or not
Note 1: Date types refers to table “Simple data types for Alarm Management”				

Table 243: patchableParams

6.1.1.6.2.2 updatedParam

Following table describes data structure to update patchable parameters for an existing alarm.

Attribute Name	Data Type	P	Cardinality	Description
Op	String	M	1	Operation to be performed on alarm parameters. Currently “replace” operation is valid
path	patchableParams	M	1	Parameter to be updated
Value	String	M	1	New value of the parameter for “path” attribute

Table 244: updatedParam

6.1.1.6.2.3 updateAlarmParams

Following table describes data structure to update patchable parameters for an existing alarm.

Attribute Name	Data Type	P	Cardinality	Description
updateAlarmParams	Array(updatedParam)	M	1..N	List of parameters and operation to be performed

Table 245: updateAlarmParams

6.1.1.6.2.4 alarmedObject

The alarmedObject resource Identifies the managed object instance associated with the alarm.

Attribute Name	Data Type	P	Cardinality	Description
id	String	M	1	Identifies the managed object instance associated with the alarm
Href	String	O	1	A reference to the managed object associated with the event

Table 246: alarmedObject

6.1.1.6.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

6.1.1.6.3.1 Simple data types

Attribute Name	Data Type	Description
alarmType	String	The categories for alarm classification are defined in X.733 8.1.1 and 3GPP TS 32.111-2 Annex A. The importan ones from OP perspective e.g., processingErrorAlarm, qualityOfServiceAlarm, securityService etc.
perceivedSeverity	String	The values are consistent with ITU-T Recommendation X.733. E.g., Critical, Major, Minor, Warning etc
probableCause	String	Provides the probable cause of the alarm
alarmedObject	String	Identifies the managed object instance associated with the alarm
sourceSystemId	String	Source system identity
state	String	Defines the alarm state during its life cycle (raised | updated | cleared).
alarmRaisedTime	String	A date time (DateTime). Indicates the time (as a date + time) at which the alarm occurred at its source
affectedService	String	A list of affected services. For OP they could be Application Lifecycle Management, Edge Discovery etc
alarmDetails	String	Contains further information on the alarm
specificProblem	String	Provides more specific information about the alarm

Attribute Name	Data Type	Description
serviceAffecting	String	Indicates whether the alarm affects service or not

Table 247: Simple data types for Alarm Management

6.1.1.6.3.2 Enumeration: alarmType

The following table enlist enumerations of the alarm categories.

Enumeration value	Description
EDGERES	In relation to Edge Resources
APPLICATION	In relation to edge applications
ARTEFACT	In relation to application artefacts
EDGEDISC	In relation to Edge Discovery Service
FEDERATION	In relation to Federation Service
SECURITY	In relation to application security
APIFEDERATION	In relation to Service APIs Federation
FILE	In relation to Applications file management

Table 248: alarmType

6.2 Event Management APIs

APIs for management of events.

6.2.1 Events Management – API

This section describes the HTTP methods and parameters that are used by the federating OPs to create events of interests dynamically and notify the Partner OPs on detection of those events.

6.2.1.1 Introduction

Following table describes the APIs for management of events associated to Edge Resources of the Availability Zones offered by the Partner OP to the Leading OP.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Event Monitoring	POST	/operatorplatform/federation/v1/{federationContextId}/events	M
Unsubscribe Event Monitoring	DELETE	operatorplatform/federation/v1/{federationContextId}/event/{event_subs_id}	M
Create Event Criteria	POST	/operatorplatform/federation/v1/{federationContextId}/event-criteria	O
Delete Event Criteria	DELETE	/operatorplatform/federation/v1/{federationContextId}/event-criteria/{criteria-id}	O
Report Event	POST	{listnerEvents}	O

Operations	HTTP Method	Resource URI	Qualifier
Retrieve Event Criterions	GET	/operatorplatform/federation/v1/{federationContextId}/event-criteria	

Table 249: Event Management HTTP Methods

6.2.1.2 POST Method: Subscribe Event Notifications

The HTTP POST request contains the following parameters towards the Partner OP to create an event subscription.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information
listnerEvents	M	1	Callback URI where the the Originating OP listen for event information from Partner OP

Table 250: Create Event Subscription request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information
listnerEvents	M	1	Callback URI where the the Originating OP listen for event information from Partner OP

Table 251: Create Event Subscription response parameters

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 252: Event Subscription Response Error Codes

6.2.1.3 POST Method: Create Event Criteria

The HTTP POST request contains the following parameters towards the Partner OP to create an event object.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
resType	M	1	Represent the object cateogry of the event to be created e.g., CPU, Memory, Network, Storage, API etc
eventCriteria	M	1	Criteria defines the conditions about the given event that the Partner OP can monitor and when the condition is satisfied it notifies the event occurance to the Originating OP

Table 253: Create Event request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
eventCriterionInfo	M	1	Event Criterion Identifier and assosiated event Info as provided for event criteria

Table 254: Create Event response parameters

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
eventCriterionInfo	C	200	Event Criterion Created
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 255: Create Event Criterion response parameters

6.2.1.4 DELETE Method: Remove Event Criteria

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP DELETE request to delete the event object earlier created using event create request.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
eventCriterionId	M	1	Identifier assigned to the given event criteria

Table 256: Remove Event request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
eventCriterionInfo	C	200	Event Criterion removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 257: Remove Event response parameters

6.2.1.5 GET Method: Retrieve Events

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP GET request to retrieve the list of events earlier created by the Originating OP.

Parameter Name	P	Cardinality	Description
resUsageType	O	1	Represent the object cateogry of the event to be created e.g., CPU, Memory, Network, Storage, API etc
eventId	O	1	String, Identifier assigned to the given event criteria

Table 258: Retrieve Events request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
eventsList	C	200	Events information retrieved successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 259: Retrieve Events response parameters

6.2.1.6 POST Method: Notify Events

The HTTP POST request sent by the Partner OP contains the following parameters for detected events towards the Originating OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
eventsList	M	1	One or more events that has been detected by the Partner OP

Table 260: Notify Events request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Notify events acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 261: Notify Events response parameters

6.2.1.7 Data Model

6.2.1.7.1 General

This section provides the data types for the Events Management API.

6.2.1.7.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

6.2.1.7.2.1 eventCriterion

Following table describes information about the event criterion which the Partner OP has created on request from the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
resUsageType	String	M	1	E.g., CPU, Memory, Disk, Network etc
triggerCondition	String	M	1	Comparison operator e.g., greater than(GT), equal to(EQ), less than(LT), etc
thresholdVal	Object	M	1	The threashold value to compare with resource metrics indicated by usage type
numOccurance	String	M	1	Number of times the trigger condition to be detected

Attribute Name	Data Type	P	Cardinality	Description
monitorDuration	Object	M	1	Time duration in which the trigger condition to be monitored

Table 262: eventCriterion

6.2.1.7.2.2 eventCriterionInfo

The events list represents the event information object.

Attribute Name	Data Type	P	Cardinality	Description
eventCriterionId	String	M	1	Identifier of the event object
eventCriterion	Object	M	1	Event criterion

Table 263: eventTypeInfo data structure

6.2.1.7.2.3 eventInfo

The events list represents the event information object.

Attribute Name	Data Type	P	Cardinality	Description
eventId	String			Identifier of the event object
detectedEvent	Object			Details of the detected Event

Table 264: eventInfo data structure

6.2.1.7.2.4 eventsList

The events list represents the events list.

Attribute Name	Data Type	P	Cardinality	Description
eventsList	Array(eventInfo)	M	1..N	List of event objects

Table 265: eventsList data structure

6.2.1.7.2.5 detectedEvent

The details of the event that is detected as per the event criterion.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	The Availability Zone identifier of the detected event
eventId	String	M	1	The eventId associated to event criterion agains which the given event is detected
startTime	Date-time	M	1	The timestamp at which event detection started
endTime	Date-time	M	1	The timestamp at which event detection ended

Attribute Name	Data Type	P	Cardinality	Description
numOccurance	integer	M	1	Number of times the given event detected in the time period between startTime and endTime

Table 266: detectedEvent data structure

6.2.1.7.2.6 thresholdVal

It indicates the threshold value of a resource type that needs to be compared to detect the given event criteria at the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
Value	String	M	1	Positive Number,that represent the usage of a given resource
Unit	Enum	M	1	“Percent”, “vCPU”, “TBytes”, “Mbps” etc

Table 267: thresholdVal

6.2.1.7.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

6.2.1.7.3.1 Enumeration: resUsageType

The enumeration resUsageType represents the resource usage types supported on which the event criterion can be created.

Enumeration value	Description
CPU	Represent the CPU metrics
MEMORY	Represent the Memory metrics
DISK	Represent the Disk/Storage metrics
NETWORK	Represent the Networking metrics
GPU	Represent the GPU metrics
FLAVOUR	Represent the Flavour metrics

Table 268: resType

7 Notification Management APIs

The notification management APIs enables the Partner OP to share the information about the network capabilities which it may applies to the applications of by the Leading OP.

7.1 Network Events Notification context Management APIs

APIs for management of network events that a Partner OP monitors and can report to the Originating OP.

7.1.1 Network Events Management – API

This section describes the HTTP methods and parameters that are used by the Partner OP to manage reporting of network events of interests by the Originating OP on detection of those events in the Partner OP environment.

7.1.1.1 Introduction

Following table describes the APIs for management of reporting of network events associated to network capabilities.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Network Event Notifications	POST	/operatorplatform/federation/v1/{federationContextId}/network-events-notifications	M
Unsubscribe Notification Monitoring	DELETE	operatorplatform/federation/v1/{federationContextId}/network-events-notifications/{notif-subs-id}	M
Add Network Capability Event	POST	operatorplatform/federation/v1/{federationContextId}/network-events-notifications/{notif-subs-id}/nw-caps	O
Delete Network Capability Event	DELETE	operatorplatform/federation/v1/{federationContextId}/network-events-notifications/{notif-subs-id}/nw-caps//{nw-caps-id}/	O
Report Network Capability Events	POST	{listnerEvents}	M
Retrieve Network Capability Events	GET	operatorplatform/federation/v1/{federationContextId}/network-events-notifications/{notif-subs-id}/nw-caps	M

Table 269: Network Event Management HTTP Methods

7.1.1.2 POST Method: Subscribe Network Event Notifications

The HTTP POST request contains the following parameters towards the Partner OP to create network event subscription.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information
listnerNetworkEvents	M	1	Callback URI where the the Originating OP listen for network event information from Partner OP

Table 270: Create Event Subscription request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information
listnerNetworkEvents	M	1	Callback URI where the the Originating OP listen for network event information from Partner OP

Table 271: Create Network Event Subscription response parameters

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 272: Network Event Subscription Response Error Codes

7.1.1.3 POST Method: Add Network Event For Monitoring

The HTTP POST request contains the following parameters towards the Partner OP to create a network event monitoring object.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
networkCapMonitorInfo	M	1	Details like capabilityID, applicationId, duration, Availability Zone etc
appId	M	1	Application Identifier
appProviderId	M	1	Application Provider Identifier
transactionId	M	1	Transaction Identifier

Table 273: Create Network Event request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
networkCapSubsInfo	M	1	Network Event information registered at Partner OP
txnIdentifier	M	1	Transaction Identifier

Table 274: Add Network Event Subscription response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 275: Network Event Subscription response parameters

7.1.1.4 DELETE Method: Remove Network Event Subscription

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP DELETE request to delete the network event object earlier created using event create request.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
networkCapMonitorInfo	M	1	Identifier assigned to the given network event

Table 276: Remove Network Event request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
networkCapMonitorInfo	C	200	Network Event removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access

Parameter Name	P	Response Codes	Description
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 277: Remove Network Event response parameters

7.1.1.5 GET Method: Retrieve Network Monitoring Info

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP GET request to retrieve the list of network events collected by the Partner OP.

Parameter Name	P	Cardinality	Description
networkEventsMonitorInfo	O	1..N	Represent the aggregated network event information for all the Capabilities registered

Table 278: Retrieve Events request parameters

The following table describes the data structures supported by the GET Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
networkEventsList	C	200	Network Events information retrieved successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 279: Retrieve Network Events response parameters

7.1.1.6 POST Method: Notify Network Events

The HTTP POST Notification sent by the Partner OP contains the following parameters for detected network events associated to the network capabilities supported at Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
networkEventsList	M	1	One or more network events that has been detected by the Partner OP

Table 280: Notify Network Events request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Notify events acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 281: Notify Network Events response parameters

7.1.1.7 Data Model

7.1.1.7.1 General

This section provides the data types for the Events Management API.

7.1.1.7.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

7.1.1.7.2.1 networkCapMonitorInfo

Following table describes information about the network event criterion which the Partner OP has created on request from the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
capabilityID	String	M	1	Represents network capability Identifier
appId	String	M	1	Application Identifier
appProviderId	String	M	1	Application Provider Identifier
appInstanceId	String	M	1	Application Instance Identifier
monitorDuration	Object	M	1	Time duration in which the trigger condition to be monitored

Table 282: networkCapMonitorInfo

7.1.1.7.2.2 networkEventInfo

The events list represents the event information object.

Attribute Name	Data Type	P	Cardinality	Description
networkEventId	String			Identifier of the event object
networkCapInvoked	Object			Details of the detected Event

Table 283: networkEventInfo data structure

7.1.1.7.2.3 networkEventsList

The events list represents the events list.

Attribute Name	Data Type	P	Cardinality	Description
networkEventsList	Array(networkCapInvoked)	M	1..N	List of event objects

Table 284: networkEventsList data structure

7.1.1.7.2.4 networkCapInvoked

The details of the event that is detected as per the event criterion.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	The Availability Zone identifier of the detected event
capabilityID	String	M	1	The network capability ID
invocationTime	Date-time	M	1	The timestamp at which network capability applied or invoked
appId	String	M	1	Application Identifier
appProviderId	String	M	1	Application Provider Identifier
appInstanceId	String	M	1	Application Instance Identifier

Attribute Name	Data Type	P	Cardinality	Description
nwCapabilitySLI	String	C	1	Observed SLI for given capabilityID

Table 285: detectedNetworkEvent data structure

7.2 Application-level Events Notification Management API

This section describes various HTTP methods and resources for the management of application-level events that a Partner OP monitors and can report to the Originating OP.

7.2.1 Application-level Events Management – API

This section describes the HTTP methods and parameters that are used by the Partner OP to manage reporting of application-level events subscribed by the Originating OPs on detection of those events in the context of federated applications in the Partner OP environment.

7.2.1.1 Introduction

Following table describes the APIs for management of reporting of application-level events associated to federated applications.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Applications Event Notifications	POST	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications	O
Update Applications Event Notifications	PATCH	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{app-notif-subs-id}	C
Unsubscribe Application Events Monitoring	DELETE	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{app-notif-subs-id}	C
Add Application-level Event Type	POST	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{notif-subs-id}	C
Retrieve Application-level Event Types	GET	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{notif-subs-id}	C
Delete Application-level Event Type	POST	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{notif-subs-id}/cancel	C
Report Application-level Events	POST	{listnerEvents}	C
Retrieve Application-level Events	POST	/operatorplatform/federation/v1/{federationContextId}/appl-event-notifications/{notif-subs-id}/app-events	C

Table 286: Application-level Event Management HTTP Methods

7.2.1.2 POST Method: Create Subscription for Applications Events Notification

The HTTP POST request contains the following parameters towards the Partner OP to create an application-level events notification subscription.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information
listnerApplEvents	M	1	Callback URI where the the Originating OP listen for applications events information from the Partner OP
numEvtsPerNotif	O	1	The number of applications events that the Partner OP should include in a notification

Table 287: Application-level Event Subscription request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
appEventSubsId	M	1	Application-level event subscription identifier resource
subscriptionsPeriodicity	M	1	Periodicity at which the Partner OP is requested to send event information

Table 288: Aplication-level Event Subscription response parameters

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 289: Application-level Event Subscription Response Error Codes

7.2.1.3 PATCH Method: Modify Subscription Information for Applications Events

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP PATCH request to modify the subscription meta-information about application-level notifications e.g. enable/disable notifications, periodicity of subscription etc.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	Federation context Identifier
appEventSubsId	M	1	Application-level event subscription identifier resource
generateNotification	O	1	Indicate if the Partner OP can start sending the notifications or not (True/False)
appEvtSubsPeriodicity	O	1	The periodicity at which the Partner OP is publishing the application events reports

Table 290: Modify Application-level Events request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource for an HTTP 200 response.

Parameter Name	P	Response Codes	Description
applEventSubsInfo	C	200	Applications Events information updated successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 291: Modify Application-level Events response parameters

7.2.1.4 GET Method: Retrieve Subscription Information for Applications Events

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP GET request to retrieve the subscription meta-information about application-level notifications e.g. duration of subscription, start timestamp, number of applications subscribed, last report sent timestamp etc.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	Federation context Identifier
appEventSubsId	M	1	Application-level event subscription identifier resource

Table 292: Retrieve Application-level Events request parameters

The following table describes the data structures supported by the POST Response Body on this resource for an HTTP 200 response.

Parameter Name	P	Response Codes	Description
applEventsSubsInfo	C	200	Applications events subscription information retrieved successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 293: Retrieve Application-level Events response parameters

7.2.1.5 DELETE Method: Remove Subscription for Applications Events Notifications

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP DELETE request to delete the subscription for application-level events reporting.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
appEventSubsId	M	1	Subscription Identifier assigned by the Partner OP for application-level events reporting

Table 294: Remove Application Event request parameters

The following table describes the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
appEventSubsId	C	200	Applications Event subscription removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 295: Remove Application Event response parameters

7.2.1.6 POST Method: Add Applications To Applications Notifications Subscription

The HTTP POST request contains the following parameters towards the Partner OP to add applications that the Originating OP wants to be notified on with the application-level events reports.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
addAppsForNotif	M	1	Set of appName, applicationId, Availability Zone etc
transactionId	M	1	Transaction Identifier

Table 296: Create Application Event Type Parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
applInfo	M	1	Applications registered at Partner OP for application-level events reporting
txnIdentifier	M	1	Transaction Identifier

Table 297: Add Application Event Subscription response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.

Parameter Name	P	Response Codes	Description
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 298: Add Application Event Type Errors

7.2.1.7 POST Method: Remove Applications From Applications Notification Context

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP POST request to delete the application-level events reporting for one or more applications.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
applInfo	M	1	Set of applications to be removed from events reporting

Table 299: Remove Application From Event Reporting Parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
applInfo	C	200	Applications removed from event reporting successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.

Parameter Name	P	Response Codes	Description
problemDetails	C	520	Web Server Returned an Unknown Error

Table 300: Remove Applications From Event Reporting Errors

7.2.1.8 GET Method: Retrieve Subscribed Applications Information From Notification Context

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP POST request to retrieve the list of applications subscribed for application-level events reporting.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
appInfo	O	1..N	List of one or more applications in various Availability Zones

Table 301: Retrieve Application-level Events request parameters

The following table describes the data structures supported by the GET Response Body on this resource for a HTTP 200 response.

Parameter Name	P	Response Codes	Description
applEventInfo	C	200	Applications information retrieved successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 302: Retrieve Application-level Events response parameters

7.2.1.9 POST Method: Applications Events Reporting Notification

The HTTP POST Notification sent by the Partner OP contains the following parameters for detected application-level events associated with the set of applications at the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
applEventsList	M	1	One or more applications events reports that the Partner OP may have observed

Table 303: Notify Applications Events request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Notify events acknowledged
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 304: Notify Applications Events response parameters

7.2.1.10 POST Method: Retrieve Application-level Events Information

The HTTP POST request sent by the Originating OP contains the following parameters for retrieval of application-level events associated with the set of applications at the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
applInfo	M	1	One or more applications for which event information list is to be retrieved

Table 305: Applications Events request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
applEventsList	C	200	Application events information

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 306: Applications Events response parameters

7.2.1.11 Data Model

7.2.1.11.1 General

This section provides the data types for the Applications Events Management API.

7.2.1.11.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

7.2.1.11.2.1 applEventsSubsInfo

Following table describes the event subscription information for the application-level events.

Attribute Name	Data Type	P	Cardinality	Description
appEventSubsId	String	M	1	Application Events Subscription Identifier
appEvtSubsStartTime	String	M	1	Represents subscription start time
appEvtSubsLastReportTime	String	M	1	Represents time of the last notification
appEvtSubsNumApps	String	M	1	Total number of applications subscribed for events reporting
appEvtSubsPeriodicity	String	M	1	Periodicity of the notifications generated by the Partner OP

Table 307: applEventsSubsInfo

7.2.1.11.2.2 aggrApplEventsList

Following table describes information that the Partner OP provides about the application-level events.

Attribute Name	Data Type	P	Cardinality	Description
startTime	String	M	1	Start timestamp of the event collection
endTime	String	M	1	End timestamp of the event collection
aggrAppsEventsList	Array(applEventsList)	M	1..N	Represents aggregated application event list for one or more applications

Table 308: aggrApplEventsList

7.2.1.11.2.3 applEventsList

Following table describes information that the Partner OP provides about the application-level events.

Attribute Name	Data Type	P	Cardinality	Description
appId	String	M	1	Application Identifier
appProviderId	String	M	1	Application Provider Identifier
aggrApplEvents	Array(zoneLevelApplEvents Container)	M	1..N	Application events across one or more Availability Zones for the given application

Table 309: applEventsList

7.2.1.11.2.4 zoneLevelApplEventsContainer

The events list represents the event information object.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Availability Zone Identifier
appsEventsLi7st	Array(applInst EventsList)	M	1..N	Represent list of application Instance and assosiated events

Table 310: zoneLevelApplEventsList

7.2.1.11.2.5 applInstEventsList

The events list represents the event information object.

Attribute Name	Data Type	P	Cardinality	Description
appInstanceEventsList	Array(applInstEventsContainer)	M	1..N	Represent list of application Instance and assosiated events

Table 311: applInstEventsList

7.2.1.11.2.6 applInstEventsContainer

The events list represents the application-level events information object for a single application instance.

Attribute Name	Data Type	P	Cardinality	Description
appInstanceId	String	M	1	Application Instance Identifier
appInstEventsList	Array(applInstEventTypeInfo)	M	1	List of application events for a given application instance

Table 312: applInstEventsContainer

7.2.1.11.2.7 applInstEventTypeInfo

The events types list represents the application-level event types supported in various Availability Zones by the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
applInstEvent	appEventTypes	M	1	Represents application event
applInstEventCount	String	M	1	Represents number of occurance of given event

Table 313: applInstEventTypeInfo data structure

7.2.1.11.2.8 addAppsForNotif

The applications and event types list represents the application-level events for various applications for which the Partner OP should send application events notifications.

Attribute Name	Data Type	P	Cardinality	Description
appId	String	M	1	Represents application event
appprovId	String	M	1	Represents number of occurance of given event
appZones	Array(zoneId)	M	1..N	List of Availability Zones
appEvents	Array(appEventTypes)	O	0..N	List of application-level events to be notified

Table 314: addAppsForNotif data structure

7.2.1.11.3 Simple data types and enumerations

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

7.2.1.11.3.1 Simple data types

Attribute Name	Data Type	Description
subscriptionsPeriodicity	String	Application events notification periodicity in combination of hours and minutes
listnerApplEvents	String	Callback URI at the Originating OP for receiving applications event notifications from the Partner OP
appEvtSubsStartTime	String	Start time of the subscritpion in Date and Time zone Info format as defined in RFC 3339
appEvtSubsNumApps	String	Number of applications for which application events being reported by the Partner OP

Table 315: Application Events Simple Datatype table

7.2.1.11.3.2 Enumeration: appEventTypes

The enumeration appEventTypes represents the application-level events supported by the Partner OP on in context of federated applications.

Enumeration value	Description
evt_type_app_relocation	Application relocations events
evt_type_app_session_cont	Application Session Continuity events
evt_type_app_restarts	Application restarts events
evt_type_app_upscale	Application upscaling events
evt_type_app_downscale	Application downscaling events

Table 316: Application-Level Event Types

8 Resource Control Policies Management API

The resource control policy management APIs enables the Originating OP to request Partner OP to perform the actions at operation or application level that conforms to the rules defined as part of a given policy created by the Partner OP.

8.1 Application-Level Policy Management

This section describes the various HTTP methods for management of the application-level policies that a Partner OP may share with the Originating OP. The Originating OP may request the Partner OP to attach such policies to one or more applications onboarded earlier with the Partner OP.

8.1.1 Application-Level Policy Methods

This section describes the HTTP methods and parameters that describes the various HTTP methods for application-level policies management.

8.1.1.1 Introduction

Following table describes the HTTP resources for management of application-level policies.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Application Policy	POST	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription	
Retrieve Application Policy Templates	GET	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription /{appl-policy-subs-id}/app-policy-templates	C
Register Application Policy	POST	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription /{appl-policy-subs-id}/app-policy-registration	C
Retrieve Application Policy	GET	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription /{appl-policy-subs-id}/app-policy-	
Apply Application Policies	POST	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription /{appl-policy-subs-id}	C
Modify Application Ploicies	PATCH	/operatorplatform/federation/v1/{federation ContextId}/app-policies-subscription /{appl-policy-subs-id}	C
Remove Application Policies	POST	/operatorplatform/federation/v1/{federation ContextId}//app-policies-subscription /{appl-policy-subs-id}/app-policy-cancel	C

Table 317: Application-Level Policies Management HTTP Methods

8.1.1.2 POST Method: Subscribe Application-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to create subscription for application-level policies for federated applications.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.

Table 318: Subscribe Application-level Policies Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
applPolicySubscriptionId	M	1	Application-level policy subscription identifier

Table 319: Subscribe Application-level Policies response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 320: Subscribe Application-level Policies response parameters

8.1.1.3 GET Method: Retrieve Application-level Policy Templates

The HTTP GET request contains the following parameters towards the Partner OP to retrieve application-level policies templates supported by the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
applPolicySubscriptionId	M	1	Application-level policy subscription identifier

Table 321: Retrieve Application-level Policies Request Parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
applPolicyTemplateList	M	1	List of application-level policy templates supported by the Partner OP

Table 322: Retrieve Application-level Policy Templates Response Parameters

The following table describes the HTTP response codes supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict

Parameter Name	P	Response Codes	Description
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 323: Retrieve Application-level Policy Templates Error Codes

8.1.1.4 POST Method: Register Application-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to register application-level policies for federated applications.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
applConcretePolicy	M	1	Application policy with policy parameter values against policy template supported by Partner OP

Table 324: Register Application-level Policy Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
applPolicySubscriptionId	M	1	Application-level policy subscription identifier
applConcretePolicy	M	1	Application policy with policy parameter values against policy template supported by Partner OP

Table 325: Register Application-level Policy response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.

