Open Gateway NBI APIs Realisation in the SBI

Version 1.0

16 February 2024

Security Classification: Non-Confidential

Copyright Notice

Disclaimer

Compliance Notice

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

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

Table of Contents

1 Introduction

1.1 Overview

The idea of this document is creating a baseline for the realisation of Open Gateways APIs at lower levels in the OP architecture. This way partners and operators can use it as a reference implementation covering aspects such as charging, federation, gaps and recommended enablers to fill those service gaps.

1.2 Scope

This document provides guidelines to operators on how service APIs should be implemented in the network identifying missing interfaces or enablers and aligning SDOs and the wider industry to the needs for service development.

1.3 Definitions

Term	Description
CAMARA	CAMARA is an open-source project within Linux Foundation to define, develop and test the APIs

1.4 Abbreviations

Term	Description
3GPP	3rd Generation Partnership Project
AS	Application Server
API	Application Programming Interface
BSS	Business Support System
CSP	Communication Service Provider
EWBI	East/Westbound Interface
HR	Home Routed
HSS	Home Subscriber Server
IP	Internet Protocol
IPX	Internetwork Packet Exchange
MSISDN	Mobile Station International Subscriber Directory Number
NEF	Network Exposure Function
OB	Operating Business
OP	Operator Platform
OTP	One-Time Password
PDN	Packet Data Network
PRD	Permanent Reference Document
PDU	Packet Data Unit
QoD	Quality on Demand
QoS	Quality of Service
SBI-NR	SouthBound Interface – Network Resources

Term	Description
SCEF	Service Capability Exposure Function
SDO	Standard Developing Organisation
SIM	Subscriber Identity Module
SMS	Short Message Service
TCP	Transmission Control Protocol
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
URL	Uniform Resource Locator

1.5 References

Ref	Doc Number	Title
[1]	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
[2]	RFC 8174	Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words https://www.rfc-editor.org/info/rfc8174
[3]		CAMARA UE Identification https://github.com/camaraproject/Commonalities/blob/main/documentation/UE-Identification.md
[4]		CAMARA API "CarrierBilling" https://github.com/camaraproject/CarrierBillingCheckOut
[5]	GSMA PRD OPG.07	Southbound Interface Charging Function APIs
[6]		CAMARA API "Location Retrieval" https://github.com/camaraproject/DeviceLocation/blob/main/codeAPI_definitions/location-retrieval.yaml
[7]	GSMA PRD OPG.03	Southbound Interface Network Resources APIs, Version 2.0
[8]		CAMARA API "Number Verification" https://github.com/camaraproject/NumberVerification/blob/main/code/API_definitions/number_verification.yaml
[9]		CAMARA API "Quality on Demand" https://github.com/camaraproject/QualityOnDemand/tree/release-0.10.0
[10]	3GPP TS 29.214	Policy and Charging Control over Rx reference point

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 RFC 2119 [1] and clarified by RFC8174 [2], when, and only when, they appear in all capitals, as shown here."

2 Northbound CAMARA APIs Analysis

2.1 UE identification

Different solutions needed by several CAMARA APIs for the API invoker to identify the UE are described by CAMARA Commonalities group in [3].
To identify the UE addressed by a CAMARA API, an application provider should be able to use:

- a GPSI in any of the possible existing formats
- Optionally using the UE IP address or Ethernet MAC address (together with the application provider Identifier)

2.2 CarrierBilling API

The version of CAMARA CarrierBilling API which is considered in this document is ver. 0.2.0 [4].

2.2.1 Description

As defined in the CAMARA Carrier Billing Check Out sub project, this API allows clients to request the payment of a (set of) good(s)/service(s), as well as to retrieve information about a specific payment or a list of payments.

In summary, this API provides the customer with the ability to:

- Trigger carrier billing payment request (in one or two steps)
- Follow up of payment processing using as Payment Method Carrier Billing, i.e.: the operator performs the billing of the goods.

NOTE: The scope of this API family is limited to 4G and 5G at a first stage.

