Reference Implementation Model (RIM) for the Area Management Security Use Case

Classification: APPROVED REFERENCE DOCUMENT

Confidentiality: PUBLIC

Version 1.0

1/27/2022

[RIM AM]

Legal

THIS DOCUMENT HAS BEEN DESIGNATED BY THE INNOVATIVE OPTICAL AND WIRELESS NETWORK GLOBAL FORUM, INC. (“IOWN GLOBAL FORUM”) AS AN APPROVED REFERENCE DOCUMENT AS SUCH TERM IS USED IN THE IOWN GLOBAL FORUM INTELLECTUAL PROPERTY RIGHTS POLICY (THIS “REFERENCE DOCUMENT”).

THIS REFERENCE DOCUMENT IS PROVIDED ”AS IS” WITH NO WARRANTIES WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT OF THIRD PARTY RIGHTS, TITLE, VALIDITY OF RIGHTS IN, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, REFERENCE DOCUMENT, SAMPLE, OR LAW. WITHOUT LIMITATION, IOWN GLOBAL FORUM DISCLAIMS ALL LIABILITY, INCLUDING WITHOUT LIMITATION LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS AND PRODUCTS LIABILITY, RELATING TO USE OF THE INFORMATION IN THIS REFERENCE DOCUMENT AND TO ANY USE OF THIS REFERENCE DOCUMENT IN CONNECTION WITH THE DEVELOPMENT OF ANY PRODUCT OR SERVICE, AND IOWN GLOBAL FORUM DISCLAIMS ALL LIABILITY FOR COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, PUNITIVE, EXEMPLARY, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE, ARISING IN ANY WAY OUT OF USE OR RELIANCE UPON THIS REFERENCE DOCUMENT OR ANY INFORMATION HEREIN.

EXCEPT AS EXPRESSLY SET FORTH IN THE PARAGRAPH DIRECTLY BELOW, NO LICENSE IS GRANTED HEREIN, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS OF THE IOWN GLOBAL FORUM, ANY IOWN GLOBAL FORUM MEMBER OR ANY AFFILIATE OF ANY IOWN GLOBAL FORUM MEMBER. EXCEPT AS EXPRESSLY SET FORTH IN THE PARAGRAPH DIRECTLY BELOW, ALL RIGHTS IN THIS REFERENCE DOCUMENT ARE RESERVED.

A limited, non-exclusive, non-transferable, non-assignable, non-sublicensable license is hereby granted by IOWN Global Forum to you to copy, reproduce, and use this Reference Document for internal use only. You must retain this page and all proprietary rights notices in all copies you make of this Reference Document under this license grant.

THIS DOCUMENT IS AN APPROVED REFERENCE DOCUMENT AND IS SUBJECT TO THE REFERENCE DOCUMENT LICENSING COMMITMENTS OF THE MEMBERS OF THE IOWN GLOBAL FORUM PURSUANT TO THE IOWN GLOBAL FORUM INTELLECTUAL PROPERTY RIGHTS POLICY. A COPY OF THE IOWN GLOBAL FORUM INTELLECTUAL PROPERTY RIGHTS POLICY CAN BE OBTAINED BY COMPLETING THE FORM AT: www.iowngf.org/join-forum. USE OF THIS REFERENCE DOCUMENT IS SUBJECT TO THE LIMITED INTERNAL-USE ONLY LICENSE GRANTED ABOVE. IF YOU WOULD LIKE TO REQUEST A COPYRIGHT LICENSE THAT IS DIFFERENT FROM THE ONE GRANTED ABOVE (SUCH AS, BUT NOT LIMITED TO, A LICENSE TO TRANSLATE THIS REFERENCE DOCUMENT INTO ANOTHER LANGUAGE), PLEASE CONTACT US BY COMPLETING THE FORM AT: https://iowngf.org/contact-us/

The IOWN GLOBAL FORUM mark and IOWN GLOBAL FORUM & Design logo are trademarks of Innovative Optical and Wireless Network Global Forum, Inc. in the United States and other countries. Unauthorized use is strictly prohibited. IOWN is a registered and unregistered trademark of Nippon Telegraph and Telephone Corporation in the United States, Japan, and other countries. Other names and brands appearing in this document may be claimed as the property of others.

Contents

List of Figures

List of Tables

Executive Summary

The IOWN Global Forum is attempting to advance the technologies that will give rise to a “smart world.” We consider a “smart world” to be one optimized to leverage the massive amounts of digital data generated by society’s physical environment, as well society themselves, in order to optimize societal infrastructure across multiple spheres, including mobility, safety and security, industrial applications, healthcare, municipal infrastructure, and more. The result will be a society able to live their lives safely and in their own way.

At the IOWN Global Forum, multiple industry leaders have joined together to research and develop network and computing infrastructure technologies to enable the next generation of ICT infrastructure. This new infrastructure will, in turn, support the transformation of lifestyles and work styles in the era of artificial intelligence (AI).

The IOWN Global Forum provided prominent future-looking use cases and identified application-specific service requirements benefiting users from different vertical industries in the use case reports at Cyber-Physical System Use Case Release-1 [IOWN GF CPS UC] and AI-Integrated Communications Use Case Release-1 [IOWN GF AIC UC].

A unique approach has been taken in the IOWN Global Forum to create the Reference Implementation Model (RIM) for implementing the future-looking use cases by leveraging Data-Centric Infrastructure (DCI) technologies [IOWN GF DCI] and Open All-Photonic Network (APN) technologies [IOWN GF Open APN], such as advancing the necessary technological development via DevOps methodology.

The purpose of this RIM work is to develop and evaluate the RIMs interactively. The RIMs provide guidance for the practical implementation of these technologies and their combinations for the overlay solution targeting specific CPS/AIC Use Cases. These are represented by the Benchmark Models the IOWN Global Forum has developed.

This document reports the overall system implementation best practice as a reference model to realize each future-looking use case based on the Benchmark Models for individual service scenarios with concreteness for each lifestyle and work style in the AI era. This document also aims to demonstrate many of the benefits of IOWN GF architecture and technology over today’s central-cloud-based implementations.

In this first edition, the Security use case (Guarding Services), categorized as an Area Management use case in Cyber-Physical System Use Case Release-1, was selected for this study to represent one of the most complex applications envisioned for systems using IOWN technologies. The Benchmark model that continuously analyzes the surveillance camera video image and LiDAR sensor data with AI to identify criminal activities or accidents for a prompt response and/or action is described in section 2. Initial RIM adapting the basic strategy of IOWN GF architecture and technology is described in section 5.

Based on this initial RIM recommendation for a Security use case (Guarding Services) in Area Management, we can expect the following benefits.

- RIM with Open APN technologies obtains benefits for high bandwidth and reduced power consumption across the core network and access network collecting massive data from surveillance cameras and LiDAR sensors. In addition, RIM with the Open APN technologies lowers latency by reducing optical-to-electrical conversion, as well as establishing a direct communication path in distributed clouds between the customer premise site and the telco edge/core site, which will help to build real-time monitoring services.

- RIM with DCI technologies obtains benefits for resource allocation flexibility from heterogeneous and disaggregated device resources pool that can allocate application’s functional node at the desired location to perform high data performance in hardware rate by Function Dedicated Network interface card such as RDMA capable Smart NIC, DPU [DPU], and IPU [IPU], in addition to CPU, GPU, and Persistent memory. This flexible resource allocation in disaggregated infrastructure helps CPU cost reduction and reduces power consumption.

1. Introduction

1.1. Purpose

The Innovative Optical and Wireless Network Global Forum (IOWN GF) is expected to accelerate the development and commercial availability of its architecture and technology in a relatively short period of time. To accomplish this goal for IOWN GF architecture and technology development, the IOWN GF develops and evaluates Reference Implementation Models (RIMs), which realize attractive IOWN GF use cases. The RIMs are also helpful for identifying potential technical issues and further improving the IOWN GF architecture and technology specifications.

The RIM utilizes IOWN GF architecture and technology developed by the IOWN GF to describe an end-to-end system that meets the requirements of a target use case and has the best metrics.

This document also aims to demonstrate many of the benefits of IOWN GF architecture and technology over today’s cloud-based implementations. The RIM adopts IOWN GF’s latest architecture and technology and continues to evolve to achieve low-cost and low-power consumption for realizing a sustainable society.

1.2. Approach

RIMs for the target use case is developed using the following three procedures.

1. Select a target use case from the various use cases of IOWN GF, such as area management security and entertainment interactive live music.
2. Develop Benchmark Model for the selected target use case, which is a defined way for evaluating implementation models by measuring selected metrics.
3. Develop a RIM for target use cases that yields the best evaluation results for the metrics defined in the benchmark model.

The RIM is expected to evolve repeatedly and achieve lower-cost and lower-power consumption by adopting new/revised IOWN GF architecture and technology.

Section 2 details the Benchmark Model for the target use case. Section 3 describes the flows and workloads of data processing in the Benchmark Model. Section 4 explains several major technology gaps and issues between today’s cloud-based implementations and use case requirements in the Benchmark Model for the target use case. Section 5 defines the initial RIM using IOWN GF architecture and technology. Section 6 concludes our first achievements and describes future studies.

1.3. Scope

This version of the document covers an initial study on the first target use case. The Area Management Security Use Case (AM Security UC) included in Cyber-Physical System Use Case Release 1 [IOWN GF CPS UC] was chosen as the first target use case. The RIM for the AM security UC in this document has been developed with reference to Open All-Photonic Network Functional Architecture [IOWN GF Open APN] and Data-Centric Infrastructure Functional Architecture [IOWN GF DCI], and Data Hub Functional Architecture [IOWN Data Hub].

IOWN GF continuously revises the RIM through Proof of Concept (PoC) and detailed specification development process, though this activity is out of the scope for the initial RIM study in this document.

2. Benchmark Model

This section describes the Benchmark Model for the Area Management Security Use Case (AM Security UC) in the Cyber-Physical System Use Case document [IOWN GF CPS UC], which defines Reference Case in 2.1 and Metrics and Evaluation Methods in 2.2.

The AM Security UC is a representative use case among CPS Use Cases. The AM Security UC contains the basic requirements for the various use cases proposed in the CPS Use Cases. Through the development of a Benchmark Model for this use case, we also aim to identify common methodologies that can be applied to the Benchmark Models for other use cases in the future (See Annex A).

2.1. The Reference Case for the AM Security UC

This subsection develops a Reference Case for the AM Security UC (Guarding Services). The Reference Case digs deeper into the target use case and specifically defines the conditions for determining functional and non-functional requirements. The aim to define the Reference Case for the target use case is to make it accurate to evaluate implementation models by measuring selected metrics in the specific conditions.

2.1.1. Description

There are myriad use cases in the security field. Among them, this document will describe a security use case that uses AI to continuously analyze surveillance camera video stream data and Light Detection And Ranging (LiDAR) sensor data to detect criminal acts or accidents and take prompt action.

In the AM Security UC, there are several situations in which security needs to be tightened, such as the Capitol when Congress is open, airports or facilities especially when VIP arrives there, and venues where the summit meeting is held, and the roadside of a marathon event. Security surveillance devices will be installed in such locations to provide automatic surveillance for peripheral security.

2.1.2. Area Size

We can assume several options for the area size of each service to keep flexibility on technology development. Depending on the service provider's choice, a service can cover a Large Area, e.g., sub-national level area, or a narrower area, e.g., a single building, which is a base unit for areas to be monitored (monitored areas).

In this AM Security UC in the document, the Area size is defined as the Large Area. Large Area sizes can contain between 100 - 10,000 monitored areas.

In this AM Security UC, these considerations and metrics are based on a secure environment based in Japan. As such, it assumes about 1,250 monitored areas (monitored areas per 100,000 people, about 650 monitored areas in eastern Japan, about 600 monitored areas in western Japan).

2.1.3. Devices Types and Number

The AM Security UC requires consideration of the various input devices for collecting information from the real world and the output devices to present information from the Guarding Services system.

The AM Security UC in this document assumes the following devices:

- Surveillance Video Cameras with Motion sensors and microphones and LiDAR sensors are connected to wired networks as input devices. Each monitored area, such as smart building or city block, has 25 sub-blocks. Each sub-block, which is about 50-meters in radius, has 40 monitoring posts with a video camera and a LiDAR sensor. As a result, each monitored area has 1,000 video cameras and 1,000 LiDAR sensors. In addition, there are 100 to 10,000 monitored areas in a Large Area. Therefore, the number of video cameras and LiDAR in Large Areas is 100,000 to 10,000,000.

- As output devices, the Local Police System and/or Smartphones of police officers and/or guards in charge of the area are connected to wired/wireless networks. The number of devices that need to be notified in each monitored area is 1,000.

2.1.4. Input Data to the Guarding Services System

The following items show the types of input data to this system and their data volumes.

- Video stream data
    - Surveillance Video Camera with Motion Sensor and microphone, that captures the Full HD movie at 15 fps and sound to monitor the area and transmit video stream data to the data center.

      Note: If Motion Jpeg is used, the stream data from each camera will have a flow rate of around 45-60Mbps. If H.264 is used, it will be 3Mbps.
- LiDAR point cloud stream data
    - LiDAR Sensor that measures the distance to the surface of surrounding objects as well as the brightness of its surface over a 100m range, and produces 100,000 points of data, each 2-4 bytes long, at 20Hz frequency, i.e., 2 million points data per second, resulting in 32-64 Mbps data stream (without compression).

2.1.5. Output Data from the Guarding Services System

The following items show the types of output data to this system and their data volumes.

- Voice message explaining the situation automatically generated by AI to the police officers and/or security guards' wearable earphones.

  Note1: When a suspicious person or action is detected, the place, situation, characteristics of the person, etc., are explained by voice.

  Note2: The message will be copied to up to 1,000 nearby police officers/security guards.
- Instant alert messages, such as SMS messages, are sent to 1) the local police systems and 2) police officers and/or security guards' smartphones.

  Note: The frequency of alerts will vary based on the threshold setting of the algorithm used. However, it will not be so high, most probably far less than once per second.
- Copied video stream data are transferred to the external security monitoring system to monitor the monitored area continuously.
- Replayed video stream data transferred to the external police system will be used for the post-event verification upon the request from the police officer in charge.

2.1.6. Functional Requirements

2.1.6.1. Main Data Processing Flow and Relevant Functions

We continuously receive the sensor data, analyze the situation to detect dangerous situations, and guide security to react quickly. The following are functional requirements on the main data processing flow and relevant functions.

- Determining the data center to receive the data
  If multiple data centers can be selected, determine the appropriate data center by considering the distance from the sensing device, network congestion, and the load status inside the data center.

  Note1: Data from multiple sensors in the same location needs to be collected in the same data center because these need to be associated together when being analyzed.

  Note2: Data would be aggregated at a local aggregation node for the monitored area and sent to the data center.

- Duplication of the stream data as necessary
  The received stream data may be duplicated so that these can be used for two or more purposes, one for long-term storage and the other for analysis.

- Data storage efficiency
  For the post-event analysis, the received data must be stored for long periods, e.g., one month. Due to storage efficiency, it may be converted to a data format with higher compression efficiency regardless of whether it is reversible or irreversible.

- Data record association
  If multiple cameras and LiDAR sensor data need to be linked and jointly analyzed, multiple separate data records must be linked together on a time and space basis.

  Note: It may be required to associate multiple records with slightly different measurement times due to the difference in frame timing and tolerate the delayed arrival of some data.

- Continuous analysis
  Each input video stream data and LiDAR point cloud stream data will be continuously analyzed by the assigned resource(s) to detect dangerous situations and immediately take required action(s).
  The below processing steps are included in the continuous analysis

  - Object recognition
    Run the CNN-type of object recognition algorithms against each frame included in the video stream data and LiDAR point cloud stream data.

    - Recognize people, cars, bikes, relevant objects - e.g., guns, knives, etc.

    - If the same location is monitored by multiple sensors, e.g., video camera and LiDAR sensor, link the object recognition results from different sensors.

  - Generation of the Live 4D Map data that represents the behavior of the identified object
    To understand the situation correctly, analysis of data at a specific point in time is insufficient, and analysis of the behavior of objects in chronological order is necessary. As the detected objects (e.g., people, cars, bikes) move, their data must be maintained as the Live 4D Maps for subsequent analysis.

  - Spatial time-series analysis
    An analytical model for understanding the behavior of detected objects that change over time is run

against the Live 4D map to understand each detected object's behavior and overall situation. If the result meets a specific condition, the alerting process is triggered by the system. There would be multiple conditions to be monitored simultaneously.

- Alerting
  When the situation of concern occurs, the system will promptly alert the police organization responsible for the target person and the police officers near the detected place.

2.1.6.2. Supplementary Functions

The use cases need to consider the interfaces / APIs that the external system can connect to provide the data requested by the external system. The following are functional requirements for supplementary functions:

- Registration/update/deletion of the object recognition models used for the continuous analysis
- Registration/update/deletion of the behavioral models used for the continuous analysis
- Registration/update/deletion of subscriber systems for the target
- Requests submitted by an external system, e.g., a local police system, to transfer a copy of video stream data and/or LiDAR point cloud stream data to the external requesting system

2.1.7. Non-Functional Requirements

The IOWN GF mission includes meeting the requirements of use cases and developing more scalable and more energy-efficient end-to-end systems.

Three non-functional requirements are essential in this use case:

- Response time
  Immediate action and interaction with relevant devices and people are required for this use case. If suspicious behavior is detected, alerts should be sent to the security officers and/or relevant devices immediately, e.g., less than 1 sec, ideally 100 milliseconds.
  - For machine-to-machine automation process: 100 milliseconds
  - For notification delivery to the security officials in charge: 1 second (mandatory) / 100 milliseconds (objective)
- Scalability
  This use case requires the ability to support up to 10,000,000 surveillance video cameras and/or LiDAR sensors.
- Energy efficiency
  There are limits to when and where near real-time monitoring is needed. To eliminate wasted energy consumption, you need to turn the real-time monitoring process on and off as needed. For example, the Capitol and its surroundings need to be much safer than usual during the Congressional session. To increase the security level for the duration that Congress is in attendance, the resources for AI analysis resources could dynamically increase to provide more powerful real-time monitoring.

2.2. The Metrics and Evaluation Method for the AM Security UC

This subsection develops a Metrics and Evaluation Method for the AM Security UC (Guarding Services).

2.2.1. Overview

A Guarding Services system that implements an AM Security UC monitors the status of the monitored area in real-time and coordinates automated actions by the system to mitigate security risks. So, the system must collect data from sensors installed in various monitoring areas, analyze the collected data through a set of intelligence applications, and take the required actions.

In addition, this system is expected to satisfy the functional and non-functional requirements of use cases and to be provided as a system with as low cost and low power consumption as possible.

A certain degree of data and process consolidation in one or several data centers is required to build an efficient system at a low cost and low power consumption. However, it inevitably requires an increase in response time due to the data transfer over the network and the data distribution process in the data center.

2.2.2. Metrics

The system needs to realize this AM security UC to meet the functional and non-functional requirements, especially the end-to-end response time. And the system should be the reference implementation model that has lower cost and lower power consumption not only during peak hours but also during idle time.

The metrics for AM Security UC are:

- System cost, and
- System Power consumption.

2.2.3. Evaluation Method

Evaluation of implementation models that realize AM security UC can be achieved by comparing Metrics. The power consumption can be obtained by measuring a network device that transmits data through Proof of Concept (PoC) or the like and a computing node that analyzes the data. In addition, the system cost can be obtained by calculating the cost of the implemented system.

This document discusses the expected qualitative benefits, and the quantitative evaluation method will be defined through PoCs starting in 2022. This document will be updated based on the results of the evaluation work. See A.3 in Annex A for more information on developing metrics and evaluating methods for the target use case.

3. Dataflow and Workloads Analysis

This section first analyzes the flows and workloads of data processing in the benchmark model. This section aims to identify a broader range of requirements necessary for system design in addition to the key requirements shown in the use case document [IOWN GF CPS UC]. Then this section subdivides data processes into sub-processes, defines the behavior of each sub-process. Finally, this document clarifies the connections and dataflows between sub-processes until they have sufficiently fine granularity for evaluation by the IOWN GF in the context of the technologies it is studying.

Through this analysis, we utilize the Dataflow and Workload Profiling framework, which the IOWN GF developed to identify service gaps/requirements of use cases accurately and efficiently. Please refer to Annex B for the details of the framework.

3.1. Data Pipeline Diagram

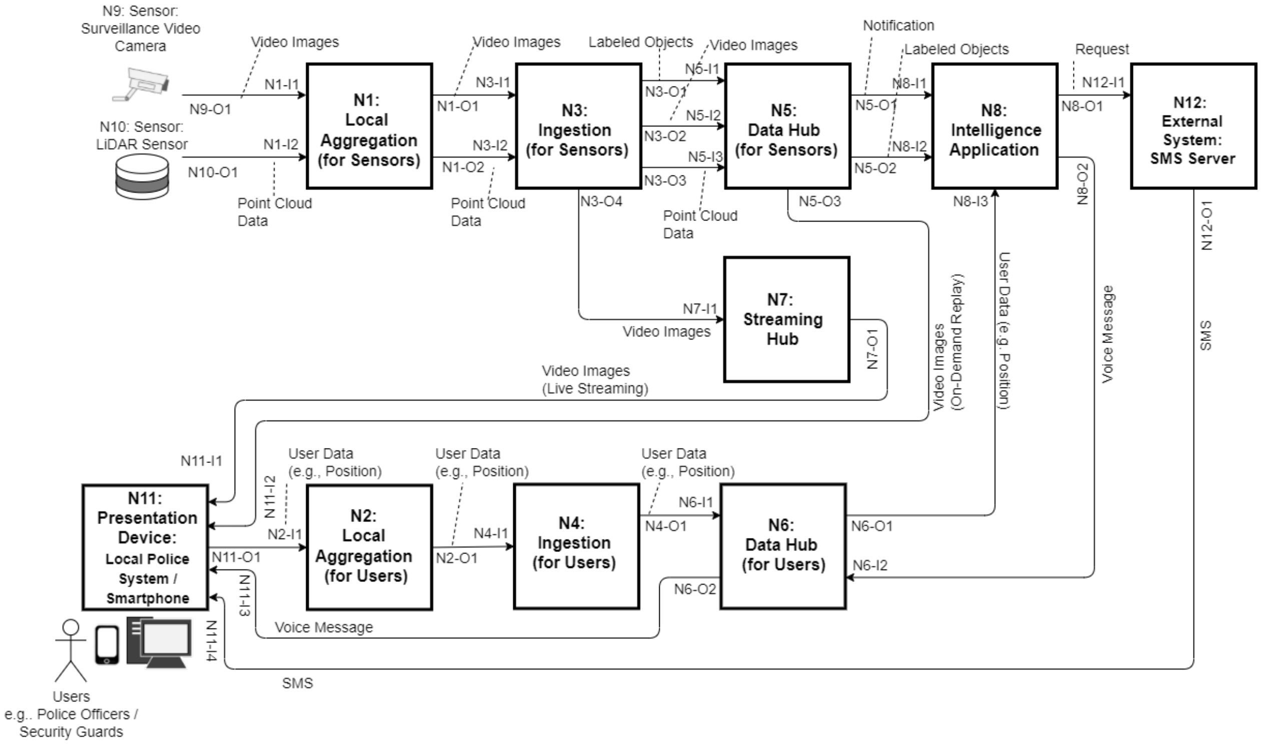

N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
Video Images
N1-I1
N9-O1
N1-I2
N10-O1
Point Cloud Data
N1: Local Aggregation (for Sensors)
N1-O1
N1-O2
Video Images
N3-I1
N3-I2
Point Cloud Data
N3: Ingestion (for Sensors)
N3-O1
N3-O2
N3-O3
N3-O4
Labeled Objects
Video Images
N5-I1
N5-I2
N5-I3
Point Cloud Data
N5: Data Hub (for Sensors)
N5-O1
N5-O2
N5-O3
Notification
Labeled Objects
N8-I1
N8-I2
N8-I3
N8: Intelligence Application
N8-O1
N8-O2
Request
N12-I1
N12: External System: SMS Server
N12-O1
SMS
N7-I1
Video Images
N7: Streaming Hub
N7-O1
Video Images (Live Streaming)
Video Images (On-Demand Replay)
User Data (e.g. Position)
Voice Message
N11-I1
N11-I2
N11: Presentation Device: Local Police System / Smartphone
N11-O1
N11-I3
N11-I4
User Data (e.g., Position)
N2-I1
N2: Local Aggregation (for Users)
N2-O1
User Data (e.g., Position)
N4-I1
N4: Ingestion (for Users)
N4-O1
User Data (e.g., Position)
N6-I1
N6: Data Hub (for Users)
N6-O1
N6-I2
N6-O2
Voice Message
SMS
Users e.g., Police Officers / Security Guards

Figure 3.1-1: A Data Pipeline Diagram for AM Security UC

Figure 3.1-1 is a DPD (Data Pipeline Diagram) used for the data processing and dataflow analysis on the Area Management Security Use Case (AM Security UC). As shown in Figure 3.1-1, the DPD has the following eight different types of functional nodes:

- Local Aggregation nodes (N1, N2): Nodes that collect data from devices and send data to the Ingestion node. They may also provide functions for efficient data collection, such as consolidating and/or reducing data from multiple devices. N1 collects sensor data from Sensor nodes such as surveillance video cameras (N9) and LiDAR sensors (N10), while N2 collects user data from Presentation Device nodes (N11).
- Ingestion nodes (N3, N4): Nodes that accept data from Local Aggregation nodes and may update the database/storage in the Data Hub node and/or Streaming Hub node. Ingestion nodes may also provide

efficient data collection and usage functions, such as data format conversion, indexing, and cognitive functions, such as image recognition and metadata creation. N3 treats sensor data, while N4 treats user data.

- Data Hub nodes (N5, N6): Nodes with a database/storage that collectively preserve data and provide these data for subsequent, possibly repeated data usages. To handle massive data usage, a Data Hub may store data replications in a distributed manner and streamline access/query by managing and utilizing data indices. In addition, the Data Hub node supports distributing notifications to listening nodes, including Intelligence Application nodes (N8). N5 is a Data Hub node for sensor data, while N6 is a Data Hub node for user data.
- Streaming Hub node (N7): A node that receives one or multiple video image streams and relays them to Presentation Device nodes (N11).
- Intelligence Application node (N8): A node that provides application services such as analysis/optimization, alerts, and data exposure to External System nodes (N12) by utilizing data received from the Data Hub nodes (N5, N6).
- Sensor node (N9, N10): Nodes that output sensed data. N9 represents surveillance video cameras which provide video image streams, and N10 represents LiDAR sensors which provide point cloud data.
- Presentation Device node (N11): A node that receives video image streams and alert messages and renders them for presentation. It also supports some functionality for uploading user data. Typically, it is the output function of a user’s communication device, such as a smartphone or a PC for a local police system.
- External System node (N12): A 3rd party’s system that consumes output data of Intelligence Application nodes (N8). In Figure 3.1-1, an SMS server is assumed as an External System node.

Figure 3.1-1 just shows the top-level view of the DPD, and the more detailed descriptions of the functional nodes are shown in Annex C.

To achieve the functional requirements described in the Benchmark Model, the DPD contains four major processing flows:

- Detection of suspicious persons or activities,
- Collection of user data,
- Alerting (SMS and Voice Message), and
- Live monitoring and on-demand replay.

