AECC

AUTOMOTIVE EDGE
COMPUTING CONSORTIUM

AECC Industry Blueprint: Shaping the Future of Automotive Innovation

Version 1.0 · February 2026

Contents

Executive Summary

The future automotive industry demands massive data collection, enhanced real-time processing such as sub-second latency AI inference, high availability, and large-scale data analytics for training machine learning and AI models at petabyte to exabyte scales. Traditional centralized clouds face challenges with data transfer loads, latency, reliability, and environmental impact, making them inadequate for these needs.

To address this, the Automotive Edge Computing Consortium (AECC) promotes widespread adoption of distributed edge computing architectures. By decentralizing data processing across geographically dispersed edge infrastructures, latency is minimized, reliability is improved through redundancy, and energy efficiency is increased by leveraging local renewable energy. However, significant gaps remain in building a fully integrated edge ecosystem supporting diverse automotive use cases.

This Industry Blueprint provides comprehensive guidance for adopting edge computing in the automotive sector. It clarifies the design principles and solutions for a holistic, end-to-end distributed architecture that integrates networking, computing, and data management. Key use cases include intelligent driving, teleoperated driving, high-definition mapping, voice-interactive AI agents, digital twins, green mobility, and data-driven development platforms.

The blueprint maps key challenges—data collection burden, big data processing, real-time responsiveness, reliability, environmental efficiency, and system configurability—to these use cases, highlighting priority areas for solutions. It also analyzes the limitations of existing technologies and stresses the need for innovations such as multi-carrier redundant networks, quality-assured network slicing, distributed AI/ML processing, and integrated API designs.

Intended readers are broad industry stakeholders involved in automotive technology strategy and system design, including original equipment manufacturers (OEMs), mobile network operators (MNOs), edge and cloud service providers, technology developers, and system integrators. Serving as a concise knowledge base and strategic guide, this document aims to help readers clearly understand foundational principles and provide practical guidance for effective implementation and use.

How to Use This Blueprint

Scope

This blueprint offers a practical framework for deploying distributed edge computing in the automotive industry. It provides actionable guidance to address critical challenges and includes:

- Detailed system requirements
- A reference architecture with defined actor roles
- Actor-specific implementation patterns
- Validation insights from proofs of concept

The document serves as a strategic guide to support solution development. It is not a technical standard or vendor-specific manual.

What This Is and Is Not

What This Is:

- A holistic end-to-end blueprint outlining architectural principles and solution approaches for automotive edge computing.
- A consolidated knowledge base linking business use cases to technical challenges and design considerations.
- A guide to accelerate the adoption of distributed edge architectures with practical implementation insights.

What This Is Not:

- A formal standards document or vendor-specific implementation manual.
- A marketing or product brochure.
- A static text; it will evolve based on ongoing and continuous research and PoC outcomes.

Navigation

The blueprint is organized into five main chapters for targeted use:

1. **Key Challenges and Design Principles**
   Summarize core challenges (e.g., data collection burden, reliability, green infrastructure) and foundational design principles shaping the AECC architecture.
2. **System Requirements**
   Defines mandatory and recommended requirements across connectivity, computation, data management, and security domains. Annex A includes detailed technology catalogs.
3. **Reference Architecture and Actor Responsibilities**
   Presents a layered, end-to-end architecture that integrates physical infrastructure and control and user plane functions. Clarifies roles and responsibilities of key actors, including OEMs, Tier-1 suppliers, MNOs, Edge/Cloud providers, and Mobility Service Providers (MSPs). Interfaces and system boundaries are described.
4. **Implementation Recipes**
   Provides actor-specific deployment patterns, example topologies, KPI targets, and common pitfalls. Includes a sample data flow for the high-definition map updates, representative of an AECC-enabled service scenario.
5. **Validation and Key Learnings**
   Summarizes insights from proof-of-concept activities to guide future development. Full PoC details appear in Annex B.

Additional annexes include detailed analyses of existing solutions and gaps, a glossary, and references for further study.

Users are encouraged to navigate the blueprint iteratively, focusing on relevant chapters while cross-referencing for a comprehensive understanding. This approach complements the Executive Summary and emphasizes practical guidance for effective use.

1. Key Challenges and Architecture Principles

The rapid growth of connected and autonomous vehicles is driving an unprecedented surge in data volume, distribution, and processing demands far beyond the capabilities of traditional centralized hyperscale cloud architectures. As fleets and services scale, meeting requirements for **real-time responsiveness, reliability, cost efficiency**, and **sustainability** becomes increasingly complex.

To overcome these limitations, the AECC advocates a **distributed-edge approach** enabling data to be processed and managed at the most appropriate location, closer to vehicles and aligned with the characteristics of each data type.

This chapter highlights the key challenges and defines the design principles essential for enabling future large-scale automotive edge systems.

1.1. Key Challenges

Several major challenges must be addressed to enable AECC use cases as envisioned in the *AECC General Principle and Vision White Paper [1]*. These include:

- **Data Transfer Burden –** Managing the massive volume of data generated by connected and autonomous vehicles.
- **Big Data Processing –** Handling complex analytics and storage requirements at scale.
- **Real-Time Capability –** Ensuring ultra-low latency for safety-critical and time-sensitive applications.
- **Reliability –** Maintaining consistent performance across diverse environments and network conditions.
- **Efficient and Green Infrastructure –** Designing energy-efficient systems that minimize environmental impact.
- **Configurability and Manageability –** Supporting flexible deployment and simplified operations across heterogeneous ecosystems.

1.1.1. Data Collection Burden

Collecting large amounts of data from vehicles over long distances significantly increases costs —an inherent limitation of today's centralized hyperscale cloud architectures. Even with limited connected car services, network expenses are already substantial. As data requirements for teleoperation and autonomous driving model training grow, reducing data transfer costs becomes critical.

The current practice of uniformly collecting data from all vehicles adds further complexity. Ideally, only relevant information should be collected. However, this approach persists because solutions based on centralized cloud models limit the ability to make timely, context-specific decisions or deliver precise, context-specific instructions to vehicles on which data to collect.

Additionally, applications such as training autonomous driving models demand continuous, high-quality data capture. Meeting these needs would require network carriers to increase capacity during peak hours, significantly driving up operational costs.

Edge computing offers a solution by processing data closer to vehicles, enabling selective, timely collection of only the necessary information—reducing both network load and cost.

1.1.2. Big Data Processing

Processing the massive volumes of data generated by the automotive industry is a significant challenge. In today's centralized hyperscale cloud data center (DC) environments, these datasets are often too large to handle efficiently. Key limitations include:

- **Centralized Storage Without Preprocessing**
  Data is typically stored without prior sorting or filtering, reducing processing efficiency.
- **Resource Mismatch**
  Computing resources such as GPUs are often located far from the data, making timely processing difficult.

AECC's Distributed Architecture Solution

To address these issues, the AECC proposes a distributed architecture designed to deliver the following capabilities:

- **Primary Data Processing Near the Source**
  Sorting, filtering, and aggregation performed close to where the data is generated.
- **Utilization and Management of Distributed Data**
  Efficient handling and coordination of data across multiple locations.
- **Hierarchical Distributed Computing**
  Multi-level processing architecture for scalability and efficiency.
- **Leveraging Distributed Computing Resources**
  Optimal use of computing power across decentralized environments.

1.1.3. Real-time Data Processing

Transferring data over long distances introduces latency, complicating real-time processing. For example, sending just a few kilobytes over 1,000 km typically adds **10–20 milliseconds** of delay. While this may seem acceptable, additional factors—such as jitter and packet loss—can extend delays to **hundreds of milliseconds**, or even **several seconds** in worst-case scenarios. Such latency makes it challenging to support use cases requiring real-time performance, for instance, digital twin use cases described in the AECC Digital Twin White Paper [2].

Why Centralized Clouds Fall Short
Current hyperscale cloud architectures prioritize scale-out above all else, resulting in:

- Massive resource pools connected through multi-tiered internal networks with significant overhead
- Complexity built on numerous small databases, queues, minibatch jobs, microservices, and slow object storage
- Variable performance of network, compute, and storage resources, fluctuating with workload and congestion
- These characteristics make real-time data processing at scale impractical in centralized environments.

Latency challenges can be mitigated by developing **new technical specifications** leveraging edge infrastructure and localized networks. Adopting **distributed computing platforms** with design patterns and technologies tailored for real-time performance.

1.1.4. Reliability

Centralized hyperscale cloud-based solutions face significant reliability issues:

1. Vulnerable Communication Paths
Vehicle-to-cloud communication typically traverses the access network and the Internet. Network outages already occur today, and the growing volume of vehicle data will further increase the risk of failures. Moreover, access networks rarely provide redundant connectivity through multiple operators, resulting in insufficient reliability.

2. Single Points of Failure at Cloud Entry
Hyperscale clouds are designed to serve massive numbers of users from centralized locations. This concentration can lead to traffic congestion and make entry points prime targets for external attacks, such as Distributed Denial of Service (DDoS).

3. Internal Cloud Failures
Reliability issues also exist within the cloud itself. Failures can impact not only individual services but entire availability zones or regions. Large resource pools do not inherently guarantee high reliability.

To overcome these challenges, a **distributed edge and multi-access architecture** is needed—eliminating single points of failure and enabling rapid failover for uninterrupted service.

1.1.5. Efficient and Green Infrastructure

Today's centralized hyperscale cloud data centers prioritize scalability over efficiency and real-time performance. These facilities are already enormous, and building even larger ones is increasingly unrealistic due to excessive power consumption and the industry's shift toward sustainability. In fact, some governments may soon restrict or prohibit the construction of such large-scale data centers.

Centralized hyperscale cloud solutions cannot fully meet AECC requirements, especially for green vision described in the AECC Connected Infrastructure for the Realization of the Green Mobility Society White Paper [3] and use cases described in the AECC Data Management Systems White in the Distributed Environment White Paper [4], which demand unprecedented levels of real-time and large-scale data processing.

AECC advocates for **edge-based distributed architectures** composed of multiple regional data centers or infrastructures. Each center would provide cloud-like capabilities while enabling efficient geo-distributed processing—even though these facilities may be much smaller than today's hyperscale data centers.

A key advantage is that regional data centers can leverage **locally generated renewable energy**, supporting both **performance** and **sustainability goals**.

1.1.6. Configurability and Manageability

Implementing emerging automotive use cases for tens of millions of vehicles is highly complex. The system must provide **extensive configurability and manageability** to:

- Handle vast amounts of data under fluctuating traffic conditions
- Deploy new data-driven services seamlessly
- Monitor operational status in real time
- Enable rapid configuration changes to maintain stability

At first glance, centralized cloud solutions may seem better suited for this task than distributed edge architectures. However, this is not the case. Future automotive scenarios require **joint configuration and management of networking and computing resources**—something current hyperscale clouds cannot deliver. Key limitations include:

- **Limited Connectivity Options**
  Hyperscale clouds offer only a few dedicated line connections per tenant and do not support dozens of dynamic connections from multiple access network breakout points.
- **Basic Load Balancing**
  Current systems rely on simple mechanisms and lack position-based load balancing, both of which are essential for AECC scenarios.

- **Restricted Real-Time Configuration**
  Centralized clouds impose limits on the frequency of configuration changes (e.g., scaling) and on real-time failure detection, making them unable to meet elasticity and reliability demands.

Adopting distributed edge architectures is critical—not only to address other major challenges but also to achieve a level of configurability and manageability that surpasses that of centralized hyperscale clouds.

1.1.7. Interoperability and Ecosystem Fragment

The rapid evolution of mobility services involves a diverse set of stakeholders—OEMs, network operators, cloud/edge providers, and application developers—each with their own platforms and standards. Without open and interoperable solutions, the risk of ecosystem fragmentation increases, leading to integration complexity and limited scalability across regions and operators. Achieving seamless interoperability is essential to enable cross-industry collaboration, accelerate innovation, and ensure that mobility services can scale globally.

1.2. Architecture Principles

The AECC architecture adopts an **end-to-end distributed-edge framework** that seamlessly integrates vehicles, edge nodes, and cloud systems. Its core principle is to **process and manage data as close to its source as possible**, reducing transfer costs, latency, and energy consumption while ensuring **real-time responsiveness and reliability**.

Hierarchical orchestration across in-vehicle, edge, and cloud layers enables scalable, resilient operations and localized intelligence—allowing services to continue even during network or cloud disruptions.

Equally critical are **openness, interoperability, and secure automation**:

- **Standardized APIs** foster collaboration among diverse stakeholders, ensure seamless integration, and prevent ecosystem fragmentation.
- **Zero-trust security and policy-based governance** safeguard sensitive mobility data.
- **Automated, intent-driven orchestration** ensures elasticity and stability for large-scale mobility services.

With **energy-aware workload placement** and **federated control**, AECC delivers a **future-ready, sustainable platform** designed to evolve alongside advancements in **6G**, **AI acceleration**, and **green computing**.

2. System Requirements

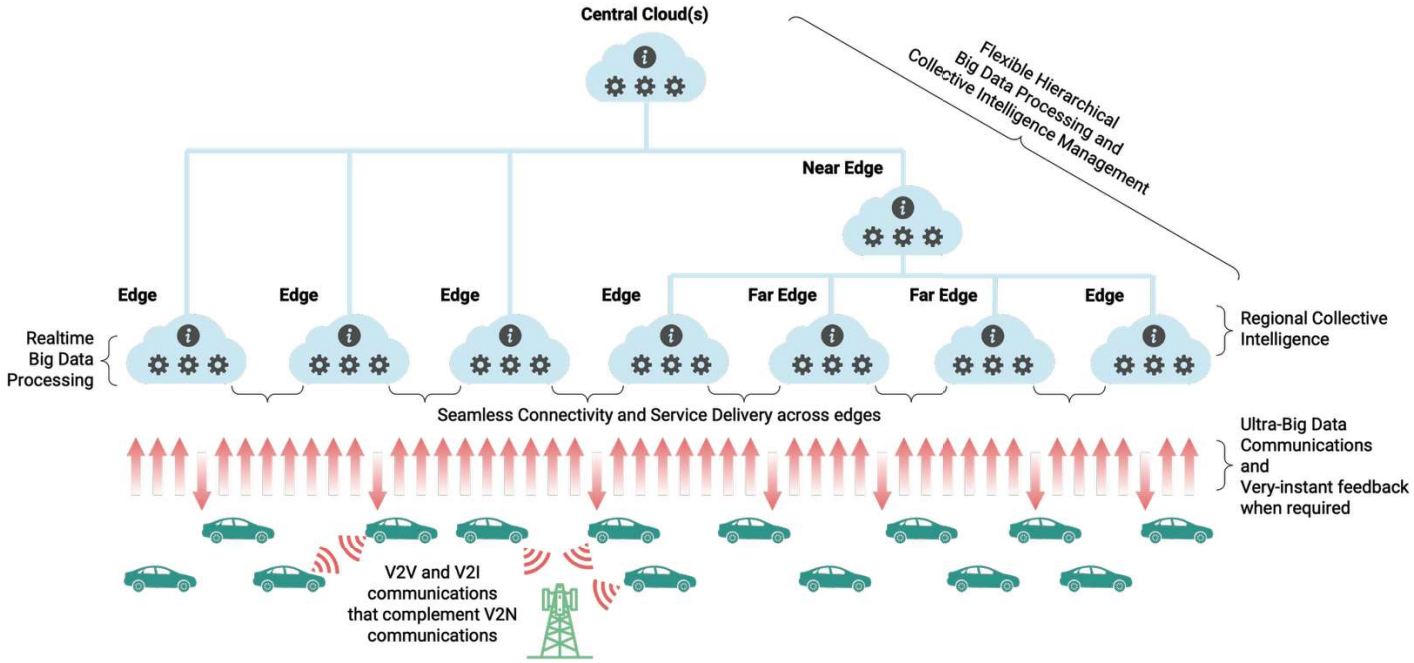

Figure 2.1. General architecture of automotive edge computing

Figure 2.1 illustrates the general architecture of automotive edge computing, highlighting how future mobility services rely on flexible, hierarchical big-data processing, real-time analytics, and collective-intelligence management distributed across the in-vehicle, edge, and cloud layers. As connected and autonomous vehicles generate massive, highly dynamic data flows, the system must support ultra-large-scale data communications, seamless connectivity and service delivery across edge nodes, and real-time regional insight generation, while enabling scalable, secure processing across heterogeneous infrastructures. These capabilities form the foundation of the system requirements for large-scale automotive edge computing. Sections 2.1–2.4 detail these requirements in terms of connectivity, computation, data management, and security, outlining the essential functions that underpin the AECC's distributed edge architecture.

