International Civil Aviation Organization

THE ELEVENTH MEETING OF SYSTEM WIDE INFORMATION MANAGEMENT TASK FORCE (SWIM TF/11)

Bangkok, Thailand, 25 – 29 May 2026

Agenda Item 7: SWIM Task Force ToR, Programme, Work Plan, and Action Items review
- Development of an explanation for synchronous REQ/REP and asynchronous REQ/REP

PROGRESS ON DEVELOPMENT OF REST API URI STRUCTURING AND NAMING GUIDANCE MATERIAL

(Presented by the Republic of Korea, on behalf of the SWIM Pioneering Group (SIPG))

SUMMARY

This document covers the REST API URI Structuring & Naming Guidance Material, a subsequent deliverable of the R/R MEP Guidance Material within the SIPG Task 3.

1. INTRODUCTION

1.1 At SWIM TF/11, the WP - Progress on Development of the R/R MEP Guidance Material will be presented. This guidance material provides high-level approach to implement the R/R MEP in the APAC region and as of April 2026, at the SIPG level, there is a tentative consensus on how to implement synchronous R/R MEP. The consensus focuses on four key technical pillars:

- Protocol: HyperText Transfer Protocol(HTTP)/HTTP(Secure)S v1.1
- Architecture: Representational State Transfer(REST) as an architectural style
- Implementation: Application Programming Interface(API) Gateways(GW) (or Proxies)
- Routing: Path-based Routing (PBR)

1.2 If future implementation proceeds based on these technical pillars, service instances using RESTful API are expected to be exposed through API GWs deployed at regional and local levels. API GWs are responsible for routing requests from the service consumer to the appropriate service provider. In such an environment, PBR provides consistent and predictable URI structures to ensure that requests are delivered to the correct destination. Also, REST is a set of architectural style, but it does not mandate strict rules for URI structuring or naming in the aspect of implementation. As a result, if there is no common convention for URI structuring and naming, different implementations may adopt inconsistent URI patterns, which can lead to interoperability issues, increased routing complexity, and operational inefficiencies when integrating services on a regional scale.

1.3 To address these challenges, common conventions and standardized URI structures ought to be defined and applied across the region. These conventions aim to support harmonized **runtime-routing** through API GWs, promote consistent service identification, and facilitate interoperable implementation of the synchronous R/R MEP in the SWIM environment.

1.4 This paper provides the progress on development of REST API URI Structuring & Naming Guidance Material, a subsequent deliverable of the R/R MEP Guidance Material within the SIPG Task 3.

2. DISCUSSION

2.1 The REST API URI Structuring & Naming Guidance Material outlines how to design and apply RESTful API URI structuring and naming conventions for implementing synchronous R/R MEP in the APAC region. This includes the definition of REST and RESTful APIs, the rationale for URI structuring and naming conventions, path element definitions, and data flow diagrams illustrating how requests are routed when the conventions defined in this guidance material are applied

2.2 Main contents of the REST API URI Structuring & Naming Guidance Material are as follows:

URI Naming Conventions and Reference Standards

a) Industry Best Practices (De-facto) The following industrial best practices are applied to ensure consistency and readability of URIs:

- URI path segments shall be lowercase;
- Multi-word path elements shall use hyphen-separated lowercase;
- Where names originally contain spaces (e.g. organization or service provider names), spaces shall be replaced with hyphens (-);
- Where version originally contain period (.) (e.g. x.y.z), shall only use major (vX) for URI path and specify exact version in the header;
- Underscores (_) and CamelCase shall not be used in URI path segments;
- Reserved characters, spaces, and special characters shall not be used; and
- Plural nouns should be used for collection resources where applicable (e.g. /services, /resources

b) Reference Standards (De-jure) The following international and regional standards are referenced to ensure harmonization and interoperability:

- ISO 3166-1 alpha-3: Used for the identification of States or territories to ensure globally consistent country codes;

c) Any Other Consideration Alignment with the Topic Naming Convention being developed under SIPG Task 2 is expected to be required.

URI Structure and Path Element Definition

a) General URI Structure and description of each element is as shown below:

```
http://{api-gateway-host}:{api-gateway-port}
/{region-code}
/{country-or-jurisdiction-code}
/{service-name}
/{service-version}
/{resource}
?{query-parameters}
```

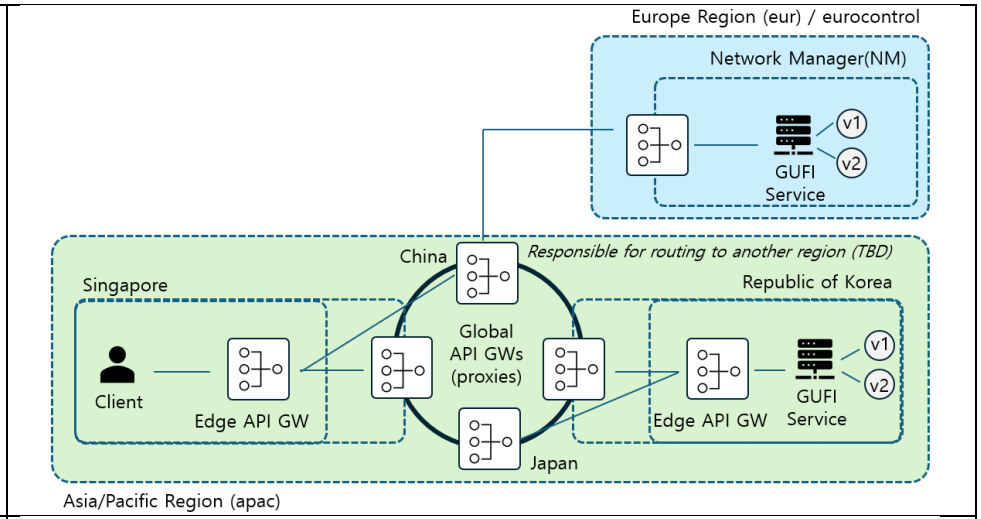
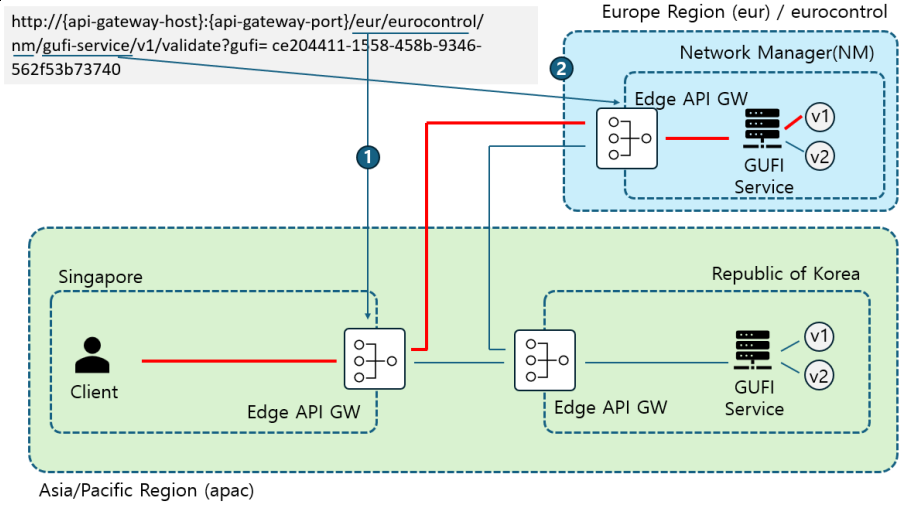
Path Element	Description	M/O	Example
{api-gateway-host}	Hostname or IP address of the API Gateway that provides a common entry point for service access	-	-
{api-gateway-port}	Network port on which the API Gateway listens for incoming requests	-	-

{region-code}	ICAO regional identifier (e.g. APAC) used to support regional-level routing and governance	M	apac
{country-or-jurisdiction-code}	{country-code} : Country or territory identifier based on ISO 3166-1 alpha-3 {jurisdiction code} : Registered jurisdictional or regional routing namespace used when the service is provided under a regional, multinational, or centrally managed SWIM environment where a single State or territory code is not applicable. Topic Naming Convention > Country Code	M	kor jpn sgp - eurocontrol
{service-provider}	Identifier of the organization or entity providing the service Topic Naming Convention > Organization	M	atmo
{service-name}	Name of the exposed resource or service Topic Naming Convention > System Name	M	gufi-service
{service-version}	Version identifier of the service (e.g. vX) Specific version of the service SHALL be explicitly indicated in the HTTP headers. GET /gufi-service/validate Host : example.aero Accept : application/json **API-Version : 1.1.1** Topic Naming Convention > Version	M	v1
{resource} <service-defined-path>	A relative URI (Uniform Resource Identifiers) to an individual endpoint. {resource} could have hierarchical path (i.e., service-defined-path) depending on service (e.g., /a/b/c/) Topic Naming Convention > System Specific Topic Hierarchy	M	Validate

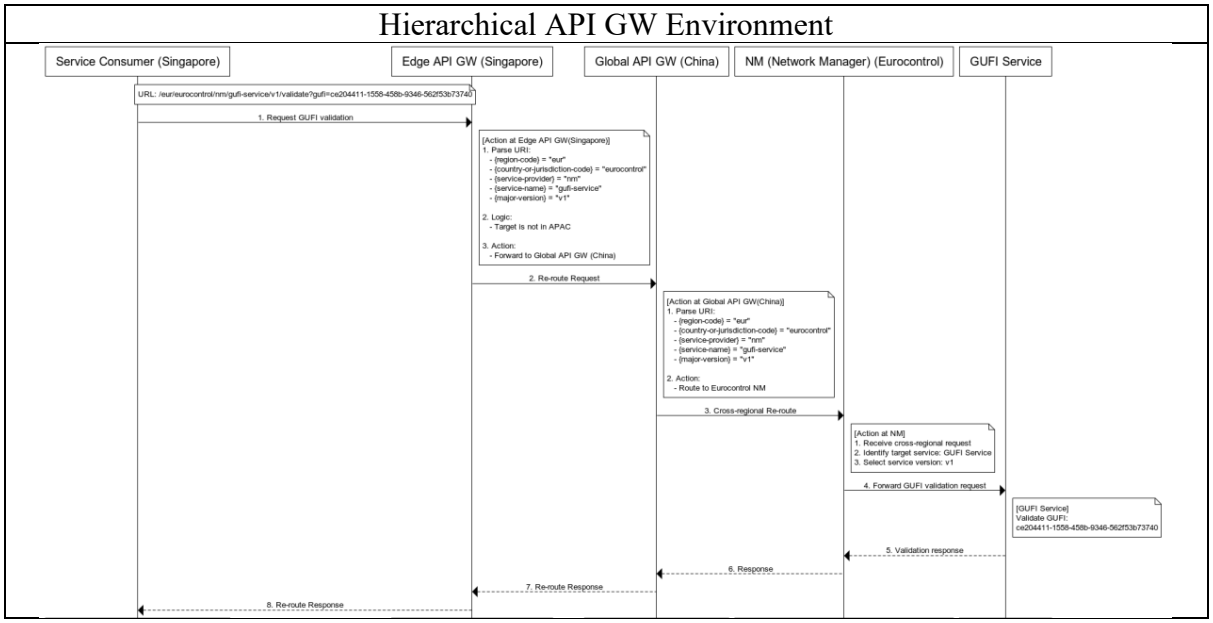
* Query parameters may be used to specify filtering criteria, selection options, or other non-identifying parameters. Their use does not violate REST principles and is commonly applied in RESTful APIs.
* As of Aril 2026, alignment between REST API URL Structuring and Naming and Topic Naming Convention is made except {region-code} and {environment}.

