An Efficient Event-driven Inference Approach to Support AI Applications in IOWN Era

Takeharu Eda, Ryosuke Kurebayashi, Xu Shi, Shohei Enomoto, Koji Iida, and Daisuke Hamuro

Abstract

Artificial intelligence (AI) in the Innovative Optical and Wireless Network (IOWN) era is expected to not only acquire capabilities beyond humans but also be energy-efficient, therefore contribute to the sustainability of future societies. This article describes an event-driven inference approach as a promising approach to balance AI capabilities and efficiency. This approach efficiently inspects continuous input stream data and generates events that trigger subsequent deeper inference tasks over geographically distributed computing resources only when they are truly necessary. This approach will significantly decrease energy consumption and computational and networking costs in AI inference.

Keywords: IOWN, AI, event-driven

1. Introduction

Artificial intelligence (AI) technologies such as deep learning are being applied to many commercial services worldwide and steadily progressing towards disruptive advances for real business [1]. NTT has initiated the concept of the Innovative Optical and Wireless Network (IOWN) and is developing more advanced AI-based cognitive, autonomous, and predictive systems that can recognize what humans cannot, handle much larger-scale problems than humans can do, and make decisions much quicker than humans. These systems will contribute to the establishment of a data-centric society and are expected to provide new value to society in terms of safety, accessibility, sustainability, and comfort.

2. Future AI applications in IOWN era

Figure 1 is an overview of our AI service platform. In this platform, we assume that a large number of various sensor devices (e.g., cameras, GPS (Global Positioning System), and accelerometers) are installed over a certain area (e.g., a commercial building, station, park, or even a whole city) and connected to the AI service platform. At the same time, various AI applications are deployed on the platform. The AI platform understands the physical world by analyzing data received from the sensor devices and uses the AI applications to extract intelligence from the data to meet customer needs.

Figure 2 shows a list of typical AI applications expected to be provided by the AI service platform. We focus on the two metrics described in the IOWN white paper [2], that is, cognitive capacity and response speed of the AI applications, and map them according to the metrics. Cognitive capacity indicates how precisely such applications need to capture and understand the physical world, and response speed indicates how quickly they need to give feedback to the physical world. As shown in this figure, some applications require human-level or beyond-human-level cognitive capacity and response speed (most are not possible with today’s technology). This gap is what we are trying to close through IOWN.

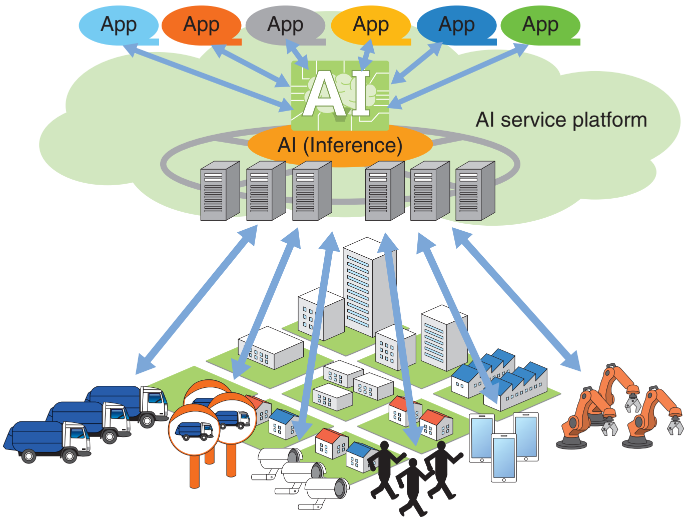


Fig. 1. Overview of our AI service platform.