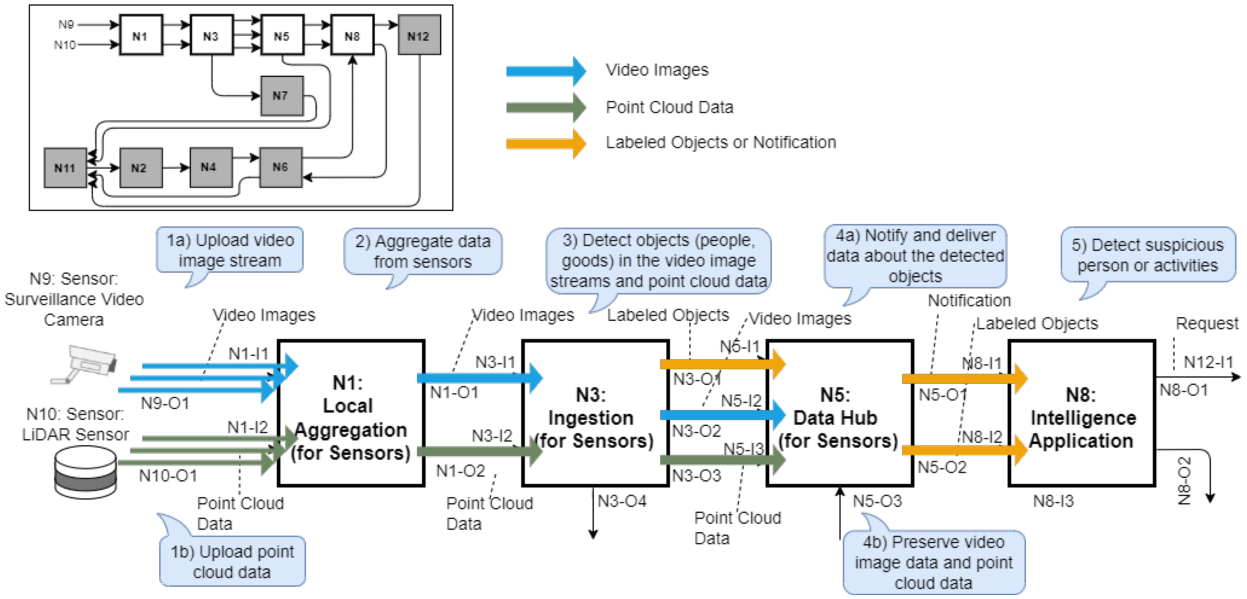

N9
N10
N1
N3
N5
N8
N12
N7
N11
N2
N4
N6
Video Images
Point Cloud Data
Labeled Objects or Notification
1a) Upload video image stream
2) Aggregate data from sensors
3) Detect objects (people, goods) in the video image streams and point cloud data
4a) Notify and deliver data about the detected objects
5) Detect suspicious person or activities
N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
Video Images
N1-I1
N9-O1
N1-I2
N10-O1
Point Cloud Data
1b) Upload point cloud data
N1: Local Aggregation (for Sensors)
Video Images
N1-O1
N3-I1
N3-I2
N1-O2
Point Cloud Data
N3: Ingestion (for Sensors)
Labeled Objects
N3-O1
N3-O2
N3-O3
N3-O4
Video Images
N5-I1
N5-I2
N5-I3
Point Cloud Data
N5: Data Hub (for Sensors)
N5-O3
4b) Preserve video image data and point cloud data
Notification
Labeled Objects
N5-O1
N5-O2
N8-I1
N8-I2
N8-I3
N8: Intelligence Application
Request
N12-I1
N8-O1
N8-O2

Figure 3.1-2: Processing Flow 1: Detection of Suspicious Persons or Activities

Figure 3.1-2 shows the processing flow of detection of suspicious persons or activities. This processing flow is for mining values from massive sensor data in real-time. The surveillance video cameras (N9) and the LiDAR sensors (N10) upload video image data and point cloud data, respectively. The Local Aggregation node (N1) aggregates data received from the sensors, then forwards them to the Ingestion node (N3). The Ingestion node (N3) performs object detection processes by utilizing the received video image and point cloud data. Results (or labeled objects) may be used by many applications. The Ingestion node (N3) also converts the received video image data and point cloud data to a format suitable for preservation at the Data Hub node (N5). The Ingestion node (N3) posts the output data (i.e., labeled objects, video imaged data, and point cloud data) to the Data Hub (N5) for further usage. The Data Hub node (N5) preserves the received data, then sends notifications to the listening nodes. In this case, the Intelligence Application node (N8) is supposed to subscribe to labeled objects, so notifications are sent from the Data Hub node (N5) to the Intelligence Application node (N8) to trigger further analysis there. The Intelligence Application node (N8) interacts with the Data Hub node (N5) and continuously maintains the Live 4D Map according to the labeled objects detected by Ingestion nodes (N3). The Intelligence Application node (N8) acquires necessary spatial time-series data from the Live 4D Map. It analyzes the behaviors of the objects on the Live 4D Map so that suspicious persons or activities can be detected.

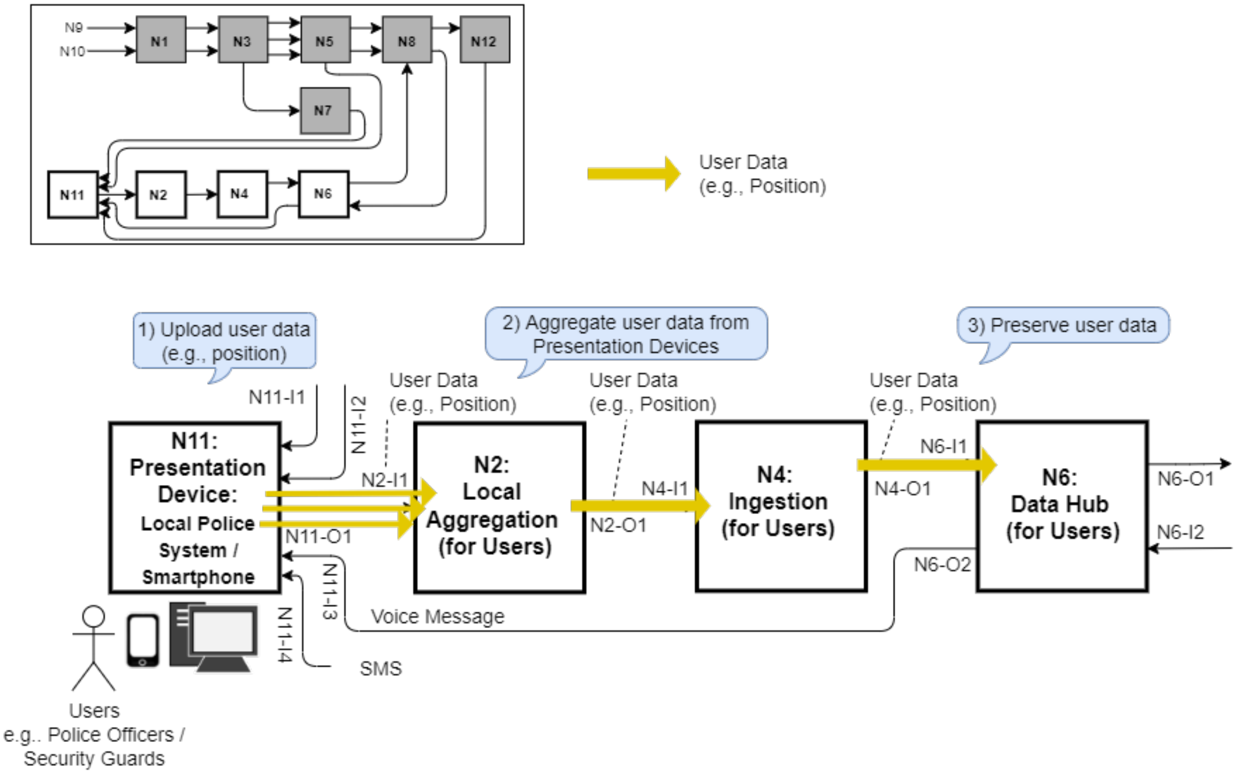

N9
N10
N1
N3
N5
N8
N12
N7
N11
N2
N4
N6
User Data
(e.g., Position)
1) Upload user data (e.g., position)
2) Aggregate user data from Presentation Devices
3) Preserve user data
N11-I1
N11-I2
User Data (e.g., Position)
User Data (e.g., Position)
User Data (e.g., Position)
N11: Presentation Device: Local Police System / Smartphone
N2-I1
N2: Local Aggregation (for Users)
N2-O1
N4-I1
N4: Ingestion (for Users)
N4-O1
N6-I1
N6: Data Hub (for Users)
N6-O1
N6-I2
N6-O2
N11-O1
N11-I3
N11-I4
Voice Message
SMS
Users
e.g.. Police Officers / Security Guards

Figure 3.1-3: Processing Flow 2: Collection of User Data

Figure 3.1-3 shows the processing flow of collecting user data (e.g., position). The collected user data is used at the Intelligence Application node (N8) for identifying target recipients of voice messages and instant alert messages, as described in the next paragraph. User data are collected from the Presentation Device nodes (N11), securely uploaded to the Data Hub node (N6) via the Local Aggregation node (N2) and the Ingestion node (N4). The Data Hub node preserves the received user data for further usage at authorized applications.

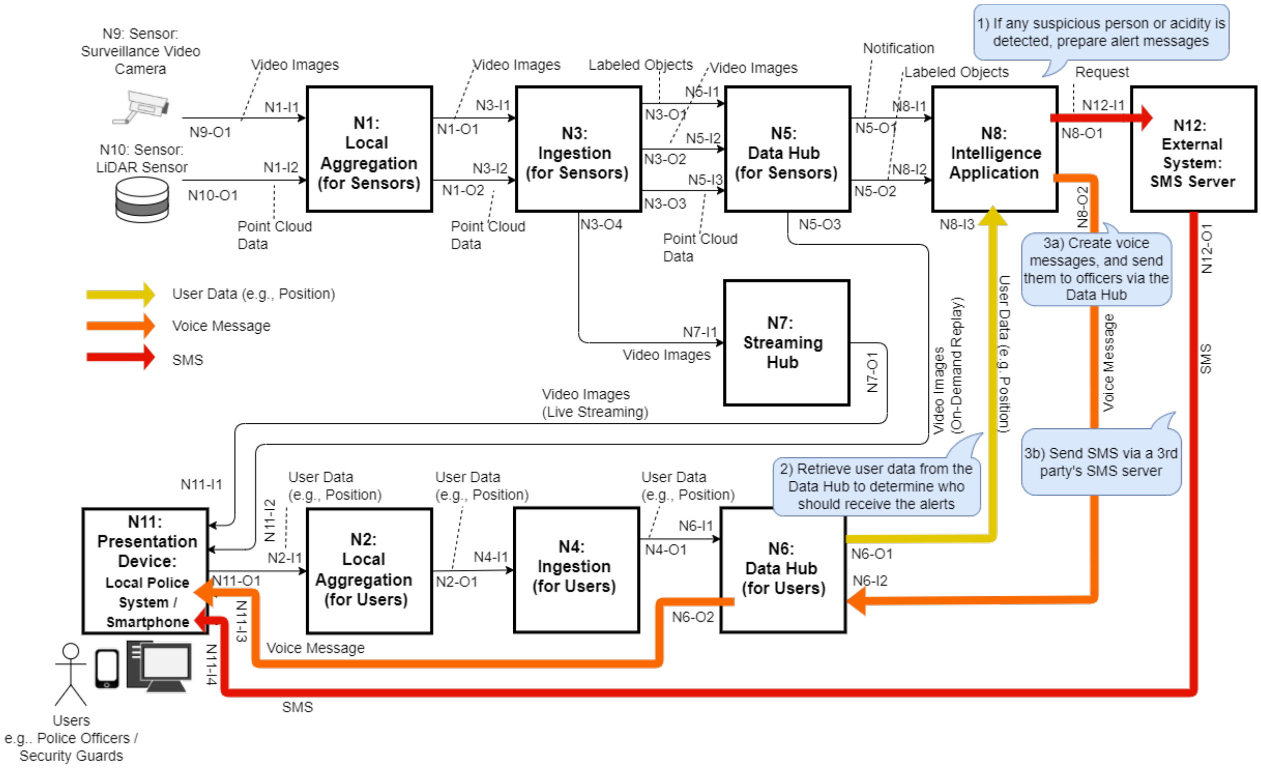

N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
N9-O1
N10-O1
Video Images
N1-I1
N1-I2
Point Cloud Data
N1: Local Aggregation (for Sensors)
N1-O1
N1-O2
Video Images
N3-I1
N3-I2
Point Cloud Data
N3: Ingestion (for Sensors)
Labeled Objects
N3-O1
N3-O2
N3-O3
N3-O4
Video Images
N5-I1
N5-I2
N5-I3
Point Cloud Data
N5: Data Hub (for Sensors)
N5-O1
N5-O2
N5-O3
Notification
Labeled Objects
N8-I1
N8-I2
N8-I3
N8: Intelligence Application
N8-O1
N8-O2
1) If any suspicious person or acidity is detected, prepare alert messages
Request
N12-I1
N12: External System: SMS Server
N12-O1
SMS
3a) Create voice messages, and send them to officers via the Data Hub
Voice Message
3b) Send SMS via a 3rd party's SMS server
User Data (e.g., Position)
Voice Message
SMS
N7-I1
Video Images
N7: Streaming Hub
N7-O1
Video Images (Live Streaming)
Video Images (On-Demand Replay)
User Data (e.g. Position)
2) Retrieve user data from the Data Hub to determine who should receive the alerts
N11-I1
N11-I2
N11: Presentation Device: Local Police System / Smartphone
N11-O1
N11-I3
N11-I4
User Data (e.g., Position)
N2-I1
N2: Local Aggregation (for Users)
N2-O1
User Data (e.g., Position)
N4-I1
N4: Ingestion (for Users)
N4-O1
User Data (e.g., Position)
N6-I1
N6: Data Hub (for Users)
N6-O1
N6-I2
N6-O2
Voice Message
SMS
Users e.g.. Police Officers / Security Guards

Figure 3.1-4: Processing Flow 3: Alerting (SMS and Voice Message)

Figure 3.1-4 shows the processing flow of the alerting. This processing flow corresponds to the delivery of voice messages and instant alert messages to the target recipients as described in 2.1.5 of the Benchmark Model. Once the Intelligence Application node (N8) detects suspicious persons and activities in the corresponding monitored area, it retrieves user data from the Data Hub node (N6) to determine which users should be notified. In this case, the Intelligence Application node (N8) has two means for notification. One is using voice messages generated by the Intelligence Application node (N8) itself, where the voice messages are delivered to the Presentation Device nodes (N11) of the selected users via the Data Hub node (N6). The other is short messaging using SMS servers (N12) operated by 3rd parties.

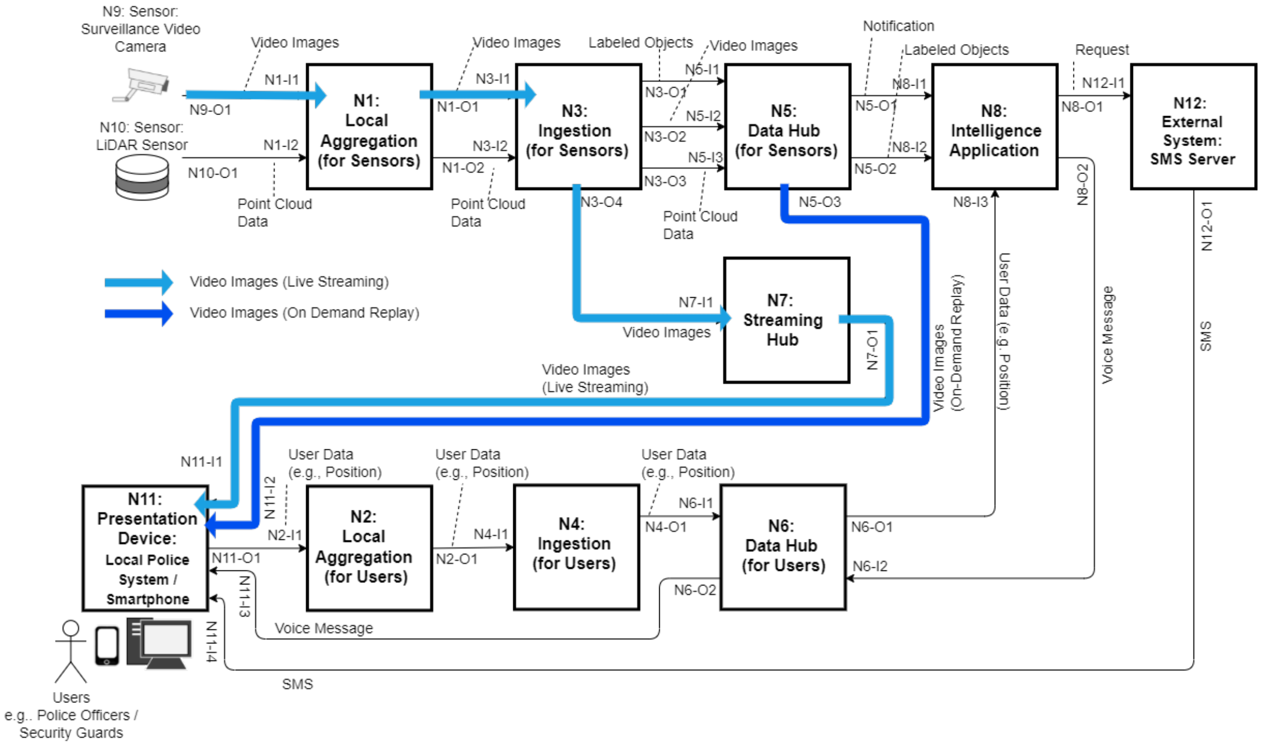

N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
N9-O1
N10-O1
Video Images
N1-I1
N1-I2
Point Cloud Data
N1: Local Aggregation (for Sensors)
N1-O1
N1-O2
Video Images
N3-I1
N3-I2
Point Cloud Data
N3: Ingestion (for Sensors)
N3-O1
N3-O2
N3-O3
N3-O4
Labeled Objects
Video Images
N5-I1
N5-I2
N5-I3
Point Cloud Data
N5: Data Hub (for Sensors)
N5-O1
N5-O2
N5-O3
Notification
Labeled Objects
N8-I1
N8-I2
N8-I3
N8: Intelligence Application
N8-O1
N8-O2
Request
N12-I1
N12: External System: SMS Server
N12-O1
SMS
Video Images (Live Streaming)
Video Images (On Demand Replay)
N7-I1
Video Images
N7: Streaming Hub
N7-O1
Video Images (Live Streaming)
Video Images (On-Demand Replay)
User Data (e.g. Position)
Voice Message
N11-I1
N11-I2
N11: Presentation Device: Local Police System / Smartphone
N11-O1
N11-I3
N11-I4
User Data (e.g., Position)
N2-I1
N2: Local Aggregation (for Users)
N2-O1
User Data (e.g., Position)
N4-I1
N4: Ingestion (for Users)
N4-O1
User Data (e.g., Position)
N6-I1
N6: Data Hub (for Users)
N6-O1
N6-I2
N6-O2
Voice Message
SMS
Users e.g., Police Officers / Security Guards

Figure 3.1-5: Processing Flow 4: Live Monitoring and On-Demand Replay

Figure 3.1-5 shows the processing flow of the live monitoring and on-demand replay. Users can watch live video images of selected surveillance video cameras (N9) via the Local Aggregation node (N1), the Ingestion node (N3), and the Streaming Hub node (N7). In the case of multiple users watching the video images of the same cameras, the Streaming Hub node (N7) duplicates the video image data flow and distributes them to the users. Meanwhile, users can also watch video images of any cameras on demand. In the on-demand replay case, the Data Hub node (N5) works as a video server to deliver the video image data in response to requests from the users.

3.2. Dataflow Profiles

This subsection shows dataflow profiles of each data type. The DPD showed that it is necessary to handle different types of data (i.e., video image data, point cloud data, labeled object data and notifications, user data, and alerts) for supporting the given benchmark model. Each data type has distinct dataflow requirements (e.g., number of sources, data rate, data size, occurrence rate, etc.).

Note that an appropriate communication scheme and compression scheme should be selected for each communication section according to the dataflow requirements. This selection will be discussed in section 5. Examples of communication schemes and compression schemes are as follows:

- Communication scheme: Shared memory, DMA, RDMA, UDS (Unix Domain Socket), (S)RTP over UDP or TCP, etc.
- Compression scheme:
    - For video images: H.264, Motion JPEG, RAW, etc.
    - For point cloud data: G-PCC, V-PCC, RAW, etc.

3.2.1. Symbols

The following symbols are defined and used in the dataflow profiles.

- *#_of_monitored_areas*
    - ○ *#_of_monitored_areas* means the number of monitored areas that one instance of the functional node accommodates. The concrete values of *#_of_monitored_areas* in a specific deployment scenario will be discussed in Annex E.
- *chunk_interval*
    - ○ *chunk_interval* means the interval time of chunked streaming data such as video image data and point cloud data. The "chunk" process is needed when the Ingestion node (N3) posts these data to the Data Hub node (N5) to let data consumers access these data as a series of objects. Optimal *chunk_interval* highly depends on the implementation and may affect delay time to get available to data consumers, compression rate, download efficiency, etc.

3.2.2. Video Image Data

The following table summarizes the characteristics and significant attributes of the dataflows related to video image data.

Table 3.2-1: Dataflow Profiles of Video Image Data

NODE / DATAFLOW ID	DESCRIPTION	ATTRIBUTES
• **Sensor (N9) → Local Aggregation (N1)** ○ **N9-O1, N1-I1** • **Local Aggregation (N1) → Ingestion (N3)** ○ **N1-O1, N3-I1** • **Ingestion (N3) → Streaming Hub (N7)** ○ **N3-O4, N7-I1** • **Streaming Hub (N7) → Presentation Device (N11)** ○ **N7-O1, N11-I1**	• Full HD video image streams at 15 fps from surveillance video cameras ○ Meta-data may be embedded.	• # of sources: ○ N9-O1, N11-I1: 1 ○ N1-I1, N1-O1, N3-I1: 1,000 cameras x *#_of_monitored_areas* ○ N3-O4, N7-I1: up to 20 cameras x *#_of_monitored_areas* ○ N7-O1: up to 20 cameras x 20 users x *#_of_monitored_area* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG) ○ 750 Mbps / source (RAW)

• **Ingestion (N3) → Data Hub (N5)** ○ **N3-O2, N5-I2** • **Data Hub (N5) → Presentation Device (N11)** ○ **N5-O3, N11-I2**	• A series of chunked data of compressed Full HD video Images at 15 fps ○ Meta-data may be embedded. ○ Chunk is necessary for efficient data access at the Data Hub.	• # of sources: ○ N3-O2, N5-I2: 1,000 cameras x *#_of_monitored_areas* ○ N5-O3: 50 cameras x *#_of_monitored_areas* ○ N11-I2: 1 • Data size: ○ (0.4 x *chunk_interval*) MB / chunk (H.264), see Note ○ (5.6~7.5 x *chunk_interval*) MB / chunk (Motion JPEG) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source

Note: In H.264, the minimum *chunk_interval* depends on the interval time of keyframes since chunked data must start with a keyframe. The shorter interval time of keyframes can lower the *chunk_interval.* This can reduce the latency to write to the Data Hub node while decreasing the compression rate.

3.2.3. Point Cloud Data

The following table summarizes the characteristics and significant attributes of the dataflows related to point cloud data.

Table 3.2-2: Dataflow Profiles of Point Cloud Data

NODE / DATAFLOW ID	DESCRIPTION	ATTRIBUTES
• **Sensor (N10) → Local Aggregation (N1)** ○ **N10-O1, N1-I2**	• Sensor data from LiDAR sensors in a vendor-specific format ○ Each LiDAR sensor captures 100,000 points of data, each 2~4 bytes long, at 20Hz frequency.	• # of sources: ○ N10-O1: 1 LiDAR sensor ○ N1-I2: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data rate: 32~64 Mbps / source (not compressed)
• **Local Aggregation (N1) → Ingestion (N3)** ○ **N1-O2, N3-I2**	• Point cloud data with 100,000 points at 20 fps in a Cartesian coordinate system ○ Meta-data may be embedded.	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data rate ○ 2 Mbps / source (V-PCC) ○ 7 Mbps / source (G-PCC) ○ 256 Mbps / source (RAW)
• **Ingestion (N3) → Data Hub (N5)** ○ **N3-O3, N5-I3**	• A series of chunked compressed point cloud data with 100,000 points at 20 fps in a Cartesian coordinate system ○ Meta-data may be embedded. ○ Chunk is necessary for efficient data access at the Data Hub node.	• # of sources: 1,000 LiDAR sensors x *#_of monitored_areas* • Data size ○ (0.26 x *chunk_interval*) MB / chunk (V-PCC), see Note ○ (0.92 x *chunk_interval*) MB / chunk (G-PCC) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source

Note: The minimum *chunk_interval* depends on the interval time of keyframes since chunked data must start with a keyframe.

3.2.4. Labeled Objects and Notification

The following table summarizes the characteristics and significant attributes of the dataflows related to labeled objects and notifications:

Table 3.2-3: Dataflow Profiles of Labeled Objects and Notification

NODE / DATAFLOW ID	DESCRIPTION	ATTRIBUTES
• **Ingestion (N3) → Data Hub (N5)** ○ **N3-O1, N5-I1** • **Data Hub (N5) → Intelligence Application (N8)** ○ **N5-O2, N8-I2**	• Labeled object data detected at a sensing point. A sensing point may be captured by multiple devices (cameras and LiDAR sensors). The Ingestion node fuses the captured data from multiple devices and produces consistent labeled data for the sensing point. • The data has a format suitable for data-sharing. It may contain its label name, feature values, and the corresponding JPEG image and point cloud data cropped from the original video image and point cloud data frame.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 3 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source
• **Data Hub (N5) → Intelligence Application (N8)** ○ **N5-O1, N8-I1**	• Notification messages • Multiple notification messages from different sensing points can be merged.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 256 Byte / message • Occurrence rate: up to 15 messages / second / source

3.2.5. User Data

The following table summarizes the characteristics and significant attributes of the dataflows related to user data.

Table 3.2-4: Dataflow Profiles of User Data

NODE / DATAFLOW ID	DESCRIPTION	ATTRIBUTES
• **Presentation Device (N11) → Local Aggregation (N2)** ○ **N11-O1, N2-I1** • **Local Aggregation (N2) → Ingestion (N4)** ○ **N2-O1, N4-I1** • **Ingestion (N4) → Data Hub (N6)** ○ **N4-O1, N6-I1** • **Data Hub (N6) → Intelligence Application (N8)** ○ **N6-O1, N8-I3**	• User data (e.g., position data) posted by applications on Presentation Devices (N11) ○ Assuming that each source posts user data every minute.	• # of sources: ○ N11-O1: 1 ○ Others: 1,000 users x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: ○ N11-O1, N2-I1, N2-O1, N4-I1, N4-O1, N6-I1: 1 message / minute / source ○ N6-O1, N8-I3: Ad hoc ▪ Assuming user data is retrieved as needed when the incident is detected.

3.2.6. Alerts (Voice Message and SMS)

The following table summarizes the characteristics and significant attributes of the dataflows related to alerts, i.e., voice messages and SMS.

Table 3.2-5: Dataflow Profiles of Alerts

NODE / DATAFLOW ID	DESCRIPTION	ATTRIBUTES
• **Intelligence Application (N8) → Data Hub (N6)** ○ **N8-O2, N6-I2** • **Data Hub (N6) → Presentation Device (N11)** ○ **N6-O2, N11-I3**	• Voice messages and the position of the corresponding sensing points ○ A user listens to voice messages 10% of the time during the day on average	• # of sources: ○ N8-O2, N6-I2: *#_of_monitored_areas* ○ N6-O2: 100 users x *#_of_monitored_areas* ○ N11-I3: 1 • Data rate: ○ 64Kbps x 10% / source
• **Intelligence Application (N8) → External System (N12)** ○ **N8-O1, N12-I1**	• Request to the SMS server ○ Texts of the short messages and their recipients' telephone number ○ Assuming 100 users will receive the short message.	• # of sources: *#_of_monitored_areas* • Data size: 1.5KB / message • Data rate: Ad hoc ○ Assuming 100 messages / source are delivered as needed when the incident is detected.

4. Technology Gaps and Issues

This section describes technical gaps and issues between today's centralized cloud-based implementation and use case requirements in the given benchmark model. Technical solutions need to be developed from a full-stack and end-to-end viewpoint to resolve these gaps and issues.

4.1. Typical Structure of Today's Centralized Cloud-Based Implementations

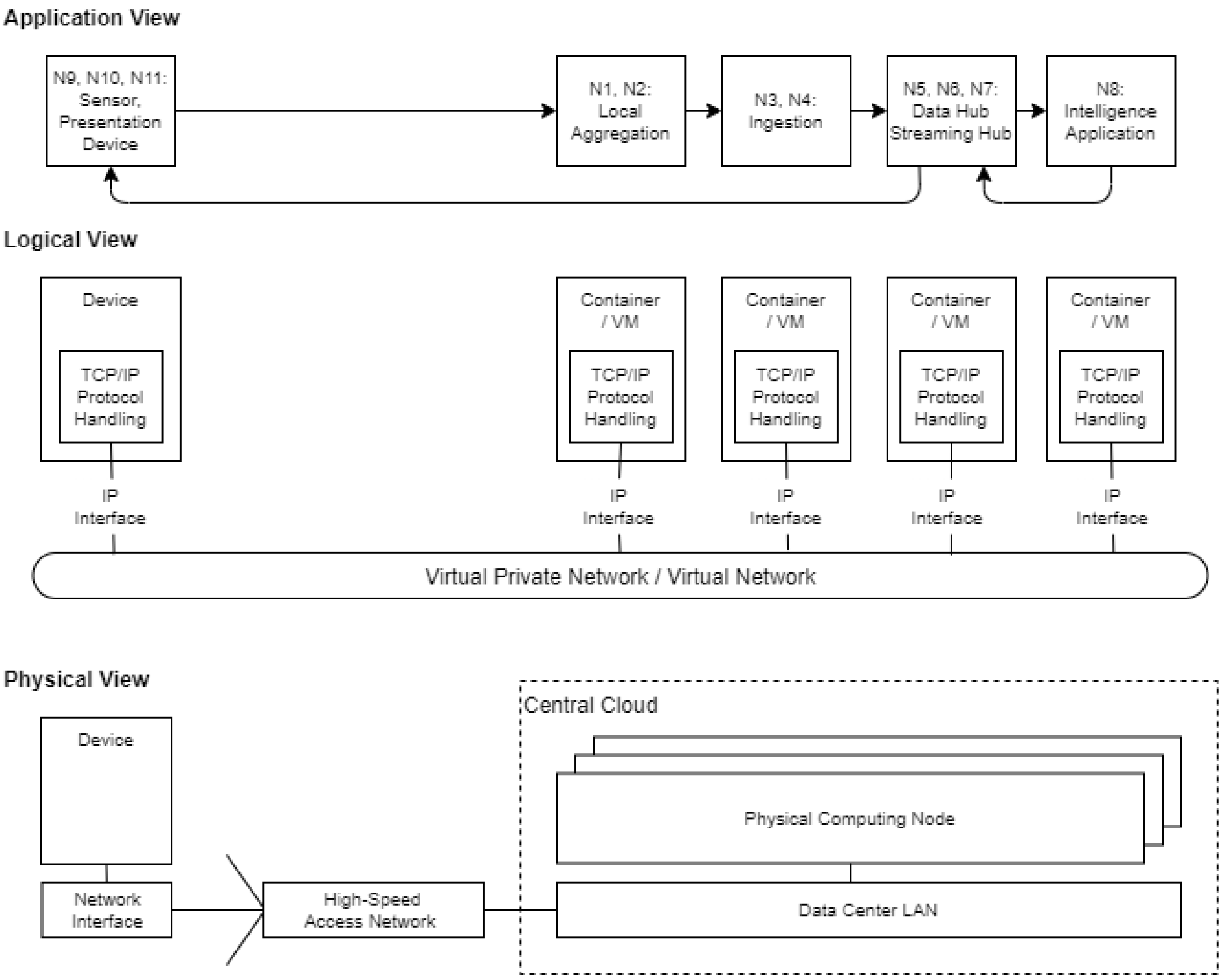


Figure 4.1-1: Today's Centralized Cloud-Based Implementation

Figure 4.1-1 shows a typical structure of today's centralized cloud-based implementation. Although there are several variants and more distributed approaches such as MEC, the basic strategy of today's centralized cloud-based implementation is a centralized computing approach. Most of the functional nodes are deployed in a central cloud of a single cloud vendor, and their application traffic is confined within the central cloud as much as possible. This is due to several reasons, such as the cost-effectiveness of a large-scale data center, data-transferring costs/overheads to other clouds within/without the cloud vendor, and the cost in time for application developers to educate themselves on a new cloud vendors' managed service.