Parameter Name	P	Response Codes	Description
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 326: Register Application-level Policies response parameters

8.1.1.5 GET Method: Retrieve Application-level Policies

The HTTP GET request by the Originating OP contains the following parameters towards the Partner OP to retrieve application-level policies registered earlier.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.

Table 327: Retrieve Application-level Policies Request Parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
applPoliciesList	M	1	List of application-level policies supported by the Partner OP

Table 328: Retrieve Application-level Policies Response Parameters

The following table describes the HTTP response codes supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 329: Retrieve Application-level Policies Error Codes

8.1.1.6 POST Method: Apply Application-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to associate application-level policies for set of applications.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocApplPolicies	M	1	Represent Association of Applications with application-level policies
txnIdentifier	M	1	Transaction Identifier as idempotent operation key

Table 330: Apply Application-level Policies Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
assocApplPolicies	M	1	Applications information and the application-level policies applied on them
txnIdentifier	M	1	Transaction Identifier

Table 331: Apply Application-level Policies response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 332: Apply Application-level Policies response parameters

8.1.1.7 POST Method: Remove Application-level Policies

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP POST request to remove the application-level policies with set of applications.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocApplPolicies	M	1	Applications policies information for removing application-level policies
txnIdentifier	M	1	Transaction Identifier

Table 333: Remove Application Policies request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
applPolicies	C	200	Application policies removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized Access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 334: Remove Application Policies response parameters

8.1.1.8 PATCH Method: Modify Application-level Policies

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP PATCH request to modify existing application policies at the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocApplPolicies	M	1	Applications information and the application-level policies for modification
txnIdentifier	M	1	Transaction Identifier

Table 335: Modify Application Policies request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource for a HTTP 200 response.

Parameter Name	P	Response Codes	Description
assocApplPolicies	C	200	Application Policies modified successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 336: Modify Application Policies response parameters

8.1.1.9 Data Model

8.1.1.9.1 General

This section provides the data types for the Events Management API.

8.1.1.9.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

8.1.1.9.2.1 appPolicyTemplate

The events list represents the application-level policy template object.

Attribute Name	Data Type	P	Cardinality	Description
applPolicyName	String	M	1	Application-level policy Name
applPolicyId	String	M	1	Application-level Policy Identifier
applPolicyType	String	M	1	Indicate what aspecto of the application behavior policy aim to influence i.e., static about application resource pool or Dynamic about application instantiation
applPolicyScope	String	M	1	Represent if policy is specific to the Availability Zone or applies to all Availability Zones i.e., local or global
applPolicyDescription	String	M	1	Brief text describing the policy intent

Attribute Name	Data Type	P	Cardinality	Description
applPolicyRules	Array(applPolicyRule)	M	1..N	Policy specific rules to indicate the conditions and actions that applies to the applications on policy rules execution

Table 337: appPolicyTemplate data structure

8.1.1.9.2.2 applPolicyTemplateList

The events list represents the list of application policy templates object.

Attribute Name	Data Type	P	Cardinality	Description
applPolicyTemplateList	Array(applPolicyTemplate)	M	1..N	List of applPolicyTemplate

Table 338: appPolicyTemplateList data structure

8.1.1.9.2.3 applPolicyRule

Fillowing table describes the elements that comprises an application policy.

Attribute Name	Data Type	P	Cardinality	Description
ruleLHSParamType	String	M	1	Represent the application property e.g., “App.Metadata.QoS.Latency”, “App.Metadata.Compute.GPU” etc
ruleOperator	ruleOperatorType	M	1	Operations that can be applied on ruleParameter e.g., EQ(EQual), LT(Less Than), GT(Greater Than),
ruleRHSParamVal	Object	M		Permitted type specific value objects for types in ruleLHSParamType
ruleAction	Object	M	1	The pre-defined action that the Partner OP takes on application of given rule
ruleDescription	String	M	1	Text describing the rule for human interpretation

Table 339: applPolicyRule data structure

8.1.1.9.2.4 ruleOperatorType

The details of the logical operations that can be perfomed on input parameters values to determine if the actions defined for the policy can be executed on associated applications.

Attribute Nme	Data Type	P	Cardinality	Description
opType	String	M	1	Operations that can be applied on Parameter e.g., “Binary Operation” EQ(EQual) or “String Operations” e.g., MATCH(String A Contains String B” etc
ruleOperator	String	M		Operations e.g., EQ(EQual), LT(Less Than), GT(Greater Than), “MATCH” etc

Table 340: ruleOperatorType data structure

8.1.1.9.2.5 ruleActionType

The details of the actions that can be applied to applications as defined for a given application-level policy.

Attribute Name	Data Type	P	Cardinality	Description
actionType	String	M	1	Operations that can be applied for a given policy e.g. application orchestration to preferred Availability Zones, restrict scaling, preferred assignment of specialized resources etc
actionTargetType	String	M		Represent application property that the action mentioned in actionType should be applied to e.g. Availability Zones, Edge Resources etc

Table 341: ruleActionType data structure

8.1.1.9.2.6 applConcretePolicy

Following table describes information about the application policy that the Originating OP registers with the Partner OP based on policy templates including the resource type and their limits for policy rule evaluation and execution.

Attribute Name	Data Type	P	Cardinality	Description
applPolicyId	String	M	1	Represents network capability Identifier

Attribute Name	Data Type	P	Cardinality	Description
policyParamValues	Object	M	1	Represent the information for the policy evaluation based on application attributes like QoS, Compute types and counts etc to determine policy applicability on an application

Table 342: applConcretePolicy

8.1.1.9.2.7 assocApplPolicies

Following table describes information about the association of a policy with one or more applications as determined by the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
applPolicyId	String	M	1	Represents network capability Identifier
applList	Array(Object)	M	1	Application Identifier
appProviderId	String	M	1	Application Provider Identifier
appInstanceId	String	M	1	Application Instance Identifier
monitorDuration	Object	M	1	Time duration in which the trigger condition to be monitored

Table 343: assocApplPolicies

8.1.1.9.2.8 registeredAppPolicyList

The following table describes the list of application-level policies that the Originated OP may have registered using the policy templates shared by the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
appPolicyTemplatesList	Array(applConcretePolicy)	M	1..N	List of application policies registered by the Originating OP

Table 344: registeredAppPolicyList data structure

8.2 Operation-Level Policy Management

This section describes the various HTTP methods for management of the operation-level policies that a Partner OP may share with the Originating OP. The Originating OP may request the Partner OP to attach such policies to subscribed Edge Resources in various Availability Zones with the Partner OP.

8.2.1 Operation-Level Policy Methods

This section describes the HTTP methods and parameters that describes the various HTTP methods for operation-level policies management.

8.2.1.1 Introduction

Following table describes the HTTP resources for management of operation-level policies.

Operations	HTTP Method	Resource URI	Qualifier
Subscribe Operations Policy	POST	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription	C
Retrieve Operation Policy Templates	GET	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/ops-policy-templates	C
Register Operation Policy	POST	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/ops-policy-eregistration	C
Retrieve Operation Policies	GET	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/operate-policies	C
Apply Operation Policies	POST	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/operate-policies	
Modify Operation Ploicies	PATCH	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/operate-policies	
Remove Operation Policies	POST	/operatorplatform/federation/v1/{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/operate-policies	

Table 345: Operation-Level Policies Management HTTP Methods

8.2.1.2 POST Method: Subscribe Operation-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to create subscription for operation-level policies for federated Edge Resources.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.

Table 346: Subscribe Operation-level Policies Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
opsPolicySubscriptionId	M	1	Operation-level policy subscription identifier

Table 347: Subscribe Operation-level Policies response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 348: Subscribe Operation-level Policies response parameters

8.2.1.3 GET Method: Retrieve Operation-level Policy Templates

The HTTP GET request contains the following parameters towards the Partner OP to retrieve operation-level policies templates supported by the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
opsPolicySubscriptionId	M	1	Operation-level policy subscription identifier

Table 349: Retrieve Operation-level Policies Request Parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
opsPolicyTemplateList	M	1	List of operation-level policy templates supported by the Partner OP

Table 350: Retrieve Operation-level Policy Templates Response Parameters

The following table describes the HTTP response codes supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found

Parameter Name	P	Response Codes	Description
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 351: Retrieve Operation-level Policy Templates Error Codes

8.2.1.4 POST Method: Register Operation-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to register operation-level policies for federated Edge Resources.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
opsConcretePolicy	M	1	Operation policy with policy parameter values against policy template supported by Partner OP

Table 352: Register Operation-level Policy Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
opsPolicySubscriptionId	M	1	Operation-level policy subscription identifier
opsConcretePolicy	M	1	Operation policy with policy parameter values against policy template supported by Partner OP

Table 353: Register Operation-level Policy response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict

Parameter Name	P	Response Codes	Description
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 354: Register Operation-level Policies response parameters

8.2.1.5 GET Method: Retrieve Operation-level Policies

The HTTP GET request by the Originating OP contains the following parameters towards the Partner OP to retrieve operation-level policies registered earlier.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.

Table 355: Retrieve Operation-level Policies Request Parameters

The following table describes the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Cardinality	Description
operatePoliciesList	M	1	List of application-level policies supported by the Partner OP

Table 356: Retrieve Operation-level Policies Response Parameters

The following table describes the HTTP response codes supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 357: Retrieve Operation-level Policies Error Codes

8.2.1.6 POST Method: Apply Operation-level Policies

The HTTP POST request contains the following parameters towards the Partner OP to associate operation-level policies for set of Edge Resources in various Availability Zones.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocOperatePolicies	M	1	Association of operation-level policies with the Availability Zones compute infrastructure
txnIdentifier	M	1	Transaction Identifier

Table 358: Apply Operation-level Policies Request

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
assocOperatePolicies	M	1	Association of operation-level policies with the Availability Zones compute infrastructure
txnIdentifier	M	1	Transaction Identifier

Table 359: Apply Operation-level Policies response

The following table describes the HTTP response codes supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 360: Apply Operation-level Policies response parameters

8.2.1.7 POST Method: Remove Operation-level Policies

Following table provides parameters which an Originating OP sends to the Partner OP in HTTP POST request to remove the operation-level policies.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocOperatePolicies	M	1	Operation-level policies to be removed
txnIdentifier	M	1	Transaction Identifier

Table 361: Remove Operation Policies request parameters

The following table describes the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
assocOperatePolicies	C	200	Application policies removed successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Unauthorized access
problemDetails	C	404	Content Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 362: Remove Operation Policies response parameters

8.2.1.8 PATCH Method: Modify Operation-level Policies

Following table provides parameters which an Originating OP sends to a Partner OP in a HTTP PATCH request to modify existing operation-level policies at the Partner OP.

Parameter Name	P	Cardinality	Description
federationContextId	M	1	This identifier shall be provided by the Originating OP to a Partner OP to identify the existing Federation relationship.
assocOperatePolicies	M	1	Operation-level policies to be modified
txnIdentifier	M	1	Transaction Identifier

Table 363: Modify Operation Policies request parameters

The following table describes the data structures supported by the PATCH Response Body on this resource for HTTP 200 response.

Parameter Name	P	Response Codes	Description
assocOperatePolicies	C	200	Application Policies modified successfully
problemDetails	C	400	Bad Request.
problemDetails	C	401	Authorization information is missing or invalid
problemDetails	C	404	Availability Zone Not Found
problemDetails	C	409	Conflict
problemDetails	C	422	Unprocessable Entity. Mandatory parameters are not sent in the request.
problemDetails	C	500	Internal Server Error
problemDetails	C	503	Service Unavailable.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 364: Modify Operation Policies response parameters

8.2.1.9 Data Model

8.2.1.9.1 General

This section provides the data types for the operation-level Policies Management API.

8.2.1.9.2 Structured Data Types

This clause defines the structured data types to be used in resource representations.

8.2.1.9.2.1 opsPolicyTemplate

The events list represents the operations policy template object.

Attribute Name	Data Type	P	Cardinality	Description
opsPolicyName	String	M	1	Identifier of the event object
opsPolicyId	String	M	1	Policy Identifier
opsPolicyType	String	M	1	Indicate what aspect of the Edge Cloud behavior policy aims to influence i.e. static (e.g. Edge Resource pool) or Dynamic (e.g. vertical or horizontal infra scaling)
opsPolicyScope	String	M	1	Represent if policy is specific to the Availability Zone or applies to all Availability Zones i.e., local or global
opsPolicyDescription	String	M	1	Brief text describing the policy intent

Attribute Name	Data Type	P	Cardinality	Description
opsPolicyRules	Array(opsPolicyRule)	M	1..N	Policy specific rules to indicate the conditions and actions that applies to the applications on policy rules execution

Table 365: opsPolicyTemplate data structure

8.2.1.9.2.2 opsIPolicyTemplateList

The events list represents the list of application policy templates object.

Attribute Name	Data Type	P	Cardinality	Description
opsPolicyTemplateList	Array(opsPolicyTemplate)	M	1..N	List of opsPolicyTemplate

Table 366: opsPolicyTemplateList data structure

8.2.1.9.2.3 opsIPolicyRule

Fillowing table describes the elements that comprises an operation policy.

Attribute Name	Data Type	P	Cardinality	Description
ruleLHSParamType	String	M	1	Represent the application property e.g., “EdgeCloud.Metadata.QoS.Latency”, “EdgeCloud.Metadata.Compute.GPU” etc
ruleOperator	ruleOperatorType	M	1	Operations that can be applied on ruleParameter e.g., EQ(EQual), LT(Less Than), GT(Greater Than),
ruleRHSParamVal	Object	M		Permitted type specific value objects for types in ruleLHSParamType
ruleAction	Object	M	1	The pre-defined action that the Partner OP takes on execution of a given rule
ruleDescription	String	M	1	Text describing the rule for human interpretation

Table 367: opsIPolicyRule data structure

8.2.1.9.2.4 opsIConcretePolicy

Following table describes information about the operation policy that the Originating OP registers with the Partner OP based on policy templates.

Attribute Name	Data Type	P	Cardinality	Description
opsPolicyId	String	M	1	Represents operation-level policy identifier
policyParamValues	Object	M	1	Represent the information for the policy evaluation based on Edge Cloud attributes like QoS, Compute types and counts etc to determine policy applicability on an application

Table 368: opsIConcretePolicy

8.2.1.9.2.5 assocOpsPolicies

Following table describes information about the association of a registered operation-level policy with one or more Edge Resources in an Availability Zone as determined by the Originating OP.

Attribute Name	Data Type	P	Cardinality	Description
opsPolicyId	String	M	1	Represents operation-level policy Identifier
compResList	Array(Object)	M	1	Operation-level policy association with Edge Resources in various Availability Zones
monitorDuration	Object	M	1	Time duration in which the trigger condition to be monitored

Table 369: assocOpsPolicies

8.2.1.9.2.6 registeredOpsPolicyList

The following table describes the list of operation-level policies that the Originated OP may have registered using the policy templates shared by the Partner OP.

Attribute Name	Data Type	P	Cardinality	Description
appPolicyTemplatesList	Array(applConcretePolicy)	M	1..N	List of application policies registered by the Originating OP

Table 370: registeredOpsPolicyList data structure

9 Security

Transport Level Security (TLS) shall be used to support the secure communication between the OPs. The access to the E/WBI APIs shall be authorized by means of OAuth2 protocol (see IETF RFC 6749 [4]), based on local configuration, using the "Client Credentials" authorization grant. If OAuth2 is used, a client, prior to consuming services offered by an OP E/WBI APIs, shall obtain a "token" from the authorization server.

9.1 Authorization Service for E/WBI APIs

Authorization Service in the context of the Federation is part of the Operator Platform. An OP shall provide the authorization service endpoints that can be used by the other OPs to request access tokens. Access token shall then be included in the E/WBI APIs by the other federating OPs to request access to the resources offered.

9.2 oAuth 2.0 Roles for E/WBI

The OAuth 2.0 roles, are as follows:

- The Partner OP shall be the OAuth 2.0 Authorization server.
- The Originating OP shall be the OAuth 2.0 client.
- The Partner OP shall be the OAuth 2.0 resource server..

9.3 Usage of signed JWT for oAuth2.0

JSON web token is an open standard (RFC 7519 [7]) that defines a compact and self-contained method for securely transmitting server-generated JSON-encoded information between parties. Access tokens shall be JSON Web Tokens (JWT) and are secured with digital signatures based on JSON Web Signature (JWS) as described in RFC 7515 [6].

Note: As an alternative, JWT can also be secured using Message Authentication Codes (MAC) which requires a pairwise pre-shared symmetric key between the federating OPs. The provisioning of such pre-shared symmetric key is outside the scope of this document.

Note: The JWT claims can be used to provide authorized information associated to an Operator which can be centrally governed. This is FFS

9.4 Authentication and TLS Certificate Validations

E/WBI shall support the standard TLS protocol for the authentication purposes. However, the Operators can agree to use mutual TLS (mTLS) protocol for authenticating each other's identities via TLS certificates.

E/WBI shall minimally support the TLS protocol version 1.2 while version 1.3 is recommended for the current version of this specification. E/WBI should be able to accept the TLS certificates issued by publicly recognized and trusted CAs.

A compliant implementation of GSMA E/WBI shall avoid TLS certificate errors including but not limited to self-signed, invalid (e.g., domain mismatch, broken chain of trust etc.), revoked or expired certificates.

9.5 Notifications and Callbacks Security

The E/WBI shall provide the API notifications by the Partner OP to be delivered securely to the Originating OP. The oAuth 2.0 shall be used with the grant type as "Client Credentials" and the Originating OP can share the oAuth 2.0 credentials to be used by the Partner OPs with the API notification to be delivered to the Originating OP.

10 Usage of HTTP Versions on E/WBI

The E/WBI is a RESTful interface that uses the HTTP protocol for communication between the OPs over a secure transport channel. The E/WBI APIs may utilize different versions of the HTTP protocol.

HTTP/1.1 is the minimum version that an OP implementation must support while HTTP/3 and HTTP/2 can also be used if the federating OPs can mutually agree prior to setting up the Federation establishment process.

The HTTP specifications allows the HTTP/1.1 compliant implementations to dynamically agree and upgrade to HTTP/2 by starting the initial communication with HTTP/1.1. It is recommended that E/WBI procedures shall also allow the standard process of version negotiation mechanism for compliant OP implementation to upgrade to HTTP/2 following the standard HTTP procedures.

11 Federation Template

This section describes the information that the telcos would exchange prior to creating Federation between the OP instances that they are operating.

S.No	Info Provider	Information Shared	Description
1.	Partner OP	Federation endpoint	Partner Federation endpoint i.e., HTTP URL or IP:Port as defined in OPG.04 API Spec
2.	Partner OP	Authorization endpoint	oAuth 2.0 authorization server endpoint i.e., HTTP URL at Partner OP as defined in OPG.04 API Spec
3.	Partner OP	Client ID (oAuth2.0)	oAuth 2.0 client ID generated at Partner OP for “Client Credentials” Grant
4.	Partner OP	Client Secret (oAuth2.0)	oAuth 2.0 client secret generated at Partner OP for “Client Credentials” Grant
5.	Partner OP & Originating OP	HTTP Protocol version	HTTP Protocol version to be used on the E/WBI
6.	Originating OP	Authorization endpoint	oAuth 2.0 endpoint corresponding to the Originating OP for E/WBI Callback Authorization
7.	Originating OP	Client ID (oAuth2.0)	oAuth 2.0 client ID generated at Originating OP for Partner OP Callbacks Authorization for “Client Credentials” Grant
8.	Originating OP	Client Secret (oAuth2.0)	oAuth 2.0 secret generated at Originating OP for Partner OP Callbacks Authorization for “Client Credentials” Grant

S.No	Info Provider	Information Shared	Description
9.	Operator Identity determination field Preference in X.509 Certificate	subjectAltName(recommended), subject, CN	The field in X.509 certificate to determine the identity of the Operator on E/WBI as recommended by this document

Table 371: Federation Creation Information Exchange Template

Annex A OpenAPI Specification Sample

Note: This OpenAPI definition is made available as a YAML file on the GSMA's public website www.gsma.com alongside this PRD.

```
openapi: 3.0.3
info:
  version: 1.4.0
  title: Federation Management Service
  description: |
    # Introduction
    ---
    RESTful APIs that allow an OP to share the Edge Resources and capabilities securely to other Partner OPs over E/WBI.

    ---
    # API Scope

    ---
    APIs defined in this version of the specification can be categorized into the following areas:
    * __FederationAPIManagement__ - Retrieves Federation resources and methods a Partner OP support on E/WBI
    * __FederationManagement__ - Create and manage Directed Federation relationship with a Partner OP
    * __AvailabilityZoneInfoSynchronization__ - Management of resources of Partner OP Availability Zones and status updates
    * __ArtefactManagement__ - Upload, remove, retrieve and update application descriptors, charts and packages over E/WBI towards a Partner OP

    * __FileManagement__ - Upload, remove, retrieve and update application binaries over E/WBI towards a Partner OP
    * __ApplicationOnboardingManagement__ - Register, retrieve, update and remove applications over E/WBI towards a Partner OP
    * __ApplicationDeploymentManagement__ - Create, update, retrieve and terminate application instances over E/WBI towards a Partner OP
    * __AppProviderResourceManagement__ - Static resource reservation for an Application Provider over E/WBI for Partner OP Availability Zones
    * __EdgeNodeSharing__ - Edge discovery procedures towards Partner OP over E/WBI.
    * __ServiceAPIManagement__ - Service APIs capability sharing, forwarding, notification and API context management
    * __SubscribeMonitoringInfo__ - The Originating OP subscribe for receiving the resource utilization reports periodically from the Partner OP for existing Federation
    * __FaultManagement__ - The Partner OP performs the alarm reporting and clearances to the Originating OP on existing Federation
    * __EventsReporting__ - The Partner OP notifies the detection of events as created by the Originating OP on the existing Federation
    * __NetworkEventsReporting__ - The Partner OP notifies the network events applied for offered network capabilities on the existing Federation
    * __ApplicationEventsReporting__ - The Partner OP notifies the applications events of the federated applications
    * __ApplicationPolicyManagement__ - The application-level policy requested by Originating OP for federated applications
    * __OperationPolicyManagement__ - The operation-level policy requested by Originating OP for federated Edge Resources

    ---
    # Definitions
    ---
    This section provides definitions of terminologies commonly referred to throughout the API descriptions.
```

* __Accepted Zones__ - List of Partner OP Availability Zones, which the Originating OP has confirmed to use for its edge applications
* __Anchoring__ - Partner OP capability to serve UEs (still in their home location) from application instances running on partner Availability Zones.
* __Application Provider__ - An Application Provider, onboarding his/her edge application on a Partner OP (MEC).
* __Artefact__ - Descriptor, charts or any other package associated with the application.
* __Availability Zone__ - Availability Zones that Partner OP can offer to share with Originating OP.
* __Device__ - Refers to User Equipment like mobile phone, tablet, IOT kit, AR/VR device etc. In context of MEC End-Users use these devices to access edge applications
* __Directed Federation__ - A Federation between two OP instances A and B, in which Edge Resources are shared by B to A, but not from A to B.
* __Edge Application__ - Application designed to run on MEC Edge Cloud
* __Edge Discovery Service__ - Partner OP service responsible to select most optimal edge(within Partner OP) for edge application instantiation. Edge Discovery Service is defined as HTTP based API endpoint identified by a well-defined FQDN or IP.
* __E/WBI__ - East Westbound interface.
* __Federation__ - Relationship among member OPs who agrees to offer services and capabilities to the Application Providers and Subscribers of member OPs
* __FederationContextId__ - Partner OP defined string identifier representing a certain Federation relationship.
* __Federation Identifier__ - Identify an Operator Platform in Federation context.
* __FileId__ - An OP defined string identifier representing a certain application image uploaded by an Application Provider
* __Flavour__ - A group of compute, network and storage resources that can be requested or granted as a single unit
* __FlavourIdentifier__ - An OP defined string identifier representing a set of compute, storage and networking resources
* __Home OP__ - Used in Federation context to identify the OP with which the Application Providers or user clients are registered.
* __Home Routing__ - Partner OP capability to direct roaming user client traffic towards application instances running on Home OP Availability Zones.
* __Instance__ - Application process running on an edge
* __LCM Service__ - Partner OP service responsible for life cycle management of edge applications. LCM Service is defined as HTTP based API endpoint identified by a well-defined FQDN or IP.
* __Offered Zones__ - Availability Zones that Partner OP offer to share to the Originating OP based on the prior agreement and local configuration.
* __Onboarding__ - Submitting an application to MEC platform
* __OP__ - Operator Platform.
* __OperatorIdentifier__ - String identifier representing the owner of MEC platform. Owner could be an enterprise, a TSP or some other organization
* __Originating OP__ - The OP when initiating the Federation creation request towards the Partner OP is defined as the Originating OP
* __Partner OP__ - Operator Platform which offers its Edge Cloud capabilities to the other Operator Platforms via E/WBI.
* __Resource__ - Compute, networking and storage resources.
* __Resource Pool__ - A group of compute, networking and storage resources. Application Provider pre-reserve resources on Partner OP Availability Zone, these resources are reserved in terms of flavours.
* __ZoneIdentifier__ - An OP defined string identifier representing a certain geographical or logical area where Edge Resources and services are provided
* __Zone Confirmation__ - Procedure via which Originating OP acknowledges Partner OP about the partner Availability Zones it wishes to use.
* __User Clients__ - Lightweight client applications used to access edge applications. Application End-Users run these clients on their devices (UE, IOT device, AR/VR device etc)
* __ServiceAPIManagement__ - Service APIs capability sharing, forwarding, notification and API context management

---

API Operations

__FederationManagement__
* __CreateFederation__ - Creates a Directed Federation relationship with a Partner OP
* __GetFederationDetails__ - Retrieves details about the Federation relationship with the Partner OP. The response shall provide info about the Availability Zones offered by the partner, Partner OP network codes, information about Edge Discovery and LCM Service etc.
* __DeleteFederationDetails__ - Remove existing Federation with the Partner OP
* __NotifyFederationUpdates__ - Call back notification used by Partner OP to update Originating OP about any change in existing Federation relationship
* __UpdateFederation__ - API used by the Originating OP towards the Partner OP, to update the parameters associated to the existing Federation
* __QueryFederationContext__ - The Originating OP retrieves federationContextId from the Partner OP
* __HealthCheckFederation__ - The Originating OP sends health check message to the Partner OP to check the health of the the existing Federation
* __RenewFederation__ - The Originating OP requests the Partner OP to renew the the existing Federation relationship
* __GetNetworkCapabilities__ - The Originating OP requests the Partner OP to share the offered network capabilities information

__AvailabilityZoneInfoSynchronization__
* __ZoneSubscribe__ - Informs Partner OP that Originating OP is willing to access the specified Availability Zones and Partner OP shall reserve Edge and Network Resources for these Availability Zones.
* __ZoneUnsubscribe__ - Informs Partner OP that Originating OP will no longer access the specified partner Availability Zone.
* __GetZoneData__ - Retrieves details about the Edge and Network Resources that Partner OP has reserved for an Partner OP Availability Zone.
* __Notify Zone Information__ - Call back notification used by Partner OP to update Originating OP about changes in the resources reserved on a partner Availability Zone.

