East-Westbound Interface APIs

Version 1.0

03 October 2022

This is a Non-binding Permanent Reference Document of the GSMA

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

Disclaimer

The GSM Association ("Association") 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 GSM Association's 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 in GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

1 Introduction

1.1 Overview

This document specifies RESTful Application Programming Interface (APIs) that allow an Operator Platform (OP) to share the edge cloud 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

Term	Description
API Initiator	API Initiator is the entity that originates the first message in the API sequences
Application Provider	The provider of the application that accesses the OP to deploy its application on the Edge Cloud, thereby using the Edge Cloud Resources and Network Resources as detailed in GSMA PRD OPG.02 [1]
Federation	Federation refers to relationship among member OPs who agrees to offer OP PRD defined services and capabilities to the application providers and end users of member OPs
Directed Federation	A Federation between two OP instances A and B, in which edge compute resources are shared by B to A, but not from A to B.
Federation Creation	Refers to the process for the establishment of the federation relationship between originating OP and partner OP on request by originating OP over the E/WBI
Discovery Service	OP service identified by a well-defined Fully Qualified Domain Name (FQDN) or IP:Port and protocol pair to assist User Clients (UCs) over User Network Interface (UNI) to discover adequate edge cloud in the current location of the end users
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.02
Home OP	The OP instance belonging to the subscriber's Operator; that is, whose PLMN identity Mobile Country Code ((MCC) and Mobile Network Code (MNC)) matches with the MCC and MNC of the subscriber's International Mobile Subscriber Identity (IMSI),
LCM Service	Lifecycle Management (LCM) Service to enable UCs for requesting dynamic application instantiation or termination
Leading OP	The Operator Platform instance as defined in GSMA PRD OPG.02 [1] connected to the Application Provider and receiving the onboarding requests, sharing them to the selected federated platforms/operators.

Term	Description
Originating OP	The OP instance initiating the federation creation request to selected federated platforms/operators. Both leading OP and Home OP will be acting as Originating OP while creating the federation with Partner OP.
OP Id	Operator id is a uniquely identifier assigned to each OP instance of the federation to identify the member OP
OP Administrator	Refers to person(s) responsible for the functions e.g., management, configuration, monitoring etc. of an OP instance
Mobility Strategy	It refers to defining an application mobility strategy that includes QoE, geographical store and privacy policies intent
Zone	Zone refers to an Availability Zone as defined in GSMA PRD OPG.02 [1]

1.4 Abbreviations

Term	Description
API	Application Programming Interface
CPU	Central Processing Unit
DNS	Domain Name System
DPDK	Data Plane Development Kit
E/WBI	East/West Bound Interface
FPGA	Field Programmable Gate Array
FQDN	Fully Qualified Domain Name
GPU	Graphical Processing Unit
HTTP	HyperText Transfer Protocol
IMSI	International Mobile Subscriber Identity
ISA	Instruction Set Architecture
ISV	Independent Software Vendor
KPI	Key Performance Indicator
LBO	Local Break Out (also defined in PRD as Local BreakOut)
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

Term	Description
REST	REpresentational State Transfer
SRIOV	Single Root Input Output Virtualisation
TLS	Transport Level Security
UC	User Client
UNI	User Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
SDK	Software Development Kit
vCPU	Virtual CPU
VM	Virtual Machine
VPU	Visual Processing Unit
YAML	YAML Ain't Markup Language

1.5 References

Ref	Doc Number	Title
[1]	OPG.02	Operator Platform Telco Edge Requirements", Version 2.0 14 April 2022
[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

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 Cloud 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 cloud resources and network capabilities to the requesting OP. Thus, if two OPs want to expose edge cloud resources and network capabilities with each other, then both the OPs would need to initiate a directional federation creation request towards each other.

2.1.3 Federation Identifier

A federation 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 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 applications 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 5), 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 Zones

The Partner OP may offer to expose one or more Availability Zone(s) and associated Edge Cloud resources to the Originating OP based on the prior agreement and local configuration. These zone(s) are defined as “Offered 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 Zones

Based on the offered 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 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 users in visited OP networks and in conjunction with MCC they can be used by home OPs to determine the roaming in partner OPs footprints.

2.1.10 Zone Meta-information

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

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 app 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).

The Home OP can also use the edge discovery service to redirect the edge discovery requests from roaming users on partner OP networks to be redirected to that partner OP’s edge discovery service based on the network identification.

2.1.12 Mobility Strategy

An Application Provider may be able to provide the mobility strategy (refer GSMA PRD OPG.02 [1]) 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 compute resources offered by the Partner OP to the Leading OP. The Leading OP may use such resources to serve its own end users in scenarios such as not having the edge clouds footprint in locations where the end users requesting access to edge services, but a Partner OP is offering edge cloud resources in those locations

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 5 and the details of obtaining credentials are outside the scope of this document.

2.2.1 Procedures for federation establishment between OP partners

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

Basic functionalities must cover:

- Create federation between OP partners

- Update an already establish federation between OP partners
- Remove a federation establishment between OP partners.

2.2.1.1 Create Federation between OP partners

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 API initiator say ‘OP-A’ requests partner OP-B permission to consume the OP-B resources and network capabilities on edge sites of 'OP-B'.

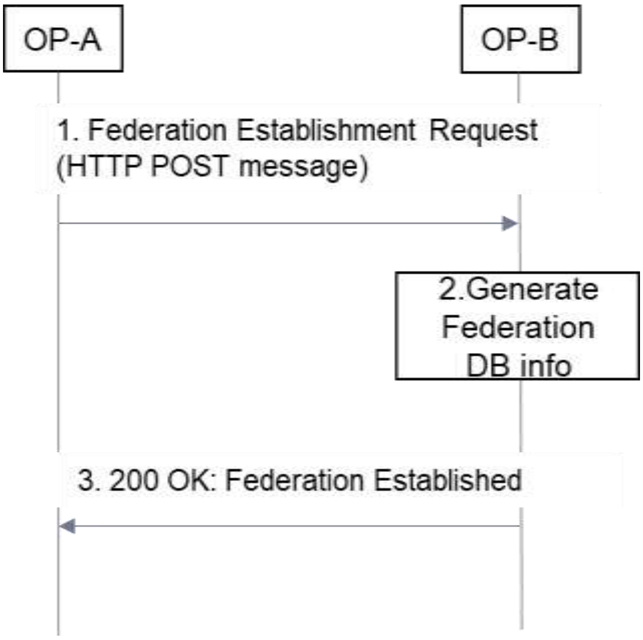
OP-A
OP-B
1. Federation Establishment Request (HTTP POST message)
2.Generate Federation DB info
3. 200 OK: Federation Established

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.
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 Partner OP edge discovery service FQDN, list of Availability Zone(s) meta-information (e.g., zone Id, geolocation details etc.) that the Partner OP can offer to the Originating 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.

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 edge discovery service FQDN shared by the Partner OP is for supporting roaming users when they visit a partner OP's network. In those cases, the Home OP on receiving the edge discovery requests from UCs, detect the roaming condition, and based on the current network code of the UE determines the Partner OP and corresponding edge discovery FQDN and redirects UCs to partner OP edge discovery service.

2.2.1.2 Update Federation between OP partners

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)).

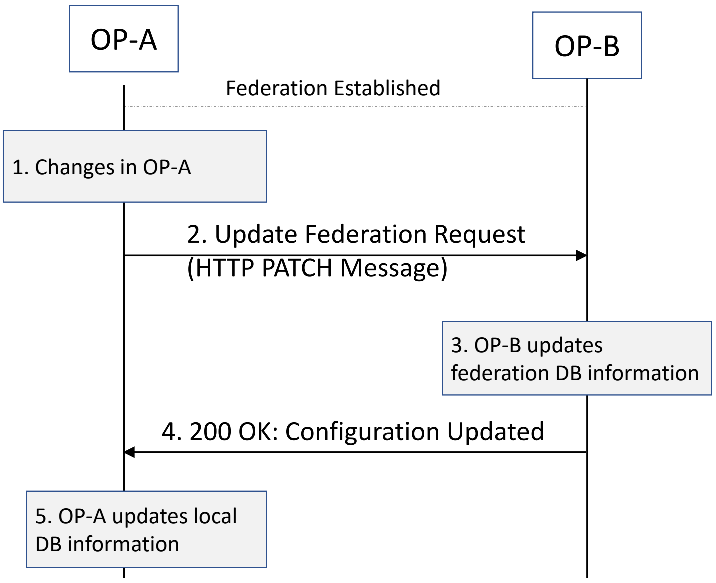

Figure 2: Update Federation

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.

2.2.1.3 Remove Federation configuration between OP partners

This procedure is intended to remove existing federation information within a Partner OP. By Remove Federation Operation, the API initiator OP 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).

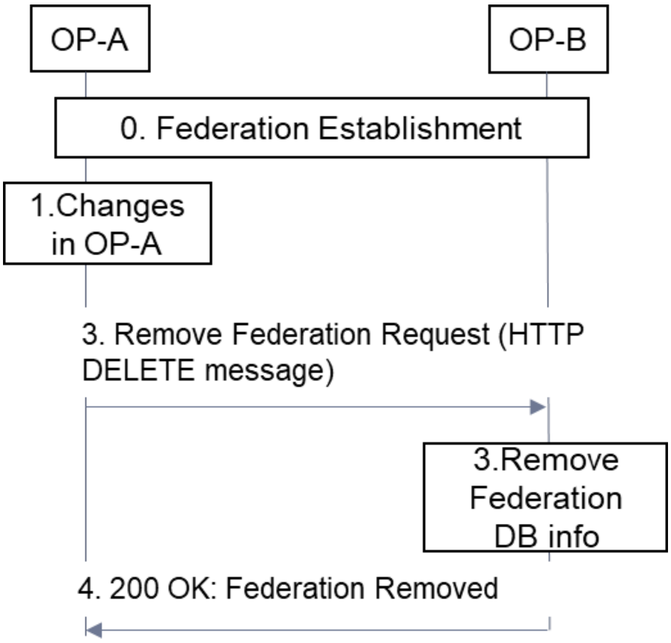
OP-A
OP-B
0. Federation Establishment
1.Changes in OP-A
3. Remove Federation Request (HTTP DELETE message)
3.Remove Federation DB info
4. 200 OK: Federation Removed

Figure 3: Remove Federation

The Partner OP can also terminate the existing federation with the Originating OP. The Partner OP say ‘OP-B’ sends an HTTP POST request to the Originating OP say ‘OP-A’ to terminate the existing directional 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.

2.2.1.4 Retrieve partner federated zone meta-information

The Originating OP may use this procedure towards federated partners OP to retrieve Availability Zone(s) meta-information e.g., zone identifier(s), 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.

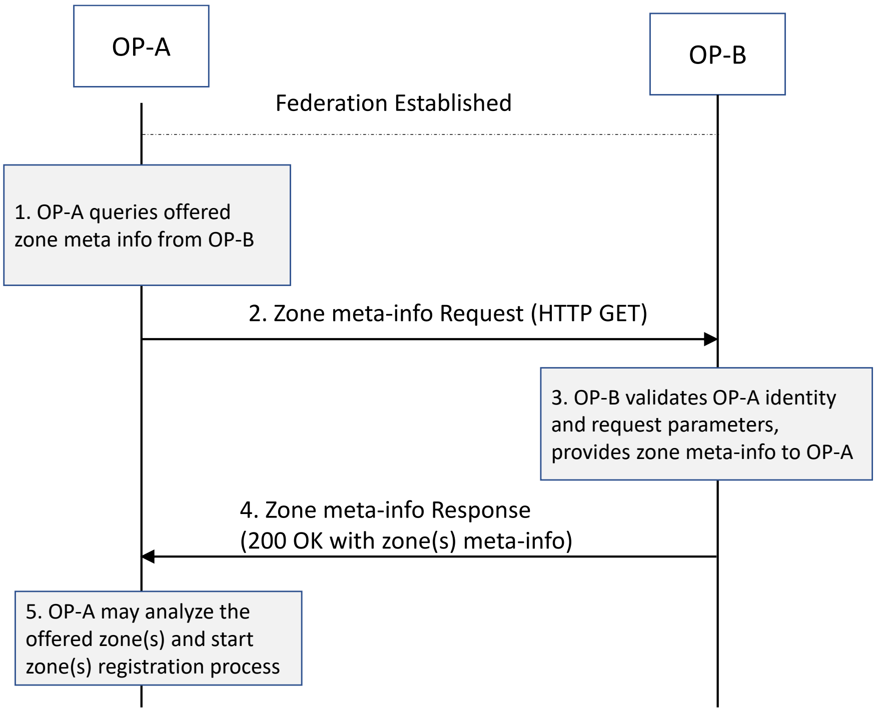
OP-A
OP-B
Federation Established
1. OP-A queries offered zone meta info from OP-B
2. Zone meta-info Request (HTTP GET)
3. OP-B validates OP-A identity and request parameters, provides zone meta-info to OP-A
4. Zone meta-info Response (200 OK with zone(s) meta-info)
5. OP-A may analyze the offered zone(s) and start zone(s) registration process

Figure 4: Retrieve partner federated 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 follow:

1. A partner federated 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 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.

2.2.2 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 zone information and updating the resource information, notifying partners if there are new 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 API initiator subscribe one or more zones from the list of zones being offered by the Partner OP to the Originating OP. The ISVs of the originating OP can deploy their applications on the zone(s) being subscribed by this API.

