Computers, Materials &
Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077008

ARTICLE

Experimental Evaluation of Spatio-Temporal Data Utilization on Floating Cyber-Physical System Platform

Daiki Nobayashi[1,*], **Meiya Tanaka**[2], **Naoki Tanaka**[2], **Riku Nakamura**[2], **Kazuya Tsukamoto**[3],
Takeshi Ikenaga[1], **Shu Sekigawa**[4] **and Myung Lee**[5]

[1]Department of Electrical and Electronic Engineering, Faculty of Engineering, Kyushu Institute of Technology, Fukuoka, Japan
[2]Electronic Engineering Course, Department of Engineering, Kyushu Institute of Technology, Fukuoka, Japan
[3]Department of Computer Science and Networks, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka, Japan
[4]KDDI Corporation, Tokyo, Japan
[5]Department of Electrical & Computer Engineering, City College of New York, New York, NY, USA

*Corresponding Author: Daiki Nobayashi. Email: nova@ecs.kyutech.ac.jp

Received: 30 November 2025; Accepted: 20 February 2026; Published: 09 April 2026

ABSTRACT: To realize local production and consumption of Spatio-temporal data (STD), it is essential to address two key challenges: (1) maintaining data locality by retaining and distributing STD close to their generation area, and (2) enabling application execution on heterogeneous and resource-constrained devices through a lightweight and portable execution platform. To address these challenges, we developed a Floating Cyber-Physical System (F-CPS) that retains both STD and the functions required to process and use the STD within a specific area. In the F-CPS, the STD Retention System directly distributes STD from the generation location and maintains the STD within the target area, thereby ensuring data localization. Furthermore, to facilitate the execution of common applications across heterogeneous devices, we designed a compact and portable application execution platform based on the WebAssembly (Wasm). In this paper, we describe the validation of the feasibility of the F-CPS platform through an integrated experimental evaluation using environmental information. The experiment demonstrated that the F-CPS enables (1) area-based retention of STD and (2) execution of data-processing functions on Wasm containers. The experimental results confirmed that the F-CPS can achieve local production and consumption of environmental data, thereby verifying the effectiveness of this platform.

KEYWORDS: Cyber physical system platform; experiment; data retention system; WebAssembly; local data production and consumption

---

1 Introduction

Cyber-physical systems (CPS) are evolving and becoming more widespread because they use Internet of Things (IoT), cloud, and artificial intelligence technologies to connect the physical and cyber spaces. By integrating these technologies, CPSs aim to solve various societal challenges and create innovative services in fields such as transportation, environmental monitoring, and commerce [1,2]. In a CPS, the physical space gathers information from IoT devices, while the cyber space analyzes the collected data. Based on the analysis results, the system delivers services to the physical space. Efficient data collection, transmission, and use in CPSs require the adoption of next-generation wireless communication technologies, such as Beyond 5G and 6G, along with the development of advanced network infrastructures.

In response to these challenges, recent CPS research has increasingly shifted toward data-driven and decentralized architectures, in which sensing data generated in the physical space are processed closer to their sources rather than being centrally collected in the cloud. In particular, data-driven CPS approaches have been investigated in domains such as smart grids and microgrids, where system behaviors are inferred directly from observed data to support detection, diagnosis, and control without relying on explicit physical models [3]. In parallel, edge- and fog-based CPS architectures have been studied to reduce latency and communication overhead by distributing computation across edge nodes located near physical devices [4]. Furthermore, recent studies on CPS and IoT architectures that anticipate the 6G era emphasize the importance of locality-aware and distributed data processing, while also identifying unresolved challenges in managing, sharing, and using data generated within specific regions in a scalable manner [5]. These studies indicate a clear trend toward decentralized and data-centric CPS designs, highlighting the need to reconsider how locally generated data should be handled within CPS architectures.

Furthermore, among the data generated in the physical world that require locality-aware handling in decentralized CPS architectures, there exist "spatio-temporal data (STD)" that depend on the location and time of data generation, such as traffic, disaster, weather, and time-sensitive advertising information. However, existing data-driven and decentralized CPS architectures do not explicitly address how such region-specific data should be distributed and used within the area where they are generated. To address this issue, our research group has been developing a "Floating cyber-physical system (F-CPS)" as a region-specific CPS that retains STD and the functions (applications) to process and use STD within the local area [6]. We aim to accelerate STD distribution and achieve effective STD use with an F-CPS.

Fig. 1 shows an overview of the F-CPS. In the F-CPS, as a method of STD distribution, we have developed an "STD Retention System (STD-RS)." The STD-RS uses a device edge (DE), which is a mobile device equipped with communication and computational capabilities, owned by users to disseminate and maintain STDs [7]. In the STD-RS, DEs within the area periodically broadcast STD. This enables direct delivery of STDs to users without the need for remote servers, such as cloud systems.

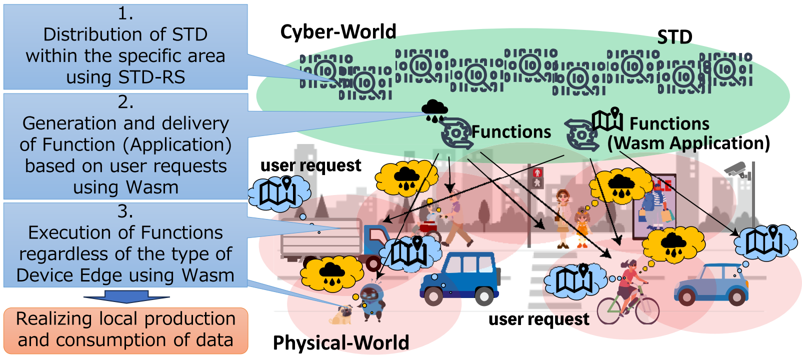


Figure 1: Overview of an F-CPS.