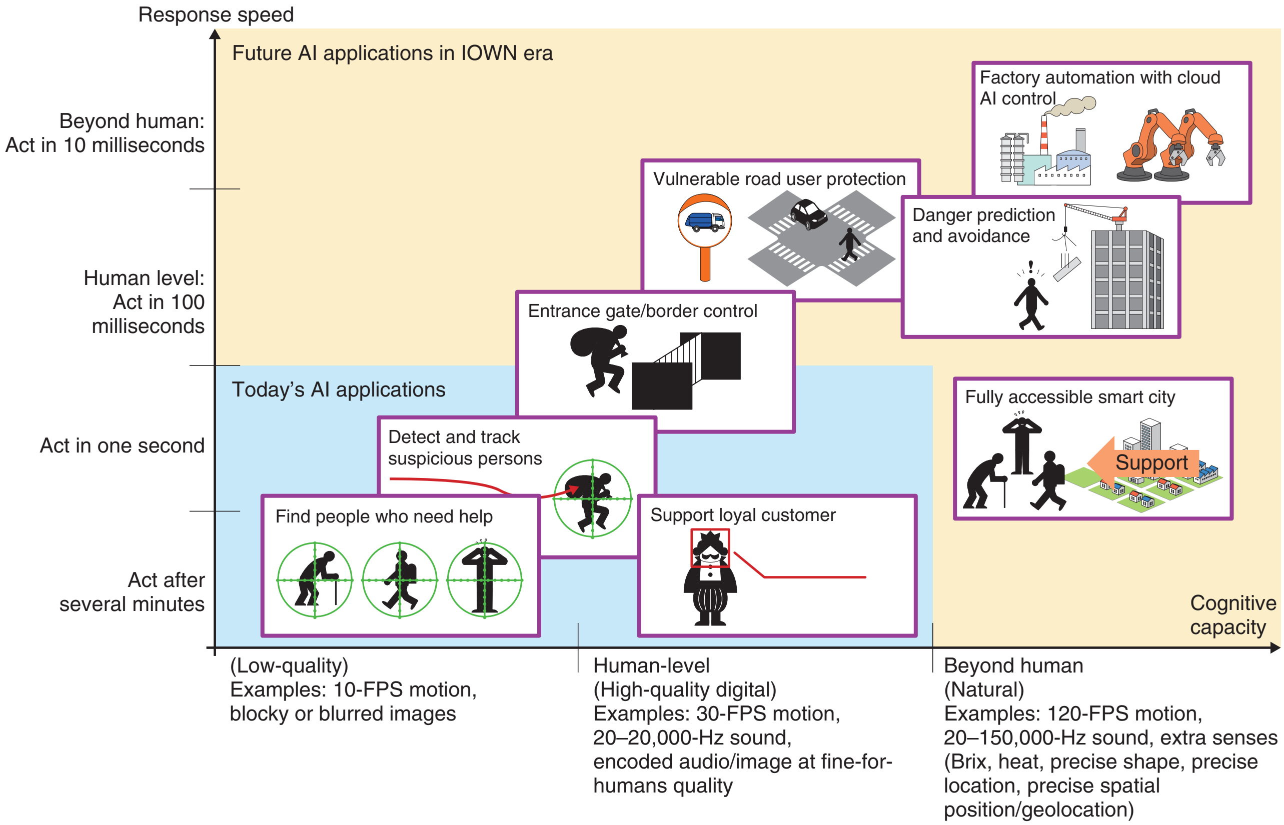

Response speed
Future AI applications in IOWN era
Beyond human:
Act in 10 milliseconds
Human level:
Act in 100
milliseconds
Act in one second
Act after
several minutes
Today's AI applications
Factory automation with cloud AI control
Vulnerable road user protection
Danger prediction and avoidance
Entrance gate/border control
Fully accessible smart city
Support
Detect and track suspicious persons
Find people who need help
Support loyal customer
Cognitive capacity
(Low-quality)
Examples: 10-FPS motion, blocky or blurred images
Human-level
(High-quality digital)
Examples: 30-FPS motion, 20–20,000-Hz sound, encoded audio/image at fine-for-humans quality
Beyond human
(Natural)
Examples: 120-FPS motion, 20–150,000-Hz sound, extra senses (Brix, heat, precise shape, precise location, precise spatial position/geolocation)

Fig. 2. List of AI applications.