2.1. Connectivity

Mandatory:

- **Support ultra-large-data communications** between vehicles, edge nodes, and clouds without congestion.
- **Enable controlled session breakout** to offload traffic to the appropriate edge based on vehicle positions, edge load, and other metrics.
- **Ensure seamless connectivity across edges**, including low-latency handover between edge nodes.
- **Provide multi-layer traffic forwarding** to enable communication paths to adapt to workload relocation.
- **Maintain consistent service** through data replication and timely migration of vehicle session data.

Recommended:

- **Optimize data transfer** by considering urgency, data size, and network congestion forecasts.
- **Enhance protocol efficiency,** for example, by tuning Transport Layer Security (TLS), optimizing congestion control, and using UDP-based protocols.

- **Improve network scalability** by supplementing Vehicle-to-Network (V2N) communications with Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications.
- **Support key and session information transfer during edge switching** to further reduce handover latency.

2.2. Computation

Mandatory:

- Provide **real-time big data processing** for hundreds of thousands to millions of vehicles per edge.
- Use a **high-speed, scalable message queue** capable of handling fluctuating data inflow.
- Support **continuous (as opposed to batch) processing** for latency-sensitive workloads.
- Enable **regional collective intelligence processing** at edge nodes.

Recommended:

- Employ **fast and scalable regional data management**, integrating vehicle and environmental data.
- Provide sufficient **heterogeneous compute resources** (Central Processing Unit (CPU), GPU, Field Programmable Gate Array (FPGA)) for dynamic scaling.
- Allow **workload relocation and rebalancing** across edges and cloud based on resource availability.

2.3. Data Management

Mandatory:

- **Support hierarchical, spatial partitioning and indexing** to manage diverse and location-specific insights efficiently.
- **Enable progressive time aggregation**, retaining detailed data for short-term and aggregated data for long-term analysis.
- **Provide an intelligent orchestrator** to distribute data-processing tasks across edge and cloud resources.
- **Ensure integrated data access** across distributed storage systems without unnecessary duplication.

Recommended:

- **Use a high-speed, scalable data platform** optimized for distributed edge–cloud distributed environments.
- **Support wide-area insight fusion**, combining regional insights into comprehensive analytics.
- **Enable efficient data locality** to ensure computation occurs where the data resides.
- **Maintain interoperability across multi-vendor environments** through standardization and abstraction layers.

2.4. Security

Mandatory:

- **Support secure, low-latency session establishment** across edge nodes, even under distributed processing conditions.
- **Ensure secure transmission of vehicle, driver, and environmental data** during handover and replication processes.
- **Provide secure orchestration and distributed system management** across multiple vendors and layers.
- **Protect privacy and sensitive data from unauthorized access and use**, as well as unauthorized cross-border transfer

Recommended:

- **Use fine-tuned encrypted communication protocols** to minimize handshake latency and enable faster secure connections.
- **Enable secure sharing of session keys and metadata** across adjacent edge nodes to ensure continuity during handovers.
- **Provide robust access control and policy enforcement** across distributed data stores and compute nodes.

2.5. Manageability, Configurability, and Interoperability

Mandatory:

- **Ensure seamless integration of systems, platforms, and services** across multiple vendors, operators, and regions.
- **Support standardized interfaces and APIs** to enable cross-domain data exchange and service orchestration.
- **Facilitate compatibility between legacy systems and emerging technologies** to protect investments and accelerate adoption.

Recommended:

- Promote open standards and collaborative frameworks to minimize ecosystem fragmentation.
- Enable dynamic onboarding and interworking of new partners, devices, and applications with minimal integration effort.
- Manage related edge infrastructures and networks together

3. Reference Architecture and Actor Responsibilities

3.1. Physical Infrastructure

In the AECC System, all functions operate on top of a distributed physical infrastructure. The diagram below illustrates this architecture, showing multiple layers of interconnected components—including vehicles, edge nodes, regional data centers, and cloud platforms—working together to enable scalable, real-time processing and resilient service delivery.

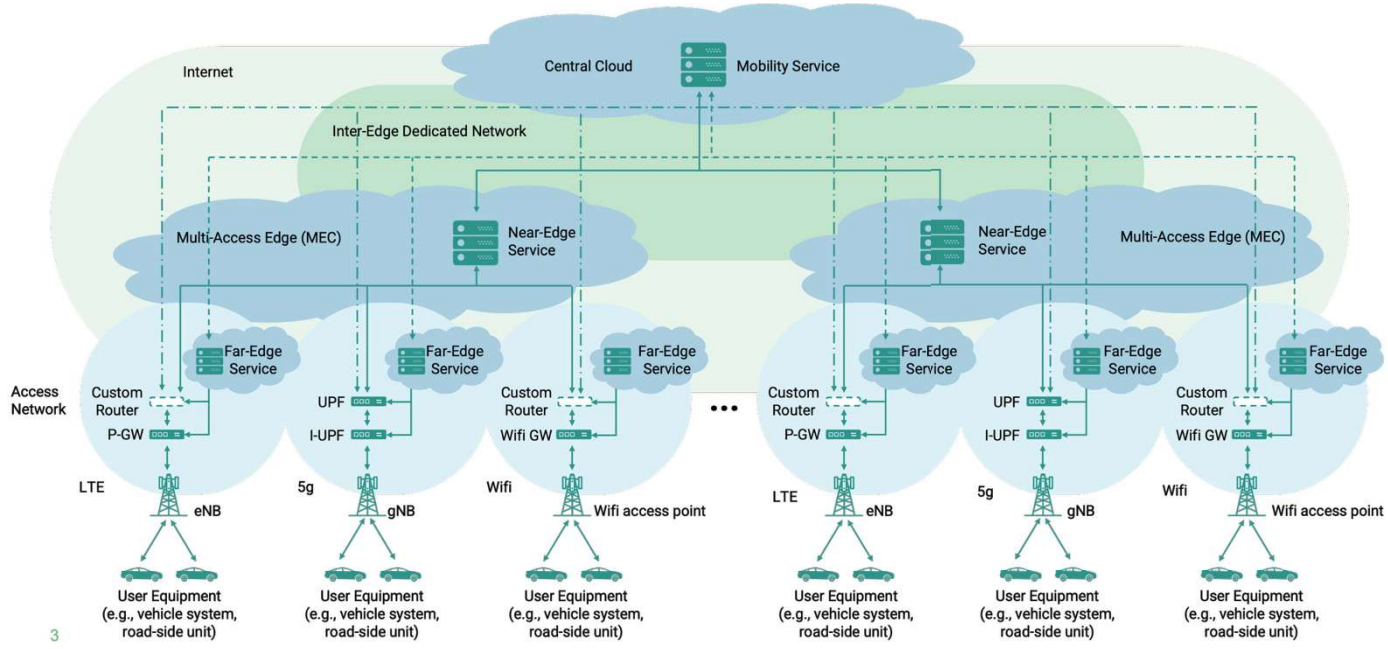

Figure 3.1. AECC physical infrastructure

As illustrated in the diagram, the AECC system assumes the following architectural principles for its underlying physical infrastructure:

- **Access network:** Support for diverse access technologies, such as cellular (LTE, 5G) and Wi-Fi
- **Edges:** Inclusion of multiple edge types, including near edge, far edge, and device edge
- **Network routing control:** Intelligent routing optimized for distributed environments
- **Distributed computing:** Computing resources operating across both central cloud(s) and edge nodes

3.2. Actors and Roles

- **Automotive OEM**
  Primary providers of connected vehicles that generate and consume data within the AECC ecosystem. They design vehicle systems, define data collection policies, and integrate AECC-compliant communication and computing modules. OEMs utilize AECC services for use cases, such as teleoperation, remote diagnostics, autonomous driving support, and/or over-the-air (OTA) updates. They may also share aggregated vehicle data with ecosystem partners under defined governance policies.
- **Tier-1 Supplier:**
  Supplies hardware and software components that implement AECC functions in vehicles or edge nodes, such as sensor-fusion modules, gateway ECUs, and onboard data-management systems. They ensure

interoperability across multiple OEM platforms and may operate localized data processing or analytics services to support OEM or third-party applications.

- **Network Operator (NO)**:
  Provides the communication infrastructure connecting vehicles, edge nodes, and clouds. NOs enable deterministic, low-latency connectivity through technologies such as network slicing, QoS control, and Multi-access Edge Computing (MEC). They also expose network capabilities via standardized APIs (e.g., CAMARA) to allow dynamic optimization of data routing, bandwidth allocation, and mobility management in line with AECC policies.
- **Edge / Cloud Infrastructure Provider**
  Supplies and operates distributed compute, storage, and orchestration platforms required for AECC workloads. They manage hierarchical edge-to-cloud resources, ensure elastic scaling, secure execution, and high availability, and support policy-driven workload placement through energy-efficient, region-compliant operations. These providers form the foundational digital substrate for the execution of AECC services.
- **Mobility Service Provider (MSP)**
  Develops and operates value-added applications that leverage AECC resources, including cooperative perception, high-definition mapping, teleoperation assistance, traffic management, user mobility services, and vehicle-centric digital services. MSPs consume edge/cloud APIs and vehicle data (subject to OEM data governance), deploy distributed applications, and deliver real-time services to vehicles, users, and enterprises.

3.3. AECC Reference Functional Architecture

Figure 3.2 shows the reference architecture of an AECC System, comprising both **control plane** and **user plane** functions. The control plane manages orchestration, policy enforcement, and session control across vehicles, edge nodes, and cloud layers, while the user plane handles data flows for real-time analytics, service delivery, and distributed processing.

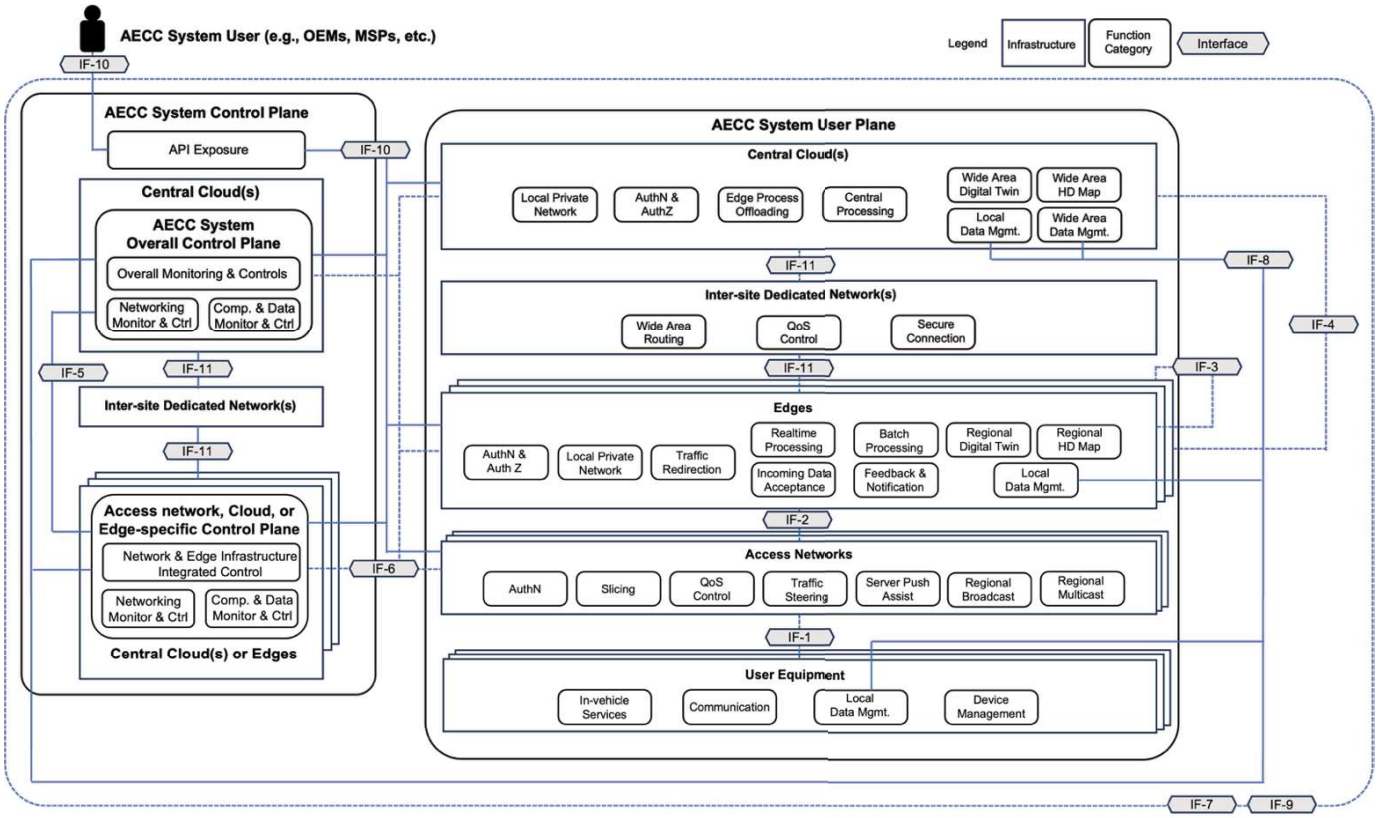

Figure 3.2 AECC reference architecture

3.3.1. Control Plane Functions

The AECC System Control Plane manages a large, distributed ecosystem comprising millions of vehicles, diverse networks, multiple edge nodes, and central clouds. It provides unified management across heterogeneous infrastructures through standardized APIs, enabling consistent configuration, integrated control, and system-wide optimization - such as intelligent traffic steering and dynamic resource allocation. The Control Plane consists of two key sub-functions:

1. **API Exposure**
   This component provides configurability of the AECC system for its users as well as providers in a cascade manner across central clouds and edges.
2. **AECC System Overall Control Plane**
   This component centrally monitors network, compute, and data resources and dynamically adjusts workload placement and data movement across access networks, inter-site networks, edge nodes, and cloud environments. It interacts with individual network and cloud/edge control planes to collect status information, configure resources, dispatch workloads, and maintain system-wide consistency. Dedicated inter-site networks securely connect all control-plane components across geographically distributed areas.
3. **Access-Network, Cloud, and Edge-specific Control Planes**
   These provide unified network–compute control, network Quality of Service (QoS) and Slicing, and monitor workloads and data across their respective domains.