Additionally, we have developed a method for retaining application functions to process STD, allowing users to use STD more flexibly. This function operates as a background application within a designated area to provide various services. Because this function must run as a common application across diverse devices, the F-CPS platform requires an application execution system that is independent of both hardware and the operating system (OS). The most widely adopted approach involves container technology, a virtualization technique that encapsulates applications, libraries, and OS components into a single package.

However, in an F-CPS, executing functions within traditional containers results in a significantly large container size, potentially creating a bottleneck in wireless data transmission. Furthermore, traditional container technologies inherently depend on the environment in which they were created, such as the OS and processor architecture. To address these challenges, we incorporated the WebAssembly (Wasm) as a compact and portable application execution platform for the F-CPS. Wasm enables applications to run on web platforms, which are widely available across various devices, thereby facilitating an execution environment independent of system architecture and kernel configurations [8].

In this study, we conducted integrated experiments on the F-CPS platform, focusing on STD-RS and the application execution platform using Wasm containers, to evaluate the feasibility of locally producing and consuming STD. The experiment used various environmental data collected within a college facility and validated the operational workflow, including (1) the retention of STD by the STD-RS and (2) the retrieval and execution of functions via multi-access edge computing (MEC) servers. While our prior studies independently developed and evaluated the STD-RS and the Wasm-based application execution platform, this study addressed the practical challenges that arise when integrating these components into a unified F-CPS platform. Through integrated experiments, we clarified the feasibility of end-to-end STD use, from local data retention to immediate function execution on user devices.

The main contributions of this paper are as follows:

- End-to-end integration of STD retention, function provisioning, and execution in an F-CPS architecture: We present a new and technically challenging integrated experiment that unifies STD retention by an STD-RS, function provisioning via MEC, and Wasm-based function execution on user devices. Unlike conceptual or simulation-based studies, our implementation validated the complete end-to-end data and function flow, from sensor-generated STD to real-time analysis and visualization on user nodes within a confined area.
- Prototype-based experimental validation in a realistic hybrid communication environment: We designed and implemented an indoor F-CPS testbed that combines local Wi-Fi—based data distribution for STD retention with Wi-Fi/B5G, connected with MEC for function delivery. Through experiments using actual sensors, DEs, a MEC server, and mobile user devices, we demonstrated one of the few practical evaluations of Wasm applications executed on mobile terminals over 5G/B5G networks in conjunction with local data distribution.
- Experimental demonstration of real-time, location-specific data use through integrated mechanisms: Through integrated experiments, we quantitatively confirmed that the F-CPS enables local production and consumption of environmental STD, achieving end-to-end data delivery and low-overhead function execution. The results highlight how the combination of STD-RS and lightweight Wasm function execution enables practical, location-aware data use without reliance on centralized cloud infrastructures.

The remainder of this paper is organized as follows. Section 2 introduces related research on information sharing and the use of Wasm containers. Section 3 gives an overview of STD-RS and Section 4 explains its function and execution in an F-CPS. Section 5 describes the experimental environment and the evaluation metrics, and Section 6 provides experimental results. Finally, Section 7 presents a summary of our paper.

Abbreviations: The abbreviations that appear frequently throughout this paper are listed in Table 1.

Table 1: List of Abbreviations.

Abbreviation	Full Term
CPS	Cyber-Physical System

(Continued)

Table 1 (continued)

Abbreviation	Full Term
F-CPS	Floating Cyber-Physical System
STD	Spatio-Temporal Data
STD-RS	Spatio-Temporal Data Retention System
MEC	Multi-access Edge Computing
Wasm	WebAssembly
UE	User Equipment
DE	Device Edge

2 Related Work

In this section, we describe related works on communication control methods for mobile nodes and related studies on the applicability of Wasm as a container for IoT devices with hardware constraints.

2.1 Information Transfer Control in Mobile Nodes

With a focus on the communication for STD in a specific area, delay tolerant networking (DTN) technologies have been researched as a relay-forwarding technique in various networks [9,10]. DTN involves coordinated control over the transmission, relay, and reception of information among communication nodes to optimize network resource sharing and communication performance. One of the communication methods in DTN is epidemic broadcasting, where messages are repeatedly forwarded from a moving node to nearby moving node via short-range wireless communication, thereby disseminating messages throughout the network without dedicated infrastructure. In [11–13], floating content sharing is described as a method for delivering information by intentionally limiting the coverage and lifetime of epidemic broadcasting. In information sharing with floating contents, messages are embedded within the available area (anchor zone) and are given a time to live (TTL). Messages are only forwarded to mobile nodes within the zone. Subsequently, messages are deleted when the TTL expires or when the mobile node that transmitted the information leaves the zone. In [11], it is mentioned that in this control method, there is a risk of messages disappearing due to buffer overflow on mobile nodes, leading to unfair propagation among messages. Therefore, the authors proposed "proportional control for floating content sharing (PFCS)" to control the message possession ratio (the fraction of mobile nodes carrying a message in the anchor zone).

In an F-CPS, adaptive transmission rate control is also implemented. However, while PFCS controls transmission to achieve the target message possession ratio, the STD-RS aims to periodically deliver STD to the entire area. Furthermore, the STD-RS introduces a mechanism that promotes data use by eliminating the need for query-response information distribution between adjacent nodes in floating contents.

2.2 WebAssembly as a Container for IoT Devices

Currently, container execution and management using Docker are widespread. However, Docker containers can hold a large amount of data and have complex structures. When running a Docker container on hardware-constrained IoT devices, challenges arise in terms of system memory consumption, startup time, and transfer time. Therefore, Napieralla [14] introduced the use of Wasm as a compact and lightweight container execution platform for IoT devices. Wasm refers to a binary code and its corresponding runtime environment, which are designed for execution on a virtual machine, thereby achieving independence of the actual hardware and OS. In [14], the author analyzed the performance of Wasm and Docker containers from the viewpoint of various metrics. Wasm outperformed a Docker container in terms of having lower system memory consumption and reduced startup time. This study concluded that Wasm is suitable for the sporadic execution of simple programs and for serverless computing.

