Draft Guidance Materials for Request and Reply Message Exchange Pattern in Asia/Pacific SWIM

Oct. 2025

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 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. Limitation of the Previous SWIM Efforts in the Asia/Pacific Region

There have been various efforts for the implementation of SWIM in the Asia/Pacific region. These efforts are not only about the establishment of regional implementation guidance or standardization, but also implementation of regional SWIM prototype, and demonstrations. However, regarding the demonstration and technical efforts for message delivery, these efforts were mostly depended on an Enterprise Messaging System (EMS) using Publish and Subscribe (Pub/Sub) Message Exchange Pattern (MEP), as mentioned in “APPROACH TO GLOBAL API GATEWAY FOR WEB SERVICE (SWIM TF/10 – WP/18)”.

A MEP refers to the fundamental interaction mechanism that defines how messages are exchanged between heterogeneous systems. There are a few mechanisms to enable MEP such as Pub/Sub, Request and Reply (R/R), and Fire and Forget. And, ICAO SWIM Implementation document (Doc. 10203) identified this mechanism for MEP in the SWIM. However, given the current emphasis on the regional SWIM prototype architecture using an EMS which the SIPG is developing in the APAC region, the primary issue is to discuss how the Request/Reply MEP should be implemented.

1.2. Guidance Materials

Guidance materials (i.e. this document) 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 for R/R MEP in the Asia/Pacific region, and it covers business and technical aspect of R/R MEP including FF-ICE/R1 based a data flow and use-case diagram, with the purpose of ensuring continuous and coherent implementation of the R/R MEP to SWIM platform in harmonized and interoperable within the region.

2. Operational Concept

This chapter introduces the operational concept of the R/R MEP. It describes the definition of the R/R MEP, identifies the core components of the R/R MEP, including participants, synchronization mechanisms, and supporting elements.

2.1. Definition of R/R MEP

The R/R MEP is a communication model where a requester sends a request message to a replier, which then processes the message and returns a reply.

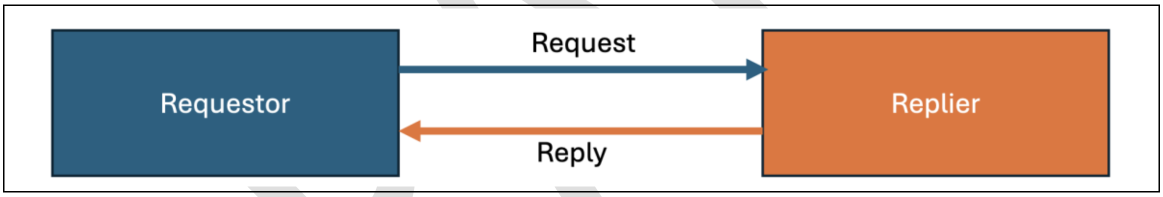


Figure X. Diagram

2.2. Components of R/R MEP

The components of R/R MEP could be distinguished as follows:

Core Participants

- **Requester:** The requester asks for something via a request; it could be simply considered as Create, Read, Update, Delete (CRUD) operation
- **Replier:** The replier processes the request and return a message in reply

Synchronization Mechanisms

- **Synchronous:** The requestor waits for the response before continuing its operation
- **Asynchronous:** The requestor's control flow is released after the request is sent, and the response is handled later, perhaps via a callback or another mechanism.

Supporting Elements

- **Payload:** it defines the actual contents of the message being exchanged. This could be Extensible Markup Language (XML), JavaScript Object Notation (JSON)
- **Transportation and Protocol Binding:** it defines how the R/R is implemented. This could be Hyper Text Transfer Protocol/Representational State Transfer (HTTP/REST), Simple

Object Access Protocol (SOAP), Advanced Messaging Queue Protocol (AMQP), Message Queuing Telemetry Transport (MQTT)

- **Error and Exception Handling:** it provides mechanism to manage any other errors or exceptions such as failures, timeouts, and invalid requests
- **Security and Policy Enforcement**: it provides authentication, authorization, encryption, compression, logging mechanism

3. 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.

3.1.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 X. Diagram

3.2.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 X. Diagram

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.

4. Mechanism

This chapter outlines the mechanism of the R/R MEP. It explains the difference between synchronous and asynchronous interactions, compares their characteristics, and highlights common points of confusion. It also clarifies how R/R MEP should be understood from a business perspective to support consistent implementation in the Asia/Pacific SWIM environment.

4.1.Synchronous R/R MEP

In ICAO SWIM Implementation Document, synchronous R/R MEP is defined as – *The consumer initiates a request to an information service; the service processes the request and generate 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 request*

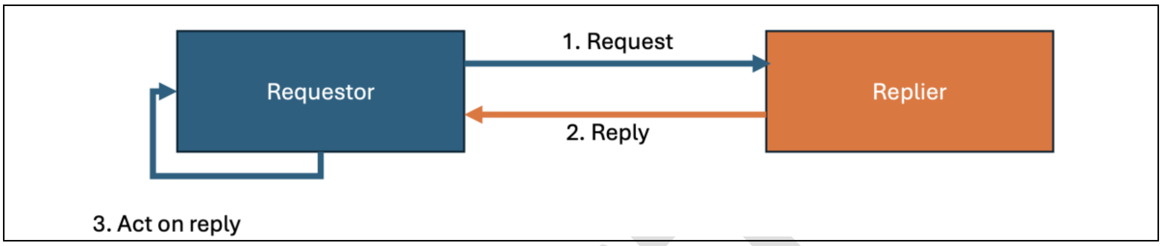


Figure X. Diagram

4.2.Asynchronous R/R MEP

In ICAO SWIM Implementation Document, synchronous R/R MEP is defined as – *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*

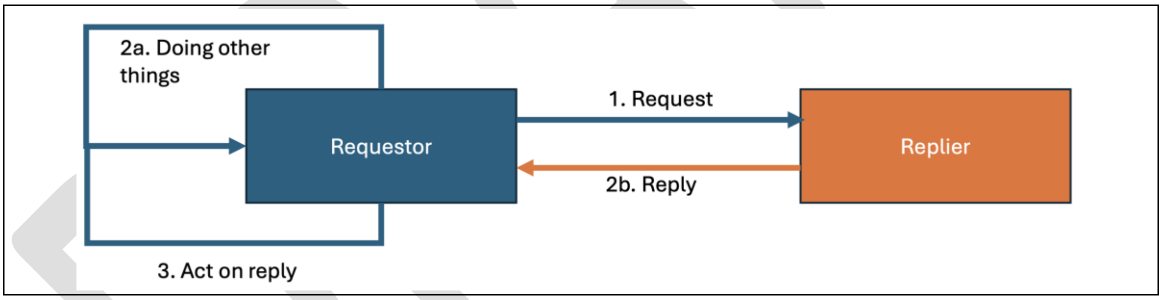


Figure X. Diagram

4.3. Comparison of Synchronous and Asynchronous R/R MEP

Comparison of synchronous and asynchronous R/R MEP is as follows:

Index	Synchronous	Asynchronous
Time Coupling	Both requester and replier are available at the same time.	Requester sends a request and continues its process; replier can send the response later when available.
Space Coupling	Requester needs to know the exact service endpoint (protocol, address, API).	Requester sends to a known endpoint, but response may arrive via callback, polling, or correlation ID; looser coupling in response handling.
Reliability Handling	Retries and error handling happen at requestor side.	Retries and correlation of delayed responses must be managed at the

		requester side (e.g., matching reply with original request).
Use Cases	• Low latency expected • Both parties are available • Immediate response interaction	• Replier may not be immediate • Deferred or background processing acceptable
Typical Scenarios	• User Authentication • User Interface Interactions • Database Read and Immediate Write	• Order processing with delayed confirmation • Flight plan filing with later validation • Weather data request with queued response • Batch data processing

Table X. Table

4.4. Confusion between Synchronous and Asynchronous R/R MEP

In real-world implementations, confusion often arises when distinguishing between synchronous and asynchronous Request/Reply (R/R) patterns, especially in cases where a service returns an immediate acknowledgment (e.g., Status code without payload) but the actual business result (e.g., Payload) is provided later.

This confusion was raised at the SWIM TF/10 – Approach to Global API Gateway for SWIM Web Services (WP/01) and SIPG WS/2 - *Request-Reply Message Exchange Pattern (SP/09).*

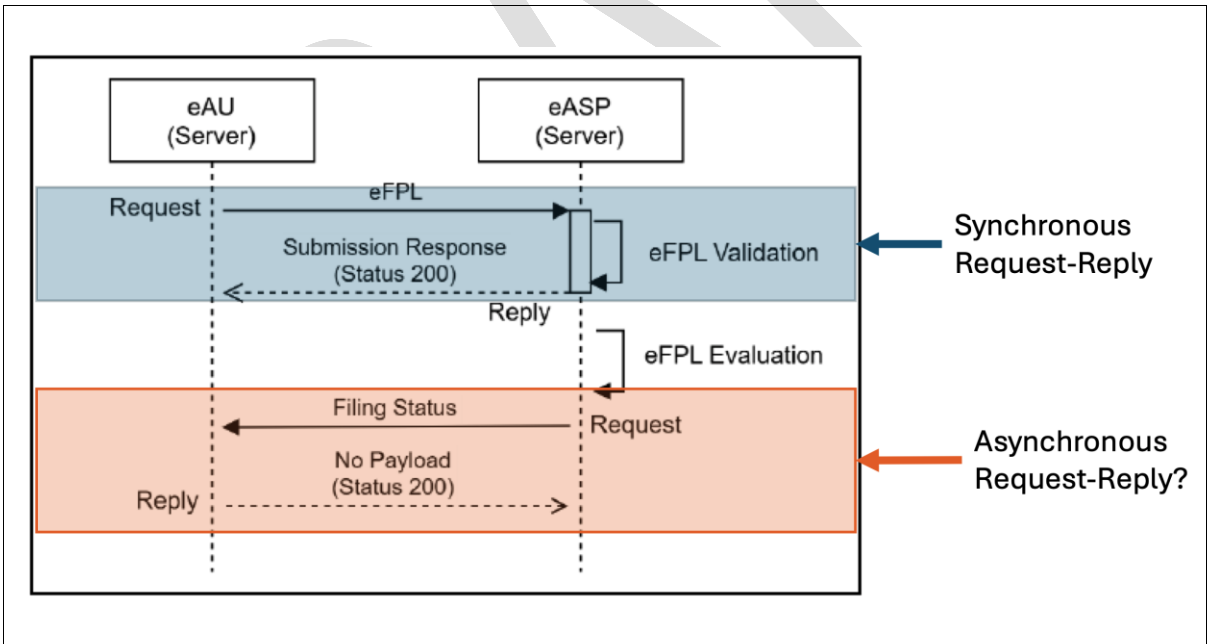


Diagram X. Diagram

From a **technical perspective**, when a requester sends a message and immediately receives a system-level response (e.g., status code with no payload), this exchange may be classified as **synchronous**. The requester technically obtains a reply within the same transaction, even if that reply does not contain the final outcome of the request.

