AECC

AUTOMOTIVE EDGE
COMPUTING CONSORTIUM

AECC PROOF OF CONCEPT

Traffic Load Balancing of Edge Server Access

By Toyota, Oracle, and KDDI

Version 1.1

Abstract

As the demand for AI-assisted connected vehicle services grows, the need for efficient and reliable edge computing solutions becomes increasingly critical. Traditional methods of handling massive computational loads are strained by the diverse and intensive service requirements of modern edge AI applications. However, when a vehicle is in operation, any essential AI or other connected vehicle services simply can't be allowed to fail, as safety may be impacted.

To address these challenges, this proof of concept (PoC) explores a groundbreaking method for dynamically distributing traffic to edge servers based on real-time server load, optimizing performance and resource usage both within and beyond mobile networks.

This PoC implements two main approaches to achieve traffic distribution: one focuses on it across wide area networks (WAN), and the other focuses on it within mobile networks. These approaches are designed to enhance the scalability and efficiency of edge AI applications in the automotive industry. The PoC also utilizes the concepts of distributed computing architecture, edge data offloading, access network selection, and opportunistic data transfer in WG2.

LISP (Locator/ID Separation Protocol) is used to distribute traffic across WAN. It is a network technology that simplifies and enhances internet routing by separating location and identity information in IP addresses. This improves scalability and allows for more flexible address management, making it easier to support multiple connections, mobility, and virtualization, while also streamlining operations for future internet developments. It was created to address the exponential growth expansion in the number of connected devices.

For traffic distribution within mobile networks, this PoC utilizes the Uplink Classifier (ULCL). The ULCL classifies uplink data (data from user devices to the network) in real-time, optimizing the prioritization and processing of data dynamically, ensuring that critical traffic is allocated to essential applications and services. The control of ULCL is managed using the CAMARA API. CAMARA is a Linux Foundation open source project that provides a standardized interface for telecom operators and cloud service providers to integrate various services and functions within edge computing environments and 5G networks through APIs. This enables developers to uniformly access different network providers and cloud services, simplifying application development and deployment.

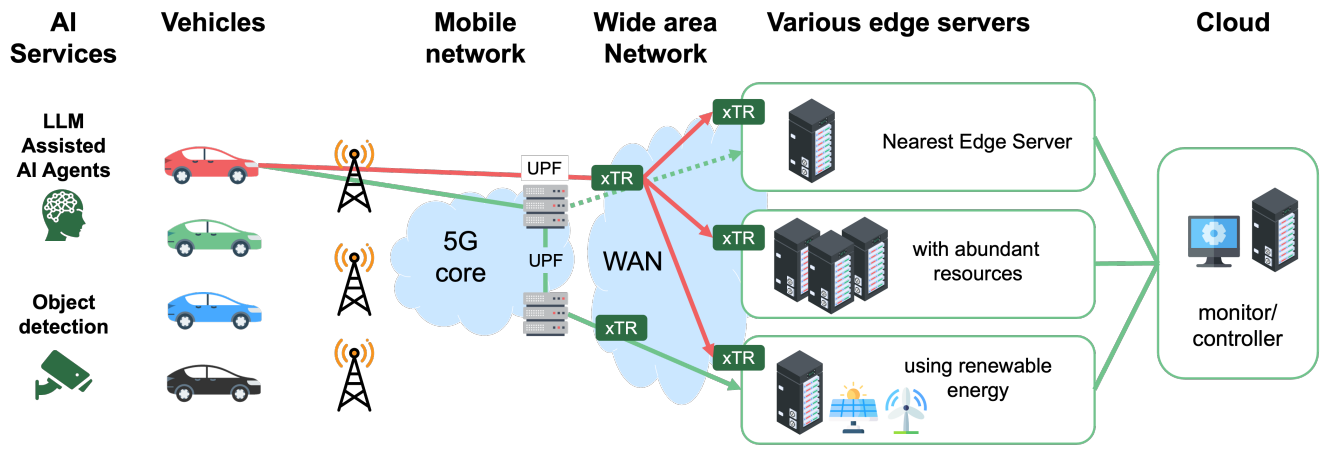

Figure 1 A technical overview of equipment and workflow.