Concerning the communication between devices (e.g., cameras and LiDAR sensors) and functional nodes on the central cloud, the devices are connected with the functional nodes via high-speed access networks and send their captured data to the applications. The TCP protocol is usually used for the transport layer protocol of this data transfer, even if the content is media type data stream (e.g., video images and point cloud data). Unlike pure UDP, TCP supports retransmission mechanisms to avoid degradation of service quality caused by packet losses in today's network at the expense of latency.

Inside a central cloud data center, computing nodes, e.g., virtual machines for Kubernetes nodes, are created with a hardware manager, and microservices for the functional nodes in Linux containers are deployed to computing nodes. Typically, gRPC or REST API is used to exchange data among the microservices, and service mesh is used for controlling the route of the traffic among the computing nodes.

4.2. Issues of Today's Centralized Cloud-Based Technology

Today's centralized cloud-based implementation model has the following technical issues.

4.2.1. Too Much Bandwidth Cost Caused by Centralized Cloud Computing

To build a system that runs various intelligence applications and situation logs based in a single central cloud data center, basically all data needs to be collected in the central cloud data center and analyzed there. The benchmark model for the Area Management Security Use Case (AM Security UC) assumes 1,000 video cameras and 1,000 LiDAR sensors are installed in each monitored area. If Motion JPEG is used for the video image compression and G-PCC is used for the point cloud data compression from LiDAR sensors, the aggregated traffic of video image and point cloud data transmissions would amount to 67 Gbps from one monitored area to the central data center. Assuming 1,250 monitored areas are deployed in a Japan case as described in the Benchmark Model, the total traffic the central cloud data center needs to accommodate would be around 84 Tbps. Sustaining such a high bandwidth with today's network would be challenging or at least very costly. This issue can be reduced by processing data in a more geographically distributed manner. Considering the data flows shown in the DPD in section 3, transferring all data over the network and aggregating it in the central cloud data center can be wasteful because most of the data collected is never used. At best, this data is only used by the instances of functional nodes associated with the monitored area. Therefore, primary data handling and storing should be done closer to the origin of the data.

4.2.2. Lack of Deterministic Service Quality Caused by Best-Effort Networking

The benchmark model requires the system to handle large volumes of sensor data (e.g., 67 Gbps from one monitored area with Motion JPEG and G-PCC) continuously with relatively low response time (i.e., less than 1 sec, ideally 100 msec including processing time at the functional nodes). However, it is hard to establish end-to-end connectivity in today's network with guaranteed network service quality between devices at customers' sites and logical computing nodes (e.g., VMs and containers) at the central cloud. This makes it challenging to ensure the service level requirements of the Benchmark Model. For example, traffic congestion at packet-based network equipment may lower the effective bandwidth and increase the queuing delay. Such traffic congestion also causes packet losses resulting in additional latency due to packet retransmission by a specific network protocol layer (e.g., TCP). Pure UDP may be able to keep data handling real-time. Still, without any retransmission mechanism, data loss is inevitable. It may degrade the accuracy of the Intelligence Application nodes and/or lead to misjudgment by users (e.g., police officers and security guards). Even though photonic networks are fully deployed and managed within each network domain, packet congestion may still occur at packet-based network equipment at exchange points between the domains. Thus, a mechanism to flexibly deploy end-to-end network connectivity with guaranteed service quality is vital for satisfying the requirements in the Benchmark Model.

4.2.3. Virtualization Overhead for Tag-Based Multi-Tenancy Operation

Cloud is a vast resource pool consisting of hundreds of thousands of servers, storage, and network equipment, connected through the software-defined network technology. For example, AWS has built cloud data centers consisting of multiple CELLs (groups of system resources) connected through a clustered network with many routers which control network flow between resources [Cloud DC]. Google has built its data centers based on the Leaf-Spine architecture and deployed many software-based configurable network switches to establish virtual private networks and control packet routing [A. Singh]. It means when the network packet is sent out from any of the user instances, a sort of tag is added to the packet header by the hypervisor or Smart NIC of the host machine to establish the virtual private network. When the network switches/routers receive the packet, they must examine the packet header to determine the network path to forward it, apply throttling, and/or perform a security check. Such a networking process increases network latency and consumes a lot of energy. In addition, significant network latency decreases the performance of various user workloads. This is because the CPU cycles may be consumed for no reason while waiting for a network I/O. As a result, a massive amount of energy is also consumed in the user plane.

4.2.4. Unnecessary Power Consumption Caused by Constant Processing

Many of today’s cognitive and intelligence functions required in the Ingestion node and Intelligence Application node are designed based on a constant rate approach. In a constant rate approach, video cameras capture (or LiDAR sensors) and generate data at constant frame rates. The cognitive and intelligence function constantly performs inference from the incoming data. That is, it consumes computational resources regardless of whether any objects of interest exist in front of the camera (or LiDAR sensors) or not. For example, according to the Benchmark Model, the frame rate of cameras is 15 fps, and the image size is Full HD. Given this condition, even if the state-of-the-art GPUs (e.g., NVIDIA A100) are fully utilized, AI functions, e.g., object detection, for one monitored area may consume 4~30KW (depending on the AI model) and become a dominant part of energy consumption in the system. Therefore, resolving energy inefficiency resulting from a constant rate approach is a crucial issue.

4.2.5. Insufficient Resource Utilization Caused by Box-Oriented Computing Platform

Cloud vendors offer a wide variety of selection of computing instances in terms of the number of vCPUs and memory size. But as for instances supporting specific accelerators, the variety of the instance types is often limited, and this limitation lowers the hardware utilization rate. For example, cloud vendors offer an instance supporting NVIDIA A100 GPU, and its typical configuration has 96 vCPUs and 8 A100 GPUs. Through our estimation for the Ingestion node, it is revealed that if a large size AI model is used for the inference, half of the 96 vCPU will not be used, while if a small size AI model is used, only 3 out of 8 GPUs can be utilized because of a CPU bottleneck. This inefficiency is because available CPUs and accelerators highly depend on the hardware configuration. Only CPUs and accelerators on the same physical computing node can be available from a single instance. In addition, as data transfer is a significant workload for computing infrastructure, processing big unstructured data in the NIC, i.e., Smart NIC, is becoming a common practice. However, this approach depends on the processor capacity of the smart NIC. IOWN technology should support a way to relax these limitations and improve their flexibility to realize rack-scale computing.

4.2.6. CPU Overwhelmed by Software-Based Data Transfer

The DPD has many receive-and-forward type processes. For example, Ingestion nodes need to copy incoming streams and ingest them for multiple purposes, and Data Hub nodes need to deliver received data to multiple subscribers. In addition, the service mesh for microservices requires an additional transparent application-level proxy in each computing node. In today’s implementation, a socket interface is commonly used for these processes, and CPUs handle TCP/IP data transfer. CPUs execute the following three read and two write tasks:

- Reads the incoming packet from the NIC buffer and writes it to the socket buffer.

1. Reads the packet in the socket buffer to verify the checksum
2. Reads the packet in the socket buffer and writes its payload to the application buffer

- As the size of incoming data gets bigger, CPUs get overwhelmed with interruption, context switching, memory copy, and the TCP/IP protocol handling tasks. In Figure 5.3-1 in 5.3, we will explain this issue more specifically in the Ingestion node case. This has long been an issue in the computing industry. TCP Offloading (TCO) will alleviate this issue to some extent. But, without DMA, CPU resources will be consumed for data transfer from the NIC to the processor. Increased CPU resource consumption results in higher energy consumption.

4.2.7. Increased Energy Consumption and Latency Caused by Data Hub Tier

When using today's cloud, it is natural to build the system by combining various cloud services, such as message broker, object storage, database, and data warehouse. These services are designed to scale well independently. However, they are not intended to provide high energy efficiency and short latency. The reason is that, in these services, the process is divided into mutually independent simple functions, and those functions are executed in parallel at multiple stages. Therefore, a lot of data is transferred across various nodes. For instance, when querying data stored in the object storage, data is transferred across different data service nodes inside the object storage. Then, it is passed to the query component, and finally, necessary data is extracted. Therefore, data processing latency and energy consumption become larger. Another example is the data warehouse. Many data warehouses are designed to store the data in a columnar format and run a query in a full-scan manner. Maintaining data indexes and statistics causes large load fluctuations when updating data, so it is not well-suited with functional decomposition's scaling method.

5. Initial Reference Implementation Model

This section illustrates an initial Reference Implementation Model (RIM) for the given Benchmark Model leveraging the IOWN GF's architecture and technology. In section 3, we analyzed what functional nodes are required for the given Benchmark Model and how much traffic needs to be transferred throughout the system. In section 4, we also identified several significant technology gaps and issues in today's centralized cloud-based technologies. In this section, 5.1 shows the basic strategy to solve these issues by utilizing IOWN technology. Then, 5.2 and 5.3 describe a concrete design of a geographically distributed data pipeline and its functional nodes for this use case. Finally, 5.4 explains the expected benefits of the IOWN GF RIM.

5.1. Basic Strategy of IOWN GF Architecture and Technology Adoption

The Area Management Security Use Case (AM Security UC) requires a highly flexible overall infrastructure that at the same time needs to fulfill higher specifications than many of the most demanding cloud use cases today. 4.2 introduced which issues arise when attempting to implement the AM Security UC with today's technology.

This subsection briefly revisits and summarizes each of these issues and highlights how the IOWN GF Overall Architecture and its high-level components will transform the communication and computation landscape to overcome the issues of today's centralized cloud-based technology for realizing the AM Security UC.

5.1.1. Shift from Centralized Cloud Computing to Distributed Cloud Computing

The AM Security UC requires high-bandwidth data flows between its functional nodes but routing to and processing these data flows inside a single data center is inefficient and costly.

As a solution, the DCI functional architecture document [IOWN GF DCI] describes seamless management of communication and computing: The DCI architecture provides interfaces that allow users to jointly request and manage both network and computing resources. This is the enabling feature to make the deployment of geographically distributed applications beyond a single data center practical. The geographically distributed data pipeline for this AM Security UC will be discussed in 5.2.

5.1.2. Shift from Best-Effort Networking to Deterministic Quality Networking

Besides the challenges of geographically distributed deployment as outlined above, the AM Security UC requires communication links with guaranteed quality of service regarding bandwidth, latency, and reliability between its functional nodes. However, current technology provides neither standardized nor cost-efficient methods to establish such high QoS links over long distances.

The Infrastructure Orchestrator of the IOWN GF overall architecture provides seamless management of geographically distributed deployments by integration with the DCI. It also enables automatic provisioning of Function Dedicated Networks (FDNs), which are end-to-end paths with guaranteed quality-of-service over multiple network segments spanning long distances. It achieves this goal via integration with the Open APN [IOWN GF Open APN]. In the context of the AM Security UC, this orchestrator provides a solution to the particularly critical issue of realizing extreme long-range user-data dataflows such as the high-bandwidth transmission of the aggregated sensor data of the AM Security UC.

Furthermore, with today's technology, such high QoS links can only practically be realized using dedicated and statically allocated optical fiber lines. This method would be highly inflexible, wasteful, and extremely costly. With its seamless and global resource management, the described DCI architecture is the key enabler to dynamically configure and cost-efficiently exploit the end-to-end optical paths underlying networks, such as an Open APN network, will offer.

5.1.3. Shift from Tag-Based Multi-Tenancy Operation to Wavelength-Based Multi-Tenancy Operation

Open APN and IOWN GF DCI technologies make a paradigm shift on to how the Cloud DCs are built. In today's cloud, software-controlled switches are installed in multiple stages to physically connect large numbers of resources and build software-defined virtual private networks. Such network configurations increase the delay in inter-node communications and increase the cost and energy consumption associated with the Cloud control plane. The IOWN GF's DCI technologies will innovate and improve the means for building the cloud's internal network. Specifically, the functional cards installed with to the physical server (such as Smart NIC, Infrastructure Processing Unit (IPU) [IPU], and Data Processing Unit (DPU) [DPU]) and interfacing with FDN (or FDN interface cards) will dynamically create a private channel between the servers assigned to the user tenant through DCI Gateway connected to Open APN. Suppose the network is accelerated where communication frequently occurs, such as between application servers and the DB server, or between two DB cluster servers; in that case, the system performance will be increased, and the system cost and energy consumption will be largely reduced.

5.1.4. Shift from Constant Processing to Event-Driven Processing

With today's technology, video AI analysis data pipelines with extremely high bandwidth need to be implemented with a constant rate approach and a fixed data flow path through accelerator hardware to achieve high energy efficiency. To reach its efficiency and performance goals simultaneously, the AM Security UC will need to break away from such constant-rate approaches but instead employ an event-driven approach. Event-driven approaches reduce the complexity of spreading computation over various accelerators and automatically adjust to varying load conditions expected from the use case. This type of approach also lends itself to building flexible multi-stage analysis pipelines that help save energy by identifying images for which no AI inference is required, thus saving compute resources. A way to apply an event-driven approach will be described in Annex D.

5.1.5. Shift from Box-Oriented Computing to Disaggregated Computing

The previous paragraphs described how the AM Security UC requires management solutions for powerful large-scale resource management. Issues and requirements also appear on a smaller scale within datacenters: the combinations and ratios of general-purpose CPUs and accelerators that are required vary over time, due to external conditions such as weather, weekday, time of day, sudden irregular events, as well as the configuration of the AM Security UC implementation. The fixed hardware configurations of today's cloud servers would cause large amounts of installed hardware to remain inefficiently utilized.

To solve this issue, the DCI architecture includes disaggregation technologies to dynamically adapt the hardware composition of individual servers with the required combination of hardware parts and AI accelerators to the varying load situation. This increases overall hardware utilization and reduces the required amount of hardware for this use case.

5.1.6. Shift from Software-Based Data Transfer to Hardware-Based Data Transfer

The global resource management provided by the IOWN technologies will remove the hurdle of provisioning low-latency and high-bandwidth connections between endpoints in a straightforward manner. With this solution in place on the infrastructure level, the remaining issue from the AM Security UC requirements is that network latency must be reduced. This means both latency due to forwarding data through the network and the communication latency due to protocol processing. Additional issues are tightly intertwined with protocol processing related to the energy consumption and overall efficiency of the AM Security UC system.

The DCI functional architecture proposes to use the following three mechanisms to reduce transmission latency and increase the efficiency of communication in terms of processing time and energy consumption:

- **Memory copy reduction:** Removing the intermediate step to copy data to kernel memory once before the transmission reduces transfer latency and energy consumption by reducing the total number of required memory accesses for sending and receiving data over the network.

- **DMA/RDMA-based data transfer protocols:** Transfer protocols such as (R)DMA that lend themselves to hardware offloading contribute to possible link utilization, latency reduction, and energy efficiency. The highest-bandwidth data flow between functional nodes of the AM Security UC exhibits highly regular traffic flow patterns that are expected to be implementable with fully hardware-off loadable RDMA protocols in a straightforward manner.

- **Three tiers of transmission distance range:** The AM Security UC dataflows traverse a wide variety of infrastructure between functional nodes. Each type of communication infrastructure poses unique challenges to reducing overhead and power consumption. For the acceleration of the data plane, the DCI functional architecture document identifies three main classes of connectivity between functional nodes based on communication distance range. Solutions are presented, laying out how today's communication protocols need to be adapted to optimize transmission performance for each of these transmission distance ranges.

5.1.7. Shift from a Single-Function, Low-Speed Data Hub to a Multi-Function, High-Performance Data Hub

In today's cloud, Data Hub services, or Platform-as-a-Service (PaaS) solutions, are designed to focus on scalability and stability. For scalability, a structural unit of the system is determined, and the system is expanded by adding the unit as the workload requirements increase. For the stability purpose, the functionality of that Data Hub is determined to cover very limited scenarios only, such as just queuing the messages, accessing data based on the primary key only, etc. The workload does not vary from user to user. By offering a variety of Data Hub services designed with such a concept, Cloud has facilitated resource sharing and attracted a large number of users. However, with digitalization advancement, the situation has changed as the need to process more data in real-time has emerged. In order to realize digital services as described in the use case document [IOWN GF CPS UC], multiple Data Hubs services need to be combined. In today’s implementation model, each Data Hub service takes a not small amount of processing time, so if the system is built by combining multiple Data Hub services, then the real-time requirement will not be fulfilled. It also creates other issues in system cost and energy consumption because the large amount of data needs to be transferred between Data Hub services. Such data hub relevant issues, combined with the inefficiencies of the cloud's internal network mentioned above, make issues, such as energy consumption, more difficult.

IOWN technologies provide us the following essential solutions to these issues:

1. Building high-performance Data Hub
   Increasing performance of each Data Hub by connecting servers that make up the Data Hub through IOWN GF’s low latency and high bandwidth network.

2. Realizing Multi-functional Data Hub
   By increasing the flexibility to combine various system resources such as CPU, GPU, memory, and accelerator that form a Data Hub cluster, it realizes the multi-functional Data Hub that causes different workloads depending on the user.

3. Streamlining Data Hub-to-Data Hub, or Data Hub-to-Client communications
   Open APN accelerates a large amount of data ingestion and export to/from any Data Hub service. This ensures that the network does not become a bottleneck and efficiency of data processing is guaranteed, even when several Data Hubs are jointly used.

5.1.8. Summary

The AM Security UC requires new approaches to data processing: seamless and global resource management, optimized communication mechanisms, and a high-volume, low-latency event-driven workload scheduling framework. DCI architecture provides the tools required for resource management and communication, and with these technologies, the basic building blocks for a new kind of Data Hub for workload scheduling.

The following subsections will elaborate in detail on the realization strategies of the AM Security UC, including the IOWN Data Hub to realize a flexible and high-performance workload scheduling framework.

5.2. Geographically Distributed Data-Pipeline of IOWN GF RIM

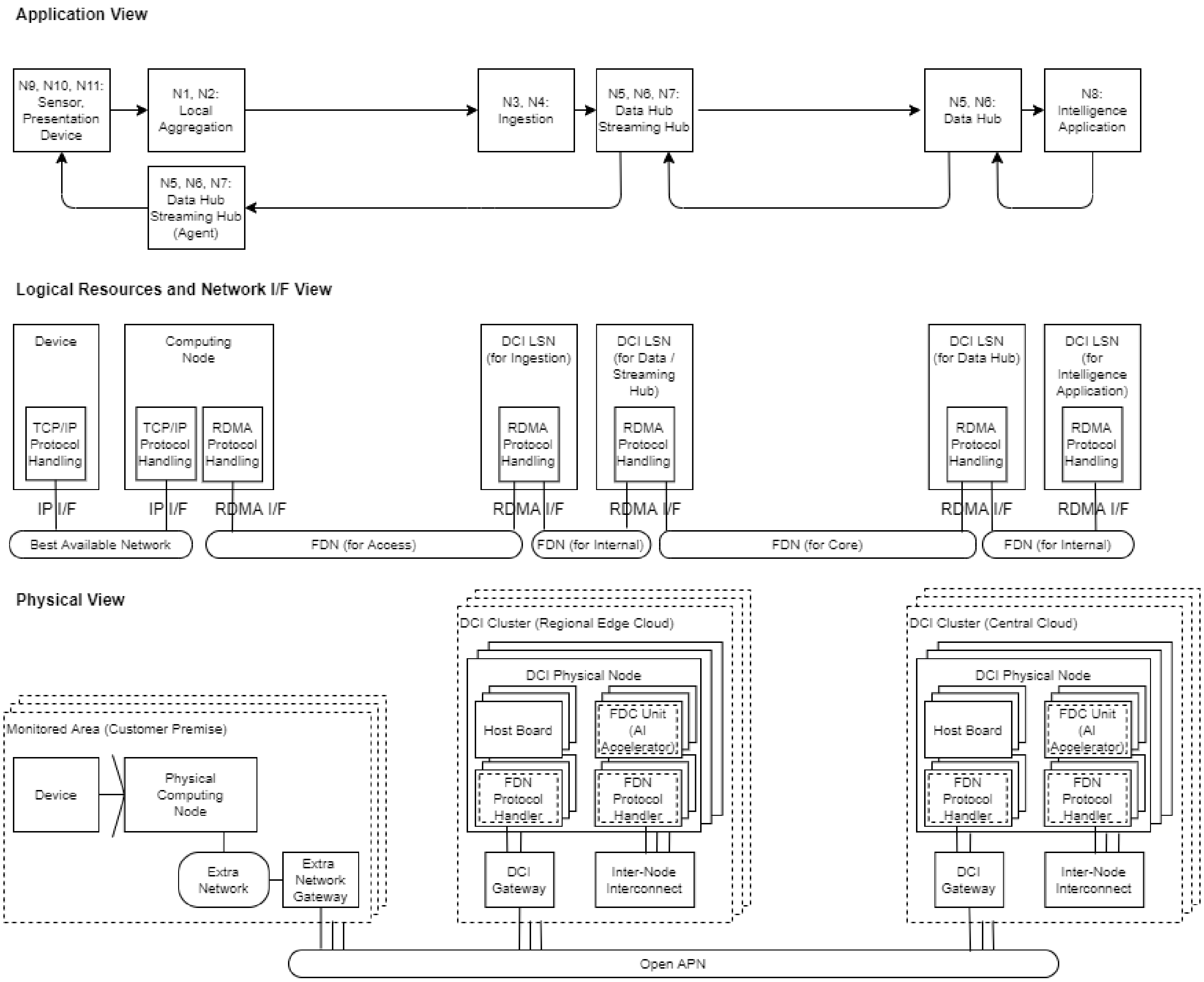


Figure 5.2-1: Overview of an IOWN GF RIM

Figure 5.2-1 shows an overview of an initial form of an IOWN GF RIM. In contrast to the centralized approach in today's implementation described in section 4, the RIM adopts a geographically distributed approach leveraging IOWN technologies such as Open APN and the DCI subsystem. This approach contributes to network-wide flexible provisioning of large-capacity and application-dedicated networking and computing resources. It significantly improves the end-to-end latency and system efficiency of this AM Security UC. For more details of DCI, see subsection 5.2 and section 7 of the DCI document [IOWN GF DCI].

As shown in Figure 5.2-1, the RIM assumes three layers of the vertical distribution of computing sites, that is,

- Customer premises,
- Regional edge clouds, and
- Central clouds.

Customer premises are supposed to be in each area to be monitored by various sensors like cameras and LiDAR sensors. Their networking environments may vary from one to another, and available computing resources are limited. Therefore, the RIM does not put substantial limitations on customer premises. The RIM only assumes that devices are connected via the best available network for the monitored area, and conventional small servers with the functionality of the Local Aggregation node are installed, working as a gateway between today's IP-based networking environment and the networking environment using IOWN technologies.

A central cloud is supposed to be a large-scale data center. It is assumed that there are a few central clouds in a nationwide area, and each of them has an abundant amount of computing resources of various types. The merit of using central clouds is the cost-effectiveness while accommodating many monitored areas with few central clouds in this use case may overwhelm the network and computing capacities as discussed in 4.2. Therefore, RIM put regional edge clouds in-between the customer premises and the central clouds to resolve the issues in today's centralized cloud-based implementations.

The regional edge clouds should be middle-scale data centers and placed in each region (e.g., local administrative unit or multiple municipalities) in the Large Area in 2.1.2. The primary roles of the regional edge clouds are to perform intermediate processing of the DPD, store the relevant data, and deliver only minimum necessary data toward the central clouds and monitored areas.

Both central clouds and regional edge clouds should support DCI clusters to accommodate the workload of the functional nodes flexibly and efficiently. Although there are several options in mapping the functional nodes, as shown in Figure 5.2-1, this RIM proposes deploying Local Aggregation nodes, Ingestion nodes, and Intelligence Application nodes onto customer premises, regional edge clouds, and central clouds, respectively. In addition, the components of the Data Hub node reside in each computing site to facilitate data preservation and exchange among the computing sites in a geographically distributed manner. In this functional mapping, video image data for on-demand replay and live streaming are returned to users' Presentation Device nodes from the Data Hub node at the regional edges cloud. In addition, only labeled data are delivered to Intelligence Application nodes at the central clouds. These functional nodes are realized as DCI LSNs with hardware configurations suitable for the workloads of the functional nodes.

These computing sites are connected via the Open APN, and FDNs are deployed over Open APN to connect LSNs on DCI clusters and/or computing nodes at customer premises to enhance the service quality of the networks required in this use case (e.g., latency, bandwidth, jitter, and loss rate). An example of an FDN for this AM Security UC is Converged Enhanced Ethernet (CEE). RDMA protocols are applied to data transferring over the FDNs instead of conventional TCP/IP protocols to efficiently exchange the large volumes of data required in this use case. Note that the IOWN GF overall architecture defines "extra networks" to extend FDNs outside of DCI clusters and Open APN to deploy FDN across customer premises and the regional edge cloud. Flexible bridging Service (FlexBr), which is defined in Annex A in the Open APN functional architecture document [IOWN GF Open APN], offers a forwarding service that aggregates traffic from the extra networks for customer premises and sends them over an optical path provided by the Open APN. The FlexBr defines several service types, and Type D2, which provides P-to-P connectivity with bandwidth reservation and strict latency management, is suitable for this AM Security UC.

The video image data and point cloud data transferred between computing sites are expected to be compressed by low-latency and inter-frame independent schemes, i.e., Motion JPEG and G-PCC, since the FDN over Open APN can efficiently support the higher bandwidth requirement of Motion JPEG and G-PCC. An exception is for video image data transferring from Data Hub nodes to Presentation Device nodes for on-demand replay. Since on-demand replay does not require severe low latency, transcoding from Motion JPEG to H.264 can be applied to save the network bandwidth.

Figure 5.2-1 shows a view of the three-layered structure in a normal condition with cost-efficiency. The structure, including the Open APN topology, should be flexible enough to follow the dynamic change of networking and computing resource demands based on the service conditions. DCI architecture and Open APN also contribute to supporting the dynamic configuration of the infrastructure to meet the change. Many reasons can cause the change of networking and computing resource demands. For example, monitored areas may be added/removed as the business progresses of this AM Security UC. An unexpected increase in the number of video streams downloaded by Presentation Device nodes may overload the provisioned networking resources. Support of the event-driven approach in Annex D also requires high elasticity of the networking and computing infrastructures. DCI architecture defines a service exposure function that exposes system operation services to cope with these changes. This function allows business owners and/or specific functional nodes in the DPD to request the addition/deletion of networking and computing resources across multiple computing sites (e.g., LSNs and FDNs over Open APN) on demand and keep the loads on the resource at an appropriate level.

These ways significantly reduce the overall network traffic demands and improve the efficiency and responsiveness of the system.

Annex E describes a possible deployment example of the functional mapping for Japan and shows their estimated traffic demands.

5.3. Application’s Functional Node Structure of IOWN GF RIM

This subsection describes a concrete design of the functional nodes in the geographically distributed data pipeline described in Figure 5.2-1.

5.3.1. Local Aggregation Node

Local Aggregation nodes are placed in customer premise sites to aggregate sensor data and efficiently transfer them to the subsequent functional nodes on DCI-based regional edge and central clouds. As described in 5.2, the Local Aggregation nodes are installed on conventional small servers and are supposed to support the basic functionality described in the DPD in section 3. In IOWN GF RIM, the Local Aggregation nodes are further recommended to keep the following features to improve the overall efficiency.

- Local Aggregation nodes should support RDMA I/F to efficiently transfer aggregated sensor data to the Ingestion node on the DCI-based regional edge cloud.

- Local Aggregation nodes may have the functionality of primary analysis for the event-driven approach described in Annex D.

5.3.2. Ingestion Node

Ingestion nodes are one of the most computationally intensive functional nodes in the DPD. Ingestion nodes are expected to replicate the received data, convert them according to their intended usage, and ingest the processed data onto the Data Hub node. Significantly, the cognition process in Ingestion nodes consumes relatively higher computing resources for CNN-based AI inference. In other words, it can be said that Ingestion nodes are nodes that can benefit from IOWN-based technologies.