From a **business perspective**, however, the true result of the request (e.g., validation outcome, processing status, or evaluation result) is delivered at a later stage. The requester must therefore rely on additional asynchronous mechanisms—such as callbacks, notifications, or correlation with a subsequent message—to complete the intended business process. In such cases, the service behavior is effectively **asynchronous**, because the requester cannot proceed with its operational workflow until the deferred reply is received.

This duality highlights that:

- At the **technical-level**, an immediate acknowledgment can be interpreted as synchronous.
- At the **business-level**, the process may still be asynchronous if the final response is decoupled from the initial request.

To make it clear and reduce the nuisance triggered by the confusion mentioned above, when designing or documenting R/R MEP in SWIM, it is important to clearly differentiate between **technical-level of synchronization** and **business-level of synchronization**, to avoid misinterpretation and ensure consistent implementation across different systems and stakeholders.

4.5. Clarification of Synchronous and Asynchronous R/R MEP

To resolve the confusion described in the previous section, it is necessary to establish a clear basis for distinguishing between synchronous and asynchronous R/R MEP.

From a **technical perspective**, any immediate acknowledgment could be appeared as a synchronous, since the requester receives a response without delay. However, this does not always reflect the completion of the underlying business process.

Therefore, for the purpose of SWIM implementation in the Asia/Pacific region, the classification of R/R MEP shall be **defined from the business perspective**:

- If the requester can complete its intended **business operation** immediately upon receiving the reply, the interaction is considered **Synchronous R/R**.

- If the requester must wait for an additional message or deferred processing result in order to complete its **business operation**, the interaction is considered **Asynchronous R/R**.

<table>
<tr><th>Classification</th><th>Case/Description</th><th>Technology</th></tr>
<tr><td>Synchronous Response</td><td>ACK + Final Result Payload
Returns an acknowledgment together with the final result (payload).
No further response is expected.)</td><td>HTTP-based Synchronous R/R
HTTP 200 OK + payload result</td></tr>
<tr><td rowspan="2">Asynchronous Response</td><td>ACK + Partial / Meta Payload
Returns an acknowledgment along with some meta information or partial result (payload).
The result will be delivered later through the same or a different channel.</td><td rowspan="2">- HTTP-based Asynchronous R/R
HTTP 200 Accepted → HTTP later callback
- AMQP-based Asynchronous R/R
AMQP ACK (Accepted) → asynchronous reply via reply-to queue with matching correlation-id
- Hybrid Asynchronous R/R
HTTP 202 Accepted → final result via MQ (reply queue)</td></tr>
<tr><td>ACK Only
Simply acknowledges that the request has been accepted
The actual result will be sent asynchronously via a designated reply mechanism (e.g., callback endpoint, reply queue, topic, or event).</td></tr>
</table>

By adopting this **business-oriented definition**, system designers and stakeholders can avoid misinterpretation caused by protocol-level acknowledgments and instead align the classification of R/R MEP with operational reality. This ensures that the design of SWIM services, including error handling, correlation mechanisms, and user expectations, is consistent with the actual business workflows they are intended to support.

Table X provides examples that bridge the business and technical perspectives of **asynchronous** R/R MEP. It highlights that a transaction may appear synchronous at the protocol level, yet still behave asynchronously from a business process standpoint.

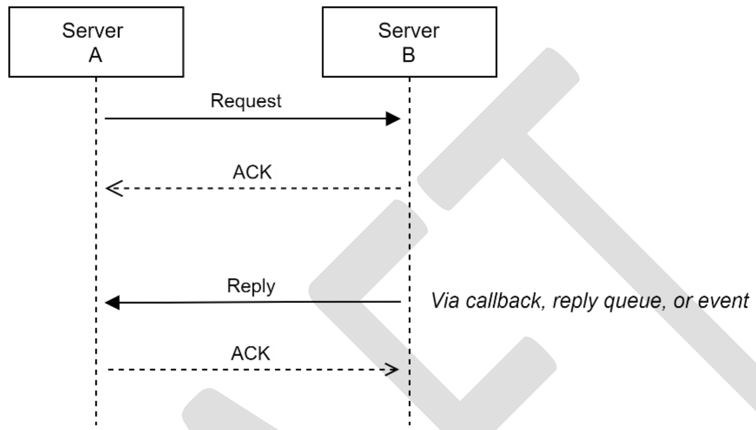
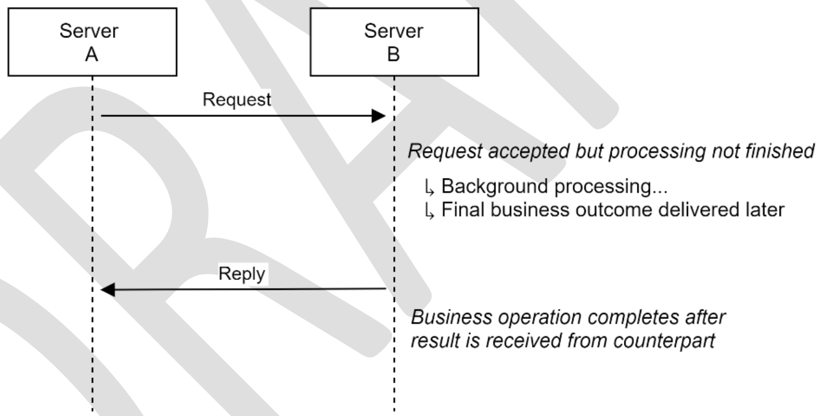
Asynchronous R/R in technical aspect	
Asynchronous R/R in business aspect	

Table X. Table

5. Implementation

This chapter explains practical methods that could be adopted to implement the R/R MEP in the SWIM environment. It describes technical options for both synchronous and asynchronous interactions, evaluates their suitability for Asia/Pacific SWIM, and provides regional approach.

5.1. Implementation of Synchronous R/R MEP

Synchronous R/R interactions can be effectively implemented using lightweight, stateless web service technologies as follows:

a. REST API (HTTP/HTTPS)
- A requester sends an HTTP request (e.g., GET, POST, or DELETE) and waits for an immediate reply from counterpart
- The response includes both status code (e.g., HTTP status code 200 - OK, HTTP status code 400 - Bad Request) and, a payload (e.g., JSON, XML)
- RESTfAPI represents the de-facto for synchronous R/R interactions in the ICT industry
- This approach aligns with the APAC SWIM regional strategy, as REST is widely supported, interoperable, and well-suited for cross-domain information exchange

b. SOAP (HTTP/HTTPS)
- SOAP provides a rigid XML-based messaging protocol and was historically used for enterprise-level synchronous R/R.
- However, as stated in previous chapters, although SOAP also could be enabled using HTTP/HTTPS, unlike REST, SOAP is not recommended to use due to its complexity, high overhead, and limited scalability due to its constraints as follows:
  - Only XML is supported, other data format like JSON is not supported
  - Requires WSDL (Web Services Description Language) for service definitions, which adds complexity at the initial setting;
  - SOAP's components such as envelop have an overhead, but a bandwidth of CRV is one of the major issues in the Asia/Pacific region;

c. Other Protocol Bindings
- While synchronous R/R can also be implemented over other protocols (e.g., gRPC, GraphQL, even AMQP), these remain optional and are not stated at the IGD at the regional level.

5.2.Implementation of Asynchronous R/R MEP

Asynchronous R/R interaction is not directly related to whether processing can be completed instantly or not, or whether replies must be deferred. Implementation options could be as follows:

a. REST API (HTTP/HTTPS) with Asynchronous Callback
- A requester sends an HTTP request and receives an immediate acknowledgment (status code).
- The actual business result is delivered later via an HTTP callback endpoint provided by the requester. This solution (a) is the callback mechanism that is limited to HTTP/HTTPS-based endpoints;
- This requires correlation mechanisms to match replies with original requests.