Developed collaboratively by Toyota, Oracle, and KDDI, this PoC effectively utilizes computational resources — such as server central processing unit (CPU) and graphics processing unit (GPU) usage — by distributing traffic

across geographically dispersed servers. By integrating advanced techniques like real-time image recognition and large language model inference, the PoC demonstrates the potential to significantly improve the performance and energy efficiency of AI-driven connected vehicle services at the network edge.

For the demonstration, four servers were available in different locations in Japan. They ran two AI applications: a general-purpose object detection application designed to simulate the use of object detection technology for driver assistance, and a learning language model (LLM)-assisted AI chat agent that can help with wayfinding.

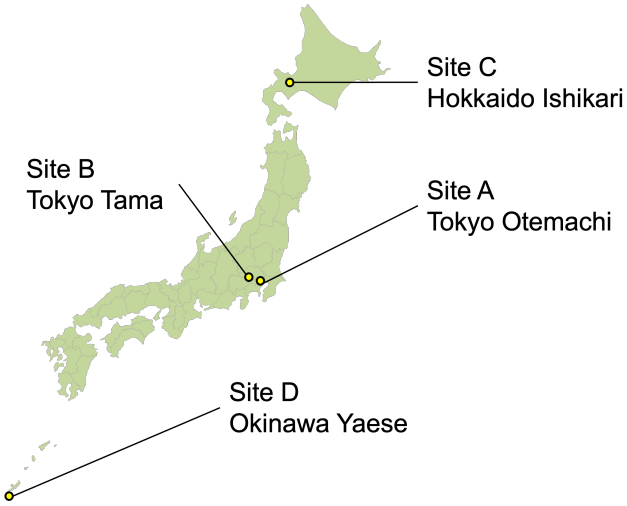

Figure 2 Locations of the four central servers used in this PoC.

Figure 3 A KDDI container data center at the KDDI site in Okinawa.

As traffic volume increased, CPU and GPU usage increased in the closest server and the traffic was distributed to other edge servers.

The results of this proof of concept confirm that traffic can be effectively directed to the ideal edge servers, both within and outside the mobile network. This approach is crucial for achieving the service stability needed for AI-driven and other connected vehicle services.

Business Strategy

The use of the architecture outlined in this PoC offers practical business applications and benefits for different types of organizations.

- **For Mobility Service Providers:** Ensuring stable mobility service delivery by appropriately distributing traffic across multiple edge servers, regardless of fluctuations in edge service demand. This increases the range of options for service deployment through effective traffic distribution using both WAN traffic distribution and mobile network traffic distribution.
- **For Mobile Network Operators:** A mobile network service implementation that allows MNOs to select an appropriate breakout point for each of the user PDU sessions by using CAMARA-API and ULCL.
- **For Edge Providers:** An appropriate edge selection service implementation using LISP over WAN. A further benefit is the understanding of how edge infrastructure can be used by the automotive industry and the design of AI workloads, which is vital for edge providers to develop attractive edge infrastructure solutions.

Proof of Concept Objective

The primary objective of this PoC is to verify the effectiveness of dynamically distributing traffic to edge servers based on real-time server load. Specifically, the PoC aims to confirm that two distinct approaches for traffic distribution are functioning correctly: using LISP over WAN and employing ULCL managed via the CAMARA API within mobile networks. By demonstrating the proper operation of these methods, the PoC seeks to prove their viability for efficient resource utilization and service stability in AI-driven connected vehicle applications. (For the efficient use of network resources, see AECC's Opportunistic Data Transfer proof of concept.)

Proof of Concept Scenario

This PoC included two communication routes:

1. In the mobile network demonstration, the optimal server was selected using ULCL, which diverts traffic to available data centers, and the CAMARA API. When the controller detected congestion, user equipment (UE) routing rules were changed in the mobile network.
2. In the WAN communication route, the optimal server was selected using LISP according to load. When the controller detected congestion, the distribution ratio of each tunnel router (xTR) was changed. For the WAN, a private network prepared by Toyota was used for this PoC.