3.3.2. User Plane Functions

The **AECC System User Plane** comprises distributed workloads deployed by OEMs and MSPs, supported by in-vehicle capabilities, including data capture, communication, local storage, and device management. It also leverages access-network features, including authentication, slicing, QoS, traffic steering, and broadcast/multicast services.

Edge site functions handle real-time and batch processing, data storage, feedback delivery, and management of regional digital twins and high-definition (HD) maps. They enable high-speed data ingestion, traffic redirection, and local private networking.

Inter-site user-plane networks provide secure and reliable cloud–edge connectivity with wide-area routing, QoS enforcement, and Virtual Private Network (VPN) integration.

Central cloud functions deliver authentication, private networking, edge processing offloading, large-scale data aggregation, and management of wide-area digital twins and HD maps. They integrate regional data, support advanced analytics, and maintain consistency across distributed edge nodes, enabling large-scale services and applications.

3.3.3. Interoperability and Standardized Interfaces

To ensure interoperability within a large-scale distributed automotive edge computing system, it is critical that interfaces between AECC system components are clearly defined and standardized. Without such standardization, coordinating connectivity, computation, and data management across heterogeneous edge and cloud environments becomes operationally complex and highly vendor-dependent.

Table 3.1 summarizes the functional interfaces required between major AECC system elements, illustrating how each interface supports end-to-end data and control flow. Table 3.2 builds on this by mapping these interfaces to the layers of the AECC architecture, clarifying how connectivity, compute, and data management capabilities interact across the in-vehicle, edge, and cloud domains.

Together, these tables establish the foundation for standardized inter-component interfaces enabling extensible and interoperable AECC deployments across multiple vendors and operational environments.

Table 3.1. AECC reference interfaces

Interface ID	Between	Primary Function	Notes and Responsibilities
IF-1: UE–Access Network Interface	User Equipment(UE) ↔ Access Networks	Vehicle connectivity for uplink/downlink data (telemetry, perception data, OTA, control feedback)	Provided by **NO**; enables low-latency transport and slicing; supports secure data transfer and service continuity.
IF-2: Access–Edge Interface	Access Networks ↔ Edges	Real-time data offloading, MEC processing, and routing of vehicle data to regional edges	Managed jointly by **NO** and **Edge Provider**; used for steering traffic and enabling ultra-low-latency services.
IF-3: Edge-to-Edge Interface	Edges ↔ Edges (through Inter-site Dedicated Network(s))	Exchange of processed or aggregated data between edges; federation of regional services	Enables regional cooperation and resilience; managed under AECC's integrated network control.
IF-4: Edge–Central Cloud Interface	Edges ↔ Central Cloud(s)	Transfer of aggregated datasets, model updates, and global coordination data	Controlled by **Edge/Cloud Provider**; supports hierarchical data flow and long-term analytics.
IF-5: Inter-Control-Plane Interface	Control Plane ↔ Control Plane	Multi-vender orchestration, policy propagation, and telemetry collection	Ensures global coordination of compute, data, and networking; AECC Control Plane's core interface.
IF-6: Control-Plane – User Plane Interface	Control Plane ↔ User Plane	Propagate configuration, e.g., network QoS, compute instantiation, and workload parameters, from the control plane to the user plane per vendor-specific environment	Propagate configurations across a vendor-specific distributed environment
IF-7: AuthN & AuthZ / Trust Interface	All AECC components (Cross-layer)	Identity, authentication (AuthN), authorization (AuthZ), and trust management	Provides zero-trust enforcement and certificate-based authentication across **OEMs, NOs,** and **Edge/Cloud** domains.
IF-8: Data Governance & Catalog Interface	Data Management Components across Edge–Cloud–OEM	Policy enforcement for data usage, residency, anonymization, and metadata catalog	Used by **OEMs**, **MSPs**, and **Cloud Providers**; enables federated governance and auditability.
IF-9: Observability & Telemetry Interface	All runtime entities ↔ Monitoring & Control	Metrics, logs, traces, event collection, Service Level Agreement (SLA) monitoring	Allows the AECC Control Plane to observe system health and enforce SLA/Service Level Objective (SLO) compliance.

Interface ID	Between	Primary Function	Notes and Responsibilities
IF-10: Service Exposure Interface	AECC Platform (Edge/Cloud) ↔ Mobility Service Provider	Expose APIs for configuring every aspect of the AECC system such as networking, computing, storage, and application, and consuming data from the AECC system. This interface also allows **user plane functions to expose services and data** to MSPs.	Help OEMs and MSPs set up and manage the multi-vendor distributed system in a streamlined manner
IF-11: Site - Dedicated Network Interface	Sites (Edges/Clouds) ↔ Inter-site Dedicated Networks	A common control interface for wide-area routing, QoS, and security configuration of the Inter-site Dedicated Networks.	Enable secure and smooth integration of compute and data across a wide area

Table 3.2. Interface-layer mapping

Layer	Connected Interfaces	Main Actor(s)
User Equipment	IF-1 IF-7, IF-9	Automotive OEM, Tier 1, NO
Access Networks	IF-1, IF-2, IF-6, IF-7, IF-9	NO, API Aggregator, Channel Partner
Edges (User Plane)	IF-2, IF-3, IF-4, IF-6, IF-7, IF-8, IF-9, IF-11	Edge / Cloud Provider, MSP
Inter-site Dedicated Networks	IF-9, IF-11	NO, Edge / Cloud Provider
Central Cloud(s) (User Plane)	IF-4, IF-6, IF-7, IF-8, IF-9, IF-11	Cloud Provider, MSP
AECC System Control Plane	IF-6, IF-10	All Actors

4. AECC System Realization Guide

The optimal implementation of an AECC-based system depends on the characteristics and operational requirements of the target service scenario. Different mobility services impose distinct demands on connectivity, computation, data management, and security, leading to varying architectural choices across in-vehicle, edge, and cloud layers.

In this chapter, we present a representative example through **High-definition (HD) Map Update**, one of the major AECC use cases. For this scenario, we outline an example implementation "recipe" from the perspectives of key ecosystem players- automotive OEMs, Tier-1 suppliers, mobile network operators (MNOs), edge and cloud infrastructure providers, and mobility service providers (MSPs). These perspectives demonstrate how system components can be integrated, how data flows can be optimized, and how distributed processing functions can be organized to support large-scale and efficient HD map generation and distribution.

It is important to emphasize that this discussion represents only one possible configuration. Actual implementations must be tailored to the detailed system requirements, application-specific characteristics, and operational constraints of each deployment environment. The purpose of this example is not to prescribe a uniform implementation design, but to illustrate how the AECC system requirements can be realized in practice under a specific service scenario.

4.1. Example Service Scenario: High-definition Map Update

The AECC's high-definition (HD) map solution is a key component of the AECC digital twin framework, designed to manage dynamic events and to provide detailed road-structure information.

To automatically create and update detailed road-structure data, or to detect relatively static objects such as fallen debris, the AECC envisions a system in which image and LiDAR data for each road segment are periodically collected and analyzed from vehicles in motion.

In addition, images and LiDAR data may be collected and analyzed on an ad hoc basis to identify hazardous dynamic objects, such as pedestrians and bicycles. In these cases, data collection is triggered by an in-vehicle mechanism, such as advanced driver-assistance systems (ADAS). Once activated, subsequent vehicles continuously gather data as needed to monitor the target objects. The detected dynamic object information is integrated into the HD map and distributed to approaching vehicles to support accident avoidance, route optimization, and other safety-critical functions.

To realize AECC HD map solutions, the following requirements must be met:

- **Intelligent Data Collection:** Gather large volumes of data efficiently (e.g., 1 MB of image data per minute for every 25-meter road section) while ensuring only necessary and sufficient data is captured.
- **Efficient Data Processing:** Implement mechanisms for rapid analysis and processing of massive datasets.
- **Real-Time Operations:** Enable real-time handling, processing, and analysis to support timely updates and decision-making

4.1.1. Guidelines for Automotive OEM

Preconditions

- Vehicle gateways and ECUs must comply with AECC interfaces IF-1 and IF-2, supporting mTLS, certificate rotation, and differential data upload.
- Vehicles maintain a local buffer of driving data with synchronized Global Navigation Satellite System (GNSS) and IMU (Inertial Measurement Unit) timestamps.
- Data governance policies (e.g., residency, anonymization, retention, consent) are distributed via interface IF-8.

Network Topology Example

- **Forward path:** Vehicle (UE) → Access (5G NR) → Far Edge (MEC) → Near Edge (Regional DC) → Central Cloud
- **Reverse path**: Central → Near/Far Edge (map/model delta distribution)

KPI Targets

- Map-delta freshness: < 1–5 min (regional reflection)
- Availability: ≥ 99.9% (including session continuity during edge handover)

Technical Challenges

- Location-aware load balancing (dynamic Far-Edge selection).
- On-board preprocessing (Region Of Interest (ROI) switching, deduplication, delta encoding).
- Store-and-forward with priority control (safety > map).

4.1.2. Guideline for Tier 1 Supplier

Preconditions

- The in-vehicle software stack must implement sensor fusion, feature extraction, and data anonymization.
- Streaming SDKs supporting Google Remote Procedure Call (gRPC)/Quick UDP Internet Connections (QUIC) or Message Queuing Telemetry Transport (MQTT) streaming Software Development Kit (SDK) and a signed OTA mechanism must be available.

Network Topology Example

- Vehicle Electronic Control Units(ECUs) ↔ Vehicle Gateway (aggregation and preprocessing) ↔ Access Network → Far Edge

KPI Targets

- On-board preprocessing latency: 5–25 ms per frame (using lightweight Deep Neural Network(DNN) and compression techniques)

Technical Challenges

- Calibration of heterogeneous sensors and alignment of metadata across different DNN versions.
- Energy-efficient inference on cost-optimized System on Chips(SoCs) while operating under strict thermal constraints.

4.1.3. Guidelines for Mobile Network Operator (MNO)

Preconditions

- Network slicing and traffic steering must be controllable via interface IF-6.
- MEC platform is connected via IF-2 and IF-5.

Network Topology Example

- Forward path: Radio Access Network(RAN) (gNodeB) → User Plane Function(UPF)@Edge (local breakout) → MEC → Inter-site Dedicated Network → Near/Central Cloud
- Control: Network Data Analytics Function(NWDAF) predicts load and triggers automatic slice resizing.

KPI Targets

- UE ↔ MEC round-trip latency: up to 100 ms

Technical Challenges

- Mitigate network congestion by rescheduling non-urgent transmissions to periods or locations with lower traffic demand.
- Multiple MSPs can securely share network resources without mutual interference while their individual API usage is precisely tracked for billing and management.
- Seamless session continuity across mobility events.

4.1.4. Guidelines for Edge & Cloud Infrastructure Provider

Preconditions

- Far and Near Edge nodes support real-time analytics; the central cloud performs model training and global reconciliation.
- AECC interface IF-9 enables intent-based workload placement optimized for latency, energy, and cost.

Network Topology Example

- Multiple Far-Edges (per city) ↔ Near-Edge (regional data center (DC)) ↔ Inter-site Dedicated Network ↔ Central Cloud
- Edge interconnect uses Segment Routing version 6 (SRv6) or Software-Defined Wide Area Network (SD-WAN); data transport via Kafka/NATS with object storage integration.

KPI Targets

- Edge processing latency: up to 100 ms (preprocessing, aggregation, QA)
- Regional fusion latency: up to 180 s (regional digital-twin alignment)
- Cloud ingestion volume: 1–10 TB/day per region
- Availability: ≥ 99.95% per site, ≥ 99.99% with regional redundancy

Technical Challenges

- Enforcing geographic sharding and data-residency policies (IF-8).
- Detecting cross-edge duplication and applying quality scoring.
- Achieving green operation through renewable-first scheduling and carbon-aware placement.

4.1.5. Guidelines for MSP

Preconditions

- APIs exposed via the AECC interfaces IF-6 and IF-10 provide access to regional digital twins, landmark candidates, driving events, and map deltas.
- OEM data access is governed by contractual agreements and scoped tokens enforced through IF-7 and IF-8.

Network Topology Example

- MSP Application → Developer Portal/API Gateway (Edge/Cloud) → Regional Digital Twin / Map Service
- Bidirectional callbacks are enabled via Webhook or gRPC streaming for regional event subscriptions.

KPI Targets

- API latency (p95): 100–250 ms (regional)

- Map-delta publication SLO: visible within 5–15 min after event capture

Technical Challenges

- Coordinating map-version rollout by region, vehicle class, or service tier
- Maintaining consistency across wide geographic areas.

4.1.6. Example Data Flow

Sensing & Local Preprocessing (OEM / Tier 1)

- Collect sensor, LiDAR, GNSS, and Controller Area Network(CAN) data.
- Perform anonymization, compression, and delta encoding (including ROI selection, deduplication, and spatiotemporal tagging).
- Transmit via **IF-1** (UE→Access); buffer locally if connectivity degrades.

Access & Traffic Steering (MNO)

- Manage QoS and network slicing to ensure low-latency routing through UPF at the Edge.
- IF-6 interface for QoS and network slicing management
- Edge Ingestion & Regional Fusion (Edge Provider)
- **Use IF-2** (Access→Edge) to steer traffic away from congested cells.

Edge Ingestion & Regional Fusion (Edge Provider)

- Far-Edge performs validation, normalization, and clustering of similar events.
- Near-Edge aggregates via IF-3, updating the Regional Digital Twin and Regional HD Map.
- Low-confidence segments trigger re-sampling requests through **IF-10** to MSP/OEM.