The development of the WebAssembly system interface, which enables input/output operations to the host system's file system, network, and other resources, has made Wasm an attractive option for application in serverless environments [15]. In [16], the authors described the design of a hybrid serverless platform for Wasm by extending Kubernetes, which is a container orchestrator. The application of Wasm could provide a new approach to reduce cold start delays in serverless scenarios, which require short execution times.

Therefore, the use of Wasm containers can be considered to be an effective approach for executing region-specific functions on various and diverse device edges (DEs) in an F-CPS.

3 Floating Cyber-Physical System

In this section, we provide an overview of the STD-RS, which serves as the STD distribution infrastructure and executes functions for effectively using STD in the F-CPS.

3.1 Overview of the Retention System

The purpose of the STD-RS is to effectively provide STD directly from its source to users. We define the region for delivering STD as the retention area. Within this area, DEs autonomously perform distributed management of STD for a certain period. Each DE within this area broadcasts STD at appropriate timings based on the control of the STD-RS, to facilitate its dissemination and maintenance throughout the entire area. This system also enables the delivery of STD in a passive manner, thereby reducing the burden on the existing network infrastructures such as the Internet.

Fig. 2 shows an overview of the STD-RS. The STD-RS consists of the source point (data origin), DEs responsible for retention function including data dissemination and maintenance of STD within the retention area, and user equipment (UE) that receives and uses the STD. The source point attaches control information such as the TTL to the STD and broadcasts it to the surrounding DEs. The DEs receiving the STD individually control STD transmission based on the control information included in the received STD and surrounding DE density, and broadcast STD if necessary for distribution. This relay operation is propagated between the DEs within the area. As a result, all UE staying in the area can passively receive STD without the need for network infrastructure. So far, we have developed various transmission control methods, some of which control the transmission probability based on the radio-wave propagation environment or retention of a large amount of STD [7]. In the F-CPS, the STD-RS enables the realization of STD circulation within a specific area.

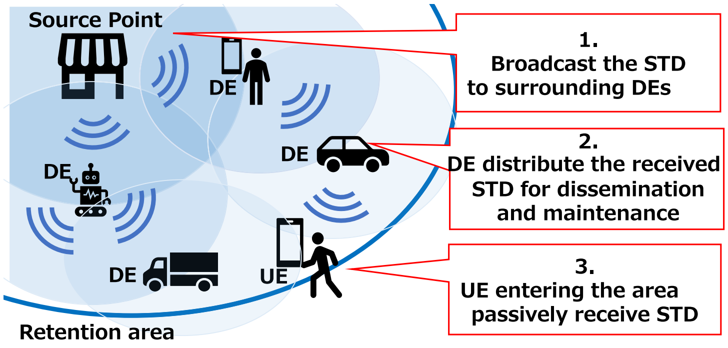


Figure 2: Overview of the STD retention system.

3.2 *Overview of the Function Platform in F-CPS*

The F-CPS enables local production and consumption of data by retaining STD and functions within the retention area. This function of the F-CPS is provided as a means for UE to analyze and process STD according to user needs, thereby accelerating flexible data use based on specific locations and times. Therefore, the F-CPS platform needs to consider how to generate functions based on user requests and how to distribute those functions within a specific area. We focused on MEC as an architecture for decentralized management of local STD and user requests.

Here, MEC is particularly suitable for function generation in the F-CPS for several reasons. First, in future B5G and 6G networks, MEC is expected to be deployed close to end users to support distributed processing and reduce latency. Second, because MEC operates within the local network domain, it can observe and manage the flow of data, such as locally generated STD, within its coverage area. This capability allows the MEC to generate applications that match the characteristics and needs of users in the local area. Furthermore, with the advancement of generative AI technologies, MEC is expected to support on-demand creation of local-oriented functions (applications) by analyzing regional data trends and user behaviors within the network. These characteristics make MEC an appropriate and promising platform for generating and distributing localized functions in the F-CPS.

Fig. 3 shows an overview of function generation and distribution in the F-CPS. In the envisioned network architecture of the B5G and 6G era, MEC is connected to base stations located close to users and has the role of data management and distribution within specific areas. In the F-CPS, MEC generates and delivers various applications as functions, based on the result of an analysis of STD and user requests. The generated functions are delivered within the local area via a base station. However, because not all DEs may be able to connect to the base station, functions are also distributed throughout the region using the STD-RS, similar to STD.

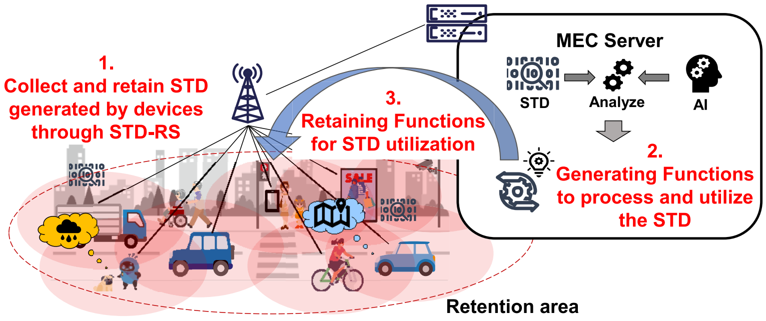


Figure 3: Overview of function retention within the F-CPS.

Furthermore, the F-CPS includes the common application execution platform using Wasm for various types of DEs [8]. The execution platform using Wasm can be run on web platforms regardless of hardware and OS. Moreover, Wasm applications generated in binary code are extremely lightweight compared to those of traditional containers, and thus fast and stable data distribution can be achieved. Using Wasm for the functions managed by the MEC server and retaining them in suitable locations and times by STD-RS allow the F-CPS to use the STD flexibly, which results in local production and consumption of data in the F-CPS. Moreover, functions and applications for the F-CPS implemented using Wasm can be executed on various types of devices regardless of their hardware or OS. This makes it possible for these functions to be used across different device roles, such as DEs and UEs, within the STD-RS. As a result, the F-CPS is expected to leverage all devices in the region as computational resources, thereby enhancing the overall flexibility and capability of the system.