__ArtefactManagement__
* __UploadArtefact__ - Uploads application artefact on Partner OP
* __RemoveArtefact__ - Removes an artefact from Partner OP.
* __GetArtefact__ - Retrieves details about an artefact from Partner OP.
* __UploadFile__ Upload application binaries to Partner OP
* __RemoveFile__ - Removes application binaries from Partner OP
* __ViewFile__ - Retrieves details about binaries associated with an application from Partner OP

__ApplicationOnboardingManagement__
* __OnboardApplication__ - Submits an application details to a Partner OP. Based on the details provided, Partner OP shall do bookkeeping, resource validation and other pre-deployment operations
* __UpdateApplication__ - Updates Partner OP about changes in application Edge Resource requirements, QoS Performance Profile, associated descriptor or change in associated components
* __DeboardApplication__ - Removes an application from Partner OP
* __ViewApplication__ - Retrieves application details from Partner OP
* __OnboardExistingAppNewZones__ - Make an application available on new additional Availability Zones
* __LockUnlockApplicationZone__ - Forbid or permit instantiation of application on an Availability Zone

__Application Instance Lifecycle Management__
* __InstallApp__ - Instantiates an application on a Partner OP Availability Zone.
* __GetAppInstanceDetails__ - Retrieves an application instance details from Partner OP.
* __RemoveApp__ - Terminate an application instance on a Partner OP Availability Zone.
* __GetAllAppInstances__ - Retrieves details about all instances of the application running on Partner OP Availability Zones.

__AppProviderResourceManagement__
* __CreateResourcePools__ - Reserves resources (compute, network and storage) on a Partner OP Availability Zone. ISVs registered with Home OP reserves resources on a Partner OP Availability Zone.
* __UpdateISVResPool__ - Updates resources reserved for a pool by an ISV
* __ViewISVResPool__ - Retrieves the resource pool reserved by an ISV
* __RemoveISVResPool__ - Deletes the resource pool reserved by an ISV

__EdgeNodeSharing__
*__GetCandidateZones__ - Edge discovery procedures towards Partner OP over E/WBI. Originating OP request Partner OP to provide a list of candidate Availability Zones where an application instance can be created.

__ServiceAPIManagement__
*__ServiceAPIRequestForwarding__ - Forward the NBI Service API requests to Partner OP over E/WBI.
*__RemoveServiceAPISession__ - Remove the existing Service API session with Partner OP over E/WBI.
*__ServiceAPIRequestForwarding__ - Retrieve Service API session context with Partner OP over E/WBI.

__ConsumptionReportingManagement__
*__SubscribeForResourceConsumption__ - Originating OP Subscription for Edge Resource consumption reporting by Partner OP over E/WBI.

__EventManagement__
*__SubscribeForEventNotifications__ - Originating OP Subscription for edge services related events reporting by Partner OP over E/WBI.

__Alarm Management__
*__SubscribeForAlarmManagement__ - Originating OP Subscription for reporting of alarms by Partner OP over E/WBI.

__Network Capabilities Event Management__
*__SubscribeForNetworkCapabilitiesNotifications__ - Originating OP Subscription for reporting of events for application of network capabilities by Partner OP over E/WBI.

__Applications Event Notifications Management__

*__SubscribeForApplicationEventsNotifications__ - Originating OP Subscription for reporting of application-level events by Partner OP over E/WBI.