Required information for using the CarrierBilling API [4]:

- **Carrier Billing:** An online payment process which allows users to make purchases by charging payments against an Operating Business's (OB) Billing Systems, accordingly to the user's configuration in the OB. In a common usage in the industry, the payment is processed on current account balance or charged on next bill generated for the line.
- **Payment:** The process of paying for a (set of) good(s)/service(s).
- **1-STEP Payment:** Payment process performed in one phase (i.e. one action), that involves all the OB Carrier Billing Systems checking and trigger the charging request against billing systems.
- **2-STEP Payment:** Payment process performed in two phases (i.e. two actions). First action deals with payment preparation request to guarantee the reservation of the involved amount. Second action is an explicit confirmation or cancellation of the

payment by the user. Any payment not confirmed/cancelled by a given user is discarded after some time in order to avoid inconsistency in the billing systems.

API functionality

This API allows to clients to request the payment of a (set of) good(s)/service(s), as well as to retrieve information about a specific payment or a list of payments.

The API provides several endpoints/operations:

- An endpoint to request the payment in one step.
- A set of endpoints to request the payment in two steps:
    - One endpoint to setup the payment reservation.
    - A couple of endpoints to confirm/cancel such payment reservation.
- A set of endpoints to retrieve information about a list of payments or a specific payment, identified by its specific `paymentId`.
- An endpoint where the API Server can send notifications about a payment procedure, towards the `webhook.notificationUrl` when provided by API client.

2.2.2 SBI Realisation

No standardised SBI is available for the OP to serve the Carrier Billing API. The OP shall interpret the Carrier Billing API requests and transform them to proprietary request(s) to the operator's Business Support System (BSS).

2.2.3 Charging Aspects

The API can be classified as a "network capabilities exposure services: without impact on device’s data usage". Possible options for charging for this category are detailed in GSMA PRD OPG.07 [5].

2.2.4 Roaming (Home Routed scenario) Aspect

The API will always be addressed to the home network, as such no roaming aspects need to be considered.

2.2.5 Federation Aspects

It is assumed that the merchant has a relation with a single operator. The request can be forwarded to the home operator when the subscriber who does the transaction does not belong to the prime operator.

2.3 Location Retrieval API

The version of CAMARA Location Retrieval API which is considered in this document is ver. 0.1.0 [6].

2.3.1 Description

With this API, customers can retrieve the area where a certain user device is localised. The area provided in the response could be described:

- by a circle determined by coordinates (latitude and longitude) and a radius.

- by a simple polygon delimited by segments connecting consecutively an array of coordinates (points). The last point connects to the first point to delimit a closed shape bounded with straight sides.

The retrieved shape depends on the network conditions at the subscriber's location and any of the supported shapes could be received.

The requester could optionally ask for a freshness of the localisation information by providing a maxAge ("I want a location not older than 600 seconds").

The result accuracy depends on the network's ability and accuracy to locate the device.

Additionally, to location information, the answer will also provide indication about the location time.

Note: Location is in most jurisdictions considered to be sensitive data and thereby consent by device owner/user must be verified before providing it to the developer.

Relevant terms and definitions

- **Device**: A device refers to any physical entity that can connect to a network and participate in network communication.
- **Area**: It specifies the geographical surface where a device may be physically located.
- **Max Age**: Maximum age of the location information which is accepted for the location retrieval (in seconds).
- **Last Location Time**: Last date and time when the device was localised.

API Functionality

The API exposes a single endpoint/operation:

- `/retrieve`: Retrieve where the device is localised. The operation returns:
  - a localisation defined with a circle with center specified by the latitude and longitude, and radius for answer accuracy,
  - a timestamp about location information freshness.

2.3.2 SBI Realisation

The CAMARA Location Retrieval API leverages 3GPP Monitoring Event (monitoringType: LOCATION_REPORTING) as mentioned in section 2.5 of OPG.03 [7].

Request:

CAMARA POST /retrieve	→	3GPP Monitoring Event POST //subscriptions

<pre>{
  "device": {
  "phoneNumber":
"33699901036"
  },
  "maxAge": 60
}</pre>
		<pre>{
		  "supportedFeatures": "4FF",
		  "msisdn": "33699901036",
		  "notificationDestination":
		"https://webhook.site/10a41b89-0d66-4bd0-
		8950-
		cfe8cd357eb9?cbInternet=true&cbEIN=false",
		  "monitoringType": "LOCATION_REPORTING",
		  "accuracy": "ENODEB",
		  "minimumReportInterval": "60",
		  "maximumNumberOfReports": 1,
		  "maximumResponseTime": 2,
		  "locationAccuracyGranularity" : "1",
		  "locationType": "LAST_KNOWN_LOCATION"
		}</pre>

Response:

The Monitoring event response is the following:

```
{
  "msisdn": "33699901036",
  "locationInfo": {
    "ageOfLocationInfo": 1,
    "cellId": "208-85-77000",
    "enodeBId": "208-85-1-300-1000",
    "trackingAreaId": "208-85-1000"
  },
  "monitoringType": "LOCATION_REPORTING",
  "locationType": " LAST_KNOWN_LOCATION "
}
```

With this answer the CAMARA service must:

1. Check if the ageOfLocationInfo is smaller than the maxAge provided in the request (if the location is too old a specific error must be send back)
2. Map cellId or enodeBid or trackingAreaId to an Operator's own geographic network reference to get either a circle or a polygon.