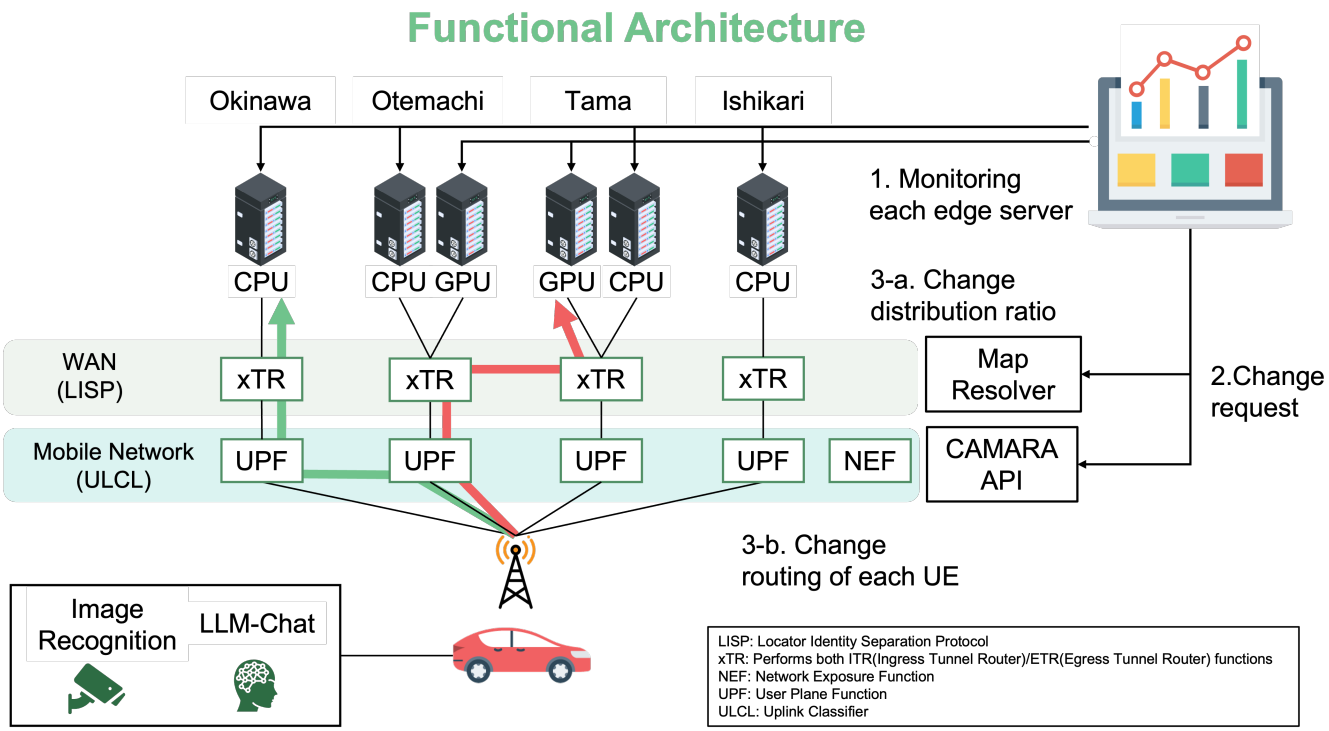

Figure 4 The functional architecture of the PoC.

Process

Each demonstration saw the following sequence of events.

1. Monitoring Each Edge Server

The servicer monitored the CPU and GPU usage of each server site in real time. Typically, traffic from each terminal was sent to the geographically closest server, which was assumed to have the lowest communication latency. For example, terminals near Otemachi sent traffic to the Otemachi server.

2. Change Request

As traffic increased in a specific area and the servicer detected a rise in CPU or GPU usage, the servicer sent a request to change the communication route to distribute the traffic to other servers, again, prioritizing closer servers. For instance, as traffic was concentrated in Otemachi, it was redistributed to Tama, as both were located in Tokyo.