4 Proposed Method

In this study, to realize the F-CPS concept described in the previous section, we constructed an architecture that integrates the STD-RS with a function platform based on Wasm. In this architecture, we implemented the functionalities of the source point, DE, and UE to demonstrate the feasibility of the F-CPS in indoor environments. This section first describes how the STD-RS controls transmission in a way tailored for realizing the F-CPS, and then explains the execution platform for integrating the STD-RS with Wasm applications.

4.1 Transmission Control Method of the STD-RS

In this study, we aimed to build an indoor F-CPS and distribute its environmental information using the STD-RS. Two types of sensors as data source points periodically collected environmental information. The environmental sensor collected temperature, humidity, discomfort index, illuminance, atmospheric pressure, and noise level. The Wi-Fi sensor counted the number of devices within approximately 10 m using a threshold of −71 dBm for the received signal strength indicator, as derived from preliminary experiments. By distributing this information through the STD-RS, we aimed to enable real-time acquisition of environmental information within the indoor space.

This information was transmitted from the sensors serving as the source point, disseminated and maintained by the DEs functioning as retention nodes, and passively received by UE. Sensors generated STD based on the measured values, and then periodically broadcast the STD. DEs as retention nodes relayed the STD received from the sensors. The DEs operated based on the previously described transmission control by the STD-RS [7], and probabilistically broadcast data according to the number of surrounding devices and the number of transmissions. In addition, to suppress excessive data transfers in indoor F-CPS environments, a TTL was assigned to each data packet so that data exceeding the specified lifetime were no longer forwarded, allowing the data to naturally disappear from the area. Finally, the UE passively received the STD retained by the DEs. A UE receiving data whose TTL had already expired discarded the data to prevent the use of outdated information.

4.2 Function Execution Platform for Integrating the STD-RS and Wasm Application

Our F-CPS system creates and provides a Wasm application that analyzes the received STD (indoor environmental information), displays the analyzed results, and notifies the user (UE) whenever newly retained STD are received. The Wasm application on the UE needs to access the received STD. However, Wasm applications typically do not allow direct access to network resources or storage on the same device for security reasons. Therefore, we used a data access method that enables a Wasm application to access data on the terminal using HyperText Transfer Protocol Secure (HTTPS) by modifying the Wasm execution platform. Furthermore, we packaged the Wasm application as a Wasm container to enable HTTPS communication.

Fig. 4 shows the operational flow of the function execution platform. First, we configured the system such that the data received by the STD-RS can be accessed only from the local environment via a web server built into UE, due to the limitations of Wasm applications. We directly delivered the Wasm application to UE from the MEC server connected to the 5G/6G network. As an F-CPS concept, applications implemented as Wasm containers should ideally be provided in a push-based manner from the MEC server whenever a device connects to an underlying network such as 5G/6G, delivering applications specialized for that specific area. However, because such a mechanism would require integration with association and authentication procedures in 5G/6G networks, this study adopted a simplified approach: when joining the 5G/6G network, users explicitly download the application through a web browser, enabling easy deployment.

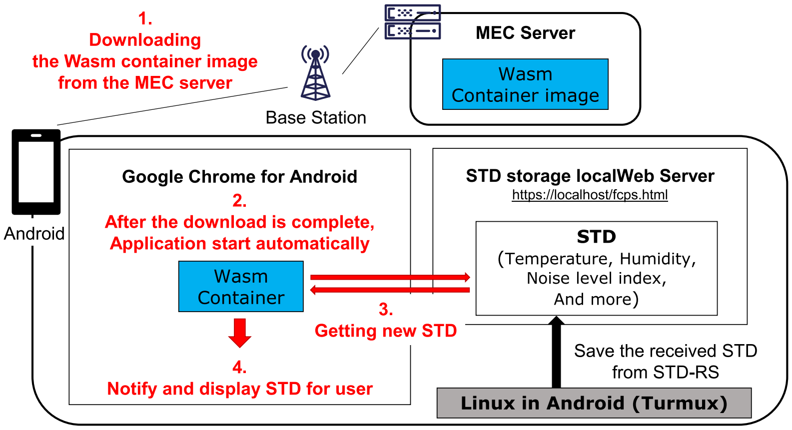


Figure 4: Operational flow of the proposed function execution platform.

The MEC server prepared a Wasm container image created from the Wasm container in advance. After that, UE accessed the MEC server to download the Wasm container image and its execution platform (Fig. 4-1). In other words, in the F-CPS, the deployment of Wasm applications from the MEC to UE follows a pull-based system. In this system, the Wasm container was automatically built after the download, and then the Wasm application was executed using a web browser on UE (Fig. 4-2). The Wasm application periodically checked the STD received from the local web server using HTTPS (Fig. 4-3). When the Wasm application detected new STD, it analyzed environmental information such as the discomfort index, noise level, and number of devices. Based on these analyses, the UE displayed information about the comfort level at each location (Fig. 4-4). In addition, the Wasm application notified the user of updated information through the web browser whenever new STD were detected.

5 Experimental Validation of End-to-End Data Flow and Function Execution

The integrated experiments performed in this study were designed to validate the feasibility of the F-CPS platform through an integrated implementation of the STD-RS and the Wasm-based execution platform using actual equipment. The purpose of the integrated experiments presented in this section was not to evaluate the communication or propagation performance of the STD-RS itself, such as dissemination confinement or redundancy behavior. Instead, the experiments aimed to validate the feasibility of the proposed F-CPS platform by confirming end-to-end data flow from sensors to UE and the real-time use of retained STD through Wasm-based applications.