Once done, the CAMARA response will be:

```
  "lastLocationTime": "1",
  "area": {
    "areaType": "Circle",
    "center": {
      "latitude": 48.801009,
      "longitude": 2.29498
    },
    "radius": 1000.0
  }
}
```

2.3.3 Charging Aspects

The Location Retrieval API is classified as a "network capabilities exposure services: without impact on device's data usage". Possible options for charging for this category are detailed in GSMA PRD OPG.07 [5].

2.3.4 Roaming (Home Routed scenario) Aspect

The API will always be addressed to the home network, as such no roaming aspects need to be considered.

2.3.5 Federation Aspects

It is assumed that the API invoker has a relation with a single operator. The request can be forwarded to the home operator when the subscriber who does the transaction does not belong to the prime operator.

2.4 Number Verification API

The version of the CAMARA Number Verification API which is considered in this document is ver. 0.3.0 [8].

2.4.1 Description

As defined in CAMARA Number Verification subproject, this API allows clients to verify that a provided "mobile phone number" is the one used in the device. It verifies that the user is using a device with the same "mobile phone number" as is associated with the SIM used in the device connected to the mobile data network.

It also makes it possible for a service provider to verify the number whereby the phone number associated to the authenticated user's phone number is returned (enabling verification to be done by the invoker of the API / application service provider).

In summary, this API provides the customer / client with the ability to:

- verify in real time the validity of the phone number of the mobile device being used to access their service. The service is offered without requiring user interaction (e.g. no OTPs (One-Time Passwords) received by SMS nor authenticator application downloads etc).
- enhance their security checks (verified number increases the likehood that number is not spoofed / cloned).

NOTE: The scope of this API family is limited to 4G and 5G at a first stage.

Required information for using the Number Verification API [8]:

- **OAuth Authorisation Code Grant**: Number Verification API uses the standard OAuth auth code grant flow. Both offered end points use 3-legged OAuth authentication flows.
- **Network-Based Authentication**: authentication mechanisms (of the device / UE) are based on the identification of the endpoint of a network connection (e.g. the mobile phone number associated to a specific mobile network connection / IP address). Network-Based Authentication is implementation specific, however one possible

example of an implementations is an “IP translation mechanism” whereby the received IP address is translated into information known to and stored by the CSP (Communication Service Provider) related to the UE’s SIM Another possible example is the use of tokens (used in the authorisation flow) generated by the UE to the CSPs network that is used to retrieve information related to the UE’s SIM.

- **PDU/PDN session:** An active PDU/PDN session is required to access this service. The mobile device requesting the service shall do so over a mobile network (Wi-Fi connections are out of this API’s scope).

API functionality

The API provides two endpoints/operations:

- **Phone Number Verify**: checks if the user mobile phone number matches the phone number associated with the mobile device. It can receive either a hashed or a plain text phone number as input and it compares the received input with the authenticated user's phone number associated to the access token to respond true/false.
- **Phone Number Share**: retrieves the phone number associated to the user's token and returns it so the verification can be made by the API invoker.

2.4.2 SBI Realisation

The OP shall interpret the CAMARA Number Verification API requests and redirects it to SBI when possible. For the network side, relevant interfaces are implementation specific.

2.4.3 Charging Aspects

The Number Verify API is classified as a "network capabilities exposure services: without impact on device’s data usage". Possible options for charging for this category are detailed in GSMA PRD OPG.07 [5].

2.4.4 Roaming (Home Routed scenario) Aspect

The API will always be addressed to the home network, as such no roaming aspects need to be considered.

2.4.5 Federation Aspects

It is assumed that the API invoker has a relation with a single operator. The request can be forwarded to the home operator when the subscriber who does the transaction does not belong to the prime operator.

2.5 QualityOnDemand API

The version of the CAMARA Quality-On-Demand API which is considered in this document is ver. 0.10.0 [9].