When changing the communication route over the WAN, a request was sent to the LISP MapResolver, and when changing the route within the mobile network, a request was sent to the CAMARA API. In the future, when implementing this system, the decision of whether to send the request to the MapResolver or the CAMARA API will depend on the deployment status of xTR and user plane function (UPF), the business model of the service, and the network configuration at that time.

3. a. When changing the communication route over the WAN, the distribution amount to each site was sent to each xTR via the MapResolver, and the traffic was distributed accordingly.

3. b. When changing the communication route over the mobile network, the communication routes of each terminal were changed via the CAMARA API, and the traffic was distributed accordingly. If the CPU and GPU

usage of multiple sites increased, the traffic was further distributed to other servers. Additionally, if the traffic decreased to normal levels, the traffic from each terminal was reallocated to the geographically closest server.

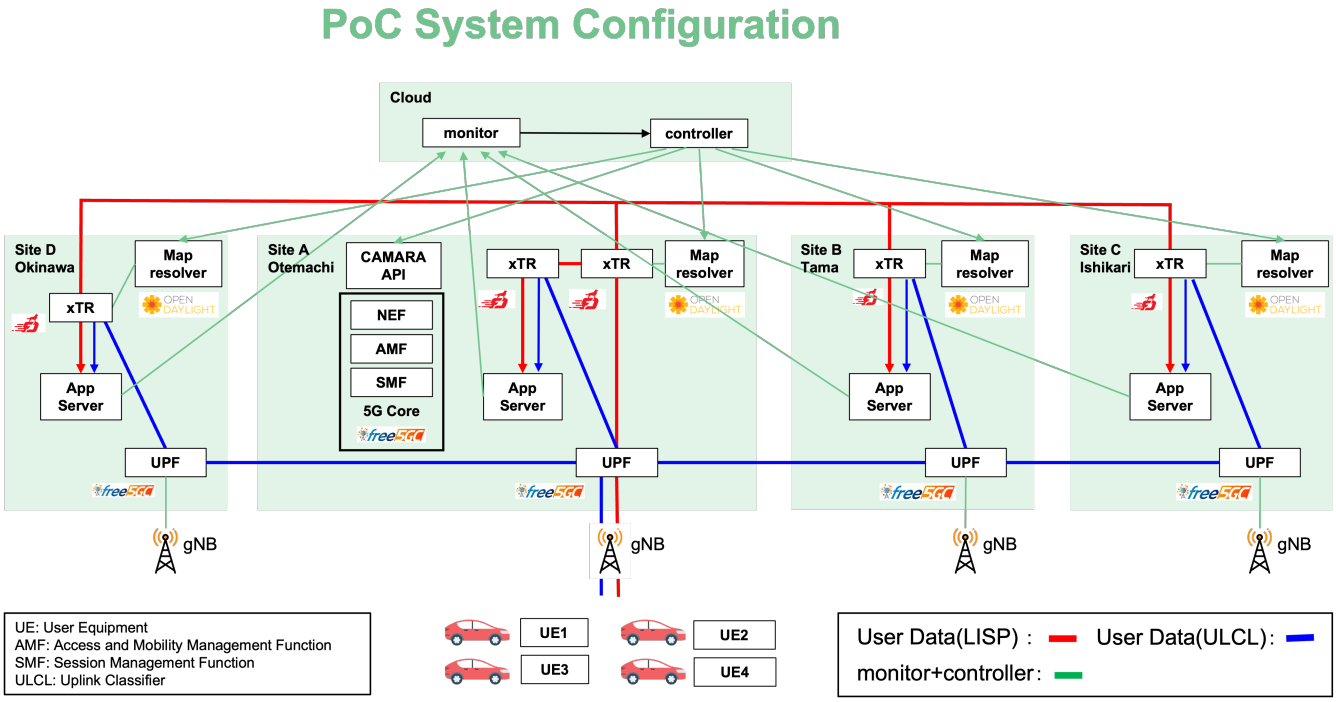

Figure 5 The system configuration.