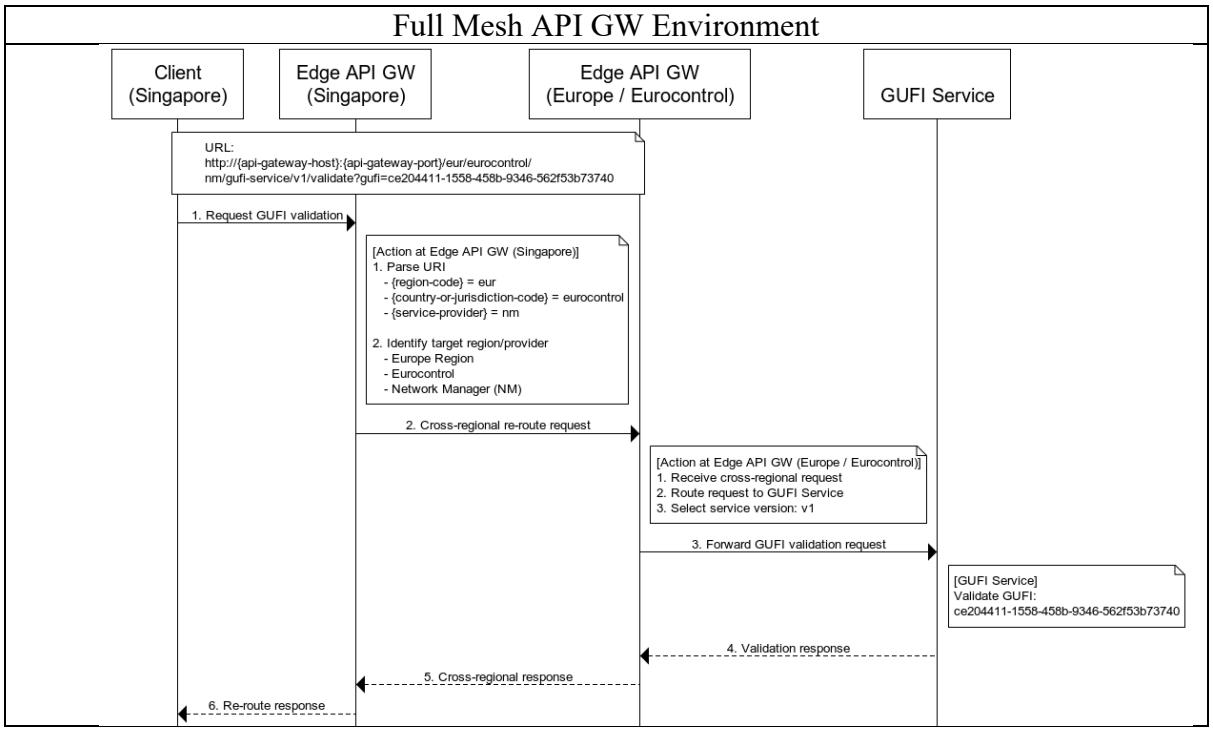
2.3 The guidance material describes path-based routing scenario of GUFI service that is one of FF-ICE services defined as an APAC SWIM common services using synchronous R/R MEP. As of April 2026, topology of R/R MEP is not yet defined, scenario consists of 1) full-mesh API GW based environment scenario and 2) hierarchical API GW based environment scenario. Also, cross-regional routing scenarios are also described for expendability of this guidance material.

Cross Regional Scenario - Overview

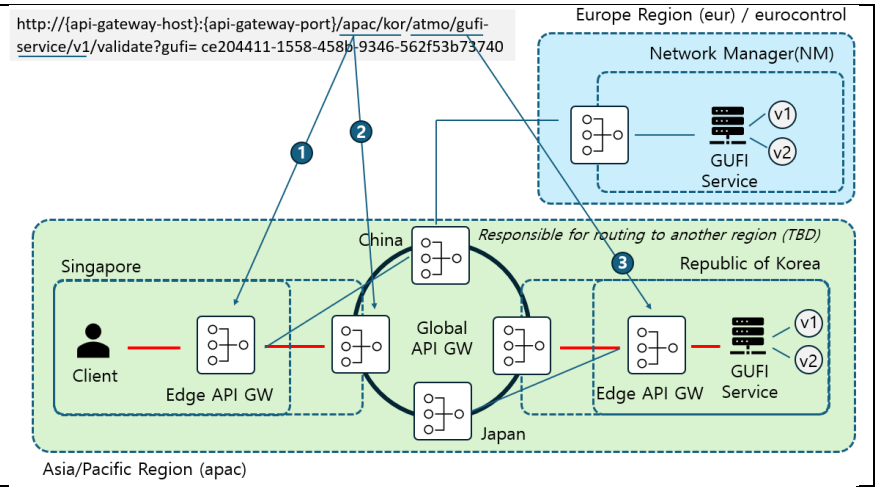
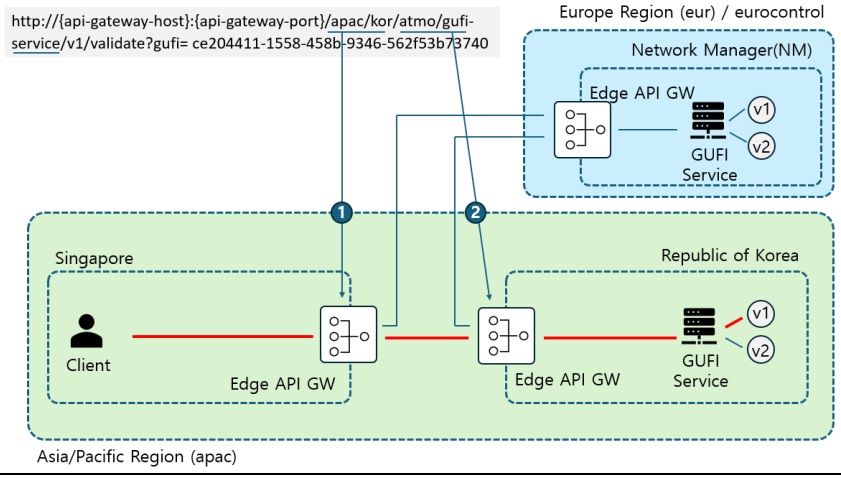
Hierarchical API GW Environment	
Full Mesh API GW Environment	