In this experiment, we assumed an indoor mobile environment in which UE simultaneously maintained a wide-area cellular connection and a local short-range communication link. Such hybrid connectivity is commonly observed in practical mobile scenarios, where smartphones access cloud or edge services via cellular networks (e.g., 4G/5G) while participating in local Wi-Fi-based communication for proximity data exchange.

5.1 Experimental Environment and Equipment

In this study, as an application for the integrated experiments of the F-CPS platform, we envisioned the use of environmental information in indoor facilities for the purpose of crowd guidance and behavior modification. Therefore, we attempted to provide an F-CPS for collecting, analyzing, and using environmental information within a college facility. The experimental environment and system configuration are described below.

The integrated experiments were conducted at GYMLABO, a co-working space located on the Tobata campus of the Kyushu Institute of Technology. This facility is equipped with the mobile application execution environment of the National Institute of Information and Communications Technology (NICT), namely the highly reliable and flexible B5G/IoT testbed. Through licensed devices, users can access the B5G testbed network. Additionally, on the NICT B5G testbed, we connected the MEC server that we prepared to the backhaul network to which the radio unit was connected. MEC server was implemented on an HP ProLiant DL160 Gen10 server, configured with Ubuntu 22.04 as the OS. With MEC server, we deployed the Wasm container management framework (F-CPS orchestrator) developed in our previous work [8], enabling the management of Wasm containers running on the UE. This enabled the provision of Wasm container images and their execution platform for areas accessible to the B5G testbed.

Fig. 5 shows the experimental environment, and Fig. 6 shows an image of GYMLABO. In the integrated experiments, we set two sensors and one DE using Raspberry Pi 3 Model B+ with the Raspberry Pi OS. We used the Omron 2JCIE-BU01 sensor for environmental sensing and the Buffalo WI-U3-866DS for Wi-Fi sensing. As UE, we employed a smartphone with Android OS 12 (FCNT SD01) compatible with the NICT B5G testbed and a laptop (Dell Latitude E5440) with Ubuntu 22.04. Google Chrome was used as the web browser for executing the application on UE.

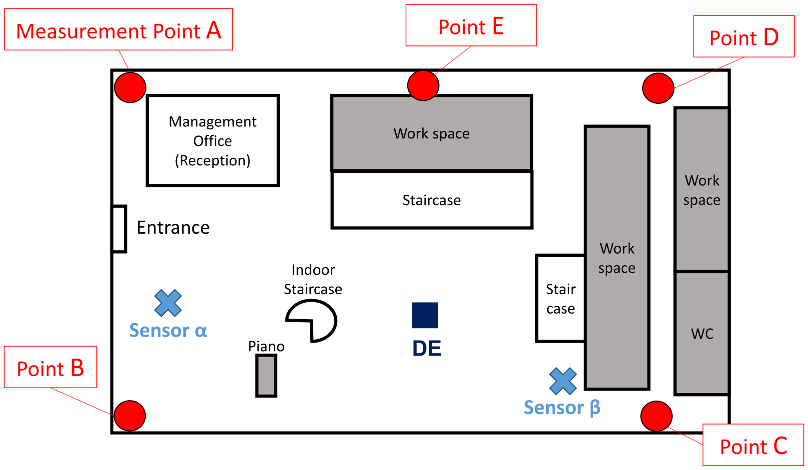
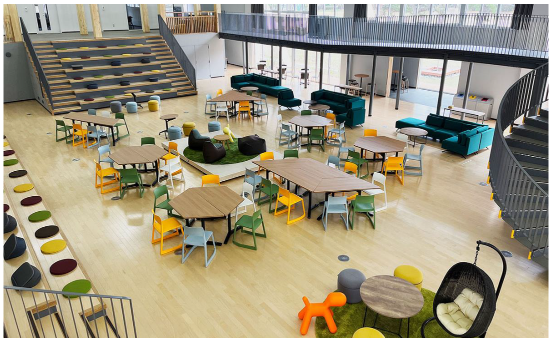


Figure 5: Experimental environment.

Fig. 7 shows the network environment in GYMLABO. In this integrated experiment, the sensors, a DE, and UE must be present on the same network to enable the UE to receive the STD delivered by the STD-RS. Based on the assumed hybrid connectivity, we constructed a hybrid communication environment in which locally generated STD were disseminated through a Wi-Fi Direct peer-to-peer (P2P) communication network using Wi-Fi 5, while Wasm-based applications were delivered to UE via the NICT B5G testbed. Each network was assigned an IPv4 address with a subnet mask length of 24, and UE could participate in both networks. This configuration enabled user devices to simultaneously participate in local STD sharing through short-range P2P communication and access wide-area computing and application provisioning resources provided by the MEC server.



https://www.gymlabo.kyutech.jp/

Figure 6: Image of GYMLABO.

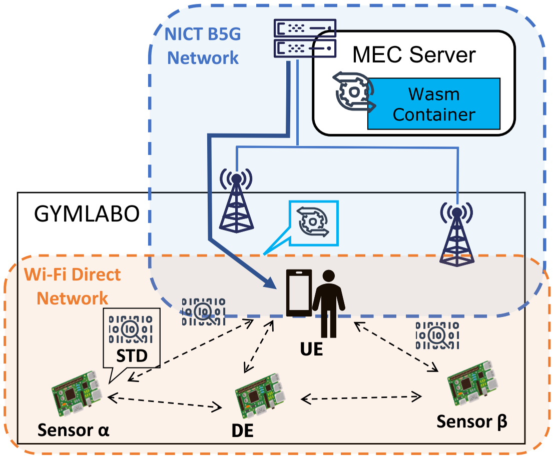


Figure 7: Network environment.