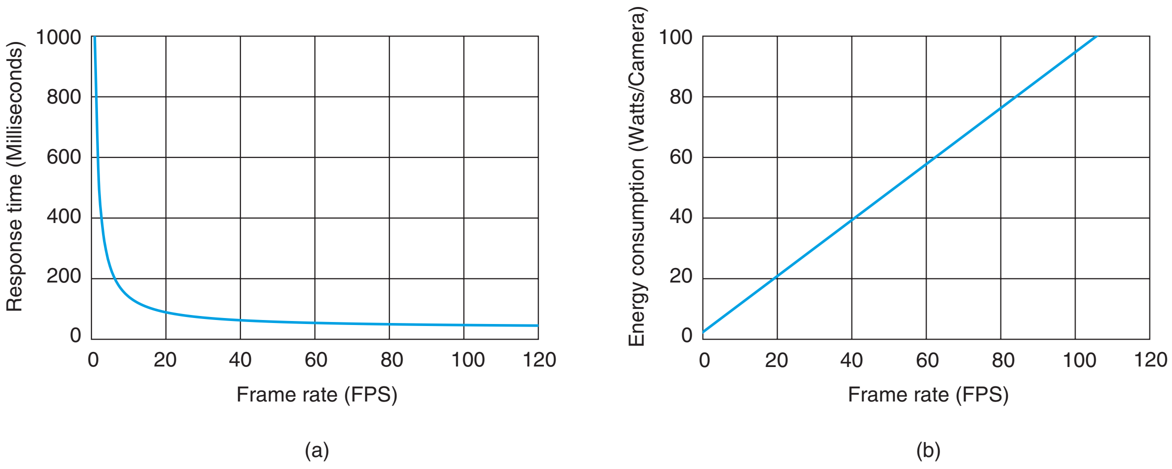


Fig. 3. Impact of frame rate on response time and energy consumption.

3. AI technologies for sustainable societies

To acquire AI capabilities beyond the human level, should we simply speed-up and scale-up AI processing? Unfortunately, while high expectations are growing given the advantages of AI, the huge energy consumption of AI is becoming a social problem [3]. In fact, improving AI capabilities often involves using more computing and networking resources, which decreases its energy efficiency. That is, there is a trade-off between AI capabilities and efficiency. Therefore, it is important to study energy-saving methods of AI in conjunction with enhancing its capabilities.

The graphs in **Fig. 3** show an example of the impact that improvements in AI capabilities may have on energy consumption. The figure shows the relationship between response time and energy consumption in AI inference processing for video images. The graphs were plotted from the results of our experiments in which we executed an inference model of Yolo v3 FP16 [4] on a server with four commercial accelerators. We also assumed a power usage effectiveness of 2. For simplicity, we focused on just inference processing and ignored network latency. Video images are generally expressed as a series of still images (or frames), and an AI inference process is applied to each frame or a set of frames. Therefore, the response time of AI inference is affected by the frame per second (FPS) of the video images as well as the processing time of AI inference. That is, a higher FPS means shorter intervals between AI inference processes, which yields better response time (see Fig. 3(a)). However, a higher FPS obviously increases the frequency of inference processing and leads to higher energy consumption per camera (see Fig. 3(b)). Let us assume that one wants to achieve 100-millisecond response time in total (i.e., human level), so one needs to keep the response time of inference processing iteration below 50 milliseconds, the energy consumption per camera will reach 45 W. This *one light bulb per camera* energy consumption is clearly not desirable from the environmental point of view. This example focuses on response time, and we can have a similar discussion with regard to the scale and complexity of AI models. Therefore, we need technical breakthroughs to break the trade-off between AI capabilities and efficiency.

4. Event-driven AI inference approach

One promising approach for breaking the trade-off is event-driven AI inference. This approach efficiently inspects continuous input stream data and generates events that trigger subsequent deeper inference tasks over geographically distributed computing resources only when they are truly necessary. This approach will significantly decrease energy consumption and computational and networking costs in AI inference.

An example of event-driven AI inference is model cascading (model splitting, early exit), where expensive but accurate AI models are split into two or more small models and the small models are properly deployed at user/edge locations to reduce computation cost, network usage, and power consumption

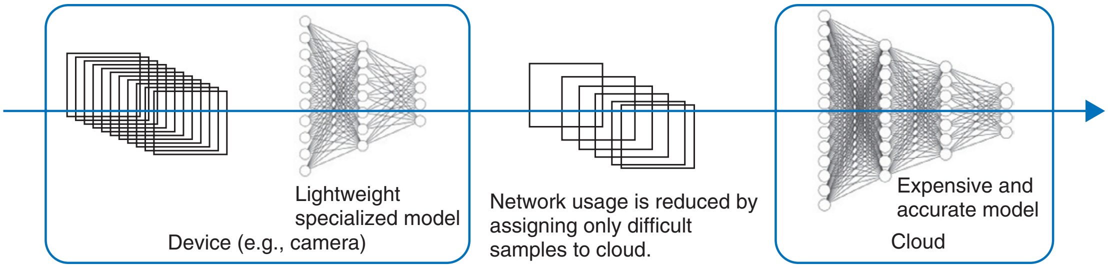

Lightweight specialized model
Device (e.g., camera)
Network usage is reduced by assigning only difficult samples to cloud.
Expensive and accurate model
Cloud

Fig. 4. Model-cascading system with camera and cloud.

without losing overall accuracy. **Figure 4** shows one example of a model-cascading system.