Cross Regional Scenario- Data Flow Diagram

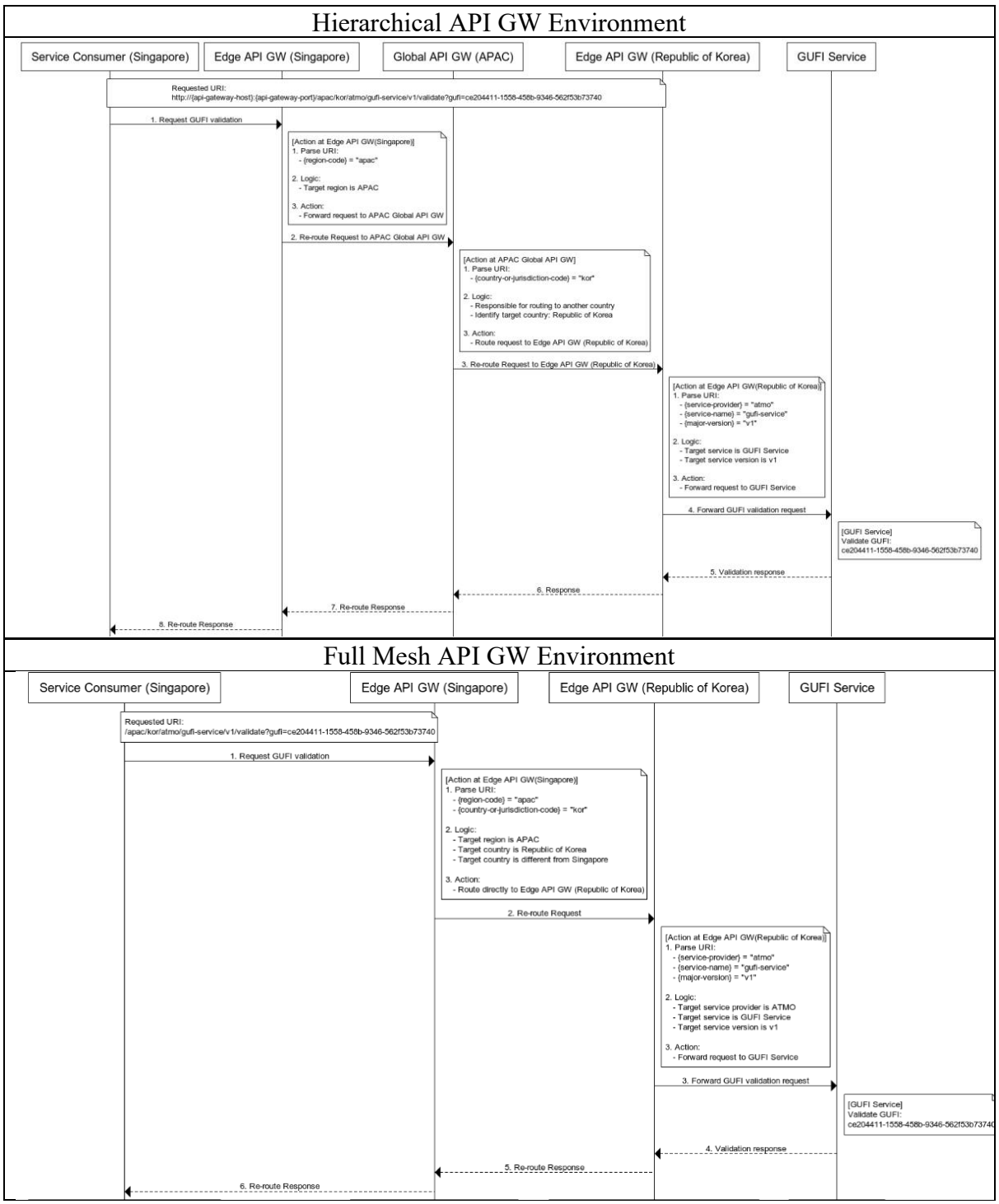




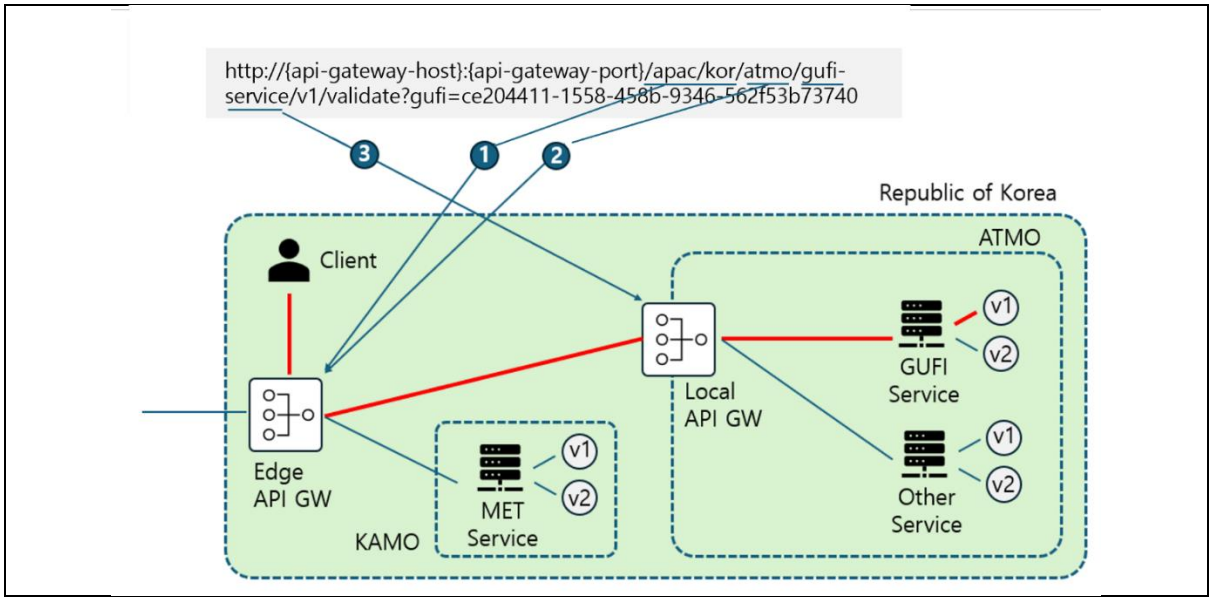
Inter-Regional Scenario - Overview

Hierarchical API GW Environment	
Full Mesh API GW Environment	

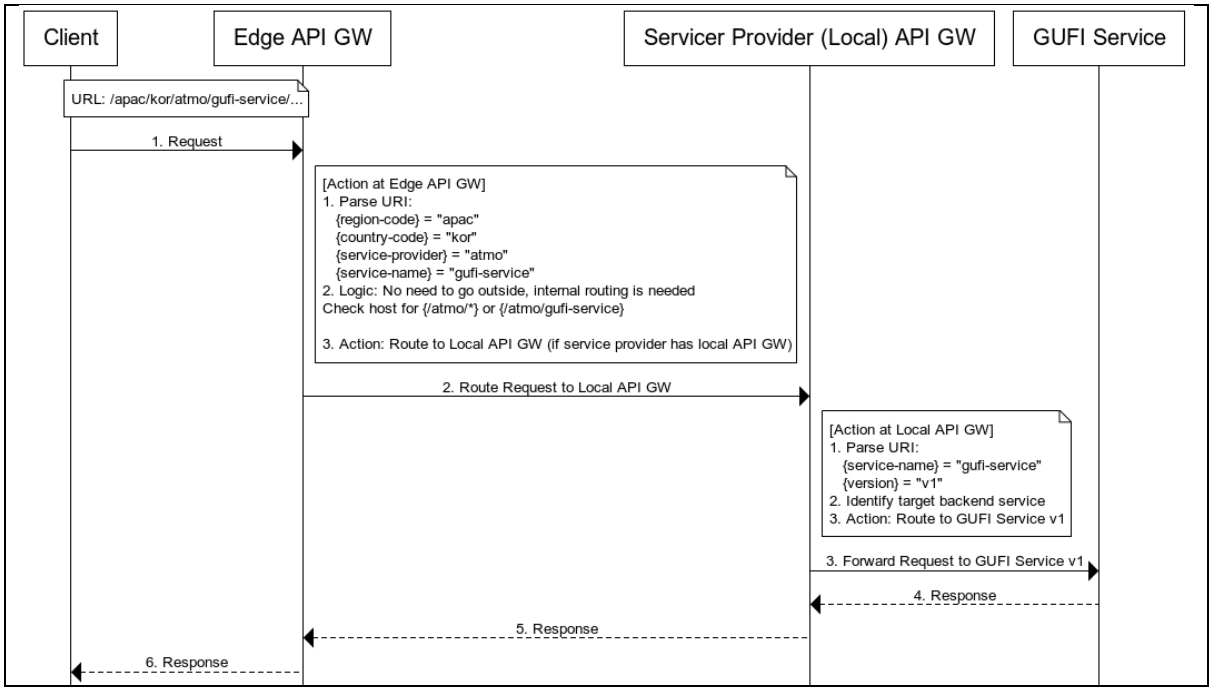
Inter-Regional Scenario- Data Flow Diagram



Local (Edge) Regional Scenario- Overview



Local (Edge) Regional Scenario- Data Flow Diagram



CONCLUSION

2.4 This paper summarized the key concept of REST API URI structing and naming for regional SWIM. Under the SIPG, R/R MEP will be implemented, tested and validated in the APAC SWIM prototype based on current version of guidance material. And, through a continuous feedback loop, any deficiencies or areas for improvement identified during the prototyping phase will be iteratively reflected in the guidance material to improve its maturity.

2.5 Same as the R/R MEP guidance material, consensus has been reached on certain topics—such as the need of URI structuring and naming convention—others remain subject to ongoing debate. These include the following:

- Should a full-mesh or hierarchical architecture be adopted?
- To what extent should the URI structure be simplified?
- How should alignment with the Topic Structuring and Naming Convention?
- How should alignment with the URI convention for MET services defined by the MET group be achieved?
- How should interoperability with other regions be ensured?

These unresolved issues will be further discussed.

3. ACTION BY THE MEETING

3.1 The meeting is invited to:

a) note the information contained in this paper; and

b) discuss any relevant matter as appropriate

– – – – – – – – – – – – – – –

Guidance Materials for REST-ful API Structuring and Naming in Asia/Pacific SWIM (Working Draft v2)

Apr. 2026

SWIM Implementation Pioneer Ad-hoc Group (SIPG)

Table of Contents

1. Introduction

1.1. Background

1.1.1. SWIM Implementation Pioneer Ad-hoc Group (SIPG)

The establishment of SIPG was decided at the SWIM TF/7 in 2023, and its Terms of Reference (TOR) was endorsed by the SWIM TF/? under the **“”**. Following SIPG’s TOR, the initial objective of the SIPG was to implement a seed/prototype version of the Asia/Pacific SWIM within 2024 as a means of kickstarting SWIM adoption in the region. Based on the initial objectives SIPG, SIPG built a protype version of Asia/Pacific SWIM and supported SWIM Demonstration over CRV and surveillance data sharing in the SWIM trial in Hong Kong, China, from 28 to 29 May 2024

After the supported SWIM Demonstration over CRV and surveillance data sharing in the SWIM trial, there were still needs for an expert group that can provide technical work for SWIM implementation in the Asia/Pacific region, and the SIPG continues its work in response to the need. In line with this, the SIPG defined sub-tasks to further materialize the implementation of SWIM in the Asia/Pacific region, and the sub-tasks, which are currently identified and in progress by the end of Dec 2026, as of Sep. 2025, are as below:

- Task 1: Requirements and Functionalities of the Edge EMS and Gateway EMS
- Task 2: New proposed hierarchical architecture review
- **Task 3: Guidance for the Sync Req/Rep and Async Req/Rep Message Exchange Pattern**
- Task 4:
- Task 5 : SWIM Technical Infrastructure Integration
- Task 6: SWIM Security Requirements and Implementation
- Task 7: SWIM Registry Requirements and Implementation
- Task 8:
- Task 9: APAC SWIM Integration Testing
- Task 10: Performance Testing SWIM TI
- Task 11: Regional SWIM TI Operationalization Guidance Material

1.1.2. Previous Achievement of Task 3, SIPG

Task 3 of SIPG developed “Guidance Materials for Request and Reply(R/R) Message Exchange Pattern(MEP) in Asia/Pacific SWIM” with the purpose of ensuring continuous and coherent implementation of the R/R MEP in the SWIM platform in a harmonized and interoperable manner within the region. The draft edition of the guidance material was released and presented at the FF-ICE Ad-hoc/3 held in Dec. 2025.

1.1.3. Limitation of Guidance Materials for Request and Reply Message Exchange Pattern in Asia/Pacific SWIM

While the document provides an overview of the R/R Message Exchange Pattern, including its operational concept, mechanisms, implementation considerations, topology, and routing mechanism, it does not specify detailed implementation guidance.

Specifically, for the synchronous R/R MEP, the document introduces Path-Based Routing (PBR) and designates it as the routing mechanism for Asia/Pacific SWIM; however, detailed descriptions of how PBR should be implemented are not provided.

1.2.Guidance Materials

Guidance material for REST(ful) API URI Structuring and Naming (i.e. this document) in the Asia/Pacific SWIM is one of the deliverables of Task 3 under SIPG. Republic of Korea, Australia, China, Hong Kong China, India, Japan, Fiji, Singapore, Thailand, Malaysia, USA, New Zealand. CANSO have volunteered and contributed to producing this document.

1.3. Purpose of the Document

This document provides guidance on the structuring and naming of REST API Uniform Resource Identifier (URI) for the R/R MEP in the Asia/Pacific region. It describes the principles for designing hierarchical URI structures, defining consistent naming conventions, and applying Path-Based Routing (PBR) to support synchronous R/R MEP in Asia/Pacific SWIM. The purpose of this document is to ensure a consistent, coherent, and harmonized implementation of the synchronous R/R MEP across the Asia/Pacific region, thereby enabling interoperability among SWIM stakeholders.

2. R/R MEP in SWIM

This chapter provides the conceptual framework of R/R MEP within the SWIM environment. It identifies the role of R/R MEP at the global level, and in the Asia/Pacific region, explains how the R/R MEP operates under different synchronization mechanisms, Furthermore, it provides comparison between synchronous and asynchronous R/R MEP, points out possible confusion part and clarifies them.