Northbound Synchronization & Global Consistency (Edge→Cloud)

- **IF-4** transfers deduplicated map deltas to the Central Cloud during off-peak.
- Central Cloud executes global fusion and model retraining to optimize consistency.

Policy & Orchestration (Control Plane)

- **IF-5/9** apply intent-based orchestration (e.g., scaling out to new edges in high-demand regions).
- **IF-7/8** enforce trust and governance policies (data residency, anonymization).

Publication & Feedback (MSP / OEM / End Users)

- **IF-10** APIs publish validated map deltas at the regional level.
- Critical deltas are pushed via Webhooks/gRPC to vehicles; OTA updates are provided if required.

5. Validation through Proof-of-Concept Trials

AECC has conducted eleven official Proofs of Concept (PoCs), complemented by numerous additional grassroots PoCs led by member companies. Collectively, these PoCs validate that distributed edge computing, standardized telco APIs, and intelligent data governance can enable scalable, sustainable connected-vehicle ecosystems. This section summarizes the key findings from the AECC PoCs, while the complete list is provided in Annex B.

5.1. Efficient Data Transfer and Network Offload

- **Opportunistic Data Transfer (ODT)**: Non-urgent uploads shifted to off-peak hours via Network Exposure Function(NEF)/Service Capability Exposure Function(SCEF) APIs; Tokyo trial required only **≈62 MB** in-vehicle cache, reducing busy-hour load.
- **Wi-Fi Offload + Edge Pre-Processing:** Edge sites using immersion cooling achieved 94% facility power savings and >50% total-power reduction while maintaining service continuity.

Key Insight: Time- and location-aware offload policies can reduce congestion and carbon footprint simultaneously.

5.2. Distributed Edge Computing & Mapping

- **HD Map Distribution**: Multi-cloud edge system achieved **10x faster queries**, **100k TPS updates**, and **<200 ms** processing for map updates.
- **Blockchain-Based Map Validation**: Permissioned Distributed Ledger Technology (DLT) confirmed the integrity and traceability of multi-vehicle data without a central authority.

Key Insight: A hierarchical, multi-edge architecture delivers deterministic latency and trusted data provenance.

5.3. Adaptive Network & Edge Management

- **Edge Relocation**: Real-time rerouting of 5G sessions among edges preserved low latency for critical workloads.
- **Traffic Balancing**: Dynamic WAN/5G distribution lowered average response time under load.
- **Optimal Edge Selection**: The controller prioritized solar-powered or less-loaded sites using the **Traffic Influence API**, balancing performance and sustainability.

Key Insight: CAMARA and 3GPP exposure APIs enable **application-driven connectivity**, dynamically matching service and network conditions.

5.4. Intelligent AI Deployment

- **Federated BEV Range Estimation**: Personalized federated learning cuts resource use by **90%** and training time by **50%**, while improving accuracy.
- **Hybrid LLM Agents**: In-vehicle, edge, and cloud LLMs cooperated for multimodal tasks, minimizing latency and bandwidth.

Key Insight: AI workloads should be distributed training at the edge, inference near data, and orchestration in the cloud.

5.5. Premium Connectivity

- **Quality-on-Demand (QoD) & Network Slicing**: Vehicles switched seamlessly from shared to **premium 5QI 6 slice**, enabling tiered video-conference quality.

Key Insight: Standardized telco APIs allow OEMs and MNOs to offer differentiated in-car experiences.

5.6. Cross-Domain Insights

- **Edge First Design**: Computing near vehicles yields order-of-magnitude latency gains and bandwidth savings.
- **Standard Exposure is Key**: NEF/Policy Control Function(PCF)/NWDAF + CAMARA APIs form a unified control plane for QoS, traffic, and mobility management.
- **Trust & Privacy by Design**: Decentralized validation and on-device learning preserve data sovereignty.
- **Sustainability & Performance Can Align**: Edge cooling, solar-aware routing, and ODT reduce both cost and emissions.

Annex A – Details of Existing Solutions and Gaps

This section provides a brief overview of existing solutions and highlights gaps related to AECC use cases.

A1. Overall Edge Solution Architecture

When considering computing platforms, centralized hyperscale clouds are often the first option that comes to mind. These providers have established numerous data centers in economically advanced countries and are collaborating with network operators to build edge computing platforms —forming what can be viewed as a distributed computing infrastructure.

However, hyperscale clouds alone cannot meet the AECC's vision and use cases, for several reasons:

- **Data Proximity and Scale**
  Future automotive applications require pre-processing near the data source to handle massive data volumes—one to two orders of magnitude greater than today. Hyperscale clouds rely on large, centralized data centers, typically limited to one or two regions per country. Even with multiple vendors, these centers are concentrated in major cities, which conflicts with AECC's need for geographically distributed processing.
- **Real-Time Service Requirements**
  Use cases such as teleoperations demand ultra-low latency and predictable performance. Hyperscale clouds, optimized for rapid scale-out, cannot guarantee the consistent, low-latency data transfer required for mission-critical real-time services.
- **Interoperability and Cost Efficiency**
  AECC envisions unified networking and computing services across providers to optimize response time, resource utilization, cost, and resiliency. This requires open specifications and cooperative data movement. Current hyperscale cloud models, with proprietary APIs and costly egress fees, promote vendor lock-in—contrary to AECC's objectives.
- **Distributed Data Processing**
  Automotive systems generate unprecedented data volumes. Processing location-based information requires platforms that understand boundaries and coordinate across distributed environments. Hyperscale clouds assume centralized processing, offering little support for true distributed data handling.

To address these limitations, several distributed edge infrastructure concepts and solutions have emerged, including: ETSI Network Function Virtualization (NFV) [5] and Industry Specification Group (ISG) MEC [6]

- 3GPP Edge Computing [7]
- GSMA Operator Platform [8]
- 5GAA Solutions [9]
- Cloud Native Computing Foundation - Edge Native Application Principles [10]
- Linux Foundation - LF Edge [11]

For example, ETSI ISG MEC defines conceptual architectures and standard APIs to ensure interoperability, while considering technical aspects such as virtualization. Similarly, 3GPP—the largest telecom standards body—develops edge solutions for network slicing, connection management, mission-critical communications, and automotive-specific features like V2V communication. Further details are available in the AECC white paper "Distributed Computing in an AECC System" [12].

Despite these efforts, gaps remain in areas such as:

1. Implementing traffic routing and timing control for moving vehicles.
2. Managing dynamic location data and delivering information efficiently across distributed edges.
3. Building and operating services on small yet widely distributed MEC platforms.

The AECC adopts a holistic, end-to-end approach, focusing on practical implementation challenges faced by stakeholders. It prioritizes acquiring actionable knowledge, defining a comprehensive edge architecture, and validating concepts through proof-of-concept exercises. This blueprint provides a concise overview of AECC's solutions and direction.

A1.1. Network Redundancy and Optimal Usage

The future automotive industry requires highly reliable communications, as connectivity with remote services is essential for enabling a mobility-driven society.

Why Reliability Matters

Autonomous vehicles require external data sources in addition to onboard sensors. While sensor accuracy has improved, they remain limited compared to human perception, making it difficult for vehicles to handle intersections or curves with poor visibility—posing safety risks. Remote services can provide situational awareness, overcoming these limitations and making autonomous driving safer than human driving.

Similarly, teleoperation—where a remote driver controls a vehicle—is critical in the short to medium term. Current AI systems may fail to make decisions in complex scenarios, leaving vehicles stalled. Teleoperation ensures continuity and also alleviates the burden on logistics drivers.

AECC's Vision for Redundancy

To achieve this, AECC envisions **access network redundancy** and mechanisms that maintain communication even during network failures. Key considerations include:

- **Multiple Carriers and Profile Management**
  Combining networks from different operators and technologies improves resilience. This requires managing multiple profiles on a device and monitoring network status. Emerging technologies like multi-carrier SIM and eSIM support this approach.
- **Satellite Communications**
  Satellite connectivity can back up mobile networks. Low Earth Orbit (LEO) satellites, such as those used by Starlink, are now practical, though bandwidth per satellite is limited (e.g., ~20 Gbps). Services must account for this—for instance, prioritizing safety-critical data during failover. Satellite SIM profiles follow similar logic to multi-carrier setups.
- **Wi-Fi Integration**
  Wi-Fi can complement mobile networks by offering high bandwidth at low cost. Use cases include bulk map downloads, sensor data uploads, and enhancing V2V communications. For dense urban areas requiring high-volume data collection, Wi-Fi reduces costs. However, seamless handover, low latency, and innovations like Wi-Fi 6/7 are essential.
- **eSIM Management and Orchestration**
  eSIM enables dynamic switching among carriers throughout a vehicle's lifecycle, supporting multi-network strategies.

Redundancy Patterns

Different redundancy models balance cost and reliability:

1. **Active-Passive**
   One network is primary; another activates upon failure. Cost-effective but involves brief service interruptions.
2. **Active-Active**
   Multiple networks maintain active sessions (e.g., TCP connections): faster failover but higher cost due to heartbeat maintenance.

3. **Active-Active with Duplicated Delivery**
   Messages are sent over multiple paths for maximum reliability and timeliness. The service processes the first received message and discards duplicates. This ensures continuity but increases cost.

AECC supports the development of these technologies and focuses on integrating them into robust systems. A proof-of-concept exercise will explore optimal configurations, with findings published in future blueprint revisions.

A1.2. Network Slicing and QoS Controls

Network slicing is an emerging technology that enables the creation of logical networks for specific purposes on a shared physical infrastructure. It provides clear traffic separation, enhanced QoS control, and guaranteed SLAs.

Unlike traditional virtualization methods such as VLAN, network slicing offers stronger isolation. VLANs share the same physical devices for packet switching, whereas network slicing uses Virtual Network Functions (VNFs) to separate logical networks. The only common component is the packet classifier within the slicing control and data planes.

Network slicing is critical for AECC scenarios such as:

- **Automated Driving Assistance**
  Vehicles require edge support to compensate for blind spots in onboard sensors.
- **V2N2V Coordination**
  Smooth and safe merging at intersections with heavy traffic and poor visibility demands high QoS connectivity.

Both examples highlight the need for **low latency, high reliability, and guaranteed performance**.

Current Standardization and Challenges

Standards for network slicing and related functions are being developed by **3GPP** and exposed via network services defined by **CAMARA**. However, several challenges remain before large-scale deployment:

- **End-to-End Virtual Networking**
  Specifications for complete communication flows—including edge computing resources—are missing. Mechanisms to route packets to the correct edge and virtual circuit have not been fully defined or tested.
- **QoS Controls**
  Autonomous driving requires high availability and redundancy. For example, redundant paths within a slice must be dynamically configured to ensure continuity in the event of failures. Such software-defined dynamic network control is not yet widely implemented.
- **VNF Platform Development and Validation**
  NFVI solutions vary across vendors (VMs, Kubernetes, Docker, Kata containers). The most suitable approach for security, scalability, real-time performance, orchestration, and long-term operations remains unclear. Performance with hardware accelerators like SmartNICs also needs further testing.
- **Beyond Demonstrations**
  Today, network slicing is mainly limited to lab environments and exhibitions. It has not been validated for large-scale, real-world deployments.

AECC will continue to design solutions and conduct proof-of-concept experiments to address these gaps, accelerating the adoption of network slicing for automotive use cases.

A1.3. Traffic Steering

Beyond QoS controls, traffic routing is a critical enabler for edge-based solutions. The goal is to direct data to the most suitable nearby edge infrastructure, reducing network load and improving performance. However, routing decisions must also consider ensuring the continuous delivery of mobility services:

- Network congestion status
- Edge infrastructure workload
- Coverage areas of location-based services

Traffic steering requires a certain level of intelligence. A key design question is whether this intelligence should reside **in the network** or **in the vehicle.** AECC assumes a **network-based approach** for the following reasons:

- **Centralized efficiency**
  Vehicles in the same region rely on shared information, making centralized management more efficient than distributing logic across individual vehicles.
- **Dynamic reconfiguration**
  Edge infrastructure failures require flexible re-routing without notifying vehicle systems.
- **Resource constraints**
  In-vehicle resources should prioritize autonomous driving and infotainment, not complex routing logic.

Therefore, AECC envisions traffic routing intelligence primarily within the network, with variations based on customer or service requirements.

Several protocols and frameworks are being explored for traffic routing:

- **AnyCast Protocol**
  Routes requests with the same target IP to multiple edge servers. Simple and static—traffic is directed to the predefined nearest edge server.
- **Dynamic AnyCast (DynCast)**
  Extends AnyCast with dynamic routing based on network and compute metrics. Uses ingress/egress D-routers to encapsulate packets, enabling distribution across multiple edge infrastructures.
- **Locator/ID Separation Protocol (LISP)**
  Introduces two namespaces: endpoint identifiers (e.g., edge servers) and routing locators (routers). Initially designed for VM mobility, it is now considered for edge routing. Conceptually similar to DynCast but includes higher-level integration elements.
- **3GPP Application Function influence on traffic routing**
  Enables applications to direct user traffic to the most optimal edge server instance. This allows the MNO network to provide efficient, low-latency traffic handling by dynamically selecting the best network path and UPF for each service.

AECC's Perspective

For connected cars and autonomous driving, AECC sees **DynCast or LISP combined with 3GPP AF influence on traffic routing** as highly promising:

- **AF influence on traffic routing,** enabling routing control in Mobile Network Operators (MNO) by Mobility Service Providers (MSP).
- **DynCast/LISP** allows MSPs to adjust routing independently for operational reasons, such as system changes.

However, practical implementation faces challenges:

- Lack of detailed specifications for real-world use cases.
- Need for software stacks and data management mechanisms to monitor edge server load.

AECC is defining **implementation best practices** and conducting **proof-of-concept (PoC) trials** to validate these approaches.

A1.4. Session and Service Continuity

Most current IoT scenarios rely on session-less and stateless architectures, as data transmission is typically periodic or ad hoc, making session maintenance costly and unnecessary. Additionally, mobile connectivity interruptions reinforce this design choice.

However, AECC use cases—such as intelligent driving and Vehicle-to-Cloud (V2C) cruise assist—require maintaining state and, in many cases, sessions. Edge services must notify vehicles in real time about other vehicles, objects, and events beyond sensor coverage, considering the vehicle's location and planned route. This enables proactive reactions to unseen hazards (e.g., curves, intersections) and smooth merging in complex traffic conditions.

Implications for Distributed Edges

In AECC's distributed edge architecture, vehicles connect to **nearby edge nodes** as they move. Maintaining session and state continuity means ensuring **seamless transitions** during connectivity changes or edge handovers.

Key challenges include:

1. **Session Management and Secure Communication**
   Current SSL/TLS protocols require re-establishing the session—including reauthentication and key exchange—when the destination server changes. To enable smooth transitions:

   Option A: Client-side software creates multiple sessions in advance and switches as needed.
   Limitation: Not feasible when the mobile network controls edge selection without client awareness.