Since hundreds of companies have started to develop cheaper and more efficient hardware accelerators than graphics processing units, we can deploy small models on devices equipped with such AI accelerators for detecting pre-defined events and deploy the original (expensive) models on the cloud for handling difficult input samples for accurate inference provided by a large amount of computing power. AI models on devices enable security cameras to detect semantically correct events compared to naive motion detection, which is a very simple function in security cameras, and detects animals as well as humans. Such model-cascading systems reduce network usage and computation cost by using lightweight models on edge devices for efficient inference. Overall this results in both significant power savings and high capacity.

5. Learning to calibrate the confidence of lightweight models on edge devices

The role of small models on edge devices in model cascading is usually to perform the same task (i.e. detecting semantic events) with the model on the cloud, triggering decisions as to whether it sends difficult input samples to the cloud or finishes the inference within the edge devices. Since edge devices, such as cameras and home gateways, have limited computation power, they cannot afford to run powerful AI models. Thus, it is important to train lightweight and accurate models for event detection. It is desirable that the lightweight models pass only difficult input samples to the cloud. In theory, we can send such samples directly to the cloud if we know whether the lightweight model is correct, which is impossible in practice. If we send easy samples that lightweight models can correctly classify to the cloud, the network resources are wasted. The most important point in edge models is to obtain the correct confidence score of the models to input samples. Usual AI models output inference results together with confidence scores (e.g., softmax), which shows how confident the model is in its output. One recently discovered issue is that AI models tend to be overconfident, and we experimentally confirmed that the overconfidence of AI models actually negatively impacted model-cascading systems. This explains why we developed a calibration technique for model-cascading systems [5]. Our technique calibrates the confidence scores of lightweight models by taking into account the accuracy of the original AI model as well as that of the lightweight model, leading to a reduction in redundant data transfer (**Fig. 5**). Our experiments verified that this technique can reduce computation cost by 36% and data-transfer cost by 41% in model-cascading systems.

6. Model-structure sharing by early exit for further efficiency

The previously introduced calibration technique requires the model-cascading system to send the raw still images (frames) to the cloud for processing. We reduced the number of frames by using a lightweight specialized model on edge devices, but the cost incurred by edge devices in sending still images to the cloud remains significant. Thus, we implemented a more sophisticated technique based on the early exit idea where the edge and cloud models share a model structure and only compressed (quantized) feature maps are sent to the cloud to further reduce both computation and network costs (**Fig. 6**) [6]. We verified that this technique reduced network usage by 65% without losing model accuracy.