2.1.R/R MEP in the Global Level

ICAO SWIM Implementation Document (Doc. 10203) defined MEP including synchronous and asynchronous R/R as follows:

5.3.2.4.2 ***Message exchange patterns***

5.3.2.4.2.1 Several types of message exchange patterns (MEPs) are expected to be supported within a SWIM environment, including synchronous request/reply, asynchronous request/reply, one-way ("fire-and-forget") and publish/subscribe. The MEP used in any given exchange is directed by the information service provider to meet information service objectives. These MEPs include:

a) **Synchronous request/reply:** The consumer initiates a request to an information service; the service processes the request and generates a reply to the consumer. The consumer waits for the information service to provide a response. During this waiting period, the consumer cannot send or receive any other requests or responses. This pattern is specifically applicable to information services that can quickly execute and respond to consumer requests;

b) **Asynchronous request/reply:** The consumer initiates a request to an information service; the service processes the request and generates a reply to the consumer. However, the consumer is not restricted from performing other operations while waiting for the information service's response. This MEP requires that the consumer be able to receive messages at any time and correlate them with prior requests;

c) **One-way ("fire-and-forget"):** The consumer initiates a message to an information service without expecting any response from the information service. This MEP is particularly useful at the lower application layer, where immediate message responses are not required;

d) **Publish/subscribe (P/S):** The consumer initiates a subscription request to an information service. The subscription may be capable of providing details (such as through a filtering parameter) on the information being subscribed to:

1) in the case of a P/S with a push mechanism, the information service sends necessary updates (publish) to the consumer, in accordance with the subscription. This MEP requires that the consumer can receive messages at any time. However, the consumer is not restricted from completing other operations while waiting for the information service to respond; and

2) in the case of a P/S with a pull mechanism, the information service would keep necessary updates available to the consumer, in accordance with the subscription. This MEP requires that the consumer send requests to the information service to receive the updates.

Figure 2. Definition of MEP in the SWIM Document

2.2.R/R MEP in the Asia/Pacific Region

ICAO APAC SWIM Implementation Guidance Document (IGD, Working Draft) defines MEP including R/R as follows:

3.3.2 Standards for Resource-oriented Interface

3.3.2.1 RESTful API

The following table makes reference to RESTful API related standards and specifications required for supporting the service or infrastructure bindings of SWIM TI.

3.3.3 Standards for Method-oriented Interface

3.3.3.1 OGC WCS

The Open Geospatial Consortium (OGC) has developed a number of Web Common Service (WCS) standards that define services for accessing and manipulating geospatial data in a web environment, such as aeronautical information and meteorologic information. The following table makes reference to some of the key WCS standards and specifications required for supporting the service or infrastructure bindings of SWIM TI.

3.3.3.2 SOAP

As most users have not applied SOAP to current web applications, this standard is not recommended for the development of SWIM services. The following table makes reference to SOAP related standards and specifications required for supporting the service bindings of SOAP applications.

4.1 Functional Capabilities

The SWIM TI functional capabilities described in this section are common features widely supported by mainstream Commercial Off The Shelf (COTS) systems and services. Implementing a SWIM TI that supports all these capabilities is recommended. The SWIM TI functional capabilities can be grouped into three categories as follows:

Table 8. SWIM TI Functional Capabilities

Capability	Description	Related Technology
Messaging	This capability employs technologies that enable information exchange using various access methods (e.g., publish/subscribe, request/reply).	- Message brokers: such as Apache Kafka, RabbitMQ, ActiveMQ.

Figure 3. R/R MEP Related Description in the ICAO APAC SWIM IGD

Note: This section is intended for the business experts' group of the ICAO APAC region to explain why this document does not select SOAP as one of the candidate technologies to be explored, despite the fact that Eurocontrol's SWIM implementation uses SOAP for R/R MEP.

3. Overview of RESTful API

This chapter introduces the concept of a Representational State Transfer (RESTful) API. It describes the definition of the RESTful API, identifies core components, and explains the differences between RESTful API and non-RESTful API.

3.1. Definition of RESTful API

RESTful API is a widely adopted architectural style for web services in the modern ICT industry as a de facto standard. This architectural style defines a resource-oriented approach in which resources are identified by hierarchical Uniform Resource Locator (URL) using HTTP. RESTful API is a robust enabler of synchronous R/R MEP.

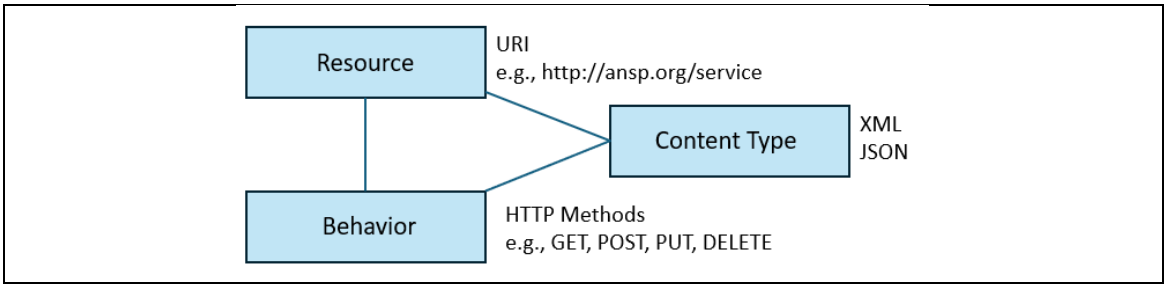


Figure 1. Concept of REST

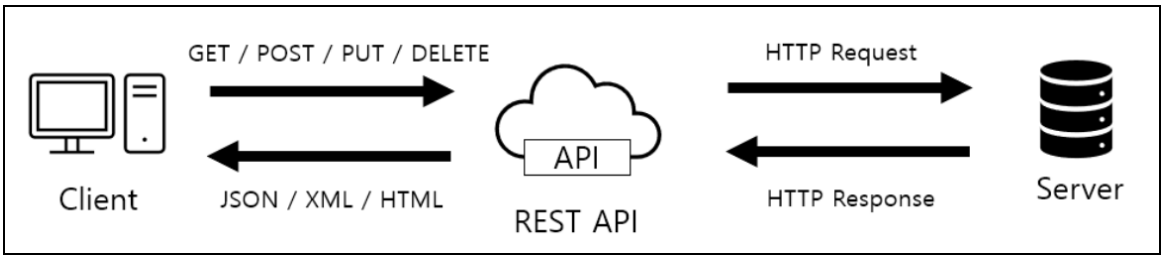


Figure 2. Basic RESTful API Interaction

3.2.Difference between REST and RESTful

The terms REST and RESTful are frequently used interchangeably in practice, despite having distinct meanings. Section 3.2 clarifies the definition of the term of REST and RESTful.

- **REST:** it refers to a conceptual architectural style that defines principles and constraints to design web services. It provides common guidelines for resource identification, interaction, and communication between clients and servers.
- **RESTful:** it refers to the practical implementation of REST principles through implementation of services or applications. A RESTful service strictly follows REST

constraints, such as resource-oriented URI design, use of standard HTTP methods, stateless communication, and uniform interfaces.

3.3. Components of RESTful API

Commonly used methods and components of RESTful API are as follows:

Methods

Methods	Description	CRUD Operation
POST	Create a new resource	CREATE
GET	Retrieve a resource representation	READ
PUT	Update or replace an existing resource	UPDATE
DELETE	Delete an existing resource	DELETE

** Note: Other methods (e.g., PATCH) defined in RFC 9110 and 5789 are also available as required*

Components

- **Resource:** it represents a entity or object exposed by the service instance. it is defined by a Uniform Resource Identifier (URI).
- **Behavior:** it defines the actions that are expressed based on HTTP methods (e.g., GET, POST, PUT, DELETE, and etc) on a resource. It describes interaction between a client and server (i.e., resource).
- **Message**: it is the actual payload delivered through request or response exchanged between the client and server. It contains metadata and payload (e.g., header, data)

3.4. Difference between RESTful API and non-RESTful API

Understanding the difference between RESTful APIs and non-RESTful APIs is essential to correctly understand its and avoid misinterpretation of implementation. This section is to clarify the differences between RESTful and non-RESTful APIs and distinguish their differences.

Category	RESTful API	Non-RESTful API
Architectural Approach	Resource-oriented architecture based on REST principles	Operation-oriented or procedure-based architecture
Conceptual Basis	REST architectural style and constraints	RPC or message-oriented service models
Resource Identification	Resources are identified using hierarchical URIs (e.g. /a/b/c)	Operations or services are identified by method or endpoint names
Use of HTTP Methods	Standard HTTP methods (GET, POST, PUT, DELETE) are used to represent operations	HTTP is often used as a transport only, typically relying on POST
State Management	Stateless interaction model	Often stateful, with server-side session or context management
Message Exchange Pattern	Request/Reply based on resource manipulation	Remote Procedure Call (RPC) or message-based invocation
Data Representation	Lightweight formats such as JSON or XML	XML-based messages with strict schemas

Service Description	Optional or lightweight (e.g. OpenAPI, WADL)	Formal service description (e.g. WSDL)
Service Registry	Not required	Often relies on service registries (e.g. UDDI)
Required Technology Stack	Minimal, HTTP-based	Requires additional middleware and WS-* stack
Scalability and Flexibility	High scalability and loose coupling	Relatively lower scalability due to tight coupling
Typical Use Cases	Data-centric services and UI-oriented applications	Transaction-oriented and enterprise integration services
Current Challenges	Lack of strict standardization across implementations	Complex usage and heavyweight protocols

4. URI Structuring and Naming

This chapter outlines URI structuring and naming. It describes why URI structuring and naming are required, and what needs to be considered.

4.1.Rationale for URI Structuring and Naming

In the Asia/Pacific SWIM environment, service instances using RESTful API are expected to be exposed through API GWs deployed at regional and local levels. API GWs are responsible for routing of requests from the service consumer to the appropriate service provider.

In such an environment, Path-based Routing (PBR) provides consistent and predictable URI structures to ensure that requests are delivered to the correct destination.

REST is a set of architectural principles, but it does not mandate strict rules for URI structuring or naming in the aspect of implementation. As a result, if there is no common convention for URI structuring and naming, different implementations may adopt inconsistent URI patterns, which can lead to interoperability issues, increased routing complexity, and operational inefficiencies when integrating services at a regional scale.

To address these challenges, this chapter defines common conventions and standardized URI structures to be applied across the Asia/Pacific region. These conventions aim to support harmonized routing through API GWs, promote consistent service identification, and facilitate interoperable implementation of the synchronous R/R MEP in the SWIM environment.