© 2024 GSM Association.
All rights reserved.

```
externalDocs:
  description: GSMA, E/WBI APIs v1.4.1
  url: http://www.xxxx.com
servers:
  - url: '{apiRoot}/operatorplatform/federation/v1'
    variables:
      apiRoot:
        default: https://operatorplatform.com
security:
  - oAuth2ClientCredentials:
      - fed-mgmt
  - notifClientCredentials:
      - fed-mgmt-notif
components:
  securitySchemes:
    oAuth2ClientCredentials:
```

```
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: /oauth2/token
          scopes:
            fed-mgmt: Access to the Federation APIs
    notifClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: /oauth2/token
          scopes:
            fed-mgmt-notif: Access to the Federation notification APIs

  schemas:
    AppIdentifier:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
      description: Identifier used to refer to an application.
    AppProviderId:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
      description: UserId of the Application Provider.  Identifier is relevant only in context of this Federation.
    ArtefactId:
      type: string
      format: uuid
      description: A globally unique identifier associated with the artefact. Originating OP generates this identifier
when artefact is submitted over NBI.

    CountryCode:
      type: string
      description: ISO 3166-1 Alpha-2 code for the country of partner Operator
      pattern: ^[A-Z]{2}$
    CPUArchType:
      type: string
      enum:
        - ISA_X86
        - ISA_X86_64
        - ISA_ARM_64
      description: CPU Instruction Set Architecture (ISA) E.g., Intel, Arm etc.

    InstanceIdentifier:
      type: string
      pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
      description: Unique identifier generated by the Partner OP to identify an instance of the application on a
specific Availability Zone.
    InstanceState:
      type: string
      enum:
        - PENDING
        - READY
        - FAILED
        - TERMINATING
      description: Running status of the application instance.

    TransactionId:
      description: A unique transaction id for this request in UUID format. It is used for tracking the request
      example: ab1d6gh5-79c2-3256-7hvb-d897549x40f7
```

```
      format: uuid
      type: string
    Ipv4Addr:
      type: string
      pattern: ^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])$
      example: 198.51.100.1
    Ipv6Addr:
      type: string
      allOf:
        - pattern: ^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))$
        - pattern: ^((([^:]+:){7}([^:]+))|((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?))$
      example: 2001:db8:85a3::8a2e:370:7334
    Fqdn:
      type: string
    FixedNetworkIds:
      type: array
      items:
        type: string
      description: List of network identifier associated with the fixed line network of the Operator Platform.
      minItems: 1
    FederationContextId:
      type: string
      pattern: ^[A-Za-z0-9][A-Za-z0-9-]*$
      readOnly: true
      description: This identifier shall be provided by the Partner OP on successful verification and validation of the Federation create request and is used by Partner OP to identify this newly created Federation context. Originating OP shall provide this identifier in any subsequent request towards the Partner OP.
    FederationIdentifier:
      type: string
      pattern: ^[A-Za-z0-9][A-Za-z0-9-]*$
      description: Globally unique identifier allocated to an Operator Platform. This is valid and used only in context of MEC Federation interface.
    FileId:
      type: string
      format: uuid
      description: A globally unique identifier associated with the image file. Originating OP generates this identifier when file is uploaded over NBI.

    FileName:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
      description: Name of the image file.  Application provides specifies this name when image is uploaded on Originating OP over NBI.

    FileDescription:
      type: string
      minLength: 8
      maxLength: 128
      description: Brief description about the image file.

    FileVersionInfo:
      type: string
      description: File version information.

    FlavourId:
      type: string
      description: An identifier to refer to a specific combination of Edge Resources
    GeoLocation:
```

```
      type: string
      description: Latitude,Longitude as decimal fraction up to 4 digit precision
      pattern: ^([-+]?)([\d]{1,2})((((\.)([\d]{1,4}))?(,)))(([-+]?)([\d]{1,3})((\.)([\d]{1,4}))?)$
    Mcc:
      type: string
      pattern: ^\d{3}$
    Mnc:
      type: string
      pattern: ^\d{2,3}$

    OnboardStatusInfo:
      type: string
      enum:
        - PENDING
        - ONBOARDED
        - DEBOARDING
        - REMOVED
        - FAILED
      description: Defines change in application status. This change could be related to application itself or an
application instance status


    PoolName:
      type: string
      pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
      description: ISV defined name of the resource pool.
    PoolId:
      type: string
      pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
      description: OP defined Identifier for the pool reserved for the ISV. It should be unique with an OP.
    Port:
      type: integer
      minimum: 0
    Status:
      type: string
      enum:
        - FAILED
        - TEMPORARY_FAILURE
        - AVAILABLE
        - LOCKED
        - NOT_AVAILABLE
    Uri:
      type: string
    Vcpu:
      type: string
      pattern: ^\d+((\.\d{1,3})|(m))?$
      description: Number of vcpus in whole, decimal up to millivcpu, or millivcpu format.
      example:
        whole:
          value: 2
        decimal:
          value: 0.500
        millivcpu:
          value: 500m
    VirtImageType:
      type: string
      enum:
        - QCOW2
```

```
      - DOCKER
      - OVA
    description: Indicate if the file is Container image or VM image (QCOW2, OVA)

  ZoneIdentifier:
    type: string
    pattern: ^[A-Za-z0-9][A-Za-z0-9-]*$
    description: Human readable name of the Availability Zone.

  FederationHealthInfo:
    type: object
    required:
      - federationStatus
      - federationStartTime
      - numOfAcceptedZones
    properties:
      federationStatus:
        $ref: '#/components/schemas/State'
      federationStartTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      numOfAcceptedZones:
        type: string
      numOfActiveAlarms:
        type: string
      numOfApplications:
        type: string

  FederationSupportedAPIs:
    type: object
    required:
      - federationBaseAPI
      - availabilityZoneAPI
      - edgeApplicationAPI
      - artefactAPI
      - fileAPI
    properties:
      federationBaseAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      availabilityZoneAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      edgeApplicationAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      artefactAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      fileAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      serviceAPIFederation:
        $ref: '#/components/schemas/FederationAPIResources'
      resourceMonitoringAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      faultManagementAPI:
        $ref: '#/components/schemas/FederationAPIResources'
      eventManagementAPI:
        $ref: '#/components/schemas/FederationAPIResources'


  FederationAPINames:
    type: string
```

```
      enum:
        - FEDERATION
        - AVAILZONE
        - ARTEFACT
        - FILE
        - SVSAPEFED
        - RESMONITOR
        - EVENTMGMT
        - FAULTMGMT

    HttpMethods:
      type: string
      enum:
        - POST
        - PUT
        - PATCH
        - DELETE
        - GET

    HttpResources:
      type: object
      required:
        - href
        - httpMethods
      properties:
        href:
          $ref: '#/components/schemas/Uri'
        httpMethods:
          type: array
          items:
            $ref: '#/components/schemas/HttpMethods'
          minItems: 1
          description: List of HTTP Methods supported for the given API category

    FederationAPIResources:
      type: object
      required:
        - name
        - apiOperations
      properties:
        name:
          $ref: '#/components/schemas/FederationAPINames'
        apiOperations:
          type: array
          items:
            $ref: '#/components/schemas/HttpResources'
          minItems: 1
          description: List of HTTP Methods supported for the given API category

    monitoringSubsType:
      type: string
      enum: ["edge_resource","app_resource","alarm","all"]
      description: Denotes types of Edge Resources, faults and events at Partner OP to be reported to Originating
OP.

    resourceSubscriptionInfo:
      type: object
      required:
```

```
      - monitoringType
      - subscriptionId
      - dateAndTime
    properties:
      monitoringType:
        $ref: '#/components/schemas/monitoringSubsType'
      dateAndTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      subscriptionId:
        type: string
        format: uuid
        description:  Partner OP managed identifier for new subscription.

  utilizationValue:
    type: object
    required:
      - resType
      - value
      - unit
    properties:
      resType:
        $ref: '#/components/schemas/resourceType'
      value:
        type: string
        description:  Whole number that represent the value of given resource type.
      unit:
        type: string
        enum:
          - Percent
          - MBPS
          - GB
          - TB
          - CORES
          - SECONDS
          - MINUTES
    description: Indicate the resource measurement Unit

  resourceType:
    type: string
    enum:
      - CPU
      - MEMORY
      - DISK
      - Network
      - FLAVOUR
    description: Indicate the type of resource

  edgeResUtilizeMetrics:
    type: object
    required:
      - edgeMetrics
      - federationContextId
      - sequenceNum
    properties:
      edgeMetrics:
        type: array
        items:
          $ref: '#/components/schemas/edgeComputeMetrics'
```

```
        minItems: 1
        description: List of Edge Resource metrics per Availability Zone
      federationContextId:
        $ref: '#/components/schemas/FederationContextId'
      sequenceNum:
        type: integer
        description: Monotonically increasing counter for sequencing resource monitoring reports

  edgeComputeMetrics:
    type: object
    required:
      - zoneId
      - startTime
      - endTime
      - cpuUtil
      - memUtil
      - diskUtil
      - networkUtil
      - flavourUtil
    properties:
      zoneId:
        $ref: '#/components/schemas/ZoneIdentifier'
      startTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      endTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      cpuUtil:
        $ref: '#/components/schemas/cpuUtilization'
      memUtil:
        $ref: '#/components/schemas/memUtilization'
      diskUtil:
        $ref: '#/components/schemas/diskUtilization'
      networkUtil:
        $ref: '#/components/schemas/networkUtilization'
      flavourUtil:
        $ref: '#/components/schemas/flavourUtilization'

  memUtilization:
    type: object
    required:
      - noOfSamples
      - averageUtilization
      - maxUtilization
      - minUtilization
    properties:
      noOfSamples:
        type: string
        description:  Number of samples used for calculating metrics.
      averageUtilization:
        $ref: '#/components/schemas/utilizationValue'
      maxUtilization:
        $ref: '#/components/schemas/utilizationValue'
      minUtilization:
        $ref: '#/components/schemas/utilizationValue'
      effectiveUtilization:
        $ref: '#/components/schemas/utilizationValue'

  diskUtilization:
```

```
    type: object
    required:
      - noOfSamples
      - averageUtilization
      - maxUtilization
      - minUtilization
    properties:
      noOfSamples:
        type: string
        description:  Number of samples used for calculating metrics.
      averageUtilization:
        $ref: '#/components/schemas/utilizationValue'
      maxUtilization:
        $ref: '#/components/schemas/utilizationValue'
      minUtilization:
        $ref: '#/components/schemas/utilizationValue'
      effectiveUtilization:
        $ref: '#/components/schemas/utilizationValue'

  networkUtilization:
    type: object
    required:
      - noOfSamples
      - ingressUsage
      - egressUsage
      - averageThroughput
      - maxThroughput
      - minThroughput
    properties:
      noOfSamples:
        type: string
        description:  Number of samples used for calculating metrics.
      ingressUsage:
        $ref: '#/components/schemas/utilizationValue'
      egressUsage:
        $ref: '#/components/schemas/utilizationValue'
      averageThroughput:
        $ref: '#/components/schemas/utilizationValue'
      maxThroughput:
        $ref: '#/components/schemas/utilizationValue'
      minThroughput:
        $ref: '#/components/schemas/utilizationValue'

  flavourUtilization:
    type: array
    items:
      $ref: '#/components/schemas/flavourMetrics'
    minItems: 1
    description: List of compute flavours metrics per Availability Zone

  flavourMetrics:
    type: object
    required:
      - noOfSamples
      - flavourId
      - averageUtilization
      - maxUtilization
      - minUtilization
```

```
    properties:
      noOfSamples:
        type: string
        description:  Number of samples used for calculating metrics.
      flavourId:
        $ref: '#/components/schemas/FlavourId'
      averageUtilization:
        $ref: '#/components/schemas/utilizationValue'
      averageThroughput:
        $ref: '#/components/schemas/utilizationValue'
      maxUtilization:
        $ref: '#/components/schemas/utilizationValue'
      minUtilization:
        $ref: '#/components/schemas/utilizationValue'

  appsResUtilizeInfo:
    type: object
    required:
      - appMetrics
      - federationContextId
      - sequenceNum
    properties:
      appMetrics:
        type: array
        items:
          $ref: '#/components/schemas/appsResUtilizeMetrics'
        minItems: 1
        description: List of Edge Resource metrics per Availability Zone
      federationContextId:
        $ref: '#/components/schemas/FederationContextId'
      sequenceNum:
        type: integer
        description: Monotonically increasing counter for sequencing Application monitoring reports


  appsResUtilizeMetrics:
    type: object
    required:
       - zoneId
       - startTime
       - endTime
       - appZoneMetrics
    properties:
      zoneId:
        $ref: '#/components/schemas/ZoneIdentifier'
      startTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      endTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      appZoneMetrics:
        $ref: '#/components/schemas/appMetrics'

  appMetrics:
    type: array
    items:
      $ref: '#/components/schemas/appAggrResUtil'
    minItems: 1
    description: List of Edge Resource metrics per Availability Zone
```

```
    appAggrResUtil:
      type: object
      required:
        - appId
        - appProvId
        - noOfAppInstances
        - appInstances
        - cpuUtil
        - memUtil
        - diskUtil
        - networkUtil
        - flavourUtil
      properties:
        appId:
          $ref: '#/components/schemas/AppIdentifier'
        appProvId:
          $ref: '#/components/schemas/AppProviderId'
        noOfAppInstances:
          type: integer
          description: No of application instances of appId in an Availability Zone
        appInstances:
          type: array
          items:
            $ref: '#/components/schemas/InstanceIdentifier'
          minItems: 1
        cpuUtil:
          $ref: '#/components/schemas/cpuUtilization'
        memUtil:
          $ref: '#/components/schemas/memUtilization'
        diskUtil:
          $ref: '#/components/schemas/diskUtilization'
        networkUtil:
          $ref: '#/components/schemas/networkUtilization'
        flavourUtil:
          $ref: '#/components/schemas/flavourUtilization'


    cpuUtilization:
      type: object
      required:
        - cpuType
        - noOfSamples
        - averageUtilization
        - maxUtilization
        - minUtilization
        - effectiveUtilization
      properties:
        cpuType:
          $ref: '#/components/schemas/monitoringSubsType'
        noOfSamples:
          type: string
          description:  Number of samples used for calculating metrics.
        averageUtilization:
          $ref: '#/components/schemas/utilizationValue'
        maxUtilization:
          $ref: '#/components/schemas/utilizationValue'
        minUtilization:
```

```
      $ref: '#/components/schemas/utilizationValue'
    effectiveUtilization:
      $ref: '#/components/schemas/utilizationValue'

thresholdVal:
  type: object
  required:
    - value
    - unit
  properties:
    value:
      type: string
    unit:
      type: string
      enum:
        - percent
        - CORES
        - TB
        - GB
        - MBPS
        - GBPS
      description:  The unit of resources measurement e.g. number of cores, mega bits per seconds etc.

EventSubscription:
  type: object
  required:
    - resUsageType
    - periodicity
    - eventListner
  properties:
    resUsageType:
      $ref: '#/components/schemas/resourceType'
    periodicity:
      $ref: '#/components/schemas/periodicityInterval'
    eventListner:
      $ref: '#/components/schemas/Uri'

EventSubscriptionInfo:
  type: object
  required:
    - resUsageType
    - periodicity
    - subscriptionId
  properties:
    resUsageType:
      $ref: '#/components/schemas/resourceType'
    periodicity:
      $ref: '#/components/schemas/periodicityInterval'
    subscriptionId:
      type: string
      format: uuid

eventCriterion:
  type: object
  required:
    - resUsageType
    - triggerCondition
    - thresholdVal
```

```
        - numOccurance
        - monitorDuration
      properties:
        resUsageType:
          $ref: '#/components/schemas/resourceType'
        triggerCondition:
          type: string
          enum:
            - GT
            - GTE
            - EQ
            - LT
            - LEQ
          description:  The condition evaluation operator to compare threshold value of a resource for event
detection.
        thresholdVal:
          $ref: '#/components/schemas/thresholdVal'
        numOccurance:
          type: integer
          description: Number of times the trigger condition is detected
        monitorDuration:
          $ref: '#/components/schemas/periodicityInterval'

    eventInfo:
      type: object
      required:
        - eventId
        - eventCriterion
      properties:
        eventId:
          type: string
        eventCriterion:
          $ref: '#/components/schemas/eventCriterion'

    eventTypeList:
      type: array
      items:
        $ref: '#/components/schemas/eventCriterion'
      minItems: 1
      description: List of event criterion

    detectedEvent:
      type: object
      required:
        - zoneId
        - eventId
        - startTime
        - endTime
        - numOccurance
      properties:
        zoneId:
          $ref: '#/components/schemas/ZoneIdentifier'
        eventId:
          type: string
        startTime:
          $ref: '#/components/schemas/dateAndTimeZoneObject'
        endTime:
          $ref: '#/components/schemas/dateAndTimeZoneObject'
```

```
        numOccurance:
          type: integer

    CapabilityID:
      type: string
      enum:
        - NW_CAP_CONN_STATE_CHANGE
        - NW_CAP_LOCATION_RETRIEVAL
        - NW_CAP_USERPLANE_MGMT_EVENTS
        - NW_CAP_DYNAMIC_QOS
      description: The enumerated list of network capabilities that an OP can use for various services via SBI-NR.

    DeviceConnStatusChangeCap:
      type: object
      required:
        - capabilityId
        - maxiDetectionTime
      properties:
        capabilityId:
          $ref: '#/components/schemas/CapabilityID'
        maxiDetectionTime:
          type: string
          description:  The maximum detection time in seconds that the OP can determine the UE change of
connectivity with the mobile network.

    LocationRetrievalCap:
      type: object
      required:
        - capabilityId
        - locationType
      properties:
        capabilityId:
          $ref: '#/components/schemas/CapabilityID'
        locationType:
          type: string
          enum:
            - CELL_LEVEL_ACCURACY
            - REGISTRATION_AREA_ACCURACY
            - TRACKING_AREA_ACCURACY
            - GEO_LOCATION_ACCURACY
          description:  The enumerated list of UE location accuracy that an OP can determine via SBI-NR.
        locationAccuracy:
          type: string
          enum:
            - LAST_KNOWN_LOCATION
            - CURRENT_LOCATION
            - INITIAL_LOCATION
          description:  The enumerated list of type of network location of an UE that an OP can determine via SBI-
NR.

    UserPlaneMgmtEvtCap:
      type: object
      required:
        - capabilityId
        - maxUserPlaneLatency
      properties:
        capabilityId:
          $ref: '#/components/schemas/CapabilityID'
```

```
        maxUserPlaneLatency:
          type: string
          description: Indicates the maximum user plane latency in units of milliseconds to decide whether edge relocation is needed to ascertain latency remain in this range.

    DynamicQoSCap:
      type: object
      required:
        - capabilityId
        - supportedQoS
      properties:
        capabilityId:
          $ref: '#/components/schemas/CapabilityID'
        supportedQoS:
          type: string
          description:  Set of one or more 5G QoS Identifier (5QI or 4G QCI) created via concatanation of Resource Type and 5QI values i.e., GBR1, GBR2, GBR65, NONGBR79 etc.


    NetworkCapAppInfoList:
      type: array
      items:
        required:
          - appProviderId
          - appId
          - AppInstNetworkCapInvoked
          - zoneId
        properties:
          appProviderId:
            $ref: '#/components/schemas/AppProviderId'
          appId:
            $ref: '#/components/schemas/AppIdentifier'
          zoneId:
            $ref: '#/components/schemas/ZoneIdentifier'
          appInstNetworkCapInvoked:
            $ref: '#/components/schemas/AppInstNetworkCapList'
      minItems: 1

    AppInstNetworkCapList:
      type: object
      required:
        - appInstanceNwCapInfo
      properties:
        appInstanceNwCapInfo:
          type: array
          items:
            type: object
            required:
              - appInstIdentifier
              - appInstanceState
              - networkCapInvoked
            properties:
              appInstIdentifier:
                $ref: '#/components/schemas/InstanceIdentifier'
              appInstanceState:
                $ref: '#/components/schemas/InstanceState'
              networkCapInvoked:
                $ref: '#/components/schemas/NetworkCapInvoked'
```

```
        minItems: 1

  NetworkCapInvoked:
    type: object
    required:
      - networkEventId
      - capabilityId
      - zoneId
      - detectionTime
      - nwCapabilitySLI
    properties:
      networkEventId:
        type: string
        format: uuid
        description: Unique identifier allocated for a network event
      capabilityId:
        $ref: '#/components/schemas/CapabilityID'
      invocationTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      nwCapabilitySLI:
        type: string
      zoneId:
        $ref: '#/components/schemas/ZoneIdentifier'

  NetworkCapSubsInfo:
    type: object
    required:
      - appId
      - appProviderId
      - capabilityId
    properties:
      appId:
        $ref: '#/components/schemas/AppIdentifier'
      appProviderId:
        $ref: '#/components/schemas/AppProviderId'
      capabilityId:
        $ref: '#/components/schemas/CapabilityID'


  NetworkEventsList:
    type: array
    items:
      $ref: '#/components/schemas/NetworkCapInvoked'
    minItems: 1
    description: List of network capabilities events detected

  EventsList:
    type: array
    items:
      $ref: '#/components/schemas/detectedEvent'
    minItems: 1
    description: List of events detected

  EventSubscriptionIdentifier:
    type: string
    format: uuid
    description: Event subscription identifier allocated for enabling event reporting
```

```
    EventIdentifier:
      type: string
      format: uuid
      description: Event identifier allocated for event detected

    SubscriptionIdentifier:
      type: object
      required:
        - subsId
      properties:
        subsId:
          type: string
          format: uuid
          description: Generic subscription identifier

    AlarmObjectInfo:
      type: object
      required:
        - alarmType
        - alarmId
        - perceivedSeverity
        - probableCause
        - alarmedObject
        - sourceSystemId
        - state
        - alarmRaisedTime
      properties:
        alarmType:
          $ref: '#/components/schemas/AlarmType'
        alarmId:
          $ref: '#/components/schemas/AlarmIdentifier'
        perceivedSeverity:
          $ref: '#/components/schemas/PerceivedSeverity'
        probableCause:
          $ref: '#/components/schemas/ProbableCause'
        alarmedObject:
          $ref: '#/components/schemas/AlarmedObject'
        sourceSystemId:
          $ref: '#/components/schemas/SourceSystemId'
        state:
          $ref: '#/components/schemas/State'
        alarmRaisedTime:
          $ref: '#/components/schemas/AlarmRaisedTime'
        affectedService:
          $ref: '#/components/schemas/AffectedService'
        alarmDetails:
          $ref: '#/components/schemas/AlarmDetails'
        specificProblem:
          $ref: '#/components/schemas/SpecificProblem'
        serviceAffecting:
          $ref: '#/components/schemas/ServiceAffecting'

    ActiveAlarmsList:
      type: array
      items:
          $ref: '#/components/schemas/AlarmObjectInfo'
      minItems: 1
      description: List of active alarms
```

```
    AlarmType:
      type: object
      required:
        - alarmType
      properties:
        alarmType:
          type: string
          enum:
            - EDGERES
            - APPLICATION
            - ARTEFACT
            - EDGEDISC
            - FEDERATION
            - SECURITY
            - APIFEDERATION
            - FILE
          description: Alarm type category

    AlarmIdentifier:
      type: object
      required:
        - alarmId
      properties:
        alarmId:
          type: string
          description: Alarm identifier to refer to an alarm instance

    PerceivedSeverity:
      type: object
      required:
        - severity
      properties:
        severity:
          type: string
          enum:
            - MAJOR
            - MINOR
            - CRITICAL
            - WARNING
            - INFOMATIONAL
          description: Alarm severity

    ProbableCause:
      type: object
      required:
        - cause
      properties:
        cause:
          type: string
          description: Probale cause of the alarm

    AlarmedObject:
      type: object
      required:
        - alarmId
        - href
      properties:
```

```
        alarmId:
          $ref: '#/components/schemas/AlarmIdentifier'
        href:
          $ref: '#/components/schemas/Uri'

    SourceSystemId:
      type: object
      required:
        - sourceSystemId
      properties:
        sourceSystemId:
          type: string
          description: Source system identity

    State:
      type: object
      required:
        - alarmState
      properties:
        alarmState:
          type: string
          enum:
            - RAISED
            - UPDATED
            - CLEAR
          description: Defines the alarm state during its life cycle (raised | updated | cleared).

    AlarmRaisedTime:
      type: object
      required:
        - alarmRaisedTime
      properties:
        alarmRaisedTime:
          type: string
          format: date-time
          description: Defines the alarm raised time at source

    AffectedService:
      type: object
      required:
        - affectedService
      properties:
        affectedService:
          type: array
          items:
            type: string
          minItems: 1
          description: Defines the affected services e.g., Edge Discovery, application services, API services etc at
source

    AlarmDetails:
      type: object
      required:
        - alarmDetails
      properties:
        alarmDetails:
          type: string
          description: Detailed information of the alarm
```

```
    SpecificProblem:
      type: object
      required:
        - specificProblem
      properties:
        specificProblem:
          type: string
          description: Specific information related to the alarm

    ServiceAffecting:
      type: string
      enum:
        - YES
        - NO
      description: Specific information related to the alarm

    PatchableParams:
      type: string
      enum:
["/perceivedSeverity","/probableCause","/alarmedObject","/sourceSystemId","/state","/affectedService","/alarmDet
ails","/specificProblem","/serviceAffecting"]

    AlarmUpdateOps:
      type: string
      enum:
        - REPLACE
      description: Operations that can be performed to update the parameters of an alarm

    UpdatedParam:
      type: object
      required:
          - alarmUpdateOps
          - patchableParam
          - patchValue
      properties:
        alarmUpdateOps:
          $ref: '#/components/schemas/AlarmUpdateOps'
        patchableParam:
          $ref: '#/components/schemas/PatchableParams'
        patchValue:
          type: string
          description: Value to be replaced for the alarm parameter being updated

    UpdatedAlarmParameters:
      type: object
      required:
        - alarmId
        - updateParams
      properties:
        alarmId:
          $ref: '#/components/schemas/AlarmIdentifier'
        updateParams:
          type: array
          items:
            $ref: '#/components/schemas/UpdatedParam'
          minItems: 1
          description: List of alarm parameters to be updated in an update operation
```

```
    serviceType:
      type: string
      enum: ["api_federation"]
      description: An identifier to refer to Partner OP capabilities for Application Providers.

    serviceAPINames:
      type: array
      items:
          type: string
          enum:
            - QualityOnDemand
            - DeviceLocation
            - DeviceStatus
            - SimSwap
            - NumberVerification
            - DeviceIdentifier
      minItems: 1
      description: List of Service API capability names an OP supports and offers to other OPs
"quality_on_demand", "device_location" etc.

    serviceAPINameVal:
      type: string
      enum:
        - QualityOnDemand
        - DeviceLocation
        - DeviceStatus
        - SimSwap
        - NumberVerification
        - DeviceIdentifier
      description: Name of the Service API

    serviceRoutingInfo:
      type: array
      items:
          type: string
          pattern: ^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\/([0-
9]|[1-2][0-9]|3[0-2]))?$
      minItems: 1
      description: List of public IP addresses Operator manages for UEs to connect with public data networks


    customerID:
      type: string
      format: uuid
      description:  Leading OP managed identifier associated to API Provider of the Leading OP.

    txnIdentifier:
      type: string
      description: A API transaction identifier generated by the Partner OP for each API request

    connectID:
      type: string
      description: An identifier generated by the Partner OP to represent the End-User identity in the Service API
request.

    apiContentType:
      type: string
```

```
      enum:
        - application/json
      description: Indicate the Service API body schema in JSON format

    serviceAPIContent:
      type: object
      required:
          - mediaType
          - APIContent
      properties:
        mediaType:
          $ref: '#/components/schemas/apiContentType'
        APIContent:
          $ref: 'https://github.com/camaraproject'

    PlatformCaps:
      type: array
      items:
        type: string
        enum:
          - homeRouting
          - Anchoring
          - serviceAPIs
          - faultMgmt
          - eventMgmt
          - resourceMonitor
          - networkEventMgmt
          - appNotificationMgmt
          - appLevelPolicyMgmt
          - opsLevelPolicyMgmt
      description: Home routing - Operator Platform is capable of routing edge application data traffic from its
edges to UEs in their home location. This is the case where UEs are served in their home Region (requesting OP
Region, non-roaming) but the corresponding edge application are in Operator Platform edges. Anchoring -
Operator Platform is capable of routing edge application traffic for roaming UEs to edge application in the UE's
home network. Service APIs - Capability to handle Service APIs (e.g., CAMARA APIs) from the Leading OP

    expiryInterval:
      type: object
      required:
          - numHours
          - numMins
          - numSecs
      properties:
        numHours:
          type: integer
          format: int32
          description: Number of Hours for Expiry (0-23)
        numMins:
          type: integer
          format: int32
          description: Number of Minutes for Expiry (0-59)
        numSecs:
          type: integer
          format: int32
          description: Number of Seconds for Expiry (0-59)

    periodicityInterval:
      type: object
```

```
    required:
      - numHours
      - numMins
    properties:
      numHours:
        type: integer
        format: int32
        description: Number of Hours for Expiry (0-23)
      numMins:
        type: integer
        format: int32
        description: Number of Minutes for Expiry (0-59)

  periodicNotifConfig:
    type: object
    properties:
      periodicity:
        $ref: '#/components/schemas/periodicityInterval'
      notificationListner:
        $ref: '#/components/schemas/Uri'

  targetUserContext:
    type: object
    required:
      - connectID
      - expiryDuration
    properties:
      connectID:
        $ref: '#/components/schemas/connectID'
      expiryDuration:
        $ref: '#/components/schemas/expiryInterval'

  serviceAPIResponse:
    type: object
    required:
      - customerID
      - targetUserContext
      - apiResponse
      - txnIdentifier
    properties:
      customerID:
        $ref: '#/components/schemas/customerID'
      targetUserContext:
        $ref: '#/components/schemas/targetUserContext'
      apiResponse:
        $ref: '#/components/schemas/customerID'
      txnIdentifier:
        $ref: '#/components/schemas/txnIdentifier'
  svcEventType:
    type: string
    enum:
      - "evt_timerexpiry"
      - "evt_network"
      - "evt_delete"


  serviceAPIEventDef:
    type: object
```

```
    required:
      - NetworkEventDef
    properties:
      NetworkEventDef:
        $ref: 'https://github.com/camaraproject'

  serviceAPINetworkEvent:
    type: object
    required:
       - connectID
       - customerID
       - EventType
    properties:
      connectID:
        $ref: '#/components/schemas/connectID'
      customerID:
        $ref: '#/components/schemas/customerID'
      EventType:
        $ref: '#/components/schemas/svcEventType'
      serviceAPIEventDef:
        $ref: '#/components/schemas/serviceAPIEventDef'
      expiryDuration:
        $ref: '#/components/schemas/expiryInterval'

  #
  # STRUCTURED DATA TYPES
  #
  ServiceNameNB:
    type: string
    pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
    description: Must be a valid RFC 1035 label name. This defines the DNS name via which the component can be accessed over NBI. Access via     serviceNameNB is restricted on specific ports. Operator Platform shall expose component access externally via this DNS name
  ServiceNameEW:
    type: string
    pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
    description: Must be a valid RFC 1035 label name. This defines the DNS name via which the component can be accessed via peer components. Access via serviceNameEW is open on all ports. Operator Platform shall not expose serviceNameEW externally outside edge.
  ComponentName:
    type: string
    pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
    description: Must be a valid RFC 1035 label name.  Component name must be unique with an application

  ApplEventsSubsInfo:
    type: object
    required:
       - appEventSubsId
       - appEvtSubsStartTime
       - appEvtSubsLastReportTime
       - appEvtSubsNumApps
       - appEvtSubsPeriodicity
    properties:
      appEventSubsId:
        $ref: '#/components/schemas/EventSubscriptionIdentifier'
      appEvtSubsStartTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      appEvtSubsLastReportTime:
```

```
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      appEvtSubsNumApps:
        type: integer
      appEvtSubsPeriodicity:
        $ref: '#/components/schemas/periodicityInterval'

  AppsForNotif:
    type: object
    required:
      - appId
      - appProviderId
      - appZones
      - appEvents
    properties:
      appId:
        $ref: '#/components/schemas/AppIdentifier'
      appProviderId:
        $ref: '#/components/schemas/AppProviderId'
      appZones:
        $ref: '#/components/schemas/AppZones'
      appEvents:
        $ref: '#/components/schemas/AppEvents'

  AddAppsForNotif:
    type: array
    items:
      $ref: '#/components/schemas/AppsForNotif'
    minItems: 1

  RemoveAppsForNotif:
    type: array
    items:
      $ref: '#/components/schemas/AppsForNotif'
    minItems: 1


  AppEventTypes:
    type: string
    enum:
      - evt_type_app_relocation
      - evt_type_app_session_cont
      - evt_type_app_restarts
      - evt_type_app_upscale
      - evt_type_app_downscale
    description: Application-level events

  AppEvents:
    type: array
    items:
      $ref: '#/components/schemas/AppEventTypes'
    minItems: 1
    description: List of Availability Zones where application events are to be monitored

  AppZones:
    type: array
    items:
      $ref: '#/components/schemas/ZoneIdentifier'
    minItems: 1
```

```
    description: List of Availability Zones where application events are to be monitored

  ApplInstEventTypeInfo:
    type: object
    required:
      - applInstEvent
      - applInstEventCount
    properties:
      applInstEvent:
        $ref: '#/components/schemas/AppEventTypes'
      applInstEventCount:
        type: integer
        description: Number of occurances of given epplication event

  ApplInstEventsContainer:
    type: object
    required:
      - appInstanceId
      - appInstEventsList
    properties:
      appInstanceId:
        $ref: '#/components/schemas/InstanceIdentifier'
      appInstEventsList:
        type: array
        items:
          $ref: '#/components/schemas/ApplInstEventTypeInfo'
        minItems: 1
        description:  Application instance events list

  ApplInstEventsList:
    type: object
    required:
      - appInstanceEventsList
    properties:
      appInstanceEventsList:
        type: array
        items:
          $ref: '#/components/schemas/ApplInstEventsContainer'
        minItems: 1
        description:  Application instance events list for one or more applications

  ZoneLevelApplEventsList:
    type: object
    required:
      - zoneId
      - appsEventsList
    properties:
      zoneId:
        $ref: '#/components/schemas/ZoneIdentifier'
      appsEventsList:
        type: array
        items:
          $ref: '#/components/schemas/ApplInstEventsList'
        minItems: 1
        description:  Applications instance events list in an Availability Zone

  ApplEventsList:
    type: object
```

```
    required:
      - appId
      - appProviderId
      - aggrApplEvents
    properties:
      appId:
        $ref: '#/components/schemas/AppIdentifier'
      appProviderId:
        $ref: '#/components/schemas/AppProviderId'
      aggrApplEvents:
        type: array
        items:
          $ref: '#/components/schemas/ZoneLevelApplEventsList'
        minItems: 1
        description:  Applications instance events list in an Availability Zone

  AggrApplEventsList:
    type: object
    required:
      - startTime
      - endTime
      - aggrAppsEventsList
    properties:
      startTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      endTime:
        $ref: '#/components/schemas/dateAndTimeZoneObject'
      aggrAppsEventsList:
        type: array
        items:
          $ref: '#/components/schemas/ApplEventsList'
        minItems: 1
        description:  Applications events list in a various Availability Zones for different Application Providers

  ApplPolicyIdentifier:
    type: string
    pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
    description: Application-level Policy unique identifier

  ApplPolicyMetaInfo:
    type: object
    required:
      - applPolicyTypeIdentifier
      - policyVersion
    properties:
      applPolicyTypeIdentifier:
        $ref: '#/components/schemas/ApplPolicyTypeIdentifier'
      policyVersion:
        type: string
        description: Policy template version using Semantic Versioning 2.0.0 in MAJOR.MINOR.PATCH format


  ApplPolicyTypeIdentifier:
    type: string
    pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
    description: Application-level Policy unique identifier
```

```
    AppPolicyTemplate:
      type: object
      required:
        - applPolicyName
        - applPolicyMetaInfo
        - applPolicyType
        - applPolicyScope
        - applPolicyDescription
        - applPolicyRules
      properties:
        applPolicyName:
          type: string
          maxLength: 64
          description: Brief policy template name on policy objective
        applPolicyMetaInfo:
          $ref: '#/components/schemas/ApplPolicyMetaInfo'
        applPolicyType:
          $ref: '#/components/schemas/ApplPolicyType'
        applPolicyScope:
          $ref: '#/components/schemas/ApplPolicyScope'
        applPolicyDescription:
          type: string
          maxLength: 256
          description: Brief policy template description on policy objective
        applPolicyRules:
          type: array
          items:
            $ref: '#/components/schemas/ApplPolicyRule'
          minItems: 1
          description: Set of policy action rules for a given policy

    ApplPolicyTemplateList:
      type: array
      items:
        $ref: '#/components/schemas/AppPolicyTemplate'
      minItems: 1
      description:  List of Application policy templates from the Partner OP

    ApplPolicyType:
      type: string
      enum:
        - static
        - dynamic
      description: Policy attribute that the given policy intent to control specific resources e.g. compute capacity
expansion statically vs dynamic scaling of application instances

    ApplPolicyScope:
      type: string
      enum:
        - zonal
        - global
      description: Application-level Policy scope defines if a policy is a set of Availability Zones or applies globally
to all Availability Zones


    ApplPolicyRule:
      type: array
      items:
```

```
      $ref: '#/components/schemas/GenericPolicyRule'
    minItems: 1
    description:  List of Application policies


  GenericPolicyRule:
    type: object
    required:
      - ruleLHSParamType
      - ruleOperator
      - ruleRHSParamVal
      - ruleAction
      - ruleDescription
    properties:
      ruleLHSParamType:
        $ref: '#/components/schemas/RuleLHSParamType'
      ruleOperator:
        $ref: '#/components/schemas/RuleOperatorType'
      ruleRHSParamVal:
        $ref: '#/components/schemas/RuleRHSParamVal'
      ruleAction:
        $ref: '#/components/schemas/RuleActionType'
      ruleDescription:
        type: string
        maxLength: 256
        description: Brief description of the actions to be performed

  RuleLHSParamType:
    type: string
    enum:
      - AppsPolicy.App.Metadata.QoS.Latency
      - AppsPolicy.App.Metadata.Compute.CPU
      - AppsPolicy.App.Metadata.Compute.GPU
      - AppsPolicy.App.Metadata.Location.AZ
      - AppsPolicy.App.Metadata.Location.Region
      - OpsPolicy.EdgeCloud.Metadata.QoS.Latency
      - OpsPolicy.EdgeCloud.Metadata.Compute.CPU
      - OpsPolicy.EdgeCloud.Metadata.Compute.GPU
      - OpsPolicy.EdgeCloud.Metadata.Network.SRIOV
    description: Resource attributes that policy will act on to determine the target pplication after applying the
policy rules

  RuleRHSParamVal:
    type: object
    properties:
      latencyRanges:
        $ref: '#/components/schemas/LatencyRanges'
      computeResourceProfile:
        $ref: '#/components/schemas/ComputeResourceProfile'
      appLocation:
        type: array
        items:
          $ref: '#/components/schemas/AppLocation'
        minItems: 1
      networkCaps:
        $ref: '#/components/schemas/NetworkCaps'
    description: Permitted type specific value objects for types in ruleLHSParamType
```

```
    AppLocation:
      type: string
      enum:
        - zones
        - regions
      description: Application Location in terms of Availability Zones or Regions

    LatencyRanges:
      type: object
      required:
        - minLatency
        - maxLatency
        - unit
      properties:
        minLatency:
          type: string
          description: Minimum latency in milliseconds
        maxLatency:
          type: string
          description: Maximum latency in milliseconds
        unit:
          type: string
          enum:
            - MS
          description: Maximum latency in milliseconds
      description: Latency ranges that can be experienced in the Partner OP environment

    ComputeResourceProfile:
      type: object
      required:
        - resourceSpec
      properties:
        resourceSpec:
          $ref: '#/components/schemas/ResourceSpec'
      description: Type and amount of Edge Resources

    ResourceSpec:
      type: object
      required:
        - resourceType
        - resourceModel
        - resourceCount
      properties:
        resourceType:
          type: string
          enum:
            - CPU
            - GPU
            - FPGA
        resourceModel:
          type: string
          enum:
            - Intel-x86_64
            - Arm64
            - Nvidia
        resourceCount:
          type: string
```

```
    description: Resource type and  architecture specification

  NetworkCaps:
    type: object
    properties:
      nwAccelType:
        type: string
        enum:
          - SRIOV
          - DPDK
      nwAccelSpeed:
        type: string
        enum:
          - 1Gbps
          - 10Gbps
          - 100Gbps
    description: Type and speed of network acceleration resources


  RuleOperatorType:
    type: object
    properties:
      StringRuleOperatorType:
        $ref: '#/components/schemas/StringRuleOperatorType'
      BinaryRuleOperatorType:
        $ref: '#/components/schemas/BinaryRuleOperatorType'
    description: Defines the logical operations that policy rule will execute on application attribute value

  BinaryRuleOperatorType:
    type: string
    enum:
      - EQ
      - LT
      - GT
    description: Operations that can be applied on Parameter e.g., “Binary Operation” EQ(EQual)

  StringRuleOperatorType:
    type: string
    enum:
      - EQ
      - NOTEQ
    description: Operations that can be applied on Parameter e.g., String Operation” EQ(EQual), NOTEQ(Not
Equal)

  RuleActionType:
    type: object
    required:
      - actionType
      - actionTargetType
    properties:
      actionType:
        $ref: '#/components/schemas/ActionType'
      actionTargetType:
        $ref: '#/components/schemas/RuleLHSParamType'

  ActionType:
    type: string
    enum:
```

```
      - restrict
      - prefer
      - priortize
      - allow
      - deny
    description: Action to be taken once a policy rule is applied on target resource indicated by
RuleLHSParamType

  ApplConcretePolicy:
    type: object
    required:
      - policyId
      - policyParamLimits
    properties:
      policyId:
        $ref: '#/components/schemas/ApplPolicyIdentifier'
      policyParamLimits:
        $ref: '#/components/schemas/ApplPolicyRule'
    description: Application policy id and policy parameter value limits registered by the Originating OP

  AssocApplPolicies:
    type: object
    required:
      - policyId
      - appIdList
    properties:
      policyId:
        $ref: '#/components/schemas/ApplPolicyIdentifier'
      appIdList:
        $ref: '#/components/schemas/AppIdLocList'

  AppIdLocList:
    type: object
    required:
      - appId
      - appProvId
    properties:
      appId:
        $ref: '#/components/schemas/AppIdentifier'
      appProvId:
        $ref: '#/components/schemas/AppProviderId'
      zoneIds:
        type: array
        items:
          $ref: '#/components/schemas/ZoneIdentifier'
        minItems: 1

  RegisteredAppPolicyList :
    type: array
    items:
      $ref: '#/components/schemas/ApplConcretePolicy'
    minItems: 1
    description:  Applications policies registered by the Originating OP


  OpsPolicyIdentifier:
    type: string
    pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
```

```
    description: Operation-level Policy unique identifier


  OpsPolicyTypeIdentifier:
    type: string
    pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
    description: Operation-level Policy template unique identifier

  OpslPolicyMetaInfo:
    type: object
    required:
      - opslPolicyTypeIdentifier
      - policyVersion
    properties:
      opslPolicyTypeIdentifier:
        $ref: '#/components/schemas/OpsPolicyTypeIdentifier'
      policyVersion:
        type: string
        description: Policy template version using Semantic Versioning 2.0.0 in MAJOR.MINOR.PATCH format


  OpsPolicyTemplateList:
    type: array
    items:
      $ref: '#/components/schemas/OpsPolicyTemplate'
    minItems: 1
    description:  List of Operation policy templates from the Partner OP
  OpsConcretePolicy:
    type: object
    required:
      - policyId
      - policyParamLimits
    properties:
      policyId:
        $ref: '#/components/schemas/OpsPolicyIdentifier'
      policyParamLimits:
        $ref: '#/components/schemas/OpsPolicyRule'
    description: Application policy id and policy parameter value limits registered by the Originating OP


  OpsPolicyTemplate:
    type: object
    required:
      - opsPolicyName
      - OpslPolicyMetaInfo
      - opsPolicyType
      - opsPolicyScope
      - opsPolicyDescription
      - opsPolicyRules
    properties:
      opsPolicyName:
        type: string
        maxLength: 64
        description: Brief policy template name on policy objective
      opslPolicyMetaInfo:
        $ref: '#/components/schemas/OpslPolicyMetaInfo'
      opsPolicyType:
        $ref: '#/components/schemas/OpsPolicyType'
```

```
      opsPolicyScope:
        $ref: '#/components/schemas/OpsPolicyScope'
      opsPolicyDescription:
        type: string
        maxLength: 256
        description: Brief policy template description on policy objective
      opsPolicyRules:
        type: array
        items:
          $ref: '#/components/schemas/OpsPolicyRule'
        minItems: 1
        description: Set of policy action rules for a given policy


  OpsPolicyRule:
    type: object
    properties:
      opsPolicyRule:
        $ref: '#/components/schemas/GenericPolicyRule'
    description:  Operation policies rule defines the action to be taken against the subscribed policy template


  OpsPolicyType:
    type: string
    enum:
      - static
      - dynamic
    description: Policy attribute that defines if the policy rules applies to static part of the infra or dynamic part of
the Edge Cloud infra

  OpsPolicyScope:
    type: string
    enum:
      - zonal
      - global
    description: Operation-level Policy scope defines if a policy is a set of Availability Zones or applies globally to
all Availability Zones


  AssocOpsPolicies:
    type: object
    required:
      - policyId
      - appIdList
    properties:
      policyId:
        $ref: '#/components/schemas/OpsPolicyIdentifier'
      appIdList:
        $ref: '#/components/schemas/AppIdLocList'

  AvailZoneIdLocList:
    type: object
    required:
      - zoneIds
    properties:
      zoneIds:
        type: array
        items:
```

```
          $ref: '#/components/schemas/ZoneIdentifier'
        minItems: 1

    RegisteredOpsPolicyList :
      type: array
      items:
        $ref: '#/components/schemas/OpsConcretePolicy'
      minItems: 1
      description:  Operation policies registered by the Originating OP


    AppComponentSpecs:
      description: An application may consist of more than one component. Each component is associated with a
descriptor and may exposes its services externally or internally.  Application Providers are required to provide
details about all these components, their associated descriptors and their DNS names.
      type: array
      items:
        type: object
        required:
          - artefactId
        properties:
          serviceNameNB:
            $ref: '#/components/schemas/ServiceNameNB'
          serviceNameEW:
            $ref: '#/components/schemas/ServiceNameEW'
          componentName:
            $ref: '#/components/schemas/ComponentName'
          artefactId:
            $ref: '#/components/schemas/ArtefactId'
      minItems: 1

    AppMetaData:
      description: Application metadata details
      type: object
      required:
        - appName
        - version
        - accessToken
      properties:
        appName:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
          description: Name of the application. Application Provider define a human readable name for the
application
        version:
          type: string
          description: Version info of the application
        appDescription:
          type: string
          minLength: 16
          maxLength: 256
          description: Brief application description provided by Application Provider
        mobilitySupport:
          $ref: '#/components/schemas/MobilitySupport'
        accessToken:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{31,63}$
```

```
        description: An application Access key, to be used with UNI interface to authorize UCs Access to a given
application
      category:
        type: string
        enum:
          - IOT
          - HEALTH_CARE
          - GAMING
          - VIRTUAL_REALITY
          - SOCIALIZING
          - SURVEILLANCE
          - ENTERTAINMENT
          - CONNECTIVITY
          - PRODUCTIVITY
          - SECURITY
          - INDUSTRIAL
          - EDUCATION
          - OTHERS
        description: Possible categorization of the application
    AppQoSProfile:
      description: Parameters corresponding to the performance constraints, tenancy details etc.
      type: object
      required:
        - latencyConstraints
      properties:
        latencyConstraints:
          $ref: '#/components/schemas/LatencyConstraints'
        bandwidthRequired:
          $ref: '#/components/schemas/BandwidthRequired'
        multiUserClients:
          $ref: '#/components/schemas/MultiUserClients'
        noOfUsersPerAppInst:
          $ref: '#/components/schemas/NoOfUsersPerAppInst'
        appProvisioning:
          $ref: '#/components/schemas/AppProvisioning'

    EdgeAppFQDN:
      type: string
      description: DNS FQDN assigned to application instances in an Availability Zone. User Clients can resolve
the FQDN to communicate with the edge instances of the application
    ClientLocation:
      type: object
      minProperties: 1
      properties:
        geo_location:
          type: string
          description: Latitude, Longitude as decimal fraction up to 4 digit precision
          pattern: ^([-+]?)([\d]{1,2})(((\.)(\d+)(,)))(\s*)(([-+]?)([\d]{1,3})((\.)(\d+))?)$
        rad_location:
          description: Information about the 4G/5G Cell ids where the client is currently served.
          type: array
          items:
            type: object
            required:
              - carrier
              - mcc
              - mnc
              - cellId
```

```
        properties:
          carrier:
            type: string
            enum:
              - 5G
              - LTE
          mcc:
            type: integer
            minimum: 1
            maximum: 999
            description: Mobile country code of the network as broadcasted in the serving cell
          mnc:
            type: integer
            minimum: 1
            maximum: 999
            description: Mobile network code of the network as broadcasted in the serving cell
          cellId:
            type: integer
            description: it could be a CGI (if carrier is LTE) or NCGI (if carrier is 5G).
          areaCode:
            type: integer
            description: Routing area code or Traffic area code where client is being served.
  CompEnvParams:
    description: Environment variables are key value pairs that should be injected when component in
instantiated
    type: object
    required:
      - envVarName
      - envValueType
    properties:
      envVarName:
        type: string
        pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
        description: Name of environment variable
      envValueType:
        type: string
        enum:
          - USER_DEFINED
          - PLATFORM_DEFINED_DYNAMIC_PORT
          - PLATFORM_DEFINED_DNS
          - PLATFORM_DEFINED_IP
      envVarValue:
        type: string
        pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
        description: Value to be assigned to environment variable
      envVarSrc:
        type: string
        description: Full path of parameter from componentSpec that should be used to generate the environment
value. Eg. networkResourceProfile[1]. interfaceId.
  CommandLineParams:
    description: List of commands and arguments that shall be invoked when the component instance is created.
This is valid only for container based deployment.
    type: object
    required:
      - command
    properties:
      command:
        type: array
```

```
        items:
          type: string
        description: List of commands that application should invoke when an instance is created.
      commandArgs:
        type: array
        items:
          type: string
        description: List of arguments required by the command.
  DeploymentConfig:
    description: Configuration used when deploying a component. May override other ComponentSpec parameters related to deployment like restart policy, command line parameters, environment variables, etc.
    type: object
    required:
      - configType
      - contents
    properties:
      configType:
        type: string
        enum:
          - DOCKER_COMPOSE
          - KUBERNETES_MANIFEST
          - CLOUD_INIT
          - HELM_VALUES
        description: Config type.
      contents:
        type: string
        description: Contents of the configuration.

  ExposedInterfaces:
    description: Each application component exposes protocol and TCP/UDP ports either for external End-Users or for inter component communication. Application Provider is required to specify the interfaces to be exposed by using ExposedInterfaces parameter either as part of componentSpec or as a request body parameter for Helm type applications.
      type: array
      items:
        $ref: '#/components/schemas/InterfaceDetails'
      minItems: 1



  ComponentSpec:
    description: Details about compute, networking and storage requirements for each component of the application. Application Provider should define all information needed to instantiate the component. If artefact is being defined at component level this section should have information just about the component. In case the artefact is being defined at application level the section should provide details about all the components.
    type: object
    required:
      - componentName
      - images
      - numOfInstances
      - restartPolicy
      - computeResourceProfile
    properties:
      componentName:
        type: string
        pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,62}[A-Za-z0-9]$
        description: Must be a valid RFC 1035 label name.  Component name must be unique with an application
      images:
```

```
        description: List of all images associated with the component. Images are specified using the file identifiers.
Partner OP provides these images using file upload api.
        type: array
        items:
          $ref: '#/components/schemas/FileId'
        minItems: 1
      numOfInstances:
        type: integer
        format: int32
        description: Number of component instances to be launched.
      restartPolicy:
        type: string
        enum:
          - RESTART_POLICY_ALWAYS
          - RESTART_POLICY_NEVER
        description: How the Operator Platform shall handle component failure
      commandLineParams:
        $ref: '#/components/schemas/CommandLineParams'
      exposedInterfaces:
        $ref: '#/components/schemas/ExposedInterfaces'
      computeResourceProfile:
        $ref: '#/components/schemas/ComputeResourceInfo'
      compEnvParams:
        type: array
        items:
          $ref: '#/components/schemas/CompEnvParams'
      deploymentConfig:
        $ref: '#/components/schemas/DeploymentConfig'
      persistentVolumes:
        description: The ephemeral volume a container process may need to temporary store internal data
        type: array
        items:
          $ref: '#/components/schemas/PersistentVolumeDetails'
        minItems: 1
  ComputeResourceInfo:
    type: object
    required:
      - cpuArchType
      - numCPU
      - memory
    properties:
      cpuArchType:
        type: string
        enum:
          - ISA_X86_64
          - ISA_ARM_64
        description: CPU Instruction Set Architecture (ISA) E.g., Intel, Arm etc.
      numCPU:
        $ref: '#/components/schemas/Vcpu'
      memory:
        type: integer
        format: int64
        description: Amount of RAM in Mbytes
      diskStorage:
        type: integer
        format: int32
        description: Amount of disk storage in Gbytes for a given ISA type
      gpu:
```

```
        type: array
        items:
          $ref: '#/components/schemas/GpuInfo'
      vpu:
        type: integer
        description: Number of Intel VPUs available for a given ISA type
      fpga:
        type: integer
        description: Number of FPGAs available for a given ISA type
      hugepages:
        type: array
        items:
          $ref: '#/components/schemas/HugePage'
      cpuExclusivity:
        type: boolean
        description: Support for exclusive CPUs

  nodeDiscoveryResponse:
    type: object
    required:
      - edgeNodes
      - discoveredAppInsts
    properties:
      edgeNodes:
        $ref: '#/components/schemas/DiscoveredEdgeNodes'
      discoveredAppInsts:
        $ref: '#/components/schemas/DiscoveredAppInsts'
    description: Candidate Availability Zones and details of already running instances of the given application


  DiscoveredEdgeNodes:
    type: array
    items:
      type: object
      required:
        - zoneId
        - latencyServiceEndPoints
      properties:
        zoneId:
          $ref: '#/components/schemas/ZoneIdentifier'
        latencyServiceEndPoints:
          $ref: '#/components/schemas/ServiceEndpoint'
    minItems: 1
    description: List of candidate Availability Zones where application instance could be created.
LatencyServiceEndpoint is responsible for responding to latency measurement request from client


  DiscoveredAppInsts:
    type: array
    items:
      type: object
      required:
        - appId
        - appProviderId
        - appInstances
      properties:
        appId:
          $ref: '#/components/schemas/AppIdentifier'
```

```
        appProviderId:
          $ref: '#/components/schemas/AppProviderId'
        appInstances:
          type: array
          items:
            type: object
            required:
              - instancesInfo
            properties:
              instancesInfo:
                type: object
                required:
                  - zoneId
                  - appProviderId
                properties:
                  zoneId:
                    $ref: '#/components/schemas/ZoneIdentifier'
                  instanceDetails:
                    $ref: '#/components/schemas/InstanceDetails'
          minItems: 1

    InstanceDetails:
      type: array
      items:
        type: object
        required:
          - appInstanceInfo
        properties:
          appInstanceInfo:
            type: object
            required:
              - instanceIdentifier
              - instanceState
            properties:
              instanceIdentifier:
                $ref: '#/components/schemas/InstanceIdentifier'
              instancestate:
                $ref: '#/components/schemas/InstanceState'
      minItems: 1

    FederationRequestData:
      type: object
      required:
        - initialDate
        - partnerStatusLink
      properties:
        origOPFederationId:
          $ref: '#/components/schemas/FederationIdentifier'
        origOPCountryCode:
          $ref: '#/components/schemas/CountryCode'
        origOPMobileNetworkCodes:
          $ref: '#/components/schemas/MobileNetworkIds'
        origOPFixedNetworkCodes:
          $ref: '#/components/schemas/FixedNetworkIds'
        initialDate:
          type: string
          format: date-time
          description: Time zone info of the Federation initiated by the Originating OP
```

```
        partnerStatusLink:
          $ref: '#/components/schemas/Uri'

    FederationResponseData:
      type: object
      required:
        - federationContextId
        - platformCaps
      properties:
        partnerOPFederationId:
          $ref: '#/components/schemas/FederationIdentifier'
        partnerOPCountryCode:
          $ref: '#/components/schemas/CountryCode'
        federationContextId:
          $ref: '#/components/schemas/FederationContextId'
        edgeDiscoveryServiceEndPoint:
          $ref: '#/components/schemas/ServiceEndpoint'
        lcmServiceEndPoint:
          $ref: '#/components/schemas/ServiceEndpoint'
        partnerOPMobileNetworkCodes:
          $ref: '#/components/schemas/MobileNetworkIds'
        partnerOPFixedNetworkCodes:
          $ref: '#/components/schemas/FixedNetworkIds'
        offeredAvailabilityZones:
          type: array
          items:
            $ref: '#/components/schemas/ZoneDetails'
          minItems: 1
          description: List of Availability Zones, which the Operator Platform wishes to make available to Application
Providers/ISVs of requesting Operator Platform.
        platformCaps:
          $ref: '#/components/schemas/PlatformCaps'

        federationExpiryDate:
          type: string
          format: date-time
          description: Date and Time zone info of the existing Federation expiry
        federationRenewalDate:
          type: string
          format: date-time
          description: Date and Time zone info of the existing Federation renewal. Shall be less than
federationExpiryDate

    dateAndTimeZoneObject:
      type: string
      format: date-time
      description: Date and Time zone info format
    Flavour:
      type: object
      required:
        - flavourId
        - cpuArchType
        - supportedOSTypes
        - numCPU
        - memorySize
        - storageSize
      properties:
        flavourId:
```

```
        $ref: '#/components/schemas/FlavourId'
      cpuArchType:
        $ref: '#/components/schemas/CPUArchType'
      supportedOSTypes:
        description: A list of operating systems which this flavour configuration can support e.g., RHEL Linux,
Ubuntu 18.04 LTS, MS Windows 2012 R2.
        type: array
        items:
          $ref: '#/components/schemas/OSType'
        minItems: 1
      numCPU:
        type: integer
        format: int32
        description: Number of available vCPUs
      memorySize:
        type: integer
        format: int32
        description: Amount of RAM in Mbytes
      storageSize:
        type: integer
        format: int32
        description: Amount of disk storage in Gbytes
      gpu:
        type: array
        items:
          $ref: '#/components/schemas/GpuInfo'
      fpga:
        type: integer
        format: int32
        description: Number of FPGAs

      vpu:
        type: integer
        description: Number of Intel VPUs available
      hugepages:
        type: array
        items:
          $ref: '#/components/schemas/HugePage'
      cpuExclusivity:
        type: boolean
        description: Support for exclusive CPUs
  GpuInfo:
    type: object
    required:
      - gpuVendorType
      - gpuModeName
      - gpuMemory
      - numGPU
    properties:
      gpuVendorType:
        type: string
        enum:
          - GPU_PROVIDER_NVIDIA
          - GPU_PROVIDER_AMD
        description: GPU vendor name e.g. NVIDIA, AMD etc.
        example: Nvidia
      gpuModeName:
        type: string
```

```
        description: Model name corresponding to vendorType may include info e.g. for NVIDIA, model name could be “Tesla M60”, “Tesla V100” etc.
      gpuMemory:
        type: integer
        description: GPU memory in Mbytes
      numGPU:
        type: integer
        description: Number of GPUs
  HugePage:
    type: object
    required:
      - pageSize
      - number
    properties:
      pageSize:
        type: string
        enum:
          - 2MB
          - 4MB
          - 1GB
        description: Size of hugepage
      number:
        type: integer
        description: Total number of huge pages
  InterfaceDetails:
    type: object
    required:
      - interfaceId
      - commProtocol
      - commPort
      - visibilityType
    properties:
      interfaceId:
        type: string
        description: Each Port and corresponding traffic protocol exposed by the component is identified by a name. Application client on the UE requires this to uniquely identify the interface.
        pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
      commProtocol:
        type: string
        enum:
          - TCP
          - UDP
          - HTTP_HTTPS
        description: Defines the IP transport communication protocol i.e., TCP, UDP or HTTP
      commPort:
        type: integer
        format: int32
        minimum: 1
        maximum: 65535
        description: Port number exposed by the component. OP may generate a dynamic port towards the UCs corresponding to this internal port and forward the client traffic from dynamic port to container Port.
      visibilityType:
        description: Defines whether the interface is exposed to outer world or not i.e., external, or internal. If this is set to "external", then it is exposed to external applications otherwise it is exposed internally to edge application components within an Edge Cloud. When exposed to external world, an external dynamic port is assigned for UC traffic and mapped to the internal container Port
        type: string
        enum:
```

```
          - VISIBILITY_EXTERNAL
          - VISIBILITY_INTERNAL
        network:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
          description: Name of the network.  In case the application has to be associated with more than 1 network then Application Provider must define the name of the network on which this interface has to be exposed.  This parameter is required only if the port has to be exposed on a specific network other than default.
        InterfaceName:
          type: string
          pattern: ^[a-z][a-z0-9]{3}$
          description: Interface Name. Required only if application has to be attached to a network other than default.
    InvalidParam:
      type: object
      properties:
        param:
          type: string
        reason:
          type: string
      required:
        - param
    MobileNetworkIds:
      type: object
      properties:
        mcc:
          $ref: '#/components/schemas/Mcc'
        mncs:
          type: array
          items:
            $ref: '#/components/schemas/Mnc'
          minItems: 1
    ObjectRepoLocation:
      type: object
      properties:
        repoURL:
          $ref: '#/components/schemas/Uri'
        userName:
          type: string
          description: Username to access the repository
        password:
          type: string
          description: Password to access the repository
        token:
          type: string
          description: Authorization token to access the repository
    OSType:
      type: object
      required:
        - architecture
        - distribution
        - version
        - license
      properties:
        architecture:
          type: string
          enum:
            - x86_64
            - x86
```

```
        example: x86_64
      distribution:
        type: string
        enum:
          - RHEL
          - UBUNTU
          - COREOS
          - FEDORA
          - WINDOWS
          - OTHER

      version:
        type: string
        enum:
          - OS_VERSION_UBUNTU_2204_LTS
          - OS_VERSION_RHEL_8
          - OS_VERSION_RHEL_7
          - OS_VERSION_DEBIAN_11
          - OS_VERSION_COREOS_STABLE
          - OS_MS_WINDOWS_2012_R2
          - OTHER

      license:
        type: string
        enum:
          - OS_LICENSE_TYPE_FREE
          - OS_LICENSE_TYPE_ON_DEMAND
          - NOT_SPECIFIED

    RepoType:
      type: string
      enum:
        - PRIVATEREPO
        - PUBLICREPO
        - UPLOAD
      description: Artefact or file repository location. PUBLICREPO is used of public URLs like GitHub, Helm repo, docker registry etc., PRIVATEREPO is used for private repo managed by the Application Provider, UPLOAD is for the case when artefact/file is uploaded from MEC web portal.  OP should pull the image from ‘repoUrl' immediately after receiving the request and then send back the response. In case the repoURL corresponds to a docker registry, use docker v2 http api to do the pull.

    ArtefactName:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
      description: Name of the artefact.

    ArtefactVersionInfo:
      type: string
      description: Artefact version information

    ArtefactDescription:
      type: string
      maxLength: 256
      description: Brief description of the artefact by the Application Provider

    ArtefactVirtType:
      type: string
      enum:
```

```
        - VM_TYPE
        - CONTAINER_TYPE

    ArtefactFileName:
      type: string
      minLength: 8
      maxLength: 32
      description: Name of the file.

    ArtefactFileFormat:
      type: string
      enum:
        - ZIP
        - TAR
        - TEXT
        - TARGZ
      description: Artefacts like Helm charts or Terraform scripts may need compressed format.

    ArtefactDescriptorType:
      type: string
      enum:
        - HELM
        - TERRAFORM
        - ANSIBLE
        - SHELL
        - COMPONENTSPEC
      description: Type of descriptor present in the artefact.  Application Provider can either define either a Helm chart or a Terraform script or container spec.


    LatencyConstraints:
      type: string
      enum:
        - NONE
        - LOW
        - ULTRALOW
      description: Latency requirements for the application.Allowed values (non-standardized) are none, low and ultra-low. Ultra-Low may corresponds to range 15 - 30 msec, Low correspond to range 30 - 50 msec. None means 51 and above

    BandwidthRequired:
      type: integer
      format: int32
      minimum: 1
      description: Data transfer bandwidth requirement (minimum limit) for the application. It should in Mbits/sec

    MobilitySupport:
      type: boolean
      default: false
      description: Indicates if an application is sensitive to User Mobility and can be relocated. Default is “FALSE”

    MultiUserClients:
      type: string
      enum:
        - APP_TYPE_SINGLE_USER
        - APP_TYPE_MULTI_USER
      description: Single user type application are designed to serve just one client. Multi user type application is designed to serve multiple clients
```

```
    NoOfUsersPerAppInst:
      type: integer
      default: 1
      description: Maximum no of clients that can connect to an instance of this application. This parameter is
relevant only for application of type multi user
    AppProvisioning:
      type: boolean
      default: true
      description: Define if application can be instantiated or not

    AppComponents:
      description: An application may consist of more than one component. Each component is associated with a
descriptor and may exposes its services externally or internally.  Application Providers are required to provide
details about all these components, their associated descriptors and their DNS names.
      type: array
      items:
        type: object
        required:
          - componentName
        anyOf:
          - required:
            - serviceNameNB
          - required:
            - serviceNameEW
          - required:
            - artefactId
        properties:
          serviceNameNB:
            $ref: '#/components/schemas/ServiceNameNB'
          serviceNameEW:
            $ref: '#/components/schemas/ServiceNameEW'
          componentName:
            $ref: '#/components/schemas/ComponentName'
          artefactId:
            $ref: '#/components/schemas/ArtefactId'
      minItems: 1


    PersistentVolumeDetails:
      type: object
      required:
        - volumeSize
        - volumeMountPath
        - volumeName
      properties:
        volumeSize:
          type: string
          enum:
            - 10Gi
            - 20Gi
            - 50Gi
            - 100Gi
          description: size of the volume given by user (10GB, 20GB, 50 GB or 100GB)
        volumeMountPath:
          type: string
          description: Defines the mount path of the volume
        volumeName:
```

```
        type: string
        description: Human readable name for the volume
      ephemeralType:
        type: boolean
        default: false
        description: It indicates the ephemeral storage on the node and contents are not preserved if containers
restarts
      accessMode:
        type: string
        enum:
          - RW
          - RO
        default: RW
        description: Values are RW (read/write) and RO (read-only)l
      sharingPolicy:
        type: string
        enum:
          - EXCLUSIVE
          - SHARED
        default: EXCLUSIVE
        description: Exclusive or Shared. If shared, then in case of multiple containers same volume will be shared
across the containers.
  ProblemDetails:
    type: object
    properties:
      title:
        type: string
        description: Summary of the problem
      detail:
        type: string
        description: Specific detail of the issue
      cause:
        type: string
        description: Fixed string indicating cause of the issue
      invalidParams:
        type: array
        items:
          $ref: '#/components/schemas/InvalidParam'
        minItems: 0
  ResourceReservationDuration:
    description: Time period for which resources are to be reserved starting from now
    type: object
    minProperties: 1
    properties:
      numOfDays:
        type: integer
        format: int32
        description: Number of days to be reserved
      numOfMonths:
        type: integer
        format: int32
        description: Number of months to be reserved
      numOfYears:
        type: integer
        format: int32
        description: Number of years to be reserved
  ServiceEndpoint:
    type: object
```

```
      required:
        - port
      anyOf:
        - required:
            - fqdn
        - required:
            - ipv4Addresses
        - required:
            - ipv6Addresses
      properties:
        port:
          $ref: '#/components/schemas/Port'
        fqdn:
          $ref: '#/components/schemas/EdgeAppFQDN'
        ipv4Addresses:
          type: array
          items:
            $ref: '#/components/schemas/Ipv4Addr'
          minItems: 1
        ipv6Addresses:
          type: array
          items:
            $ref: '#/components/schemas/Ipv6Addr'
          minItems: 1
    ZoneDetails:
      type: object
      required:
        - zoneId
        - geographyDetails
      properties:
        zoneId:
          $ref: '#/components/schemas/ZoneIdentifier'
        geolocation:
          $ref: '#/components/schemas/GeoLocation'
        geographyDetails:
          type: string
          description: Details about cities or state covered by the edge. Details about the type of locality for eg rural,
urban, industrial etc. This information is defined in human readable form.
    ZoneRegistrationRequestData:
      type: object
      required:
        - acceptedAvailabilityZones
      properties:
        acceptedAvailabilityZones:
          type: array
          items:
            $ref: '#/components/schemas/ZoneIdentifier'
          minItems: 1
        availZoneNotifLink:
          $ref: '#/components/schemas/Uri'
    ZoneRegistrationResponseData:
      type: object
      required:
        - acceptedZoneResourceInfo
      properties:
        acceptedZoneResourceInfo:
          type: array
          items:
```

```
          $ref: '#/components/schemas/ZoneRegisteredData'

        minItems: 1
    ZoneRegisteredData:
      type: object
      required:
        - zoneId
        - reservedComputeResources
        - computeResourceQuotaLimits
        - flavoursSupported
      properties:
        zoneId:
          $ref: '#/components/schemas/ZoneIdentifier'
        reservedComputeResources:
          description: Resources exclusively reserved for the originator OP.
          type: array
          items:
            $ref: '#/components/schemas/ComputeResourceInfo'
          minItems: 1
        computeResourceQuotaLimits:
          description: Max quota on resources Partner OP allows over reserved resources.
          type: array
          items:
            $ref: '#/components/schemas/ComputeResourceInfo'
          minItems: 1
        flavoursSupported:
          type: array
          items:
            $ref: '#/components/schemas/Flavour'
          minItems: 1
        networkResources:
          type: object
          required:
            - egressBandWidth
            - dedicatedNIC
            - supportSriov
            - supportDPDK
          properties:
            egressBandWidth:
              type: integer
              format: int32
              description: Max dl throughput that this edge can offer. It is defined in Mbps.
            dedicatedNIC:
              type: integer
              format: int32
              description: Number of network interface cards which can be dedicatedly assigned to application pods
on isolated networks. This includes virtual as well physical NICs
            supportSriov:
              type: boolean
              description: If this Availability Zone support SRIOV networks or not
            supportDPDK:
              type: boolean
              description: If this Availability Zone supports DPDK based networking.
        zoneServiceLevelObjsInfo:
          type: object
          description: It is a measure of the actual amount of data that is being sent over a network per unit of time
and indicates máximum supported value for an Availability Zone
          required:
```

```
        - latencyRanges
        - jitterRanges
        - throughputRanges
      properties:
        latencyRanges:
          type: object
          properties:
            minLatency:
              type: integer
              format: int32
              minimum: 1
              description: The time for data/packet to reach from UC to edge application. It represent mínimum
latency in milli seconds that may exist between UCs and edge apps in this Availability Zone but it can be higher in
actual.
            maxLatency:
              type: integer
              format: int32
              description: The maximum limit of latency between UC and Edge Application in milli seconds.
        jitterRanges:
          type: object
          properties:
            minJitter:
              type: integer
              format: int32
              minimum: 1
            maxJitter:
              type: integer
              format: int32
              description: The maximum limit of network jitter between UC and Edge Application in milli seconds.
        throughputRanges:
          type: object
          properties:
            minThroughput:
              type: integer
              format: int32
              minimum: 1
              description: The minimum limit of network throughput between UC and Edge Application in Mega bits
per seconds (Mbps).
            maxThroughput:
              type: integer
              format: int32
              description: The maximum limit of network throughput between UC and Edge Application in Mega bits
per seconds (Mbps).

        #
        # HTTP responses
        #
responses:
  "400":
    description: Bad request
    content:
      application/problem+json:
        schema:
          $ref: '#/components/schemas/ProblemDetails'
  "401":
    description: Unauthorized
    content:
      application/problem+json:
```

```
        schema:
          $ref: '#/components/schemas/ProblemDetails'
    "404":
      description: Not Found
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "409":
      description: Conflict
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "412":
      description: Precondition Failed
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "422":
      description: Unprocessable Entity
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "500":
      description: Internal Server Error
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "501":
      description: Not Implemented
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "503":
      description: Service Unavailable
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    "520":
      description: Web Server Returned an Unknown Error
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'

    "400BadRequest":
      description: Bad request
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
          examples:
```

```
          InvalidFedParameters:
            description: Sufficient parameters must be specified to allow the Partner OP to validate Federation
request
            value:
              {
                "title": "Insufficient parameters",
                "details": "Incorrect values received in federation request",
                "cause": "INVALID_FED_RQST_PARAMS"
              }

    "404NotFound":
      description: Resource Not Found
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
          examples:
            FederationContextNotFound:
              description: Federation context does not exist
              value:
                {
                  "title": "Federation context Id not found",
                  "details": "Partner OP does not recognize the federationContextId from Originating OP",
                  "cause": "INVALID_FED_CTX_ID"
                }
            FederationNotFound:
              description: Federation terminated parmanently
              value:
                {
                  "title": "Federation context Id not found",
                  "details": "Partner OP does not recognize the federationContextId from Originating OP",
                  "cause": "FED_PERMANENTLY_TERMINAT"
                }
            ZoneNotFound:
              description: Zone Not Found
              value:
                {
                  "title": "Requested Zone Id not found",
                  "details": "Requested zone by the Originating OP does not exist with Partner OP",
                  "cause": "ZONE_ID_NOT_FOUND"
                }
            AppNotFound:
              description: Application Not Found
              value:
                {
                  "title": "Requested Application Id not found",
                  "details": "Requested Application by the Originating OP does not exist with Partner OP",
                  "cause": "APP_ID_NOT_FOUND"
                }
            AppInstNotFound:
              description: Application Instance Not Found
              value:
                {
                  "title": "Requested App instance Id not found",
                  "details": "Requested application instance by the Originating OP does not exist with Partner OP",
                  "cause": "APP_INST_NOT_FOUND"
                }
```

```
    default:
      description: Generic Error
paths:
  /federation-resources:
    get:
      summary: Retrieves REST APIs supported by an OP for Federation services.
      operationId: GetFederationAPIs
      tags:
        - FederationAPIManagement
      responses:
        "200":
          description: Federation meta-info request accepted
          content:
            application/json:
              schema:
                type: object
                required:
                  - federationSupportedAPIs
                properties:
                  federationSupportedAPIs:
                    $ref: '#/components/schemas/FederationSupportedAPIs'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /partner:
    post:
      summary: Creates one direction Federation with Partner OP.
      operationId: CreateFederation
      tags:
        - FederationManagement
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/FederationRequestData'
      responses:
        "200":
          description: Federation meta-info request accepted
          content:
            application/json:
              schema:
```

```
            $ref: '#/components/schemas/FederationResponseData'
        headers:
          Location:
            description: 'Contains the URI of the newly created resource, according to the structure:
{apiRoot}/operatorplatform/federation/v1/partner/{federationContextId}'
            required: true
            schema:
              type: string
          Accept-Encoding:
            description: Accept-Encoding, described in IETF RFC 7694
            schema:
              type: string
          Content-Encoding:
            description: Content-Encoding, described in IETF RFC 7231
            schema:
              type: string
      "400":
        $ref: '#/components/responses/400BadRequest'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onPartnerStatusEvent:
        '{$request.body#/partnerStatusLink }':
          post:
            requestBody:
              description: |
                OP uses this Callback API to notify Partner OP about change in Federation status, Federation
metadata or offered Availability Zone details. Allowed combinations of objectType and operationType are
                - FEDERATION - STATUS: Status specified by parameter 'federationStatus'.
                - ZONES - STATUS: Status specified by parameter 'zoneStatus'.
                - ZONES - ADD: Use parameter 'addZones' to define add new zones
                - ZONES - REMOVE: Use parameter 'removeZones' to define remove zones.
                - EDGE_DISCOVERY_SERVICE - UPDATE: Use parameter 'edgeDiscoverySvcEndPoint' to specify
new endpoints
                - LCM_SERVICE - UPDATE: Use parameter 'lcmSvcEndPoint' to specify new endpoints
                - MOBILE_NETWORK_CODES - ADD: Use parameter 'addMobileNetworkIds' to define new mobile
network codes.
                - MOBILE_NETWORK_CODES - REMOVE: Use parameter 'removeMobileNetworkIds' to remove
mobile network codes.
                - FIXED_NETWORK_CODES - ADD: Use parameter 'addFixedNetworkIds' to define new fixed
network codes.
                - FIXED_NETWORK_CODES - REMOVE: Use parameter 'removeFixedNetworkIds' to remove fixed
network codes.
```