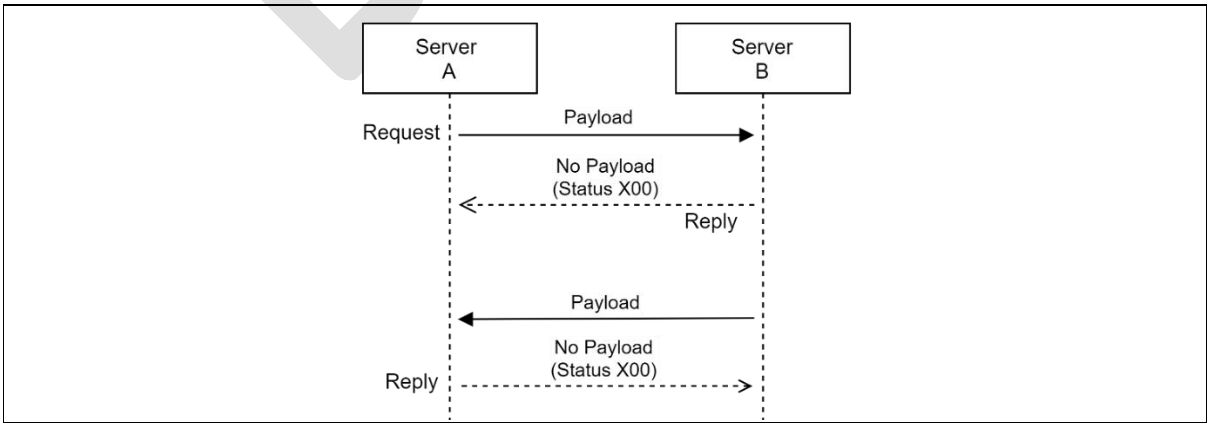


Diagram X. Diagram

b. REST API (HTTP/HTTPS) with Polling
- The requester periodically polls the service to check the status of its request.
- While simple, polling may cause inefficiencies in bandwidth usage and should be applied only for services with low response frequency.

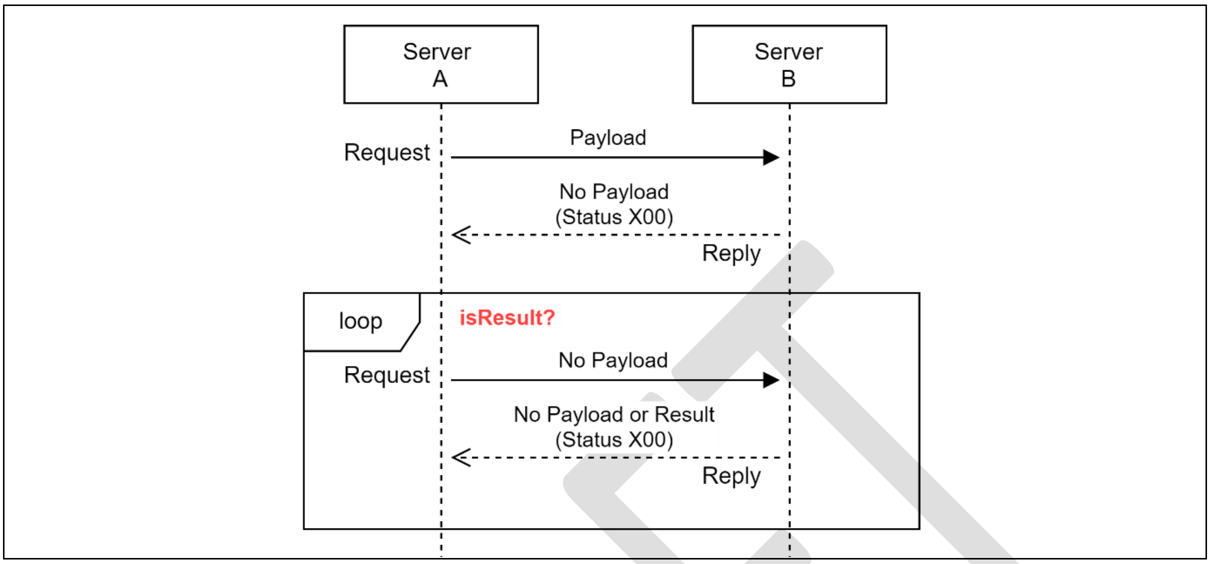


Diagram X. Diagram

c. Message Queue
- AMQP or MQTT can be used to implement asynchronous R/R interactions in distributed environments.
- AMQP properties (e.g., correlation-id, reply-to) must be configured to send request a query, and the message broker uses or creates shared or exclusive queue to handle the R/R MEP.
- Counterpart processes a request query, generates a corresponding reply using the same correlation-id, and returns it to the queue or topic specified in the reply-to property.

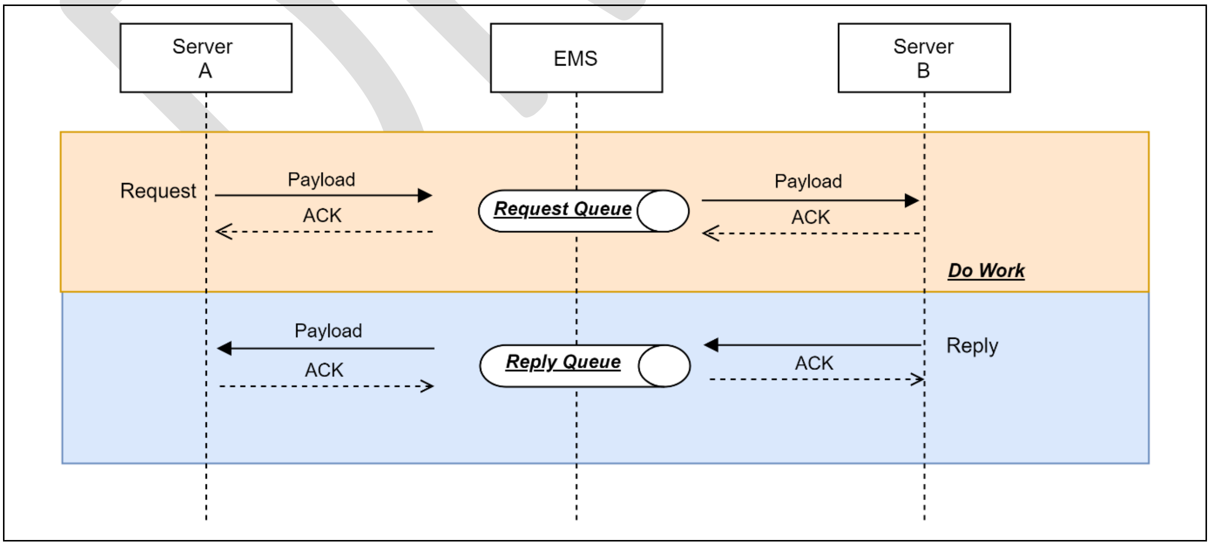


Diagram X. Diagram

d. REST API–Message Queue Bridge

- Most modern message brokers provide a REST API interface. A gateway service, either built into the broker or deployed as a standalone component, performs protocol conversion between AMQP and HTTP.
- This provides a REST API interface externally while using asynchronous message queue–based message delivery internally. Internal message delivery is the same as described in the (c)

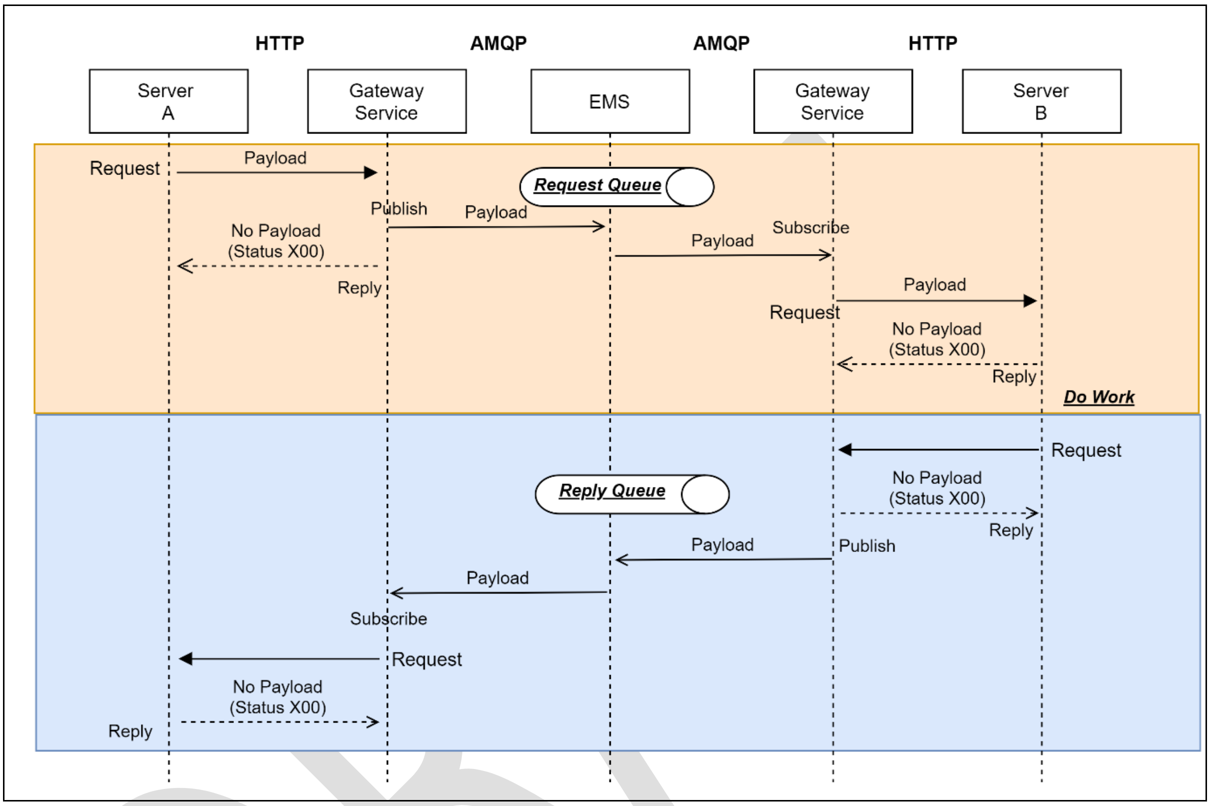


Diagram X. Diagram

a. Synchronous REST with Asynchronous Event Delivery using Message Queue

- This pattern is asynchronous R/R MEP implementation of Network Manager (NM), Eurocontrol. The pattern is composed of two parts:
  - Synchronous part: The server sends HTTP request, and the counterpart returns reply. If the status of the information reply is OK, the server extracts the information on how to consume the asynchronous reply message, the pattern continues with the asynchronous part
  - Asynchronous part: The server connects to the broker and consumes the asynchronous reply message
- This pattern allows integration of REST API with asynchronous processing workflows using message queue