4.2. Consideration for URI Structuring and Naming

While primarily intended for the Asia/Pacific region, the URI structuring and naming guidance in this document has been developed with future cross-regional interoperability in mind.

The proposed URI structure is intended to be extensible and adaptable, so that it may support service interactions not only within the Asia/Pacific region but also across regions as SWIM implementation matures at the global level.

To develop this document, commonly adopted IT industry best practices and widely recognized standards have been taken into account, including REST architectural principles, established API design patterns, and relevant international and ICAO specifications.

By aligning regional conventions with de facto industry practices and applicable standards, this guidance aims to minimize divergence from global implementational use case in the ICT industry while addressing the specific operational and routing requirements of the Asia/Pacific SWIM environment.

4.3. URI Naming Conventions and Reference Standards

Industry Best Practices (De-jure)

The following industrial best practices are applied to ensure consistency and readability of URIs:

- URI path segments shall be lowercase;
- Multi-word path elements shall use hyphen-separated lowercase;
- Where names originally contain spaces (e.g. organization or service provider names), spaces shall be replaced with hyphens (-);
- Where version originally contain period (.) (e.g. x.y.z), shall only use major (vX) for URI path and specify exact version in the header;
- Underscores (_) and CamelCase shall not be used in URI path segments;
- Reserved characters, spaces, and special characters shall not be used; and
- Plural nouns should be used for collection resources where applicable (e.g. /services, /resources

Reference Standards (De-jure)

The following international and regional standards are referenced to ensure harmonization and interoperability:

- ISO 3166-1 alpha-3: Used for the identification of States or territories to ensure globally consistent country codes;

Any Other Consideration

Alignment with the Topic Naming Convention being developed under SIPG Task 2 is expected to be required.

4.4. URI Structure and Path Element Definition

4.4.1. General URI Structure

```
http://{api-gateway-host}:{api-gateway-port}
/{region-code}
/{country-or-jurisdiction-code}
/{service-provider}
/{service-name}
/{service-version}
/{resource}
?{query-parameters}
```

* Query parameters may be used to specify filtering criteria, selection options, or other non-identifying parameters. Their use does not violate REST principles and is commonly applied in RESTful APIs.

4.4.2. Description of URI Path Element

Path Element	Description	M/O	Example
{api-gateway-host}	Hostname or IP address of the API Gateway that provides a common entry point for service access	-	-
{api-gateway-port}	Network port on which the API Gateway listens for incoming requests	-	-
{region-code}	ICAO regional identifier (e.g. APAC) used to support regional-level routing and governance	M	apac
{country-or-jurisdiction-code}	{country-code} : Country or territory identifier based on ISO 3166-1 alpha-3 {jurisdiction code} : Registered jurisdictional or regional routing namespace used when the service is provided under a regional, multinational, or centrally managed SWIM environment where a single State or territory code is not applicable. Topic Naming Convention > Country Code	M	kor jpn sgp - Eurocontrol
{service-provider}	Identifier of the organization or entity providing the service Topic Naming Convention > Organization	M	atmo - nm
{service-name}	Name of the exposed resource or service Topic Naming Convention > System Name	M	gufi-service
{service-version}	Version identifier of the service (e.g. vX) Specific version of the service SHALL be explicitly indicated in the HTTP headers. GET /gufi-service/validate Host : example.aero Accept : application/json **API-Version : 1.1.1** Topic Naming Convention > Version	M	v1
{resource} <service-defined-path>	A relative URI (Uniform Resource Identifiers) to an individual endpoint. {resource} could have hierarchical path (i.e., service-defined-path) depending on service (e.g., /a/b/c/) Topic Naming Convention > System Specific Topic Hierarchy	M	Validate

* M: Mandatory, O: Optional

4.5. URI Structure Example

4.5.1. Decentralized Environment (e.g., APAC)

```
http://{api-gateway-host}:{api-gateway-port}
/apac
/kor
/atmo
/gufi-service
/v1
/validate
?gufi= ce204411-1558-458b-9346-562f53b73740
```

4.5.2. Centralized Environment (e.g., Eurocontrol)

```
http://{api-gateway-host}:{api-gateway-port}
/eur
/eurocontrol
/nm
/gufi-service
/v1
/validate
?gufi= ce204411-1558-458b-9346-562f53b73740
```

* In a centralized SWIM environment, such as Europe, "eurocontrol" may be used as a {jurisdiction code}, rather than a {country-code}.

5. Path-Based Routing Scenarios using URI Structure

This chapter describes path-based routing scenario of GUFI service that is one of FF-ICE services defined as an APAC SWIM common services using synchronous R/R MEP.

5.1.1. Overview of GUFI Service

GUFI service is defined in the APAC SWIM Common Services (APAC Common SWIM Information Services, WP11, SWIM TF/10), but it is briefly described what it is, to make use it for scenarios, pseudo-operation of GUFI service is defined as follows:

Operation	Use Case	Resource
ValidateGufi	Retrieving a GUFI from GUFI service instance	/gufi-service /validate?gufi={gufi}

5.1.2. Overview of Scenario Topology

As R-R MEP Guidance Material (draft) doesn't define the specific topology of API GW in the Asia/Pacific region, but identifies candidate topologies, this chapter uses two different topologies to describe scenarios.

5.1.2.1. Full Mesh API GW Environment

In a full mesh architecture, every Edge API GW establishes a direct communication link with all other Edge API GWs in the network. This direct gateway-to-gateway connection model removes the need for an intermediary global routing hub. It allows incoming requests to be routed immediately to the appropriate service provider across the region based on the established RESTful URI structures.

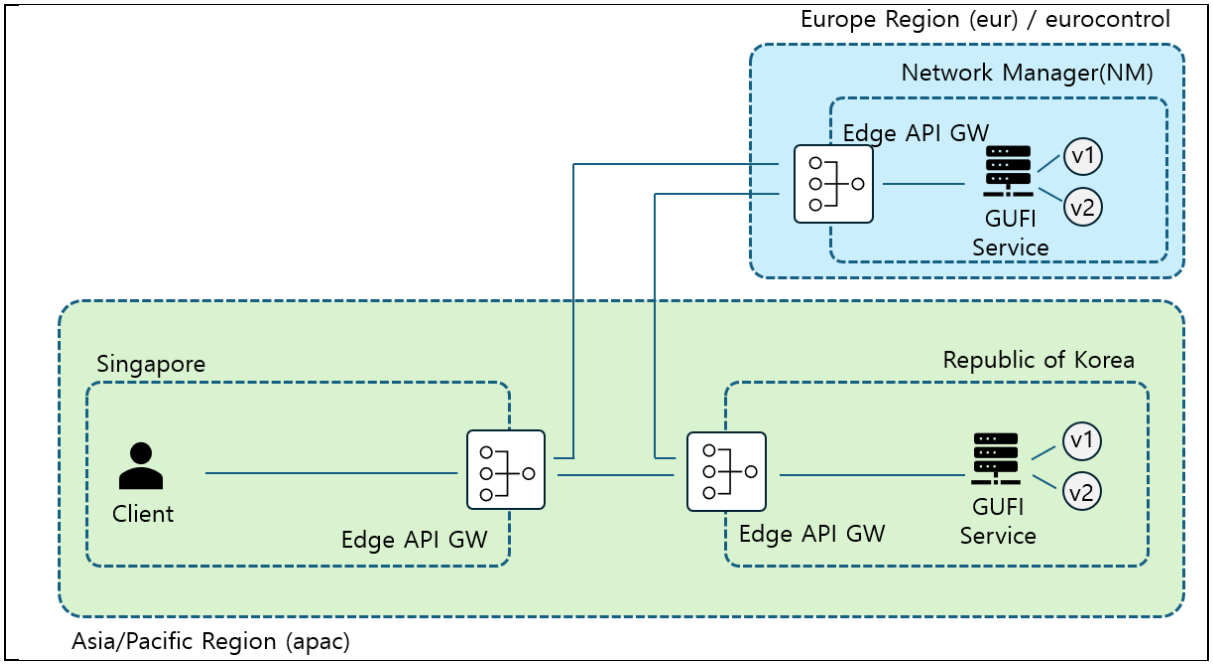


Figure 3 Full Mesh API GW Environment Scenario

5.1.2.2. Hierarchical API GW Environment

In a hierarchical architecture, Edge API GWs do not connect directly to every other gateway in the network. Instead, local Edge API GW requests the delivery to a cluster of regional global API GW (or proxies).

A cluster of regional global API GWs (or proxies) act as the primary router for the network. It receives incoming requests from edge API GWs and analyses the established RESTful URI structures to forward the traffic to the appropriate destination at the global level. The destination could be another edge API GW within the same region, such as routing from Singapore to the Republic of Korea. For cross-regional interoperability, some of global API GWs (or global proxies) could take a responsibility for message delivery between regions.

Unlike the full mesh topology, a hierarchical architecture could reduce the number of direct point-to-point connections required across the network, making it a highly scalable approach for integrating complex, cross-regional SWIM environments.

To ensure high availability and operational continuity against malfunctioning of a certain regional global API GW (or proxy), failover mechanism could be configured. Edge API GW and global API GWs (or proxies) could support upstream level redundancy, and active health check & retries. This capability would be defined in the API GW requirement specification document (TBD).