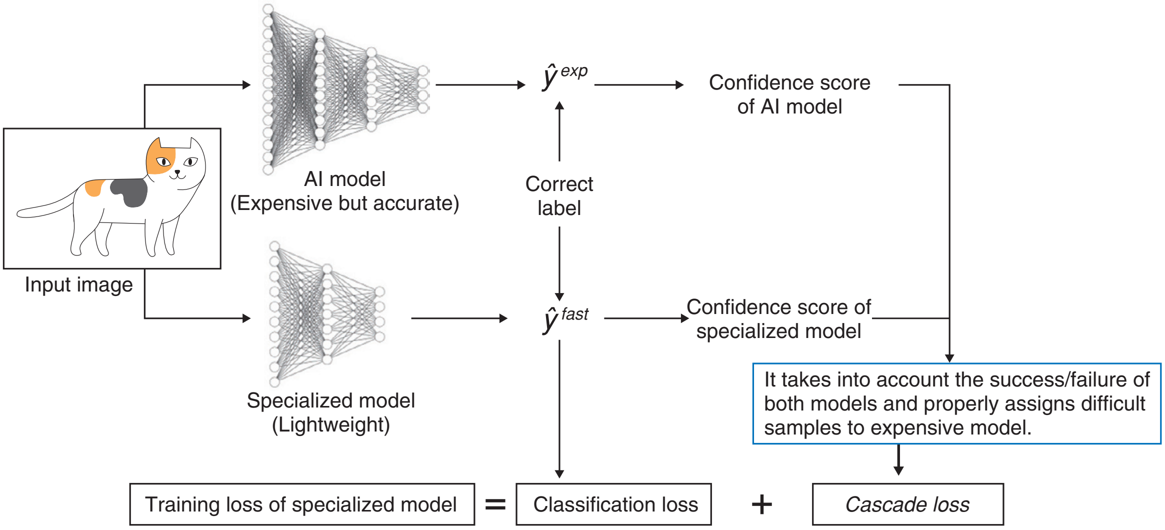

$\hat{y}^{exp}$
Confidence score of AI model
Input image
AI model (Expensive but accurate)
Correct label
$\hat{y}^{fast}$
Confidence score of specialized model
Specialized model (Lightweight)
It takes into account the success/failure of both models and properly assigns difficult samples to expensive model.
Training loss of specialized model
=
Classification loss
+
Cascade loss

Fig. 5. Training loss of specialized models for edge devices.

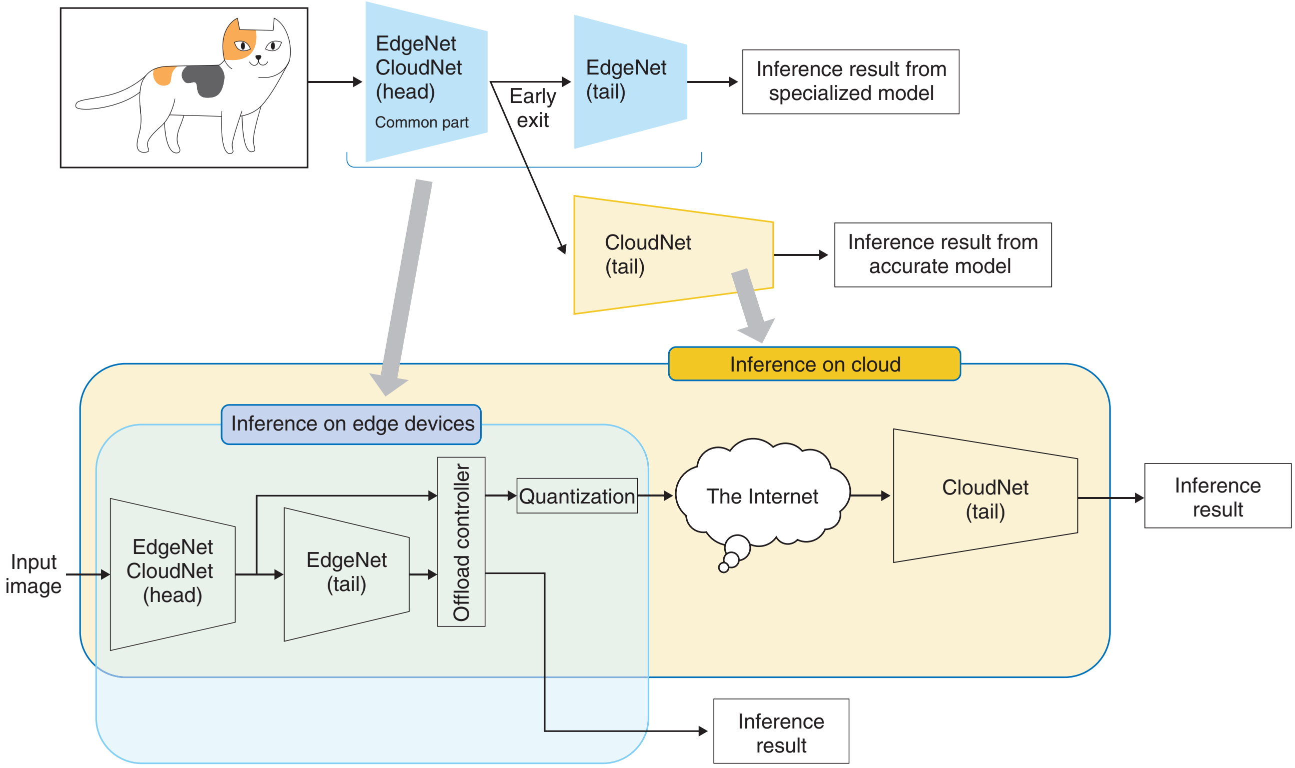


Fig. 6. Model structure sharing and feature quantization by early exit.