2.5.1 Description

As defined in CAMARA Quality-On-Demand (QoD), this API allows application developers and other users (capability consumers) to request a specific packet delay budget[1] or throughput from the mobile networks without the necessity to have an in-depth knowledge of the 4G/5G system or the overall complexity of the Telecom Systems.

In summary, this API provides the customer with the ability to:

- Set quality for a mobile connection (e.g. required packet delay (latency), throughput).
- Get a notification if the network cannot fulfil the request.

NOTE: The scope of this API family is limited to 4G and 5G at a first stage.

Required information for using the QoD API [9]:

- **QoD service endpoint**: The Uniform Resource Locator (URL) pointing to the RESTful resource of the QoD API.
- **Authentication**: security access keys such as OAuth 2.0 client credentials used by client applications to invoke the QoD API.
- **QoS profiles and QoS profile labels:** Latency or throughput requirements of the application mapped to relevant QoS profile class.
- **Identifier for the User Equipment (UE):** At least one identifier for the user equipment out of four options: IPv4 address, IPv6 address, MSISDN, or external ID assigned by the mobile network operator for the user equipment.
- **Identifier for the Application Server (AS):** IPv4 and/or IPv6 address of the application server (application backend)
- **Application data Flow** (between the application client and application server): The precise application data flow the developer wants to prioritise and have stable latency or throughput for. This flow is in the current API version determined by the identifiers used for the user equipment and the application server. And it can be further elaborated with details such as ports or port ranges. Future versions of the API might allow more detailed flow identification features.
- **Duration:** Duration (in seconds) for which the QoS session (between application client and application server) should be created. This parameter is optional. When not specified, a default session duration (e.g. 24 hours) is applied. The user may request a termination before its expiration.
- **Notification URL and token:** Developers may provide a callback URL on which notifications (e.g. session termination) regarding the session can be received from the service provider. This is an optional parameter.

[1] The packet delay budget is the maximum allowable one-way latency between the customer's device and the gateway from the operator's network to other networks. By limiting the delay, the network can provide an acceptable level of performance for various services, such as voice calls, video streaming, and data. The end-to-end or round trip latency will be about two times this value plus the latency not controlled by the operator

API functionality

The usage of the CAMARA QoD API is based on QoS profile classes and parameters which define application data flows.

The QoS session resources can be created, queried, and deleted. The expectation is that once the QoS profile class is requested, application users get a prioritised service with requested latency or throughput, even in the case of congestion.

How QoS profiles are mapped to connectivity characteristics is subject to agreements between the operators and the Application Providers/developers.

2.5.2 SBI Realisation

The OP shall interpret the CAMARA QoD API requests and redirects it to the Service Capability Exposure Function (SCEF) or Network Exposure Function (NEF) via Southbound Interface – Network Resources (SBI-NR) as defined in GSMA PRD OPG.03 section 2.2 [7] to realise it when a request is accepted.

The below flow sequence in Figure 1 details this functionality from a high-level point of view.