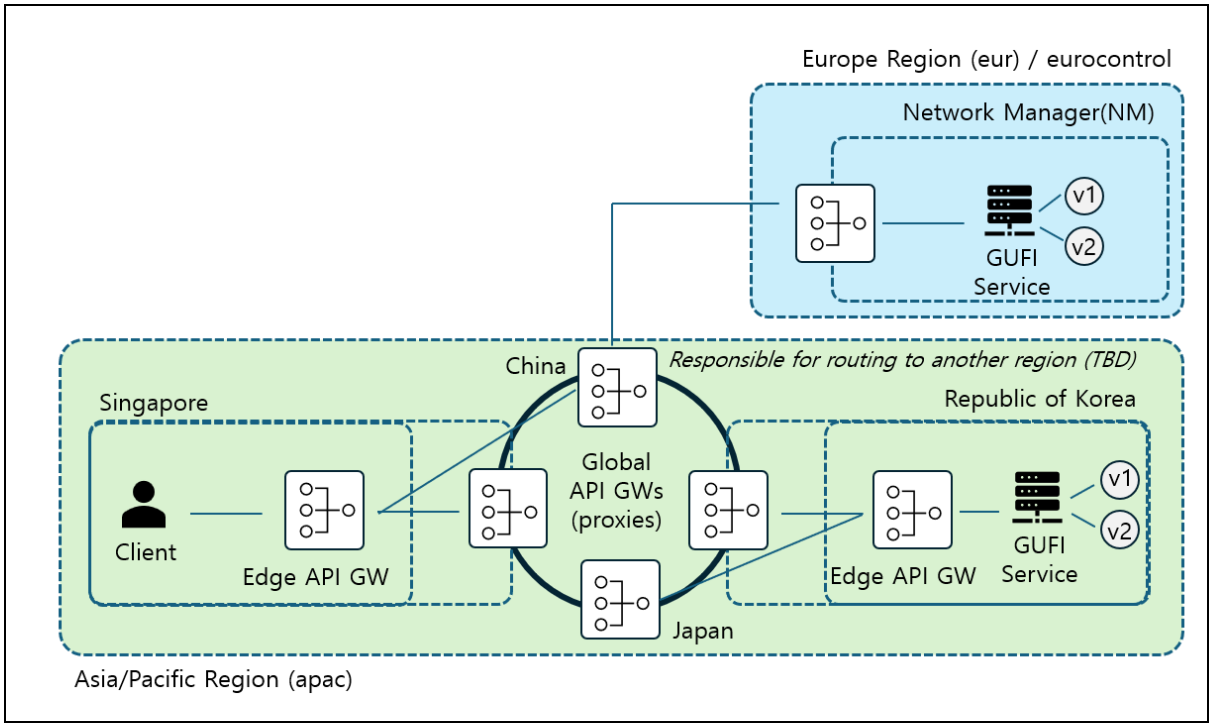


Figure 4 Hierarchical API GW Environment Scenario

5.1.3. Cross-regional Routing Scenario

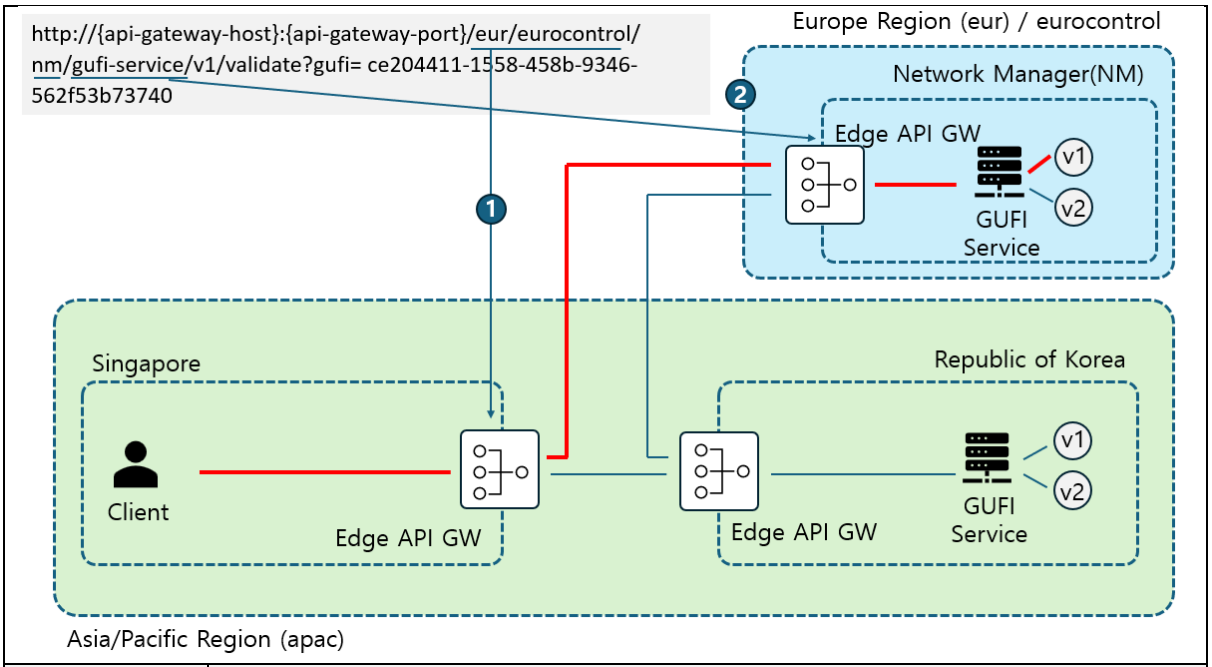
To enable cross-regional routing, the {region-code}, {country-or-jurisdiction-code} and {service-provider} path elements are referenced. The edge API GW or global API GW (or global proxies) responsible for cross-regional routing interprets the URI as follows:

1. checks {region-code} and follow step 2 or 3 when {region-code} does not match its own region.

2. If the destination region operates a centralized SWIM Technical Infrastructure (TI), the registered jurisdictional or regional routing namespace ought to be filled in {country-or-jurisdiction-code}. (e.g., eurocontrol). And, the edge API GW reroutes the request to the centralized API GW of that region based on the IP and port information of the {service-provider} registered in the edge API GW.

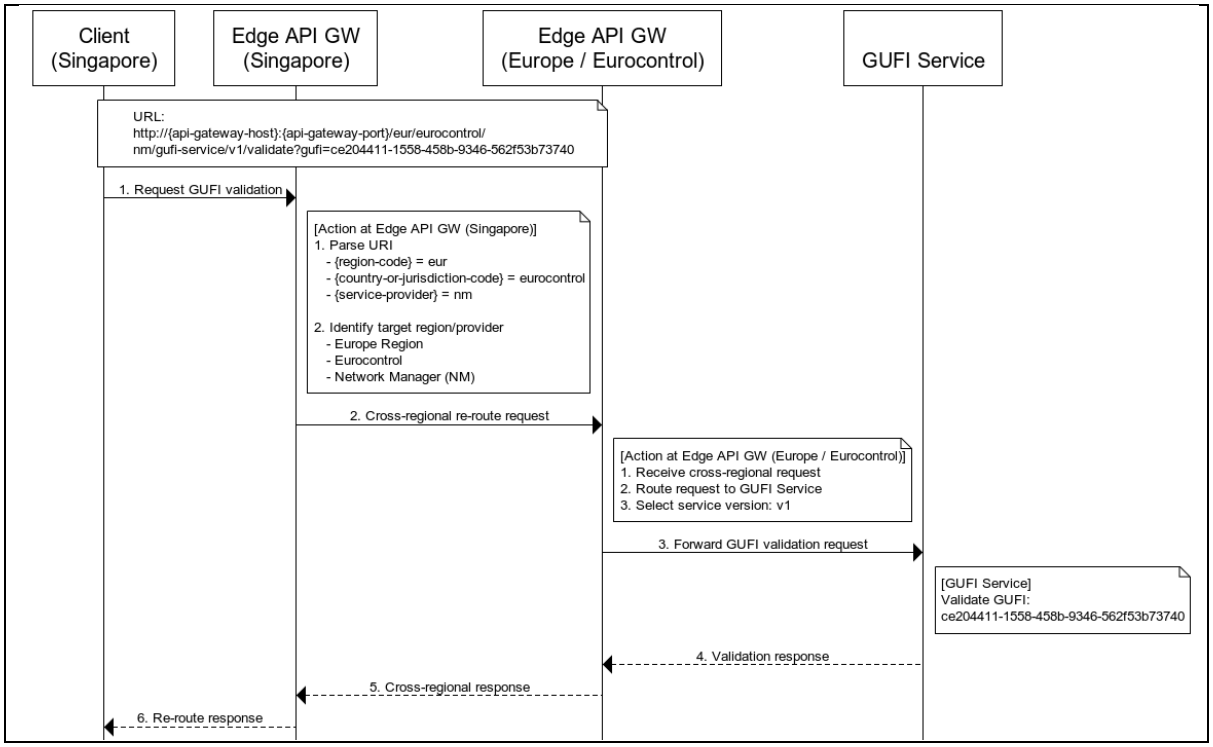
3. If the destination region operates a decentralized SWIM TI, the gateway reroutes the request to one of Edge API GWs in the target region based on the IP and port information associated with the {country-or-jurisdiction-code} registered in the edge API GW.

5.1.3.1. Full Mesh API GW Environment

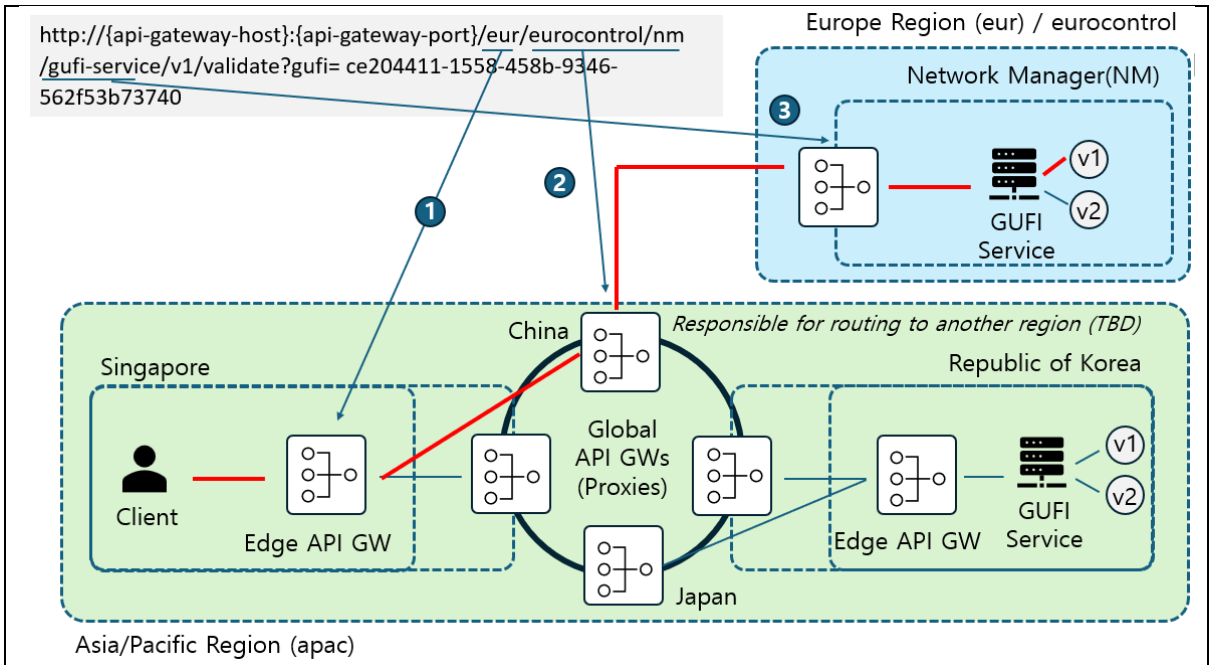


Step	Mechanism
1	**Determine the Routing Type and Forward to the Edge API GW** - When a client initiates a request, the edge API GW checks the {region-code} in the URI to determine whether the routing is inter-regional or cross-regional. - If it is determined to be cross-regional routing, the request is forwarded to the direct external API GW that isn't registered in own region.
2	**Final Routing within the Destination Region** - Upon receiving the request, the edge API GW in the destination region checks the remaining detailed path elements. /{country-or-jurisdiction-code}/{service-provider} - Routing mechanisms may vary depending on the topology of each region. - Refers to 5.1.5 - Local Routing Scenario

* In this scenario, each Edge API GW SHALL recognize other API GWs both within and outside its region. Whenever a region code outside of the local region is identified in the path, the originating Edge API GW is required to route the request directly to the destination region's API GW.