2.2.2.1 Zone Subscription

On receiving the Availability Zones meta information (e.g., 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 zone(s) and provides the details of resources configuration and quantity in the Availability Zones to the Originating OP in the response to zone subscription request.

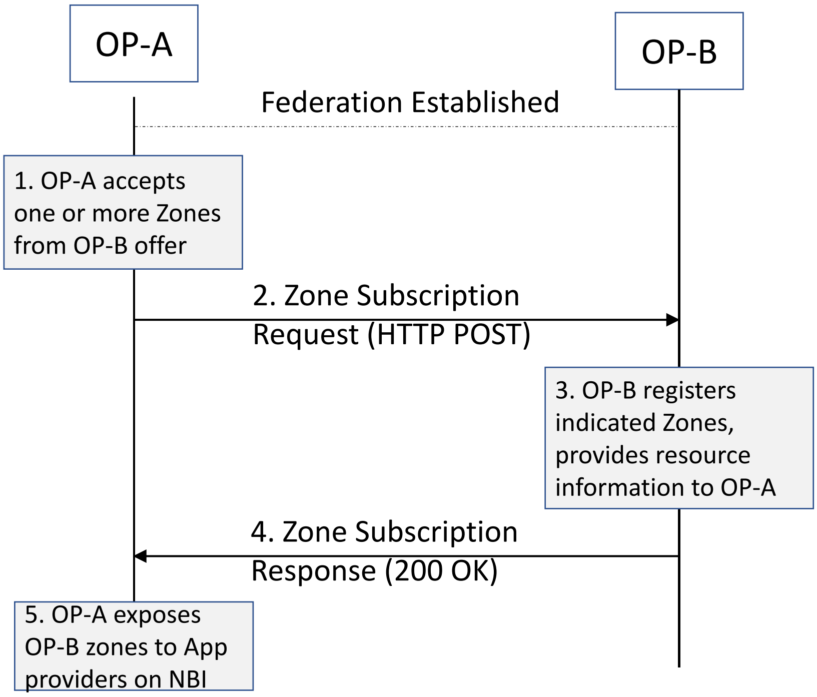
OP-A
OP-B
Federation Established
1. OP-A accepts one or more Zones from OP-B offer
2. Zone Subscription Request (HTTP POST)
3. OP-B registers indicated Zones, provides resource information to OP-A
4. Zone Subscription Response (200 OK)
5. OP-A exposes OP-B zones to App providers on NBI

Figure 5: Availability Zone subscription

2.2.2.2 View 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.

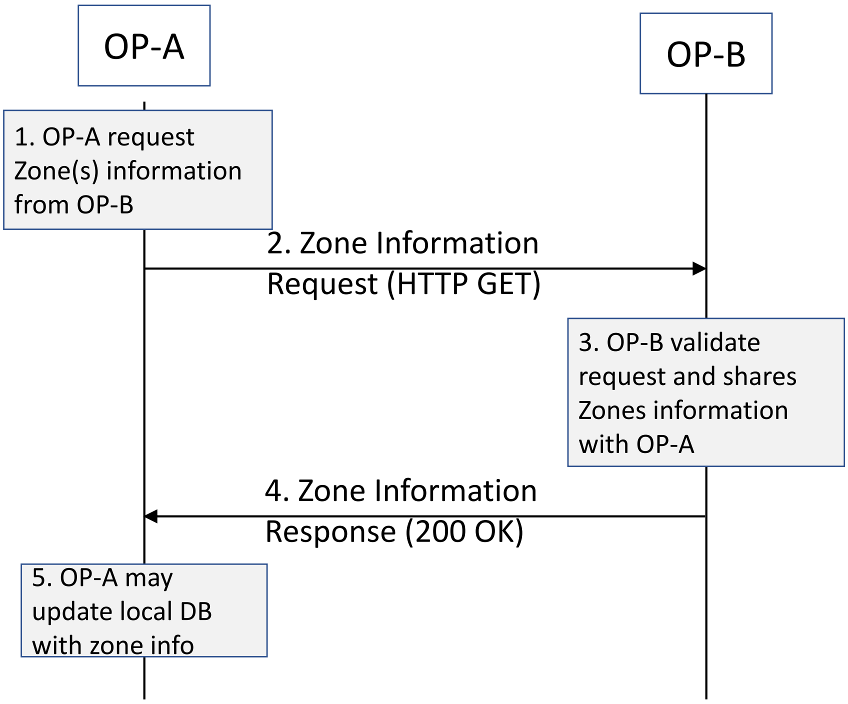
OP-A
OP-B
1. OP-A request Zone(s) information from OP-B
2. Zone Information Request (HTTP GET)
3. OP-B validate request and shares Zones information with OP-A
4. Zone Information Response (200 OK)
5. OP-A may update local DB with zone info

Figure 6: View Availability Zone information

2.2.3 Procedures for registration and authorization of end users in a federated OP partner

As defined in the GSMA PRD OPG.02 [1] when moving to a visited network, the end user shall first contact the home network OP platform. In case the visited network is a federated partner and that local break out is available the end user is redirected to the visited OP platform.

2.2.3.1 Authorization of end users by federated OP

The Visited OP platform needs to authenticate and authorize the service to the end users it can access edge nodes available in the visited network. This model is preferred because the edge cloud service is provided closer to the User Client.

As described in the GSMA PRD OPG.02 [1], the Home OP is involved managing the subscriber’s authentication and authorization. The following figure is intended to describe the interactions between OP partners to validate and authenticate end users.

1. The UE A while in OP-B network, registers to OP-A (Home Domain).

a) Authentication/Authorization procedures in the home network
b) OP-A retrieve UE location information

2. The OP-A steers the user to OP-B based on the user location and considering that both operators have agreed that Local Breakout (LBO) can be used. Information regarding UE access to OP-B must be included in the redirection message (e.g., IP address, FQDN
3. After receiving OP-B access information the UE-A proceed to register in the Visited OP
4. These steps represent the federation connection for enabling the application availability on Operator B by sharing and validating user's authorization information (HTTP GET).
5. In case of failure, the cause should be reflected in the response message, so that it can be notified to UE-A
6. Finally, UE-A gets authorized in OP-B and can request access to edge services provided based on the UE's location.
7. In case of failure the corresponding status message must be provided showing the cause.

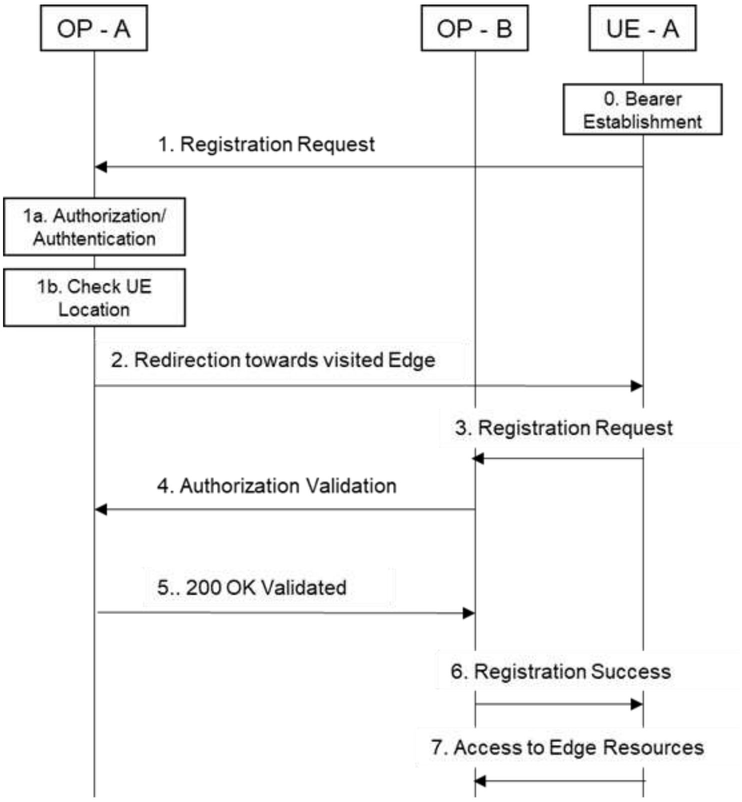

Figure 7: User Client registration on access from a visited OP

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 partners OP 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 an OP partner, considering that a federation has been established between partners 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 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.

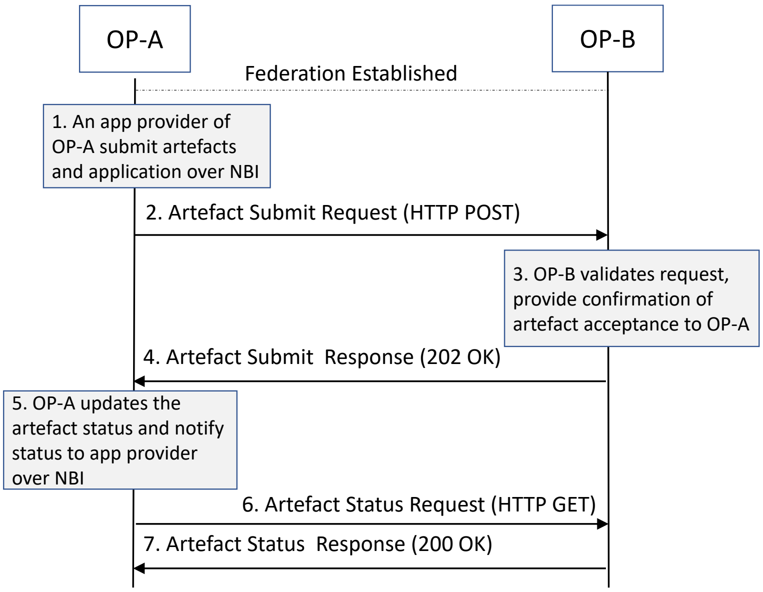
OP-A
OP-B
Federation Established
1. An app provider of OP-A submit artefacts and application over NBI
2. Artefact Submit Request (HTTP POST)
3. OP-B validates request, provide confirmation of artefact acceptance to OP-A
4. Artefact Submit Response (202 OK)
5. OP-A updates the artefact status and notify status to app provider over NBI
6. Artefact Status Request (HTTP GET)
7. Artefact Status Response (200 OK)

Figure 8: 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 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.

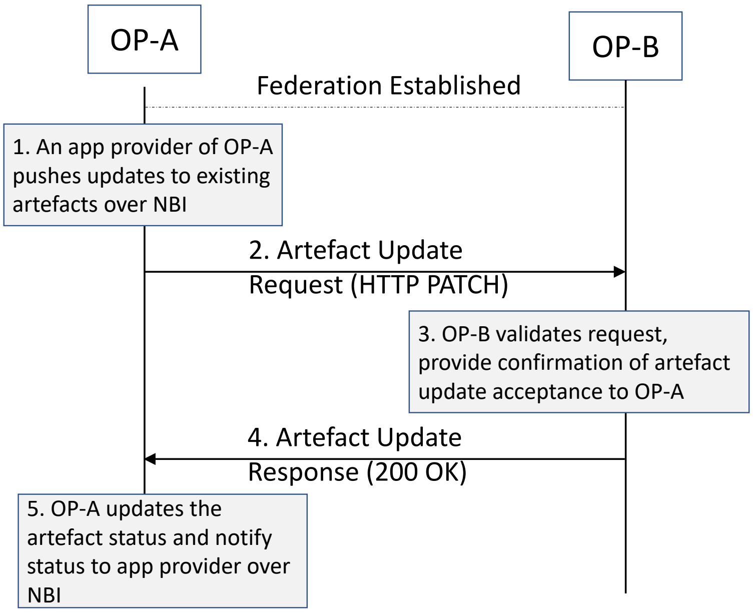
OP-A
OP-B
Federation Established
1. An app provider of OP-A pushes updates to existing artefacts over NBI
2. Artefact Update Request (HTTP PATCH)
3. OP-B validates request, provide confirmation of artefact update acceptance to OP-A
4. Artefact Update Response (200 OK)
5. OP-A updates the artefact status and notify status to app provider over NBI

Figure 9: 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 compute 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., compute 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 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 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)

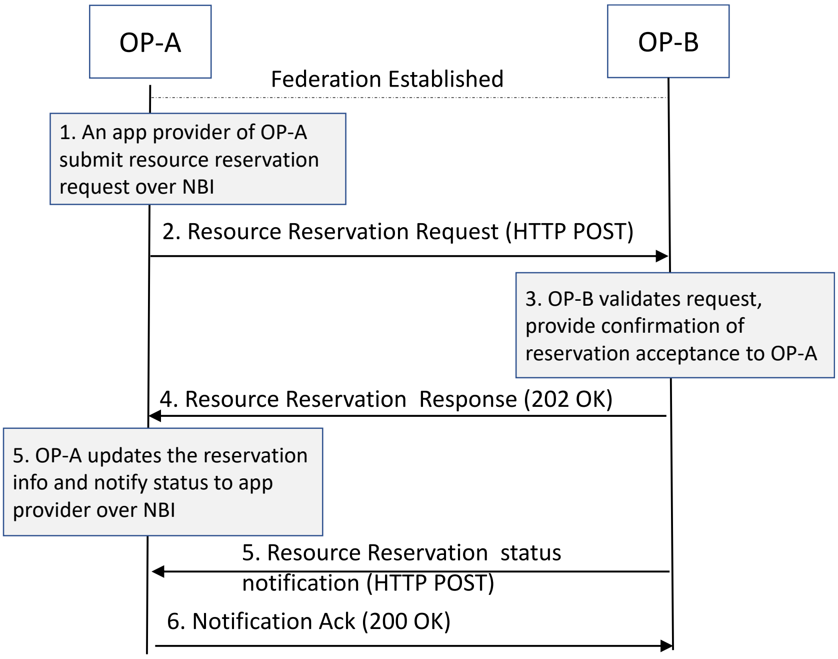
OP-A
OP-B
Federation Established
1. An app provider of OP-A submit resource reservation request over NBI
2. Resource Reservation Request (HTTP POST)
3. OP-B validates request, provide confirmation of reservation acceptance to OP-A
4. Resource Reservation Response (202 OK)
5. OP-A updates the reservation info and notify status to app provider over NBI
5. Resource Reservation status notification (HTTP POST)
6. Notification Ack (200 OK)

Figure 10: 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.