It is important to note that the goal of this integrated experiment was not to evaluate retention performance in the STD-RS, such as the confinement of STD dissemination within a specified area or the behavior of duplicate STD generation. Rather, the purpose was to verify that, once devices join the Wi-Fi Direct P2P network, STD can be successfully delivered from the STD-RS to UE and immediately used by applications in the environment. For this reason, we intentionally configured only a single DE, which enabled us to focus on demonstrating real-time use of STD by applications while keeping the data delivery process simple and aligned with the intended use case. Comprehensive evaluations involving multiple DEs, including node mobility, multi-hop environments, the impact of interference, analysis of multiple STD handling, and region-constrained dissemination have already been presented in our prior work [6,7], and these aspects were outside the scope of the present experiment.

This hybrid configuration does not represent an unconventional experimental setup but rather reflects a feasible and commonly observed operational model in real-world mobile environments, where cloud–edge computing resources and local data ecosystems coexist. Accordingly, the experimental design is appropriate for validating the feasibility of real-time, location-specific use of STD and the concept of "local production and consumption" of environmental information within a practical communication environment.

In addition, it should be emphasized that the use of Wi-Fi Direct P2P for STD dissemination and the use of the B5G network for Wasm application distribution is consistent with practical operational scenarios. In real-world mobile environments, it is common for smartphones to maintain a cellular (e.g., 5G) connection while simultaneously participating in a local Wi-Fi network for device-to-device communication. Our experimental configuration reflects such commonly observed hybrid connectivity, where computing resources and applications are provisioned via the B5G infrastructure, while locally generated STD are shared and consumed within the environment through a short-range P2P network.

This design choice aligns with the aim of the present experiment: to demonstrate that real-time, location-specific use of STD can be achieved in a merged communication environment combining wide-area 5G/B5G infrastructure and local networking. Rather than simulating an abstract architecture, our setup reflects a feasible operational model in which cloud–edge resources and local data ecosystems coexist. Thus, the configuration used in this study is appropriate for validating the feasibility of local production and consumption of environmental data within a practical communication environment.

5.2 Evaluation Metrics

In this study, we evaluated various aspects of the constructed F-CPS platform as part of its verification, including a series of operational tests, STD retention by STD-RS, and Wasm application execution.

For evaluating the STD-RS, we verified the coverage area of STD retention and the time taken to retrieve STD. As a metric to evaluate the coverage area, we defined the STD delivery rate as the ratio of the number of STD receptions by UE to the number of STD transmissions by the sensors and DE. UE with the Android OS was placed at each measurement point shown in Fig. 5. We evaluated the delivery rate at each point when the environmental data of approximately 160 bytes was transmitted 100 times from each sensor using the STD-RS. Additionally, for the STD retrieval time, we measured the time from the start of STD transmission by each sensor and DE until UE received the STD.

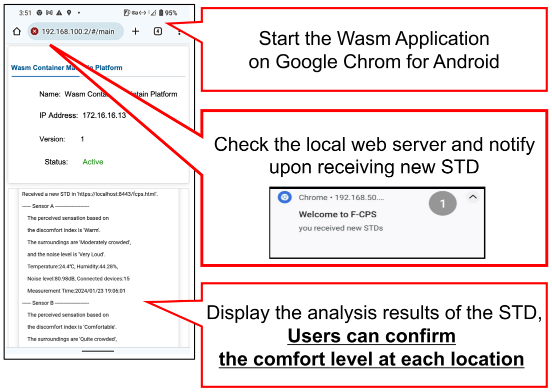
For the evaluation of Wasm application execution, we verified the startup time of the Wasm application and the percentage of battery usage during execution. Regarding the startup time, we conducted the assessment on Google Chrome on the UE with Android OS. We measured the time from initiating access to the MEC server until the complete display of the STD on the screen. We also measured the impact of a cache function of the web browser on startup time. In this evaluation, cache refers to browsing site history, site data, and cached images/files in Google Chrome. Furthermore, to assess the load imposed by the F-CPS system on UE, we measured the battery usage percentage. This battery usage was measured by referencing the logs collected by the Android OS in the background. Note that the CPU, RAM, and input/output overhead introduced by Wasm itself depends on the scale of the Wasm application. However, the overhead of the Wasm container platform has already been evaluated in prior work [8], and therefore is not discussed in this paper. Furthermore, this study evaluated the feasibility of end-to-end operations in the F-CPS. Comparisons between the Wasm container management platform for the F-CPS and conventional Docker containers have already been conducted in prior work [8], which demonstrated that our Wasm container management platform imposes sufficiently low overhead.

6 Experimental Results

In this section, we describe the verification results and evaluations of the integrated experiments.

6.1 Verification with a Series of Operations on the F-CPS Platform

For the operational verification of the F-CPS platform, we conducted experiments using UE in the environment shown in Fig. 5. The results of the integrated experiments confirmed successful operation from the collection of STD at sensors to the retention of STD by the STD-RS and to the execution of functions by the Wasm application. Fig. 8 shows a screen of the Wasm application running on Google Chrome on UE with an Android device. The Wasm application allows the display of the comfort level at each location to the user by receiving and analyzing STD based on parameters such as discomfort index, the number of devices, and noise level. Additionally, this Wasm application was confirmed to operate on a laptop with Ubuntu OS. Therefore, we successfully used Wasm to realize the provision of functions that promote data use on UE with a web platform, using the same application.



Received a new STD in 'https://localhost:8443/fcps.html'.
------ Sensor A ----------------------
The perceived sensation based on
the discomfort index is 'Warm'.
The surroundings are 'Moderately crowded',
and the noise level is 'Very Loud'.
Temperature:24.4℃, Humidity:44.28%,
Noise level:80.98dB, Connected devices:15
Measurement Time:2024/01/23 19:06:01
------ Sensor B ----------------------
The perceived sensation based on
the discomfort index is 'Comfortable'.
The surroundings are 'Quite crowded',

Start the Wasm Application
on Google Chrom for Android

Check the local web server and notify
upon receiving new STD

Chrome • 192.168.50....
Welcome to F-CPS
you received new STDs

Display the analysis results of the STD,
Users can confirm
the comfort level at each location

Figure 8: Function execution screen on UE with android OS.