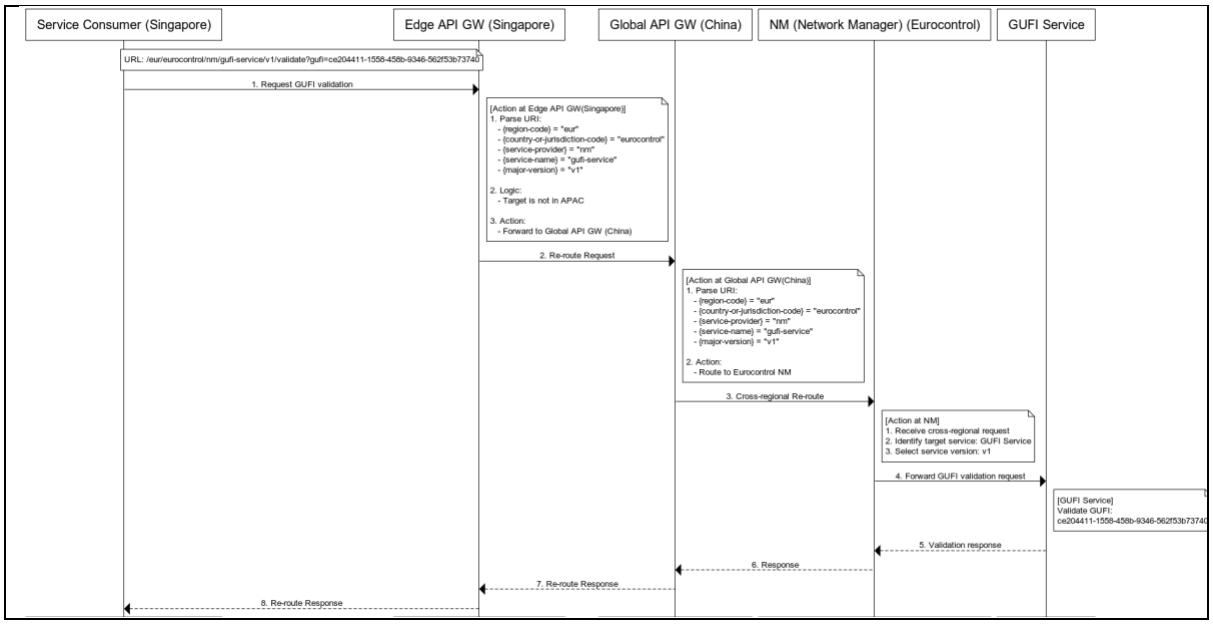
5.1.3.2. Hierarchical API GW Environment



Step	Mechanism
1	**Determine the Routing Type and Forward to the Responsible Global API GW (or global proxies)** - When a client initiates a request, the Edge API GW checks the {region-code} in the URI to determine whether the routing is inter-regional or cross-regional. - If it is determined to be cross-regional routing, the request is forwarded to the Global API GWs (or global proxies) responsible for cross-regional message transfer (e.g., the China node in the diagram).

	- If the originating Edge API GW is not directly connected to the responsible Global API GW (or global proxies), the request is relayed through an adjacent connected Global API GW (or global proxies) to reach the responsible node.
2	**Route to the Destination Region's Edge API GW** - The Global API GW (or global proxies) responsible for cross-regional transfer interprets the URI elements to route the request to its destination. - When routing to a region with a centralized SWIM TI, such as Europe, the {country-or-jurisdiction-code} field ought to contain a jurisdictional or regional routing namespace, while the {service-provider} field should identify the relevant service provider, such as nm. - On the other hand, {country-or-jurisdiction-code} is used for target region where decentralized environment is adopted
3	**Final Routing within the Destination Region** - Upon receiving the request, the edge API GW or global API GW (or global proxies) in the destination region checks the remaining detailed path elements. - Routing mechanisms may vary depending on the topology of each region. - Refers to 5.1.5 - Local Routing Scenario

* In this scenario, a specific global API GW (or global proxies) must be authorized to handle cross-regional routing. Whenever a region code outside of the local region is identified in the path, all other edge and global API GWs (or global proxies) are required to route the request to that designated global API GW (or global proxies).



5.1.4. Inter-Regional Routing Scenario

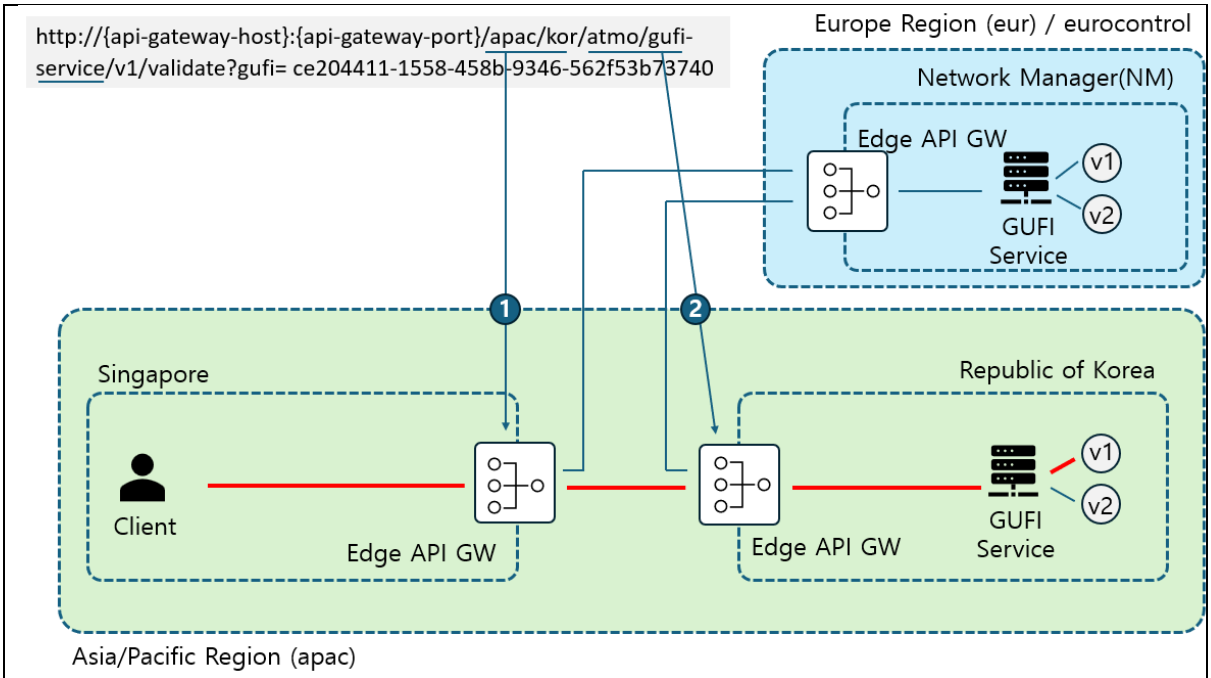
To enable inter-regional routing for Asia/Pacific region, the {region-code}, {country-or-jurisdiction-code} path elements are referenced. The edge API GW or global API GW (or proxies) responsible for inter-regional routing interprets the URI as follows:

Step 1. The edge API GW checks {region-code} and follow step 2 when {region-code} matches its own region. For full mesh API GW environment, go to Step 2, and for hierarchical API GW environment, go to Step 3.

Step 2. In the full mesh API GW environment, the edge API GW reroutes the request to one of edge API GWs within the Asia/Pacific region based on the IP and port information associated with the {country-or-jurisdiction-code} registered in the API GW.

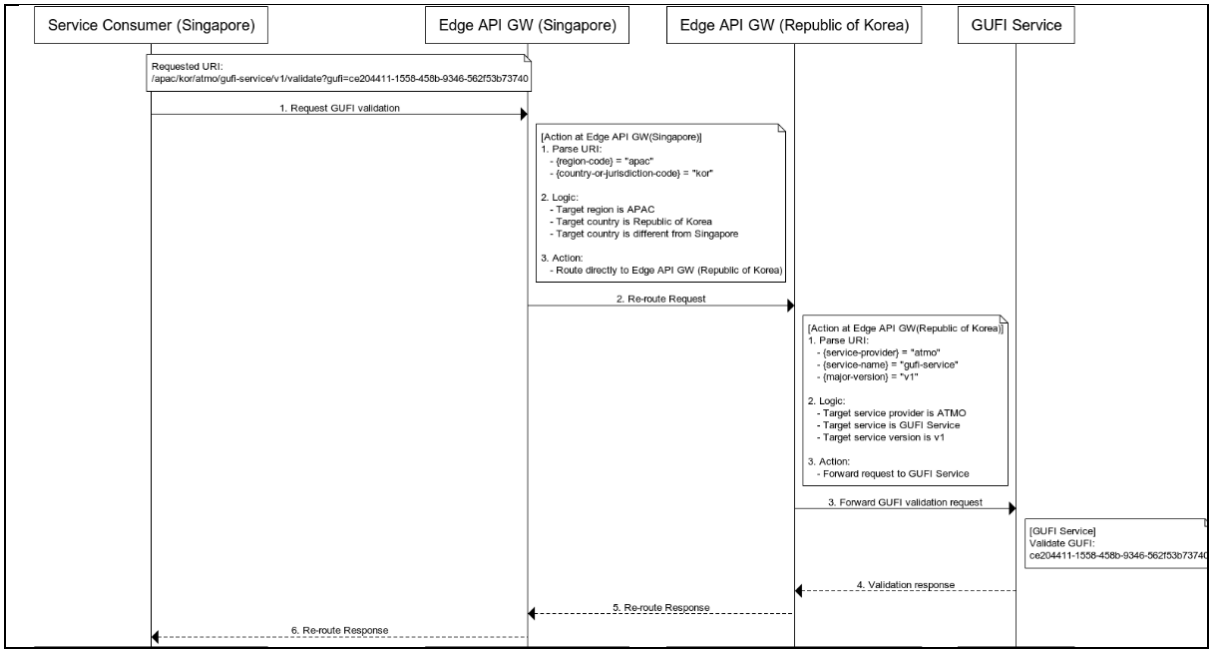
Step 3. In the hierarchical API GW environment, the edge API GW reroute to the global API GW (or proxies). The global API GW (or proxies) reroutes the request to another global API GWs (or proxies) or edge API GW associated with routing path elements.