- SERVICE_APIS - ADD/REMOVE: Parameter Usage 'addServiceAPIs / removeServiceAPIs' to add or remove Service APIs support.

```
          content:
            application/json:
              schema:
                type: object
                required:
                  - federationContextId
                  - objectType
                  - operationType
                  - modificationDate
                properties:
                  federationContextId:
                    $ref: '#/components/schemas/FederationContextId'
                  objectType:
                    type: string
                    enum:
                      - FEDERATION
                      - ZONES
                      - EDGE_DISCOVERY_SERVICE
                      - LCM_SERVICE
                      - MOBILE_NETWORK_CODES
                      - FIXED_NETWORK_CODES
                      - SERVICE_APIS
                  operationType:
                    type: string
                    enum:
                      - STATUS
                      - UPDATE
                      - ADD
                      - REMOVE
                  edgeDiscoverySvcEndPoint:
                    $ref: '#/components/schemas/ServiceEndpoint'
                  lcmSvcEndPoint:
                    $ref: '#/components/schemas/ServiceEndpoint'
                  addMobileNetworkIds:
                    $ref: '#/components/schemas/MobileNetworkIds'
                  removeMobileNetworkIds:
                    $ref: '#/components/schemas/MobileNetworkIds'
                  addFixedNetworkIds:
                    $ref: '#/components/schemas/FixedNetworkIds'
                  removeFixedNetworkIds:
                    $ref: '#/components/schemas/FixedNetworkIds'
                  addZones:
                    type: array
                    items:
                      $ref: '#/components/schemas/ZoneDetails'
                    description: List of Availability Zones, which the Operator Platform wishes to make available to Application Providers/ISVs of requesting Operator Platform.
                    minItems: 1
                  removeZones:
                    type: array
                    items:
                      $ref: '#/components/schemas/ZoneIdentifier'
                    description: List of Availability Zones, which the Operator Platform no longer wishes to share.
                    minItems: 1
                  addServiceAPIs:
```

```
                $ref: '#/components/schemas/serviceAPINames'
              removeServiceAPIs:
                $ref: '#/components/schemas/serviceAPINames'
              zoneStatus:
                type: array
                items:
                  type: object
                  required:
                    - zoneId
                    - status
                  properties:
                    zoneId:
                      $ref: '#/components/schemas/ZoneIdentifier'
                    status:
                      $ref: '#/components/schemas/Status'
                minItems: 1
              federationStatus:
                $ref: '#/components/schemas/Status'
              modificationDate:
                type: string
                format: date-time
                description: Date and time of the Federation modification by the originating Partner OP
          responses:
            "204":
              description: Expected response to a successful call back processing
            "400":
              $ref: '#/components/responses/400'
            "401":
              $ref: '#/components/responses/401'
            "404":
              $ref: '#/components/responses/404'
            "409":
              $ref: '#/components/responses/409'
            "422":
              $ref: '#/components/responses/422'
            "500":
              $ref: '#/components/responses/500'
            "503":
              $ref: '#/components/responses/503'
            "520":
              $ref: '#/components/responses/520'
            default:
              $ref: '#/components/responses/default'
          security:
            - notifClientCredentials: [fed-mgmt-notif]
  /{federationContextId}/partner:
    get:
      summary: Retrieves details about the Federation context with the ¨Partner OP. The response shall provide info about the Availability Zones offered by the partner, Partner OP network codes, information about Edge Discovery and LCM Service etc.
      operationId: GetFederationDetails
      tags:
        - FederationManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
```

```
          $ref: '#/components/schemas/FederationContextId'
      responses:
        "200":
          description: Federation meta-info request accepted
          content:
            application/json:
              schema:
                type: object
                properties:
                  edgeDiscoveryServiceEndPoint:
                    $ref: '#/components/schemas/ServiceEndpoint'
                  lcmServiceEndPoint:
                    $ref: '#/components/schemas/ServiceEndpoint'
                  allowedMobileNetworkIds:
                    $ref: '#/components/schemas/MobileNetworkIds'
                  allowedFixedNetworkIds:
                    $ref: '#/components/schemas/FixedNetworkIds'
                  offeredAvailabilityZones:
                    type: array
                    items:
                      $ref: '#/components/schemas/ZoneDetails'
                    minItems: 1
                    description: List of Availability Zones, which the Operator Platform wishes to make available to
Application Providers/ISVs of requesting Operator Platform.
                platformCaps:
                  $ref: '#/components/schemas/PlatformCaps'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
    patch:
      summary: API used by the Originating OP towards the Partner OP, to update the parameters associated to
the existing Federation
      operationId: UpdateFederation
      tags:
        - FederationManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        required: true
```

```
      description: Details about changes origination OP wished to apply
      content:
        application/json:
          schema:
            type: object
            required:
              - objectType
              - operationType
              - modificationDate
            properties:
              objectType:
                type: string
                enum:
                  - MOBILE_NETWORK_CODES
                  - FIXED_NETWORK_CODES
                  - OPS_POLICY
                  - APP_POLICY
              operationType:
                type: string
                enum:
                  - ADD_CODES
                  - REMOVE_CODES
                  - UPDATE_CODES
                  - ADD_POLICY
                  - REMOVE_POLICY
                  - UPDATE_POLICY
              addMobileNetworkIds:
                $ref: '#/components/schemas/MobileNetworkIds'
              removeMobileNetworkIds:
                $ref: '#/components/schemas/MobileNetworkIds'
              addFixedNetworkIds:
                $ref: '#/components/schemas/FixedNetworkIds'
              removeFixedNetworkIds:
                $ref: '#/components/schemas/FixedNetworkIds'
              assocAppPolicies:
                $ref: '#/components/schemas/AssocApplPolicies'
              assocOpsPolicies:
                $ref: '#/components/schemas/AssocOpsPolicies'
              modificationDate:
                type: string
                format: date-time
                description: Date and time of the Federation modification by the originating Partner OP
    responses:
      "200":
        description: Federation meta-info request accepted
        content:
          application/json:
            schema:
              type: object
              properties:
                edgeDiscoveryServiceEndPoint:
                  $ref: '#/components/schemas/ServiceEndpoint'
                lcmServiceEndPoint:
                  $ref: '#/components/schemas/ServiceEndpoint'
                allowedMobileNetworkIds:
                  $ref: '#/components/schemas/MobileNetworkIds'
                allowedFixedNetworkIds:
                  $ref: '#/components/schemas/FixedNetworkIds'
```

```
            offeredAvailabilityZones:
              type: array
              items:
                $ref: '#/components/schemas/ZoneDetails'
              minItems: 1
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  delete:
    summary: Remove existing Federation with the Partner OP
    operationId: DeleteFederationDetails
    tags:
      - FederationManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
    responses:
      "200":
        description: Federation removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /fed-context-id:
    get:
```

```
    summary: Retrieves the existing federationContextId with Partner OP.
    operationId: GetFederationContextId
    tags:
      - FederationManagement
    responses:
      "200":
        description: Federation context identifier retrieval request accepted
        content:
          application/json:
            schema:
              type: object
              required:
                - FederationContextId
              properties:
                FederationContextId:
                  $ref: '#/components/schemas/FederationContextId'
        headers:
          Location:
            description: 'Contains the URI of the existing resource, according to the structure:
{apiRoot}/operatorplatform/federation/v1/partner/{federationContextId}'
            required: true
            schema:
              type: string
          Accept-Encoding:
            description: Accept-Encoding, described in IETF RFC 7694
            schema:
              type: string
          Content-Encoding:
            description: Content-Encoding, described in IETF RFC 7231
            schema:
              type: string
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/health:
    get:
      summary: Retrieves health status of the Federation context with the Partner OP.
      operationId: GetFederationHealth
      tags:
        - FederationManagement
      parameters:
        - name: federationContextId
          in: path
```

```
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      responses:
        "200":
          description: Federation health status information object
          content:
            application/json:
              schema:
                type: object
                required:
                  - federationHealthStatus
                properties:
                  federationHealthStatus:
                    $ref: '#/components/schemas/FederationHealthInfo'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /{federationContextId}/renew:
    post:
      summary: API used by the Originating OP to renew the existing Federation
      operationId: RenewFederation
      tags:
        - FederationManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      responses:
        "200":
          description: Federation renewal request accepted
          content:
            application/json:
              schema:
                type: object
                required:
                  - FederationContextId
                  - federationRenewalDate
                  - federationExpiryDate
                properties:
                  FederationContextId:
```

```
              $ref: '#/components/schemas/FederationContextId'
            federationRenewalDate:
              $ref: '#/components/schemas/dateAndTimeZoneObject'
            federationExpiryDate:
              $ref: '#/components/schemas/dateAndTimeZoneObject'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/platform-caps:
    get:
      summary: Retrieves details about OP capabilities of the federated partner.
      operationId: GetPlatformCapabilities
      tags:
        - FederationManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: capType
          in: query
          required: false
          schema:
            $ref: '#/components/schemas/CapabilityID'

      responses:
        "200":
          description: Federation meta-info request accepted
          content:
            application/json:
              schema:
                type: object
                anyOf:
                  - required:
                    - deviceConnStatusChangeCap
                  - required:
                    - locationRetrievalCap
                  - required:
                    - userPlaneMgmtEvtCap
                  - required:
                    - dynamicQoSCap
```

```
            properties:
              deviceConnStatusChangeCap:
                $ref: '#/components/schemas/DeviceConnStatusChangeCap'
              locationRetrievalCap:
                $ref: '#/components/schemas/LocationRetrievalCap'
              userPlaneMgmtEvtCap:
                $ref: '#/components/schemas/UserPlaneMgmtEvtCap'
              dynamicQoSCap:
                $ref: '#/components/schemas/DynamicQoSCap'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  /{federationContextId}/partner/service/{serviceType}:
    get:
      summary: Retrieves the list of Service APIs and associated information that a Partner OP supports
      operationId: GetServiceAPIsDetails
      tags:
        - FederationManagement
      parameters:
        -   name: federationContextId
            in: path
            required: true
            schema:
              $ref:  '#/components/schemas/FederationContextId'
        -   name: serviceType
            in: path
            required: true
            schema:
              $ref:  '#/components/schemas/serviceType'
      responses:
        '200':
          description: List of Service APIs names and associated configuration info as supported capabilities
          content:
            application/json:
              schema:
                type: object
                required:
                  - ServiceType
                  - serviceCaps
                  - apiRoutingInfo
                properties:
```

```
            serviceCaps:
              $ref: '#/components/schemas/serviceAPINames'
            serviceType:
              $ref: '#/components/schemas/serviceType'
            apiRoutingInfo:
              $ref: '#/components/schemas/serviceRoutingInfo'
      '400':
        $ref: '#/components/responses/400'
      '401':
        $ref: '#/components/responses/401'
      '404':
        $ref: '#/components/responses/404NotFound'
      '409':
        $ref: '#/components/responses/409'
      '422':
        $ref: '#/components/responses/422'
      '500':
        $ref: '#/components/responses/500'
      '503':
        $ref: '#/components/responses/503'
      '520':
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/zones:
    get:
      summary: Retrieves details about the Edge and Network Resources that Partner OP has reserved for this
Availability Zone.
      operationId: GetZoneData
      tags:
        - AvailabilityZoneInfoSynchronization
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: zoneId
          in: query
          required: false
          schema:
            $ref: '#/components/schemas/ZoneIdentifier'
      responses:
        "200":
          description: Zone metadata
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ZoneRegisteredData'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
```

```
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  post:
    summary: Originating OP informs Partner OP that it is willing to access the specified Availability Zones and
Partner OP shall reserve Edge and Network Resources for these Availability Zones.
    operationId: ZoneSubscribe
    tags:
      - AvailabilityZoneInfoSynchronization
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ZoneRegistrationRequestData'
      required: true
    responses:
      "200":
        description: Zone registered successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ZoneRegistrationResponseData'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onZoneResourceUpdateEvent:
        '{$request.body#/availZoneNotifLink}':
          post:
```

```
      requestBody:
        description: Notification about resource availability.
        content:
          application/json:
            schema:
              type: object
              required:
                - federationContextId
                - zoneId
                - zoneResUpdInfo
              properties:
                federationContextId:
                  $ref: '#/components/schemas/FederationIdentifier'
                zoneId:
                  $ref: '#/components/schemas/ZoneIdentifier'
                zoneResUpdInfo:
                  type: array
                  items:
                    type: object
                    minProperties: 1
                    properties:
                      availableCompResources:
                        description: Resources exclusively reserved for the originator OP.
                        type: array
                        items:
                          $ref: '#/components/schemas/ComputeResourceInfo'
                        minItems: 1
                      availableNetResources:
                        type: object
                        properties:
                          egressBandWidth:
                            type: integer
                            format: int32
                            description: Max dl throughput that this edge can offer. It is defined in Mbps.
                          dedicatedNIC:
                            type: integer
                            format: int32
                          supportSriov:
                            type: boolean
                            description: If this Availability Zone support SRIOV networks or not
                          supportDPDK:
                            type: boolean
                            description: If this Availability Zone supports DPDK based networking
                        minProperties: 1
      responses:
        "200":
          description: Zone info notification acknowledged
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
```

```
            $ref: '#/components/responses/500'
          "503":
            $ref: '#/components/responses/503'
          "520":
            $ref: '#/components/responses/520'
          default:
            $ref: '#/components/responses/default'
        security:
          - notifClientCredentials: [fed-mgmt-notif]

  /{federationContextId}/zones/{zoneId}:
    delete:
      summary: Assert usage of a Partner OP Availability Zone. Originating OP informs Partner OP that it will no
longer access the specified Availability Zone.
      operationId: ZoneUnsubscribe
      tags:
        - AvailabilityZoneInfoSynchronization
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: zoneId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/ZoneIdentifier'
      responses:
        "200":
          description: Zone deregistered successfully
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
    get:
      summary: Retrieves details about the Edge and Network Resources that Partner OP has reserved for this
Availability Zone.
      operationId: GetZoneDetails
      tags:
        - AvailabilityZoneInfoSynchronization
      parameters:
        - name: federationContextId
          in: path
```

```
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: zoneId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ZoneIdentifier'
    responses:
      "200":
        description: Zone metadata
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ZoneRegisteredData'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/artefact:
    post:
      summary: Uploads application artefact on Partner OP. Artefact is a zip file containing scripts and/or
packaging files like Terraform or Helm which are required to create an instance of an application.
      operationId: UploadArtefact
      tags:
        - ArtefactManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        description: An application can consist of multiple components. Application Providers are allowed to define
separate artefacts for each component or they could define a consolidated artefact at application level.
        content:
          multipart/form-data:
            schema:
              type: object
              required:
                - artefactId
                - appProviderId
                - artefactName
```

```
            - artefactVersionInfo
            - artefactVirtType
            - artefactDescriptorType
          properties:
            artefactId:
              $ref: '#/components/schemas/ArtefactId'
            appProviderId:
              $ref: '#/components/schemas/AppProviderId'
            artefactName:
              $ref: '#/components/schemas/ArtefactName'
            artefactVersionInfo:
              $ref: '#/components/schemas/ArtefactVersionInfo'
            artefactDescription:
              $ref: '#/components/schemas/ArtefactDescription'
            artefactVirtType:
              $ref: '#/components/schemas/ArtefactVirtType'
            artefactFileName:
              $ref: '#/components/schemas/ArtefactFileName'
            artefactFileFormat:
              $ref: '#/components/schemas/ArtefactFileFormat'
            artefactDescriptorType:
              $ref: '#/components/schemas/ArtefactDescriptorType'
            repoType:
              $ref: '#/components/schemas/RepoType'
            artefactRepoLocation:
              $ref: '#/components/schemas/ObjectRepoLocation'
            artefactFile:
              type: string
              format: binary
              description: Helm archive/Terraform archive/container spec file or Binary image associated with an
application component.
            componentSpec:
              type: array
              items:
                $ref: '#/components/schemas/ComponentSpec'
              minItems: 1
            exposedInterfaces:
              $ref: '#/components/schemas/ExposedInterfaces'
      required: true
    responses:
      "200":
        description: Artefact uploaded successfully
      "202":
        description: Artefact upload request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
```

```
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/artefact/{artefactId}:
    get:
      summary: Retrieves details about an artefact.
      operationId: GetArtefact
      tags:
        - ArtefactManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: artefactId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/ArtefactId'
      responses:
        "200":
          description: Artefact details
          content:
            application/json:
              schema:
                type: object
                required:
                  - artefactId
                  - appProviderId
                  - artefactName
                  - artefactVersionInfo
                  - artefactVirtType
                  - artefactDescriptorType
                properties:
                  artefactId:
                    $ref: '#/components/schemas/ArtefactId'
                  appProviderId:
                    $ref: '#/components/schemas/AppProviderId'
                  artefactName:
                    $ref: '#/components/schemas/ArtefactName'
                  artefactDescription:
                    $ref: '#/components/schemas/ArtefactDescription'
                  artefactVersionInfo:
                    $ref: '#/components/schemas/ArtefactVersionInfo'
                  artefactVirtType:
                    $ref: '#/components/schemas/ArtefactVirtType'
                  artefactFileName:
                    $ref: '#/components/schemas/ArtefactFileName'
                  artefactFileFormat:
                    $ref: '#/components/schemas/ArtefactFileFormat'
                  artefactDescriptorType:
                    $ref: '#/components/schemas/ArtefactDescriptorType'
                  repoType:
                    $ref: '#/components/schemas/RepoType'
                  artefactRepoLocation:
                    $ref: '#/components/schemas/ObjectRepoLocation'
```

```
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  delete:
    summary: Removes an artefact from Partner OP.
    operationId: RemoveArtefact
    tags:
      - ArtefactManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: artefactId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ArtefactId'
    responses:
      "200":
        description: Artefact deletion successful
      "202":
        description: Artefact deletion request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/files:
```

```
    post:
      summary: Uploads an image file. Originating OP uses this API to onboard an application image to Partner
OP.
      operationId: UploadFile
      tags:
        - ArtefactManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        content:
          multipart/form-data:
            schema:
              type: object
              required:
                - fileId
                - appProviderId
                - fileName
                - fileVersionInfo
                - fileType
                - imgOSType
                - imgInsSetArch
              properties:
                fileId:
                  $ref: '#/components/schemas/FileId'
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                fileName:
                  $ref: '#/components/schemas/FileName'
                fileDescription:
                  $ref: '#/components/schemas/FileDescription'
                fileVersionInfo:
                  $ref: '#/components/schemas/FileVersionInfo'
                fileType:
                  $ref: '#/components/schemas/VirtImageType'
                checksum:
                  type: string
                  description: MD5 checksum for VM and file-based images, sha256 digest for containers
                imgOSType:
                  $ref: '#/components/schemas/OSType'
                imgInsSetArch:
                  $ref: '#/components/schemas/CPUArchType'
                repoType:
                  $ref: '#/components/schemas/RepoType'

                fileRepoLocation:
                  $ref: '#/components/schemas/ObjectRepoLocation'
                file:
                  type: string
                  format: binary
                  description: Binary image associated with an application component.
        required: true
      responses:
        "200":
          description: File uploaded successfully
```

```
      "202":
        description: File upload request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/files/{fileId}:
    delete:
      summary: Removes an image file from Partner OP.
      operationId: RemoveFile
      tags:
        - ArtefactManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: fileId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FileId'
      responses:
        "200":
          description: Image deletion successful
        "202":
          description: Image deletion request accepted
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
```

```
      default:
        $ref: '#/components/responses/default'
  get:
    summary: View an image file from Partner OP.
    operationId: ViewFile
    tags:
      - ArtefactManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: fileId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FileId'
    responses:
      "200":
        description: Image details
        content:
          application/json:
            schema:
              type: object
              required:
                - fileId
                - appProviderId
                - fileName
                - fileVersionInfo
                - fileType
                - imgOSType
                - imgInsSetArch
              properties:
                fileId:
                  $ref: '#/components/schemas/FileId'
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                fileName:
                  $ref: '#/components/schemas/FileName'
                fileDescription:
                  $ref: '#/components/schemas/FileDescription'
                fileVersionInfo:
                  $ref: '#/components/schemas/FileVersionInfo'
                fileType:
                  $ref: '#/components/schemas/VirtImageType'
                checksum:
                  type: string
                  description: MD5 checksum for VM and file-based images, sha256 digest for containers
                imgOSType:
                  $ref: '#/components/schemas/OSType'
                imgInsSetArch:
                  $ref: '#/components/schemas/CPUArchType'
                repoType:
                  $ref: '#/components/schemas/RepoType'

                fileRepoLocation:
                  $ref: '#/components/schemas/ObjectRepoLocation'
```

```
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/application/onboarding:
    post:
      summary: Submits an application details to a Partner OP. Based on the details provided,  Partner OP shall do bookkeeping, resource validation and other pre-deployment operations.
      operationId: OnboardApplication
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        required: true
        description: Details about application Edge Resource requirements, associated artefacts, QoS Performance Profile and Regions where application shall be made available etc.
        content:
          application/json:
            schema:
              type: object
              required:
                - appId
                - appProviderId
                - appMetaData
                - appQoSProfile
                - appComponentSpecs
              properties:
                appId:
                  $ref: '#/components/schemas/AppIdentifier'
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                appDeploymentZones:
                  description: Details about Partner OP Availability Zones where the application should be made available;  This field when specified will instruct the OP to restrict application instantiation only on the listed Availability Zones.
                  type: array
                  items:
                    $ref: '#/components/schemas/ZoneIdentifier'
                  minItems: 1
```

```
            appMetaData:
              $ref: '#/components/schemas/AppMetaData'
            appQoSProfile:
              $ref: '#/components/schemas/AppQoSProfile'
            appComponentSpecs:
              $ref: '#/components/schemas/AppComponentSpecs'
            appStatusCallbackLink:
              $ref: '#/components/schemas/Uri'
            edgeAppFQDN:
              $ref: '#/components/schemas/EdgeAppFQDN'

    responses:
      "200":
        description: Application onboarded successfully
      "202":
        description: Application onboarding request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onApplicationOnboardStatusEvent:
        '{$request.body#/appStatusCallbackLink}':
          post:
            requestBody:
              description: Notification payload.
              content:
                application/json:
                  schema:
                    type: object
                    required:
                      - federationContextId
                      - appId
                      - statusInfo
                    properties:
                      federationContextId:
                        $ref: '#/components/schemas/FederationIdentifier'
                      appId:
                        $ref: '#/components/schemas/AppIdentifier'
                      statusInfo:
                        type: array
                        items:
                          type: object
                          required:
```

```
                - zoneId
                - onboardStatusInfo
              properties:
                zoneId:
                  $ref: '#/components/schemas/ZoneIdentifier'
                onboardStatusInfo:
                  $ref: '#/components/schemas/OnboardStatusInfo'
            minItems: 1
      responses:
        "204":
          description: Application status updated
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
      security:
        - notifClientCredentials: [fed-mgmt-notif]

  /{federationContextId}/application/onboarding/app/{appId}:
    delete:
      summary: Deboards the application from all Availability Zones, if any, and deletes the Application.
      operationId: DeleteApp
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
        - name: appId
          in: path
          required: true
          schema:
           $ref:  '#/components/schemas/AppIdentifier'
      responses:
        '200':
          description: App deletion successful
        '202':
          description: App deletion request accepted
        '400':
            $ref: '#/components/responses/400'
        '401':
            $ref: '#/components/responses/401'
```

```
        '404':
          $ref: '#/components/responses/404NotFound'
        '409':
          $ref: '#/components/responses/409'
        '422':
          $ref: '#/components/responses/422'
        '500':
          $ref: '#/components/responses/500'
        '503':
          $ref: '#/components/responses/503'
        '520':
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
    patch:
      summary: Updates Partner OP about changes in application Edge Resource requirements, QoS Performance
Profile, associated descriptor or change in associated components
      operationId: UpdateApplication
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: appId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/AppIdentifier'
      requestBody:
        required: true
        description: Details about application Edge Resource requirements, associated artefact and QoS
Performance Profile that needs to be updated.
        content:
          application/json:
            schema:
              type: object
              minProperties: 1
              properties:
                appUpdQoSProfile:
                  description: Parameters corresponding to the performance constraints, tenancy details etc.
                  type: object
                  anyOf:
                    - required:
                        - latencyConstraint
                    - required:
                        - bandwidthRequired
                    - required:
                        - mobilitySupport
                    - required:
                        - multiUserClients
                    - required:
                        - appProvisioning
                  properties:
                    latencyConstraints:
                      $ref: '#/components/schemas/LatencyConstraints'
```

```
            bandwidthRequired:
              $ref: '#/components/schemas/BandwidthRequired'
            mobilitySupport:
              $ref: '#/components/schemas/MobilitySupport'
            multiUserClients:
              $ref: '#/components/schemas/MultiUserClients'
            noOfUsersPerAppInst:
              $ref: '#/components/schemas/NoOfUsersPerAppInst'
            appProvisioning:
              $ref: '#/components/schemas/AppProvisioning'
          appComponents:
            $ref: '#/components/schemas/AppComponents'

    responses:
      "200":
        description: Application update successful
      "202":
        description: Application update request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  get:
    summary: Retrieves application details from Partner OP
    operationId: ViewApplication
    tags:
      - ApplicationOnboardingManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppIdentifier'
    responses:
      "200":
        description: Application details
        content:
          application/json:
            schema:
```

```
            type: object
            required:
              - appId
              - appProviderId
              - appDeploymentZones
              - appMetaData
              - appQoSProfile
              - appComponentSpecs
              - onboardStatusInfo
            properties:
              appId:
                $ref: '#/components/schemas/AppIdentifier'
              appProviderId:
                $ref: '#/components/schemas/AppProviderId'
              appDeploymentZones:
                description: Details about Partner OP Availability Zones where the application should be made
available;  This field when specified will instruct the OP to restrict application instantiation only on the listed
zones.
                type: array
                items:
                  type: object
                  required:
                    - countryCode
                    - zoneInfo
                  properties:
                    countryCode:
                      $ref: '#/components/schemas/CountryCode'
                    zoneInfo:
                      $ref: '#/components/schemas/ZoneIdentifier'
                minItems: 1
              appMetaData:
                $ref: '#/components/schemas/AppMetaData'
              appQoSProfile:
                $ref: '#/components/schemas/AppQoSProfile'
              appComponentSpecs:
                $ref: '#/components/schemas/AppComponentSpecs'
              onboardStatusInfo:
                $ref: '#/components/schemas/OnboardStatusInfo'

      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/application/onboarding/app/{appId}/zone/{zoneId}:
```

```
    delete:
      summary: Deboards an application from specific Partner OP Availability Zones
      operationId: DeboardApplication
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: appId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/AppIdentifier'
        - name: zoneId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/ZoneIdentifier'
      responses:
        "200":
          description: Application deboarded successfully
        "202":
          description: Application deboard request accepted
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /{federationContextId}/application/onboarding/app/{appId}/additionalZones:
    post:
      summary: Onboards an existing application to a new Availability Zone within Partner OP.
      operationId: OnboardExistingAppNewZones
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: appId
          in: path
```

```
        required: true
        schema:
          $ref: '#/components/schemas/AppIdentifier'
      requestBody:
        required: true
        description: Details about new Availability Zones where application shall be made available
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/ZoneIdentifier'
              minItems: 1
      responses:
        "200":
          description: Application onboarding successful
        "202":
          description: Application onboarding request accepted
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /{federationContextId}/application/onboarding/app/{appId}/zoneForbid:
    post:
      summary: Forbid/allow application instantiation on a partner Availability Zone
      operationId: LockUnlockApplicationZone
      tags:
        - ApplicationOnboardingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: appId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/AppIdentifier'
      requestBody:
        required: true
        content:
          application/json:
            schema:
```

```
            type: array
            items:
              type: object
              description: List of Availability Zones where application instantiation shall be forbidden or allowed.
            required:
              - zoneId
              - forbid
            properties:
              zoneId:
                $ref: '#/components/schemas/ZoneIdentifier'
              forbid:
                type: boolean
                description: Value 'true' will forbid application instantiation on this Availability Zone.  No new instance
of the application can be created on this Availability Zone.
            minItems: 1
      responses:
        "200":
          description: Application forbid/permit request accepted
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /{federationContextId}/application/lcm:
    post:
      summary: Instantiates an application on a Partner OP Availability Zone.
      operationId: InstallApp
      tags:
        - ApplicationDeploymentManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: Idempotency-Key
          in: header
          required: true
          schema:
            $ref: '#/components/schemas/TransactionId'

      requestBody:
        description: Details about application and Availability Zones where application instance should be created. It
also definea call back URI which the Partner OP shall use update Home OP about a change in instance status.
        content:
```

```
        application/json:
          schema:
            type: object
            required:
              - appId
              - appProviderId
              - appVersion
              - zoneInfo
              - appInstCallbackLink
            properties:
              appId:
                $ref: '#/components/schemas/AppIdentifier'
              appVersion:
                type: string
                description: Version info of the application
              appProviderId:
                $ref: '#/components/schemas/AppProviderId'
              zoneInfo:
                type: object
                required:
                  - zoneId
                  - flavourId
                properties:
                  zoneId:
                    $ref: '#/components/schemas/ZoneIdentifier'
                  flavourId:
                    $ref: '#/components/schemas/FlavourId'
                  resourceConsumption:
                    type: string
                    enum:
                      - RESERVED_RES_SHALL
                      - RESERVED_RES_PREFER
                      - RESERVED_RES_AVOID
                      - RESERVED_RES_FORBID
                    default: RESERVED_RES_AVOID
                    description: Specifies if the application can be instantiated using pre-reserved resource or not.
Application Provider can pre-reserve a pool of Edge Resources on each Availability Zone.
'RESERVED_RES_SHALL' instruct OP to use only the pre-reserved resources. 'RESERVED_RES_PREFER'
instruct to first try using pre-reserved resource, if none available go for non-reserved resources.
'RESERVED_RES_AVOID' instruct OP not to use pre-reserved resource if possible, it is a choice depending
upon circumstances 'RESERVED_RES_FORBID' instruct OP not to use pre-reserved resources.
                  resPool:
                    type: string
                    pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                    description: Resource pool to be used for application instantiation on this Availability Zone.  Valid
only if IE 'resourceConsumption' is set to 'RESERVED_RES_SHALL' or 'RESERVED_RES_PREFER'
              appInstCallbackLink:
                $ref: '#/components/schemas/Uri'
      responses:
        "202":
          description: Application instance creation request accepted.
          content:
            application/json:
              schema:
                type: object
                required:
                  - zoneId
                  - appInstIdentifier
```

```
          properties:
            zoneId:
              $ref: '#/components/schemas/ZoneIdentifier'
            appInstIdentifier:
              $ref: '#/components/schemas/InstanceIdentifier'
    "400":
      $ref: '#/components/responses/400'
    "401":
      $ref: '#/components/responses/401'
    "404":
      $ref: '#/components/responses/404NotFound'
    "409":
      $ref: '#/components/responses/409'
    "422":
      $ref: '#/components/responses/422'
    "500":
      $ref: '#/components/responses/500'
    "503":
      $ref: '#/components/responses/503'
    "520":
      $ref: '#/components/responses/520'
    default:
      $ref: '#/components/responses/default'
  callbacks:
    onInstanceStatusEvent:
      '{$request.body#/appInstCallbackLink}':
        post:
          requestBody:
            description: Notification payload.
            content:
              application/json:
                schema:
                  type: object
                  required:
                    - federationContextId
                    - appId
                    - appInstanceId
                    - zoneId
                    - appInstanceInfo
                  properties:
                    federationContextId:
                      $ref: '#/components/schemas/FederationIdentifier'
                    appId:
                      $ref: '#/components/schemas/AppIdentifier'
                    appInstanceId:
                      $ref: '#/components/schemas/InstanceIdentifier'
                    zoneId:
                      $ref: '#/components/schemas/ZoneIdentifier'
                    appInstanceInfo:
                      type: object
                      properties:
                        appInstanceState:
                          type: string
                          enum:
                            - PENDING
                            - READY
                            - FAILED
                            - TERMINATING
```

```
                description: Running status of the application instance.
              message:
                type: string
                description: Event information or failure message.
              accesspointInfo:
                description: Information about the IP and Port exposed by the OP. Application clients shall use these access points to reach this application instance
                type: array
                items:
                  type: object
                  required:
                    - interfaceId
                    - accessPoints
                  properties:
                    interfaceId:
                      type: string
                      pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                      description: This is the interface Identifier that Application Provider defines when Application is onboarded.
                    accessPoints:
                      $ref: '#/components/schemas/ServiceEndpoint'
                minItems: 1
            minProperties: 1
          modificationDate:
            type: string
            format: date-time
            description: Date and time of the instance state modification by Partner OP.
      responses:
        "204":
          description: Application instance state notification acknowledged
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
      security:
        - notifClientCredentials: [fed-mgmt-notif]

  /{federationContextId}/application/lcm/app/{appId}/instance/{appInstanceId}/zone/{zoneId}:
    get:
      summary: Retrieves an application instance details from Partner OP.
      operationId: GetAppInstanceDetails
      tags:
        - ApplicationDeploymentManagement
      parameters:
```

```
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppIdentifier'
      - name: appInstanceId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/InstanceIdentifier'
      - name: zoneId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ZoneIdentifier'
    responses:
      "200":
        description: Application instance details
        content:
          application/json:
            schema:
              type: object
              properties:
                appInstanceState:
                  $ref: '#/components/schemas/InstanceState'
                accesspointInfo:
                  description: Information about the IP and Port exposed by the OP. Application clients shall use these
access points to reach this application instance
                  type: array
                  items:
                    type: object
                    required:
                      - interfaceId
                      - accessPoints
                    properties:
                      interfaceId:
                        type: string
                        pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                        description: This is the interface identifier that Application Provider defines when application is
onboarded.
                      accessPoints:
                        $ref: '#/components/schemas/ServiceEndpoint'
                  minItems: 1
              minProperties: 1
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
```

```
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  delete:
    summary: Terminate an application instance on a Partner OP Availability Zone.
    operationId: RemoveApp
    tags:
      - ApplicationDeploymentManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppIdentifier'
      - name: appInstanceId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/InstanceIdentifier'
      - name: zoneId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ZoneIdentifier'
    responses:
      "200":
        description: Application instance termination request accepted
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
/{federationContextId}/application/lcm/app/{appId}/appProvider/{appProviderId}:
  get:
```

```
    summary: Retrieves all application instance of Partner OP
    operationId: GetAllAppInstances
    tags:
      - ApplicationDeploymentManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppIdentifier'
      - name: appProviderId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppProviderId'
    responses:
      "200":
        description: Application Instance details
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                required:
                  - zoneId
                  - appInstanceInfo
                properties:
                  zoneId:
                    $ref: '#/components/schemas/ZoneIdentifier'
                  appInstanceInfo:
                    type: array
                    items:
                      type: object
                      required:
                        - appInstIdentifier
                        - appInstanceState
                      properties:
                        appInstIdentifier:
                          $ref: '#/components/schemas/InstanceIdentifier'
                        appInstanceState:
                          $ref: '#/components/schemas/InstanceState'
                    minItems: 1
              minItems: 1
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
```

```
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}:
    post:
      summary: Reserves resources (compute, network and storage)  on a Partner OP Availability Zone.   ISVs
registered with Home OP reserves resources on a Partner OP Availability Zone.
      operationId: CreateResourcePools
      tags:
        - AppProviderResourceManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: zoneId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/ZoneIdentifier'
        - name: appProviderId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/AppProviderId'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              required:
                - resRequest
                - resourceReservationCallbackLink
              properties:
                resRequest:
                  description: Compute flavours to be reserved and their time duration
                  type: object
                  required:
                    - poolName
                    - flavours
                    - reserveDuration
                  properties:
                    poolName:
                      $ref: '#/components/schemas/PoolName'
                    flavours:
                      type: array
                      items:
                        type: object
                        required:
                          - flavourId
                          - numFlavour
```

```
                properties:
                  flavourId:
                    $ref: '#/components/schemas/FlavourId'
                  numFlavour:
                    type: integer
                    format: int32
                    description: Total number of flavours to be reserved
                  minNumOfFlavours:
                    type: integer
                    format: int32
                    description: If specified, indicate the minimum numbers of flavours to be reserved up to
maximum as given in “count” member. If Partner OP cannot reserve the minimum number of flavours, then the
request shall fail.
                minItems: 1
              reserveDuration:
                $ref: '#/components/schemas/ResourceReservationDuration'
            resourceReservationCallbackLink:
              $ref: '#/components/schemas/Uri'