Click link to see more about asynchronous R/R MEP of NM, Eurocontrol - NM Release Notes

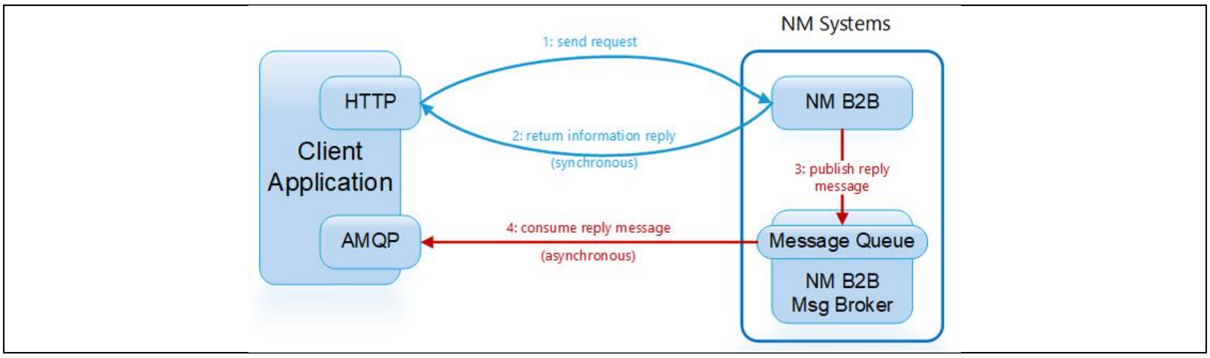


Diagram X. Eurocontrol Network Manager 26.0 Release Note ed.5

5.3.Approach on Implementation of R/R MEP in Asia/Pacific Region

For the Asia/Pacific SWIM environment, the following implementation approaches could be explored and adopted to ensure harmonization and interoperability among diverse stakeholders:

Synchronous R/R MEP – REST API via HTTP/HTTPS

For synchronous Request/Reply MEP, ***REST API via HTTP/HTTPS*** could be considered

Appendix X shows the sequence diagram of synchronous R/R MEP using REST API via HTTP/HTTPS

Asynchronous R/R MEP – REST API via HTTP/HTTPS with Asynchronous Callback

For asynchronous interactions, **1) *REST API* with asynchronous callback, 2) Message Queue 3) Synchronous REST API with Asynchronous Event Delivery using Message Queue** could be considered.

Appendix X shows the sequence diagram of 1) REST API via HTTP/HTTPS with asynchronous callback, 2) Message Queue-based Asynchronous R/R via AMQP, MQTT

In some cases, to support its business operation, information exchange may involve a combination of **Synchronous R/R MEP**, **Asynchronous R/R MEP**, and **P/S MEP** patterns. However, this document doesn't describe it in detail.

6. Regional R/R MEP Architecutre

This chapter outlines the topology of the R/R MEP. Specifically, it explains which R/R MEP solutions could be considered, and what architecture could be used to deploy those products or solutions to enable both synchronous and asynchronous R/R MEP within the region.

6.1.Introduction of R/R MEP Solution

a. Forward/Reverse Proxy
   - A forward proxy acts on behalf of an internal resource (e.g., client, server, or system), managing outbound requests to an external resource (e.g., client, server, or system) to provide capabilities such as access control, caching, and monitoring.
   - A reverse proxy acts on behalf of an external resource (e.g., client, server, or system), managing inbound requests to internal resource (e.g., client, server, or system) to security, load balancing, and routing, while hiding internal system details.
   - Comparison of forward and reverse proxy is as follows:

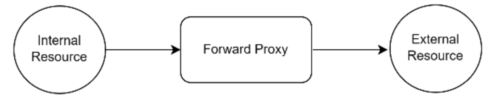
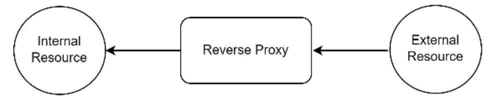
Aspect	Forward Proxy	Reverse Proxy
Diagram		
Traffic Direction	Handles outbound requests from internal resource to external resource	Handles inbound requests from external resource to internal resource
Visibility	Hides the internal resource's identity from external resource	Hides the internal resource's identity to external resource

Table X. Table

- Forward and reverse proxy supports OSI 3rd, 4th, and 7th layer protocols such as HTTP, Web-socket, TCP, UDP, IP. Main focus of forward and reverse proxy is message routing.

b. API Gateway
- An API Gateway (GW) is built on top of a reverse proxy, primarily supporting HTTP and providing advanced API management capabilities. The main difference between an API GW and a reverse proxy lies in how policies are applied and managed from an API management perspective.

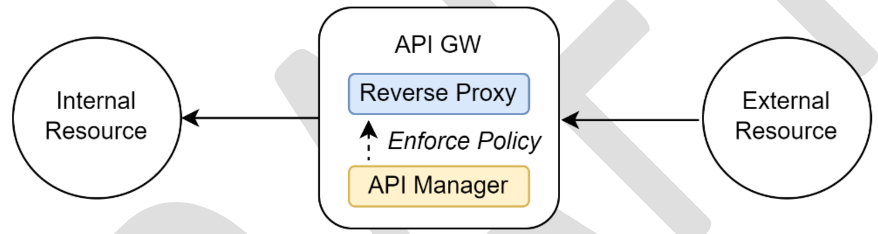


Diagram X. Diagram

6.2.Candidate Architecture using API GW or Forward/Reverse Proxy

This section describes candidate architectures that could be applicated for synchronous and asynchronous R/R MEP using REST API (HTTP)

a. Full Mesh Architecture
- Same as Decentralized Approach of EMS interconnectivity architecture presented in WP05,SWIM/TF8. A service interacts with its own Edge API GW, and the Edge API GW directly forwards the request to another Edge API GW.

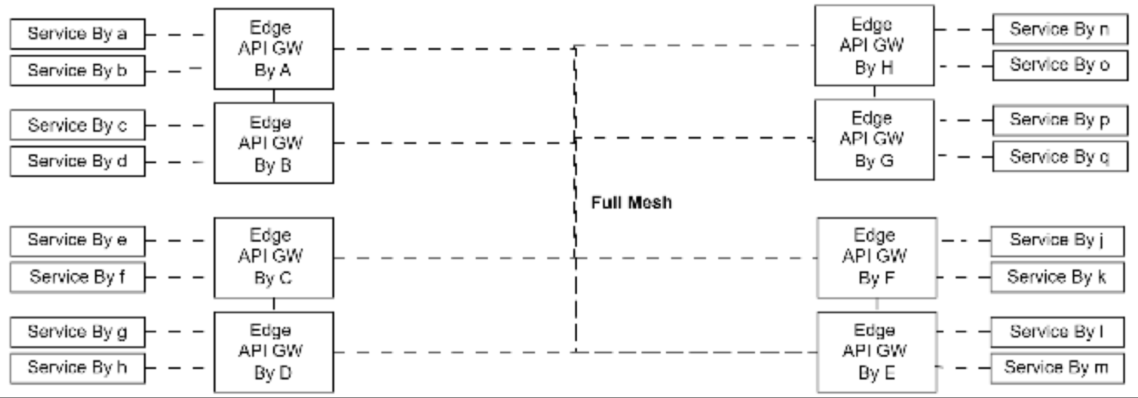


Diagram X. Diagram

b. Centralized Architecture

- Same as Centralized Approach of EMS interconnectivity architecture presented in WP05, SWIM/TF8 API GW acts as a single-entry point for all services between member states

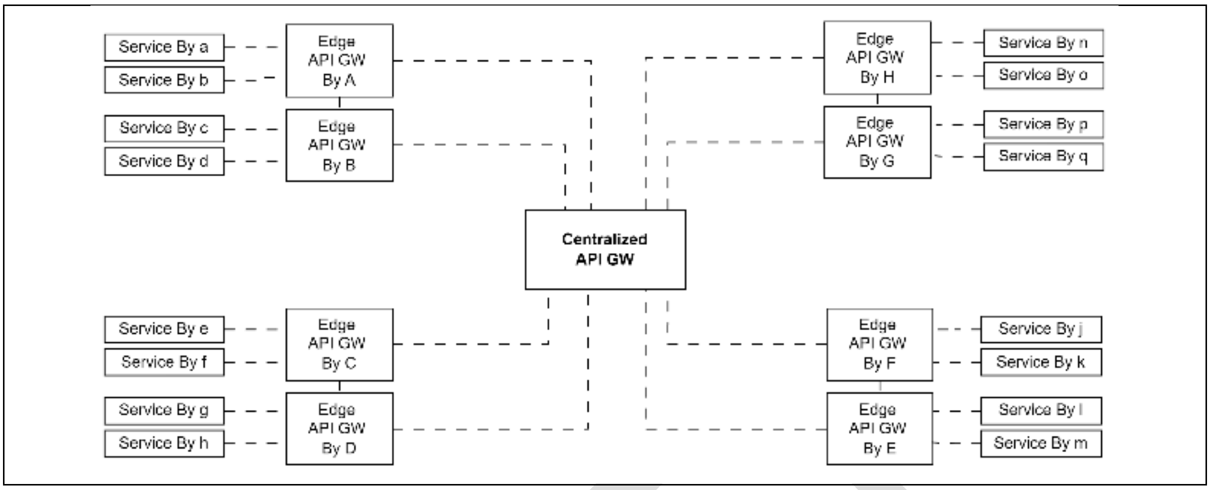


Diagram X. Diagram

c. Hierarchical Architecture
   - Same as Hierarchy Approach of EMS interconnectivity architecture presented in WP05, SWIM/TF8. Edge API GW interacts with Regional API GW, which handle routing within sub-community or forward requests to other sub-community.

