International Civil Aviation Organization

The Ninth Meeting of System Wide Information Management Task Force (SWIM TF/9)

Bangkok, Thailand, 14 – 17 May 2024

Agenda Item 5: Updates on the assigned tasks by task leads/contributors, including progress report and issues

Consideration on Guaranteed Message Delivery for Regional SWIM Architecture

(Presented by SIPG, presenter ROK)

SUMMARY

This paper presents a consideration on guaranteed-message delivery in regional SWIM architecture (i.e., a hierarchical architecture). This paper is to introduce vulnerabilities which possibly break guaranteed message delivery in a hierarchical SWIM architecture in the APAC region, and considerations to ensure reliable and guaranteed message delivery.

1. INTRODUCTION

1.1 At the 8th SWIM TF meeting in 2023, a hierarchical architecture was proposed to avoid the issue of having a single point of failure present in the centralized approach while at the same time avoiding the case of a very complex topology in the decentralized approach. *Proposal for detailed Enterprise Messaging Service architecture and its impact on the use of message headers – Japan, Singapore, and Thailand (WP/05)*

1.2 SWIM Implementation Pioneering Group (SIPG) was requested to undertake the task of defining the message header format and contents. And, a hierarchical architecture is implemented to support SWIM over CRV demonstration and surveillance data over SWIM trial to be held in May 2024.

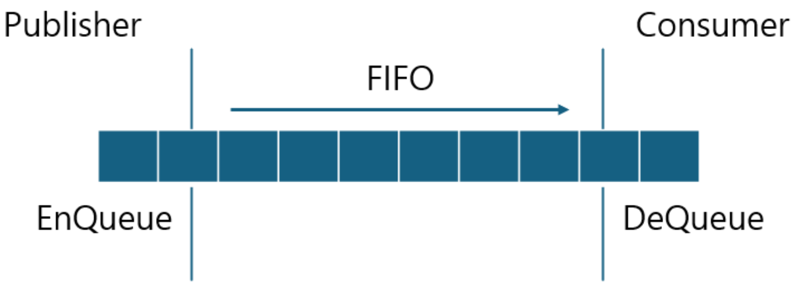
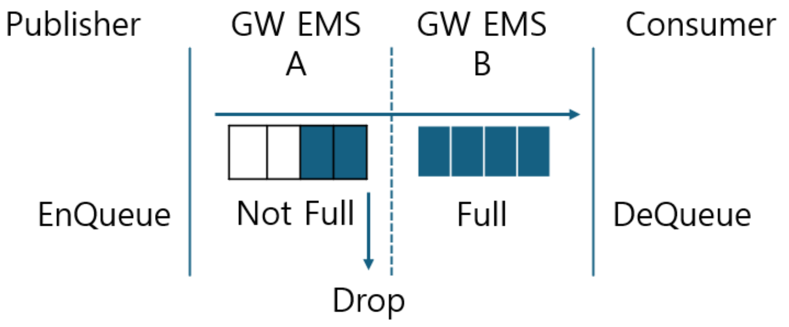
1.3 The hierarchical architecture is an implementation of the edge or fog computing, and through it, this could enable efficient usage of bandwidth on a CRV network and prevent the propagation of failures. However, potential architectural vulnerabilities have been identified (e.g., guaranteed message delivery failure due to the improper failover or message handling during partitioning phase) during the implementation of the hierarchical architecture. These vulnerabilities could impair reliable messaging, (i.e., provides support for various types of guarantees for message delivery) a core capability defined in the SWIM ConOps (Doc. 10039).