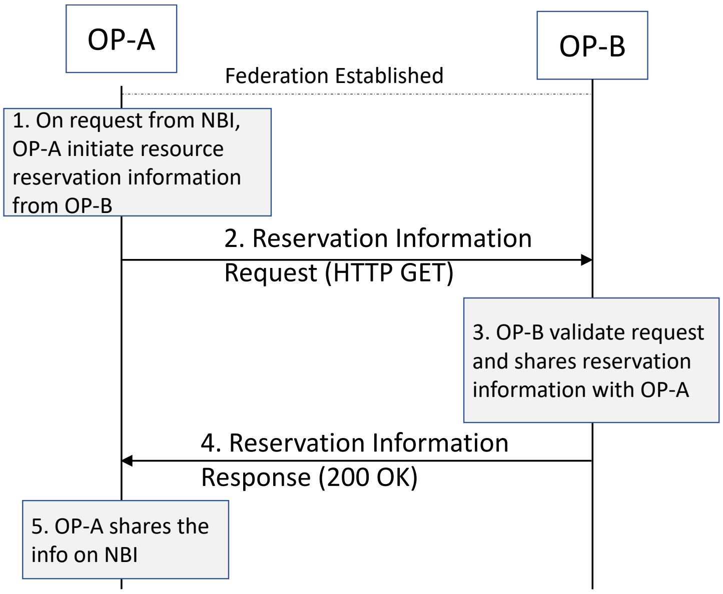
OP-A
OP-B
Federation Established
1. On request from NBI, OP-A initiate resource reservation information from OP-B
2. Reservation Information Request (HTTP GET)
3. OP-B validate request and shares reservation information with OP-A
4. Reservation Information Response (200 OK)
5. OP-A shares the info on NBI

Figure 11: 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, zone identifier and resources to be updated.

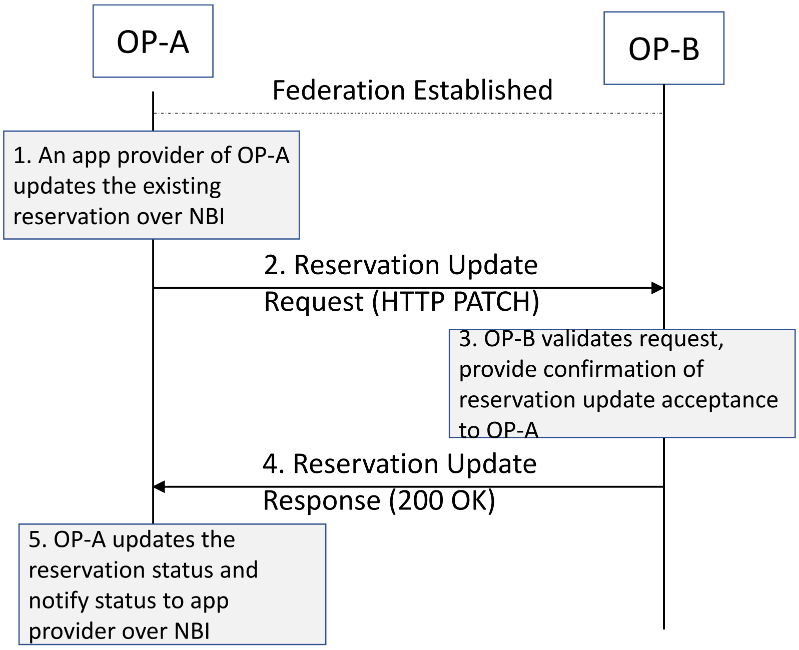
OP-A
OP-B
Federation Established
1. An app provider of OP-A updates the existing reservation over NBI
2. Reservation Update Request (HTTP PATCH)
3. OP-B validates request, provide confirmation of reservation update acceptance to OP-A
4. Reservation Update Response (200 OK)
5. OP-A updates the reservation status and notify status to app provider over NBI

Figure 12: 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 partners 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 clouds as a follow-up action after onboarding.

- Transfer of other application-specific files, e.g., application manifest, specifying the workload information like mobility strategy, Quality of Service (QoS) and privacy policies etc., and other optional characteristics indicating the application's needs (latency, prioritization, reservation)
- 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 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.

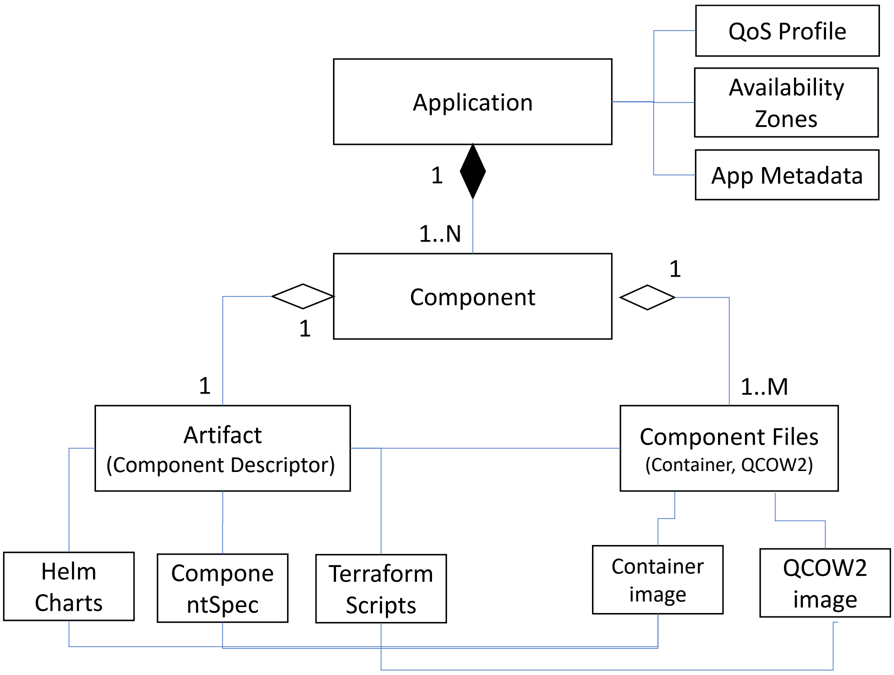

Figure 13: Application Schema

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 app 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., app name, app identifier, Application Provider identifier, Availability Zone(s), QoS profile etc.

2. The Partner OP validates the OP identity and authorization info, federation keys and zone onboarding status

    a) If application is already onboarded or is ongoing a correspondent failure response will be sent.
    b) Otherwise, after OP validation the Partner OP proceed to push application data (container images) to the edges and update/store all the info related to a database.

3. Once the application push is finish

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) 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.

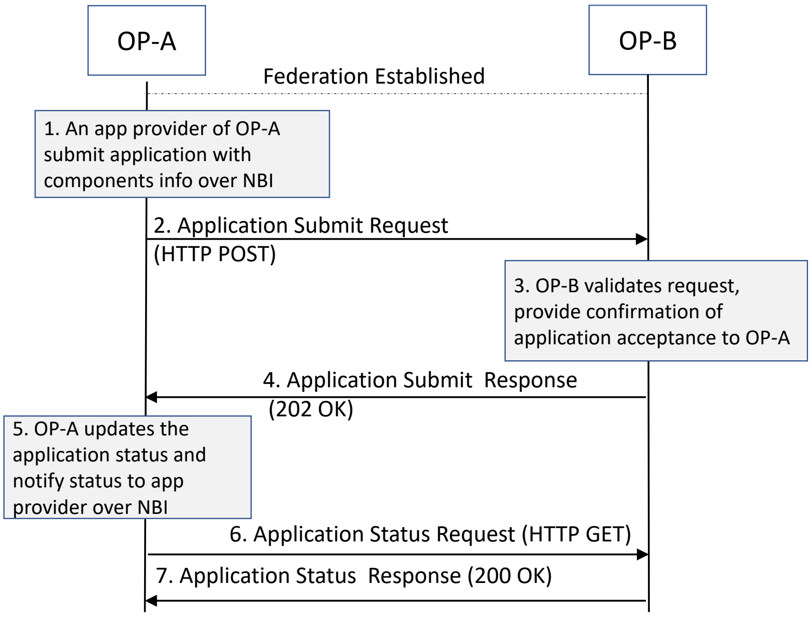
OP-A
OP-B
Federation Established
1. An app provider of OP-A submit application with components info over NBI
2. Application Submit Request (HTTP POST)
3. OP-B validates request, provide confirmation of application acceptance to OP-A
4. Application Submit Response (202 OK)
5. OP-A updates the application status and notify status to app provider over NBI
6. Application Status Request (HTTP GET)
7. Application Status Response (200 OK)

Figure 14: 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., app name, app identifier, Application Provider identifier, Availability Zone(s), QoS profile etc.

2. The Partner OP validates the OP identity and authorization info, federation keys etc.
3. On successful validation, if application indicated by app 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 “201: Application updated successfully” shall be sent from the Partner OP. (app name and app Identifier information can be included as well).
b) In other case a correspondent failure message will be generated by the partner OP.

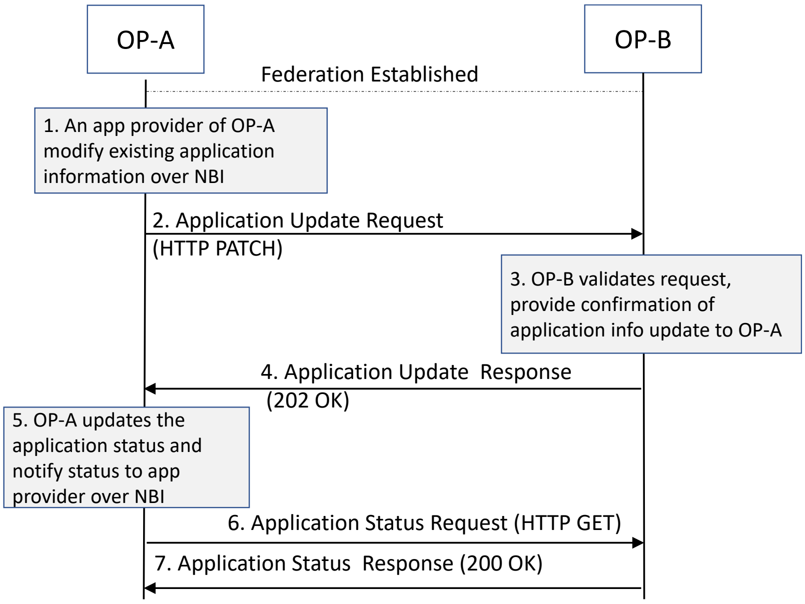
OP-A
OP-B
Federation Established
1. An app provider of OP-A modify existing application information over NBI
2. Application Update Request (HTTP PATCH)
3. OP-B validates request, provide confirmation of application info update to OP-A
4. Application Update Response (202 OK)
5. OP-A updates the application status and notify status to app provider over NBI
6. Application Status Request (HTTP GET)
7. Application Status Response (200 OK)

Figure 15: 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 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].

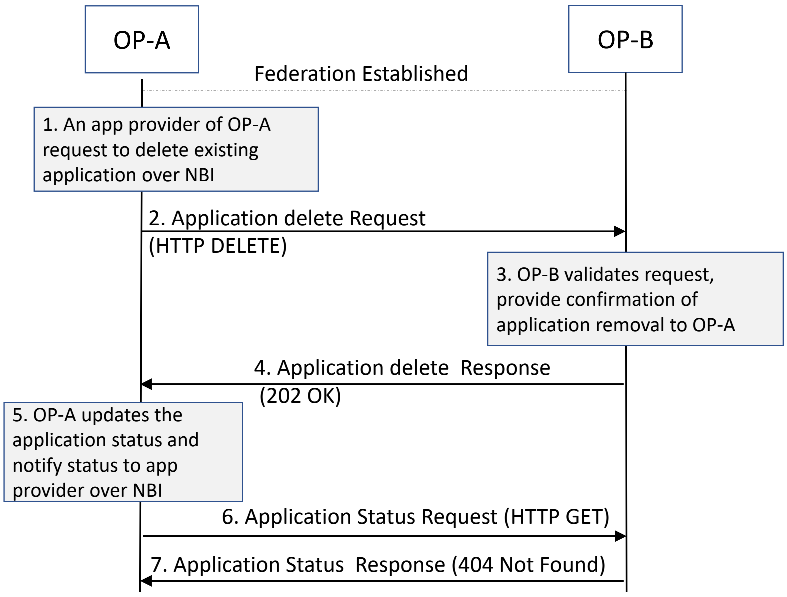
OP-A
OP-B
Federation Established
1. An app provider of OP-A request to delete existing application over NBI
2. Application delete Request (HTTP DELETE)
3. OP-B validates request, provide confirmation of application removal to OP-A
4. Application delete Response (202 OK)
5. OP-A updates the application status and notify status to app provider over NBI
6. Application Status Request (HTTP GET)
7. Application Status Response (404 Not Found)

Figure 16: Delete Application

2.3.1.3.6 Onboard Application On New Zones

This Originating OP can use this API to request the Partner OP to make already onboarded applications available on additional 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 zone onboarding status
   a) If application is already onboarded on the indicated 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 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 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 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 zone onboarding status

    a) If application on indicated 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 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 “202: Application restriction updated successfully” 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.