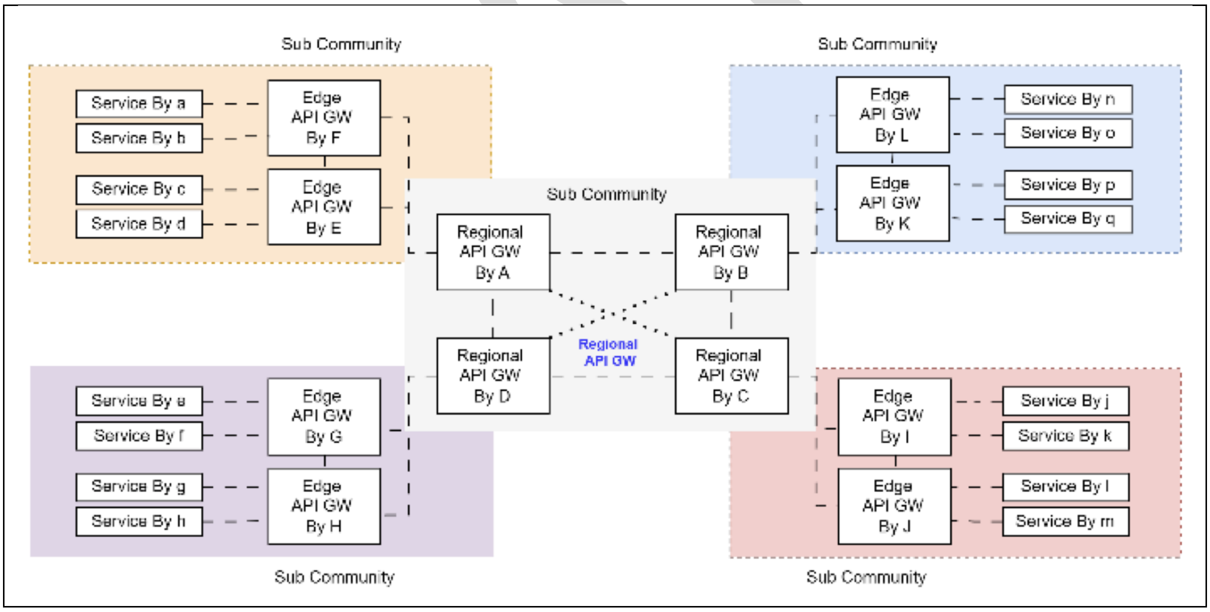


Diagram X. Diagram

d. Two-layered Hierarchical Architecture
   - Same as Modified Hierarchy Approach of EMS interconnectivity architecture. presented in SP07 SIPG/WS8. Two-layer hierarchical API GW architecture with a clustered regional layer

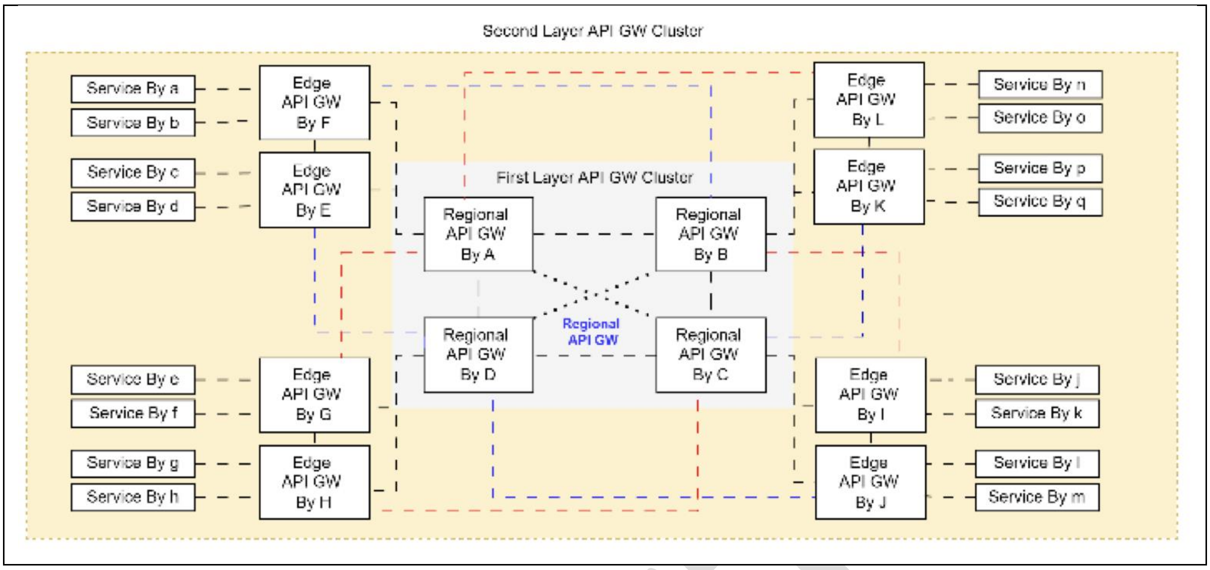


Diagram X. Diagram

Comparison of candidate architectures is as follows:

Architecture	Pros	Cons
Full Mesh	- Low dependency between API GWs - Easy to implement - High Sovereign	- All API GW should be ready for cross-border data exchange (e.g., bridge to SOAP for European region - API GW should be able to get connected with all API GW interested - All API GW should be updated when there is a policy/configuration change
Centralized	- Simple communication - Easy to configure policy	- Low Sovereign - Centralized API GW is SPOF
Hierarchical	- Efficient for cross border data exchange - A configuration change could be affected to only regional API GWs	- Regional API GW is SPOF
Two Layered Hierarchical	- Higher Fault tolerance - Better HA and scalability	- Hight cost for maintenance - Management Complexity

6.3.Candidate Architecture using Message Broker (AMQP)

This section describes the candidate architecture that could be applicable to synchronous and asynchronous R/R MEP using Message Broker (AMQP).

As modified hierarchy approach of EMS interconnectivity architecture, presented in SP07 SIPG/WS8 is information backbone for P/S MEP in the region. To enable synchronous and asynchronous R/R MEP using Message Broker (AMQP), same architecture is better to be used.

As of Oct. 25, Implementation of two layered hierarchical architecture (i.e., APAC SWIM P/S MEP architecture) is in progress under SIPG Task 1 and Task2.

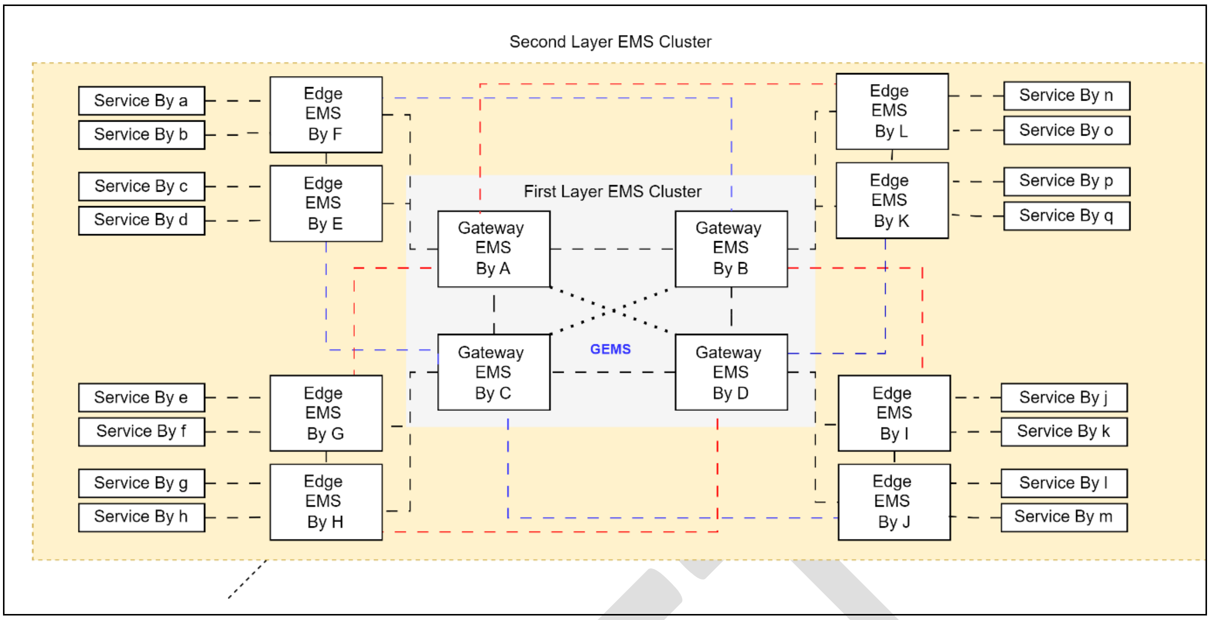


Diagram X. Diagram

6.4.Approach on Topology in Support of R/R MEP in Asia/Pacific Region

For the Asia/Pacific SWIM environment, the following solution and architecture could be explored in support of R/R MEP in the region before adoption.