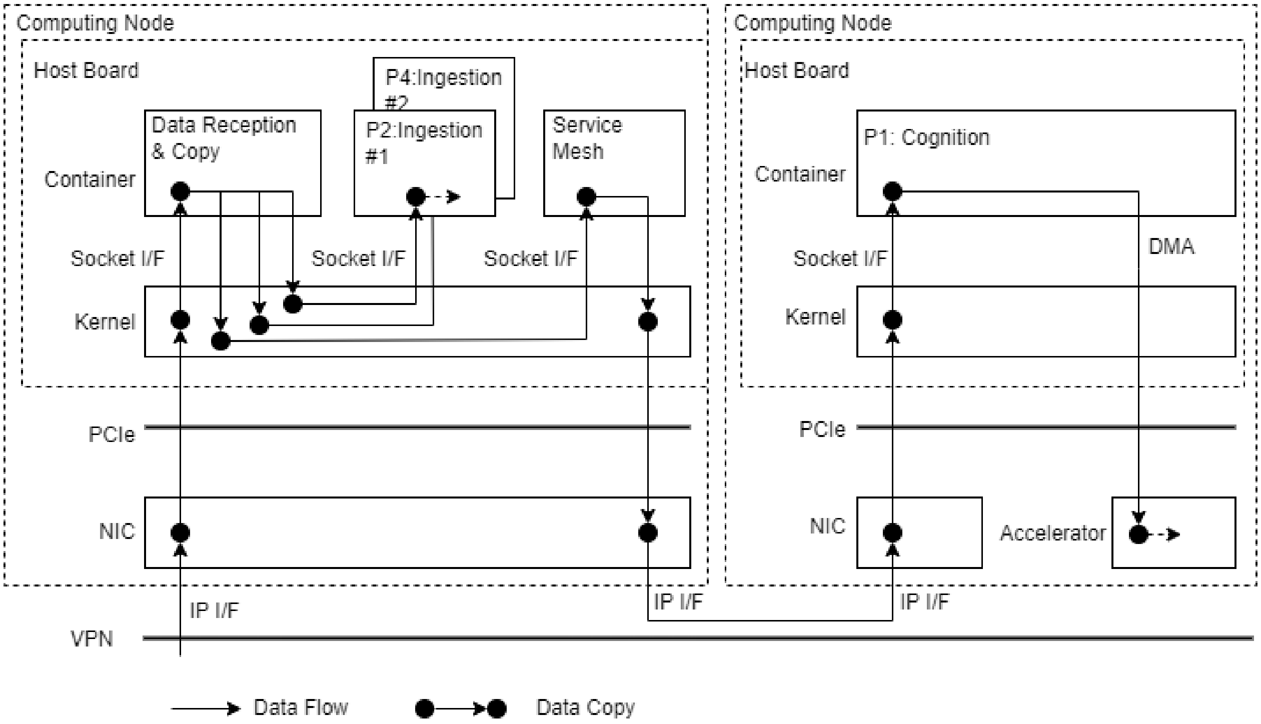


Figure 5.3-1: Typical Today’s Implementation of Ingestion Node (Flow of Video Image Data)

Figure 5.3-1 shows typical today’s implementation of an Ingestion node employing the microservice architecture. Figure 5.3-1 focuses on the flow of video image data and illustrates why the “CPU Overwhelmed by Software-Based Data Transfer” issue in 4.2 can be critical. As described in C.3.3 in Annex C, the Ingestion node has three major processes, i.e., “P1: Cognition”, “P2: Ingestion for On-Demand Replay (or P2: Ingestion #1)”, and “P4: Ingestion for Live Monitoring (or P4: Ingestion #2)”. As shown in Figure 5.3-1, these processes are deployed as containers running in the user space on two computing nodes with different hardware configurations, i.e., with accelerators and without accelerators. In addition, there are two additional containers for appropriately controlling the dataflows. One is to queue and duplicate received video image data for the three processes (“Data Reception and Copy” in Figure 5.3-1), and the other is to route one of the duplicated video image data streams to the “P1: Cognition” container on the other computing node (“Service Mesh” in Figure 5.3-1). As shown in Figure 5.3-1, as long as conventional IP and socket interfaces are used, many memory copies of the whole video image data are necessary at any layer and can be a bottleneck restricting the efficiency of the Ingestion nodes.

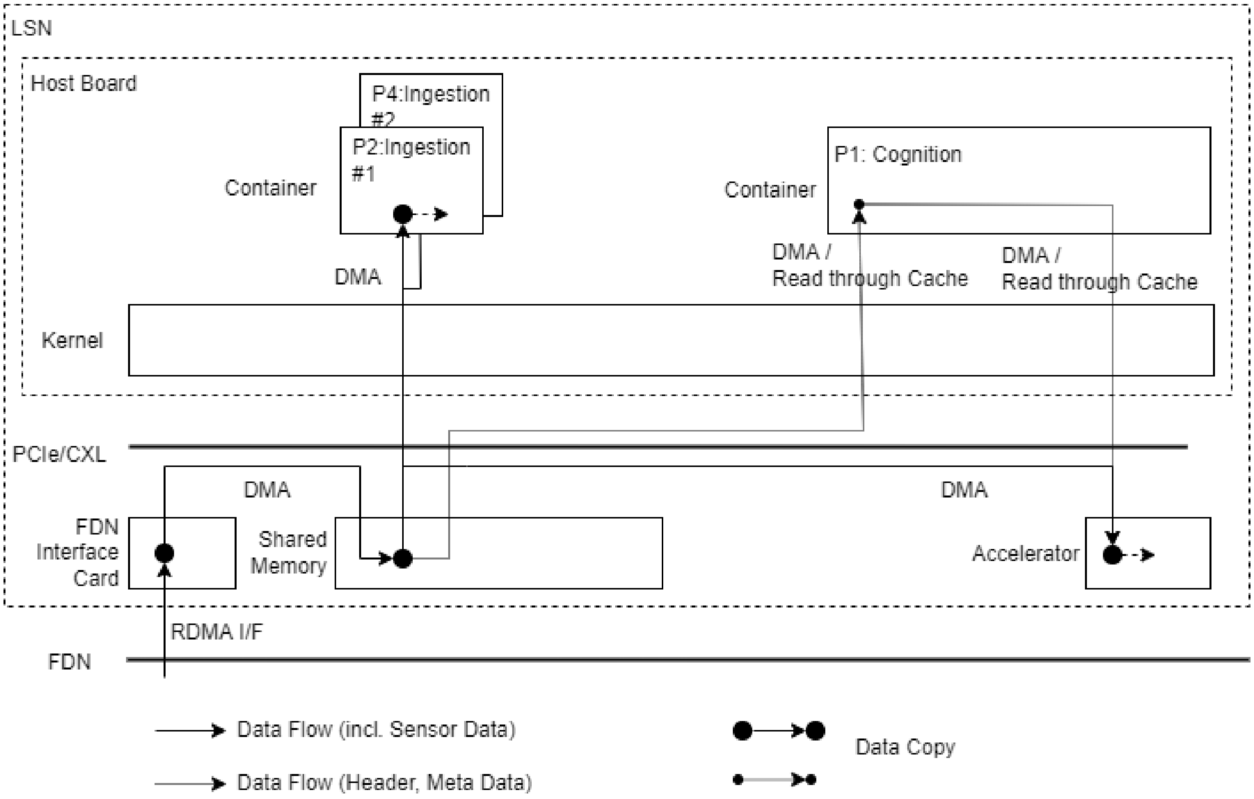


Figure 5.3-2: Example of Implementation of Ingestion Node with IOWN technologies (Flow of Video Image Data)

As summarized in 5.1, the IOWN GF RIM recommends utilizing state-of-the-art data-plane acceleration techniques (e.g., memory copy reduction and DMA/RDMA-based data transfer protocols) together with DCI architectures to streamline the redundant data flow. Figure 5.3-2 shows an example of the implementation of the Ingestion node using IOWN technologies. The data transfer can be terminated within the FDN interface card by utilizing RDMA-based data transfer between the Local Aggregation nodes and the Ingestion node. The received data can be pushed to the shared memory without disturbing any CPUs on the host board. The shared memory can be embedded either on the FDN interface card or an external functional card. Then the data consumers (i.e., P1: Cognition, P2: Ingestion#1, and P4: Ingestion#2) pull data necessary for processing from the shared memory with DMA. Since IOWN GF RIM adopts the event-driven approach shown in Annex D for the Ingestion node, the Cognition process (P1) can only fetch the metadata (or the results of primary analysis) of the frame, check the necessity of the secondary analysis, and trigger transfer of just the related image data to the accelerators for secondary analysis. This way significantly decreases the number of memory copies and the amount of transferred data within a computing node, resulting in the improvement of energy efficiency of the Ingestion node.

Note that, in Figure 5.3-2, it is assumed that one instance of the Ingestion node is implemented on a single LSN. However, due to the limitation of the number of hardware resources on a DCI physical computing node, the LSN may be split into two or more LSNs on different DCI physical computing nodes, e.g., one for the Ingestion processes (P2 and P4), and the other is for the Cognition process (P1) with accelerators. In this case, the stored data onto the shared memory may be transferred from one LSN to the other by using RDMA I/F, but it would still be efficient compared to the data transferring with conventional socket I/F.

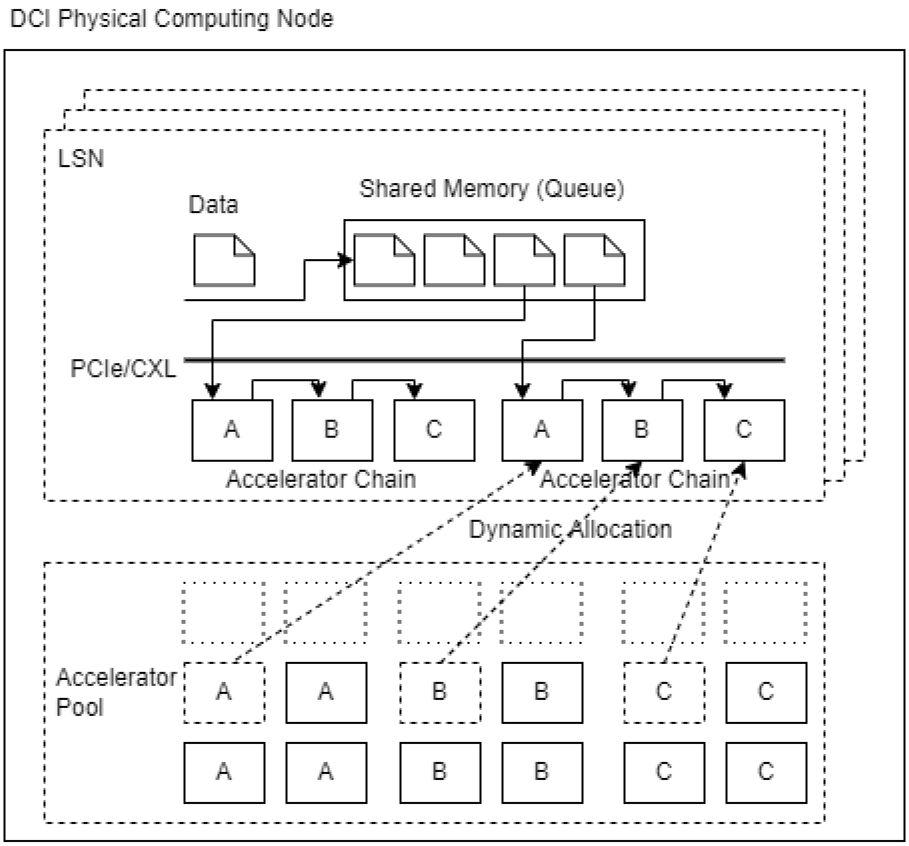


Figure 5.3-3: Utilization of Heterogeneous Accelerators

The cognition process in the Ingestion nodes is supposed to perform AI Inference with CNN for a massive number of video image streams. In addition, because of the adoption of an event-driven approach, the workloads for the secondary analysis in the cognition process may dynamically change depending on the primary analysis results. Thus, efficient and flexible utilization of hardware accelerators can be a key requirement for the Ingestion node. When comparing the energy efficiency of FPGAs and GPUs, there are also advantages and disadvantages depending on the type of the workloads [M. Qasaimeh]. Many special-purpose accelerators with lower energy consumption are also emerging [K. Matsubara] [Silicon Photonics]. Therefore, the IOWN GF RIM is recommended to support flexibly building accelerator chains combining heterogeneous accelerators optimal for each portion of the workloads, as shown in Figure 5.3-3. In addition, accelerator chains should be dynamically added/removed from/to accelerator pools to flexibly scale the performance of LSNs for the Ingestion nodes according to the change of the workloads.

5.3.3. Data Hub Node

The Data Hub node is a message broker system in the Benchmark Model for the AM Security UC. Its roles are to stably receive data, such as video stream segments, point cloud data, labeled data which represents recognition results, user data such as position, and/or voice message, with high throughput, and to ensure its persistence while delivering data to its consumer applications with low latency. To accelerate its performance and improve functionality, the solutions will include the IOWN Data Hub service for the message broker with the following characteristics, as described in the IOWN GF Reference Document for the IOWN Data Hub [IOWN Data Hub].

- The local resource pools for the message broker system will consist of multiple interconnected servers that are interconnected through the inter-node interconnect mechanism, which provides the peer-to-peer direct optical paths between nodes assigned to the same tenant/purpose.

- To guarantee the data persistence locally, replica sets that consist of multiple nodes are formed in the regional edge cloud, and the message data is replicated across them.

Note 1: The number of replica sets is configurable, depending upon the write throughput requirement.
Note 2: The number of nodes that make up each replica set is configurable, depending on the availability and read throughput requirements.

- To shorten the latency for data processing, persistent memory is used to store the data. When the data is replicated across nodes locally, IOWN GF DCI technologies that enable RDMA communications for the persistent memory are used.

- To support distributed data processing (e.g., the central cloud to consume data stored in the regional edge cloud), the message data may be replicated to the remote site asynchronously and/or request-based through the Open APN at low latency. In addition, when replicating data, data may be compressed through the irreversible method with the support of an FDN interface card accelerator, to reduce the network traffic while if it is specified.

- If the global data consistency needs to be achieved, the message arrival event is notified to the remote node synchronously to allow remote applications to wait for the message body arrival.

Note: The message arrival event can be notified asynchronously if the eventual consistency is acceptable.

- To balance the workload across the nodes in the replica set, 1) IOWN Data Hub technologies will configure it all-active (all-master) so that all nodes can accept data ingestion for assigned partitions, and 2) the message arrival event is sent from the node that accepts the data ingestion, but the message body data is transferred from one of the other nodes in the corresponding replica set to the remote site.

Note 1: Inter-site data transfer will be triggered upon both the request from remote clients in an event-driven manner or the server-push event trigger created by the Data Hub node itself.
Note 2: The remote client could be the remote Data Hub node and other types of application node such as the Streaming Hub node as described in the next section.

- When transferring the data to the remote site, Open APN network between two DCI Gateways is used. In addition, RDMA over Open APN WAN technology will also be used.

Figure 5.3-4 below shows such an IOWN GF RIM for the Data Hub node.

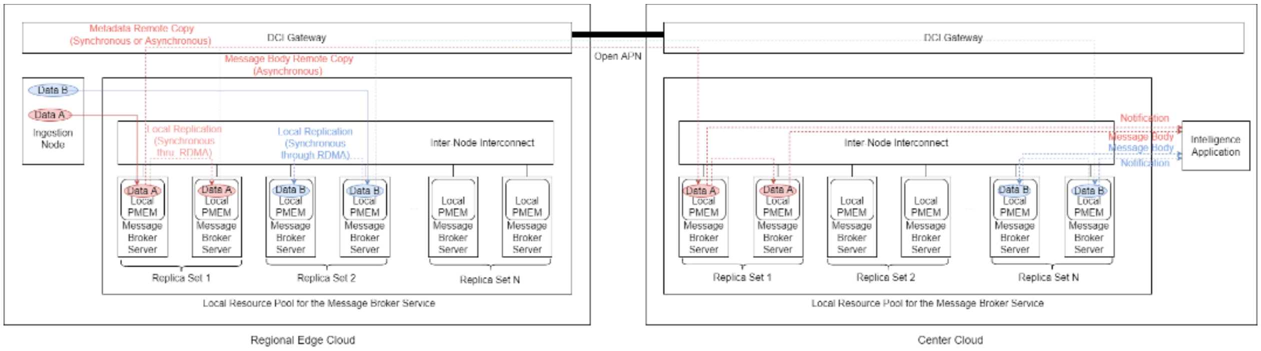


Figure 5.3-4: IOWN Data Hub RIM

5.3.4. Streaming Hub Node

The Streaming Hub is a special variant of the IOWN Data Hub for the message broker, in which data persistence is less critical as it will be guaranteed by the preceding node in a typical configuration. Still, it is very important to deliver the video streams to many remote clients at a minimum delay, e.g., less than a second. To enable real-time monitoring of monitored areas, the solutions will realize the IOWN technology-based Streaming Hub component which has the following characteristics.

- Data Transfer between the upstream regional edge cloud site and the downstream regional edge cloud site is conducted through the Open APN with the advanced protocol, such as RDMA over WAN
- Depending upon the number of clients that monitor the area, the video stream data is dynamically replicated to multiple streaming service servers in the downstream regional edge cloud.
- The IOWN GF DCI technologies accelerate video stream data replication that enables RDMA communication between nodes.

Figure 5.3-5 shows such an IOWN GF RIM for the Streaming Hub node.

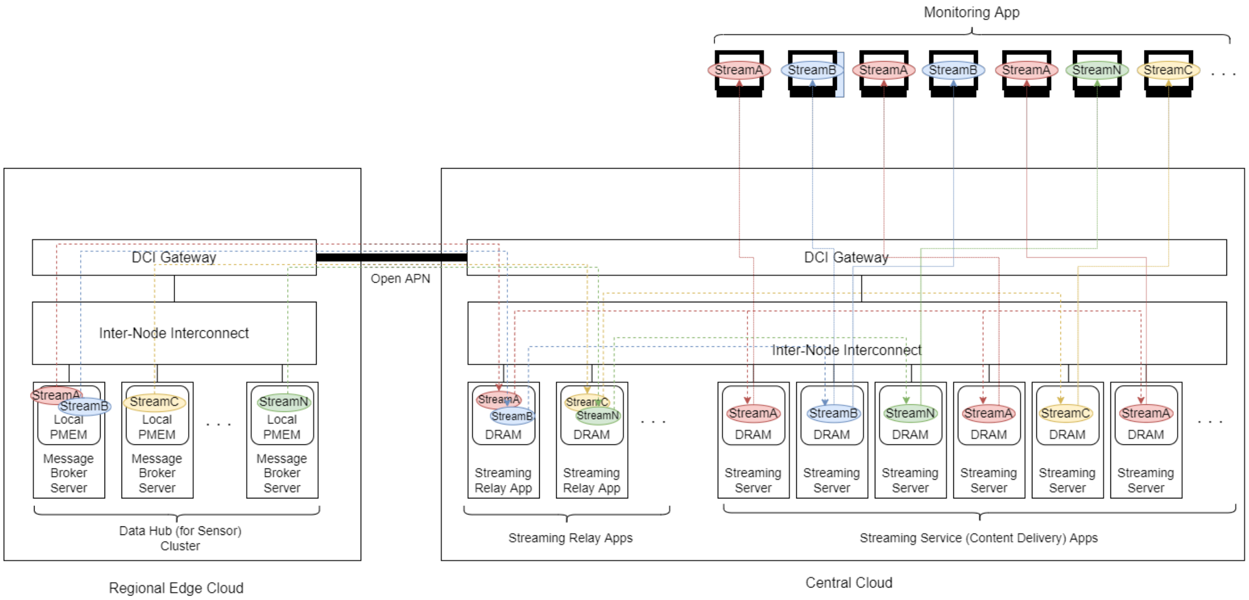


Figure 5.3-5: IOWN GF Streaming Hub RIM

5.3.5. Intelligence Application Node

In the central cloud, one or several Intelligence Application nodes will consume the labeled object and sensor data from the Data Hub nodes (for sensors) in a real-time manner, investigate the situation from different angles and make decisions automatically to mitigate the security risk. When a risk is detected, it also queries the positions of the security officers, generates a voice message explaining the risk situation, and delivers it to the security officers near the risk location. It also sends SMS messages to selected parties based on the predefined rules. In that sense, the Intelligence Application node is the most complicated functional node in the AM Security UC, consisting of the following features:

- Consume the labeled object and sensor data from the Data Hub nodes (for sensors) in a real-time manner
  Note: Due to the nature of inference processing, the time to process data varies, so rather than the Data Hub node pushing the data to the Intelligence Application node, the Intelligence Application node is basically pulling the data from the Data Hub node. Of course, if the inference application is light and there are enough resources available, then the data may be pushed to reduce latency.
- Validate and record the labeled objects, or inference results, on a Live 4D Map to keep tracking the movement and behaviors of recognized objects for a certain period of time.

Note 1: Inference applications in the ingestion node may contain errors. To eliminate errors, time-series consistency and the validity of spatial positions of the inference results are assessed before updating Live 4D Map.
Note 2: Live 4D Map may consist of four layers, each representing 1) the latest situation, 2) the history of situation change and detected object movement, 3) normal and/or abnormal movement path, and 4) static structure of the monitored area.
Note 3: Live 4D Map shall timestamp each data record, and store them on top of static structural information.
Note 4: Live 4D Map may associate relevant records to streamline the subsequent analysis when storing data records, or after storing them. To speed up the search for associations, such association information is recorded inside each data record, not in a separate table, as described for the Graph DB in the IOWN GF Reference Document for the IOWN Data Hub [IOWN Data Hub].

- Run the anomaly detection analysis against the Live 4D Map data to detect anomaly situations, e.g., where a suspicious man is trying to break through the security gate forcibly, etc.
  Note 1: Such advanced inference algorithms may include time-series analysis, geospatial analysis as well as pattern matching.
  Note 2: Those models may be trained by the deep learning algorithm, such as recurrent Neural networks, graph Neural networks, etc.

- Determine the required actions automatically
  Note: Actions include 1) sending alerts to the security officers, 2) highlighting detected anomalies on the monitoring monitor, and 3) controlling the security gates and/or the security robots

- Write the determined actions to the Data Hub node (for users) to initiate automated processes

The following figure 5.3-6 shows the expected data layers in Live 4D Map, 5.3-7 shows the data model to analyze the real-world situation, and 5.3-8 shows such a logical breakdown of the intelligence application platform.

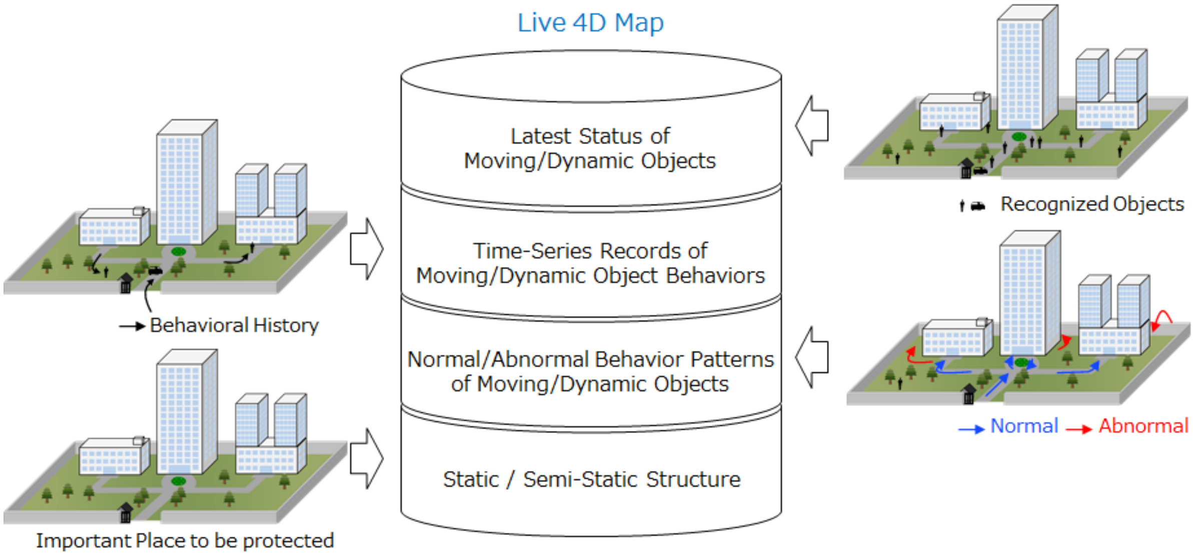
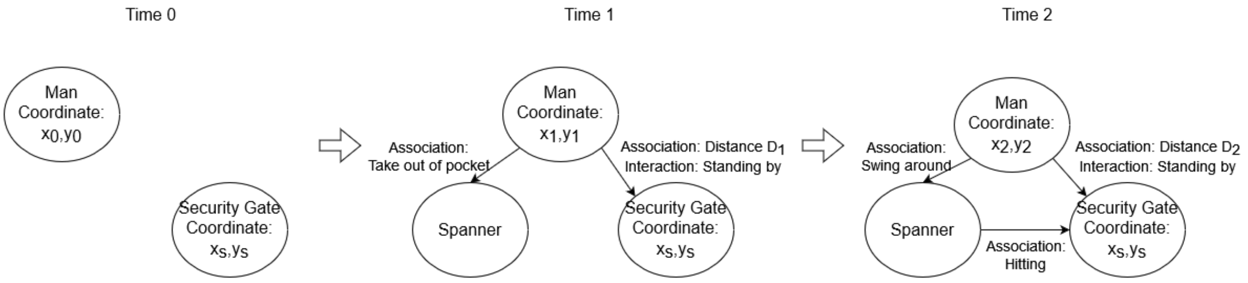

Live 4D Map
Latest Status of Moving/Dynamic Objects
Time-Series Records of Moving/Dynamic Object Behaviors
Normal/Abnormal Behavior Patterns of Moving/Dynamic Objects
Static / Semi-Static Structure
Behavioral History
Important Place to be protected
Recognized Objects
Normal
Abnormal

Figure 5.3-6: Data Layers Within the Live 4D Map using IOWN technologies



Object ID	Object Type	Time Stamp	Coordinate	Movement / Bahavior	Interaction with other object(s)
3	Security Gate	T0	x_s, y_s	Closed	N/A
1	Man	T0	x_0, y_0	Walking to direction A	N/A
3	Security Gate	T1	x_s, y_s	Closed	Man: There is a man in front of the security gate
1	Man	T1	x_{m1}, y_{m1}	Stopping	Spanner: Take spanner out of pocket Security Gate: Standing by the security gate
2	Spanner	T1	x_{s1}, y_{s1}	Being Lifted	Man: Held by man
2	Security Gate	T2	x_s, y_s	Closed	Man: There is a man attacking in front of the security gate Spanner: Hit by a spanner
1	Man	T2	x_{m2}, y_{m2}	Move his body violently	Spanner: Swing the spanner around Security Gate: Standing by the security gate
2	Spanner	T2	x_{s2}, y_{s2}	Being swayed	Man: Held and swung around by a man Security Gate: Hitting the security gate

Location information is managed with index to speed up (geo-)spatial query

Association information is included inside each record, to accelerate the relationship & interaction analysis

Time stamp information is added to the data records, to partition and/or sort them to speed up time series analysis

Figure 5.3-7: Data Model to Represent the Real-World Situation for the Anomaly Detection

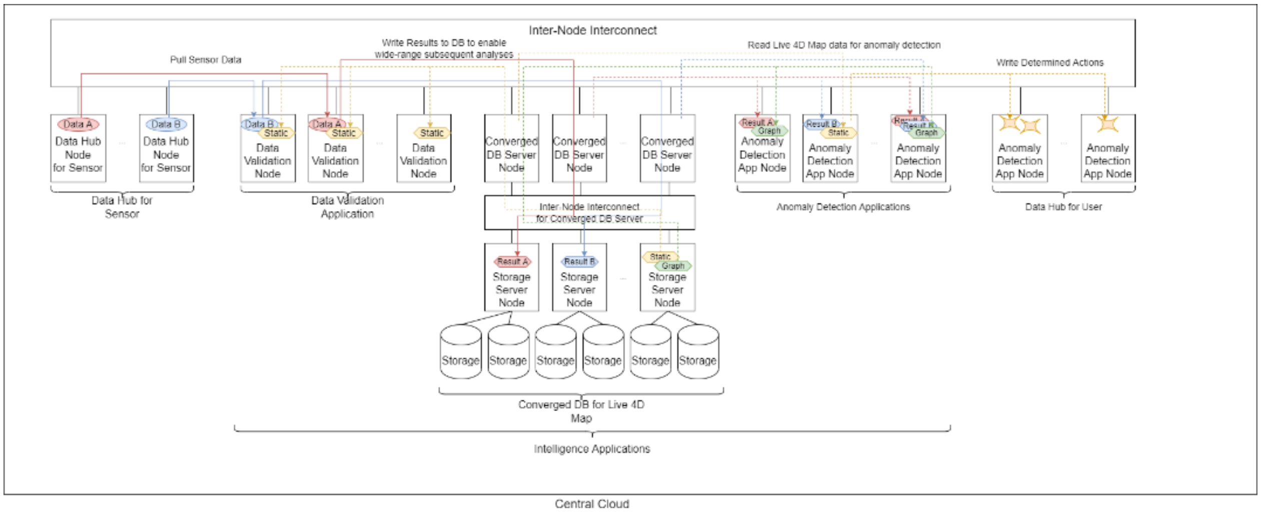


Figure 5.3-8 IOWN GF Intelligence Application RIM

5.4. Expected Benefits

In this section, we will discuss the benefits of IOWN technologies for AM Security UC. To highlight it, we will compare today's centralized cloud-based implementation model, and the IOWN GF RIM exhibited in Figure 5.4-1 below and then discuss how to make technology breakthrough for each component.