Roles

Toyota investigated, designed, and implemented the distributed architecture using LISP on WAN. Additionally, Toyota provided servers at three locations other than the edge center in Okinawa (which was provided by KDDI) and also supplied all the evaluation applications used in this PoC. Oracle supported the overall PoC architecture and scenario development. KDDI, besides providing the edge data center environment, also worked on the distributed architecture using CAMARA-API within mobile networks.

Proof of Concept Results

Figure 6 shows the elapsed time and CPU usage of each site when using LISP for distributing the traffic in the object detection application. Initially, all traffic is distributed to Site A (Otemachi). However, as the number of application sessions increases and the CPU usage at the site rises, traffic is redistributed to other sites, leading to an increase in CPU usage at those sites. The CPU usage at each site was adjusted to ensure it did not exceed the threshold of 40%.

Figure 7 shows the average response time for object detection using LISP, from sending the request to receiving the results. It was confirmed that the average response time decreased when using this system (blue line) compared to not using it (orange line).

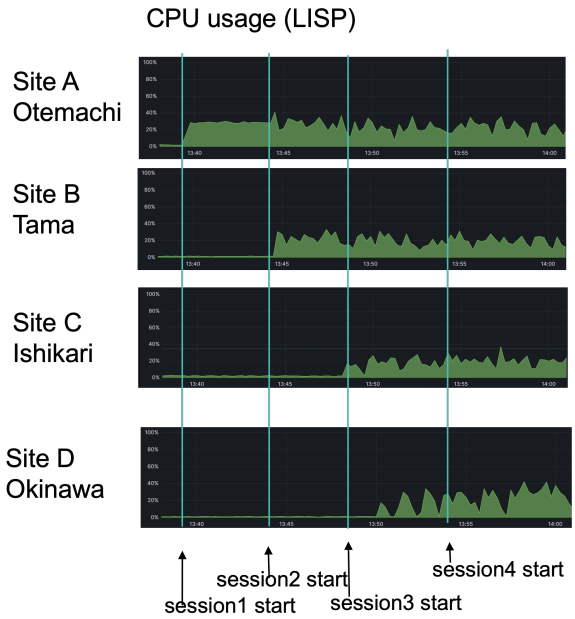

Figure 6 CPU usage for each server, showing CPU usage kept below 40% for optimal performance.

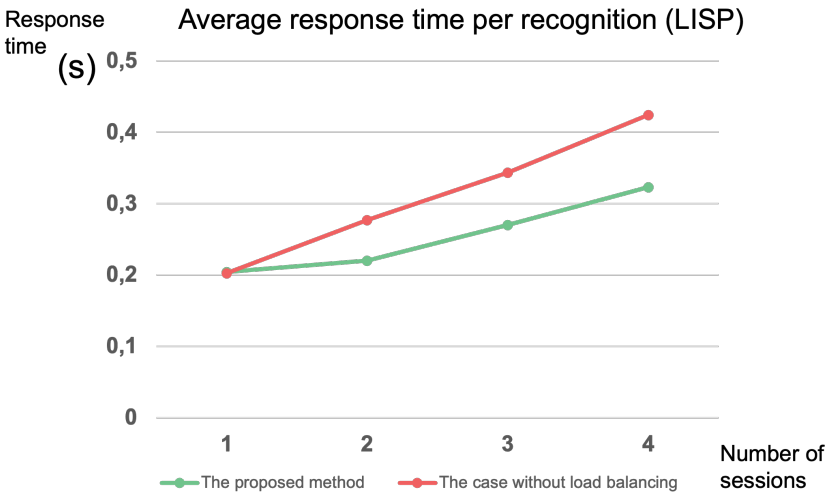

Figure 7 The PoC architecture has a shorter response time than without the load balancing capability.

Next Steps

To further develop this architecture, the engineering team is considering how to realize a seamless transition of sessions and applications between edge servers and prioritizing sites utilizing renewable energy. Although this PoC does not directly reduce energy consumption, prioritizing servers using renewable energy can enhance the usage rate of these more climate-friendly energy sources.