   Option B: Develop a protocol that maintains sessions across edge changes by securely transferring authentication and encryption keys between edge servers.
   This foundation would support secure, seamless sessions for protocols such as **MQTT** across distributed edge devices.

2. **State Management Across Edges**

State information (e.g., vehicle location, data transfer progress) must move between edge servers to ensure service continuity. Challenges include:

- Predicting which edge a vehicle will connect to next based on **driving direction and route**.
- Designing a **data management solution** for efficient state transfer.

To date, most efforts—such as those by 3GPP—focus on handovers between radio units (RU), distributed units (DU), centralized units (CU), and user plane functions (UPF) in 5G networks. However, edge server transitions beyond RU/DU/CU and UPF remain largely unexplored. Similarly, technical proposals for distributed edge state management are scarce.

AECC's Next Steps

AECC plans to **propose and demonstrate a functional architecture** for session and state management in distributed edge environments, addressing these gaps through **proof-of-concept trials**.

A1.5. Vehicle System Reachability

Achieving **real-time, pinpoint push services** is essential for future connected car applications. For example:

- Autonomous vehicles have limited sensing ranges. Remote servers can notify them of obstacles in blind spots.
- Servers may request image data from nearby vehicles to gain insight into specific locations.

Such push-based communication improves both data collection efficiency and the timeliness of location updates.

Server-Push Methods

Several methods exist or are under development (by 3GPP and others):

- **Keep-Alive TCP Connections**
  The vehicle periodically sends keep-alive messages to maintain sessions (e.g., TCP/HTTP). Servers can then push messages over these connections.
  Pros: Already widely used (e.g., Apple Push Notification Service).
  Cons: Communication overhead and delayed detection of connection loss due to keep-alive intervals.
- **Network Exposure Function (NEF)**
  Uses functions such as SCEF (3GPP TS 23.682) and NEF (3GPP TS 29.522) to trigger the vehicle system to re-establish connections to remote servers.
- **Short Message Service (SMS)**
  A legacy GSM-based solution for server push.
  Cons: High cost and poor stability make it unsuitable for large-scale or frequent pushes.
- **3GPP New Push Service**
  Under development (TS 22.174). Automotive industry contributions will ensure support for scenarios like sending messages to groups of vehicles on specific road sections.
- **Geocasting**
  Distributes data to devices within a geographic area using location-based routing. Examples:
  - **Location-Based Multicast:** Extends multicast group management to geographic regions.
  - **GeoTORA:** Uses unicast-like routing to reach the target area, then floods locally.

AECC's Perspective

SMS is too heavy and unreliable for in-vehicle push. Among the other methods, there is no clear consensus yet on the best approach. AECC is actively exploring this space and plans proof-of-concept trials with members to identify the most suitable solution.

A1.6. Network Protocol Optimization for Big Data Transfer and Continuous Connectivity

To enable future automotive services, vehicles must reliably collect and transmit large volumes of data, process it, and receive feedback within strict timeframes—sometimes as short as one second. This requires optimizing communication protocols for both performance and continuity.

Key Optimization Areas

- **Dynamic Maximum Transmission Unit (MTU) and TCP Window Size Adjustment**
  Fine-tuning these parameters improves throughput and reduces latency.
- **Congestion Control Algorithm Tuning**
  Selecting or adapting algorithms for mobile and edge environments.
- **UDP for Multipart Uploads**
  Leveraging UDP for large data transfers where speed is critical.

- **Seamless Connectivity Across Distributed Edges**
  Maintaining encrypted communication during edge handovers is challenging. Current TLS protocols require re-establishing both TCP and TLS sessions when the destination server changes, causing unacceptable delays for real-time services such as autonomous driving.

Several initiatives aim to address these challenges:

- **TCP Variant Analysis [17]**
  Research comparing TCP BBR, Reno, and CUBIC in MEC environments
- **TCP Window Scaling for Satellite Links [18]**
  Investigating performance impacts in satellite communications
- **Windows Server TCP Tuning [19]**
  Microsoft's ongoing improvements to TCP algorithms.

TCP was designed for an era of limited bandwidth and compute resources. Despite improvements, inefficiencies remain—especially for real-time edge computing handling massive data volumes.

AECC's Focus Areas

AECC plans to explore protocol optimization through:

- **Identifying better congestion control protocols**
- **Proactive TCP window scaling** based on network conditions, topology, and vehicle position
- **Seamless communication continuity during edge handovers**
- **Balancing performance and security** through configurable trade-offs

A1.7. Distributed Computing for Edges

AECC envisions a system of distributed computing platforms deployed across centralized clouds, near edges, far edges, and device edges, interconnected by networks. The placement and distribution of workloads and data across these platforms is critical for performance and scalability.

Current Landscape

Several **frameworks** and **technologies** have been proposed and commercialized globally:

- **Frameworks (Functional Architectures)**
  - ETSI NFV and ISG MEC
  - 3GPP Edge Solutions
  - GSMA Operator Platform (OP)
  - 5GAA
  - CNCF / Linux Foundation Edge
- **Technologies**
  - Grid computing: hierarchical Hadoop & Spark, multi-cluster Kubernetes, Virtual Kubelet, BOINC
  - Distributed databases: Oracle Globally Distributed DB, YugabyteDB
- **Generic Distributed Computing Use Cases**
  - Distributed learning
  - Federated learning
  - Crypto mining
  - Blockchain

These trends indicate a strong future for distributed computing. However, **none of these initiatives fully address the automotive industry's unique technical challenges and use case requirements** (see Section 4).

To adapt distributed computing for automotive applications, AECC identifies the following key components:

- **Digital Twin of Data, Computing Platforms, and Networks**
  To optimize workload execution, systems must know:
  - Where the source data resides
  - Available resources (location and timing)
  - Network connectivity options
- **Workload Orchestrator**
  For large-scale workloads handling big data:
  - Divide workloads and data intelligently
  - Execute tasks on the most suitable resources
  - Monitor progress and ensure consistency
- **Data and Workload Pipelining**
  Distributed workloads must exchange data periodically:
  - Peer-to-peer interconnection: Suitable for small-scale environments; requires embedded agents for coordination
  - Bus or database-based interconnection: Better for large-scale environments or limited-resource scenarios; requires a scalable, reliable backbone.

AECC's Role

Other industry groups have not developed these solution elements extensively. AECC will:

- Define **recommended functional architecture specifications**
- Validate them through **proof-of-concept (PoC) activities**

A1.8. Distributed ML/AI Processing Running on Edges

One major advantage of a **distributed edge environment** is its ability to support **machine learning (ML) and artificial intelligence (AI) workloads**. Training and inference for ML/AI models require significant computational resources, making it practical to **bundle GPUs and other accelerators across multiple locations** rather than concentrating them in massive data centers—especially when leveraging **renewable energy** through local production and consumption.

Challenges in Distributed ML/AI

- **Training**
  Distributed ML/AI training is highly sensitive to network latency and bandwidth between nodes. Effective coordination is needed to split workloads and dispatch them intelligently across nodes, considering these constraints.
- **Inference**
  Selecting the most appropriate node for inference requires balancing:
  - Data transmission size
  - Real-time performance requirements
  - Computational capacity
  - Current ML/AI workload distribution

Existing frameworks—such as **MapReduce** and **Apache Spark**—offer entry-level distributed processing capabilities but are **not designed for advanced ML/AI orchestration in edge environments**. There is no widely adopted framework that addresses these requirements.

AECC's Proposed Framework

To fill this gap, AECC proposes and will validate a **distributed ML/AI framework** with the following capabilities:

- **Management and Monitoring**
  Comprehensive visibility into distributed edges and interconnecting networks.
- **Interoperable Runtime Environment**
  Support for diverse ML/AI workloads across heterogeneous platforms.
- **Execution Time Estimation**
  Predict workload completion based on data size and resource availability.
- **Workload Execution Planner**
  Split and dispatch workloads intelligently, detect errors, and enable recovery.
- **Open APIs**
  For workload dispatch and real-time monitoring of execution status.

A1.9. Federated ML/AI Learning Running on Edges

Training ML/AI models in a distributed environment requires effective utilization of data owned by multiple parties. However, protecting sensitive information remains a major barrier. In most cases, confidential data must be manually sanitized before use in the training process, significantly slowing progress.

Federated ML/AI learning addresses this challenge by enabling multi-party data collaboration while ensuring **data sovereignty and protection**. This approach ensures that confidential data is never exposed to the party training the model, enabling secure, privacy-preserving model development.

Currently, federated learning for edge environments is under active research at universities and other institutions. While progress has been promising, practical deployment is still limited. To accelerate adoption, it is essential to build knowledge in the following areas:

- **Scalable Architecture Design –** How to efficiently orchestrate federated learning across distributed edge nodes.
- **Privacy and Security Mechanisms –** Advanced techniques for encryption, anonymization, and zero-trust frameworks.
- **Performance Optimization –** Strategies to minimize latency and bandwidth usage during federated training.
- **Standardization and Interoperability –** Common APIs and protocols to enable collaboration across diverse stakeholders.

To make federated learning practical in automotive edge environments, the AECC is focusing on the following critical areas:

- **Model and data parallelism**
  Model and data parallelism, and some combinations of these two, are well-known techniques for running distributed learning. These provide a basis for how to divide a larger learning process. However, a parallelization model that accounts for the characteristics of the automotive industry, such as location-based data generation, and the characteristics of a hierarchical computing environment comprising MEC, central cloud, etc., has not yet been well-defined. The AECC is trying to gain such knowledge through conducting PoCs, etc.
- **Parameter synchronization mechanism and protocol**
  Another challenge in training in a distributed environment is synchronizing model parameters. In the case of a large language model, the number of parameters can reach trillions, and it is essential to synchronize these parameters in a distributed environment. In the context of the AECC System, it is necessary to determine what parameter synchronization mechanisms and protocols are optimal in a hierarchical distributed computing environment, taking into account the parallelism techniques mentioned above. The AECC is also trying to gain such knowledge.

- **Confidentiality of data**
  In federated learning, at least the model parameters are sent to an external server. The problem here is that an attacker could embed a malicious program into the federated learning workload and spoof the parameters to send sensitive data externally. This will require a mechanism to pre-quarantine distributed workloads and data before they are sent externally. The AECC will define the functional architecture and technical elements for implementing such a system.
- **Confidentiality of computing**
  It is also vital to protect the content of the distributed federated learning workload, because the way in which the data is analyzed is crucial intellectual property of the company. Specifically, once a workload is distributed, it can be reverse-engineered or dumped and analyzed at the platform provider's discretion, which is likely the same entity as the data owner. The solution to this may be the use of more advanced confidential computing techniques, which are expected to protect not only data but also workloads. This is still a research field, but the AECC will monitor the progress of related technology development and prepare the necessary functional architecture in parallel.

A1.10. Real-time Data Processing for Edges

An edge-based distributed architecture must process large volumes of incoming data with **ultra-low latency**, while remaining **scalable** and **reliable**. Several technologies address these needs, each with strengths and limitations:

- **Apache Kafka and Extensions (e.g., ksqlDB)**
  Kafka is a widely adopted open-source platform for scalable data pipelines. It stores data on clustered servers and delivers it to consumers sequentially.
  Limitation: Because data is written to and read from disk, latency is typically several hundred milliseconds—even in on-premises deployments. Extensions like **ksqlDB** enable stream queries but operate as external consumers, so the delay persists.
- **Redis and Valkey**
  Redis, historically open source, is an in-memory key-value store often used for caching. It can scale horizontally by adding nodes and supports **Redis Streams** for continuous data ingestion, functioning like an in-memory Kafka.
  Limitations:
  - No embedded processing (e.g., consistent reads, in-place updates, complex queries).
  - Requires external consumers, reducing real-time performance.
  - Redis is no longer OSI-compliant; **Valkey** is the community-driven alternative with similar features.
- **Apache Flink**
  Flink is an open-source framework for **stream and batch processing**. It supports in-memory, distributed stream processing with minimal reliance on storage (only for state recovery).
  Strengths: High-speed, real-time analytics.
  Limitations:
  - Memory exhaustion can cause crashes, limiting scalability under fluctuating loads.
  - Often paired with Kafka or Redis for ingestion, which adds latency.
- **Apache Spark (Streaming)**
  Originally designed for big-data analytics, Spark now supports streaming with **Spark Streaming** and **Structured Streaming**.
  Strengths: Rich ecosystem for ML, AI, and data science.
  Limitations:
  - Processes streams in micro-batches, introducing latency (typically 100 ms–1 s).
  - Continuous processing mode (introduced in Spark 2.3) aims to reduce latency but remains experimental.
- **Other Solutions**
  Various open-source and commercial products attempt to fill gaps in real-time performance, scalability, and elasticity—such as proprietary IoT streaming services from cloud vendors. However, technical specifications often reveal **insufficient real-time capabilities**.

Given these limitations, the **AECC** will:

- Endorse the development of advanced data management technologies for edge-based architectures.
- Conduct **PoCs** for candidate technologies.
- Explore **geo-distributed data management systems** to handle automotive-specific requirements, such as:
  - Data in diverse sizes and formats.
  - Dynamic data movement aligned with vehicle mobility
    Current database solutions offer partial capabilities (e.g., data propagation, integrated search) but do not fully meet these unique needs.

A1.11. A Digital Twin Platform Designed for Edges

The primary advantage of implementing services on distributed edge nodes is the ability to **localize data processing by region**, thereby reducing communication traffic, minimizing the volume of data transmitted, and lowering computational and storage demands at any single edge site.

However, even in such a distributed architecture, mobility services introduce unique challenges. A well-designed platform is required to manage **dynamic states of vehicles and drivers**, monitor **traffic conditions** such as congestion and restrictions, and respond to **events occurring on or near the road** in real time.

At the same time, large-scale distributed environments significantly increase operational complexity. This necessitates **continuous monitoring of communication and edge server workloads**, along with the ability to **instantly determine where and how processing should occur** to maintain efficiency and reliability.

Digital Twin as a Solution

A promising approach to these challenges is the **digital twin—**a virtual representation of physical assets, processes, and environments. Numerous organizations and companies are actively researching and developing digital twin solutions, including:

- **International Data Space Association (IDSA) and GAIA-X**
  Focused on reference architectures for data spaces, emphasizing **data exchange and reusability**.
- **European Big Data Value Forum**
  Developing a **digital twin data processing pipeline** within the context of big data.
- **ETSI Specialist Task Force (STF) 628**
  Modeling IoT digital twins and creating **reference architectures and interoperability guidelines**.
- **3GPP Study on Digital Twins for Network Management**
  Exploring digital twins to **manage complex networks efficiently**.
- **Commercial Initiatives**
  Companies such as **NTT, Ericsson, Oracle, AWS, Microsoft, Google, and IBM** are proposing solutions that integrate their products and services into digital twin platforms.