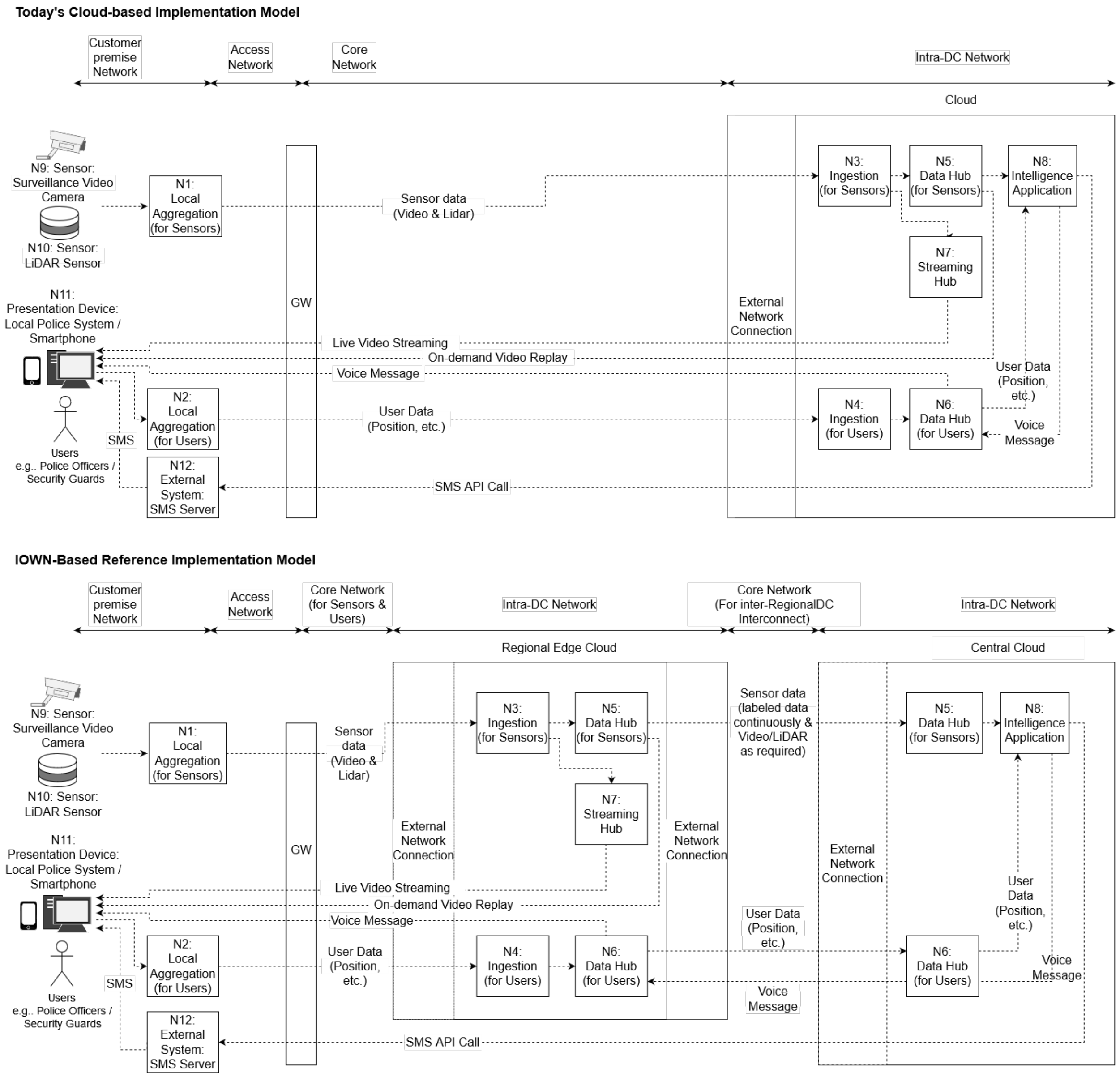

Today's Cloud-based Implementation Model
Customer premise Network
Access Network
Core Network
Intra-DC Network
Cloud
N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
N1: Local Aggregation (for Sensors)
GW
Sensor data (Video & Lidar)
External Network Connection
N3: Ingestion (for Sensors)
N5: Data Hub (for Sensors)
N8: Intelligence Application
N7: Streaming Hub
N11: Presentation Device: Local Police System / Smartphone
Live Video Streaming
On-demand Video Replay
Voice Message
User Data (Position, etc.)
N2: Local Aggregation (for Users)
SMS
Users e.g., Police Officers / Security Guards
N4: Ingestion (for Users)
N6: Data Hub (for Users)
User Data (Position, etc.)
Voice Message
N12: External System: SMS Server
SMS API Call
IOWN-Based Reference Implementation Model
Customer premise Network
Access Network
Core Network (for Sensors & Users)
Intra-DC Network
Core Network (For inter-RegionalDC Interconnect)
Intra-DC Network
Regional Edge Cloud
Central Cloud
N9: Sensor: Surveillance Video Camera
N10: Sensor: LiDAR Sensor
N1: Local Aggregation (for Sensors)
Sensor data (Video & Lidar)
GW
External Network Connection
N3: Ingestion (for Sensors)
N5: Data Hub (for Sensors)
N7: Streaming Hub
External Network Connection
Sensor data (labeled data continuously & Video/LiDAR as required)
External Network Connection
N5: Data Hub (for Sensors)
N8: Intelligence Application
N11: Presentation Device: Local Police System / Smartphone
Live Video Streaming
On-demand Video Replay
Voice Message
N2: Local Aggregation (for Users)
SMS
Users e.g., Police Officers / Security Guards
User Data (Position, etc.)
N4: Ingestion (for Users)
N6: Data Hub (for Users)
User Data (Position, etc.)
Voice Message
N6: Data Hub (for Users)
User Data (Position, etc.)
Voice Message
N12: External System: SMS Server
SMS API Call

Figure 5.4-1: Today's Implementation Model and IOWN GF RIM

5.4.1. Network

5.4.1.1. Access Network

The access network that typically spans up to 20 km to cover a local area connects the sensors and users' devices to the edge router in order to intermediate communications to/from the cloud services in the AM Security UC. Due to the amount of network traffic described below, we first consider the wired access network only when it comes to collecting data from surveillance video cameras and LiDAR sensors. Of course, the wireless access network remains a future study item.

Workload Profile

The amount of data traffic flowing through the access network is the same for both models. We pick up the deployment scenario in Japan and assume the network traffic volume as follows, based on the dataflow analysis described in section 3 and the deployment scenario described in Annex E.

- Upstream traffic
    - Video sensor data: 1,250 monitored areas x 1,000 cameras x 60 Mbps = 75 Tbps
      Note: Motion JPEG is assumed as a video data format
    - LiDAR sensor data: 1,250 monitored areas x 1,000 LiDAR sensors x 7 Mbps = 8.75 Tbps
      Note: G-PCC is assumed as a point cloud data format
    - User data (such as position, etc.): 1,250 monitored areas x 1,000 users x 1.5 KB/min = 250 Mbps
    - **Sub-Total: Approximately 83.75 Tbps**
- Downstream
    - Live video streaming: 1,250 monitored areas x 20 cameras to be monitored x 20 independent display systems per camera x 60 Mbps = 30 Tbps
      Note: Motion JPEG is assumed as a video data format
    - On-demand video replay: 1,250 monitored areas x 50 concurrent per-area VOD requests x 3 Mbps = 187.5 Gbps
      Note: H.264 is assumed as a video stream format (as latency does not matter)
    - Voice message: 125 concurrent streams x 100 x 64 Kbps = 800 Mbps
      Note: Assuming that voice messages are delivered at a rate of 6 seconds per minute for each monitoring area, that is, there are 125 concurrent voice messages, and each message is delivered to 100 security officers on average.
    - SMS traffic: 1,250 monitored areas x 100 messages per sec x 1.5 KB / message = 1.5 Gbps
      Note: Assuming the peak hour only when a security incident happens, 1,000 messages are sent to the people in the monitored area in a second.
    - **Sub-Total: Approximately 30.19 Tbps**
- **Up & Down-stream Total: Approximately 113.94 Tbps**

Expected Benefits

As mentioned above, the network traffic for the AM Security UC will be massive. Therefore, the required network equipment and power consumption for networking services will equally sizable. Open APN technologies, which provide high bandwidth and reduced power consumption, help mitigate this issue. In addition, the Open APN technologies that shorten the latency by reducing optical-to-electrical conversion and establishing a direct communication path between the customer premise site and the cloud DC will contribute to building the real-time monitoring services.

5.4.1.2. Core Network

The core network provides wide-area data transfer services, directs data sent from the sensor to the cloud services, and mediates information exchange between the user devices and the cloud services in the AM Security UC for both today's implementation model and IOWN GF RIMs. It also provides the inter-DC interconnect network services for the IOWN GF RIM.

Workload Profile

The amount of network traffic that comes from or flows into the customer premise site over the core network is the same as one for the access network, namely 113.94 Tbps in total. In addition to that, the IOWN GF RIM has the following network flows between the regional edge clouds and the central cloud. It should be noted that there are multiple edge clouds connected to one central cloud, and the numbers below represent the total amounts considering this.

- From regional edge clouds to central clouds
    - Remote data replication within the Data Hub node (N5 for sensors)
      Labeled data ingested into the regional edge cloud is replicated to the remote central cloud to execute AI algorithms against it to infer the situations and make required decisions. Label data is much smaller than raw data, so it's not a big issue for bandwidth.
      In addition, if it is required, video stream and LiDAR point cloud data are also replicated to the central cloud, to run the detailed analysis for confirmation there. As this data is of an enormous volume, we assume an event-driven mechanism to save bandwidth, which means that only when the Intelligence Application node (N8) detects any risks from labeled data and needs to conduct a more detailed analysis, then replication of the original sensor data with the lossless algorithm is triggered for a while. Through such a mechanism, we can reduce the amount of data transfer, for instance, by a factor of 100.
    - Query-based data transfer from the Data Hub node (N6 for users) to the Intelligence Application node (N8)
      When an incident occurs, a query is made to retrieve the user's latest location, etc., to promptly send alert messages to people nearby. However, this rarely occurs in practice because it is triggered only when an imminent danger is apparent. Therefore, we'll ignore this part regarding the amount of data transferred and only consider it for the end-to-end latency later.
- From central cloud to regional edge clouds
    - Voice Messages sent from the Intelligence Application node (N8) to the Data Hub node (N6 for users)
      Voice messages are delivered continuously to inform security officers of the status of the monitored area, and its total network traffic is 8 Mbps, which is one-hundredth of the number described in Section 5.4.1.1 because 100 security officers receive the same message.

Expected Benefits

In today's implementation model, the core network may transfer data between the customer premise site to the cloud over a long distance, such as 500 km, in the AM Security UC. To forward network packets over such a long distance, it must have more than several optical transceivers, and the latency will become significant at the end. For example, the one-way latency will be 4-5ms for a 500 km long network, although it shall be only 2.5 ms considering the speed of pure light. In addition, there need to be optical-to-electric and electric-to-optical conversions in the optical transceiver, to control the packet routing and amplify the signal. Such processing consumes a lot of energy and results in a lot of CO_2 emissions.

In the IOWN GF RIM, these issues are mitigated with the following two perspectives:

- The optical transceivers become all-photonic, which eliminates the necessity of having optical-electric-optical conversion, and contribute to reducing the network latency and energy consumption
- Regional edge clouds, which have relatively small resources, are used to ingest and preprocess the data, and only forward the detailed raw data to the central cloud as required through the event-driven mechanism in order to reduce significant data transfers over a distance. Such a geo-distributed processing model becomes

feasible because IOWN technologies resolve the real-time-ness issue by lowering the latency for the long-distance network.

5.4.1.3. Intra-DC Network

Cloud DC is a huge resource pool, and the role of intra-DC Network is to establish a virtual network for its user, connect resources for the user flexibly to run the user workloads, and mediate communications with an external system. Since multiple user tenants share the cloud, it is essential to reduce the overhead to control multi-tenancy, to reduce latency, and increase bandwidth.

Workload Profile

For the CPS Area Management Security Use Case, the network traffics relevant to cloud services are summarized below:

- For today's implementation model
    - Ingress traffic to the cloud DC
        - Sensor data flow from the Local Aggregation node (N1) to the Ingestion node (N3 for sensors), totaling 83.75 Tbps
        - User data flow from the Presentation Device node (N11) to the Local Aggregation node (N2 for users), totaling 250 Mbps
    - Egress traffic from the cloud DC
        - Video streaming from the Streaming Hub node (N7), totaling 30 Tbps (assuming Motion JPEG is used)
        - On-demand video replay from the Data Hub node (N5 for sensors), totaling 187.5 Gbps (assuming H.264 is used)
        - Voice messages from the Data Hub node (N6 for users), totaling 800 Mbps
        - SMS messages from the External System node (N12), totaling 1.5 Gbps (at peak time)
- For IOWN GF RIM
    - Ingress traffic to the regional edge cloud DC
        - Sensor data flow from the Local Aggregation node (N1) to the Ingestion node (N3 for sensors), totaling 83.75 Tbps
        - User data flow from the Presentation Device node (N11) to the Local Aggregation node (N2 for users), totaling 250 Mbps
    - Egress traffic from the regional edge cloud DC to the customer premise site
        - Live video streaming from the Streaming Hub node (N7), totaling 30 Tbps (assuming Motion JPEG is used)
        - On-demand video replay from the Data Hub node (N5 for sensors) node, totaling 187.5 Gbps (assuming H.264 is used)
        - Voice messages from the Data Hub node (N6 for users), totaling 800 Mbps (assuming 100 security officers receive the same message)

- Egress traffic from the regional edge cloud DC to the central cloud
    - Replication data flow within the Data Hub node (N5 for sensors)
        - For labeled objects, totaling 4.5 Tbps (resulting from 1,250 monitored areas x 1,000 sensing points per area x 3 KB per object x 10 objects per frame x 15 frames per sec)
        - For sensor data, totaling 837.5 Gbps (assuming one-hundredths of the original data, as only replicated partially based on the event-driven mechanism.)
    - Ad hoc query data flow from the Data Hub node (N6 for users), to know the nearby users, negligible
- Ingress traffic to the central cloud:
    - Replication data flow within the Data Hub node (N5 for sensors)
        - For labeled objects, totaling 4.5 Tbps
        - For sensor data, totaling 837.5 Gbps (assuming one-hundredths of the original data, as only replicated partially based on the event-driven mechanism.)
    - Ad hoc query data flow from the Data Hub node (N6 for users), to know the nearby users, negligible
- Egress traffic from the central cloud to the regional edge cloud
    - Voice messages from the Intelligence Application node (N8) to Data Hub, totaling 8 Mbps (125 concurrent streams x 64 Kbps = 8 Mbps. 100 security officers receive the same message but the source data is the same, thus egress traffic from the central cloud becomes one-hundredth of one from the regional cloud)
- Egress traffic from the central cloud to the customer premise site (via External System)
    - SMS messages from the External System node (N12), totaling 1.5 Gbps (at peak time)

Expected Benefits

In today's implementation model for cloud infrastructure, there are many software-controlled switches/routers to establish virtual private networks for tens of thousands of users and connect hundreds or thousands of resources to build a private environment for the user on top of a large resource pool.

To establish the virtual private network, a sort of tag is assigned to the packet header by the hypervisor or Smart NIC of the host machine running user workloads, and these switches/routers read it to control packet routing and apply throttling in order to handle a huge amount of network traffic in a stable and SLA / SLO compliant manner. Such software-defined network controls increase latency, consume more energy, and become more costly as the size of the resource pool grows.

Typically, the network packets must go through 5 - 10 switches/routers for internal communications, and each time they go through them, a specific network latency, such as 10 - 15 μs, will be added. Thus communications between two resources would be more than 100 μs. For external communications, more switches/routers are needed, and additional controls for private connection management and global IP mapping management; thus, the minimum latency accessing the cloud from the outsider is typically 0.2 - 1 milliseconds.

It should be noted that these delays would be accumulated in the actual workload. For instance, when an application instance tries to update data in a DB instance in the cloud, the DB instance will contact a block storage server. The block storage will contact other block storage servers to replicate data, etc. Therefore, when performing distributed processing with multiple servers in the cloud, effective efficiency is generally lower than on-premises.

In addition, the intra-cloud network service is often oversubscribed, resulting in even more uncertain delays during network congestion. These are also the reasons for the limited use of RDMA in the cloud. Because RDMA is a low-level protocol that requires additional controls to ensure that data is delivered and stored properly at the destination site, it is challenging to apply it to a production workload under such a large and fluctuating latency.

In summary, today's centralized cloud implementation models are less efficient at distributed data processing than on-premises and cause higher system costs and energy consumption when assuming constant workloads.

In the IOWN GF RIM, these issues will be mitigated by Open APN and DCI technologies. Open APN will establish the direct communication path between user instances, which belongs to the virtual private network of the tenant, by wavelength and mode of optical signals. DCI technologies control the establishment of direct communication paths dedicated to the tenant. That means, from a resource pool consisting of hundreds of thousands of servers, DCI picks up a requested number, such as 100 servers for the tenant, and connects them through direct or near-direct communication paths based on available wavelengths, modes, and fiber cores of the network. As a result, it is expected that latency between resources will be reduced, user workloads will be executed more efficiently, and the cost and energy consumption for the inter-cloud network will be reduced. In addition, the IOWN technology-based model opens up the possibility of applying RDMA to a production deployment on top of a larger resource pool.

5.4.2. Application’s Functional Node

As the first step of evaluation of applications' functional nodes, we focused on dominant parts of the data pipeline placed on DCI clusters, that is, Ingestion nodes (N3 and N4), Data Hub nodes (N5 and N6), Streaming Hub nodes (N7), and Intelligence Application nodes (N8).

5.4.2.1. Ingestion Node

The role of the Ingestion node (N3 for sensors) is to continuously receive data (such as videos and LiDAR Point Cloud from the Local Aggregation node (N1)) and convert it to multiple different forms such as chunked data and labeled objects resulting from AI inference, and insert them into the Data Hub node. The role of the Ingestion node (N4 for users) is to collect user data and post them to the Data Hub node.

Workload Profile

The workload of these Ingestion nodes can be roughly divided into two parts, that is, A) data forwarding & conversion and B) AI inference. The following shows the workloads required to support one monitored area. The details of the data pipeline of these Ingestion nodes are shown in C.3.3 and C.3.4 in Annex C.

- For Ingestion nodes (N3 for sensors)
    - A) Data forwarding & conversion
        - Video image data (P2 and P4 in C.3.3)
            - Throughput: 60 Gbps in total, 60 Mbps per camera (assuming Full HD Motion JPEG at 15 fps)
        - Point cloud data (P3 in C.3.3)
            - Throughput: 7 Gbps in total, 7 Mbps per sensor (assuming 100,000 points with G-PCC compression)
    - B) AI inference (P1 in C.3.3)

        - Object recognition with video image data of Full HD
            - Throughput: 15,000 fps in total, 15 fps per camera.
        - Object recognition for point cloud data with 100,000 points
            - Throughput: 20,000 fps in total, 20 fps per LiDAR sensor
        - Fusion of the object recognition results with video image data and point cloud data:
            - Throughput: 15,000 fps in total, 15 fps per sensing point
            - Assuming that there are 1,000 sensing points in a monitored area. Multiple devices may capture a sensing point.

        Note: The frame rate (fps) and the resolution of video images and point cloud data can be lowered based on the event-driven approach.

- For Ingestion nodes (N4 for users)
    - A) Data forwarding & conversion (P1 in C.3.4)
        - Throughput: 0.2 Mbps in total, 1.5 KB/min = 0.2 Kbps per user

Expected Benefits

As discussed in 5.3, with IOWN technologies, which will utilize an RDMA-type network protocol overall photonics network, the workloads of data forwarding can be largely decreased. By referencing several RDMA benchmarks results [RoCE] [Y. Kwak], we could say we can more than halve the CPU utilization or more than double the performance of the processes for A) data forwarding & conversion. In addition, by combining shared memory techniques, the IOWN GF RIM can streamline the data plane and significantly eliminate the overheads caused by additional processes required in today's centralized cloud model, i.e., "Data Reception & Copy" and "Service Mesh" in Figure 5.3-1.

As for B) AI inference, although it highly depends on the situation (i.e., what is captured by sensors), leveraging an event-driven approach in Annex D, an IOWN GF RIM will significantly reduce the workloads of the AI inference. The disaggregated computing feature of the DCI architecture can adjust the number of CPU and accelerator resources assigned for the LSNs appropriately to match the dynamically changing workloads.

These effects of the IOWN GF RIM will improve the system cost, energy consumption, and latency of the Ingestion node. With the progress of the hardware technologies related to IOWN technologies, such as the use of more advanced accelerator chains and the extension of the all-photonic network to the inside of DCI clusters, these effects will be further improved.

5.4.2.2. Data Hub Node

The roles of the Data Hub are to retain continuously ingested data for a certain period of time and pass it on to subsequent applications such as the Intelligence Application node (N8). For example, the Ingestion node (N5) retains video stream data, LiDAR point cloud data, and labeled objects resulting from image recognition analysis, and the Data Hub node (N6) retains user data such as position.

To prevent data loss and guarantee high availability, data needs to be replicated to multiple physically isolated server nodes which make up the Data Hub node. This causes a certain time lag before the data becomes available in subsequent applications.

Depending upon the subsequent application profile, data stored in the Data Hub nodes will be read out sequentially or conditionally. For instance, in the AM Security UC, at least two types of applications will retrieve data from the Data

Hub node (N5). One is an application that continuously retrieves data from the Data Hub node (N5) and reflects it in the Live 4D Map, which tracks changes in real-world situations and helps detect anomalies quickly. The other is an application that reads data from the Data Hub node (N5) conditionally to reanalyze relevant raw data in more detail. Such a process is triggered when a risk is detected in the first rough analysis, but it is not deterministic.

Workload Profile

In the AM Security UC, data flows into the Data Hub nodes are summarized below:

- For the Data Hub node (N5 for sensors)
  - Video streams
    - Write throughput: 75 Tbps in total, 60 Mbps per camera (assuming raw data is 15 fps Motion JPEG)
    - Unit of ingestion: 0.5 MB Full HD JPEG file (assuming image files included in the Motion JPEG file are extracted and ingested to the Data Hub node (N5) separately.)
    - # of records ingested per sec: 18.75 million/sec
    - Read throughput: 1/100 of the write (assuming data is retrieved as needed based on the event-driven mechanism. In today’s implementation model, it should be noted that the Data Hub node is also used as local storage for the Ingestion node, and the same amount of data flow as the write will be added.
  - LiDAR data
    - Write throughput: 8.75 Tbps in total, 7 Mbps per sensor (assuming raw data is G-PCC)
    - Unit of ingestion: 0.044 MB scan data (assuming scan records included in the 20 Hz LiDAR raw data are extracted and ingested to the Data Hub node (N5) separately.)
    - # of records ingested per sec: 25 million/sec
    - Read throughput: 1/100 of the write (assuming data is retrieved as needed based on the event-driven mechanism. In today’s implementation model, it should be noted that the Data Hub node is also used as local storage for the Ingestion node, and the same amount of data flow as the write will be added.
  - Labeled objects
    - Write throughput: 4.5 Tbps in total, 3.6 Mbps per sensing point (a sensing point consists of one video camera and one LiDAR sensor)
    - Unit of ingestion: 3 KB (corresponding to the size of each labeled object.)
    - # of records ingested per sec: 187.5 million/sec
    - Read throughput: several times, or several tens of times the write
- To the Data Hub node (N6 for users)
  - User data
    - Write throughput: 250 Mbps in total, 1.5 KB/min = 0.2 Kbps per user

    - Unit of ingestion: 1.5 KB (corresponding to the size of single-user data.)
    - # of records ingested per sec: 20.8 thousand/sec
    - Read throughput: Ad hoc (assuming data is retrieved as needed when the incident is detected.)
  - Voice message
    - Write throughput: 8 Mbps in total, 6.4 Kbps per monitored area (assuming there is a total of 125 concurrent messages simultaneously and 0.1 concurrent messages per monitored area.)
    - Unit of ingestion: 160 Byte (assuming voice stream data is created in units of 20 milliseconds)
    - # of records ingested per sec: 6,250 per sec
    - Read throughput: 100 times the write (assuming hundred officers receive the same message)

To guarantee data persistence, data ingested to the Data Hub nodes are replicated locally in a synchronous manner and when specified based on data type or triggered by the event-driven mechanism, may also be replicated remotely in an asynchronous manner to utilize global resources to process data, as described in section 5.3.

Expected Benefits

In today’s implementation model, Kafka and/or a proprietary Cloud solution is used to build the Data Hub services. In these implementation models, data is replicated across multiple servers that are located in different data centers (availability zones) to guarantee data persistence and high availability. It is true that replicating data across multiple server nodes and writing it to the storage is costly. The latency from data ingestion to data retrieval at Data Hub is not very low, typically around 50 - 100 milliseconds in today’s Cloud environment.

Also, to realize real-time processing, it is desirable to push data to the subsequent applications at the timing of data insertion, but this is not well supported very well by many of today’s implementation models. This is because the workload of sending push notifications varies greatly depending on the number of subscribing clients. However, the resources allocation is fixed at each configuration unit, namely shard or partition. Therefore, the subsequent applications need to pull the data in a micro-batch manner, increasing the end-to-end latency.

Furthermore, in today’s message broker, each stream is configured to have multiple shards or partitions based on the throughput requirements. In this configuration, each shard/partition is designed to achieve a specific throughput such as 1MB/s. Thus various sensor data will be mixed in one shard/partition, or single sensor data will be spitted into multiple shards/partitions. Therefore, ideally, it is desired to extract data conditionally such as per sensor, etc., to streamline subsequent analysis. However, in reality, such conditional queries are not supported. Thus, the subsequent applications have to retrieve a certain amount of data in bulk, and sort, filter, and preprocess data locally. This is also due to the fact that the workload to create indexes and run queries fluctuates greatly, and today’s models based on fixed resource allocation cannot handle it well.

Lastly, the requirements for message brokers vary by scenario, such as different message sizes, different write-read ratios, different data persistence levels, etc. Still, with today’s implementation model, it is required to use the uniform configuration. This is because resource allocation to each system is inflexible, and design and operating costs will be too high to use individually tuned, multiple systems. Therefore, it forces users to replicate small messages with low persistence requirements to remote servers through a long-distance network.

The IOWN technologies will solve these issues by increasing the network performance and flexibility of system configuration. The Open APN technologies minimize delays for data replication. The IOWN GF DCI technologies add more flexibility in allocating resources to the message broker system, allowing us to choose memory or flash storage to store data and configure the system to push notifications and support conditional queries.

5.4.2.3. Streaming Hub Node

The Streaming Hub is a variant of the Data Hub, which delivers the video stream data to many viewers with minimum delay. To minimize delay, the video stream is divided into smaller chunks and continuously written to Streaming Hub and read from there. The ordered message delivery is also required to eliminate the need to sort messages on the client-side for video display. To minimize the delay to monitor the area, Motion JPEG is assumed to deliver the video stream. However, depending on the latency requirement, the number of jpeg files included in a single block will be configured. In this paper, we assume that each Motion JPEG file contains 5 JPEG files that correspond to a one-thirds second long video.

Workload Profile

In the AM Security UC, the amount of data flowing through the Streaming Hub node is summarized below:

- Writing data flow
  - Write throughput: 1.5 Tbps (assuming 1,250 monitored areas, 20 cameras per monitored area, and 60 Mbps per video stream)
  - Unit of ingestion: 2.5 MB (assuming five JPEG files with a size of 0.5 MB are included in each ingestion unit)
  - # of records ingested per sec: 75,000 per sec
- Reading data flow
  - Read throughput: 30 Tbps (assuming 1,250 monitored areas, 20 cameras per monitored area, 20 display systems per camera, and 60 Mbps per video stream)

Expected Benefits

In today’s implementation model of the Streaming Hub node, the data chunks need to be replicated to multiple servers to obtain the required delivery throughput. In the cloud, this type of data replication is based on TCP communications, and consumes a lot of CPU resources, as it increases latency just to replicate data. In addition, such a continuous large stream of communications is easily subject to throttling by the cloud control plane mechanism, which causes additional delays. For large broadcast live stream services, such a delay will typically be seconds, and in some cases, tens of seconds.

The application of IOWN technologies improves that situation. It will replicate data directly to memory on the other Streaming Hub nodes through RDMA over Open APN. This results in lower latency for live stream delivery services and lower CPU usage, which means lower costs and energy consumption, to run the services.

5.4.2.4. Intelligence Application Node

The roles of the Intelligence Application node (N8) are to read labeled object data generated by primary inference analysis from the Data Hub node (N5), record it on a Live 4D Map database to track changes in the situation, detect anomalies immediately, and send voice or SMS messages to notify the security officers.

Workload Profile

The characteristics of these Live 4D Maps, anomaly detection, and voice message generation, are described in more detail below.

- Live 4D Map
    - Each labeled object represents a recognized object, such as a person, belongings, etc.
    - These recognition results are recorded on the static structural information of the monitored area with the time stamp.
    - To analyze the spatial relationships among recognized objects, spatial indexes are generated and managed for each data record.
    - To quickly grasp the conceptual relationship between recognized objects, link information representing the relationship is also generated and appended to each data record at the same time.
    - Data can be retrieved by designating various spatial structures, relationships, and time series-based conditions.
    - The workload for reading data on the Live 4D Map can be tens, hundreds, or even thousands of times higher than one for writing data. This is because the data is retrieved under various conditions and involves complex operations on the relationships between the data.
- Anomaly detection
    - Analyze Live 4D Map data continuously in a micro-batch manner and ad hoc in an event-driven manner, e.g., when predefined labeled data, such as knife, is registered, etc.
    - Analyze the position and behavior of each object and the relationship between objects over time.
    - Anomaly detection is basically based on pattern analysis. When the Live 4D Map data matches the registered anomalous pattern, or when it deviates significantly from the normal pattern, it is recognized as an anomalous pattern.
      Note: For the machine learning-based analysis, a model based on the recurrent neural network (RNN) and/or the graph neural network (GNN) will be used against linked geospatial and time-series data.
- Voice message generation
    - If an anomaly is detected, a natural language text is first generated to explain it.
    - A natural language text is determined on top of relevant linked data. Related object data and relationship link information are passed to the model that represents a natural language structure.
    - Once the text is confirmed, it will be converted to voice data by voice reading processing.