      responses:
        "200":
          description: ISV Resource reservation request accepted
          content:
            application/json:
              schema:
                type: object
                required:
                  - poolId
                  - poolName
                properties:
                  poolName:
                    $ref: '#/components/schemas/PoolName'

                  poolId:
                    $ref: '#/components/schemas/PoolId'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
      callbacks:
        onResourceStatusChangeEvent:
          '{$request.body#/resourceReservationCallbackLink}':
            post:
              requestBody:
                description: Notification payload.
```

```
      content:
        application/json:
          schema:
            type: object
            required:
              - federationContextId
              - zoneId
              - appProviderId
              - poolId
              - grantedFlavours
            properties:
              federationContextId:
                $ref: '#/components/schemas/FederationIdentifier'
              zoneId:
                $ref: '#/components/schemas/ZoneIdentifier'
              appProviderId:
                $ref: '#/components/schemas/AppProviderId'
              poolId:
                $ref: '#/components/schemas/PoolId'
              grantedFlavours:
                type: array
                items:
                  type: object
                  required:
                    - flavourId
                    - numFlavour
                  properties:
                    flavourId:
                      $ref: '#/components/schemas/FlavourId'
                    numFlavour:
                      type: integer
                      format: int32
                      description: Count of flavour
                minItems: 1
    responses:
      "204":
        description: Updated Resource reservation status updated
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    security:
      - notifClientCredentials: [fed-mgmt-notif]
```

```
  get:
    summary: Retrieves the resource pool reserved by an ISV
    operationId: ViewISVResPool
    tags:
      - AppProviderResourceManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: zoneId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ZoneIdentifier'
      - name: appProviderId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/AppProviderId'
    responses:
      "200":
        description: Reserved Resources Details
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                required:
                  - poolName
                  - reservedPoolId
                  - reservedFlavours
                properties:
                  poolName:
                    $ref: '#/components/schemas/PoolName'
                  reservedPoolId:
                    $ref: '#/components/schemas/PoolId'
                  reservedFlavours:
                    type: array
                    items:
                      type: object
                      required:
                        - flavourId
                        - count
                      properties:
                        flavourId:
                          $ref: '#/components/schemas/FlavourId'
                        count:
                          type: integer
                          format: int32
                          description: Total number of flavours reserved
                    minItems: 1
                  reserveDuration:
                    $ref: '#/components/schemas/ResourceReservationDuration'
                  reservationTime:
                    type: string
```

```
            format: date-time
            description: Date and time when resources were reserved in UTC format
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  /{federationContextId}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}/pool/{poolId}:
    patch:
      summary: Updates resources reserved for a pool by an ISV
      operationId: UpdateISVResPool
      tags:
        - AppProviderResourceManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: zoneId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/ZoneIdentifier'
        - name: appProviderId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/AppProviderId'
        - name: poolId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/PoolId'
      requestBody:
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                required:
                  - updateType
                  - flavourId
                  - count
```

```
            properties:
              updateType:
                type: string
                enum:
                  - ADD
                  - REMOVE
                  - DURATION
                description: Specify if resource corresponding this flavour needs to added or removed. Field 'count'
gives the final total no of such flavours that should be reserved.  count 0 means remove all the resources.
              flavourId:
                $ref: '#/components/schemas/FlavourId'
              count:
                type: integer
                format: int32
                description: Total number of flavours to be reserved
              reserveDuration:
                $ref: '#/components/schemas/ResourceReservationDuration'
    responses:
      "200":
        description: Resource pool updated
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  delete:
    summary: Deletes the resource pool reserved by an ISV
    operationId: RemoveISVResPool
    tags:
      - AppProviderResourceManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: zoneId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/ZoneIdentifier'
      - name: appProviderId
        in: path
        required: true
        schema:
```

```
          $ref: '#/components/schemas/AppProviderId'
        - name: poolId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/PoolId'
      responses:
        "200":
          description: Resource pool deleted
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
  /{federationContextId}/edgenodesharing/edgeDiscovery:
    post:
      summary: Edge discovery procedures towards Partner OP over E/WBI. Originating OP request Partner OP to provide a list of candidate Availability Zones where an application instance can be created. Partner OP applies a set of filtering criteria's to select candidate Availability Zones.
      operationId: GetCandidateZones
      tags:
        - EdgeNodeSharing
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              required:
                - appProviderId
                - appId
              properties:
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                appId:
                  $ref: '#/components/schemas/AppIdentifier'
                edgeDiscoveryFilters:
                  type: object
                  minProperties: 1
                  properties:
```

```
              location:
                $ref: '#/components/schemas/ClientLocation'
    responses:
      "200":
        description: List of candidate Availability Zones
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/nodeDiscoveryResponse'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/apiservice/{serviceAPINameVal}:
    post:
      summary: Service API request forwarding to the Partner OP
      operationId: APIForwarding
      tags:
        - ServiceAPIManagement
      parameters:
        -   name: federationContextId
            in: path
            required: true
            schema:
               $ref: '#/components/schemas/FederationContextId'

        -   name: serviceAPINameVal
            in: path
            required: true
            schema:
               $ref: '#/components/schemas/serviceAPINameVal'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              required:
                - apiServiceId
                - customerID
                - customerInfo
                - txnIdentifier
                - ServiceAPIBody
              properties:
                customerID:
                  $ref: '#/components/schemas/customerID'
                txnIdentifier:
                  $ref: '#/components/schemas/txnIdentifier'
                ServiceAPIBody:
                  $ref: '#/components/schemas/serviceAPIContent'
```

```
          eventNotificationDest:
            $ref: '#/components/schemas/Uri'
    responses:
      '200':
        description: Service API request accepted
        headers:
          Location:
            description: Contains the URI of the newly created Service API Context resource.
            required: false
            schema:
              type: string
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/serviceAPIResponse'
      '400':
        $ref: '#/components/responses/400'
      '401':
        $ref: '#/components/responses/401'
      '404':
        $ref: '#/components/responses/404NotFound'
      '422':
        $ref: '#/components/responses/422'
      '500':
        $ref: '#/components/responses/500'
      default:
          $ref: '#/components/responses/default'
    callbacks:
      onServiceAPISessionEvent:
        '{$request.body#/eventNotificationDest}':
          post:
            parameters:
              - name: federationContextId
                in: path
                required: true
                schema:
                  $ref:  '#/components/schemas/FederationContextId'
              - name: apiServiceId
                in: path
                required: true
                schema:
                  $ref:  '#/components/schemas/serviceAPINames'
            requestBody:
              description: Notification about network event.
              content:
                application/json:
                  schema:
                    type: object
                    required:
                      - txnIdentifier
                      - serviceAPIEvent
                    properties:
                      serviceAPIEvent:
                        $ref:  '#/components/schemas/serviceAPINetworkEvent'
                      txnIdentifier:
                        $ref:  '#/components/schemas/txnIdentifier'
            responses:
              '200':
```

```
            description: Event info notification acknowledged
          '400':
            $ref: '#/components/responses/400'
          '401':
            $ref: '#/components/responses/401'
          '404':
            $ref: '#/components/responses/404'
          '409':
            $ref: '#/components/responses/409'
          '422':
            $ref: '#/components/responses/422'
          '500':
            $ref: '#/components/responses/500'
          '503':
            $ref: '#/components/responses/503'
          '520':
            $ref: '#/components/responses/520'
          default:
            $ref: '#/components/responses/default'
        security:
          - notifClientCredentials: [fed-mgmt-notif]

/{federationContextId}/apiservice/connid/{connectID}/custid/{customerID}:
  delete:
    summary: Remove the Service API Session earlier created with Service API forwarding request.
    operationId: RemoveServiceAPISession
    tags:
      - ServiceAPIManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref:  '#/components/schemas/FederationContextId'
      - name: connectID
        in: path
        required: true
        schema:
          $ref:  '#/components/schemas/connectID'
      - name: customerID
        in: path
        required: true
        schema:
          $ref:  '#/components/schemas/customerID'

    responses:
      '200':
        description: Service API Session removed successfully
        content:
          application/json:
            schema:
              type: object
              required:
                - expiryDuration
                - connectID
              properties:
                expiryDuration:
                  $ref: '#/components/schemas/expiryInterval'
```

```
              connectID:
                $ref: '#/components/schemas/connectID'
      '400':
        $ref: '#/components/responses/400'
      '401':
        $ref: '#/components/responses/401'
      '404':
        $ref: '#/components/responses/404NotFound'
      '409':
        $ref: '#/components/responses/409'
      '422':
        $ref: '#/components/responses/422'
      '500':
        $ref: '#/components/responses/500'
      '503':
        $ref: '#/components/responses/503'
      '520':
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
  get:
    summary: Retrieve the Service API context information of an existing API session identified by connectID,
customerID
    operationId: GetServiceAPISessionInfo
    tags:
      - ServiceAPIManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: connectID
        in: path
        required: true
        schema:
          $ref:  '#/components/schemas/connectID'
      - name: customerID
        in: path
        required: true
        schema:
          $ref:  '#/components/schemas/customerID'

    responses:
      "200":
        description: Device Auth Token validated
        content:
          application/json:
            schema:
              type: object
              required:
                - expiryDuration
                - connectID
              properties:
                expiryDuration:
                  $ref: '#/components/schemas/expiryInterval'
                connectID:
                  $ref: '#/components/schemas/connectID'
```

```
            ServiceAPIRespBody:
              $ref: '#/components/schemas/serviceAPIContent'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/monioring-subscriptions:
    post:
      summary: Originating OP subscribe for Edge Resource monitoring info with Partner OP.
      operationId: SubscribeMonitoringInfo
      tags:
        - ConsumptionReportingManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: monType
          in: query
          required: true
          schema:
            $ref: '#/components/schemas/monitoringSubsType'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                periodicity:
                  $ref: '#/components/schemas/periodicityInterval'
                resMonNotificationListner:
                  $ref: '#/components/schemas/Uri'

      responses:
        "200":
          description: Subscription for resource monitoring created successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/resourceSubscriptionInfo'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
```

```
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onPeriodicMonitoringEvent:
        '{$request.body#/resMonNotificationListner}':
          post:
            requestBody:
              description: Periodic Notification about resource monitoring info.
              content:
                application/json:
                  schema:
                    anyOf:
                      - $ref: '#/components/schemas/edgeResUtilizeMetrics'
                      - $ref: '#/components/schemas/appsResUtilizeInfo'
            responses:
              "200":
                description: Resource monitoring info notification acknowledged
              "400":
                $ref: '#/components/responses/400'
              "401":
                $ref: '#/components/responses/401'
              "404":
                $ref: '#/components/responses/404'
              "409":
                $ref: '#/components/responses/409'
              "422":
                $ref: '#/components/responses/422'
              "500":
                $ref: '#/components/responses/500'
              "503":
                $ref: '#/components/responses/503'
              "520":
                $ref: '#/components/responses/520'
              default:
                $ref: '#/components/responses/default'
            security:
              - notifClientCredentials: [fed-mgmt-notif]

  /{federationContextId}/events:
    post:
      summary: Originating OP uses this procedure to request enabling event reporting with Partner OP.
      operationId: CreateEventSubscription
      tags:
        - EventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
```

```
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              eventSubscriptionConfig:
                $ref: '#/components/schemas/EventSubscription'
    responses:
      "200":
        description: Subscription for reporting of events created successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/EventSubscriptionInfo'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onEventCriterionDetectionEvent:
        '{$request.body#/eventListner}':
          post:
            requestBody:
              description: Notification about event being detected as per event criterion.
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/EventsList'
            responses:
              "200":
                description: Event report acknowledged
              "400":
                $ref: '#/components/responses/400'
              "401":
                $ref: '#/components/responses/401'
              "404":
                $ref: '#/components/responses/404'
              "409":
                $ref: '#/components/responses/409'
              "422":
                $ref: '#/components/responses/422'
              "500":
                $ref: '#/components/responses/500'
```

```
            "503":
              $ref: '#/components/responses/503'
            "520":
              $ref: '#/components/responses/520'
            default:
              $ref: '#/components/responses/default'
          security:
            - notifClientCredentials: [fed-mgmt-notif]

/{federationContextId}/events/{event_subs_id}:
  post:
    summary: Originating OP uses this procedure to create an event criterion at Partner OP.
    operationId: CreateEventCriterion
    tags:
      - EventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: event_subs_id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              eventCriterion:
                $ref: '#/components/schemas/eventCriterion'
    responses:
      "200":
        description: Subscription for resource monitoring created successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/eventInfo'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
```

```
        $ref: '#/components/responses/default'

    get:
      summary: Retrieves events list with the Partner OP.
      operationId: GetEventsList
      tags:
        - EventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: event_subs_id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
        - name: event_type
          in: query
          required: false
          schema:
            type: string
            enum:
              - event_criterion
              - event_id
      responses:
        "200":
          description: Events criterion and detected events report request accepted
          content:
            application/json:
              schema:
                type: object
                properties:
                  eventCriterionList:
                    $ref: '#/components/schemas/eventTypeList'
                  eventIdList:
                    $ref: '#/components/schemas/EventsList'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'

    delete:
```

```
    summary: Remove existing event subscription with the Partner OP
    operationId: DeleteEventSubscription
    tags:
      - EventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: event_subs_id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'

    responses:
      "200":
        description: Event subscription removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/events/{event_subs_id}/event-id/{eventId}:
    delete:
      summary: Remove existing event criterion with the Partner OP
      operationId: DeleteEventCriterion
      tags:
        - EventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: event_subs_id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
        - name: eventId
          in: path
          required: true
```

```
        schema:
          $ref: '#/components/schemas/EventIdentifier'
      responses:
        "200":
          description: Event criterion removed successfully
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'

  /{federationContextId}/alarms:
    post:
      summary: Originating OP uses this procedure to request enabling alarm reporting with Partner OP.
      operationId: CreateAlarmReportingSubscription
      tags:
        - AlarmManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                alarmListnerCallback:
                  $ref: '#/components/schemas/Uri'

      responses:
        "200":
          description: Subscription for alarm reporting created successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/SubscriptionIdentifier'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
```

```
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onAlarmStateReportEvent:
        '{$request.body#/alarmListnerCallback}':
          post:
            requestBody:
              description: Notification about alarm management events at Partner OP.
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/AlarmObjectInfo'
            responses:
              "200":
                description: Event report acknowledged
              "400":
                $ref: '#/components/responses/400'
              "401":
                $ref: '#/components/responses/401'
              "404":
                $ref: '#/components/responses/404'
              "409":
                $ref: '#/components/responses/409'
              "422":
                $ref: '#/components/responses/422'
              "500":
                $ref: '#/components/responses/500'
              "503":
                $ref: '#/components/responses/503'
              "520":
                $ref: '#/components/responses/520'
              default:
                $ref: '#/components/responses/default'
            security:
              - notifClientCredentials: [fed-mgmt-notif]

          delete:
            requestBody:
              description: Alarm clear notification for an earlier alarm by Partner OP.
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/AlarmObjectInfo'
            responses:
              "200":
                description: Alarm clear event acknowledged
              "400":
                $ref: '#/components/responses/400'
```

```
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    security:
      - notifClientCredentials: [fed-mgmt-notif]

  patch:
    requestBody:
      description: Notification about alarm management events at Partner OP.
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/UpdatedAlarmParameters'
    responses:
      "200":
        description: Alarm state update report acknowledged
        content:
          application/json:
            schema:
              type: object
              properties:
                updatedAlarmId:
                  $ref: '#/components/schemas/AlarmIdentifier'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    security:
      - notifClientCredentials: [fed-mgmt-notif]
```

```
  /{federationContextId}/events/{alarm_subs_id}:
    get:
      summary: Retrieves active alarms list with the Partner OP.
      operationId: GetAlarmsList
      tags:
        - AlarmManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: alarm_subs_id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/SubscriptionIdentifier'
        - name: alarm_type
          in: query
          required: false
          schema:
            $ref: '#/components/schemas/AlarmType'
      responses:
        "200":
          description: Active alarms report request accepted
          content:
            application/json:
              schema:
                type: object
                properties:
                  activeAlarmsList:
                    $ref: '#/components/schemas/ActiveAlarmsList'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'

    delete:
      summary: Remove existing alarm subscription with the Partner OP
      operationId: DeleteAlarmSubscription
      tags:
        - AlarmManagement
      parameters:
        - name: federationContextId
```

```
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: alarm_subs_id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/SubscriptionIdentifier'

    responses:
      "200":
        description: Alarm subscription removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  /{federationContextId}/network-caps-events:
    post:
      summary: Originating OP uses this procedure to request enabling network capabilities events reporting by the
Partner OP.
      operationId: CreateNetworkCapsEventSubscription
      tags:
        - NetworkCapsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              required:
                - networkCapsEventSubscriptionConfig
              properties:
                networkCapsEventSubscriptionConfig:
                  $ref: '#/components/schemas/periodicNotifConfig'
      responses:
        "200":
```

```
        description: Subscription for notification of network events created successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/periodicNotifConfig'
        headers:
          Location:
            description: 'Contains the URI of the newly created resource'
            required: true
            schema:
              type: string
          Accept-Encoding:
            description: Accept-Encoding, described in IETF RFC 7694
            schema:
              type: string
          Content-Encoding:
            description: Content-Encoding, described in IETF RFC 7231
            schema:
              type: string
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onNetwEventDetectionEvent:
        '{$request.body#/notificationListner}':
          post:
            requestBody:
              description: Notification about events being detected as per network capabilities are applied by Partner
OP.
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/NetworkCapAppInfoList'
            responses:
              "200":
                description: Network Events report acknowledged
              "400":
                $ref: '#/components/responses/400'
              "401":
                $ref: '#/components/responses/401'
              "404":
                $ref: '#/components/responses/404'
              "409":
```

```
            $ref: '#/components/responses/409'
          "422":
            $ref: '#/components/responses/422'
          "500":
            $ref: '#/components/responses/500'
          "503":
            $ref: '#/components/responses/503'
          "520":
            $ref: '#/components/responses/520'
          default:
            $ref: '#/components/responses/default'
        security:
          - notifClientCredentials: [fed-mgmt-notif]

  /{federationContextId}/network-events/{nw-event-subs-id}:
    post:
      summary: Originating OP uses this procedure to add an intent to Partner OP to report network capability
applied by Partner OP.
      operationId: CreateNetworkCapEvent
      tags:
        - NetworkCapsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: nw-event-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
        - name: nw-cap-id
          in: query
          required: true
          schema:
            $ref: '#/components/schemas/CapabilityID'

      requestBody:
        content:
          application/json:
            schema:
              type: object
              required:
                - appId
                - appProviderId
              properties:
                appId:
                  $ref: '#/components/schemas/AppIdentifier'
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                txnIdentifier:
                  $ref: '#/components/schemas/txnIdentifier'

      responses:
        "200":
          description: Subscription for network event created successfully
          content:
```

```
        application/json:
          schema:
            type: object
            required:
              - networkCapSubsInfo
              - txnIdentifier
            properties:
              networkCapSubsInfo:
                $ref: '#/components/schemas/NetworkCapSubsInfo'
              txnIdentifier:
                $ref: '#/components/schemas/txnIdentifier'

    "400":
      $ref: '#/components/responses/400'
    "401":
      $ref: '#/components/responses/401'
    "404":
      $ref: '#/components/responses/404NotFound'
    "409":
      $ref: '#/components/responses/409'
    "422":
      $ref: '#/components/responses/422'
    "500":
      $ref: '#/components/responses/500'
    "503":
      $ref: '#/components/responses/503'
    "520":
      $ref: '#/components/responses/520'
    default:
      $ref: '#/components/responses/default'

  delete:
    summary: Remove existing network events notification subscription with the Partner OP
    operationId: DeleteNwEventNotifSubscription
    tags:
      - NetworkCapsEventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: nw-event-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    responses:
      "200":
        description: Network Event Notification subscription removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
```

```
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/network-events/{nw-event-subs-id}/nw-caps:
    get:
      summary: Retrieves network capabilities subscribed list with the Partner OP.
      operationId: GetNetworkCapsSubscribedList
      tags:
        - NetworkCapsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: nw-event-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
        - name: nw-event-type
          in: query
          required: true
          schema:
            type: string
      responses:
        "200":
          description: Network capabilities subscription list returned successfully
          content:
            application/json:
              schema:
                type: object
                properties:
                  subscribedNwCaps:
                    type: array
                    items:
                      $ref: '#/components/schemas/NetworkCapSubsInfo'
                    minItems: 1
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
```

```
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  delete:
    summary: Remove existing network event notification with the Partner OP
    operationId: DeleteNetworkCapSubscription
    tags:
      - NetworkCapsEventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: nw-event-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
      - name: nw-event-id
        in: query
        required: true
        schema:
          type: string
    responses:
      "200":
        description: Network Event subscription removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  /{federationContextId}/appl-event-notifications:
    post:
      summary: Originating OP uses this procedure to Subscribe for Application's Events Notifications.
      operationId: CreateApplicationEventSubscription
      tags:
        - ApplicationsEventManagement
      parameters:
```

```
    - name: federationContextId
      in: path
      required: true
      schema:
        $ref: '#/components/schemas/FederationContextId'
  requestBody:
    content:
      application/json:
        schema:
          type: object
          required:
            - applicationEventSubscriptionConfig
          properties:
            applEventSubscriptionConfig:
              $ref: '#/components/schemas/periodicNotifConfig'
            numEvtsPerNotif:
              type: integer
              description: The number of applications events that the Partner OP should include in a notification
  responses:
    "200":
      description: Subscription for notification of network events created successfully
      content:
        application/json:
          schema:
            type: object
            properties:
              appEventSubsId:
                type: string
              applEventSubscriptionConfig:
                $ref: '#/components/schemas/periodicNotifConfig'
              numEvtsPerNotif:
                type: integer
                description: The number of applications events that the Partner OP should include in a notification
      headers:
        Location:
          description: 'Contains the URI of the newly created resource'
          required: false
          schema:
            type: string
        Accept-Encoding:
          description: Accept-Encoding, described in IETF RFC 7694
          schema:
            type: string
        Content-Encoding:
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
    "400":
      $ref: '#/components/responses/400'
    "401":
      $ref: '#/components/responses/401'
    "404":
      $ref: '#/components/responses/404NotFound'
    "409":
      $ref: '#/components/responses/409'
    "422":
      $ref: '#/components/responses/422'
    "500":
```

```
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'
    callbacks:
      onApplEventDetectionEvent:
        '{$request.body#/notificationListner}':
          post:
            requestBody:
              description: Notification about applications LCM events being detected by Partner OP.
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/AggrApplEventsList'
            responses:
              "200":
                description: Applications events notification acknowledged
              "400":
                $ref: '#/components/responses/400'
              "401":
                $ref: '#/components/responses/401'
              "404":
                $ref: '#/components/responses/404'
              "409":
                $ref: '#/components/responses/409'
              "422":
                $ref: '#/components/responses/422'
              "500":
                $ref: '#/components/responses/500'
              "503":
                $ref: '#/components/responses/503'
              "520":
                $ref: '#/components/responses/520'
              default:
                $ref: '#/components/responses/default'
            security:
              - notifClientCredentials: [fed-mgmt-notif]


  /{federationContextId}/appl-event-notifications/{app-notif-subs-id}:
    post:
      summary: Originating OP uses this procedure to add applications for reporting of application events by
Partner OP.
      operationId: SubscribeApplsEvtNotif
      tags:
        - ApplicationsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: app-notif-subs-id
          in: path
          required: true
```

```
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
      - name: Idempotency-Key
        in: header
        required: true
        schema:
          $ref: '#/components/schemas/TransactionId'

    requestBody:
      content:
        application/json:
          schema:
            type: object
            required:
              - addAppsForNotif
            properties:
              addAppsForNotif:
                $ref: '#/components/schemas/AddAppsForNotif'

    responses:
      "200":
        description: Subscription for network event created successfully
        content:
          application/json:
            schema:
              type: object
              properties:
                addAppsForNotif:
                  $ref: '#/components/schemas/AddAppsForNotif'

      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  delete:
    summary: Remove existing application notification subscription with the Partner OP
    operationId: DeleteApplNotifSubscription
    tags:
      - ApplicationsEventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
```

```
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: app-notif-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    responses:
      "200":
        description: Application Event Notifications subscription removed successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  patch:
    summary: Modify existing application events notification subscription with the Partner OP
    operationId: ModifyApplEventNotifSubscription
    tags:
      - ApplicationsEventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: app-notif-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              applEventSubscriptionConfig:
                $ref: '#/components/schemas/periodicNotifConfig'
              numEvtsPerNotif:
                type: integer
                description: The number of applications events that the Partner OP should include in a notification
    responses:
      "200":
```

```
        description: Event Notification subscription modified successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  get:
    summary: Originating OP uses this procedure to retrieve subscription meta-information about application-
level notifications.
    operationId: RetrieveApplSubsMetaInfo
    tags:
      - ApplicationsEventManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: app-notif-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
      - name: info-type
        in: query
        required: true
        schema:
          type: string
          enum:
            - subs-info
            - apps-info

    responses:
      "200":
        description: Application events Subscription information successful retrieval
        content:
          application/json:
            schema:
              oneOf:
                - $ref: '#/components/schemas/ApplEventsSubsInfo'
                - $ref: '#/components/schemas/ApplEventsSubsInfo'
      "400":
        $ref: '#/components/responses/400'
      "401":
```

```
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/appl-event-notifications/{app-notif-subs-id}/cancel:
    post:
      summary: Remove applications from the reporting of application-level event notifications.
      operationId: RemoveAppsEventSubscription
      tags:
        - ApplicationsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: app-notif-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/RemoveAppsForNotif'
      responses:
        "200":
          description: Network capabilities subscription list returned successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/RemoveAppsForNotif'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
```

```
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/appl-event-notifications/{app-notif-subs-id}/app-events:
    post:
      summary: Remove applications from the reporting of application-level event notifications.
      operationId: RetrieveAppsEventsInfo
      tags:
        - ApplicationsEventManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: app-notif-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
      requestBody:
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/AppsForNotif'
              minItems: 1
      responses:
        "200":
          description: Network capabilities subscription list returned successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AggrApplEventsList'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'
```

```
  /{federationContextId}/app-policies-subscription:
    post:
      summary: Originating OP uses this procedure to Subscribe for Application's policy capability at Partner OP.
      operationId: CreateApplicationPolicySubscription
      tags:
        - ApplicationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      responses:
        "200":
          description: Subscription for application policy management created successfully
          content:
            application/json:
              schema:
                type: object
                properties:
                  applPolicySubscriptionId:
                    type: string
          headers:
            Location:
              description: 'Contains the URI of the newly created resource'
              required: false
              schema:
                type: string
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
          $ref: '#/components/responses/500'
        "503":
          $ref: '#/components/responses/503'
        "520":
          $ref: '#/components/responses/520'
        default:
          $ref: '#/components/responses/default'

  /{federationContextId}/appl-policies-subscription/{appl-policy-subs-id}/app-policy-templates:
    get:
      summary: Originating OP uses this procedure to retrieve application policy templates from Partner OP.
```

```
    operationId: RetrieveAppPolicyTemplates
    tags:
      - ApplicationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appl-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
      - name: appl-policy-type
        in: query
        required: false
        schema:
          $ref: '#/components/schemas/ApplPolicyType'

    responses:
      "200":
        description: Successfully retrieved application policy templates
        content:
          application/json:
            schema:
              type: object
              properties:
                applPolicyTemplateList:
                  $ref: '#/components/schemas/ApplPolicyTemplateList'

      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/appl-policies-subscription/{appl-policy-subs-id}/app-policy-registration:
    post:
      summary: Register an application-level policy with the Partner OP
      operationId: RegisterApplicationPolicy
      tags:
        - ApplicationPolicyManagement
      parameters:
        - name: federationContextId
```

```
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appl-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              applConcretePolicy:
                $ref: '#/components/schemas/ApplConcretePolicy'
    responses:
      "200":
        description: Application policy registered successfully
        content:
          application/json:
            schema:
              type: object
              required:
                - pplConcretePolicy
                - policyId
              properties:
                pplConcretePolicy:
                  $ref: '#/components/schemas/ApplConcretePolicy'
                policyId:
                  type: string

      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  /{federationContextId}/app-policies-subscription/{appl-policy-subs-id}:
    post:
      summary: Origination OP uses this procedure to apply application-level policies to federated applications at
Partner OP.
      operationId: ApplyApplicationPolicy
```

```
    tags:
      - ApplicationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appl-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/AssocApplPolicies'
    responses:
      "200":
        description: Application Policy processed successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AssocApplPolicies'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  get:
    summary: Origination OP uses this procedure to retrieve application-level policies to federated applications at
Partner OP.
    operationId: RetrieveApplicationPolicy
    tags:
      - ApplicationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: appl-policy-subs-id
```

```
      in: path
      required: true
      schema:
        $ref: '#/components/schemas/EventSubscriptionIdentifier'
    - name: policy-search-type
      in: query
      required: false
      schema:
        type: string
        enum:
          - app-prov-id
          - app-id
    - name: policy-search-value
      in: query
      required: false
      schema:
        type: string
      description: Refers to either Application Provider identifier or the Application Identifier

  responses:
    "200":
      description: Application Policy list retrieved successfully
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/AssocApplPolicies'
    "400":
      $ref: '#/components/responses/400'
    "401":
      $ref: '#/components/responses/401'
    "404":
      $ref: '#/components/responses/404NotFound'
    "409":
      $ref: '#/components/responses/409'
    "422":
      $ref: '#/components/responses/422'
    "500":
      $ref: '#/components/responses/500'
    "503":
      $ref: '#/components/responses/503'
    "520":
      $ref: '#/components/responses/520'
    default:
      $ref: '#/components/responses/default'



  patch:
    summary: Modify application-level policy associated with federated applications with the Partner OP
    operationId: ModifyApplicationPolicy
    tags:
      - ApplicationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
```

```
      - name: appl-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/AssocApplPolicies'
    responses:
      "200":
        description: Application policies modified successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/app-policies-subscription/{appl-policy-subs-id}/app-policy-cancel:
    post:
      summary: Remove applications from federated applications at Partner OP.
      operationId: RemoveApplicationPolicies
      tags:
        - ApplicationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: appl-policy-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AssocApplPolicies'
      responses:
        "200":
          description: Successfully removed application policies
          content:
```

```
        application/json:
          schema:
            $ref: '#/components/schemas/AssocApplPolicies'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/ops-policies-subscription:
    post:
      summary: Originating OP uses this procedure to Subscribe for Operation's policy capability at Partner OP.
      operationId: CreateOperationPolicySubscription
      tags:
        - OperationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
      responses:
        "200":
          description: Subscription for operation's policy management created successfully
          content:
            application/json:
              schema:
                type: object
                properties:
                  opslPolicySubscriptionId:
                    type: string
          headers:
            Location:
              description: 'Contains the URI of the newly created resource'
              required: false
              schema:
                type: string
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
```

```
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/ops-policy-templates:
    get:
      summary: Originating OP uses this procedure to retrieve operations policy templates from Partner OP.
      operationId: RetrieveOpsPolicyTemplates
      tags:
        - OperationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: ops-policy-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
        - name: ops-policy-type
          in: query
          required: false
          schema:
            $ref: '#/components/schemas/OpsPolicyType'

      responses:
        "200":
          description: Successfully retrieved operations policy templates
          content:
            application/json:
              schema:
                type: object
                properties:
                  opsPolicyTemplateList:
                    $ref: '#/components/schemas/OpsPolicyTemplateList'

        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
```

```
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/ops-policy-registration:
    post:
      summary: Register an operation-level policy with the Partner OP
      operationId: RegisterOperationPolicy
      tags:
        - OperationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: ops-policy-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                opsConcretePolicy:
                  $ref: '#/components/schemas/OpsConcretePolicy'
      responses:
        "200":
          description: Operations policy registered successfully
          content:
            application/json:
              schema:
                type: object
                required:
                  - opsConcretePolicy
                  - policyId
                properties:
                  opsConcretePolicy:
                    $ref: '#/components/schemas/OpsConcretePolicy'
                  policyId:
                    type: string

        "400":
          $ref: '#/components/responses/400'
        "401":
```

```
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  /{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/policy-association:
    post:
      summary: Origination OP uses this procedure to apply application-level policies to federated applications at
Partner OP.
      operationId: ApplyOperationPolicy
      tags:
        - OperationPolicyManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/FederationContextId'
        - name: ops-policy-subs-id
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/EventSubscriptionIdentifier'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AssocOpsPolicies'
      responses:
        "200":
          description: Operation Policy processed successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AssocOpsPolicies'
        "400":
          $ref: '#/components/responses/400'
        "401":
          $ref: '#/components/responses/401'
        "404":
          $ref: '#/components/responses/404NotFound'
        "409":
          $ref: '#/components/responses/409'
        "422":
          $ref: '#/components/responses/422'
        "500":
```

```
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  get:
    summary: Origination OP uses this procedure to retrieve application-level policies to federated applications at
Partner OP.
    operationId: RetrieveOperationPolicy
    tags:
      - OperationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: ops-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
      - name: policy-search-type
        in: query
        required: false
        schema:
          type: string
          enum:
            - zone-id
      - name: policy-search-value
        in: query
        required: false
        schema:
          type: string
        description: Refers to Availability Zone identifier

    responses:
      "200":
        description: Operations Policy list retrieved successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AssocOpsPolicies'
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
```

```
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'


  patch:
    summary: Modify operation-level policy associated with federated applications with the Partner OP
    operationId: ModifyOperationPolicy
    tags:
      - OperationPolicyManagement
    parameters:
      - name: federationContextId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/FederationContextId'
      - name: ops-policy-subs-id
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/EventSubscriptionIdentifier'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/AssocOpsPolicies'
    responses:
      "200":
        description: Application policies modified successfully
      "400":
        $ref: '#/components/responses/400'
      "401":
        $ref: '#/components/responses/401'
      "404":
        $ref: '#/components/responses/404NotFound'
      "409":
        $ref: '#/components/responses/409'
      "422":
        $ref: '#/components/responses/422'
      "500":
        $ref: '#/components/responses/500'
      "503":
        $ref: '#/components/responses/503'
      "520":
        $ref: '#/components/responses/520'
      default:
        $ref: '#/components/responses/default'

  /{federationContextId}/ops-policies-subscription/{ops-policy-subs-id}/ops-policy-cancel:
    post:
      summary: Remove applications from federated applications at Partner OP.
      operationId: RemoveOperationPolicies
      tags:
        - OperationPolicyManagement
      parameters:
```

```
    - name: federationContextId
      in: path
      required: true
      schema:
        $ref: '#/components/schemas/FederationContextId'
    - name: ops-policy-subs-id
      in: path
      required: true
      schema:
        $ref: '#/components/schemas/EventSubscriptionIdentifier'
  requestBody:
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/AssocOpsPolicies'
  responses:
    "200":
      description: Successfully removed operation policies
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/AssocOpsPolicies'
    "400":
      $ref: '#/components/responses/400'
    "401":
      $ref: '#/components/responses/401'
    "404":
      $ref: '#/components/responses/404NotFound'
    "409":
      $ref: '#/components/responses/409'
    "422":
      $ref: '#/components/responses/422'
    "500":
      $ref: '#/components/responses/500'
    "503":
      $ref: '#/components/responses/503'
    "520":
      $ref: '#/components/responses/520'
    default:
      $ref: '#/components/responses/default'
```