For synchronous Request/Reply MEP***, 1) API GW + Full Mesh Architecture*** *or* ***2) API GW + Two Layered Hierarchical Architecture*** *could be considered*

For asynchronous Request/Reply MEP***, 1) API GW + Full Mesh Architecture*** *or* ***2) API GW + Two Layered Hierarchical Architecture****, and* ***3) Message Broker + Two Layered Hierarchical Architecture (i.e., APAC SWIM P/S MEP Architecture (co-use))*** *could be considered*

7. Routing Mechanism

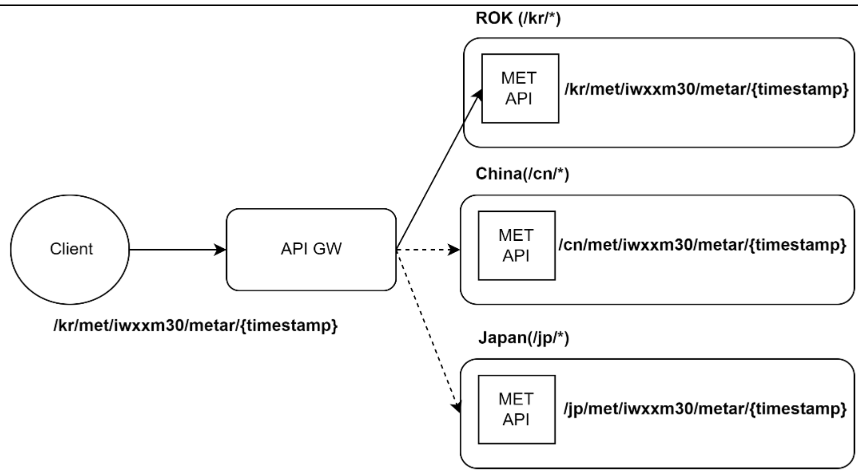
This chapter introduces routing mechanisms for R/R MEP. In R/R interactions, a message sent by a requester must be delivered to the correct replier through the API GW or message broker. Therefore, routing mechanism plays a critical role in ensuring that requests are directed to the right service and that responses are returned correctly. There are many different routing mechanisms for R/R MEP such as Path-based Routing, Content-based Routing, Header-based Routing, Policy-based Routing, and so on, but this chapter only describes Path-based Routing, and Contents-based Routing. They are mentioned as they represent the most fundamental and widely applicable approaches. These two mechanisms provide a clear contrast between simplicity and flexibility: Path-based routing offers transparency and ease of configuration, while Content-based routing enables dynamic and context-driven service delivery. This could be mapped with message routing for Topic and Queue in Pub/Sub MEP.

7.1. Routing Mechanisms

7.1.1. Path Based Routing

Path-based routing is the simplest and most commonly used routing mechanism. In this approach, the Uniform Resource Identifier (URI) path or topic within the HTTP or AMQP request determines the destination of the message.

HTTP-based R/R MEP



AMQP-based R/R MEP

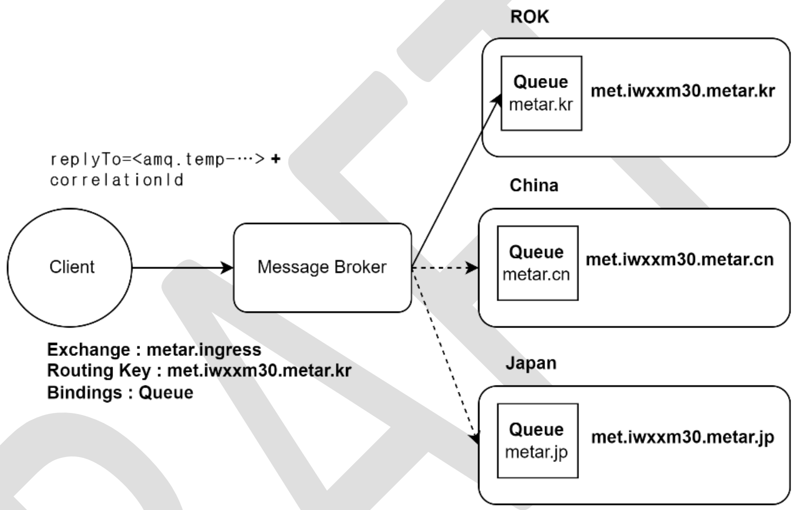


Diagram X. Diagram

a. **Mechanism:** The requester specifies the region, service name, or function within the routing path or topic naming structure (e.g., (HTTP) /kr/met/iwxxm30, /jp/met/iwxxm30, or (AMQP) met.iwxxm30.metar.kr).

b. **Advantages:**

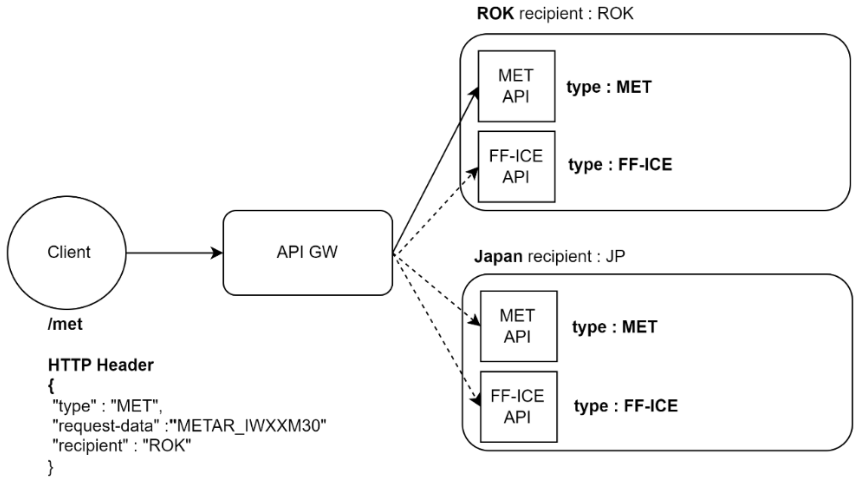
- Easy to implement and configure
- High clarity and transparency, since routing is explicitly defined in the URL
- Works well with both HTTP-based gateways and messaging systems that support topic-based or header-based routing.

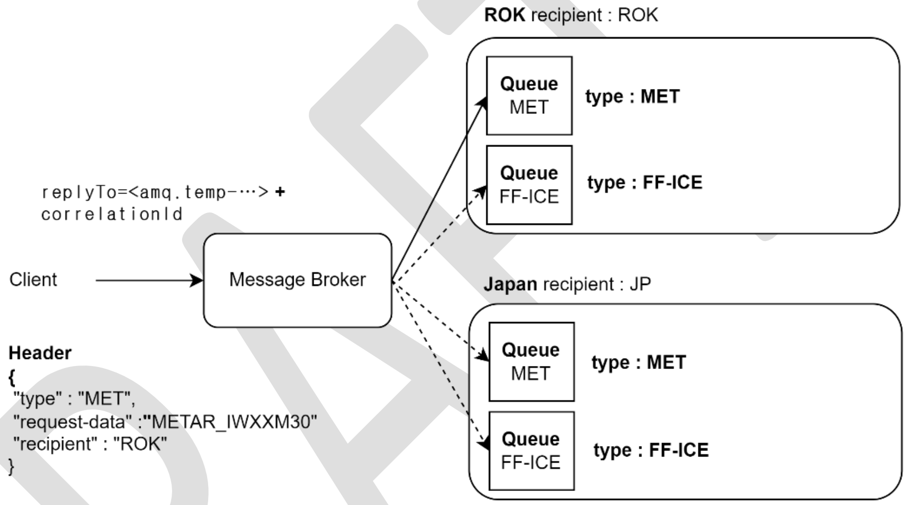
c. **Disadvantages (Consideration):**

- Need to have commonly agreed naming convention for the routing path and topic
- Front-facing path routing structure (e.g., URL or topic schema) need to be carefully designed; backend endpoints remain abstracted from requestor

7.1.2. Contents Based Routing

Content-based routing provides more flexibility by making routing decisions based on the message content (headers, parameters, or payload), rather than just the routing path or topic.

<table>
<tr>
<td>HTTP-based
R/R MEP</td>
<td>


</td>
</tr>
<tr>
<td>AMQP-based
R/R MEP</td>
<td>


</td>
</tr>
</table>

Diagram X. Diagram

a. **Mechanism:** The requester sends a generic request while including additional routing information in headers or message body (e.g., type, request-data, and recipient). The API GW or message broker inspects the content and dynamically determines the correct destination.

b. **Advantages:**

- High flexibility in handling dynamic services
- Decouples clients from internal routing logic; clients only need to know the front-facing service address, not the exact backend service.
- Supports complex service ecosystems where routing depends on data attributes (e.g., airspace ID, flight identifier)

c. **Disadvantages (Consideration):**

- More complex to configure and manage.
- Potential security risks such as injection attacks (e.g., malicious payloads in headers or message body) must be carefully mitigated through validation, sanitization, and strict policy enforcement.

- Inspection of the full message body is generally not permitted under ICAO provisions (e.g., Annex 15). Therefore, routing decisions should primarily rely on HTTP or AMQP headers or query parameters, rather than deep inspection of the payload.
- Routing decisions should therefore rely on standardized HTTP or AMQP headers or query parameters, rather than payload inspection.
- A commonly agreed set of routing-related headers should be established at the regional level to ensure consistency.