Expected Benefits

The expected benefits of these Live 4D Maps, anomaly detection and voice message generation, are described in more detail below.

- For Live 4D Map
  In today’s implementation model, it is required to combine multiple IDH services to build the Live 4D Map. For example, JSON Document DB is used to manage labeled objects, Spatial DB is used to manage the location of objects, and Graph Store is used to manage the interrelationships between Objects. Then, in order to perform the processing described above, the application needs to access multiple DBs and combine the data on the application side. In such a configuration, data analysis can be initiated after all the data has been collected. In particular, if it is required to obtain data from another DB to narrow down the data extracted from one DB, the data processing time will increase significantly.

In the IOWN GF RIM, such complexity is significantly reduced by the IDH Converged DB service. It becomes possible to manage various types and structures of data in one place by combining various resources via a high-speed network flexibly. This is difficult to achieve with scalability based on today's technologies because different data types and structures have different requirements such as index generation, query processing, storage access I/O pattern, etc.

- For anomaly detection
  When performing the pattern analysis to detect anomalies against a large amount of linked data, it is often necessary to manage data in a distributed manner on multiple nodes and execute analysis in parallel since the amount of data and the amount of calculation are too large. In such distributed data processing, communication between nodes becomes chatty, and the network becomes a bottleneck. In particular, in today's cloud environment, network packets are exchanged via multiple switches, which increases network delay and reduces both the response time and throughput of such pattern analysis processing.

  Such distributed data analysis processing can be accelerated by IOWN Technologies, too. RDMA over Open APN will shorten the latency for inter-node communications, and data processing speed and throughput will be greatly improved, and energy consumption will be reduced.

- For voice message generation
  Regarding voice message generation, the operation itself can be completed by a single node. However, to automatically generate a voice that explains the situation continuously, it is necessary to constantly send input data to that node. However, in today's cloud-based implementation model, sudden network congestion increases the jitter of the inter-node communications, which hinders the realization of stable real-time services.

  In the IOWN GF RIM, such communications will be more efficient and stable because of RDMA over Open APN, which interconnects the anomaly detection node and the voice message generation node.

6. Conclusion

We developed the initial Reference Implementation Model (RIM) for the given benchmark model to the Area Management Security Use Case (AM Security UC) (Guarding Services) leveraging the Open APN technologies and IOWN GF DCI technologies.

To quickly recap some of the findings:

Using Open APN technologies, this RIM can deliver:

- High bandwidth and reduced power consumption across core and access networks to collect images and data from massive surveillance cameras and LiDAR sensors.
- Real-time monitoring services at low latency by reducing optical-to-electrical conversion and establishing a direct APN path between the customer premise and the telco edge/central clouds.

Using IOWN GF DCI technologies, this RIM can deliver:

- Flexible resource management from heterogeneous and disaggregated device resources pool that can logically compose application’s functional node, Network service workload, and Data Hub workload at the desired location to perform high data performance in hardware rate by Function Dedicated Network interface card such as Smart NIC, DPU, and IPU, in addition to CPU, GPU, and Persistent memory.
- RDMA capable network across multiple sites building Geographically Distributed Data-Pipeline through customer premise edge, telco edge, and telco central cloud. RDMA makes data transfers more efficient and enables fast data movement between servers and storage without involving its CPU. Throughput is increased, latency reduced, CPU power consumption reduced, and CPU resource is freed up for the application’s functional node.

In this RIM, we focus our study on the real-time case, where IOWN technology can be used more effectively. Analyzing and utilizing the past data is in the scope of our future work.

Globally managing resources to enable end-to-end network service quality guarantees is the key enabler to dynamically and cost-effectively configure the optical transmission lines and conversion elements that underlying networks, such as an Open APN network [IOWN GF Open APN], will offer. We will need further study through PoC.

Many countries are now under-developing privacy regulations in the AM Security UC. Solution for privacy management technology is an important study item and our future work in the IOWN Global Forum.

Based on this RIM, we are planning to evaluate the feasibility of leveraging IOWN technologies through PoC.

Abbreviations

A

AIC: AI-Integrated Communication

AM: Area Management

AM Security UC: Area Management Security Use Case

B

BM: Benchmark Model

C

CNN: Convolutional Neural Network

CPS: Cyber-Physical System

D

DCI: Data-Centric Infrastructure

DMA: Direct Memory Access

DPA: Data Plane Acceleration

DPD: Data Pipeline Diagram

DPU: Data Processing Unit

F

FDN: Function Dedicated Network

FlexBr: Flexible bridging Service

FPGA: Field Programmable Gate Array

FPS: Frame Per Second

G

GNN**:** Graph Neural Network

GPU: Graphics Processing Unit

G-PCC: Geometry based Point Cloud Compression

gRPC: gRPC Remote Procedure Calls[*]

[*] https://grpc.io/docs/what-is-grpc/faq/#what-does-grpc-stand-for

I

I/O: Input/Output

IOWN: Innovative Optical and Wireless Network

IOWN GF: IOWN Global Forum

IPU: Infrastructure Processing Unit

J

JPEG: Joint Photographic Experts Group

L

LiDAR: Light Detection and Ranging

LSN: Logical Service Node

N

NIC: Network Interface Card

O

Open APN: Open All-Photonic Network

R

RAW: Raw image format

RDMA: Remote Direct Memory Access

RIM: Reference Implementation Model

RNN**:** Recurrent Neural Network

S

SRTP: Secure Real-time Transport Protocol

SQuaRE: Systems and Software Quality Requirements and Evaluation

T

TLS: Transport Layer Security

U

UDS: Unix Domain Socket

V

VM: Virtual Machine

V-PCC: Video Point Cloud Compression

References

[IOWN GF AIC UC] IOWN Global Forum, “AI-Integrated Communications Use Case Release-1,” 2021. https://iowngf.org/use-cases/

[IOWN GF CPS UC] IOWN Global Forum, “Cyber-Physical System Use Case Release-1,” 2021 https://iowngf.org/use-cases/

[IOWN GF DCI] IOWN Global Forum, “Data-Centric Infrastructure Functional Architecture,” 2022.

[IOWN GF Open APN] IOWN Global Forum, “Open All-Photonic Network Functional Architecture,” 2022.

[DPU] Data Processing Units (DPUs), https://www.nvidia.com/en-us/networking/products/data-processing-unit/

[IPU] Infrastructure Processing Units (IPUs), https://www.intel.com/content/www/us/en/products/network-io/smartnic.html

[ISO/IEC 25000] ISO/IEC 25000:2014 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE, https://www.iso.org/standard/64764.html

[Cloud DC] Amazon Web Services, “Behind the Scenes: Exploring the AWS Global Network (NET305) - AWS re:Invent 2018,” 2018. https://www.slideshare.net/AmazonWebServices/behind-the-scenes-exploring-the-aws-global-network-net305-aws-reinvent-2018

[A. Singh] A. Singh, et al., ”Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network,” SIGCOMM ’15 August 17-21, 2015. https://static.googleusercontent.com/media/research.google.com/ja//pubs/archive/43837.pdf

[M. Qasaimeh] M. Qasaimeh, K. Denolf, J. Lo, K. Vissers, J. Zambreno and P. H. Jones, "Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels," *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, 2019, pp. 1-8, DOI: 10.1109/ICESS.2019.8782524.

[K. Matsubara] K. Matsubara, et al., "4.2 A 12nm Autonomous-Driving Processor with 60.4TOPS, 13.8TOPS/W CNN Executed by Task-Separated ASIL D Control," *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 56-58, DOI: 10.1109/ISSCC42613.2021.9365745.

[Silicon Photonics] Ligtmatter, https://lightmatter.co/products/envise/

[IOWN Data Hub] IOWN Global Forum, “Data Hub Functional Architecture,” 2022.

[RoCE] InfiniBand Trade Association, “RoCE Accelerates Data Center Performance, Cost Efficiency, and Scalability,” 2017. https://www.roceinitiative.org/wp-content/uploads/2017/01/RoCE-Accelerates-DC-performance_Final.pdf

[Y. Kwak] Y. Kwak and J. Jeong, “Performance Evaluation of RDMA Transfer Method for Security Added on Edge Computing,” International Journal of Future Computer and Communication, Vol. 8, No. 3, September 2019. http://www.ijfcc.org/vol8/544-IAS19-658.pdf

[DFD] Lucid Software Inc., https://www.lucidchart.com/pages/data-flow-diagram

[D. Graziosi 1] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC),” APSIPA Transactions on Signal and Information Processing, vol. 9, 2020.

[D. Graziosi 2] D. Graziosi, O. Nakagami, K. Mammou, M. Preda, “Point Cloud Compression in MPEG,” ICIP 2020.

[YOLO] https://pjreddie.com/darknet/yolo/

[Y. Guo] Y. Guo, et al., "Deep Learning for 3D Point Clouds: A Survey" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 43, no. 12, pp. 4338-4364, 2021.

[J. Ngiam] J. Ngiam et al., “StarNet: Targeted Computation for Object Detection in Point Clouds,” Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019).

[IOWN GF Vision] IOWN Global Forum, “Innovative Optical and Wireless Network Global Forum Vision 2030 and Technical Directions,” 2020. https://iowngf.org/white-papers/

[Vision Sensor] SONY, “The world's first Intelligent Vision Sensor with edge processing.” https://developer.sony.com/develop/imx500/

[JPN48] Technical Committee on Photonic Network, “Japan Photonic Network Model“. https://www.ieice.org/cs/pn/eng/jpnm_en.html

Annex A. Development Method of Benchmark Model

This Annex describes the definition of the Benchmark Model in IOWN GF and how to develop the Benchmark Model for target use cases used in Section 2.

A.1. What is the Benchmark Model in IOWN GF?

A "benchmark" is generally defined as an indicator for measuring computer hardware and software's performance and operating speed. A "model" means a norm that should be the basis for judgment, evaluation, and action.

A Benchmark Model in IOWN GF defines Reference Case, Metrics, and Evaluation Methods. A Benchmark Model is also developed for a target use case.

Reference Cases dig deeper into target use cases and clearly define the conditions for determining functional and non-functional requirements, output/input data flow, system size, and parameters. In some cases, they also describe features and implementation procedures.

Metrics are the numbers to evaluate the performance and quality of the implementation model. They are assigned to essential requirements when providing services.

Evaluation Methods are ways to evaluate the implementation model. When a new architecture or technology is incorporated into a system, the evaluation uses the same evaluation methods and metrics as before and compares the evaluation results.

A.2. How to Develop the Benchmark Model for the Target Use Case

The Benchmark Model for a target use case will be developed by selecting one use case from the AIC/CPS use cases proposed by the IOWN GF.

First, for the selected use case, detailed functional, non-functional, and performance requirements will be defined, including key requirements in the use case document to define the Reference Case.

Next, capture the characteristics of the selected use case and define the Metrics that will be used as a basis for the Evaluation Method of the implementation model. This is the end of the development of the Benchmark Model.

Use case requirements change over time. In addition, various technologies and products will evolve, including IOWN GF architecture and technology. Therefore, the Benchmark Model needs to be rebuilt and evaluated to suit the changes.

A.3. How to Develop the Evaluation Method and Metrics for the Target Use Case

Determining the correct Evaluation Method is essential for making the right decision since there will be multiple technical options to design the system that satisfies the Reference Case's mandatory requirements. The Evaluation Method is also necessary to demonstrate how advanced the technology developed by IOWN GF is compared to the current technology.

A.3.1. What to Evaluate

According to the ISO/IEC 25000 [ISO/IEC 25000] or “Systems and Software Quality Requirements and Evaluation” (SQuaRE), the quality of the system can be assessed from the perspectives of, Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability. Since the current situation is in the early stages of technological development, we will focus on a limited set of Metrics, i.e., 1) response time, 2) system cost, and 3) energy efficiency, assuming that other mandatory requirements, such as the Minimum Required Response Time, Functional Suitability and Security, are met.

A.3.2. Evaluation Strategy

As discussed above, multiple metrics can be used to evaluate the system, so the question is how to assess the system's quality as a whole. In IOWN GF, we decided to take the following approach:

- Designing the implementation model so that it satisfies all mandatory requirements, such as:
  - **Functionalities**, as described in the Reference Case
    The functionalities here include data aggregation, collection, recognition, decision making, notification delivery, etc., which are all required to build the system which supports the Reference Case.
  - **End-to-end response time:**
    The End-to-end response time means the total amount of time from the actual occurrence of the target event in the real world to the execution of necessary action by the system.
- Then, analyzing following Metrics of the designed implementation model, with equal weighting:
  - **End-to-end response time**:
    The End-to-end Response time is assumed to meet the number specified by the Reference Case as a mandatory requirement as described above, but on top of that, we think the shorter, the better technology; thus, we will evaluate it, too.
  - **System resources**
    The system resources are defined as metrics required to calculate the hardware and software costs, e.g., the number of CPU cores, GPU cards, and switches. When designing the implementation model, the system resources are determined by considering scale Metrics defined in the Reference Case, e.g., the number of sensors, etc.
  - **System cost**
    The system includes hardware and software costs, power costs, location/facility costs, and labor costs required for the system operation. In the case of today’s Cloud solutions, a subscription fee that includes most of them is charged. In the case of today’s on-premise model, the labor cost is differs based on the quality of the system design. We have to consider all of these and conduct analysis in detail to estimate the cost; however, for simplicity, we make the following assumptions in this technical paper:
    - Hardware used for the system sizing estimate is standardized, i.e., we will define a set of the standard shapes as described in A.3.3
    - The hardware depreciation period is set to 5 years to calculate the hardware cost
    - Software annual cost (for depreciation and supports) is twice that of the underlying hardware
    - Every 100 physical or virtual server instances will incur a monthly labor cost of $10,000

  - These numbers are defined to be well-aligned with cloud costs. For details, please see A.3.3

- **Energy consumption**
  Energy consumption is the amount of energy consumed by the system or the system element.
  It is noted that energy consumption is a part of the system cost, but it has independent importance as the sustainability debate progresses. Therefore, we will analyze it in parallel with the system resources analysis.

A.3.3. Structure of Evaluation

The implementation model consists of various data processes and data flows, as described in the next chapter for the Reference Implementation Model. Therefore, it is not possible to immediately analyze the Metrics of the entire system. IOWN Global Forum has decided to take a solid approach to this challenge. It means the entire system must be broken down into the elements, i.e., the system nodes and the networks that connect the system nodes. Each metric is analyzed for each element and summed up to evaluate the entire system as a whole.

Figure A.3-1 and Table A.3-1 show such an evaluation framework to assemble various Metrics.

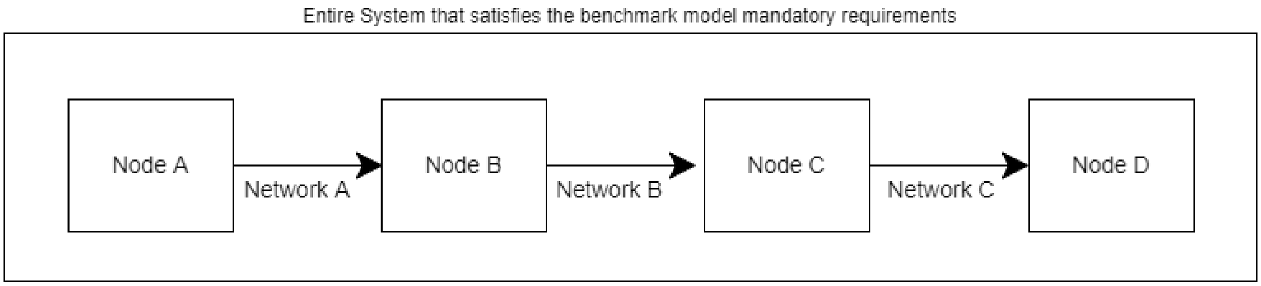

Entire System that satisfies the benchmark model mandatory requirements
Node A
Network A
Node B
Network B
Node C
Network C
Node D

Figure A.3-1: Implementation Model Breakdown Image

Table A.3-1: An Example of Implementation Model Evaluation

	RESPONSE TIME	SYSTEM RESOURCES	SYSTEM COST	ENERGY CONSUMPTION
Node A	20 milliseconds	CPU Core x 36 RAM 512 GB, etc.	$ 500	1,000 W
Network A	Ten milliseconds	Switch x 4, Etc.	$ 100	50 W
Node B	15 milliseconds	CPU Core x 6 GPU Card x 5 RAM 256 GB, etc.	$ 1,000	3,000 W
Network B	Five milliseconds	Switch x 2, Etc.	$ 50	25 W
Node C	Five milliseconds	CPU Core x 24 RAM 1024 GB, etc.	$ 400	500 W
Network C	Eight milliseconds	Switch x 4, Etc.	$ 50	25 W

Node D	Two milliseconds	CPU Core x 48 RAM 512 GB, etc.	$ 400	1,200 W
Entire System	70 milliseconds	-	$ 2,500	5,800 W

A.3.4. Comparative Evaluation

In the process of technological development, we will find multiple alternatives in determining the implementation model. Also, the current implementation model needs to be compared against them to clarify the benefits of the new solution using IOWN technologies. It means that we have to evaluate each option one-by-one based on the above framework, and at the end, we will determine the best implementation model.

Table A.3-2 below shows such a comparison.

Table A.3-2: An Example of Implementation Model Comparison

	RESPONSE TIME	SYSTEM COST	ENERGY CONSUMPTION	COMMENT
Current Implementation Model	150 msec	$ 10,000	25,000 W	Mandatory requirement of the response time (100 msec) is not met
Implementation Model - Option A	70 msec	$ 2,500	5,800 W	
Implementation Model - Option B	60 msec	$ 3,000	10,000 W	
Implementation Model - Option C	75 msec	$ 2,200	4,000 W	The best option, as achieving minimum cost and energy consumption while satisfying the mandatory requirements (response time is less than 100msec)

Annex B. Dataflow and Workload Profiling Framework

This Annex describes a dataflow and workload profiling framework used in section 3. IOWN GF developed this framework to identify service gaps/requirements of use cases accurately and efficiently.

B.1. Framework Overview

This framework consists of the following two steps:

- Step 1: Develop a data pipeline diagram that depicts the use case with the end-to-end flow of data and processes. A data pipeline diagram comprises functional nodes, processes, dataflows, and database/storage elements as illustrated in B.2.

- Step 2: Develop a profile for each of the functional nodes, processes, and dataflows of the developed data pipeline diagram.

B.2. Data Pipeline Diagram

Data Pipeline Diagram (DPD) is an extended form of Data-Flow Diagram (DFD) [DFD]. We have developed DPD with some customizations on DFD to profile dataflows and workload for end-to-end systems that span across data centers, networks, and customer premises. We have introduced the concept of a functional node to represent a group of processes as one node and visualize an end-to-end system in a simpler way. Besides simplification, clarifying how many functional nodes should be accommodated and how distributed they are will help us identify networking challenges for distributed systems.

B.2.1. Elements

A data pipeline diagram is composed of the following elements:

- **Functional Node**: A logical node that executes a set of processes. It is a logical node because multiple functional nodes may be deployed to the same physical node. As to the place of physical deployment, there may be multiple options. For example, some functional nodes may be deployed to either customer premises or edge data centers.

- **Dataflow:** Flow of data transferred from one element to another.

  Note: This DPD does not distinguish between “pull” type dataflows and “push” type dataflows.

- **Process**: A set of autonomous operations including getting input data, generating output data, and sending out output data. We describe a process with a set of micro-processes to clarify its scope. We also describe conditions that may affect the behaviors of outgoing dataflows from the process (e.g., conditional branch, relocation of destinations, etc.).

- **Database/Storage**: Database or storage. This element is typically used for realizing asynchronous interaction among processes or consolidating multiple dataflows.

  Note: This data pipeline diagram does not distinguish types of databases/storages such as RDBMS, object storage, and so on.

B.2.2. Legends

As illustrated in Figure B.2-1, elements should be represented with the following notation rules:

- A functional node should be drawn with a rectangle showing its identifier and/or its name in the middle.
  Note: You can introduce multiple instances derived from one functional node so that you can describe the specific characteristics of the functional node. The instances, when shown, should be labeled at the top of the rectangle with <>. Examples for the labels are <User 1> and <User 2>, which stand for the personas associated with the instances.
- A dataflow should be drawn with a one-directional arrow showing the direction of flow. Each arrow should be labeled with an identifier for reference.
  Note 1: You may assign different identifiers to the start points and endpoints of an arrow if it is necessary to distinguish between them.
  Note 2: A functional node may include one or more processes. If you want to clearly show the relation between a dataflow and one of the processes, you can connect the arrow of the dataflow from/to the process. Otherwise, for simplicity, you may connect it just from/to its functional node.
- A process should be drawn with a rounded rectangle (or oval) showing its identifier and/or name in the middle.
- A database/storage should be drawn with a drum showing its identifier and/or name in the middle.

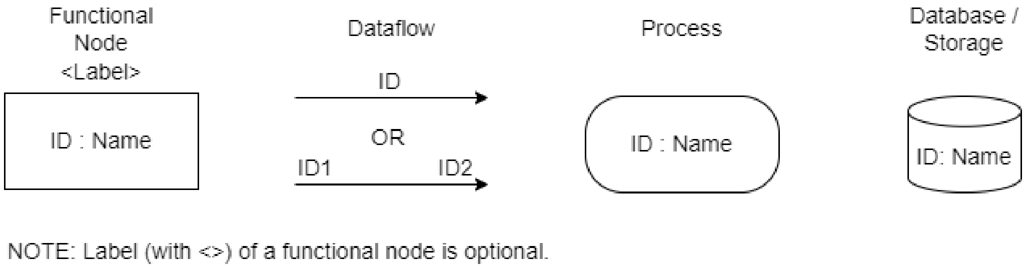


Figure B.2-1: Elements and Legends of Data Pipeline Diagram

B.2.3. Diagram Example

Shown below is an example of a data pipeline diagram. This example shows an e-mail service system supporting very large attachments.

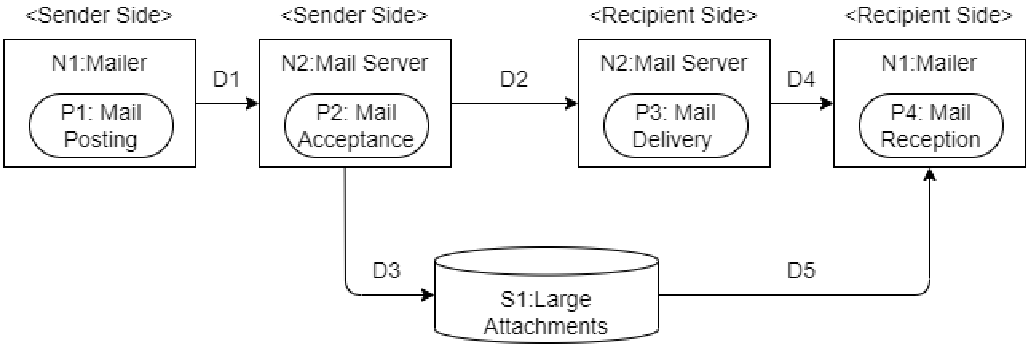


Figure B.2-2: An Example of Data Pipeline Diagram (Big Mail System)

B.3. Profiling

B.3.1. Profiling Functional Nodes

The following attributes should be clarified for each functional node. Table B.3-1 is an example of functional node profiles.

- Fixed / Mobile / Semi-Fixed
- Typical places of deployment, e.g., customer premises, regional edge data centers, and centralized cloud
- Total number of nodes

Table B.3-1: Example of Functional Node Profiles

ID	NAME	DESCRIPTION	ATTRIBUTES
N1	Mailer	Sending/Receiving e-mails	• Fixed/Mobile • Place: Customer Premises • #: billions
N2	Mail Server	Accepting e-mails and delivering them to their recipients.	• Fixed/Semi-Fixed • Place: Customer Premises/ Carrier's Local Premises • #: FFS

B.3.2. Profiling Processes

The following attributes should be clarified for each process. Table B.3-2 shown below is an example of process profiles.

- The node where the process is implemented
- Micro-processes
- The volume of computations, e.g., OPS (Operations Per Second)
- Energy consumption
- Inherent latency

Table B.3-2: Example of Process Profiles

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Mail Posting	• Generate MIME Data • Send	• OPS: FFS • Power Consumption: FFS • Inherent Latency: FFS

P2	Mail Acceptance	• Extract MIME parts • Detach large attachments and store them into the database/storage named “Large Attachments” if their total size exceeds 10 MB (dataflow D3) • Regenerate MIME data, adding attachments containing the URLs of the original attachments • Send the regenerated MIME data to the recipient-side mail server	• OPS: FFS • Power Consumption: FFS • Inherent Latency: FFS
P3	Mail Delivery	• Receive the MIME data from the sender-side mail server and deliver it to the recipient-side Mailer	• OPS: FFS • Power Consumption: FFS • Inherent Latency: FFS
P4	Mail Reception	• Receive the MIME data • Extract MIME parts • Fetch the original attachment from the large object storage	• OPS: FFS • Power Consumption: FFS • Inherent Latency: FFS

B.3.3 Profiling Dataflows

The following attributes should be clarified for each Dataflow. Table B.3-3 shown below is an example of dataflow profiles.

- Data size
- Occurrence rate
- Other requirements: security measures, QoS requirements, etc.

Table B.3-3: Example of Dataflow Profiles

ID	DESCRIPTION	ATTRIBUTES
D1	E-mails sent to sender-side Mail Server	• Occurrence Rate: varying, up to 10 e-mails per minute • Data Size: up to 10 GB • Other Requirements: mutual authentication, encryption
D2	E-mails sent to recipient-side Mail Server	• Occurrence Rate: varying, around 10,000 e-mails per minute • Data Size: up to 10 MB • Other Requirements: encryption, anti domain spoofing
D3	Detached large attachments	• Occurrence Rate: varying, around 50,000 puts per minute • Data Size: up to 10 GB • Other Requirements: mutual authentication, encryption
D4	E-mails sent to recipient-side Mailer	• Occurrence Rate: varying, up to 10 e-mails per minute • Data Size: up to 10 MB • Other Requirements: mutual authentication, encryption
D5	Large attachments fetched by recipient-side Mailer	• Occurrence Rate: varying, up to 50 gets per minute • Data Size: up to 10 GB • Other Requirements: mutual authentication, encryption

Annex C. Detailed DPD for the Area Management Security Use Case

This annex shows the details of the DPD (Data Pipeline Diagram) for the Area Management Security Use Case (AM Security UC).

C.1. Data Pipeline Diagram

Figure C.1-1 is a DPD used for processing and dataflow analysis on the AM Security UC.

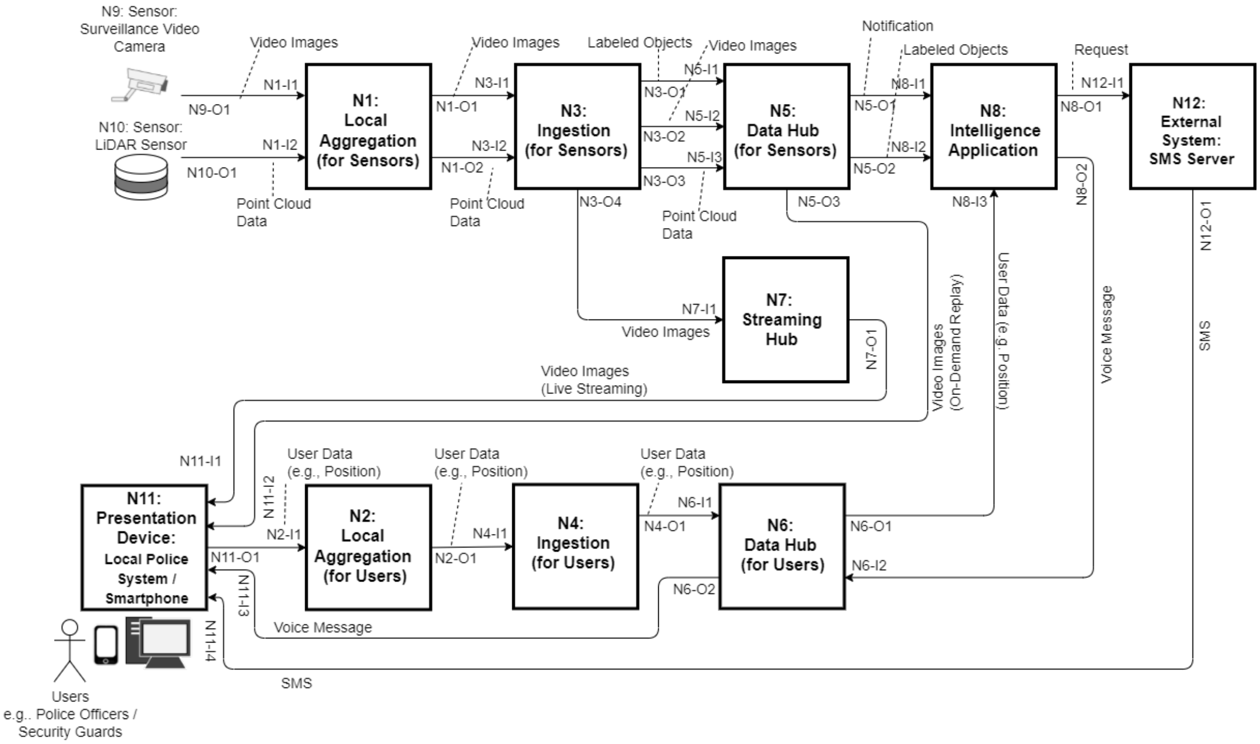


Figure C.1-1: A Data Pipeline Diagram for the AM Security UC (Reprint of Figure 3.1-1)

C.2. Functional Node Profiles

The profiles of the functional nodes described in Figure C.1-1 are shown in Table C.2-1.