2.3.1.4 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 clouds of the Partner OP and to a partner OP 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.

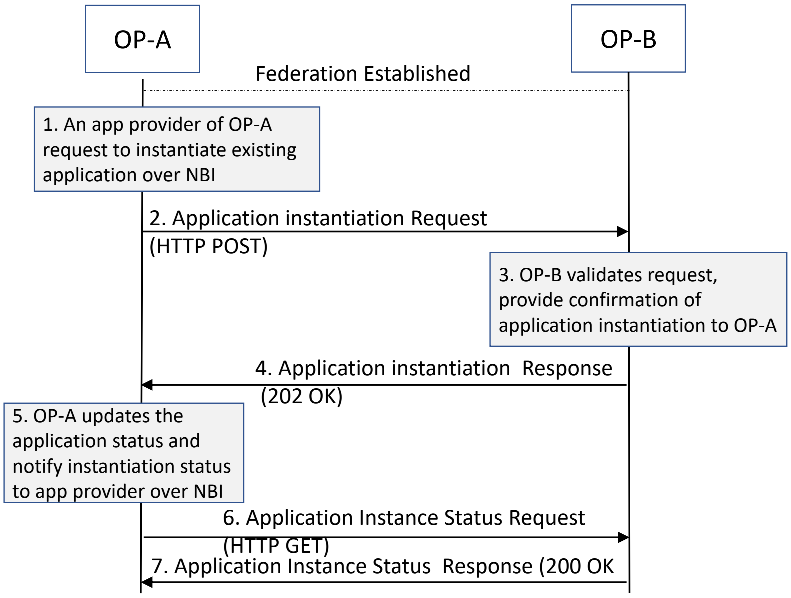
OP-A
OP-B
Federation Established
1. An app provider of OP-A request to instantiate existing application over NBI
2. Application instantiation Request (HTTP POST)
3. OP-B validates request, provide confirmation of application instantiation to OP-A
4. Application instantiation Response (202 OK)
5. OP-A updates the application status and notify instantiation status to app provider over NBI
6. Application Instance Status Request (HTTP GET)
7. Application Instance Status Response (200 OK

Figure 17: Application Instantiation

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., app 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, zone meta-information, application instance endpoints etc.

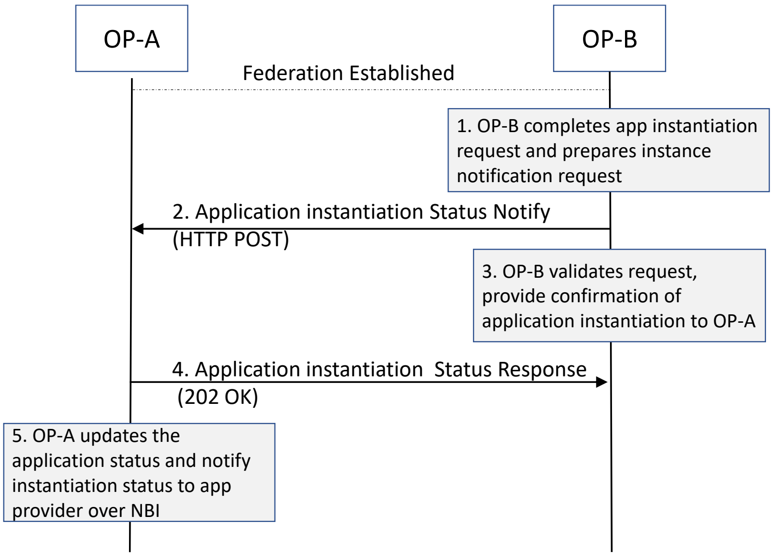
OP-A
OP-B
Federation Established
1. OP-B completes app instantiation request and prepares instance notification request
2. Application instantiation Status Notify (HTTP POST)
3. OP-B validates request, provide confirmation of application instantiation to OP-A
4. Application instantiation Status Response (202 OK)
5. OP-A updates the application status and notify instantiation status to app provider over NBI

Figure 18: 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 clouds.

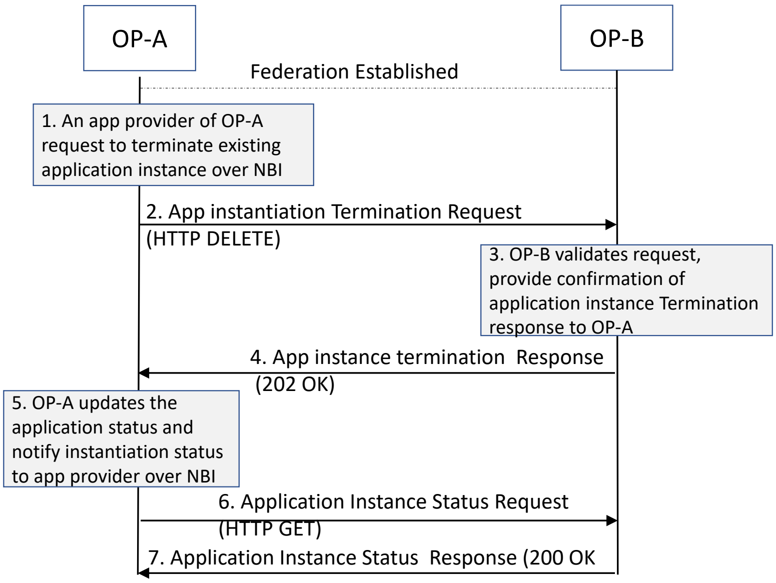
OP-A
OP-B
Federation Established
1. An app provider of OP-A request to terminate existing application instance over NBI
2. App instantiation Termination Request (HTTP DELETE)
3. OP-B validates request, provide confirmation of application instance Termination response to OP-A
4. App instance termination Response (202 OK)
5. OP-A updates the application status and notify instantiation status to app provider over NBI
6. Application Instance Status Request (HTTP GET)
7. Application Instance Status Response (200 OK

Figure 19: Application Instance Termination

2.3.1.5 Procedures for sharing edge resources between federated OP partners

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 compute 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/operator platform and requests for the required Edge-Enhanced or Edge-Native 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 partner OP edge clouds. Alternatively, a 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 Cloudlet of OP-B at which the subscriber can connect to the application.

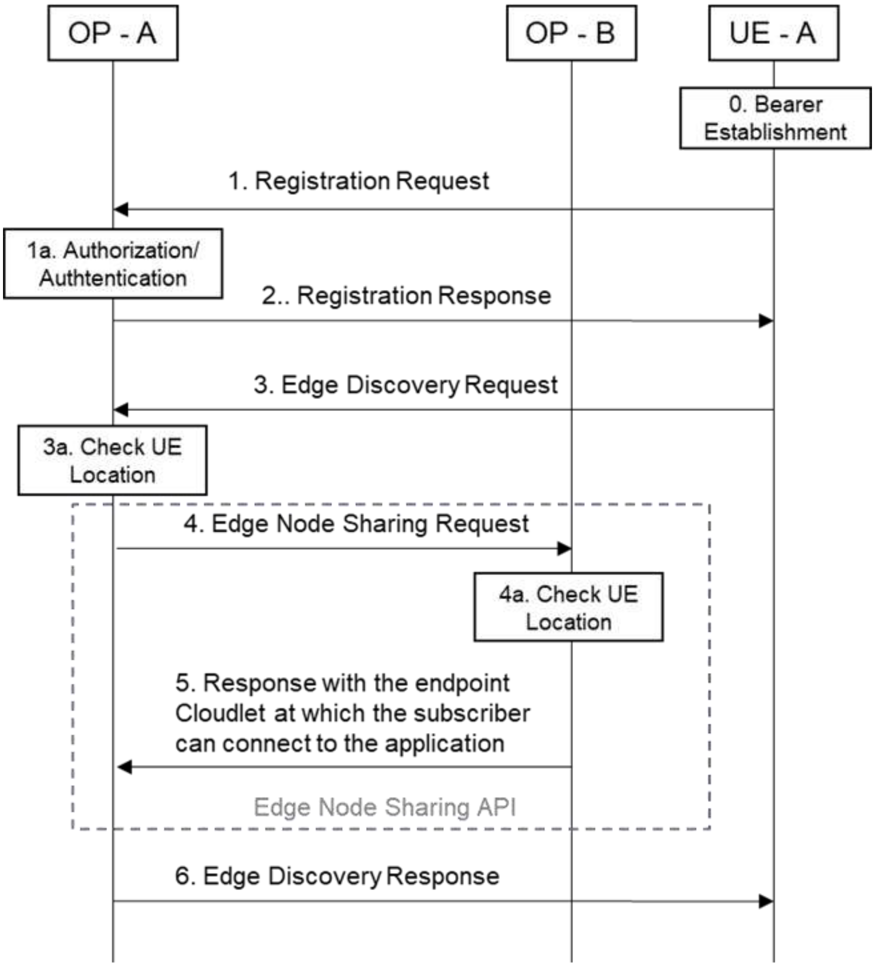
OP - A
OP - B
UE - A
0. Bearer Establishment
1. Registration Request
1a. Authorization/ Authtentication
2.. Registration Response
3. Edge Discovery Request
3a. Check UE Location
4. Edge Node Sharing Request
4a. Check UE Location
5. Response with the endpoint Cloudlet at which the subscriber can connect to the application
Edge Node Sharing API
6. Edge Discovery Response

Figure 20: Sharing edge resources between federated OP partners

Based on the OP-B response to edge node (compute resources in 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.

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 operator platforms 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 operator platforms 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 describe 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	

Table 1: E/WBI Interface Management APIs

3.1.1.2 Create Federation : POST Method

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 Table 2 below describes the data structures supported by the POST Request Body on this resource.

Parameter Name	P	Cardinality	Description
origOPFederationId	M	1	Operators in federation shall be governing the namespace and operator identifier assigned to it.

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
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 Table 3 below describe the data structures supported by the POST Response Body on this resource for 200 OK.

Parameter Name	P	Cardinality	Description
partnerOPFederationId	M	1	Operators in federation shall be governing the namespace and operator identifier assigned to it.
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 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 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 HomeRouting, Anchoring as supported by the Partner OP.
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 Table 4 below describe 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 Table 5 below describe 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
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 Table 6 below 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, zone status, edge discovery URL, network codes etc.
operationType	M	1	Type of update for e.g., Change in status, add network code, update edge discovery URL etc.
modificationDate	M	1	Date and time of the federation modification by the Partner OP
edgeDiscoverySvcEndPoint	O	1	Edge discovery service URL for UNI interface.
lcmSvcEndPoint	O	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
removeFixedNetworkIds	O	1..N	List of fixed network codes to be removed
addZones	O	1..N	New zones to be added. List of 'availabilityZone'.
removeZones	O	1..N	List of zoneIds to be removed
zoneStatus	O	1..N	Availability status of zones

Table 6: Notify federation updates request parameters

The Table 7 below describe 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 PATCH 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 Table 9 below describe 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

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.

Table 10: Zone meta info request parameters

The Table 11 below describe the data structures supported by the GET Response Body on this resource for response code 200 OK.

Parameter Name	P	Cardinality	Description
edgeDiscoveryServiceEndPoint	M	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 zones a partner OP is willing to share. The Partner OP may configure such information using system management interface
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	M	1	IP and Port of LCM Service URL of the Partner OP. This can also be a FQDN.

Table 11: Federation meta info response parameters

The Table 12 below describe 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 zone meta-information Request

3.1.1.6 PATCH Method : Update Federation

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 Table 14 below describe 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 Data Model

3.1.1.7.1 General

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

Data Type	Clause Defined	Description
federationContextId	3.1.2.3.1	Federation relationship Identifier generated by the Partner OP
partnerOPFederationId	3.1.2.3.1	Unique public identifier for the Partner OP
partnerOPCountryCode	3.1.2.3.1	Mobile Country Code (MCC) of the Partner OP
partnerOPNetworkCodes	3.1.2.3.1	Mobile Network Codes (MNCs) of the Partner OP
partnerOPFixedNetworkCodes	3.1.2.3.1	Fixed Network Codes of the Partner OP
origOPFederationId	3.1.2.3.1	Unique public identifier for the originating OP
origOPCountryCode	3.1.2.3.1	Mobile Country Code (MCC) of the originating OP
origOPNetworkCodes	3.1.2.3.1	Mobile Network Codes (MNCs) of the originating OP
origOPFixedNetworkCodes	3.1.2.3.1	Fixed Network Codes of the originating OP
offeredAvailabilityZones	3.1.2.2.1	List of zones Partner OP offers to share with originating OP
edgeDiscoveyServiceEndPoint	3.1.2.2.3	IP and Port of Edge Discovery Service URL of Partner OP
updateType	3.1.2.3.1	Indicates which Parameter being updated by Partner OP for existing federation
mncChangeInfo	3.1.2.3.3	Structure for add or remove mobile network code(s)

Table 15: 5.1 East/West Bound Interface Management Params

3.1.1.7.2 Structured Data Types

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

3.1.1.7.2.1 offeredAvailabilityZones

Following Table 16 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 16: Availability Zones meta information

3.1.1.7.2.2 availabilityZone

Following Table 17 describe the data elements of an Availability Zone.