7.2.Approach on Routing Mechanism of R/R MEP in Asia/Pacific Region

For the Asia/Pacific SWIM environment, the following routing mechanism could be explored:

Path- or Topic-Based Routing (PBR)

Path-Based Routing is both applicable to HTTP-based API GW and AMQP-based message broker more consistent with the principles of REST API or topic design architecture

8. Any Other Considerations

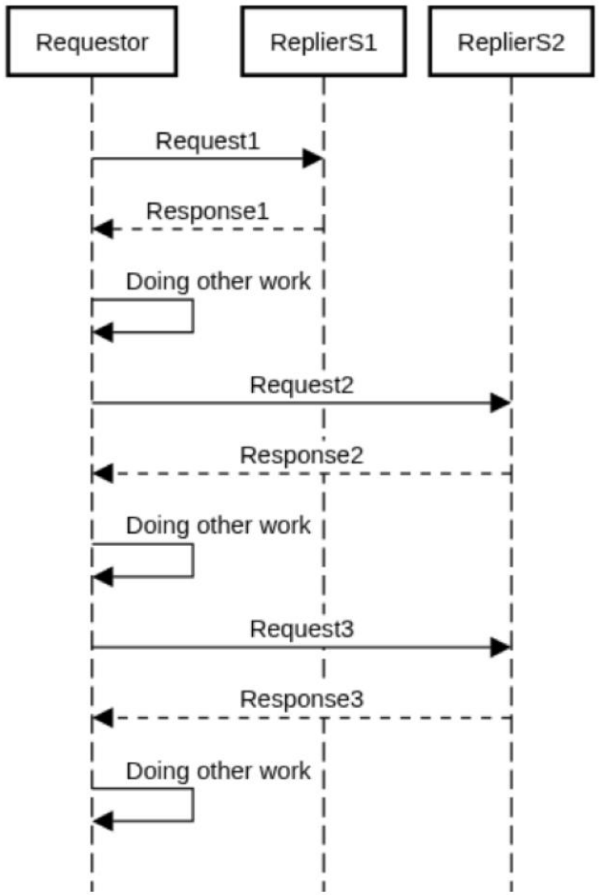
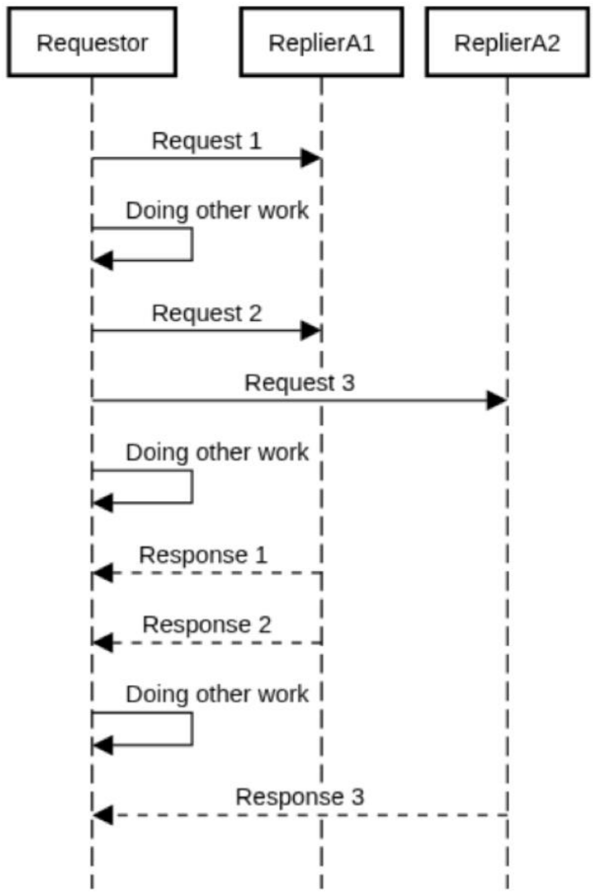
This document proposes approaches to implementing R/R MEP in the APAC region. Proof of Concept (POC) will be conducted to evaluate and select specific technologies after the POC is completed.

This document does not consider interoperability of R/R MEP between regions; this will be addressed in a later phase.

It also does not define the requirements for the API Gateway or the message broker to enable R/R MEP; these will be defined in a separate document.

9. Annexes

9.1. Annex 1 – Synchronous and Asynchronous R/R Data Flow Diagram

<table>
<tr><th>Synchronous R/R</th><th>Asynchronous R/R</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

9.2.Annex 2 – Applicability of Synchronous and Asynchronous R/R MEP to FF-ICE Service

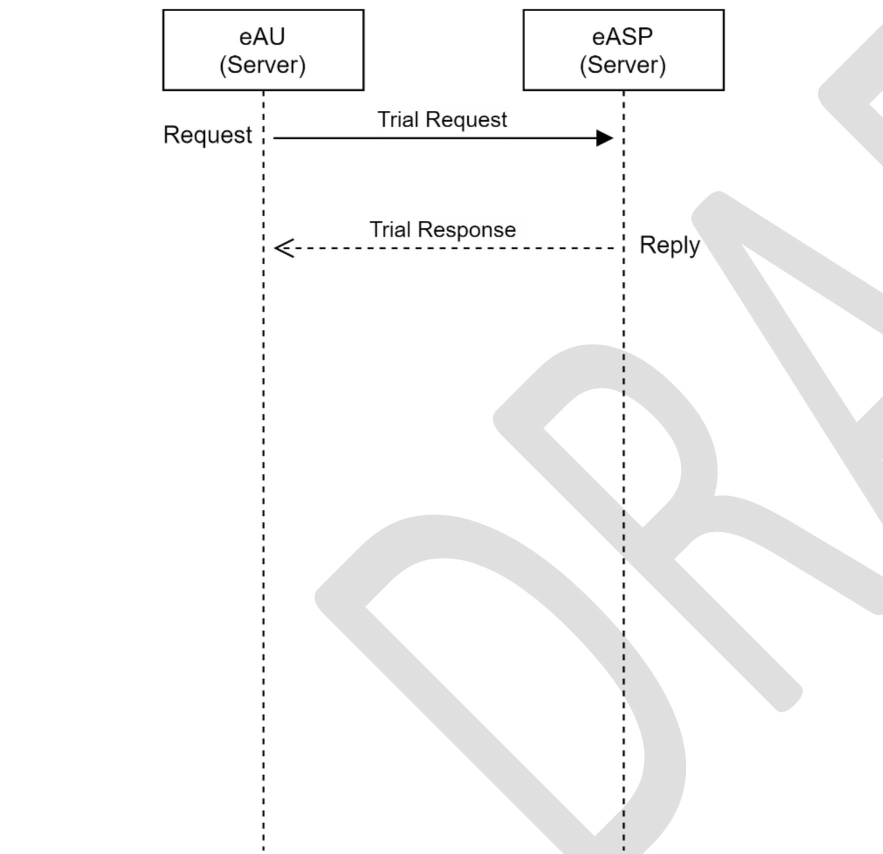
Service	MEP	Synchronous R/R	Asynchronous R/R	Note
GUFI Service	R/R	O	O	
FF-ICE Filing Service	P/S, R/R	X	O	Separate messages ought to be responded from replier (Submission Response + Filing Status)
FF-ICE Data Publication Service	P/S	-	-	
FF-ICE Trial Service	R/R	O	O	
FF-ICE Flight Data Request Service	R/R	O	O	
FF-ICE Notification	P/S, R/R	O	O	
FF-ICE Planning Service	P/S, R/R	X	O	Separate messages ought to be responded from replier (Submission Response + Planning Status)

* FF-ICE services mentioned above are defined in the APAC SWIM Common Services (APAC Common SWIM Information Services, WP11, SWIM TF/10)

9.3. Annex 3 – FF-ICE Service Data Flow Diagrams of Synchronous and Asynchronous R/R MEP

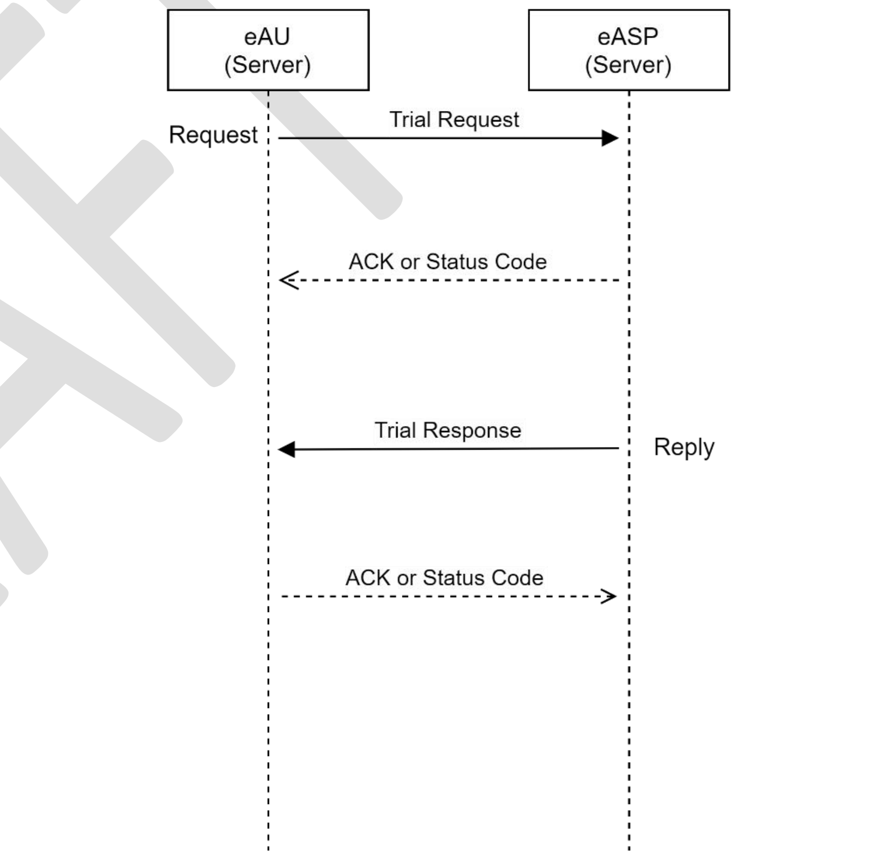
- FF-ICE Trial Service

Synchronous R/R (HTTP)