Table C.2-1: Functional Node Profiles

ID	NAME	DESCRIPTION	ATTRIBUTES
N1, N2	Local Aggregation	Nodes that collect data from devices and send data to the Ingestion node. They may also provide functions for efficient data collection, such as consolidating and/or reducing data from multiple devices. N1 collects sensor data from Sensor nodes such as surveillance video cameras (N9) and LiDAR sensors (N10), while N2 collects user data from Presentation Device nodes (N11).	• Fixed (for sensors), Fixed/Mobile (for users) • Place and #: an example is shown in Annex E.
N3, N4	Ingestion	Nodes that accept data from Local Aggregation nodes and may update the database/storage in the Data Hub node and/or Streaming Hub node. Ingestion nodes may also provide efficient data collection and usage functions, such as data format conversion, indexing, and cognitive functions, such as image recognition and metadata creation. N3 treats sensor data, while N4 treats user data.	• Fixed • Place and #: an example is shown in Annex E.
N5, N6	Data Hub	Nodes with a database/storage that collectively preserve data and provide these data for subsequent, possibly repeated data usages. To handle massive data usage, a Data Hub node may store data replications in a distributed manner and streamline access/query by managing and utilizing data indices. In addition, the Data Hub node supports distributing notifications to listening nodes, including Intelligence Application nodes (N8). N5 is a Data Hub node for sensor data, while N6 is a Data Hub node for user data.	• Fixed • Place and #: an example is shown in Annex E.
N7	Streaming Hub	A node that receives one or multiple video image streams and relays them to Presentation Device nodes (N11).	• Fixed • Place and #: an example is shown in Annex E.
N8	Intelligence Application	A node that provides application services such as analysis/optimization, alerts, and data exposure to External System nodes (N12) by utilizing data received from the Data Hub nodes (N5, N6).	• Fixed • Place and #: an example is shown in Annex E.
N9, N10	Sensor	Nodes that output sensed data. N9 represents surveillance video cameras which provide video image streams, and N10 represents LiDAR sensors which provide point cloud data.	• Fixed • Place and #: an example is shown in Annex E.
N11	Presentation Device	A node that receives video image streams and alert messages and renders them for presentation. It also supports some functionality for uploading user data. Typically, it is the output function of a user's communication device, such as a smartphone or a PC for a local police system.	• Fixed • Place and #: an example is shown in Annex E.

N12	External System	A 3rd party's system that consumes output data of Intelligence Application nodes (N8). In Figure C.1-1, an SMS server is assumed as an External System node.	• Fixed • Place and #: an example is shown in Annex E.

C.3. Process and Dataflow Profiles of Each Functional Node

The following subsections describe the internal processes of the functional nodes in Figure C.1-1, the dataflows between them, and their profiles.

An appropriate communication scheme and compression scheme should be selected for each communication section according to the dataflow requirements. This selection will be discussed in section 5. Examples of communication schemes and compression schemes are as follows:

- Communication scheme: Shared memory, DMA, RDMA, UDS (Unix Domain Socket), (S)RTP over UDP or TCP, etc.
- Compression scheme:
    - For video images: H.264, Motion JPEG, RAW, etc.
    - For point cloud data: G-PCC, V-PCC, RAW, etc.

Symbols

The following symbols are defined and used in the dataflow profiles.

- *#_of_monitored_areas*
    - *#_of_monitored_areas* means the number of monitored areas that one instance of the functional node accommodates. The concrete values of #_of_monitored_areas in a specific deployment scenario will be discussed in Annex E.
- *chunk_interval*
    - *chunk_interval* means the interval time of chunked streaming data such as video image data and point cloud data. The "chunk" process is needed when the Ingestion node (N3) posts these data to Data Hub (N5) to let data consumers access these data as a series of objects. Optimal *chunk_interval* highly depends on the implementation and may affect delay time to get available to data consumers, compression rate, download efficiency, etc.

C.3.1. N1: Local Aggregation Node (for Sensors)

Description

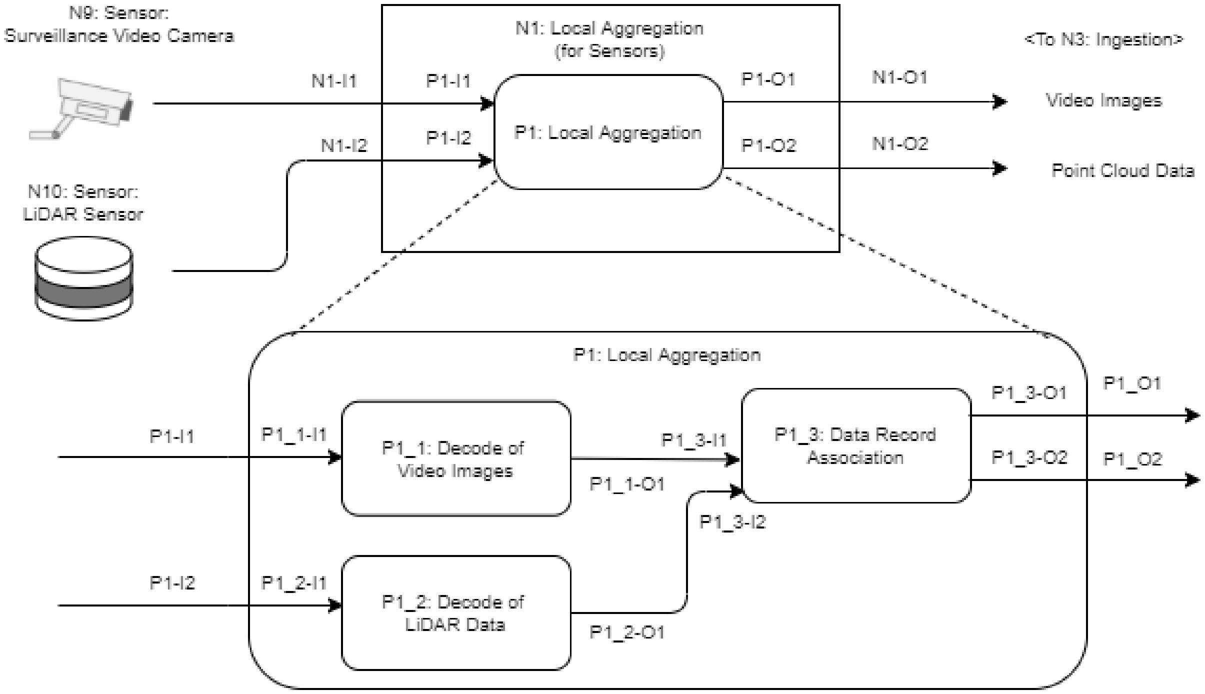


Figure C.3-1: N1: Local Aggregation Node (for Sensors)

The Local Aggregation node (N1) is to collect the output of sensors installed in monitored areas, i.e., surveillance video cameras (N9) and LiDAR sensors (N10), associate multiple separate data collected from the sensors, and send data to the Ingestion node (N3) for further analysis. The Local Aggregation node (N1) consists of the following processes.

- Local Aggregation (P1)
  - Decode of Video Images (P1_1)
  - Decode of LiDAR Data (P1_2)
  - Data Record Association (P1_3)

Process Profiles

The following table shows the detailed description and major attributes of each process.

Table C.3-1: Process Profiles for N1: Local Aggregation Node (for Sensors)

ID	NAME	DESCRIPTION	ATTRIBUTES
P1_1	Local Aggregation / Decode of Video Images	• Receive video images from surveillance video cameras. • Decode the received data into the raw data format.	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling and decoding.
P1_2	Local Aggregation / Decode of LiDAR Data	• Receive data from LiDAR sensors. • Decode and convert the received data into point cloud data in a Cartesian coordinate system.	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling, decoding, and conversion of a coordinate system
P1_3	Local Aggregation / Data Record Association	• Synchronize and associate the decoded video images and point cloud data on a time and space basis. Some meta-data (e.g., time, position, angle, and owner in a unified format) may be embedded into each record from each sensor so that subsequent nodes/processes can efficiently collect and tie relevant records together for their analysis. • Send video images and point cloud data to an Ingestion node.	• # of sources: ○ Camera: 1,000 cameras x *#_of_monitored_areas* ○ LiDAR: 1,000 LiDAR sensors x *#_of_monitored_areas* • Occurrence rate: 15~20 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and coding

Dataflow Profiles

The following table shows the detailed description and major attributes of the dataflows.

Table C.3-2: Dataflow Profiles for N1: Local Aggregation Node (for Sensors)

ID	DESCRIPTION	ATTRIBUTES
N1-I1 **P1_I1** **P1_1-I1**	Compressed Full HD video image streams at 15 fps from surveillance video cameras.	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG) ○ 750 Mbps / source (RAW)

P1_1-O1 **P1_3-I1**	Uncompressed Full HD video images at 15 fps	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Compression scheme: raw • Data size: 6 MB / frame • Occurrence rate: 15 fps / source
N1-I2 **P1-I2** **P1_2-I1**	Sensor data from LiDAR sensors. Each LiDAR sensor captures 100,000 points of data, each 2~4 bytes long, at 20Hz frequency.	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data rate: 32~64 Mbps / source (not compressed)
P1_2-O1 **P1_3-I1**	• Point cloud data in a Cartesian coordinate system • 100,000 points of data, each 16 bytes long (=4 bytes x 4 values) , at 20Hz frequency	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Compression scheme: raw • Data size: 1.6 MB / frame • Occurrence rate: 20 fps / source
P1_3-O1 **P1-O1** **N1-O1**	• Compressed Full HD video images at 15 fps with embedded meta-data • The size of the meta-data is negligible in comparison to the size of the video images.	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG) ○ 750 Mbps / source (RAW)
P1_3-O2 **P1-O2** **N1-O2**	• Compressed point cloud data with embedded meta-data [D. Graziosi 1] [D. Graziosi 2], see Note • The size of the meta-data is negligible in comparison to the size of the point cloud.	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data rate ○ 2 Mbps / source (V-PCC) ○ 7 Mbps / source (G-PCC) ○ 256 Mbps / source (RAW)

Note: Assuming that the compression rate of V-PCC is 125:1, and the compression rate of G-PCC with acceptable quality is 35:1 [D. Graziosi 2]

C.3.2. N2: Local Aggregation Node (for Users)

Description

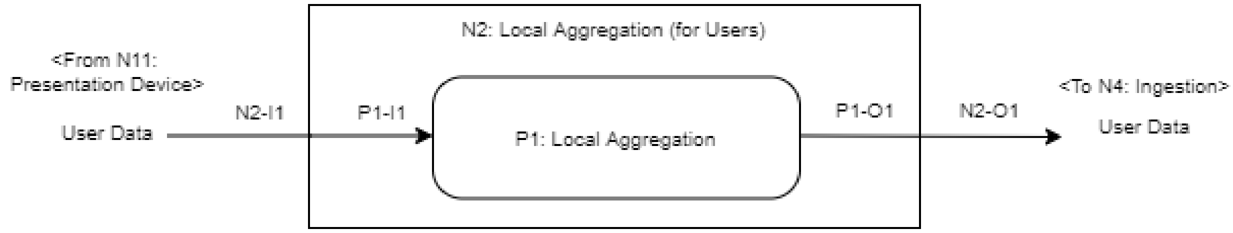


Figure C.3-2: N2: Local Aggregation Node (for Users)

The Local Aggregation node (N2) is a node that communicates with applications on Presentation Device nodes (N11) and collects user data, which may dynamically change. The user data may include user id, time, position, endpoint, etc. This node then posts collected data to the Ingestion nodes (N4) to make them available to authorized functional nodes. The Local Aggregation node (N2) consists of the following processes.

- Local Aggregation (P1)

Process Profiles

The following table shows the detailed description and major attributes of each process.

Table C.3-3: Process Profiles for N2: Local Aggregation Node (for Users)

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Local Aggregation	• Receive user data from Presentation Device nodes (N11). • Post the received data to the Ingestion node (N4).	• # of sources: 1,000 users x #*_of_monitored_areas* • Occurrence rate: 1 occurrence / minute / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-4: Dataflow Profiles for N2: Local Aggregation Node (for Users)

ID	DESCRIPTION	ATTRIBUTES
N2-I1 **P1-I1**	• User data (e.g., position data) posted by applications on Presentation Device nodes (N11) • Assuming that each source posts user data every minute.	• # of sources: 1,000 users x #*_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: 1 message / minute / sources
P1-O1 **N2-O1**	• User data sent to the Ingestion node (N4) for further use by authorized functional nodes.	• # of sources: 1,000 users x #*_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: 1 message / minute / source

C.3.3. N3: Ingestion Node (for Sensors)

Description

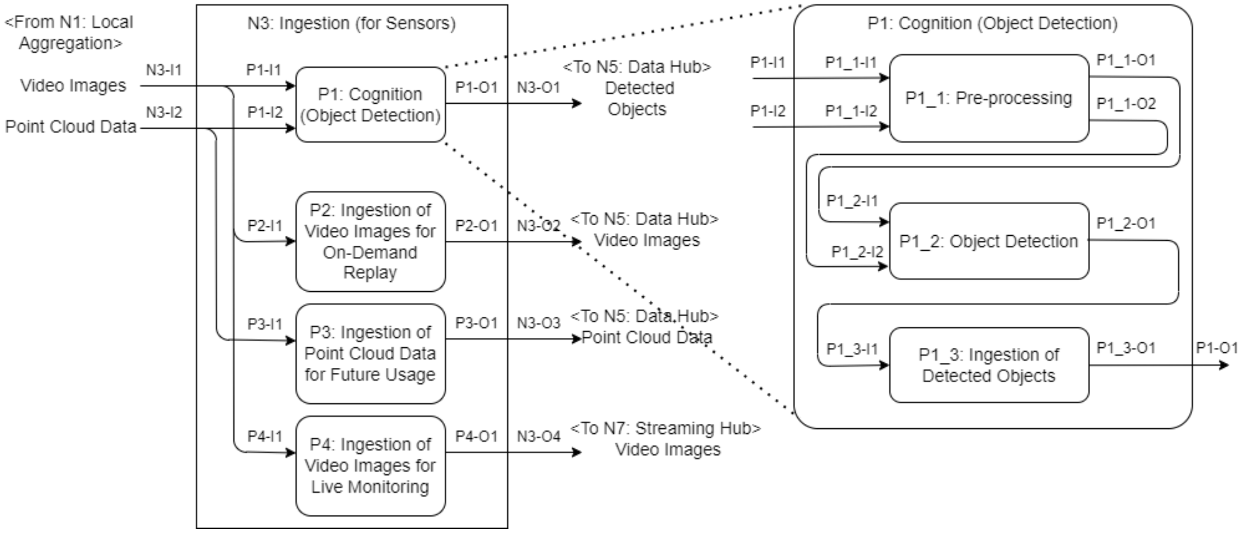

<From N1: Local Aggregation>
Video Images
Point Cloud Data
N3-I1
N3-I2
N3: Ingestion (for Sensors)
P1-I1
P1-I2
P1: Cognition (Object Detection)
P1-O1
N3-O1
<To N5: Data Hub> Detected Objects
P2-I1
P2: Ingestion of Video Images for On-Demand Replay
P2-O1
N3-O2
<To N5: Data Hub> Video Images
P3-I1
P3: Ingestion of Point Cloud Data for Future Usage
P3-O1
N3-O3
<To N5: Data Hub> Point Cloud Data
P4-I1
P4: Ingestion of Video Images for Live Monitoring
P4-O1
N3-O4
<To N7: Streaming Hub> Video Images
P1: Cognition (Object Detection)
P1-I1
P1-I2
P1_1-I1
P1_1-I2
P1_1: Pre-processing
P1_1-O1
P1_1-O2
P1_2-I1
P1_2-I2
P1_2: Object Detection
P1_2-O1
P1_3-I1
P1_3: Ingestion of Detected Objects
P1_3-O1
P1-O1

Figure C.3-3: N3: Ingestion Node (for Sensors)

The Ingestion node (N3) is a node that accepts video images and point cloud data from Local Aggregation nodes (N1), applies primary data processing to them such as data format conversion, compression, indexing, and cognition with AI inference, and post the processed data to the Data Hub node (N5) and/or Streaming Hub node (N7) for further usage. The Ingestion node (N3) consists of the following processes.

- Cognition (P1)
  - Pre-Processing (P1_1)
  - Object Detection (P1_2)
  - Ingestion of Detected Object (P1_3)
- Ingestion of Video Images for On-Demand Replay (P2)
- Ingestion of Point Cloud Data for Future Usage (P3)
- Ingestion of Video Images for Live Monitoring (P4)

The input dataflows (video images and point cloud data) from the Local Aggregation node are appropriately replicated and securely dispatched to the corresponding processes according to the system configuration.

Process Profiles

The following table shows the detailed description and major attributes of each process.

Table C.3-5: Process Profiles for N3: Ingestion Node (for Sensors)

ID	NAME	DESCRIPTION	ATTRIBUTES
P1_1	Cognition / Pre-Processing	• Receive video images and point cloud data related to the monitored areas. • Using a simple image scanning algorithm, check the possibility of objects of interest in the scanned frame. If the possibility is very low, subsequent processes (i.e., P1_2 and P1_3) may be skipped. • Convert the selected data according to the data format suitable for the AI engines used in P1_2 (e.g., HMC to CHW conversion, type conversion, and normalization).	• # of sources: ○ Camera: 1,000 cameras x *#_of_monitored_areas* ○ LiDAR sensor: 1,000 cameras x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling, image scanning, and data format conversion
P1_2	Cognition / Object Detection	• Recognize people, and some relevant goods. • Fuse recognition results from multiple different sensors. That is, if the same sensing point is monitored by multiple sensors, e.g., a video camera and a LiDAR sensor, link their object recognition results. ○ Assuming that two sensors monitor the same sensing point on average. • Assuming that, on average, ten labeled objects of interest are detected in each sensing point. ○ A record for one labeled object includes class name, 2D/3D bounding box size and position, angles, cropped image and point cloud data (uncompressed), and other feature values.	• # of sources: ○ Camera: 1,000 cameras x *#_of_monitored_areas* ○ LiDAR sensor: 1,000 sensors x *#_of_monitored_areas* ○ Sensing point: 1,000 points x *#_of_monitored_areas* • Occurrence rate: 15 OPS / sensing point • Computation cost: 1,000 billion floating operations / occurrence ○ Video image-based recognition: 800 billion floating-point operations / occurrence. See Note 1 ○ Point cloud-based recognition: 200 billion floating-point operations / occurrence. See Note 2 • Possible tasks that can be offloaded to accelerators: AI inference for recognition
P1_3	Cognition / Ingestion of Detected Objects	• Convert the records of the labeled objects for post analysis by Intelligence Application nodes. ○ e.g., compress the cropped image • Send the records to the Data Hub node (N5).	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: 15~20 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling, encoding, and encryption

P2	Ingestion of Video Images for On-Demand Replay	• Receive video images related to the monitored areas. • Create a series of chunked data from incoming video image streams according to *chunk_interval*. • Post the chunked and compressed video image streams to the Data Hub node (N5).	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling, encoding, and encryption
P3	Ingestion of Point Cloud Data for Future Usage	• Receive point cloud data related to the monitored areas. • Create a series of chunked data from incoming point cloud data according to *chunk_interval*. • Post the chunked and compressed point cloud data to the Data Hub node (N5).	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling, encoding, and encryption
P4	Ingestion of Video Images for Live Monitoring	• Receive video images related to the monitored areas. • Select video image streams for live video monitoring according to the system configuration. ○ Assuming less than 5 % of the cameras are selectively monitored in real-time by officers at the same time (i.e., 50 cameras / monitored area). • Post the compressed video image streams to the Streaming Hub node (N7).	• # of sources: up to 50 cameras x *#_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling, encoding, and encryption

Note 1: Yolo v3 [YOLO] requires 140.69 billion floating-point operations for one inference execution against a 608x608 image. Assuming that computation cost increases in proportion to the number of pixels of the image, the computation cost of inference execution for a Full HD image can be around 800 billion floating-point operations.

Note 2: Many schemes for 3D detection are being studied and proposed. One of the most practical schemes is "PointPillars." Its computation cost is around a few hundred floating-point operations for one inference execution [Y. Guo] [J. Ngiam].

Dataflow Profiles

The following table shows the detailed description and major attributes of the dataflows.

Table C.3-6: Dataflow Profiles for N3: Ingestion Node (for Sensors)

ID	DESCRIPTION	ATTRIBUTES
N3-I1 **P1-I1** **P1_1_I1** **P2-I1** **P4-I1**	• Compressed Full HD video images at 15 fps with embedded meta-data • N3-I1 is copied and sent to P1-I1, P2-I1, and P4-I1.	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG) ○ 750 Mbps / source (RAW)
N3-I2 **P1-I2** **P1_1-I2** **P3-I1**	• Compressed point cloud data with embedded meta-data • N3-I2 is copied and sent to P1-I2 and P3-I1.	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data rate ○ 2 Mbps / source (V-PCC) ○ 7 Mbps / source (G-PCC) ○ 256 Mbps / source (RAW)
P1_1-O1 **P1_2-I1**	Uncompressed Full HD video Images at 15 fps with embedded meta-data, which are converted to the format suitable for the AI engine used in P1_2	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Compression scheme: raw • Data size: 6 MB / frame • Occurrence rate: 15 fps / source
P1_1-O2 **P1_2-I2**	Uncompressed point cloud data with embedded meta-data, which are converted to the format suitable for the AI engine used in P1_2	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Compression scheme: raw • Data size: 1.6 MB / frame • Occurrence rate: 20 fps / source
P1_2-O1 **P1_3-I1**	• Labeled data of detected objects • Assuming cropped images and point cloud data are not compressed.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* ○ Assuming two sensors (e.g., one camera and one LiDAR sensor) are used for monitoring one sensing point. • Data size: 50 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source
P1_3-O1 **P1-O1** **N3-O1**	• Labeled data of detected objects with a format suitable for data-sharing • Assuming cropped images and point cloud data are compressed.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 3 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source

P2-O1 **N3-O2**	A series of chunked data of compressed Full HD video images at 15 fps with embedded meta-data	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Data size: ○ (0.4 x *chunk_interval*) MB / chunk (H.264), see Note ○ (5.6~7.5 x *chunk_interval*) MB/ chunk (Motion JPEG) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source
P3-O1 **N3-O3**	A series of chunked data of compressed point cloud data with embedded meta-data	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data size ○ (0.26 x *chunk_interval*) MB / chunk (V-PCC), see Note ○ (0.92 x *chunk_interval*) MB / chunk (G-PCC) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source
P4-O1 **N3-O4**	Compressed Full HD video Image streams at 15 fps with embedded meta-data	• # of sources: up to 20 cameras x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG)

C.3.4. N4: Ingestion Node (for Users)

Description

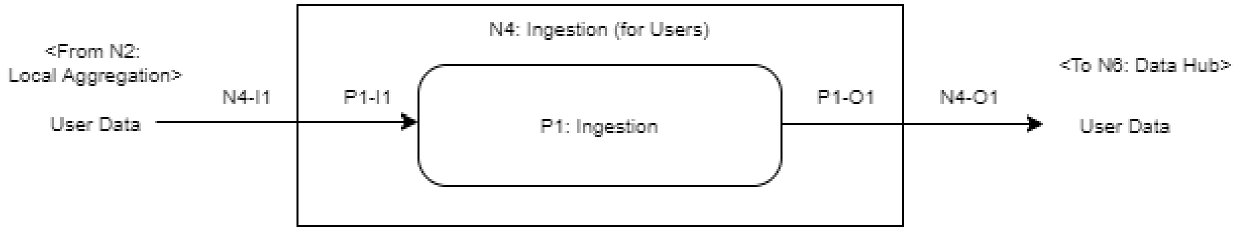


Figure C.3-4: N4: Ingestion Node (for Users)

The Ingestion node (N4) is a node that communicates with Local Aggregation nodes (N2) and collects user data that may dynamically change. The user data may include user id, time, position, endpoint, etc. This node then posts collected data to the Data Hub node (N6) to make them available to authorized functional nodes. The Ingestion node (N4) consists of the following processes.

- Ingestion (P1)

Process Profile

The following table shows the detailed description and major attributes of each process.

Table C.3-7: Process Profile for N4: Ingestion Node (for Users)

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Ingestion	• Receive user data from Local Aggregation nodes (N2). • Post the received data to the Data Hub node (N6).	• # of sources: 1,000 users x *#_of_monitored_areas* • Occurrence rate: 1 occurrence / minute / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-8: Dataflow Profiles for N4: Ingestion Node (for Users)

ID	DESCRIPTION	ATTRIBUTES
N4-I1 **P1-I1**	• User data (e.g., position data) posted by Local Aggregation nodes (N2) • Assuming that each source posts user data every minute.	• # of sources: 1,000 users x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: 1 message / minute / source
P1-O1 **N4-O1**	User data sent to the Data Hub node (N6) for further use by authorized functional nodes	• # of sources: 1,000 users x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: 1 message / minute / source

C.3.5. N5: Data Hub Node (for Sensors)

Description

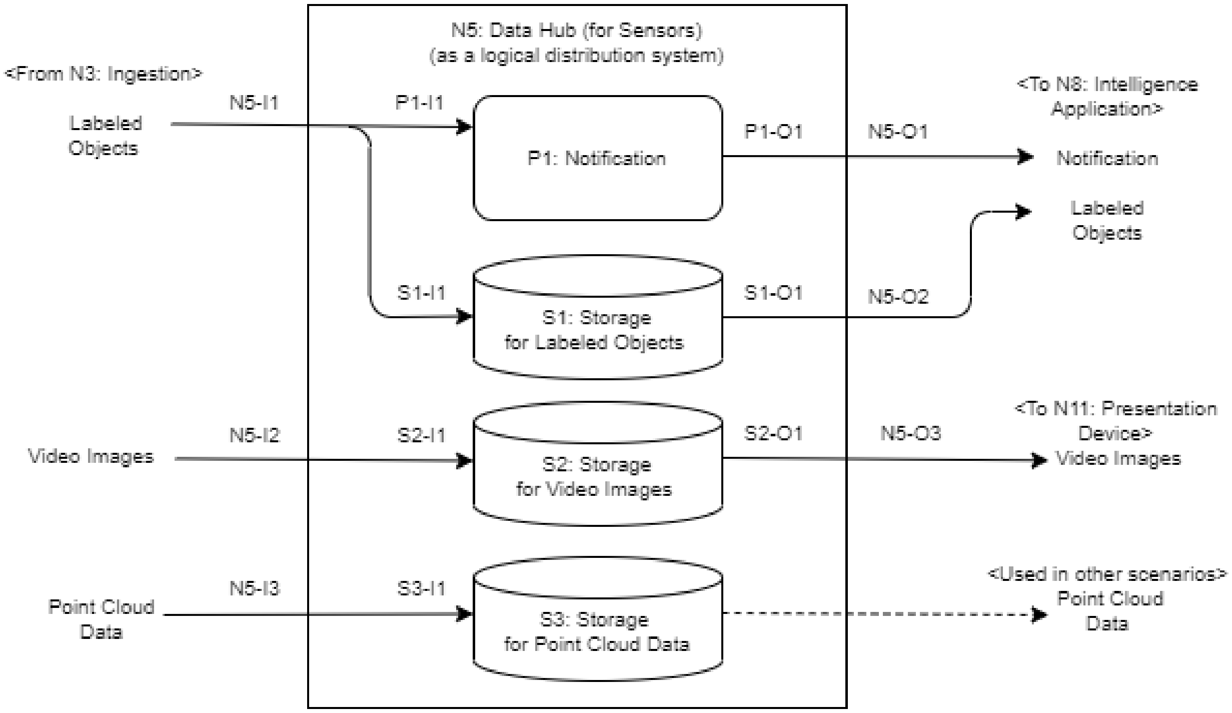

N5: Data Hub (for Sensors)
(as a logical distribution system)
<From N3: Ingestion>
Labeled Objects
N5-I1
P1-I1
P1: Notification
P1-O1
N5-O1
<To N8: Intelligence Application>
Notification
Labeled Objects
S1-I1
S1: Storage for Labeled Objects
S1-O1
N5-O2
Video Images
N5-I2
S2-I1
S2: Storage for Video Images
S2-O1
N5-O3
<To N11: Presentation Device>
Video Images
Point Cloud Data
N5-I3
S3-I1
S3: Storage for Point Cloud Data
<Used in other scenarios>
Point Cloud Data

Figure C.3-5: N5: Data Hub Node (for Sensors)

The Data Hub node (N5) is a functional node that accepts labeled object data, video images, and point cloud data from Ingestion nodes (N3) and distributes them to listening nodes, such as Intelligent Application nodes (N8) and Presentation Device nodes (N11), or store them for further use. This node consists of the following processes and storage:

- Process
  - Notification (P1)
- Storage
  - Storage for Labeled Objects (S1)
  - Storage for Video Images (S2)
  - Storage for Point Cloud Data (S3)

Process Profiles

The following table shows the detailed description and the major attributes of each storage.