6.2 Evaluation of the STD-RS

We evaluated the coverage rate of the retention area by the STD-RS. Fig. 9 shows the delivery rates of STD directly transmitted from each sensor to the UE with Android OS and those relayed through the DE. At measurement points A, B, and C, both the delivery rates from sensors α and β and those from the DE exceeded 99.0%, because at these points, there is a direct line of sight between the sensors and DE. However, at measurement point D, the delivery rate from sensor α was 96.6%, and at measurement point E, the delivery rate from sensor β was 96.0%, which were lower compared to other points because walls and other obstacles were present between these points and the sensors, resulting in degraded communications. Nevertheless, through relay via the DE, the delivery rates improved to 99.0% and 98.6%, respectively, because the DE transmitted the data from a location closer to measurement points D and E. These results show that the STD-RS can distribute STD within GYMLABO and improve the delivery rate by relaying them via the DE. In this experiment, although the experimental environment was limited and had few devices, the delivery rates were already high. However, in the case of a wider retention area, we anticipate even higher delivery rates through data relay by the STD-RS.

Next, we discuss the STD retrieval time. Table 2 shows the STD retrieval time at each measurement point. Table 2 shows that UE can receive one STD packet within 10 ms, and two packets within 300 ms. Additionally, it is apparent that at measurement points D and E, where the STD delivery rate from the sensor is low, it takes 290.24 and 232.60 ms, respectively, to receive both packets, which is longer compared to reception at other points.

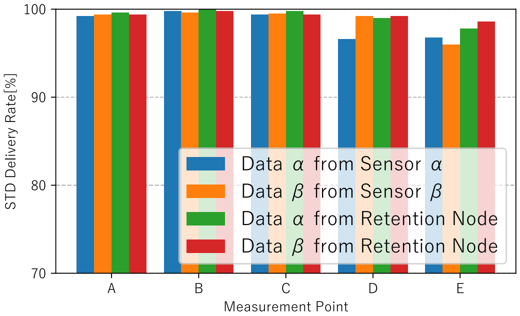


Figure 9: STD delivery rate at each measurement point.

Table 2: STD retrieval time at each measurement point.

Measurement Point	Time to Retrieve One STD [ms]	Time to Retrieve Both STD [ms]
A	8.920	182.103
B	9.052	172.283
C	8.033	221.017
D	9.052	290.244
E	7.990	232.599

6.3 Evaluation of Function Execution Using the Wasm Application

We evaluated the Wasm application created as a function in the F-CPS. The Wasm container image size created in this integrated experiment was 2.42 MB. If a similar application were created using traditional containers, the amount of data could be increased by approximately 30 times [8]. This is because the Wasm application is a program written in binary code, eliminating the need for the configuration files and libraries required by traditional containers. Therefore, we achieved a lightweight function.

Next, we evaluated the startup time of the Wasm application. Table 3 shows the startup time on Android devices. Table 3 shows that startup without caching took approximately 20.7 s. In contrast, with caching, startup took at approximately 5.3 s, resulting in a reduction of approximately 15.4 s. When UE connect to the MEC server, an authentication key is sent to fetch Wasm container information, and the authentication process is executed. In this experiment, we implemented the authentication process to ensure reliable operation verification. However, the startup time can be reduced by parameter changes and optimizations such as processing frequency and waiting time.

Table 3: Startup time of the wasm application.

Startup Time without Cache [s]	Startup Time with Cache [s]
20.678	5.274

Finally, we evaluated the battery usage percentage of the Wasm application. Fig. 10 shows the battery usage percentages measured on UE running the Android OS during the execution of the STD-RS operations and Wasm application. The battery usage percentage for Termux, a tool used to boot Linux on Android for running the STD-RS operations, and the domain name system, used to connect to the retention network, was 2.2% of the total. Additionally, the battery usage percentage for Google Chrome and Google Service, which were used for executing the Wasm application, was 27.2% of the total. During this validation, Google Chrome was running for approximately 682.8 s, consuming 1.58% of the battery. The remaining power consumption is attributed to Google Service processes running in the background to effectively support the Android OS, indicating that the impact of Chrome, where Wasm is executed, on overall power consumption is small. Thus, the consumption per second is approximately 0.002314%, indicating minimal load on UE and suggesting it is a practical application.

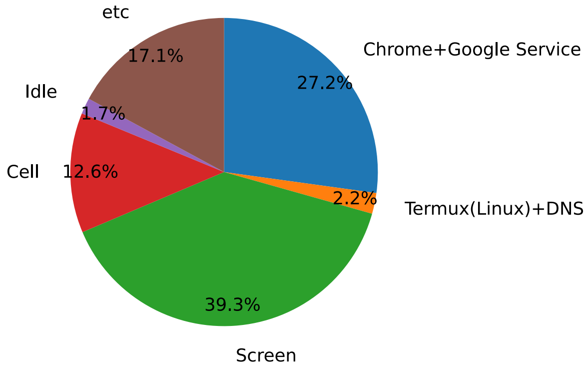


Figure 10: Battery usage percentage during function execution.

Based on the results of this integrated experiment, we successfully retained the STD obtained from sensors in the college building and executed functions on UE. By analyzing and displaying the received STD using the Wasm application, we achieved local production and consumption of data in the F-CPS platform.