* API GW and Message Broker are not Depicted

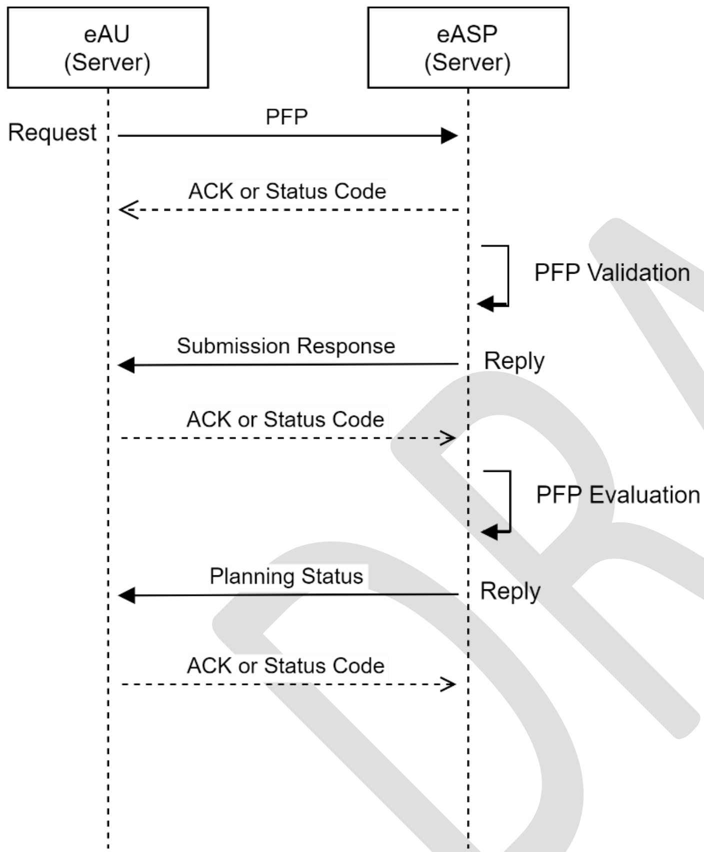
Asynchronous R/R (HTTP or AMQP)



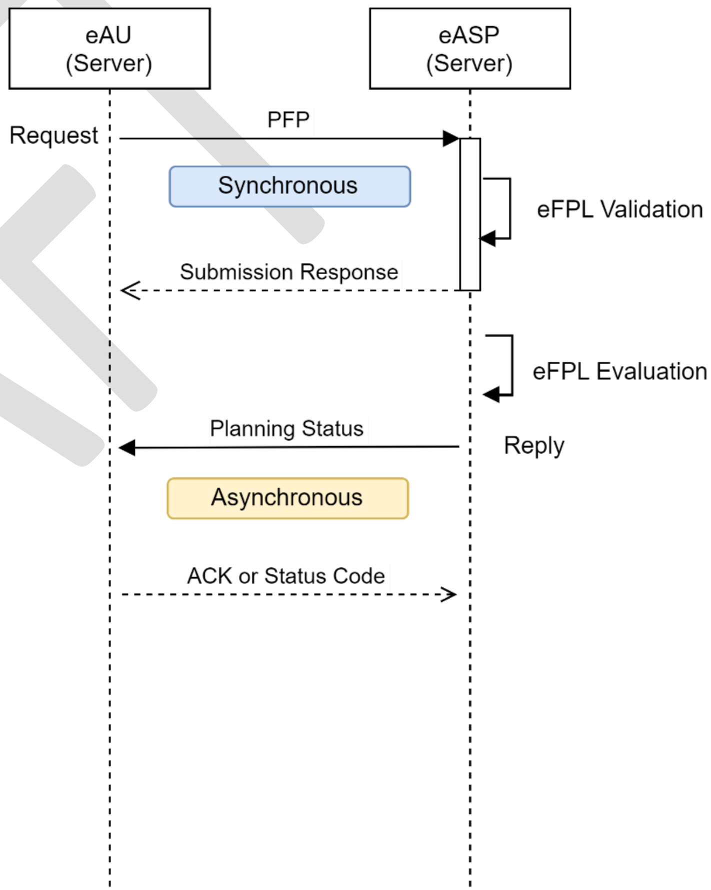
* API GW and Message Broker are not Depicted

- FF-ICE Planning Service

Asynchronous R/R (HTTP or AMQP)	**Synchronous R/R (HTTP) + Asynchronous R/R (HTTP or AMQP)**

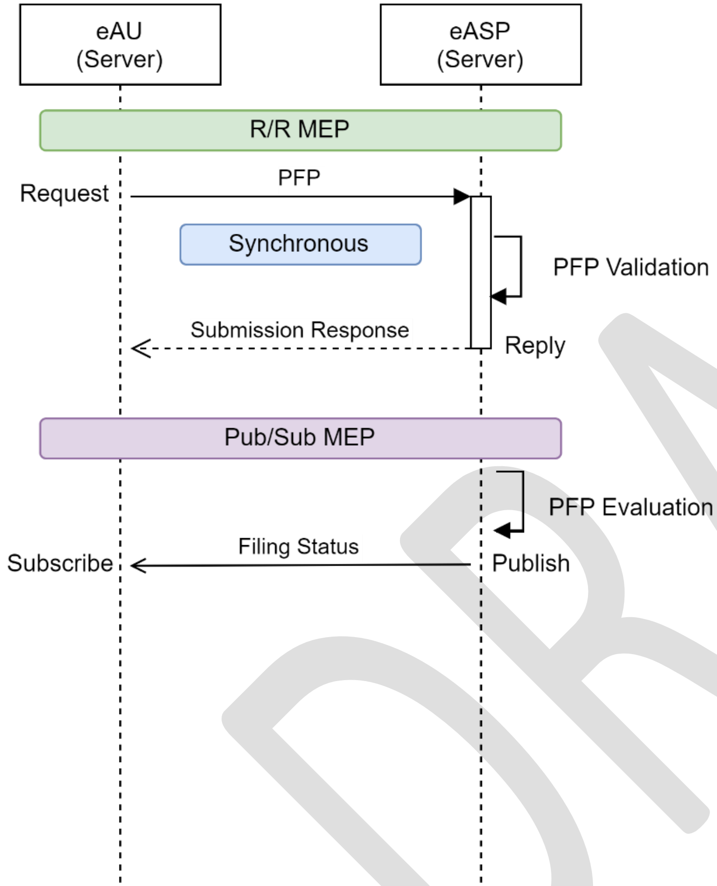


* API GW and Message Broker are not Depicted



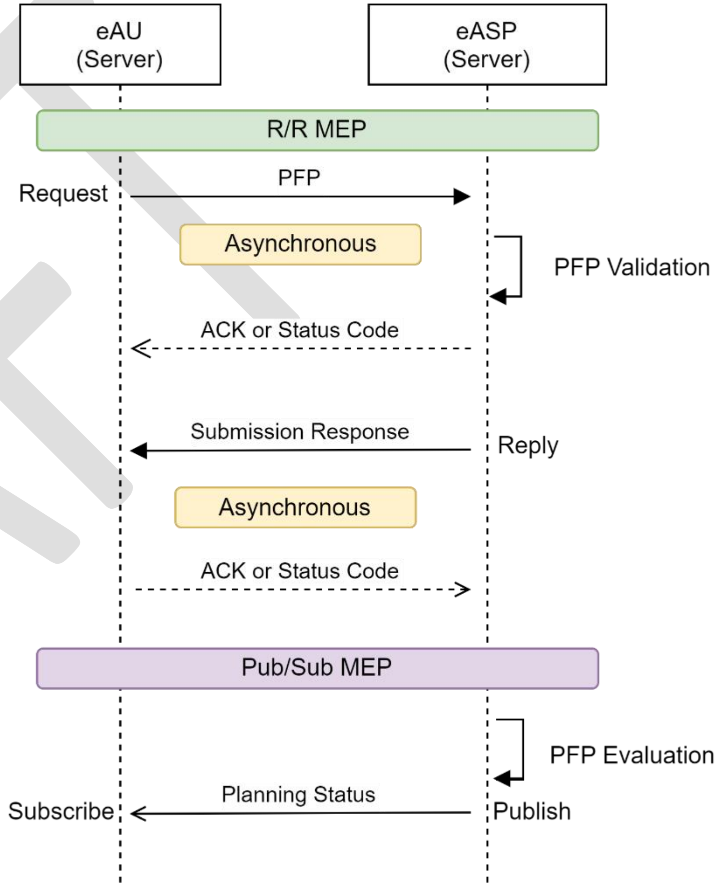
* API GW and Message Broker are not Depicted

Synchronous R/R (HTTP) + Pub/Sub



* API GW and Message Broker are not Depicted

Asynchronous R/R (HTTP or AMQP) + Pub/Sub



* API GW and Message Broker are not Depicted