Despite these efforts, existing initiatives fall short for **edge-based digital twin platforms** because:

- **Consortia** prioritize interoperability across **heterogeneous domains**, rather than regional segmentation.
- **Private companies** often design digital twin implementations for **single data centers**, focusing on scalability but neglecting **real-time responsiveness**.

AECC's Approach

In contrast, the **AECC** is pursuing a fundamentally different strategy:

- **Real-time digital twin services** through **regional segmentation**.
- **Interoperability and seamless service continuity** among regional digital twins operating across diverse edge environments.

This approach ensures that digital twins are not only interoperable but also optimized for **low-latency, real-time operations** in highly dynamic, geographically distributed automotive ecosystems.

A1.12. Network and Edge Management APIs

As we move toward an era of **distributed edge technology integrated with mobile and fixed networks**, various **Standards Development Organizations (SDOs)** and industry consortia are actively defining **APIs and common frameworks** to enable interoperability and programmability.

Network exposure—or service exposure within the network domain—refers to making network capabilities (such as data and network services) easily accessible for external consumption. With proper security and data integrity policies, network data and resources can be shared across ecosystems to **enrich applications and drive innovation**. This capability is essential for **programmable networks** that support diverse 5G use cases, including **gaming, drones, smart manufacturing, automotive applications**, and more.

Key Industry Initiatives

GSMA

The GSMA, representing mobile network operators globally, plays a pivotal role in shaping mobile communications and driving innovation through:

- **Standards and Best Practices**
  Defines specifications for **5G, IoT, and mobile identity**, ensuring interoperability and enabling new services and business models.
- **Industry Collaboration**
  Initiatives like the **IoT program** standardize device connectivity and security, simplifying IoT service delivery.
- **Advocacy and Regulatory Support**
  Engages with governments and regulators to create policies that foster investment and innovation in technologies such as **5G and IoT**.

TM Forum

TM Forum focuses on **digital transformation and business process standardization**, offering frameworks and tools such as:

- **Frameworx Suite (eTOM, SID, TAM)**
  - **eTOM**: Business process framework
  - **SID**: Standardized data model
  - **TAM**: Reference architecture for telecom applications
- **Open Digital Architecture (ODA)**
  Provides a blueprint for **modular, cloud-native architectures**, enabling agility and scalability.
- **Catalyst Projects**
  Collaborative initiatives leveraging **AI, 5G, and blockchain** to solve industry challenges.
  Example: *Network service monetization through standardized APIs*, demonstrating how GSMA Open Gateway, TM Forum, and CAMARA can support multiple use cases.

CAMARA

An open-source project under the **Linux Foundation**, launched in collaboration with GSMA, CAMARA focuses on **defining and validating user network APIs**:

- **Open-Source Approach**
  Uses **Apache 2.0 license** for API definitions and reference implementations, reducing entry barriers and accelerating adoption.
- **Technical Scope**
  Defines **northbound service APIs** and **service management APIs**, abstracting southbound APIs to hide telco complexity while maintaining operator control and compliance with regulatory and privacy requirements. Complete API inventory: CAMARA API Backlog.

Joint Initiative: Open Gateway

The **Open Gateway** initiative aims to create a **standardized API framework** for universal access to operator networks, enabling telcos to expose network capabilities as services that integrate seamlessly into digital platforms and applications.

- **GSMA**: Maintains the overall framework
- **CAMARA**: Develops API standards for service provision and management
- **TM Forum**: Defines API standards for platform management

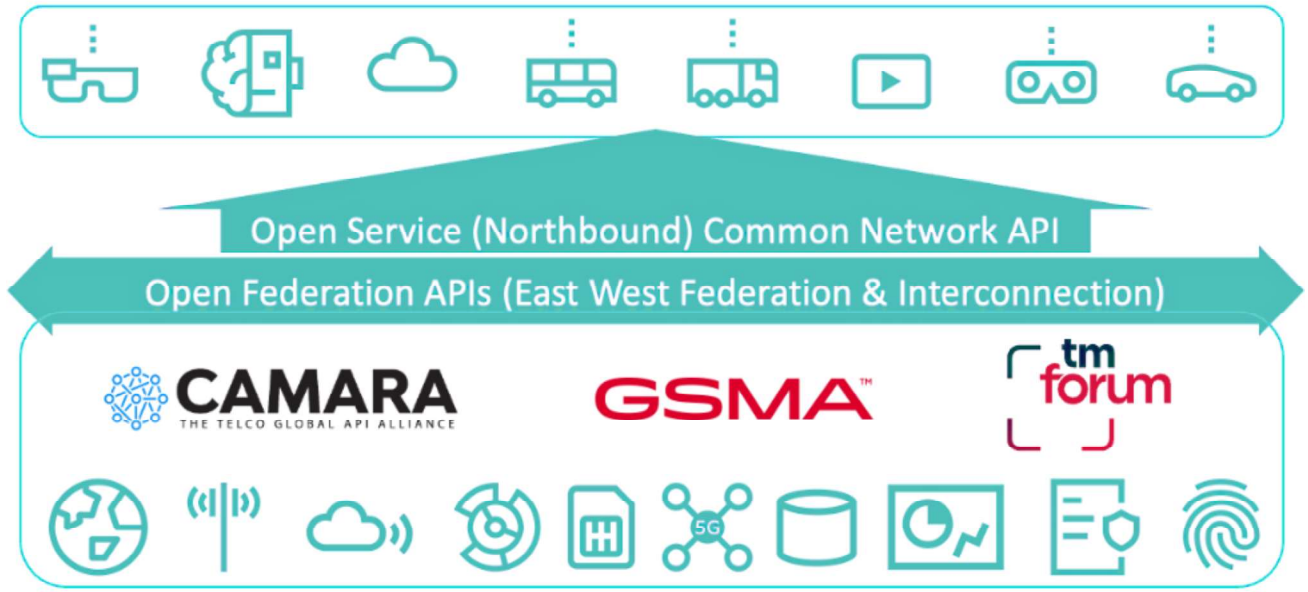

Figure A.1 Open Gateway overview

Benefits of API Standardization and Interoperability

- **API Standardization and Interoperability**
  By defining standardized APIs, **Open Gateway** simplifies integration for third-party developers. This is essential for creating a **seamless digital ecosystem** where telco services can be easily leveraged by industries such as **automotive, healthcare, and smart cities**.
- **Accelerating Innovation and Service Delivery**
  Standardized APIs enable **faster and more efficient service deployment**. Telcos can expose capabilities such as **location-based services, identity verification, and network slicing**, enabling developers to build innovative applications that leverage these features.
- **Enhancing Customer Experience**
  Exposing network capabilities through APIs enables **personalized and responsive services**. For example,

APIs can provide **real-time analytics and monitoring**, empowering telcos to deliver **proactive maintenance and improved customer support**.

GSMA and AECC Collaboration

In **October 2024**, the **GSMA** and the **Automotive Edge Computing Consortium (AECC)** signed a formal agreement to accelerate the introduction of **connected vehicle services.** This collaboration leverages **GSMA Open Gateway** and **GSMA Fusion** to align the automotive and mobile industries, fostering innovation in areas such as:

- **Intelligent in-vehicle services**
- **Driver assistance systems**
- **Green mobility solutions**
- **Edge AI services**

Open Gateway API Categories

Open Gateway defines and standardizes APIs exposed by **network operators**, **edge service providers**, and **cloud service providers.** These APIs are grouped into three categories:

1. **Operate APIs –** For operational control and monitoring
2. **Service APIs –** For accessing network and edge capabilities
3. **Service Management APIs –** For lifecycle and configuration management

(See figure below for conceptual architecture.)

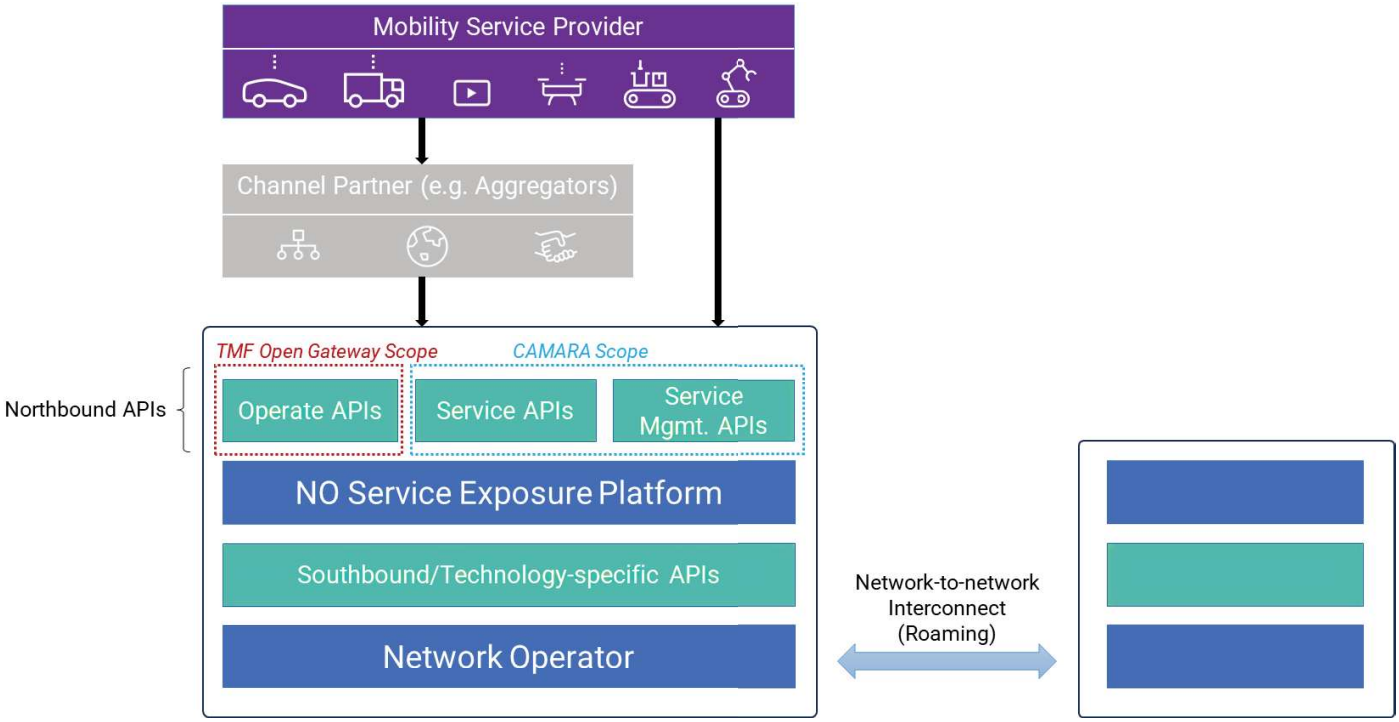

Figure A.2 Open Gateway APIs

Clarifying API Needs for Edge-Based Solutions

During ongoing standardization efforts, numerous APIs targeting edge solutions are being proposed and partially defined across different organizations. However, most of these APIs remain **incomplete**, and development is progressing without a clear view of the **overall edge architecture.** This lack of architectural clarity creates confusion about **how to combine multiple APIs** to build edge solutions for AECC use cases.

AECC's Approach

To address these challenges, the **AECC** plans to:

1. **Define Use Cases and Architecture**
   Clarify the specific automotive edge use cases and establish the **overall architecture** for an edge-based distributed system.
2. **Identify and Organize APIs**
   Determine **which APIs are needed**, their roles, and **how they should be combined** to implement the edge-based solution effectively.
3. **Validate Through PoCs**
   Conduct **proof-of-concept (PoC) testing** to verify API functionality, interoperability, and performance within the defined architecture.

A1.13. Edge Design and Technology

Edge Design and Technology

To enable the **edge-based solution era**, it is critical to define **how to build an effective and efficient edge** and develop the necessary technologies. If edge infrastructure fails to match or exceed the efficiency of today's **hyperscale cloud platforms**, industry adoption will be limited.

Currently, many consortia focus primarily on **virtual machines and container environments** for near and far edges. While these are important, they are **not sufficient** to make edge solutions compelling. Hyperscale clouds offer rich **platform services** that simplify IoT and other applications. To compete, **similar platform capabilities must be embedded into edge environments**, along with **new services not available in hyperscale clouds**, such as:

- **Unified configurability across multiple edges**
- **Distributed data management and intelligent data movement**

Additionally, edge platforms must incorporate **underlying mechanisms found in hyperscale clouds**, including:

- **Control planes**
- **Software-defined networking**
- **Storage services**
- **Security mechanisms for multi-tenancy**

Integration and Scalability

- **Access Network and Edge Integration**
  Running control software on user equipment (UE) reduces productivity and resource efficiency, and introduces configuration inconsistencies that compromise system integrity. Seamless integration between access networks and edges is essential.
- **Small-Scale Deployability**
  Unlike hyperscale clouds built on thousands of racks, near-edge deployments may require only **a few racks**. Edge design must support **efficient resource pooling** at this scale.

Key Design Considerations

1. **Integration of Access Networks and Edges**
   - Seamless integration of **network slices** and **virtual networks** on edges
   - Robust security with low overhead
   - Intelligent traffic steering based on **network congestion** and **edge workload status**

2. **Computing Infrastructure**
    - Virtual machines, containers, and orchestration frameworks
3. **Platform Services**
    - Scalable data queuing
    - Real-time stream processing
    - Intelligent data relocation
    - Distributed data processing
4. **Efficient Resource Pooling**
    - Incorporate all components above within **dozens of racks**, ideally, **a few racks.**
5. **Isolation Mechanisms**
    - Software-defined networking
    - Storage isolation
    - Security checks for multi-tenancy

AECC's Role

AECC welcomes and encourages all activities that contribute to proper edge design and technology development. Furthermore, AECC plans to **collaborate with technology vendors** to realize advanced edge computing and conduct **joint PoCs**.

A1.14. Security and Privacy Protection on Edges

Security and privacy are **critical components** in realizing edge-based solutions. With increasing regulatory requirements—such as the **EU General Data Protection Regulation (GDPR)**—and growing concerns about **economic security** due to risks of nation-level espionage and sabotage, edge environments face unique challenges. Unlike centralized systems, **distributed edges are more exposed**, making unauthorized physical or logical access easier.

To address these risks, **robust and holistic mechanisms** are needed to:

- Prevent **data theft or tampering**
- Ensure **personal information is used only within the scope of user consent**

Key Security and Privacy Requirements for Edge Environments

1. **Removal of Unnecessary Private Information**
   Vehicle data often includes personal information. Examples:
    - **Image data** may capture pedestrians.
    - **Driving data** can reveal home addresses via starting points.
      Mechanisms should be implemented at the **device edge or far edge** to remove unnecessary personal data based on the intended purpose.
2. **Secure Storage and Exchange of Encryption Keys**
    - Encryption keys must be securely stored and managed, typically using **Hardware Security Modules (HSMs).**
    - Keys should be generated by the **data owner** and securely transferred to HSMs.
    - Emerging technologies such as **quantum-secure cryptographic communication** are desirable, as conventional public-key encryption may become vulnerable in the **post-quantum era**.
3. **Data Confidentiality**
    - Data in distributed edge environments must be protected against unauthorized access and misuse—even by legitimate users.