7 Conclusion

In this study, we demonstrated the feasibility of local production and consumption of STD through the our F-CPS platform. Through integrated experiments conducted in a college facility, we validated three key functions of the system: (1) the retention and distribution of STD within a specific area using the STD-RS, (2) the retrieval and execution of a lightweight Wasm-based application on UE, and (3) the collaboration of heterogeneous networks, where STD dissemination was achieved via local Wi-Fi communication while application delivery was supported by Wi-Fi/B5G networks and a MEC server. The experimental results confirmed that STD could be delivered from sensors to UE within approximately 300 ms at all measurement points, and that the Wasm application could be executed efficiently with a small memory footprint and minimal battery consumption, independent of hardware and OS. Based on these results, we confirmed the practical potential of the F-CPS platform for enabling location-specific data use through local production and consumption of STD. However, several limitations should be noted. First, the integrated experiments were conducted in a relatively small environment compared to the potential wireless communication range. Therefore, further performance evaluation, including the behavior of the STD-RS in larger and more diverse environments, is required to assess scalability and robustness. Second, due to the security restrictions inherent in current Wasm execution environments, data access from the Wasm application was realized via an HTTPS server on the user device. While this approach ensures safe operation, more direct data exchange mechanisms, such as WebSocket-based communication, should be investigated in future work to improve efficiency and reduce overhead. Furthermore, as future work, we should investigate dynamic and adaptive function distribution mechanisms, including optimized transmission control and on-demand function placement, to support more complex and practical applications in real-world environments.

Acknowledgement: None.

Funding Statement: This work was supported in part by the Japan Society for the Promotion of Science under KAKENHI Grant Number 24K02916, and a research grant from Telecommunications Advancement Foundation. Furthermore, this work was partly based on the results of a project previously supported by JPJ012368C05501 from NICT, Japan. The authors would like to express their gratitude.

Author Contributions: The authors confirm their contribution to the paper as follows: Conceptualization, Daiki Nobayashi, Kazuya Tsukamoto and Takeshi Ikenaga; methodology, Meiya Tanaka and Naoki Tanaka; software, Shu Sekigawa; validation, Meiya Tanaka, Naoki Tanaka and Riku Nakamura; writing—original draft preparation, Daiki Nobayashi and Meiya Tanaka; writing—review and editing, Kazuya Tsukamoto, Takeshi Ikenaga and Myung Lee; supervision, Takeshi Ikenaga and Myung Lee. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, Daiki Nobayashi, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: Author Shu Sekigawa is full-time employees of KDDI Inc. The authors affirm that they have no additional financial or personal conflicts of interest that could have influenced the work reported in this manuscript.

References

1. Cassandras CG. Smart cities as cyber-physical social systems. Engineering. 2016;2(2):156–8. doi:10.1016/J.ENG.2016.02.012.
2. Mishra A, Jha AV, Appasani B, Ray AK, Gupta DK, Ghazali AN. Emerging technologies and design aspects of next generation cyber physical system with a smart city application perspective. Int J Syst Assur Eng Manag. 2023;13(3):699–721. doi:10.1007/s13198-021-01523-y.
3. Wang X, Zhu H, Luo X, Guan X. Data-driven-based detection and localization framework against false data injection attacks in DC microgrids. IEEE Intern Things J. 2025;12(17):36079–93. doi:10.1109/JIOT.2025.3579915.
4. Sanchez JM, Jorgensen N, Torngren M, Inam R, Berezovskyi A, Feng L. Edge computing for cyber-physical systems: a systematic mapping study emphasizing trustworthiness. ACM Trans Cyber-Phys Syst. 2022;6(3):1–28. doi:10.1145/3539662.
5. Kato T, Fukumoto N, Sasaki C, Tagami A, Nakao A. Challenges of CPS/IoT network architecture in 6G era. IEEE Access. 2024;12(1):62804–17. doi:10.1109/ACCESS.2024.3395363.
6. Nobayashi D, Akashi R, Tsukamoto K, Ikenaga T, Lee MJ. Performance evaluation of spatio-temporal data retention system supporting a floating cyber-physical system. J Inform Process. 2024;32:566–74. doi:10.2197/ipsjjip.32.566.
7. Nobayashi N, Goto I, Teshiba H, Tsukamoto K, Ikenaga T, Gerla M. Adaptive data transmission control for spatio-temporal data retention over crowds of vehicles. IEEE Trans Mob Comput. 2022;21(11):3822–35. doi:10.1109/TMC.2021.3066438.
8. Sekigawa S, Sasaki C, Tagami A. Web application-based WebAssembly container platform for extreme edge computing. In: Proceeding of the IEEE Global Communications Conference; 2023 Dec 4–8; Kuala Lumpur, Malaysia. p. 3616–21.
9. Georgios K, Safouri K, Tsaoussidis V. All about delay-tolerant networking (DTN) contributions to future internet. Future Internet. 2024;16(4):129. doi:10.3390/fi16040129.
10. Wang R, Liu X, Xi Y, Sanctis MD, Zhao K. A study of DTN for reliable data delivery from space station to ground station. IEEE J Sel Areas Commun. 2024;42(5):1344–58. doi:10.1109/JSAC.2024.3365876.
11. Hagihara R, Yamasaki Y, Ohsaki H. On delivery control for floating contents sharing with epidemic broadcasting. In: Proceeding of the 14th IEEE Annual Consumer Communications & Networking Conference (CCNC); 2017 Jan 8–11; Las Vegas, NV, USA. p. 353–6.

12. Ott J, Hyyti E, Lassila P, Kangasharju J, Santra S. Floating content for probabilistic information sharing. Pervasive Mob Comput. 2011;7(6):671–89. doi:10.1016/j.pmcj.2011.09.001.
13. Yamamoto R, Kashima A, Yamazaki T, Tanaka Y. Adaptive contents dissemination method for floating contents. In: 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall); 2019 Sep 22–25; Honolulu, HI, USA. p. 1–5.
14. Napieralla J. Considering webassembly containers for edge computing on hardware-constrained IoT devices [master's thesis]. Karlskrona, Sweden: Blekinge Institute of Technology; 2020.
15. Gadepalli PK, Peach G, Cherkasova L, Aitken R, Parmer G. Challenges and opportunities for efficient serverless computing at the edge. In: The 38th Symposium on Reliable Distributed Systems (SRDS); 2019 Oct 1–4; Lyon, France. p. 261–6.
16. Kjorveziroski V, Filiposka S. WebAssembly orchestration in the context of serverless computing. J Netw Syst Manag. 2023;31(3):62. doi:10.1007/s10922-023-09753-0.