Attribute Name	Data Type	P	Cardinality	Description
zoneId	String	M	1	Unique Identity of a Zone
geolocationInfo	String	M	1	Latitude/Longitude of 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 17: Availability Zone location parameters

3.1.1.7.2.3 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 18: Edge Discovery Service Endpoint

3.1.1.7.2.4 mncChangeInfo

Following Table 19 describe 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 19: Availability Zones meta information

3.1.1.7.2.5 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 20: Response body for error responses

3.1.1.7.2.6 InvalidParam

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

Table 21: InvalidParam

3.1.1.7.2.7 zoneStatus

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

status	String	M	1	Availability Status for the zone.

Table 22: InvalidParam

3.1.1.7.3 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.

3.1.1.7.3.1 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
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
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 a zone

Table 23: E/WBI Interface Management Simple Datatype table

3.1.1.7.3.2 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
ZONES	Change in the availability status of a 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

Table 24: Federation Modification Parameter types

3.1.1.7.3.3 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_CHANGE	Availability status of the resource has changed
UPDATE_SERVICE	Update endpoint of a service
ADD_CODES	Add network (mobile or fixed) network codes
REMOVE_CODES	Remove network (mobile or fixed) network codes
UPDATE_CODES	Update network (mobile or fixed) network codes. Some new codes are added, and some old codes are removed.
ADD_ZONES	Add new zones. Partner OP is willing to share new zones
REMOVE_ZONES	Remove old zones. Partner OP is no longer willing to share some previous zones
UPDATE_ZONES	Add some new zones and remove some old zones.
UPDATE_ZONE_STATUS	Change in the availability status of a zone
REM_FEDRATION_NOTIFY	Terminate the existing federation from the Partner OP

Table 25: Operations type for network code change

3.1.1.7.3.4 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 26: Allowed status values

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 Cloud resources provided to another.

3.1.2.1 Introduction

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

Operations	HTTP Method	Resource URI	Qualifier
Zone Subscribe	POST	/operatorplatform/federation/v1/{federationContextId}/zones	M

Zone Unsubscribe	DELETE	/operatorplatform/federation/v1/{federationContextId}/zones/{zoneId}	M
View Zone Information	GET	/operatorplatform/federation/v1/{federationContextId}/zones/{zoneId}	M
Notify Zone Information	POST	{ availZoneNotifLink}	M

Table 27: Availability Zone Synchronization APIs

3.1.2.2 Zone Subscribe : POST Method

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 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 zone information e.g., resource quota updates, addition of new zones etc. asynchronously

Table 28: Availability Zones subscription request parameters

The Table 29 below describe the data structures supported by the POST Response Body on this resource.

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

Table 29: Availability Zones subscription response parameters

The Table 30 below describe 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.
problemDetails	C	520	Web Server Returned an Unknown Error

Table 30: Availability Zones subscription response parameters

3.1.2.3 Zone Unsubscribe : DELETE Method

Table 31 provides parameters which an Originating OP sends to the Partner OP in 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	Zone identifier of partner operator. The Partner OP shall deregister the indicated zone and may reclaim the resources.

Table 31: Availability Zones Unsubscribe request parameters

The Table 32 below describe 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 32: Availability Zones Unsubscribe response parameters

3.1.2.4 View Zone Information : GET Method

Table 33 below provides parameters which an Originating OP sends to a Partner OP in view zone information request for 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	Zone identifier of partner operator Availability Zone. The Partner OP shall deregister the indicated zone.

Table 33: Availability Zones information request parameters

The Table 34 below describe 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 Zone Resource information provided by Partner OP for accepted 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 34: Availability Zones information response parameters

3.1.2.5 Notify Zone Information : POST Method

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 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 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 35: Availability Zones notify request parameters

The tTable 36 below describe 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 36: 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 Zone Resource information provided by the Partner OP for accepted zone IDs to originating OP. It includes zoneId, guaranteed Resources and upper Limit Quota (E.g., vCPU, Memory, Storage, GPU etc.)
partnerAvailabilityZones		List of zones a partner OP is willing to share. Partner may configure such information using system management interface

Table 37: 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 38 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 (zoneResourceInfo)	M	1..N	Partner edge resource information available for applications consumptions

Table 38: List of Availability Zones with offered resources

3.1.2.6.2.2 zoneResourceInfo

The 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	Zone identifier to refer to a zone
reservedResources	Array (computeResourceInfo)	M	1..N	Resources exclusively reserved for a partner OP
resourceQuotaLimits	Array (computeResourceInfo)	M	1..N	Max quota on Resources that an OP may allow to partner OP over reserved resources if available
flavoursSupported	Array(computeFlavour)	M	1..N	Compute resources flavours are set of OP defined compute resources combination which a partner OP supports and offers to

				application providers to be link them to applications for runtime resource requirements
networkResources	Array(networkResourceInfo)	O	1..N	Type of networks supported by the partner zone

Table 39: Availability Zones information data structure

3.1.2.6.2.3 computeResourceInfo

Compute 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(gpuResourceInfo)	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 40: Availability Zone Compute resource information

3.1.2.6.2.4 gpuResourceInfo

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 41: 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 edge clouds.

Attribute Name	Data Type	P	Cardinality	Description
flavourId	Int	M	1	An identifier to refer to the combination of compute 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 zone
cpuExclusivity	Boolean	O	1	If the zone supports exclusive allocation of Intel CPUs.

Table 42: Compute flavour for Virtual Machines

3.1.2.6.2.6 operatingSystemInfo

The tTable 43 below 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 43: 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 zone supports Single Root Input Output Virtualisation (SRIOV) based networking or not.
supportDPDK	Boolean	M	1	If the zone supports Data Plane Development Kit (DPDK)-enabled userspace networking or not.

Table 44: 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 45: 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 zone

Table 46: Definition of zoneResUpdInfo

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 47: 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 48: 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 49 : 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 50 : 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 51 : 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 52 : 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 53 describe 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 53: Artefacts Management APIs

4.1.1.2 Onboard Artefact : POST Method

The Table 54 below describe 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 user clients, 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.
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 54: Onboard Artefact request parameters

The Table 55 below describe 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 partners 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 55: Onboard Artefact response parameters

4.1.1.3 DELETE Method : Remove Artefact

The Table 56 below describe 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 56: Remove Artefact request parameters

The Table 57 below describe the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Code	Description
Status	C	200	Artefact 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 57: Remove Artefact response parameters

4.1.1.4 GET Method : View Artefact Information

The Table 58 below describe 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 58: View Artefact request parameters

The Table 59 below describe the data structures supported by the DELETE 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.
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 59: View Artefact response parameters

The Table 60 below describe 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 60: Non-200 Response Codes for View Artefact Response

4.1.1.5 POST Method : Upload File

The tTable 61 below describe 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
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 61: Upload File request Parameters

The Table 62 below describe the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Cardinality	Description
N/A	C	200	File uploaded 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	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 62: Upload File response Parameters

4.1.1.6 DELETE Method : Remove Upload File

The Table 63 below describe 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 63: Remove File request parameters

The Table 64 below describe 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
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 64: Remove File response parameters

4.1.1.7 GET Method : View File Information

The Table 65 below describe 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 65: View File request parameters

The Table 66 below describe 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 66: 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 tTable 67 describe 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 67: 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 compute 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 68: 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 69: 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 edge cloud. When exposed to external world, an external dynamic port is assigned for UC traffic and mapped to the internal containerPort
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 70: 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(hugepageInfo)	O	1..N	Huge pages for workload for a given ISA type
cpuExclusivity	Boolean	O	0..1	Support for exclusive CPUs

Table 71: 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". 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 72: 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 73: Persistent Volume

4.1.1.8.2.8 artefactRepoLocation

Following Table 74 describe 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 74: 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 75: 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 76: 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 77: 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 78: 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 79: 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 80: 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 81: commProtocol table

4.1.2 Application Provider Resource Management - APIs

The REST APIs mentioned in this section provides the capabilities to reserve and manage compute resources for an application provider within the 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 on per 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 82: Compute Resource Reservation Management Methods

4.1.2.2 POST Method : Reserve Compute Resources

The Table 83 below describe 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 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 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 83: Reserve Compute Resources request parameters

The Table 84 below describe 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 84: Reserve Compute Resource response parameters

4.1.2.3 PATCH Method : Update Compute Resource Reservation

The Table 85 below describe 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	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 85: Update Compute Resource Reservation request parameters

The Table 86 below describe 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 86: Update Compute Resource Reservation response parameters

4.1.2.4 GET Method : View Reserved Resources

The Table 87 below describe 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	Zone where resources are reserved.
appProviderId	M	1	A unique Application Provider Identifier referring an application provider account with Originating OP

Table 87: View Reserved Resource request parameters

The Table 88 below describe 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
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 88: Notify resource reservation status response parameters

4.1.2.5 DELETE Method : Remove Reserved Resources

The Table 89 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
zoneId	M	1	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 89: Remove Reserved Resource request parameters

The Table 90 below describe 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

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 90: 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 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 91: ISV resource reservation status notification parameters

The Table 92 below describe 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 92: 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 93describe 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 compute 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 93: resRequest

4.1.2.7.2.2 flavourResvInto

The Table 94 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 94: flavourResvInfo

4.1.2.7.2.3 reservedPoolId

The Table 95 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 95: reservedPoolId

4.1.2.7.2.4 UpdResInfo

The data structure in the below Table 96 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 96: updResInfo

4.1.2.7.2.5 reservedPools

The following Table 97 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 97: Reserved Pool Info

4.1.2.7.2.6 reserveDuration

The following Table 98 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 98: reservationDuration

4.1.2.7.2.7 grantedFlavours

The following Table 99 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 99: 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 100 describe 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
App Onboard at new 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 100: Application Onboarding Management APIs

4.1.3.2 Onboard Applications : POST Method

The Table 101 below describe 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
appIInformation	M	1	Application compute resource, component images, QoS, Availability Zone information

Table 101: Application Onboarding request parameters

The Table 102 below describe the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	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 102: Application Onboarding response parameters

4.1.3.3 Update Application Information : PATCH Method

The Table 103 below describe 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 compute resource, components, associated artefactId,
appUpdQoSProfile	O	1	Application resource requirement or deployment attributes that needs to be updated
appComponentSpecs	O	1	Application components and their assosiated artefacts or Domain Name System (DNS).

Table 103: Modify application information request parameters

The Table 104 below describe the data structures supported by the PATCH Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	201	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 104: Modify application information response parameters

4.1.3.4 DELETE Method : Remove Application Information

The Table 105 below describe 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	zone identifiers from where application must be deboarded.

Table 105: Remove application request parameters

The Table 106 below describe the data structures supported by the DELETE Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	Application Updated successfully
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 106: Remove application response parameters

4.1.3.5 POST Method: Notify resource reservation Status Information

The Table 107 below 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
statusInfo	M	1	Status of an application on zone.

Table 107: Resource reservation notification parameters

The Table 108 below describe 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 108: Resource reservation notification response parameters

4.1.3.6 Application Onboarding At New Zones : POST Method

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

The Table 109 below describe 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 zone identifiers where application shall be made available.

Table 109: Application Onboarding on new zones request parameters

The Table 110 below describe the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	202	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 110: Application Onboarding response parameters

4.1.3.7 Restrict Application : POST Method

The Originating OP request partner OP to restrict or allow instantiation of the application on specified zones.

The Table 111 below describe 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 zone identifier and access info

Table 111: Application Onboarding request parameters

The Table 112 below describe 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 112: Application Onboarding response parameters

4.1.3.8 Data Model

4.1.3.8.1 General

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

4.1.3.8.2 Structured Data Types

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

4.1.3.8.2.1 appInformation

Following Table 113 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, compute resources etc.
appStatusCallbackLink	uri	M	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.

Table 113: appInformation

4.1.3.8.2.2 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 114: regionInfo

4.1.3.8.2.3 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”

Table 115: Application meta data

4.1.3.8.2.4 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 116: Application QoS profile

4.1.3.8.2.5 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 117: Application Components

4.1.3.8.2.6 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. 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. 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 118: Application Component Details

4.1.3.8.2.7 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 119: Country Code

4.1.3.8.2.8 zoneInfo

Table 120 describe zone identifier where an application shall be onboarded.

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

Table 120: Zone identifier info for application onboarding

4.1.3.8.2.9 appUpdQoSProfile

Update request shall contain atleast one of the optional parameter defined in below Table 121.

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 app supports single user or multiple user clients (UCs)
noOfUsersPerAppInst	Integer	C	1	For multi user client’s app, how many UCs an app instance can support
appProvisioning	Bool	O	1	Define if application can be instantiated or not

Table 121: Application QoS profile Update Parameters

4.1.3.8.2.10 statusInfo

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

Table 122: StatusInfo

4.1.3.8.2.11 appInstantiationCtrlList

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

Table 123: appInstantiationCtrlList

4.1.3.8.3 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.

4.1.3.8.3.1 Enumeration: multiUserClients

The Table 124 define 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 124: multiUserClients

4.1.3.8.3.2 Enumeration: onboardingStatusInfo

The Table 125 below 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 125: Onboarding status info

4.1.3.8.3.3 Enumeration: resourceConsumption

The Table 126 define 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 126: 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 127 describe the applicable HTTP methods for applications lifecycle management.