- Recommended measures:
  - Encrypt data with securely managed keys
  - Rotate encryption keys regularly and segment them by data units
  - Implement **access control** to authenticate and authorize users based on agreed data usage policies

4. **Confidentiality of Computing**
   - Distributed environments increase the risk of **physical tampering**.
   - While stored data can be encrypted, **data in use during computation** is often exposed.
   - Solutions must include:
     - **Malware prevention mechanisms**
     - **Confidential computing technologies** to protect data during processing

Industry Status and AECC's Role

Few consortia are actively addressing these challenges. However, without robust security and privacy solutions, **edge adoption will stall**. For this reason, the **AECC** has initiated efforts to **develop security and privacy frameworks tailored for edge computing**, ensuring compliance, resilience, and trustworthiness.

Annex B – AECC Proofs-of-Concept

The AECC reference architecture (Section 3) and implementation guidelines (Section 4) outline how to achieve the automotive industry's desired state—where in-car and around-the-car experiences are fully digital. However, their feasibility must be validated. To this end, AECC has conducted and continues to conduct proof-of-concept (PoC) initiatives.

The table below summarizes PoC activities completed to date, some of which are published on the AECC public website. Additional PoCs are currently in planning or execution and will be published as they become available.

Table 8 AECC proof-of-concept activities

Category	PoC Title	PoC Proponents & Executors	PoC document
Vehicular communications	Using Opportunistic Data Transfer for Connected Vehicles to Reduce Cell Congestion in Existing Mobile Networks	Denso, Toyota, KDDI	PoC1 [23]
Vehicular communications	Wi-Fi Data Offloading and Edge Computing for Greener Mobility Services	Toyota, KDDI, Cisco	PoC4 [26]
Vehicular communications	A V2X Network Digital Twin	Toyota	
Vehicular communications	A Replaceable Next-generation Connected System	Toyota, Itochu Techno-Solutions, SORACOM	
Vehicular communications	Premium Communication Services Utilizing Telco APIs	Ericsson, KDDI, Toyota	PoC11 [33]
Edge computing	Enabling Distributed Edges for HD Mapping	Oracle	PoC2 [24]
Edge computing	Edge Relocation: Optimizing 5G Resources and Edge Networks to Enable Varied Mobility Services	Toyota, Ericsson	PoC6 [28]
Edge computing	Enabling Trusted HD Mapping Data Updates in a Multi-organizational Distributed Edge	Toyota, Intel	PoC5 [27]
Edge computing	Hierarchical Edge AI	Toyota, Techno-Accel Networks	
Edge computing	Traffic Load Balancing of Edge Server Access	Toyota, KDDI, Oracle	PoC9 [31]
Edge computing	Green Connected Platform Field Trial	KDDI, Cisco, Toyota	
UX application	Distributed Battery Electric Vehicle (BEV) Range Estimation via Personalized Federated Learning	FEDML, Toyota	PoC7 [29]
UX application	AI Agents Utilizing End-to-end LLMs	Toyota, Oracle, KDDI	PoC8 [30]
UX application	Enabling a Geolocation Parking Service with AECC Distributed Edge Architecture	KDDI, Nexar, Oracle	PoC3 [25]
UX application	Service-oriented Vehicle Diagnostics (SOVD)	Toyota, Denso, Vector Japan	
Safety application	Cooperation for Automation	Toyota,	

Category	PoC Title	PoC Proponents & Executors	PoC document
Safety application	Vulnerable Road User Awareness & GLOSA / Time-to Green via STEP	Vodafone, Toyota	
Safety application	Real-time precise positioning with Edge and 5G	Toyota, Orange, Here Technologies	
Safety application	Optimal Edge Selection for Realizing Digital Twins and Green Energy Utilization	Toyota, KDDI, Oracle, Equinix	PoC10 [32]
API exposure	Next Generation of 5G & API Enabled Cars	Ericsson, Toyota	
API exposure	Autonomous Vehicle Enablement Via CSPs Network APIs	Ericsson, NTT Data, Toyota (Special Thanks: AT&T)	
API exposure	An Open Gateway as Use Case Accelerator and Technology Ecosystem Enabler	NTT Data	

Annex C – API Framework

C1. Operate API Inventory

Following GSMA requirements, APIs are defined in the Operate APIs domain to fulfill required operations, administration, and maintenance needs. The APIs currently assessed for this domain are summarized below, along with expected plans for future additions and a detailed breakdown of the API functions.

Table C.1 Operate API list

Function	Description	TM Forum specification number	Status
Application & developer management	This function enables automation when registering developers and onboarding their applications through an aggregator/channel partner portal. Operators shall provide the capability for aggregators to request the registration of developers and the onboarding of their applications. This comprises end-to-end lifecycle management, including validation of the legal entity, modification of existing applications, or management of T&C acceptance.	TMF931	Published
API catalog management	This denotes the need to provide a standard catalog among operators and aggregator/channel partners to align the lifecycle management of the APIs. Operators shall provide the capability for aggregators/channel partners to manage (i.e., create, read, update, and delete, or CRUD) APIs, building up the northbound Open Gateway catalog. Catalog synchronization is needed so that aggregators/channel partners can keep up-to-date lists of Open Gateway APIs available for developers to consume.	TMF936	Published
Usage reporting	This defines a unified way to report usage information for the Open Gateway services, so that aggregators/channel partners can make it available to the developers accordingly. Information related to the consumption of service APIs is managed by operators, who have end-to-end views of the services and network resources. This information is exposed to aggregators/channel partners for consolidation and billing management.	TMF937	Planned
Service assurance	This function provides a common framework for supervising Open Gateway services. Operators need to provide developers and aggregators/channel partners the capability to supervise the status of running Open Gateway services, using appropriate performance management (e.g., monitoring) and fault management (i.e., issue/ticket management) mechanisms.	To be assigned	Planned

C2. API Use Cases

An end-to-end flow in which a mobility service provider (MSP) subscribes to and uses services of network operators (NOs), cloud service providers (CSPs), and edge service providers (ESPs) can be summarized in the following diagram. The process begins with onboarding the mobility provider and defining its organizational structure within the network. Then, devices are activated, and SIM subscriptions are set up. Next, the MSP registers applications on top of cloud/edge infrastructure in most cases, including those that access network APIs. The network services are aligned and synchronized with the MSP's offerings. The MSP then requests access to network APIs and begins consuming network services. Usage is monitored, and assurance requests can be raised if needed. Finally, billing and settlement occur based on service consumption and usage.

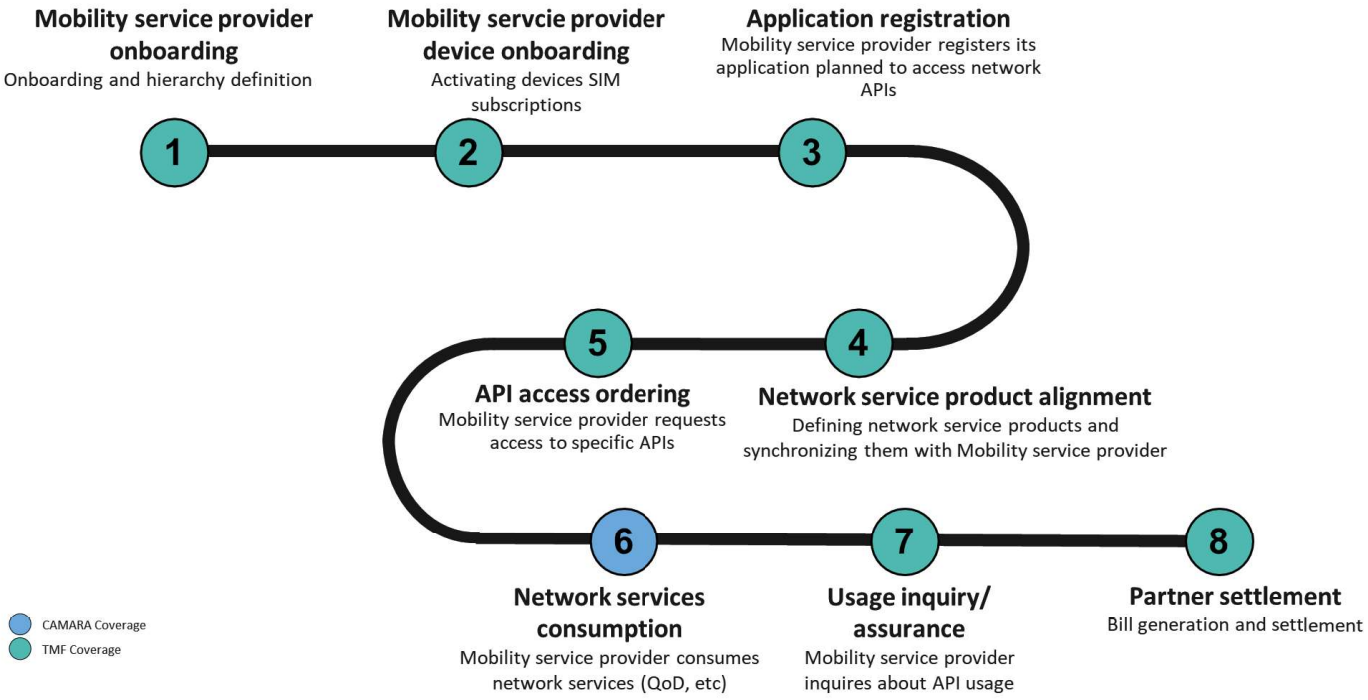

Figure E.1 Use case flow

C2.1. Use Case 1: Mobility Service Provider Onboarding

This use case registers an MSP on a network or platform, verifies its credentials, and establishes the necessary agreements for collaboration. The required steps include gathering relevant business details and technical requirements, and ensuring compliance with industry standards and regulations. There are multiple variations in this use case, relying on the extent of the relation between the MSP and other parties, as well as the agreement channel employed.

This use case can be implemented:

- For later support of SIM activation and management, and for covering network service API consumption.
- For support of SIM activation and management for vehicle modems of the MSP, a comprehensive set of TMF APIs will be used for managing parties, party roles, accounts, etc. Such a set of APIs falls outside the Open Gateway's scope.
- For covering network service consumption, APIs will be used for managing either channel partner agreement or a direct mobility provider and NOs/CSPs/ESPs relationship:

- In the case of a channel partner agreement, the MSP representative will use the channel partner portal and its APIs to populate the information that will later be reflected in the NOs/CSPs/ESPs' portal, using the TMF931 Open Gateway Onboarding and Ordering Component Suite API.
- For managing a direct mobility provider and NOs/CSPs/ESPs relationship, the MSP representative will use the NOs/CSPs/ESPs' portals and their APIs. The TMF931 Open Gateway Onboarding and Ordering Component Suite API specification has been defined to standardize these APIs and streamline their development.
- It should be noted that scenarios in which non-technical systems are used, e.g., documented agreements that are reflected manually in NOs/CSPs/ESPs' systems, are not addressed in this document.

C2.2. Use Case 2: Mobility Provider Device Onboarding

This use case refers to the process of activating devices, such as embedded modems or IoT sensors, by provisioning them with SIM cards, data plans, and secure network access. For example, when deploying connected vehicles, the onboarding process ensures that each vehicle's telematics device is correctly activated, enabling real-time tracking of the vehicles and data communication to and from them. This use case is not relevant in scenarios where an MSP uses the network APIs to request data unrelated to their devices. The Open Gateway scope does not cover this scenario.

C2.3. Use Case 3: Application Registration

This use case registers applications that will interact with the network services. This registration ensures that the application is authorized to consume network services by assigning the necessary permissions and that application usage is logged and later provided to the MSP. MSP can rely on a channel partner or directly engage with NOs and/or CSPs/ESPs. In the former case, MSPs will use the channel partner's portal and its APIs; in the latter case, the NOs/CSPs/ESPs' portals and their APIs. The TMF931 Open Gateway Onboarding and Ordering Component Suite API specification has been defined to standardize these APIs and streamline their development, too.

C2.4. Use Case 4: Network Service Product Alignment

This use case enables CSPs/ESPs to leverage network services and integrate them into their products, such as connectivity and location-based services. For example, an MSP offering a vehicle tracking service would rely on the NO's pre-defined location-based service, such as geofencing, that leverages a specific network service from one or several NOs. Such alignment streamlines the development of CSP/ESP services and optimizes their performance. The TMF936 Open Gateway Product Catalog API specification has been defined to standardize APIs for this use case and streamline their development.

C2.5. Use Case 5: API Access Ordering

This use case allows MSPs to request access to network APIs that they will later consume, based on pre-agreed product pricing. For example, an MSP developing a vehicle tracking service may place an order to access a CSP/ESP's location-based APIs, such as geofencing, based on the agreed pricing model. This ordering process ensures that the MSP has authorized access to the necessary network APIs and enables the CSP/ESP to manage and allocate resources efficiently, while also ensuring transparency and accurate billing for the API access. The use case can then be fulfilled via a channel partner portal or NO/CSP/ESP, and the exposed APIs from that portal. The TMF931 Open Gateway Onboarding and Ordering Component Suite API specification has been defined to standardize these APIs and streamline their development.

C2.6. Use Case 6: Network Service Consumption

This use case enables an MSP to consume network services offered by an NO via standardized APIs. For example, an MSP managing autonomous vehicles may consume a dedicated network service to ensure low-latency, high-quality connectivity for real-time data transmission, adhering to the CAMARA standard.

This usage ensures interoperability between the NO's network and the MSP's platform, which will be built on top of cloud/edge infrastructure, enabling seamless service delivery, automated provisioning, and standardized billing. By leveraging CAMARA standards, the process becomes more efficient, scalable, and secure, allowing the MSP to integrate and manage network services easily. This is a key use case that delivers real network value from the NO to the MSP.

CAMARA APIs can be categorized into different API portfolios covering anti-fraud detection, mobile connectivity, fixed connectivity, cloud and edge use, and payments and charging. Different mobility requirements can be met by one or more APIs, depending on the business requirement and context. A full inventory of APIs is available at the APIbacklog repository of the Camara Project on the GitHub [21].

C2.7. Use Case 7: Usage Inquiry and Assurance

This use case enables an MSP to monitor and manage the usage of the network APIs they consume and ensure that their quality meets predefined service levels. For example, an MSP using location-based APIs for a ride-hailing app may inquire about API call volume, latency, and data accuracy to ensure the service meets the agreed-upon key performance indicators. If performance deviates from expected standards, the MSP can submit an assurance request to the NO/CSP/ESP to address issues such as API downtime or degraded performance, ensuring reliable service delivery and maintaining a high-quality user experience. The TMF937 Open Gateway Product Usage Management API specification has been defined to standardize APIs for this use case and streamline their development; however, additional TMF API specifications may be determined.