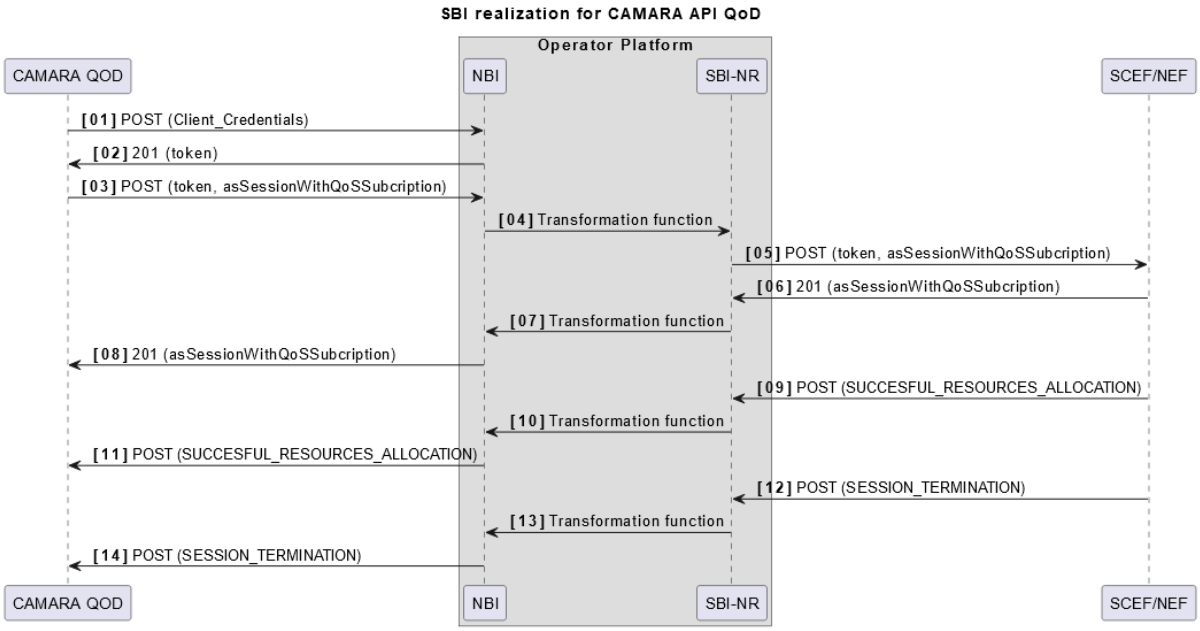

Figure 1: High level flow sequence of QoD API

The realisation of the QoD request through the Operator Platform requires a transformation function. Hereafter, a simple case of mapping is detailed.

The simple case consists of a QoD API request where the IP addresses and ports represents scalar values.

- IP address with no mask or prefix
- Port represents scalar value, i.e. not a range or list

An example of this request (noted 3 in the flow in Figure 1) is the one below.

```
      {
         "duration": 60,
         "device": {
           "ipv4Address": {
             "publicAddress": "84.125.93.10",
             "publicPort": 59765

           }
         },
         "applicationServer": {
           "ipv4Address": "172.20.120.84"
         },
         "devicePorts": {
           "ports": [
             50984
           ]
         },
         "applicationServerPorts": {
           "ports": [
             10000
           ]
         },
         “qosProfile”: “middle”,
         "webhook": {
           "notificationUrl": "https://webhook.site/27284495-859a-
      48bc-ba2a-9a3cc013b784"
         }
}
```


The corresponding SCEF/NEF compliant request is:


```
      {
         "supportedFeatures": "0",
         "notificationDestination":  "https://webhook.site/c1ae90f8-
      b7b6-4185-a848-2d1f63a08ff7",
         "flowInfo": [
           {
              "flowId": 17,
             "flowDescriptions": [
               "permit  in  “ip”  from  172.20.120.105  55289  to
      172.20.120.84 10000",
               "permit  out  “ip”  from  172.20.120.84  10000  to
      172.20.120.105 55289"
              ]
           }
         ],
         "qosReference": "b55e2cc8-b386-4d90-9f95-b98ba20be050",
         "ueIpv4Addr": "172.20.120.105",
         "usageThreshold": {
           "duration": 60,
           "totalVolume": 0,
           "downlinkVolume": 0,
           "uplinkVolume": 0
         },
      }
```


Comparing these two requests we can see that we have different mapping problems to solve:

- The CAMARA QoD API does not allow indicating the transport protocol. However, "ip" can be used as key word used in the protocol. In this case, the port(s) are used to describe the port(s) of any protocol if available ([10] section 5.3.8).
- qosReference must be pre-existent in the system
- the qosProfile field must be transformed into the qosReference that corresponds to the expected behaviour
- this qosReference must match the flowId injected in flowInfo according to the rules of T8 API
- supportedFeatures must be set to “0” as there is no input for it
- notificationDestination must be set to an url that contains original notificationUrl and notificationAuthToken to be able to transform and forward event coming from SCEF if necessary (if we have a notificationUrl in the request) or alternatively to a proxy

The reverse mapping could be used to map response from SCEF :

```
{
   "self":     "https://{baseURL}/3gpp-as-session-with-qos/v1/
/subscriptions/<token>"
   "supportedFeatures": "0",
   "notificationDestination":  "https://{baseURL}notifications
",
   "flowInfo": [
      {
         "flowId": 1,
         "flowDescriptions": [
          "permit  in  “ip”  from  172.20.120.105  50984  to
172.20.120.84 10000",
          "permit  out  “ip”  from  172.20.120.84  10000  to
172.20.120.105 50984"
         ]
      }
   ],
   "qosReference": "173047be-d805-4df9-8cd7-9cb0b78cbc2b",
   "ueIpv4Addr": "172.20.120.105",
   "status": "accepted",
   ""usageThreshold": {
      "duration": 60,
      "totalVolume": 0,
      "downlinkVolume": 0,
      "uplinkVolume": 0
   },
}
```