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

		/{appId}/instance/{appInstance}/zone/{zoneId}	
List Application Instances	GET	/operatorplatform/federation/v1/{federationContextId}/application/lcm/app/{appId}/appProvider/{appProviderId}	M
Notify Application Instance state information	POST	{appInstCallbackLink}	M

Table 127: Application Instance Management Methods

4.1.4.2 POST Method : Instantiate Applications

The Table 128 below describe 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 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	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 128: Application instantiation request parameters

The Table 129 below describe the data structures supported by the POST Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	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 129: 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	Zone Identifier where app instance is running
appInstIdentifier	M	1	Application instance identifier to refer to a running instance of an application denoted by appId

Table 130: Application instance termination request parameters

The Table 131 below describe 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
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: Application instance termination response parameters

4.1.4.4 Notify Application Instance Information : POST Method

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	Zone identifier of the app referred by appId
appInstanceInfo	M	1	Application instance information e.g., communication end points of various components of the app, zone where it is deployed denoted by appId.

Table 132: Application instance async request parameters

The Table 133 below describe 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
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: Application instance async response parameters

4.1.4.5 View Application Instance Details : Get Method

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 zone where application instance is created.

Table 134: Application instance async request parameters

The Table 135 below describe 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.
problemDetails	C	520	Server Returned an Unknown Error

Table 135: View application instance details response parameters

4.1.4.6 List Application Instances : Get Method

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	zone identifier where app referred by appId is deployed

Table 136: List application instance parameters

The Table 137 below describe 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 137: 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 138 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 138: appInstanceInfo

4.1.4.7.2.2 accessPointInfo

Following Table 139 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 app component structure
accessPoints	Object	M	1	Details of IP address, port, FQDN etc.

Table 139: accessPointInfo

4.1.4.7.2.3 accessPoints

Following Table 140 describes the protocol level details of the connectivity information of an edge application instance.

Attribute Name	Data Type	P	Cardinality	Description
fqdn	String	C	1	fqdn of the app component instance on requested zone where UC can connect with app instance on edge
ipv4Addresses	IPv4 Address	C	1	IPv4 address of the app component instance on requested zone where UC can connect with app instance on edge
ipv6Addresses	IPv6 Address	C	1	IPv6 address of the app component instance on requested zone where UC can connect with app instance on edge
port	string	M	1	Port of the app component instance on requested zone where UC can connect with app instance on edge

Table 140: accessPoints

4.1.4.7.2.4 appInstanceList

Following Table 141 describe 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 app component instance on requested zone where UC can connect with app instance on edge
appInstanceInfo	Array	M	1..N	List for app instance Identifier and instance state

Table 141: application Instance list

4.1.4.7.2.5 InstanceIdentifiers

List of zoneIds and application instances created on that zone

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

Table 142: Application Instance Identifiers

4.1.4.7.2.6 zoneInfo

The Table 143 defines the 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	Zone identifier
flavourId	String	M	1	Flavour that should be used for the application on a zone
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 compute resource on each zone.

Table 143: Zone and flavour info for application instantiation

4.1.5 Edge Node Sharing - API

4.1.5.1 Introduction

Following Table 144 describe 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 144: 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 145: Edge Node Discovery Request Parameters

The following Table 146 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 146: Edge node discovery response Parameters

The Table 147 below describe 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 147: 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 148: 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 easDiscoveryResp

The following Table 148 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
discoveredEdgeNodes	Array (discoveredEdgeNodes)	M	1..N	List of Edge discovery information (e.g. URI, FQDN, IP address)

Table 149: easDiscoveryResp

4.1.5.3.2.2 discoveredEdgeNodes

The following Table 150 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 user device to determine traffic latency.

Table 150: discoveredEdgeNodes

4.1.5.3.2.3 edgeDiscoveryFilters

The following Table 151 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 151: edgeDiscoveryFilters

4.1.5.3.3 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.

4.1.6 LBO Roaming Authentication – API

4.1.6.1 Introduction

An OP uses the HTTP POST method on the resource URI described in table below to authenticate roaming users of the Partner OP. Following Table 152 describe the applicable HTTP methods and the resource URI for LBO roaming authentication API .

Operations	HTTP Method	Resource URI	Qualifier
User Authentication	GET	/operatorplatform/federation/v1 /{federationContextId}/roaminguserauth/device/{deviceId}/token/{authToken}	M

Table 152: Roaming user authentication Operations

4.1.6.2 GET Method : Authenticate roaming user

The Table 153 below describe 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 Visited OP to the Home OP of the user to identify the existing federation relationship

deviceId	M	1	Roaming user device Id to identify and authenticate the roaming user by home mobile network
authToken	M	1	An authentication token assigned to the roaming user by the Home OP when UC tries to register from visited network. The token is provided to the Visited OP by the user client when it is redirected to register with visited OP. It is used by the Visited OP to authenticate the roaming user by the Home OP

Table 153: Roaming user authentication request parameters

The Table 154 below describe the data structures supported by the GET Response Body on this resource.

Parameter Name	P	Response Codes	Description
N/A	C	200	Device Auth Token validated
problemDetails	C	400	Bad Request
N/A	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 154: Roaming user authentication response parameters

4.1.6.3 Data Model

4.1.6.3.1 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.

4.1.6.3.1.1 Simple data types

Attribute Name	Data Type	Description
deviceId	String	Device identifier as determined by the visited mobile network for roaming user

authToken	String	Temporary token to be used by the client application to authenticate itself to the Partner OP

Table 155: Roaming user authentication simple datatype table

5 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.

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: '1.0.0'
  title: 'Federation Management Service'
  description:  |
    # Introduction
    ---
    RESTful APIs that allow an OP to share the edge cloud 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:
    * __FederationManagement__ - Create and manage directed federation relationship with a partner OP
    * __AvailabilityZoneInfoSynchronization__ - Management of resources of partner OP 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 zones
    * __EdgeNodeSharing__ - Edge discovery procedures towards partner OP over E/WBI.
    * __LBORoamingAuthentication__ -  Validation of user client authentication from home OP

    ---
    # Definitions
    ---
    This section provides definitions of terminologies commonly referred to throughout the API descriptions.

    * __Accepted Zones__  - List of partner OP zones, which the originating OP has confirmed to use for its edge applications
    * __Anchoring__ - Partner OP capability to serve application clients (still in their home location) from application instances running on partner zones.
    * __Application Provider__ - An application developer, onboarding his/her edge application on a partner operator platform (MEC).
    * __Artefact__ - Descriptor, charts or any other package associated with the application.
    * __Availability Zone__ - 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 users use these devices to access edge applications
    * __Directed Federation__ - A Federation between two OP instances A and B, in which edge compute 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 west bound interface.
* __Federation__ - Relationship among member OPs who agrees to offer services and capabilities to the application providers and end users 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 developers or user clients are registered.
* __Home Routing__ - Partner OP capability to direct roaming user client traffic towards application instances running on home OP 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__ - 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.
* __OperatorIdentfier__ - 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 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 zones it wishes to use.
* __User Clients__ - Lightweight client applications used to access edge applications. Application users run these clients on their devices (UE, IOT device, AR/VR device etc)