Table C.3-9: Process Profiles for N5: Data Hub Node (for Sensors)

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Notification	• Accept posted labeled object data. • Check if any listening nodes subscribe to the accepted data. • Send a notification message to the listening nodes.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: up to 15 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption
S1	Storage for Labeled Objects	• Accept labeled object data posted by Ingestion nodes (N3). • Store the accepted data. • Send labeled object data in response to pull requests from listening nodes. See Note	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: ○ In/Out: up to 15 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption
S2	Storage for Video Images	• Accept video images posted by Ingestion nodes (N3). • Store the accepted data. • Send video images in response to pull requests from listening nodes. See Note ○ Assuming less than 5 % of the cameras are selectively replayed on demand by officers at the same time (i.e., 50 cameras / monitored area).	• # of sources: ○ In: 1,000 video cameras x *#_of_monitored_areas* ○ Out: 50 video cameras x *#_of_monitored_areas* • Occurrence rate: ○ In: 15 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption
S3	Storage for Point Cloud Data	• Accept point cloud data posted by Ingestion nodes (N3). • Store the accepted data. • Send point cloud data in response to pull requests from listening nodes. See Note	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Occurrence rate: ○ In: 20 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

Note: The pull requests are not explicitly shown in the DPD.

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-10: Dataflow Profiles for N5: Data Hub Node (for Sensors)

ID	DESCRIPTION	ATTRIBUTES
N5-I1 **P1-I1** **S1-I1**	• Labeled data of detected objects with a format suitable for data-sharing • Assuming cropped images and point cloud data are compressed.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 3 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source
N5-I2 **S2-I1**	A series of chunked data of compressed Full HD video images at 15 fps with embedded meta-data	• # of sources: 1,000 cameras x *#_of_monitored_areas* • Data size: ○ (0.4 x *chunk_interval*) MB / chunk (H.264) ○ (5.6~7.5 x *chunk_interval*) MB / chunk (Motion JPEG) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source
N5-I3 **S3-I1**	A series of chunked data of compressed point cloud data with embedded meta-data	• # of sources: 1,000 LiDAR sensors x *#_of_monitored_areas* • Data size ○ (0.26 x *chunk_interval*) MB / chunk (V-PCC), see Note ○ (0.92 x *chunk_interval*) MB / chunk (G-PCC) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source
P1-O1 **N5-O1**	• Notification messages • Multiple notification messages from different sensing points can be merged.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 256 Byte / message • Occurrence rate: up to 15 messages / second / source
S1-O1 **N5-O2**	• Labeled data of detected objects with a format suitable for data-sharing • Assuming cropped images and point cloud data are compressed.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 3 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source
S2-O1 **N5-O3**	A series of chunked data of compressed Full HD video Images at 15 fps with embedded meta-data	• # of sources: 50 cameras x *#_of_monitored_areas* • Data size: ○ (0.4 x *chunk_interval*) MB / chunk (H.264) ○ (5.6~7.5 x *chunk_interval*) MB / chunk (Motion JPEG) • Occurrence rate: (1 / *chunk_interval*) chunks / second / source

C.3.6. N6: Data Hub Node (for User)

Description

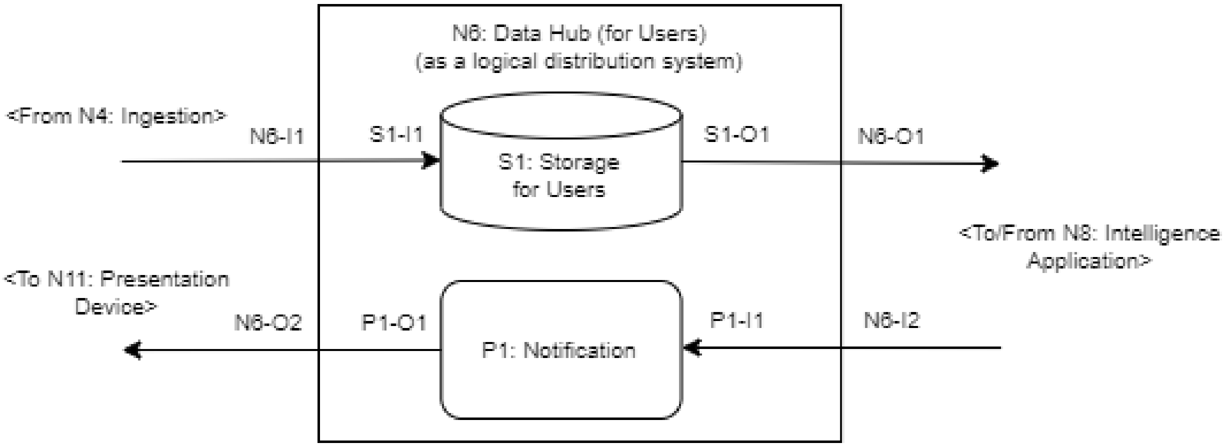


Figure C.3-6: N6: Data Hub Node (for Users)

The Data Hub node (N6) is a functional node that accepts the user data (e.g., user id, time, position, endpoint, etc.) from Ingestion nodes (N4) and notification voice messages from Intelligent Applications (N8). It distributes them to listening nodes, such as Intelligent Applications nodes (N8) and Presentation Device nodes (N11). This node consists of the following processes and storage.

- Storage
  - Storage for Users (S1)
- Process
  - Notification (P1)

Process Profiles

The following table shows the detailed description and the major attributes of each storage.

Table C.3-11: Process Profiles for N6: Data Hub Node (for Users)

ID	NAME	DESCRIPTION	ATTRIBUTES
S1	Storage for Users	• Accept user data (e.g., position) posted by Ingestion nodes for users (N4). • Store the accepted data • Send user data in response to pull requests from listening nodes. See Note 1	• # of sources: 1,000 users x *#_of_monitored_areas* • Occurrence rate: 1 occurrence / minute / source • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

P1	Notification	• Receive voice messages related to sensing points in monitored areas. • Replicate the received voice messages. • Notify the voice message to users who have subscribed messages related to the sensing pints of the voice message. See Note 2	• # of sources: #*_of_monitored_areas* • Occurrence rate: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

Note 1: The pull requests are not explicitly shown in this DPD.

Note 2: Subscription procedures are not explicitly shown in this DPD

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-12: Dataflow Profiles for N6: Data Hub Node (for Users)

ID	DESCRIPTION	ATTRIBUTES
N6-I1 **S1-I1**	• User data received from Ingestion nodes (N3) for further use by authorized functional nodes	• # of sources: 1,000 users x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: 1 message / minute / source
S1-O1 **N6-O1**	• User data sent to authorized listening nodes (e.g., N8) for further usage ○ This is necessary only when an alert message needs to be sent to users.	• # of sources: 1,000 users x *#_of_monitored_areas* • Data size: 1.5 KB / message • Data rate: Ad hoc
N6-I2 **P1-I1**	• Voice messages and the position of the corresponding sensing points ○ Assuming a user listens to voice messages 10% of the time during the day on average.	• # of sources: #*_of_monitored_areas* • Data rate: 64Kbps x 10% / source
N6-O2 **P1-O2**	• Voice messages and their recipients' identifiers ○ Assuming a voice message is delivered to 100 users in a monitored area.	• # of sources: 100 users x *#_of_monitored_areas* • Data rate: 64Kbps x 10% / source

C.3.7. N7: Streaming Hub Node

Description

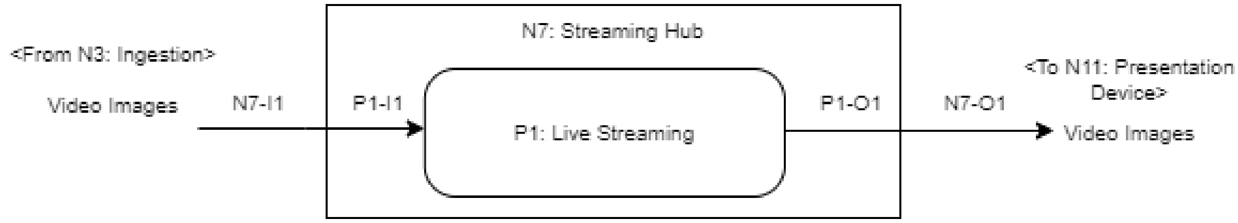


Figure C.3-7: N7: Streaming Hub Node

The Streaming Hub node (N7) is a node that receives one or multiple video image streams from Ingestion nodes (N3) and relays them to Presentation Device nodes (N11).

Process Profile

The following table shows the detailed description and the major attributes of each process.

Table C.3-13: Process Profile for N7: Streaming Hub Node

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Live Streaming	• Receive video image streams. • Duplicate streams if multiple users are watching the video image from the same cameras. ○ Assuming a video image stream from a single camera is duplicated and distributed to 20 users (i.e., 20 Presentation Device nodes). • Send video image streams.	• # of sources: up to 20 cameras x *#_of_monitored_areas* • Occurrence rata: Continuous streaming processing • Possible tasks that can be offloaded to accelerators: duplication of video streams, communication protocol handling, decryption, and encryption

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-14: Dataflow Profiles for N7: Streaming Hub Node

ID	DESCRIPTION	ATTRIBUTES
N7-I1 **P1-I1**	Compressed Full HD video image streams at 15 fps with embedded meta-data	• # of sources: up to 20 cameras x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG)
P1-O1 **N7-O1**	Compressed Full HD video image streams at 15 fps with embedded meta-data	• # of sources: up to 20 cameras x 20 users x *#_of_monitored_areas* • Data rate: ○ 3 Mbps / source (H.264) ○ 45~60 Mbps / source (Motion JPEG)

C.3.8. N8: Intelligence Application Node

Description

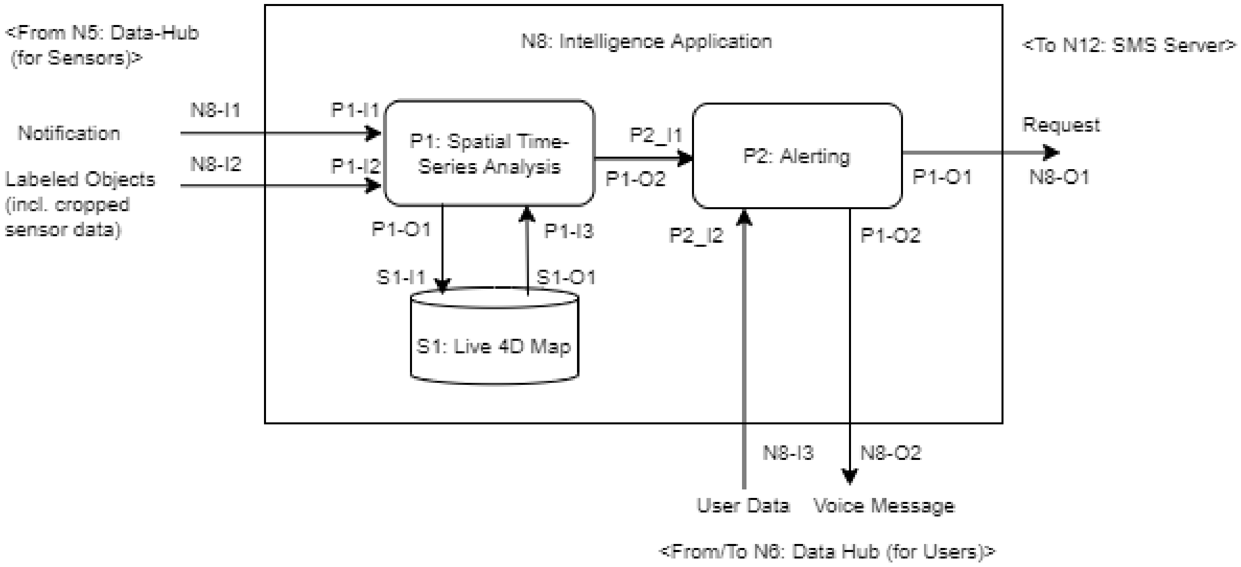

<From N5: Data-Hub (for Sensors)>
N8: Intelligence Application
<To N12: SMS Server>
Notification
N8-I1
P1-I1
Labeled Objects (incl. cropped sensor data)
N8-I2
P1-I2
P1: Spatial Time-Series Analysis
P2_I1
P1-O2
P2: Alerting
P1-O1
Request
N8-O1
P1-O1
P1-I3
S1-I1
S1-O1
S1: Live 4D Map
P2_I2
P1-O2
N8-I3
N8-O2
User Data
Voice Message
<From/To N6: Data Hub (for Users)>

Figure C.3-8: N8: Intelligence Application Node

This Intelligence Application node (N8) is designed to support the Guarding Services scenario in the AM Security UCs. The Intelligence Application node continuously updates the Live 4D Map according to the labeled objects detected by Ingestion nodes (N3). It analyzes the behaviors of the objects on the Live 4D Map so that dangerous situations can be detected and required action(s) can be taken immediately. One of the actions is delivering alert messages to users via Presentation Device nodes (N11). It supports several types of messages, e.g., SMS and voice messages that AI may automatically generate. The recipients of alert messages can be directly pre-configured or be dynamically chosen according to the alert delivering policy.

This Intelligence Application node (N8) consists of the following processes.

- Spatial Time-Series Analysis (P1)
- Alerting (P2)

Process Profiles

The following table shows the detailed description and the major attributes of each process.

Table C.3-15: Process Profiles for N8: Intelligence Application Node

ID	NAME	DESCRIPTION	ATTRIBUTES
P1	Pre-Processing	• Receive notification. • Retrieve labeled objects, including cropped sensor data, relevant to the received notification if necessary. • Decode cropped sensor data, i.e., images and point cloud data. • Update the Live 4D Map. • Load the history date of the state of the sensing point in a specific time interval from the 4D Map. • Perform analysis against the live 4D Map to understand the behavior of detected objects and the overall situation. • If the result meets a certain condition, send requests for making alerts.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: up to 15 OPS / source • Possible tasks that can be offloaded to accelerators: communication protocol handling, decryption, decoding, and analysis with the live 4D Map
P2	Alerting	• According to the detected situation and the position of users (i.e., police officers and/or security guards), decide what messages should be sent and who should receive the messages. • Generate the text of a short message to be sent to users. • Generate a voice message to be sent to users.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: Ad hoc • Possible tasks that can be offloaded to accelerators: message generation, protocol handling, and encryption
S1	Live 4D Map	• Record information about the position of labeled objects on the static structural information of the monitored area with the time stamp. • Return data in response to retrieval queries designating various spatial structures, relationships, and time series-based conditions.	• # of sources and occurrence rate: See S1-I1 and S1-O1 in Table C.3-16 • Possible tasks that can be offloaded to accelerators: communication protocol handling and encryption

Dataflow Profiles

The following table shows the detailed description and the major attributes of the dataflows.

Table C.3-16: Dataflow Profiles for N8 : Intelligence Application Node

ID	DESCRIPTION	ATTRIBUTES
N8-I1 **P1-I1**	• Notification messages • Multiple notification messages from different sensing points can be merged.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 256 Byte / message • Occurrence rate: up to 15 messages / second / source
N8-I2 **P1-I2**	• Labeled data of detected objects with a format suitable for data-sharing • Assuming cropped images and point cloud data are compressed.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 3 KB / labeled object • Occurrence rate: up to 10 labeled objects x 15 fps / source
P1-O1 **S1-I1**	• Write data to the Live 4D Map	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: up to 10 labeled objects x 15 fps / source
P1-I3 **S1-O1**	• Read data from the Live 4D Map	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Occurrence rate: tens of times or higher than S1-I1
P1_O2 **P2-I1**	• Requests for making alerts ○ These requests are sent only when this system decides to issue alerts to users in the monitored area.	• # of sources: 1,000 sensing points x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: Ad hoc
N8-I3 **P2-I2**	• User data in the monitored area ○ This is necessary only when this system decides to issue alerts to users in the monitored area.	• # of sources: 1,000 users x *#_of_monitored_areas* • Data size: 1.5KB / message • Occurrence rate: Ad hoc
P2-O1 **N8-O1**	• Texts of the short messages and their recipients' telephone numbers ○ Assuming 100 users will receive the short message.	• # of sources: *#_of_monitored_areas* • Data size: 1.5KB / message • Data rate: Ad hoc
P2-O2 **N8-O2**	• Voice messages and the position of the corresponding sensing points ○ Assuming a user listens to voice messages 10% of the time during the day on average.	• # of sources: *#_of_monitored_areas* • Data rate: 64Kbps x 10% / source

C.3.9. N9: Sensor: Surveillance Camera

A surveillance video camera with a motion sensor captures the Full HD movie at 15 fps to monitor the area. In this analysis, it is assumed that 1,000 cameras are installed in each monitored area.

Note: If Motion JPEG is used, the stream data from each camera will have a flow rate of around 45~60Mbps. If H.264 is used, it will be 3Mbps.

C.3.10. N10: Sensor: LiDAR Sensor

A LiDAR sensor measures the distance to the surface of surrounding objects as well as the brightness of its surface over a 100m range. It produces 100,000 points of data, each 2~4 bytes long, at 20Hz frequency, i.e., 2 million points data per second, resulting in 32~64 Mbps data stream (without compression). In this analysis, it is assumed that 1,000 LiDAR sensors are installed in each monitored area.

C.3.11. N11: Presentation Device: Local Police System and/or Smartphones

Presentation Devices are the Local Police System and/or Smartphones of users (e.g., police officers and/or guards in charge of the area). Presentation Devices are used for uploading user data, live streaming and/or on-demand replay of video image streams of any camera, and reception of voice / short messages.

C.3.12. N12: External System: SMS Server

An SMS Server is a system operated by 3rd parties to distribute short messages.

Annex D. Workload Optimization with an Event-Driven Approach

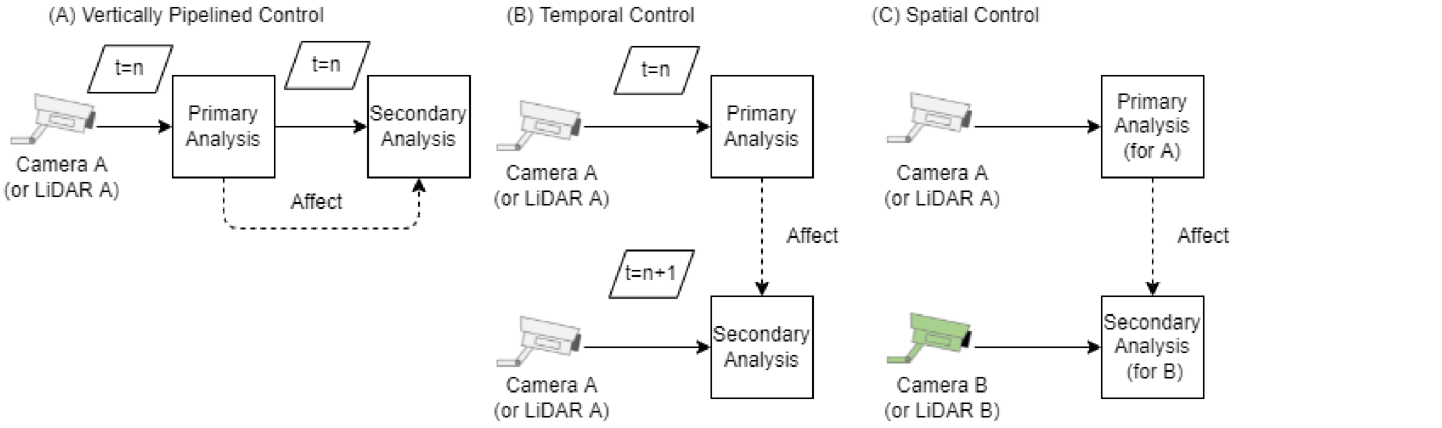


Figure D-1: Basic Strategies of the Event-Driven Approach

To tackle the “Unnecessary Power Consumption Caused by Constant Processing” issue in 4.2, the initial RIM recommends adopting an event-driven approach described in the first whitepaper of IOWN GF [IOWN GF Vision]. This scheme efficiently inspects continuous input stream data (i.e., primary analysis). It generates events that control subsequent deeper analysis tasks (i.e., secondary analysis) over geographically distributed computing resources to improve the overall system efficiency.

As shown in Figure D-1, the following three basic strategies of the event-driven approach will apply to the Area Management Security Use Case (AM Security UC).

- (A) Vertically Pipelined Control: A primary analysis inspects an input image and sends events that trigger secondary analysis for (a part of) the image if needed. An example of vertically pipelined control is that a primary analysis may be an AI inference with a small model and relatively lower image resolution (or just a simple difference detector) and ask for a secondary deeper analysis when any object of interest (e.g., a person) is detected. In another example, a primary analysis may trigger a secondary analysis only when the confidence score of the inference result is lower than the threshold.

- (B) Temporal Control: A primary analysis grasps the current context of the target environment and then dynamically adjusts the analysis parameters for subsequent images to save energy consumption. For example, the benchmark model assumes the frame rate of 15 fps for video images. Although this is needed for human use (e.g., live monitoring and on-demand replay in 3.1), this frame rate is not always necessary in terms of AI-based analysis. Only some special situations, e.g., a situation requiring rapid response time, needs such a high frame rate. Therefore, the RIM lowers (or filters) the frame rate for analysis (of both primary and secondary) in normal situations. But, when the primary analysis detects the necessity of quicker response time (e.g., when any suspicious persons are approaching VIP), it switches to analysis with a higher frame rate.

- (C) Spatial Control: AI analysis tasks for different cameras interwork in a harmonized manner. For example, in the AM Security UC, only AI analysis tasks for the cameras at the entrances of the monitored area may be usually enabled as the primary analysis. Only when it detects persons moving to another area monitored by adjacent cameras, the primary analysis enables analysis for the adjacent cameras as the secondary analysis.

Depending on the use case and the situation, these strategies can be combined and applied together. The initial RIM is recommended to support (A) vertically pipelined control at least and will be extended to support (B) temporal control and (C) spatial control.

With regard to (A) vertically pipelined control, the primary analysis tasks may be deployed in a geographically distributed manner. It can be said that the DPD for the AM Security UC natively supports (A) vertically pipelined control in terms of the relation between the Ingestion node and the Intelligence Application node. This is because the executions of the Intelligence Application node are triggered by the notification of detected objects from the Ingestion nodes. In addition, (A) vertically pipelined control applies to the cognition processes in the Ingestion node as well. The Ingestion node needs energy-consuming CNN-based AI inference, so reduction of the workloads is also important. In this case, Sensor nodes and/or Local Aggregation nodes at the customer premises may be extended to support such primary analysis tasks for event detection. Then the Ingestion nodes execute deeper CNN-based inference only when they receive events from the former functional nodes. Several of today’s sensor devices have lightweight AI functionality inside [Vision Sensor]. The Ingestion node can be a self-contained node supporting both primary analysis and secondary analysis tasks.

As for (B) temporal control and (C) spatial control, such interworking between the primary analysis and the secondary analysis can be implemented as a part of the Ingestion node at the regional edge cloud with assistance from Intelligence Application nodes.

In an event-driven approach, resource consumption, especially at nodes for secondary analysis, will be a function of the event occurrence rate. That is, there is no notion of frame rate or image size. An image, which is just a partial region of the camera frame, is used for inference tasks upon the occurrence of an event. This approach will allow the RIM to reduce computing resources consumed for AI-based analysis while achieving higher cognitive capacity and response speed.

On the other hand, this cannot be achieved with static network bandwidth allocation and computing resource allocation. The demands for networking and computing resources dynamically change according to external conditions, i.e., what is captured by sensor devices. The high elasticity of the DCI architecture will contribute to the efficient support of this event-driven approach. For example, functional nodes that execute a primary analysis can be DCI service consumers in the “procedure for service orchestration and chaining” in 6.3 of the DCI functional architecture document [IOWN GF DCI]. The functional nodes ask the DCI service exposure function to add/remove resources of LSNs and FDNs for the secondary analysis. In another scenario, functional nodes with a second analysis may monitor themselves and trigger service requests for reconfiguration of their resources to fit the dynamically changing workloads. Figure 5.3-3 in 5.3 illustrates how an LSN with heterogeneous accelerators can be reconfigured by utilizing IOWN technologies.

Annex E. Deployment Example of the Reference Implementation Model

5.2 described a geographically distributed data pipeline of IOWN GF RIM. The scale and the structure of the infrastructure for the data pipeline may vary depending on the country and the deployment scenario. This annex shows an example of a deployment scenario and its infrastructure design assuming in Japan for better understanding.

As shown in 5.2, there are three types of computing sites in the geographically distributed data pipeline: monitored area (or customer premise), regional edge cloud, and central cloud. The multiplicity of these sites in Japan's scenario are as follows:

- Customer premise (or monitored area)
    - Around 1,250 monitored areas in total in Japan
    - As described in 2.1.2, we assumed there is one monitored area per 100,000 people
- Regional edge cloud
    - Forty-eight sites in Japan, roughly one regional edge cloud per prefecture.
    - One regional edge is connected to 6 to 96 monitored areas in proportion to the population of the prefecture.
- Central cloud
    - Two sites in Japan (Tokyo and Osaka).
    - The central clouds of Tokyo and Osaka are collocated with the regional edge clouds.
    - The Tokyo site is connected to 19 regional edge clouds, while the Osaka site is connected to 29 regional edge clouds.

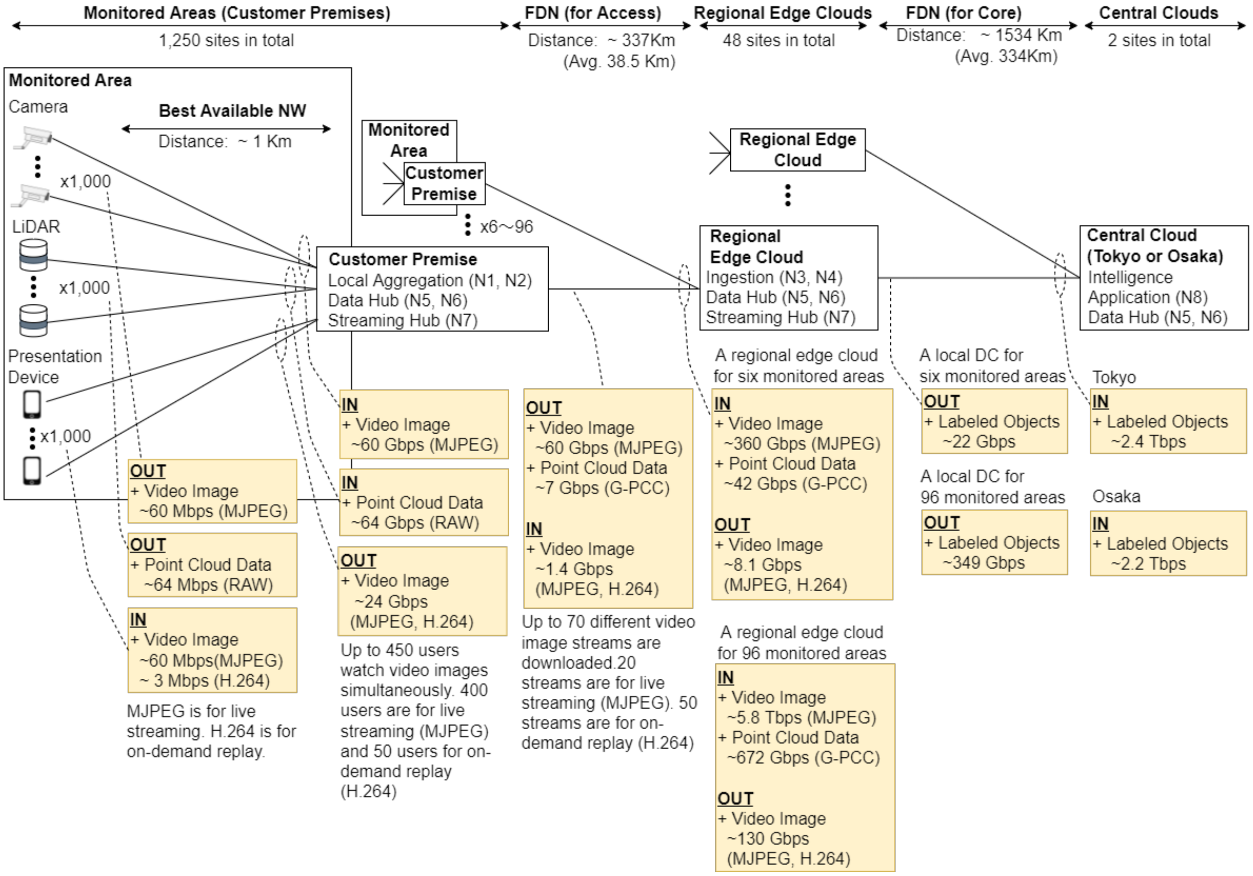


Figure E-1: Overview of a Geographically Distributed Data Pipeline for Japan

Figure E-1 shows an overview of a geographically distributed data pipeline for Japan, that is, connections between computing sites and their properties (e.g., distance and traffic demands).

Here are the assumptions in the network and traffic estimation of Figure E-1.

- The JPN48 topology [JPN48] is applied to the core network.
- The regional edge clouds are placed in the exact locations as the nodes in the JPN48 topology.
- The data volume of "user data" in 3.2.5 and “alerts” in 3.2.6 is much smaller than that of “video data” in 3.2.2, “point cloud data” in 3.2.3, and “labeled objects” in 3.2.4. Therefore, they are omitted in Figure E-1.
- Each component of the Data Hub in different locations only exchanges minimum and necessary data to execute this scenario.
- The distance of the FDN (for access), i.e., from a regional edge cloud to customer premises, is roughly estimated based on the JPN48 topology. The maximum distance from a regional edge cloud to a customer premise is equivalent to one-half of the length from a node for the regional edge cloud to the adjacent nodes, and the average distance is equivalent to a quarter of it.

History

Revision	Release Date	Summary of Changes
1.0	January 27, 2022	Initial Release