5.1.4.1. Full Mesh API GW Environment

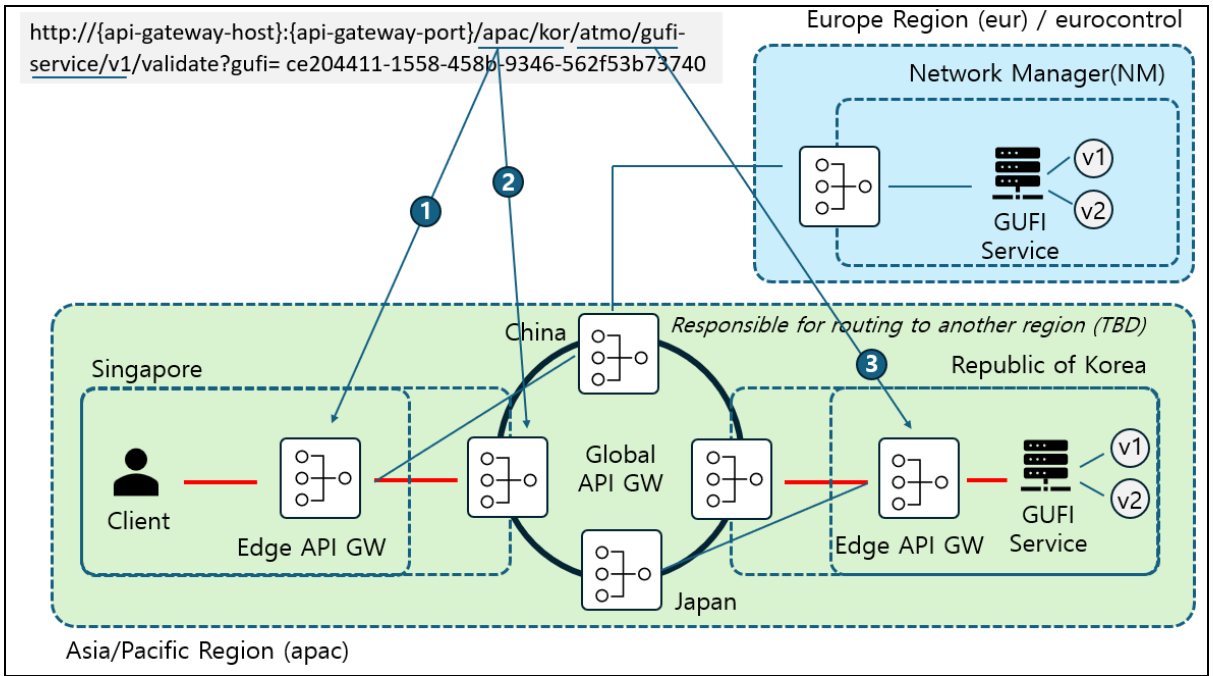


Step	Mechanism
1	**Determine the Routing Type and Forward to the Edge API GW** - When a client initiates a request, the Edge API GW checks the {region-code} in the URI to determine whether the routing is inter-regional or cross-regional. - If it is determined to be inter-regional routing, the request is forwarded to the direct edge API GW registered in own region.
2	**Conduct Local Routing** - Refers to 5.1.5 - Local Routing Scenario

* In this scenario, each Edge API GW SHALL recognize other API GWs both within and outside its region. Whenever a region code outside of the local region is identified in the path, the originating Edge API GW is required to route the request directly to the target API GW within the region.



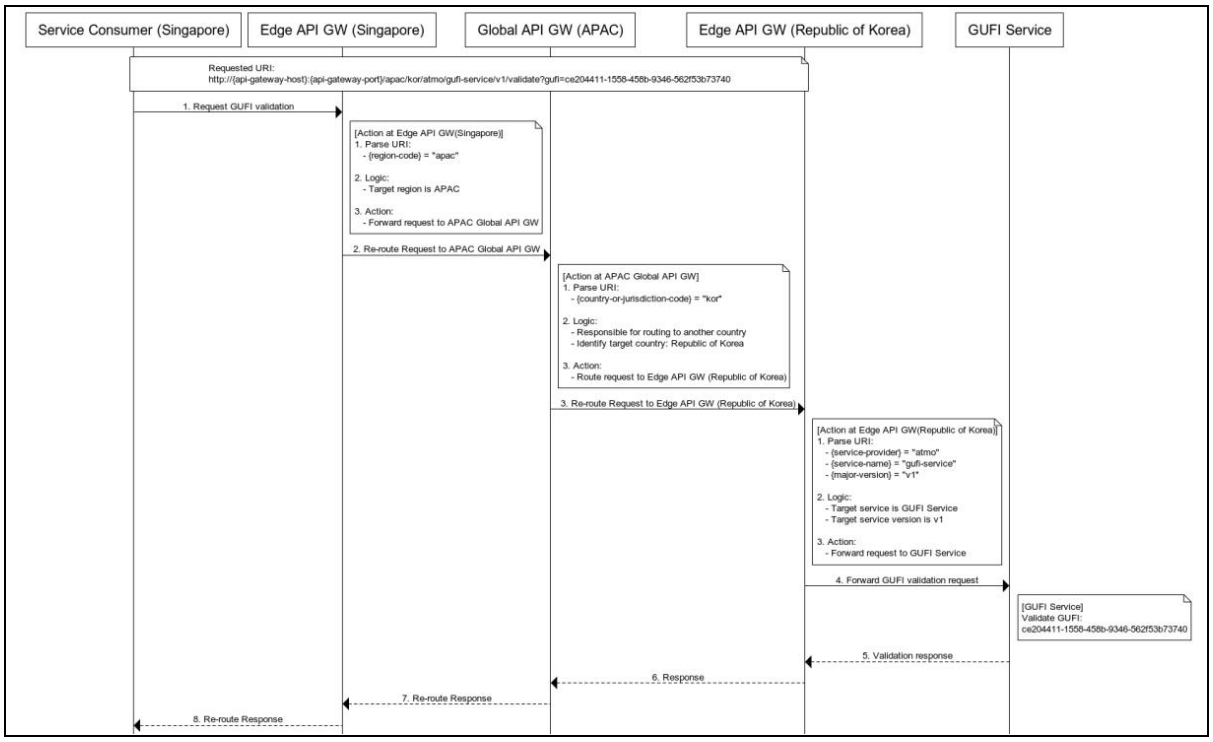
5.1.4.2. Hierarchical API GW Environment



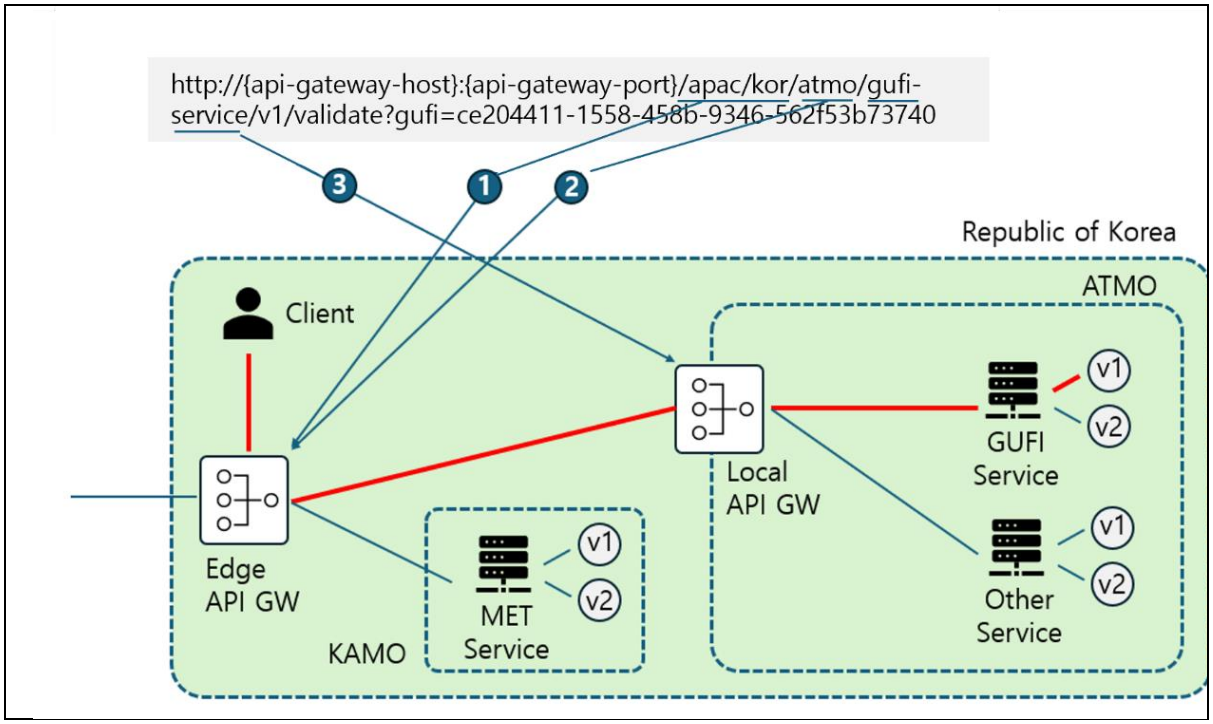
Step	Mechanism
1	**Determine the Routing Type and Forward to the Global API GW** - When a client initiates a request, the Edge API GW checks the {region-code} in the URI to determine whether the routing is inter-regional or cross-regional. - If it is determined to be inter-regional routing, the request is forwarded to the global API GW
2	**Forward to Another Global API GW Node and Route to Target Country** - The global API GW checks the {country-or-jurisdiction-code} in the URI.

	- The global API GW routes the message to another Global API GW node connected to the target country code.
3	**Conduct Local Routing** - Refers to 5.1.5 - Local Routing Scenario

* In this scenario, an edge API GW does not need to establish direct connections with every other gateway. Instead, the global API GW SHALL recognize other global API GW nodes and their associated {country-or-jurisdiction-code} within the region.



5.1.5. Local Routing Scenario



Step	Mechanism
1	**Determine the Routing Type** - The edge API GW checks the {country-code} in the URI to determine whether the request is intended for domestic routing. - If the target is not domestic, the request is routed to an external edge API GW or a global API GW.
2	**Route to Service Provider** - For domestic routing, the edge API GW checks the service-provider path element. - Routing mechanisms may vary depending on the specific service-provider. Some service providers might not operate their own local API GW, exposing their service instances directly to the Edge API GW. In contrast, some service providers may maintain their own local API GW, requiring the Edge API GW to route the request to the provider's local gateway.
3	**Final Routing to the Service Instance** - The local API GW (or the edge API GW, if a local GW doesn't exist) checks the remaining detailed path elements, such as {service-name}, and {service-version}. Based on these elements, the gateway routes the request directly to the final service instance for execution.