Into a CAMARA API compliant response:

```
{
   "duration": 60,
   "device": {
      "ipv4Address": {
        "publicAddress": "84.125.93.10",
        "publicPort": 59765

      }
   },
   "applicationServer": {
       "ipv4Address": "172.20.120.84"
```

```
        },
        "devicePorts": {
            "ports": [
              57498
            ]
        },
        "applicationServerPorts": {
            "ports": [
              10000
            ]
        },
        "qosProfile": "low",
        "webhook": {
            "notificationUrl": "https://webhook.site/bed223f3-3263-
    4373-8f25-5dcef97c2f60"
        },
        "sessionId": "16f6e90d-431a-46a1-99e4-1fee437689b8",
        "startedAt": 1698137659,
        "expiresAt": 1698137719,"qosStatus": "REQUESTED"
    }
```

The notification sent by the SCEF/NEF when the resource is available is as follows:

```
    {
        "transaction":      https://{baseURL}/3gpp-as-session-with-
    qos/v1/ /subscriptions/<token>"
        "eventReports": [
        {
            "event": "SUCCESSFUL_RESOURCES_ALLOCATION",
            "flowIds": [
              1
            ]
        }
    ]
    }
```

This notification is transformed to a CAMARA API as follows:

```
    {
        "eventType": "QosStatusChangedEvent",
        "eventDetail": {
            "sessionId": "8e11ef7b-c7cd-4b8a-91cd-9948ada815e1",
            "qosStatus": "AVAILABLE"
        }
    }
```

The CAMARA QoD API also defines a QoS Profiles endpoint. This endpoint allows the developer to retrieve the QoS profiles. For this endpoint, there is no internal logic needed as it returns a static list.

2.5.3 Charging Aspects

The QoD API is classified as a network capabilities exposure services with impact on the device's data usage. Refer to SBI-CHF API [5] specification for details on possible options for charging for this category.

2.5.4 Roaming (Home Routed scenario) Aspect

In the Home Routed (HR) roaming scenario, all end user traffic is routed back to the home network (Operator A) as depicted in the Figure 2 below.

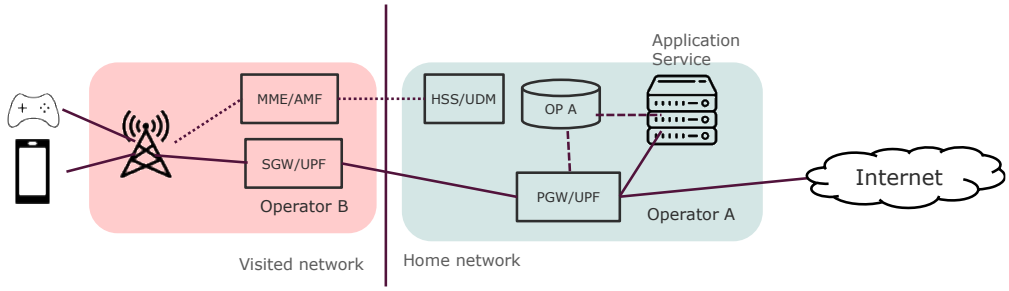

Figure 2: Simplified roaming architecture for home routed scenario

When the devices register into the visited network (Operator B), the visited network contacts home network (Operator A) and requests the service and QoS from Home Subscriber Server (HSS) in 4G or Unified Data Management (UDM) in 5G that the devices can use. Once the data sessions are established, all device data is routed back to the home network.

If an application provider or a developer requests the CAMARA QoD API, the home network is responsible for modifying the network. However, requesting a lower latency/packet delay budget may not result in better user experience. The highest latency gain is in the radio network that belongs to the Operator B, and Operator A cannot modify that. On top of it, there are Internetwork Packet Exchange (IPX) networks that introduce additional latency and are not managed by the Operator A.

2.5.5 Federation Aspects

In the HR roaming scenario as described in Section 2.5.4, there is no need to contact the visited network (Operator B) via East/Westbound Interface (EWBI).

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	16 Feb 2024	New PRD (OPG.09)	ISAG	H. Mariotte/Orange

A.2 Other Information

Type	Description
Document Owner	Operator Platform Group
Editor / Company	H. Mariotte/Orange

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.