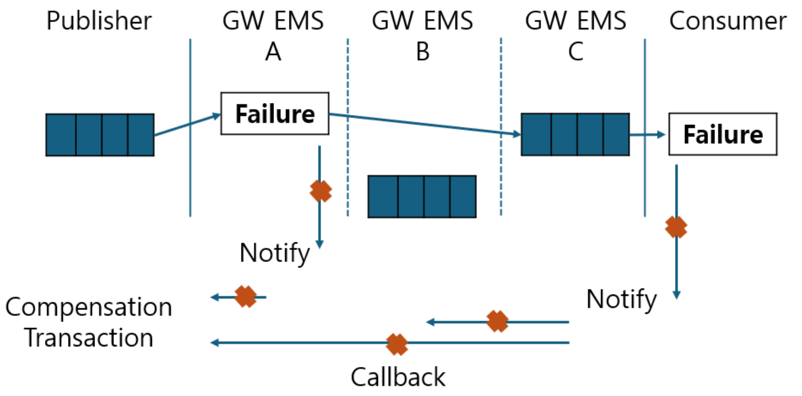
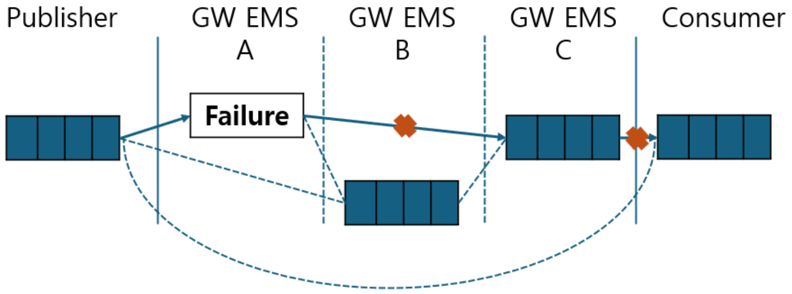
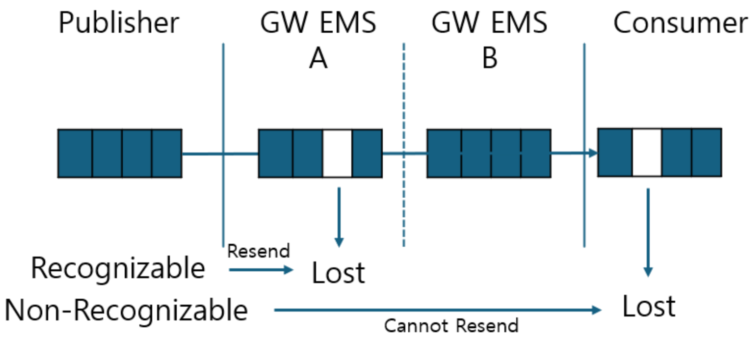
1.4 Guaranteed message delivery is a capability commonly provided by Commercial, off-the-shelf (COTs) or Open Source (OS) message broker (e.g., both Solace Pub/Sub platform and RabbitMQ provide federation capability between homogeneous message brokers), but in a hierarchical architecture in the APAC region, this is not applicable as it does not force to the use of a specific message broker. In the practical implementation of the architecture, heterogeneous message brokers are adopted by states in SIPG.

1.5 This paper describes architectural vulnerabilities identified during the implementation of the hierarchical architecture in terms of guaranteed message delivery, and benchmarks other implementations of guaranteed data or message delivery in Open Systems Interconnection Reference Model (OSI) 7 layers and elicits considerations in aspect of technical and business solution.

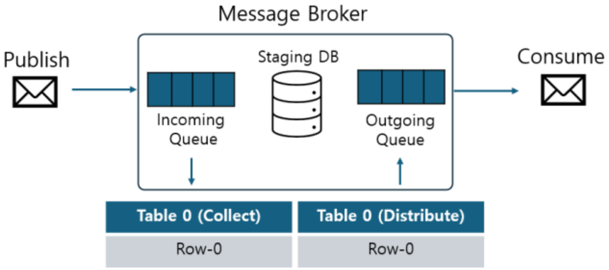
2. DISCUSSION

2.1 Substantial architectural vulnerabilities identified during the implementation of the hierarchical architecture in terms of guaranteed message delivery are as follows:

Vulnerabilities	Use-Case / Rationale / Impact
Priority messaging is not possible depending on the importance of the information	When congestion occurs in the network or messaging level, message delivery cannot be prioritized 
	In a single queue, messages are delivered in a FIFO manner regardless of the importance of the message
	Assuming that the importance of the FF-ICE message is higher than the surveillance message, if the FF-ICE message and surveillance message are delivered without priority using a single queue, when congestion occurs, FF-ICE message would be processed equally, and this could cause a delay in delivering of FF-ICE message
Guaranteed message delivery is destroyed when malfunctioning of a message broker occurs in the message delivery chain	When the message queue is full or shutdown, the message is dropped  Publisher GW EMS A GW EMS B Consumer EnQueue Not Full Full DeQueue Drop
	When the message broker reaches the maximum number of messages, it drops subsequent messages or deletes messages using the FIFO method
	Assuming an environment where network delay occurs, if the publisher publishes the message at 100msg/s and the consumer consumes the message at 50msg/s, the message queue becomes full and message omission occurs