Annex B E/WBI APIs Implementation

B.1 Kubernetes operator pattern

This section provides an overview of an implementation choice of the East/West Bound Interface (E/WBI) APIs, based on the Kubernetes Operator pattern.

Kubernetes [17], primarily known for container orchestration, provides extensibility mechanisms that allow to define custom resources and controllers [18]. Using Kubernetes Custom Resource Definitions (CRDs), high-level intents can be defined for Federation services expressed as custom resources. The custom resources represent the declarative model of E/WBI APIs operations and parameters (such as the Create Federation method, and its parameters, e.g., initialDate, federationContextId, or offeredAvailabilityZones). These custom resources are managed by custom controllers, commonly referred to as operators,

which continuously reconcile the declared desired state with the actual system state, thereby automating Federation configuration and lifecycle management within a Kubernetes cluster.

This implementation choice realises the E/WBI APIs without introducing any change to their definitions or behaviour. As illustrated in Figure 56, within OP A the standard E/WBI APIs remain exposed as the external interface towards Partner OPs (e.g., OP B), and Federation between OPs continues to take place exclusively via the defined REST APIs. The inter-OP reference point is therefore unchanged, ensuring full compatibility with any OP, including those not adopting this implementation approach based on Kubernetes.

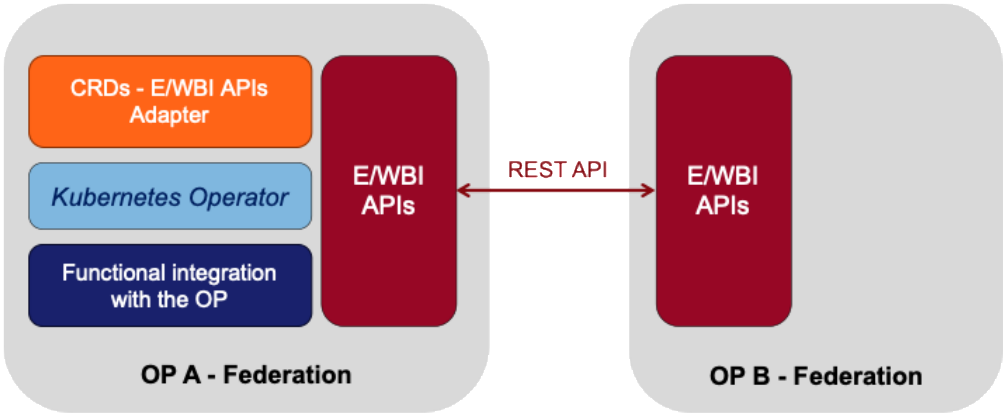


Figure 56: E/WBI APIs implementation choice representation.

Internally, OP A implements the E/WBI functionality through a Kubernetes-based Federation implementation composed of three main elements:

1. The *CRDs – E/WBI APIs Adapter* translates E/WBI REST API calls initiated by the Partner OP B towards the Originating OP A, following Figure 56, into Kubernetes Custom Resources that represent the declarative model of the Federation method. At the same time, the *E/WBI APIs Adapter* translates the Kubernetes Custom Resources into E/WBI REST API calls initiated by the Originating OP A towards the Partner OP B, following Figure 28.
2. The *Kubernetes Operator* acts as the Federation control logic. It continuously reconciles the desired state expressed in the OP Custom Resources (OP A) with the actual state of the OP's underlying resources and capabilities (OP A), and the actual state of the Partner OP (OP B). In doing so, it manages the lifecycle of Federation relationships, handles asynchronous workflows and Callbacks, performs retries, and interacts with the OP's underlying resources and capabilities.
3. The *Functional integration with the OP* connects the Kubernetes operator with the OP´s internal functionalities. This ensures that the OP can integrate the Kubernetes-based Federation implementation with its own infrastructure stack, regardless of its specific implementation technologies.

In this approach, Kubernetes is used exclusively as an internal implementation mechanism within the OP, and it is not exposed to Partner OPs. Since the E/WBI APIs remain unchanged and continue to be the inter-Operator interface, interoperability with non-Kubernetes-based OPs is preserved. At the same time, for OPs leveraging Kubernetes internally, this approach enables declarative Federation, encapsulating most of the E/WBI

API implementation logic and operational complexity into the reconciliation logic of the Federation Kubernetes operator.

This approach contributes to accelerating the adoption of Federation mechanisms across the ecosystem, encapsulating Federation expertise within reusable operator logic. A good example of this is the Linux Foundation NeoNephos Katalis project, providing a reference implementation of a subset of E/WBI APIs logic using Kubernetes operators, publicly available at the NeoNephos Katalis project GitHub repository [19]. The repository includes the CRDs that model Federation-related methods and its parameters in a declarative manner, available under the config/crd/bases directory [20]. These artefacts illustrate how E/WBI concepts can be represented as Kubernetes resources and reconciled by controller logic, providing a practical and reusable foundation for Operators adopting the Kubernetes-based Federation implementation choice.

Note: It is subject to study if the future evolution of the NeoNephos Katalis project will be adopted in the OP.

Annex C Document Management

C.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	03 Oct 2022	New PRD defining the East/Westbound Interface of the Operator Platform	ISAG	Deepak Gunjal / Capgemini
2.0	29 Mar 2023	Update implementing OPG.04 CR1002	ISAG	Deepak Gunjal / Capgemini
3.0	26 Jul 2023	Update implementing OPG.04 CR1003	ISAG	Deepak Gunjal / Capgemini
4.0	16 Feb 2024	Update implementing OPG.04 CR1004	ISAG	Deepak Gunjal / Capgemini
5.0	20 Sep 2024	Update implementing OPG.04 CR1005	ISAG	Deepak Gunjal / Capgemini
6.0	28 Feb 2025	Update implementing OPG.04 CR1006	ISAG	Deepak Gunjal / Capgemini
7.0	17 Jul 2026	Update implementing OPG.04 CR1007	ISAG	Deepak Gunjal / Capgemini

C.2 Other Information

Type	Description
Document Owner	Operator Platform Group
Editor / Company	Deepak Gunjal / Capgemini

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.