# 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 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

__AvailabilityZoneInfoSynchronization__

* __ZoneSubscribe__ Informs partner OP that originating OP is willing to access the specified zones and partner OP shall reserve compute and network resources for these zones.

```
  * __ZoneUnsubscribe__  Informs partner OP that originating OP will no longer access the specified partner OP zone.
  * __GetZoneData__  Retrieves details about the computation and network resources that partner OP has reserved for an partner OP  zone.
  * __Notify Zone Information__ Call back notification used by partner OP to update originating OP about changes in the resources reserved on a partner zone.

  __ArtefactManagement__
  * __UploadArtefact__  Uploads application artefact  on partner operator platform.
  * __RemoveArtefact__  Removes an artefact from partner operator platform.
  * __GetArtefact__  Retrieves details about an artefact from partner operator platform.
  * __UploadFile__ Upload application binaries to partner operator platform
  * __RemoveFile__  Removes application binaries from partner operator platform
  * __ViewFile__  Retrieves details about binaries assosiated with an application from partner operator platform

  __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 compute resource requirements, QOS Profile, associated descriptor or change in assosiated components
  * __DeboardApplication__ - Removes an application from partner OP
  * __ViewApplication__ - Retrieves application details from partner OP
  * __OnboardExistingAppNewZones__ - Make an application available on new additional zones
  * __LockUnlockApplicationZone__ - Forbid or permit instantiation of application on a zone

  __Application Instance Lifecycle Management__
  * __InstallApp__ - Instantiates an application on a partner OP zone.
  * __GetAppInstanceDetails__ - Retrieves an application instance details from partner OP.
  * __RemoveApp__ - Terminate an application instance on a partner OP zone.
  * __GetAllAppInstances__ - Retrieves details about all instances of the application running on partner OP zones.


  __AppProviderResourceManagement__
  * __CreateResourcePools__  Reserves resources (compute, network and storage)  on a partner OP zone. ISVs registered with home OP reserves resurces on a partner OP 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 zones where an application instance can be created.


  __LBORoamingAuthentication__
  *__AuthenticateDevice__ Validates the authenticity of a roaming user from home OP


  © 2022 GSM Association.
  All rights reserved.

externalDocs:
 description: GSMA, E/WBI APIs v0.9
 url: 'http://www.xxxx.com'
servers:
 - url: '{apiRoot}/operatorplatform/federation/v1'
```

```
    variables:
      apiRoot:
        default: https://operatorplatform.com


security:
  - oAuth2ClientCredentials:
    - fed-mgmt

components:
  securitySchemes:
    oAuth2ClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: '/oauth2/token'
          scopes:
            fed-mgmt: Access to the federation APIs

  schemas:
    AppIdentifier:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
      description: Identifier used to refer to an application. This identifier is globally unique so that application can
be identified uniquely across different OPs.

    AppProviderId:
      type: string
      pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
      description: UserId of the app 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.

    AuthorizationToken:
      type: string
      minLength: 8
      maxLength: 128
      description:  A token assigned to the roaming user’s during registration with home OP and the token is
provided back to the visited OP by
                          the user client on end user device when redirected to register with visited OP

    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_64
            - ISA_ARM_64
          description: CPU Instruction Set Architecture (ISA) E.g., Intel, Arm etc.

    DeviceId:
```

```
    type: string
    pattern:   ^[A-Za-z0-9][A-Za-z0-9_]{6,128}[A-Za-z0-9]$
    description: The identifier of the application user  (i.e., GPSI or preferably an identity token)

  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 zone.

  InstanceState:
    type: string
    enum:
      - PENDING
      - READY
      - FAILED
      - TERMINATING
    description: Running status of the application instance.

  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: '^([0-9]{5,6}-)?[^-]+$'
    readOnly: true
    description: This key 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 key in any subsequent request towards the partner op.

  FederationIdentifier:
    type: string
    pattern: '^[A-Fa-f0-9]*$'
    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.

    GeoLocation:
      type: string
      description: "Latitude, Longitude as decimal fraction upto 4 digit precision"
      pattern: ^([-+]?)([\d]{1,2})(((\.)(\d+)(,)))(\s*)(([-+]?)([\d]{1,3})((\.)(\d+))?)$

    Mcc:
      type: string
      pattern: '^\d{3}$'

    Mnc:
      type: string
      pattern: '^\d{2,3}$'

    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

    Uri:
      type: string

    Version:
      type: string
      pattern:  ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
      description: Versioning info in the format major.minor.patch

    ZoneIdentifier:
      type: string
      pattern: '^[A-Fa-f0-9]*$'
      description:  Human readable name of the zone.

#
# STRUCTURED DATA TYPES
#

    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
            - PLATFRM_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.


    ComponentSpec:
      description: Details about compute, networking and storage requirements for each component of the
application.
                         App 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
        - osType
        - InstSetArch
        - images
        - numOfInstances
        - restartPolicy
        - networkResourceProfile
        - 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
        osType:
          $ref: '#/components/schemas/OSType'
        InstSetArch:
          $ref: '#/components/schemas/CPUArchType'
        imagesPath:
          description: List of all images associated with the component. Images are uploaded or specified using
Upload File apis
          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 platform shall handle component failure
        commandLineParams:
           $ref: '#/components/schemas/CommandLineParams'
        exposedInterfaces:
          description: Each application component exposes some ports either for external users or for inter
component communication.
                              Application provider is required to specify which ports are to be exposed and the type of traffic
that will flow through these ports.

          type: array
          items:
             $ref: '#/components/schemas/InterfaceDetails'
          minItems: 1
        computeResourceProfile:
          $ref: '#/components/schemas/ComputeResourceInfo'
        compEnvParams:
          type: array
          items:
             $ref: '#/components/schemas/CompEnvParams'
        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
        - diskStorage
      properties:
        cpuArchType:
          type: string
          enum:
            - ISA_X86_64
            - ISA_ARM_64
          description: CPU Instruction Set Architecture (ISA) E.g., Intel, Arm etc.
        numCPU:
          type: integer
          format: int32
          description: Number of available vCPUs.
        memory:
          type: integer
          format: int64
          description: Amount of RAM  in mega bytes
        diskStorage:
          type: integer
          format: int32
          description: Amount of disk storage  in giga bytes 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

    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 zones where application instance could be created. LatencyServiceEndpoint is
responsible for responding to
                          latency measurement request from client


    FederationRequestData:
      type: object
      required:
        - origOPFederationId
        - initialDate
        - federationNotificationDest
      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
        federationNotificationDest:
          $ref: '#/components/schemas/Uri'
    FederationResponseData:
      type: object
      required:
        - partnerOPFederationId
        - partnerOPCountryCode
        - edgeDiscoveyServiceEndPoint
        - lcmServiceEndPoint
        - offeredAvailabilityZones
        - capabilities
      oneOf:
        - required:
          -  partnerOPMobileNetworkCodes
        - required:
          - partnerOPFixedNetworkCodes
      properties:
        partnerOPFederationId:
          $ref: '#/components/schemas/FederationIdentifier'
        partnerOPCountryCode:
          $ref: '#/components/schemas/CountryCode'
        federationContextId:
          $ref: '#/components/schemas/FederationContextId'
        edgeDiscoveyServiceEndPoint:
          $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 zones, which the operator platform wishes to make available to developers/ISVs of requesting operator platform.
        platformCaps:
          type: array
          items:
            type: string
            enum:
              - homeRouting
              - Anchoring
            description:  Home routing - Operator platform is capable of routing edge application data traffic from its edges to user device in their home location.
                                  This is the case where user devices are  served in their home region (requesting platform 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 user devices to  edge application in user device home network.

    Flavour:
      type: object
      required:
        - flavourId
        - cpuArchType
        - supportedOSTypes
        - numCPU
        - memorySize
        - storageSize
      properties:
        flavourId:
          type: integer
          format: int32
          description: An identifier to refer to this combination of compute resources.
        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 mega bytes
        storageSize:
          type: integer
          format: int32
          description: Amount of disk storage  in giga bytes
        gpu:
          type: array
          items:
            $ref: '#/components/schemas/GpuInfo'
        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 mega bytes
        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
        - network
        - interfaceName
      properties:
        interfaceId:
          type: string
          description:  Each Port and corresponding traffic protocol exposed by the component is identified by a
name.
                              Application client on user device 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 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 app 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
   required:
    - repoType
    - repoURL
   properties:
    repoType:
     type: string
     enum:
      - PRIVATEREPO
      - PUBLICREPO
      - UPLOAD
     description: Artefact repository location. PUBLICREPO is used of public URLs like GitHub, Helm repo etc,
                              PRIVATEREPO  is used for private repo managed by the application developer,
                              UPLOAD is for the case when artefact is uploaded from MEC web portal.
        repoURL:
          $ref: '#/components/schemas/Uri'
        userName:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
          description:  Username to access the artefact repository
        password:
          type: string
          description:  Password to access the artefact repository
        token:
          type: string
          description:  Authorization token to access the artefact 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
        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
        license:
          type: string
          enum:
            - OS_LICENSE_TYPE_FREE
            - OS_LICENSE_TYPE_ON_DEMAND

    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
        detail:
          type: string
        cause:
          type: string
        invalidParams:
          type: array
          items:
              $ref: '#/components/schemas/InvalidParam'
          minItems: 1

    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/Fqdn'
        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
        - geolocation
        - 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.

    ZoneRegisterationRequestData:
      type: object
      required:
        - acceptedAvailabilityZones
        - availZoneNotifLink
      properties:
        acceptedAvailabilityZones:
          type: array
          items:
            $ref: '#/components/schemas/ZoneIdentifier'
          minItems: 1
        availZoneNotifLink:
          $ref: '#/components/schemas/Uri'

    ZoneRegisterationResponseData:
      type: object
      required:
        - acceptedZoneResourceInfo
      properties:
        acceptedZoneResourceInfo:
          type: array
          items:
            type: object
            required:
              - zoneId
              - reservedResources
              - resourceQuotaLimits
              - 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 zone support SRIOV networks or not
                  supportDPDK:
                    type: boolean
                    description: If this zone supports DPDK based networking.
          minItems: 1

#
# 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'
  default:
    description: Generic Error

paths:
 /partner:
  post:
   summary: Creates one direction federation with partner operator platform.
   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/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:
        $ref: '#/components/responses/default'
    callbacks:
      onPartnerStatusEvent:
        '{$request.body#/federationNotificationDest}':
          post:
            parameters:
              -   name: federationContextIdentifier
                  in: path
                  required: true
                  schema:
                      $ref:  '#/components/schemas/FederationContextId'
            requestBody:
              description: Notification payload.
              content:
                application/json:
                  schema:
                    type: object
                    required:
                      - objectType
                      - operationType
                      - modificationDate
                    properties:
                      objectType:
                        type: string
                        enum:
                          - FEDERATION
                          - ZONES
                          - EDGE_DISCOVERY_SERVICE
                          - LCM_SERVICE
                          - MOBILE_NETWORK_CODES
                          - FIXED_NETWORK_CODES
                      operationType:
                        type: string
                        enum:
                          - STATUS_CHANGE
                          - UPDATE_SERVICE
                          - ADD_CODES
                          - REMOVE_CODES
                          - UPDATE_CODES
                          - ADD_ZONES
                          - REMOVE_ZONES
                          - UPDATE_ZONES
                          - UPDATE_ZONE_STATUS
                          - REM_FEDRATION_NOTIFY
                      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 zones, which the operator platform wishes to make available to 
developers/ISVs of  requesting operator platform.
                          minItems: 1
                        removeZones:
                          type: array
                          items:
                            $ref: '#/components/schemas/ZoneIdentifier'
                          description: List of zones, which the operator platform no longer wishes to share.
                          minItems: 1
                        zoneStatus:
                          type: array
                          items:
                            type: object
                            required:
                             - zoneId
                             - status
                            properties:
                              zoneId:
                               $ref: '#/components/schemas/ZoneIdentifier'
                              status:
                                $ref: '#/components/schemas/Status'
                          minItems: 1
                        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'

  /{federationContextId}/partner:
    get:
      summary: Retrieves details about the federation context with the partner OP. The response shall provide info about the 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
                required:
                  - edgeDiscoveryServiceEndPoint
                  - lcmServiceEndPoint
                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/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'

    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
                operationType:
                  type: string
                  enum:
                    - ADD_CODES
                    - REMOVE_CODES
                    - UPDATE_CODES
                addMobileNetworkIds:
                  $ref: '#/components/schemas/MobileNetworkIds'
                removeMobileNetworkIds:
                  $ref: '#/components/schemas/MobileNetworkIds'
                addFixedNetworkIds:
                  $ref: '#/components/schemas/FixedNetworkIds'
                removeFixedNetworkIds:
                  $ref: '#/components/schemas/FixedNetworkIds'
                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
                required:
                  - edgeDiscoveryServiceEndPoint
```

```
           - lcmServiceEndPoint
          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/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'

    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/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}/zones:
    post:
      summary: Originating OP informs partner OP that it is willing to access the specified zones  and partner OP
shall reserve compute and network resources for these 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/ZoneRegisterationRequestData'
        required: true
      responses:
        '200':
          description: Zone registered successfully
          content:
            application/json:
              schema:
                $ref:  '#/components/schemas/ZoneRegisterationResponseData'
        '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:
          $ref: '#/components/responses/default'
      callbacks:
        onZoneResourceUpdateEvent:
          '{$request.body#/availZoneNotifLink}':
            post:
              parameters:
```

```
 - name: federationContextId
  in: path
  required: true
  schema:
   $ref:  '#/components/schemas/FederationContextId'
 - name: zoneId
  in: path
  required: true
  schema:
   $ref:  '#/components/schemas/ZoneIdentifier'
requestBody:
 description: Notification about resource availability.
 content:
  application/json:
   schema:
    type: object
    required:
     - zoneResUpdInfo
    properties:
     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 thoughput that this edge can offer. It is defined in Mbps.
          dedicatedNIC:
            type: integer
            format: int32
          supportSriov:
            type: boolean
            description: If this zone support SRIOV networks or not
          supportDPDK:
            type: boolean
            description: If this 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'

  /{federationContextId}/zones/{zoneId}:
    delete:
      summary: Asservate usage of  a partner OP zone. Originating OP informs partner OP that it will no longer access the specified 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/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'
    get:
      summary: Retrieves details about the computation and network resources that partner OP has reserved for this zone.
      operationId: GetZoneData
      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:
           type: object
           required:
            - zoneId
            - reservedResources
            - resourceQuotaLimits
            - 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
              supportSriov:
                  type: boolean
                  description: If this zone support SRIOV networks or not
                supportDPDK:
                  type: boolean
                  description: If this zone supports DPDK based networking.

        '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}/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. App 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
                - artefactFileName
                - artefactFileFormat
                - artefactDescriptorType
              oneOf:
                - required:
```

```
          - artefactRepoLocation
        - required:
          - componentSpec
        - required:
          - artefactFile
      properties:
        artefactId:
          $ref:  '#/components/schemas/ArtefactId'
        appProviderId:
          $ref:  '#/components/schemas/AppProviderId'
        artefactName:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
          description: Name of the artefact.
        artefactVersionInfo:
          type: string
          pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
          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:
            - WINZIP
            - 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.  App provider can either define either a Helm
chart or a Terraform script or container spec.
        artefactRepoLocation:
          $ref: '#/components/schemas/ObjectRepoLocation'
        componentSpec:
          description: Details about compute, networking and storage requirements for each component of the
application.
                             App 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: array
         items:
           $ref: '#/components/schemas/ComponentSpec'
         minItems: 1
        artefactFile:
         type: string
         format: binary
         description: Helm archive or Terraform archive or container spec file.
    required: true
   responses:
    '200':
     description: Artefact uploaded successfully
    '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}/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
          - artefactFileName
          - artefactFileFormat
          - artefactDescriptorType
          - artefactRepoLocation
        properties:
          artefactId:
            $ref:  '#/components/schemas/ArtefactId'
          appProviderId:
            $ref:  '#/components/schemas/AppProviderId'
          artefactName:
            type: string
            pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
            description: Name of the artefact.
          artefactDescription:
            type: string
            maxLength: 256
            description: Brief description of the artefact by the application provider
          artefactVersionInfo:
            type: string
            pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
            description: Artefact version information
          artefactVirtType:
            type: string
            enum:
              - VM_TYPE
              - CONTAINER_TYPE
          artefactFileName:
            type: string
            minLength:  8
            maxLength: 32
            description: Name of the file.
          artefactFileFormat:
            type: string
            enum:
              - WINZIP
              - 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.  App provider can either define either a Helm
chart or a Terraform script or container spec.
          artefactRepoLocation:
            type: object
            required:
              - repoType
```

```
                      - repoURL
                    properties:
                      repotype:
                        type:     string
                        enum:
                          - PRIVATEREPO
                          - PUBLICREPO
                          - UPLOAD
                        description: Artefact repository location. PUBLICREPO is used of public URLs like GitHub, Helm
repo etc,
                                          PRIVATEREPO  is used for private repo managed by the application developer,
                                          UPLOAD is for the case when artefact is uploaded from MEC web portal.
                      repoURL:
                        $ref: '#/components/schemas/Uri'
                      userName:
                        type: string
                        pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
                        description:  Username to access the artefact repository
                      password:
                        type: string
                        description:  Password to access the artefact repository
                      token:
                        type: string
                        description:  Authorization token to access the artefact repository
        '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'

    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
    '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}/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
       oneOf:
        - required:
          - file
        - required:
          - repolocation
       properties:
        fileId:
         $ref:  '#/components/schemas/FileId'
        appProviderId:
         $ref:  '#/components/schemas/AppProviderId'
        fileName:
          type: string
          pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
          description: Name of the image file.   App  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
          pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
          description: File version information
        fileType:
          type: string
          enum:
            - QCOW2
            - DOCKER
          description: Indicate if the file is Container image or VM image (QCOW2)
        imgOSType:
          type: string
          enum:
            - LINUX
            - WINDOWS
            - SOLARIS
          description: Base OS for the image. Currently only “Linux” is supported
        imgInsSetArch:
          type: string
          enum:
            - X86_64
            - ARM64
        repolocation:
          type: object
          required:
            - repoType
            - repoURL
          properties:
            repotype:
              type:     string
              enum:
                - PRIVATEREPO
                - PUBLICREPO
              description: Artefact repository location. PUBLICREPO is used of public URLs like GitHub, Helm repo etc,
                           PRIVATEREPO  is used for private repo managed by the application developer,
                           UPLOAD is for the case when artefact is uploaded from MEC web portal.
            repoURL:
              $ref: '#/components/schemas/Uri'
            userName:
              type: string
              pattern: ^[A-Za-z][A-Za-z0-9_]{7,63}$
              description:  Username to access the artefact repository
            password:
              type: string
              description:  Password to access the artefact repository
            token:
              type: string
```

```
                      description:  Authorization token access the artefact repository

                file:
                  type: string
                  format: binary
                  description: Binary image associated with an application component.
        required: true
      responses:
        '200':
          description: File uploaded successfully
        '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}/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
        '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'


    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
             oneOf:
               - required:
                 - file
               - required:
                 - repolocation
             properties:
               fileId:
                 $ref:  '#/components/schemas/FileId'
               appProviderId:
                 $ref:  '#/components/schemas/AppProviderId'
               fileName:
                 type: string
                 pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
                 description: Name of the image file.   App  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
                 pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
                 description: File version information
               fileType:
                 type: string
                 enum:
                   - QCOW2
                   - DOCKER
                 description: Indicate if the file is Container image or VM image (QCOW2)
               imgOSType:
                 type: string
                 enum:
                   - LINUX
                   - WINDOWS
                   - SOLARIS
                 description: Base OS for the image. Currently only “Linux” is supported
               imgInsSetArch:
                 type: string
                 enum:
                   - X86_64
                   - ARM64

        '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}/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 compute resource requirements, associated artefacts, QoS profile and regions where application shall be made available etc.
    content:
     application/json:
      schema:
       type: object
       required:
        - appId
        - appProviderId
        - appDeploymentZones
        - appMetaData
        - appQoSProfile
        - appComponentSpecs
        - appStatusCallbackLink
       properties:
        appId:
         $ref:  '#/components/schemas/AppIdentifier'
        appProviderId:
         $ref:  '#/components/schemas/AppProviderId'
        appDeploymentZones:
         description: Details about partner OP zones where the application should be made available
         type: array
         items:
          type: object
          required:
           - countryCode
           - zoneInfo
          properties:
           countryCode:
            $ref:  '#/components/schemas/CountryCode'
           zoneInfo:
            $ref:  '#/components/schemas/ZoneIdentifier'
         minItems: 1
        appMetaData:
         description: Application metadata details
         type: object
         required:
          - appName
          - version
          - accessToken
         properties:
          appName:
           type: string
           pattern: ^[A-Za-z][A-Za-z0-9_]{7,31}$
           description: Name of the application.   Application provider define a human readable name for the application
          version:
           type: string
           pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
           description: Version info of the application
          appDescription:
           type: string
           minLength: 16
           maxLength: 256
           description: Brief application description provided by application provider
          mobilitySupport:
           type: boolean
           default: false
           description: Indicates if an application is sensitive to user mobility and can be relocated. Default is
“FALSE”
          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
            - SURVELIANCE
            - ENTERTAINMENT
            - CONNECTIVITY
            - PRODUCTIVITY
            - SECURITY
            - IDUSTRIAL
            - 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:
           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
          multiUserClients:
           type: string
           enum:
            - APP_TYPE_SINGLE_USER
            - APP_TYPE_MULTI_USER
           default: APP_TYPE_SINGLE_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
            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.  App providers are required to provide details about all these
components, their associated descriptors and their
                                  DNS names.
              type: array
              items:
                type: object
                required:
                  - serviceNameNB
                  - componentname
                  - artefactId
                properties:
                  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. 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.   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
                  artefactId:
                    $ref:  '#/components/schemas/ArtefactId'
              minItems: 1
            appStatusCallbackLink:
              $ref: '#/components/schemas/Uri'
      responses:
        '202':
          description: Application onboarded request accepted
        '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:
      $ref: '#/components/responses/default'
   callbacks:
    onApplicationOnboardStatusEvent:
     '{$request.body#/appStatusCallbackLink}':
      post:
       parameters:
        -  name: federationContextIdentifier
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
       requestBody:
        description: Notification payload.
        content:
         application/json:
          schema:
           type: object
           required:
            - appId
            - statusInfo
           properties:
            appId:
             $ref:  '#/components/schemas/AppIdentifier'
            statusInfo:
             type: array
             items:
              type: object
              required:
               - zoneId
               - onboardStatusInfo
              properties:
               zoneId:
                $ref: '#/components/schemas/ZoneIdentifier'
               onboardStatusInfo:
                description: Defines change in application status. This change could be related to application itself or an application instance status
                type: string
                enum:
                 - PENDING
                 - ONBOARDED
                 - DEBOARDING
                 - REMOVED
                 - FAILED
             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'

  /{federationContextId}/application/onboarding/app/{appId}:
    patch:
      summary: Updates partner OP about changes in  application compute resource requirements, QOS 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 compute resource requirements, associated artefact and QOS 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:
                      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
                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.  App 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:
            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. 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.   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
           artefactId:
            $ref:  '#/components/schemas/ArtefactId'
         minItems: 1
   responses:
    '202':
     description: Application update request accepted
    '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'

  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 deboard request accepted
          content:
            application/json:
              schema:
                type: object
                required:
                  - appId
                  - appProviderId
                  - appDeploymentZones
                  - appMetaData
                  - appQoSProfile
                  - appComponentSpecs
                properties:
                  appId:
                    $ref:  '#/components/schemas/AppIdentifier'
                  appProviderId:
                    $ref:  '#/components/schemas/AppProviderId'
                  appDeploymentZones:
                    description: Details about partner OP zones where the application should be made available
                    type: array
                    items:
                      type: object
                      required:
                        - countryCode
                        - zoneInfo
                      properties:
                        countryCode:
                          $ref:  '#/components/schemas/CountryCode'
                        zoneInfo:
                          type: object
                          required:
                            - zoneId
                            - flavourId
                          properties:
                            zoneId:
                              $ref:  '#/components/schemas/ZoneIdentifier'
                            flavourId:
                              type: integer
                              format: int32
                              description: Flavour identifier. Should corresponds to flavours indicated by Partner OP for the
corresponding zone.
                            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.
App provider can pre-reserve
                                                   a pool of compute resources on each 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 zone.  Valid only if IE
'resourceConsumption' is set to
                                                  'RESERVED_RES_SHALL' or 'RESERVED_RES_PREFER'
                    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
                        pattern: ^(\d{1,2}\.)?(\d{1,2}\.)?(\d{1,2})$
                        description: Version info of the application
                      appDescription:
                        type: string
                        minLength: 16
                        maxLength: 256
                        description: Brief application description provided by application provider
                      mobilitySupport:
                        type: boolean
                        default: false
                        description: Indicates if an application is sensitive to user mobility and can be relocated. Default is
"FALSE"
                      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
                          - SURVELIANCE
                          - ENTERTAINMENT
                          - CONNECTIVITY
                          - PRODUCTIVITY
                          - SECURITY
                          - IDUSTRIAL
```

```
             - 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:
            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
           multiUserClients:
            type: string
            enum:
             - APP_TYPE_SINGLE_USER
             - APP_TYPE_MULTI_USER
            default: APP_TYPE_SINGLE_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
         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.  App providers are required to provide details about all these
components, their associated descriptors and their
                    DNS names.
          type: array
          items:
           type: object
           required:
            - serviceNameNB
            - componentname
            - artefactId
           properties:
            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. 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.   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
            artefactId:
             $ref:  '#/components/schemas/ArtefactId'
          minItems: 1

    '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}/application/onboarding/app/{appId}/zone/{zoneId}:
  delete:
   summary: Deboards an application from partner OP 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:
        '202':
          description: Application deboard request accepted
        '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}/application/onboarding/app/{appId}/additionalZones:
    post:
      summary: Onboards an existing application to a new 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 zones where application shall be made available
        content:
          application/json:
            schema:
              type: array
              items:
                  $ref:  '#/components/schemas/ZoneIdentifier'
              minItems: 1
      responses:
        '202':
          description: Application onboarding request accepted
        '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}/application/onboarding/app/{appId}/zoneForbid:
    post:
      summary: Forbid/allow application instantiation on a partner 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 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 zone.  No new instance of the
application can be created on this 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/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}/application/lcm:
    post:
      summary: Instantiates an application on a partner OP zone.
      operationId: InstallApp
      tags:
        - ApplicationDeploymentManagement
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
      requestBody:
        description: Details about application and zones where application instance should be created. It also define
a 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:
                  $ref: '#/components/schemas/Version'
                appProviderId:
                  $ref: '#/components/schemas/AppProviderId'
                zoneInfo:
                  type: object
                  required:
                    - zoneId
          - flavourId
         properties:
          zoneId:
           $ref:  '#/components/schemas/ZoneIdentifier'
          flavourId:
           type: integer
           format: int32
           description: flavour identifier. Should corresponds to flavours indicated by Partner OP for the
corresponding zone.
          resourceConsumption:
           type: string
           enum:
            - RESERVED_RES_SHALL
            - RESERVED_RES_PREFER
            - RESERVED_RES_AVOID
            - RESERVED_RES_FORBID
           default: RESERVED_RES_AVOID
           description: Specifies if the application can be instantiated using  pre-reserved resource or not.
App provider can pre-reserve
                     a pool of compute resource on each zone.  'RESERVED_RES_SHALL' instruct OP to
use only the pre-reserved resources.
                     'RESERVED_RES_PREFER' instruct to first try using  pre-reserved resource, if none
available go for non-reserved resources.
                     'RESERVED_RES_AVOID' instruct OP  not to use pre-reserved resource if possible, it
is a choice depending upon circumstances
                     'RESERVED_RES_FORBID' instruct OP not to use pre-reserved resources.
          resPool:
           type: string
           pattern:  ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
           description: Resource pool to be used for application instantiation on this 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/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:
      $ref: '#/components/responses/default'
   callbacks:
    onInstanceStatusEvent:
     '{$request.body#/appInstCallbackLink}':
      post:
       parameters:
        -   name: federationContextIdentifier
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
       requestBody:
        description: Notification payload.
        content:
         application/json:
          schema:
           type: object
           required:
            -  appId
            - appInstIdentifier
            - zoneId
            - appInstanceInfo
           properties:
            appId:
             $ref: '#/components/schemas/AppIdentifier'
            appInstIdentifier:
             $ref: '#/components/schemas/InstanceIdentifier'
            zoneId:
              $ref: '#/components/schemas/ZoneIdentifier'
            appInstanceInfo:
             type: object
             properties:
              appInstanceState:
               type: string
               enum:
                - PENDING
                - READY
                - FAILED
                - TERMINATING
               description: Running status of the application instance.
              accesspointInfo:
               description: Information about the IP and Port exposed by the OP. Application clients shall use these access points to reach this application instance
               type: array
               items:
                type: object
                required:
                 - interfaceId
                 - accessPoints
                properties:
                 interfaceId:
                  type: string
                  pattern: ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                  description:  This is the interface Identifier that app 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'

 /{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 app 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/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'

  delete:
   summary: Terminate an application instance on a partner OP 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/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}/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/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}/isv/resource/zone/{zoneId}/appProvider/{appProviderId}:
  post:
    summary: Reserves resources (compute, network and storage)  on a partner OP zone.   ISVs registered with home OP reserves resources on a partner OP 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:
          $ref:  '#/components/schemas/AppProviderId'
    requestBody:
      content:
        application/json:
          schema:
            type: object
            required:
              - poolName
              - resRequest
              - resourceReservationCallbackLink
            properties:
              poolName:
                type: string
                pattern:  ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                description: ISV defined name of the resource pool.
              resRequest:
                description: Compute flavours to be reserved and their time duration
                type: object
                required:
                  - flavours
                  - reserveDuration
                properties:
                  flavours:
                    type: array
                    items:
                      type: object
                      required:
                        - flavourId
                        - numFlavour
                      properties:
                        flavourId:
                          type: integer
                          format: int32
                          description: Flavour identifier. Should corresponds to flavours indicated by Partner OP for the
corresponding zone.
                        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
        properties:
         poolId:
          $ref: '#/components/schemas/PoolId'

    '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:
      $ref: '#/components/responses/default'
   callbacks:
    onResourceStatusChangeEvent:
     '{$request.body#/resourceReservationCallbackLink}':
      post:
       parameters:
        -   name: federationContextIdentifier
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
       requestBody:
        description: Notification payload.
        content:
         application/json:
          schema:
           type: object
           required:
            - appProviderId
            - zoneId
            - poolId
            - grantedFlavours
           properties:
            appProviderId:
             $ref: '#/components/schemas/AppProviderId'
            zoneId:
             $ref: '#/components/schemas/ZoneIdentifier'
            poolId:
             $ref: '#/components/schemas/PoolId'
            grantedFlavours:
             type: array
             items:
              type:  object
              required:
               - flavourId
               - numFlavour
              properties:
               flavourId:
                type: integer
                format: int32
                description: An identifier to refer to this combination of compute resources.
               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'



  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:
            $ref: '#/components/schemas/AppProviderId'
    responses:
      '200':
        description: Reserved Resources Details
        content:
         application/json:
          schema:
            type: object
            required:
              - poolName
              - reservedPoolId
              - reservedFlavours
            properties:
              poolName:
                type: string
                pattern:  ^[A-Za-z0-9][A-Za-z0-9_]{6,30}[A-Za-z0-9]$
                description: ISV defined name of the resource pool.
              reservedPoolId:
                $ref: '#/components/schemas/PoolId'
              reservedFlavours:
                type: array
                items:
                  type: object
                  required:
                    - flavourId
                    - count
                  properties:
                    flavourId:
                      $ref: '#/components/schemas/Flavour'
                    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/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}/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:
          type: integer
          format: int32
          description: An identifier to refer to this combination of compute resources.
         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/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'

  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 zones
                       where an application instance can be created. Partner OP applies a set of filtering criteria’s to
select candidate 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 zones
          content:
           application/json:
            schema:
              $ref: '#/components/schemas/DiscoveredEdgeNodes'
        '400':
          $ref: '#/components/responses/400'
        '401':
          $ref: '#/components/responses/401'
        '404':
          $ref: '#/components/responses/404'
        '422':
          $ref: '#/components/responses/422'
        '500':
          $ref: '#/components/responses/500'
        default:
            $ref: '#/components/responses/default'


  /{federationContextId}/roaminguserauth/device/{deviceId}/token/{authToken}:
    get:
      summary: Validates the authenticity of a roaming user from home OP
      operationId: AuthenticateDevice
      tags:
        - LBORoamingAuthentication
      parameters:
        - name: federationContextId
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/FederationContextId'
        - name: deviceId
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/DeviceId'
        - name: authToken
          in: path
          required: true
          schema:
            $ref:  '#/components/schemas/AuthorizationToken'
      responses:
        '200':
           description: Device Auth Token validated
        '401':
            $ref: '#/components/responses/401'
        '404':
```

```
    $ref: '#/components/responses/404'
  '422':
    $ref: '#/components/responses/422'
  '500':
    $ref: '#/components/responses/500'
  '503':
    $ref: '#/components/responses/503'
  default:
    $ref: '#/components/responses/default'
```

Annex B Document Management

B.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

B.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.