Vulnerabilities	Use-Case / Rationale
Compensation transactions cannot be performed to compensate transaction failure in the message delivery chain	Ensuring integrity of respective messages are not achievable without a compensation transaction  Publisher GW EMS A GW EMS B GW EMS C Consumer Failure Failure Notify Notify Compensation Transaction Callback
	The publisher can know a failure from the message broker that directly published the message (e.g., Edge to GW), but is not aware of failures that occurred in the message broker afterwards (e.g., GW to GW, GW to Edge) if there is no notification or trigger.
Detouring cannot be performed if a failure occurs in the message delivery chain	The publisher is not able to change delivery responsibility even if the publisher recognizes a failure in the message delivery chain.  Publisher GW EMS A GW EMS B GW EMS C Consumer Failure Detouring during the partitioning phase is non-configurable
	All edges are accessible to other edges in the CRV level, but direct access is restricted architecturally, and edges is not able to use a detour route.
The edge node does not know which message to resend when message loss occurs	The publisher does not know which message to resend  Publisher GW EMS A GW EMS B Consumer Resend Recognizable Lost Non-Recognizable Cannot Resend Lost
	In the case of a retransmitting to message broker directly connected, the publisher can specify the missing message and retransmit it, but in the case of a missing message that occurs in a subsequent message broker, the publisher cannot specify the message and try to resend it.

2.2 The vulnerabilities addressed in Section 2.1 are commonly encountered problems in other domains (e.g., network level, or messaging in the finance domain), and many different troubleshooting methods are designed as follows:

Vulnerabilities	Troubleshooting	
Priority messaging is not possible depending on the importance of the information	OSI 3rd Layer (Network)	Queueing algorithms in the OSI 3rd layer typically prioritize packets based on various factors, including Quality of Service (QoS) requirements, packet type, and destination. Queueing plays a crucial role in managing packet traffic, minimizing delays, and maintaining the quality of service.
	OSI 7th Layer (Application)	Message brokers (e.g., RabbitMQ) support message prioritization through the use of message priorities. Prioritized messaging allows to ensure that messages with higher priorities are consumed before messages with lower priorities.
Guaranteed message delivery is destroyed when malfunctioning of a message broker occurs in the message delivery chain	OSI 3rd Layer (Network)	Leaky bucket algorithm serves as a method to control the rate of data flow into the buffer, thereby managing congestion and ensuring a consistent data transfer rate.
	OSI 7th Layer (Application)	JMS Server with staging DB is one of the de-facto architectural patterns for reliable messaging. Incoming queue is only to receive message from publisher and out coming queue is only to send message to consumer. There is staging DB between incoming queue and outgoing queue, so it acts like an buffer.  Message Broker Publish Staging DB Consume Incoming Queue Outgoing Queue Table 0 (Collect) Row-0 Table 0 (Distribute) Row-0
Compensation transactions cannot be performed to compensate transaction failure in the message delivery chain	OSI 7th Layer (Application)	Saga pattern is a design pattern used in distributed systems to maintain data consistency across multiple microservices or transactions. It's particularly relevant in the context of Event-Driven Messaging (EDM) architectures, where services communicate asynchronously through events. This consistency is achieved by executing conservative transactions for failover such as pivot, compensable, retriable transaction.
Detouring cannot be performed if a failure occurs in the message delivery chain	OSI 3rd Layer (Network)	OSPF (Open Shortest Path First) is a dynamic routing protocol commonly used in large-scale enterprise and service provider networks. OSPF provides several mechanisms for creating detour routes within a network to optimize traffic flow, enhance network resilience, and mitigate congestion or failures.

Vulnerabilities	Use-Case / Rationale	
The edge node does not know which message to resend when message loss occurs	OSI 3rd Layer (Network)	Automatic Repeat reQuest (ARQ) is a communication protocol technique used to ensure the reliable delivery of data packets over unreliable communication channels. One of the primary functions of ARQ is to detect and resend lost or corrupted packets
	OSI 7th Layer (Application)	In the FF-ICE, the message type "SUB_RESP" is generated by recipients enlisted in the FF-ICE message to notify if the message is received to message originator

Conclusion

2.3 In conclusion, the implementation of a hierarchical architecture in the APAC SWIM presents promising opportunities for enhancing efficiency and reliability in message delivery. However, as highlighted in this paper, the adoption of a hierarchical architecture with heterogeneous message brokers could introduce certain vulnerabilities, especially concerning guaranteed message delivery.

2.4 Given the critical nature of SWIM operations and the need for reliable messaging, it becomes imperative to address these vulnerabilities effectively to ensure guaranteed message delivery within the hierarchical architecture. To do this, several considerations must be taken into account:

2.4.1 Standardize common logic or process for message handling;

2.4.2 Design failover mechanisms and redundancy both in technical and business aspect;

2.4.3 Design common monitoring and notification mechanism;

2.4.4 Identify abnormal use-case and conduct testing and validation; and

2.4.5 Conduct collaborative efforts and knowledge sharing;

3. ACTION BY THE MEETING

3.1 The meeting is invited to:

a) note the information contained in this paper;

b) deliberate on the proposed considerations; and

c) discuss any relevant matter as appropriate

– – – – – – – – – – – – – –