7. Future direction

In this article, we introduced an event-driven AI inference approach to support data analytics and provide new value to a data-centric society. By properly using the AI-based model-cascading framework, we expect our approach will reduce computation/network costs and power consumption dramatically. By exerting continuous effort towards the research and development of IOWN, we believe our approach will yield an AI infrastructure that allows applications to perform inference and reasoning at speeds beyond human ability. Such a system will contribute to resolving many social problems associated with safety, accessibility, sustainability, and comfort.

References

[1] D. Hamuro, K. Iida, K. Usami, S. Yura, Y. Matsuo, T. Eda, A. Sakamoto, M. Toyama, K. Mikami, N. Inoue, R. Nakayama, S. Enomoto, T. Sasaki, X. Shi, Y. Hirokawa, and K. Inaya, "Carrier Cloud for Deep Learning to Enable Highly Efficient Inference Processing—R&D Technologies as a Source of Competitive Power in Company Activities," NTT Technical Review, Vol. 18, No. 1, pp. 15–21, 2020. https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr202001fa2.html

[2] IOWN Global Forum, "Innovative Optical and Wireless Network Global Forum Vision 2030 and Technical Directions," 2020.

[3] K. Knight, "AI Can Do Great Things—if It Doesn't Burn the Planet," WIRED, Jan. 2020. https://www.wired.com/story/ai-great-things-burn-planet/

[4] Yolo, https://pjreddie.com/darknet/yolo/

[5] S. Enomoto and T. Eda, "Learning to Cascade: Confidence Calibration for Improving the Accuracy and Computational Cost of Cascade Inference Systems," 35th AAAI Conference on Artificial Intelligence, Feb. 2021.

[6] L. Hu, T. Wang, H. Watanabe, S. Enomoto, X. Shi, A. Sakamoto, and T. Eda, "ECNet: A Fast, Accurate, and Lightweight Edge-Cloud Network System Based on Cascading Structure," IEEE 9th Global Conference on Consumer Electronics (GCCE 2020), Kobe, Japan, Oct. 2020.

Takeharu Eda

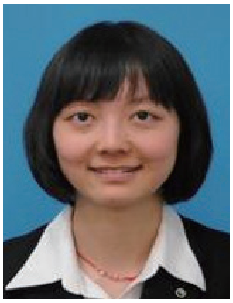
Senior Research Engineer, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

He received a B.S. in mathematics from Kyoto University in 2001 and an M.S. in engineering from Nara Institute of Science and Technology in 2003. He joined NTT in 2003 and his research interests include a wide range of topics in MLSys (machine learning and systems). He is a member of the Information Processing Society of Japan (IPSJ) and the Association for Computing Machinery (ACM).

Ryosuke Kurebayashi

Senior Research Engineer, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

He received a B.S., M.S., and Ph.D. from University of Tsukuba in 1998, 2000, and 2003. He joined NTT in 2003 and his research interests include computing and networking technologies for IoT and AI. He is a member of IPSJ and the Institute of Electronics, Information and Communication Engineers (IEICE).



Xu Shi

Engineer, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

She joined NTT Software Innovation Center in 2014. Her current research interests include deep learning and computer vision.

Shohei Enomoto

Engineer, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

He received a B.S. from Tohoku University, Miyagi, in 2014 and an M.S. from Tokyo Institute of Technology in 2016 and joined NTT Software Innovation Center in 2016. His current research interest is in deep learning.

Koji Iida

Senior Research Engineer, Supervisor, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

He received a B.E. and M.Sc. from Keio University, Kanagawa, in 1993 and 1995. He joined NTT Information Platform Laboratories in 1995 and studied enterprise communication middleware and distributed object technologies. He moved to NTT Information Sharing Platform Laboratories in 2007 and investigated identity management technology and cloud computing technology. As a result of organizational changes in July 2012, he is now with NTT Software Innovation Center.

Daisuke Hamuro

Executive Research Engineer, Distributed Data Processing Platform SE Project, NTT Software Innovation Center.

He received a B.S. and M.S. in physics from Tokyo Institute of Technology in 1992 and 1994 and joined NTT Network Service Systems Laboratories in 1994. He is a member of IEICE.