C2.8. Use Case 8: Partner Settlement

The use case is to support an MSP in settlement for the usage of network APIs with a NO. It covers the capabilities of billing and settling payments between the MSP and the NO based on the network services used. For example, an MSP using connectivity APIs for real-time vehicle tracking would be billed based on the volume of API calls, data usage, or service-level agreements. After consuming the network APIs, the NO generates a bill reflecting the agreed-upon pricing model, which is then reviewed and settled by the MSP. There are multiple variations of this use case, depending on the extent of the relationship between the MSP and the NO, as well as on the channel partner involved.

Annex D – Glossary and Acronyms

Term	Definition
Access Network	A network that connects end-user equipment or devices to various remote services
Advanced driver-assistance system (ADAS)	A system that assists the driver in driving a vehicle to avoid the risk of accidents and to increase comfort, which primarily runs in the vehicle, but also runs at remote edge services
AECC System	A network system that is built according to AECC specifications and/or recommendations *Note 1: The AECC System spans multiple sites.* *Note 2: The AECC System's physical infrastructure includes various network, computing, and storage devices.* *Note 3: The AECC System runs two kinds of workloads: AECC System control plane workloads and AECC System user plane workloads.*
AECC System Control Plane	A mechanism for governing the entire AECC System, and for streamlining the deployment and execution of various AECC System user plane workloads
AECC System User Plane	A set of distributed workloads and data that is provisioned for specific business purposes, with support of the AECC System control plane
API Aggregator	Any party that provides aggregation of multiple MNOs' network APIs
Breakout Point	An exit and entry point between the mobile network and the external Internet or other wide area networks
Business Service	Any service built for a business purpose, which includes mobility services, internal data usage services, and in-vehicle services *Note: A business service may serve as the foundation for other business services.*
Central Cloud	A platform that can perform various tasks due to its large-scale networking, computing, and storage resources, as well as various development support and data management tools *Note: The number of available central clouds tends to be limited as each one requires a large data center and a large resource pool within it, and its computing efficiency and real-time capability might not be very good, as it prioritizes scalability.*
Central Server	A unit of the computing and storage resources that run in the central cloud
Channel Partner	A partner company, such as a reseller, service distributor, service aggregator, vendor, retailer, or agent, that partners with another organization to market or sell its services, products, or technologies
Cloud Service Provider (CSP)	A company that builds central cloud data centers and provides various commercial services from there
Commercial Service	A type of business service that is provided to other parties for the purpose of generating revenue or enhancing customer satisfaction
Data Center	A facility that consists of multiple networks, computing, and storage resources, with robust power supplies, air conditioning, and security systems
Device Edge	A concept of running various services within a device, such as pre-processing data, making certain decisions, and taking automated actions *Note: Device edge services are not affected by network communication latency or failures, as data processing is performed inside the device. However, when the device's computing and storage resources are more limited, only limited services can be executed.*

Term	Definition
Digital Twin	A solution concept for understanding the state and condition of real-world objects of interest in detail, accurately simulating what will happen with these objects in the near future, and determining actions to be performed to optimize the results *Note: A digital twin can be positioned as a common business service that enables various other business services.*
Distributed Computing	A solution framework for processing data in a distributed environment, which may consist of central clouds, near edges, far edges, and device edges
Edge	Some infrastructure or platform that can be used to execute assigned tasks and store some data as part of the whole, and that works with other edges to construct a distributed system
Edge Server	A unit of the computing and storage resources used in near edge and far edge infrastructures
Edge Service Provider (ESP)	A company that builds near-edge and/or far-edge infrastructures and provides various commercial services from there
Far Edge	An infrastructure or a simple platform that sits close to user equipment and is therefore more distributed in operation. It provides services that can be used to preprocess collected data from devices and send various feedback to the devices. *Note 1: The far edge may be embedded inside or placed just outside the access network to reduce the access latency seen by devices.* *Note 2: On the other hand, to be compact for such embedding, the number of services offered would be limited compared to the central cloud or the near edge; e.g., infrastructure services only.*
Infrastructure	A collection of networking, computing, and storage resources that can be used to build and run various services
Internal Data Usage Service	Various internal data usage services, such as data analysis, are used by OEMs to improve vehicle quality
Inter-site Dedicated Network	A network that connects sites (edge sites and cloud sites) through dedicated lines
In-vehicle Service	Various services running in the vehicle to enhance drivers' and passengers' experiences, or to utilize and process data stored in the vehicle. *Note: In-vehicle services may be considered part of mobility services.*
Locator ID Separation Protocol (LISP)	A new network protocol that allows for the separation of IP addresses and locations, providing greater flexibility in the routing of communications
Mobile Network	An access network that includes a wireless section *Note: Mobile networks are also called cellular networks.*
Mobile Network Operator (MNO)	Any party that provides one or more cellular network service(s) and exposure of related network APIs
Mobility Service	Various services are provided to end users, such as vehicle drivers and travelers. *Note: Examples of such services include monitoring vehicle and driver conditions, providing driving assistance to warn of fallen objects on the road, confirming safety at low-visibility junctions, and so on.*
Mobility Service Provider (MSP)	Any party that provides one or more mobility service(s)
Multi-access Edge Computing (MEC)	A platform that can be accessed through various access networks and that provides various services to streamline business service development and operations

Term	Definition
	Note: MEC generally corresponds to a near edge integrated with various access networks.
Near Edge	A type of edge platform that is deployed in a larger number than the central clouds, and it sits closer to devices than the central cloud. *Note: The near edge is expected to provide services similar to those of the central cloud, as well as additional services for distributed computing and distributed data management, to streamline system development across a set of near edges.*
Network Function (NF)	A function related to networking, such as routing, device authentication, packet filtering, network address translation, load balancing, VPN gateways, packet encapsulation, etc.
Network Function Virtualization (NFV)	A solution to run various network functions on top of a virtualized environment, such as virtual machines and/or containers
Network Operator (NO)	Any party that provides one or more network service(s)
Network Slicing	A solution framework that is used to establish a private network in the access network(s) among selected user equipment and remote services
Original Equipment Manufacturer (OEM)	A term refers to the companies that design, manufacture, and sell vehicles
Platform	An extended offering on top of the infrastructure that provides various higher-level services to streamline development of the system and system operations
Private Network	A network that connects servers and possibly user equipment in a private and secure manner within the context of the service provision *Note 1: A larger private network may be constructed by combining multiple private sub-networks.* *Note 2: Private networks can be constructed virtually using VLANs, etc.*
Service	A generic concept that represents any business, data usage, computing, storage, networking, and system management services *Note: A service could be a business service offered to external users for monetization, a data analytics service to help internal users conduct their business duties, or a component service, such as data collection, storage, and retrieval, that can be used repeatedly to build these larger services.*
Site	A location in which networking, computing, and storage resources are available and can be used in an integrated manner to build and run various services
Tier 1 Supplier	A company directly supplies major components, systems, or finished goods to OEMs
User Equipment (UE)	A device that is equipped with a communication module and connected to the access network *Note: In many AECC use cases, the UE corresponds to a vehicle.*
Vehicle-to-Infrastructure (V2I)	A term refers to communication and interaction between vehicles and roadside infrastructure that are used to assist safe vehicle driving and enable smooth traffic flow instantly
Vehicle-to-Network (V2N)	A term refers to communication and interaction between vehicles and remote services via at least one, and possibly multiple, access network services supporting a wide geography seamlessly and with sufficiently low latency
Vehicle-to-Vehicle (V2V)	A term refers to communication and interaction between two or possibly more vehicles that must be completed instantly.

Annex E – References

[1] AECC, “General Principles and Vision”, White Paper, Oct. 2024. [Online]. Available: https://aecc.org/wp-content/uploads/2024/10/AECC_General_Principle_and_Vision_v4.0.4_Oct_3.pdf

[2] AECC, “Digital Twins”, White Paper, Aug. 2024. [Online]. Available: https://aecc.org/wp-content/uploads/2024/08/AECC_Digital_Twins_FINAL.pdf

[3] AECC, “Connected Infrastructure for the Realization of the Green Mobility Society”, White Paper, Nov. 2023. [Online]. Available: https://aecc.org/wp-content/uploads/2023/11/20230915_EN_Connected_infrastructure_for_the_realization_of_Green_Mobility_Society.Designed_10.20pdf.pdf

[4] AECC, “Data Management Systems in the Distributed Environment”, White Paper, Sep. 2023. [Online]. Available: https://aecc.org/wp-content/uploads/2023/09/AECC_Data_Management_Whitepaper_FINAL.pdf

[5] ETSI, “Network Functions Virtualisation”, Technology Overview. [Online]. Available: https://www.etsi.org/technologies/nfv

[6] ETSI, “Multi-access Edge Computing (MEC); Framework and Reference Architecture”, Group Specification GS MEC 003 v3.1.1, Jan. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_mec003v030101p.pdf

[7] 3GPP “Edge Computing”, Technology. [Online]. Available: https://www.3gpp.org/technologies/edge-computing

[8] GSMA Operator Platform Group, [Online]. Available: https://www.gsma.com/solutions-and-impact/technologies/networks/operator-platform-hp/

[9] 5GAA, “A visionary roadmap for advanced driving use cases, connectivity, and technologies ”, Nov. 2024. [Online]. Available: https://5gaa.org/content/uploads/2025/01/5gaa-roadmap-iii-wp.pdf

[10] Cloud Native Computing Foundation, “Edge Native Application Principles”, Whitepaper. [Online]. Available: https://www.cncf.io/reports/edge-native-applications-principles-whitepaper/

[11] Linux Foundation, “LF Edge”, Project Webpage. [Online]. Available: https://lfedge.org/

[12] AECC, “Distributed Computing in an AECC System”, White Paper, Sep. 2021. [Online]. Available: https://aecc.org/wp-content/uploads/2021/09/Distibuted_Computing_White_Paper_v1.0.0.pdf

[13] 3GPP, “Architecture enhancements to facilitate communications with packet data networks and applications”, TS 23.682. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.682/

[14] 3GPP, ” Network Exposure Function Northbound APIs”, TS 29.522. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/29_series/29.522/

[15] Y. -B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks: location-based multicast algorithms," Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 1999, pp. 101-110, doi: 10.1109/MCSA.1999.749282.

[16] Young-Bae Ko and N. H. Vaidya, "GeoTORA: a protocol for geocasting in mobile ad hoc networks," *Proceedings 2000 International Conference on Network Protocols*, Osaka, Japan, 2000, pp. 240-250, doi: 10.1109/ICNP.2000.896308.

[17] Z. Wang, Y. Tan and X. Zhang, "Experimental Evaluation of Modern TCP Variants in MEC-enabled Cellular Networks," *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Hangzhou, China, 2018, pp. 1-5, doi: 10.1109/WCSP.2018.8555607.

[18] A. Politis, A. Manitsaris and I. Mavridis, "Implementation and Evaluation of TCP Enhancements over Satellite Links," *2006 International Conference on Wireless and Mobile Communications (ICWMC'06)*, Bucharest, Romania, 2006, pp. 24-24, doi: 10.1109/ICWMC.2006.51.

[19] Microsoft, “Algorithmic improvements boost TCP performance on the Internet,” Technical Report. [Online] Available: https://techcommunity.microsoft.com/blog/networkingblog/algorithmic-improvements-boost-tcp-performance-on-the-internet/2347061

[20] TM Forum, Project ”Network service monetization through standardized APIs”, Catalyst Project. [Online]. Available: https://www.tmforum.org/catalysts/projects/M24.0.707/network-service-monetization-through-standardized-apis-phase-iii

[21] Camara Project Repository “APIBacklog”, GitHub. [Online]. Available: https://github.com/camaraproject/APIBacklog/blob/main/documentation/APIbacklog.md

[22] GSMA and AECC, “GSMA and Automotive Edge Computing Consortium Work Together to Drive Forward Interoperable Connected Vehicle Services”, Press Release. [Online]. Available: https://www.gsma.com/newsroom/press-release/gsma-and-automotive-edge-computing-consortium-work-together-to-drive-forward-interoperable-connected-vehicle-services/

[23] AECC, "Using Opportunistic Data Transfer for Connected Vehicles to Reduce Cell Congestion in Existing Mobile Networks", PoC, 2022. [Online]. Available: https://aecc.org/proof-of-concepts/entry/751/

[24] AECC, "Enabling Distributed Edges for HD Mapping", PoC, 2022. [Online]. Available: https://aecc.org/proof-of-concepts/entry/752/

[25] AECC, "Enabling a Geolocation Parking Service with AECC Distributed Edge Architecture", PoC, 2022. [Online]. Available: https://aecc.org/proof-of-concepts/entry/853/

[26] AECC, "Wi-Fi Data Offloading and Edge Computing for Greener Mobility Services", PoC, 2023. [Online]. Available: https://aecc.org/proof-of-concepts/entry/936/

[27] AECC, "Enabling Trusted HD Mapping Data Updates in a Multi-organizational Distributed Edge", PoC, 2023. [Online]. Available: https://aecc.org/proof-of-concepts/entry/949/

[28] AECC, "Edge Relocation: Optimizing 5G Resources and Edge Networks to Enable Varied Mobility Services", PoC, 2023. [Online]. Available: https://aecc.org/proof-of-concepts/entry/956/

[29] AECC, "Distributed Battery Electric Vehicle (BEV) Range Estimation via Personalized Federated Learning", PoC, 2024. [Online]. Available: https://aecc.org/proof-of-concepts/entry/995/

[30] AECC, "AI Agents Utilizing End-to-End LLMs", PoC, 2024. [Online]. Available: https://aecc.org/proof-of-concepts/entry/1233/

[31] AECC, "Traffic Load Balancing of Edge Server Access", PoC, 2024. [Online]. Available: https://aecc.org/proof-of-concepts/entry/1235/

[32] AECC, "Optimal Edge Selection for Realizing Digital Twins and Green Energy Utilization", PoC, 2025. [Online]. Available: https://aecc.org/proof-of-concepts/entry/1272/

[33] AECC, "Premium Communication Services Utilizing Telco APIs", PoC, 2025. [Online]. Available: https://aecc.org/proof-of-concepts/entry/1276/

Annex F – Contributors

- Lead Editor and Lead Author: Yoshisato Fukatsu (Oracle)
- Contributor – Takamasa Higuchi (Toyota)
- Contributor – Akira Itoh (Toyota)
- Contributor – Lei Zhong (Toyota)
- Contributor – Chunghan Lee (Toyota)
- Contributor – Péter Suskovics (Ericsson)
- Contributor – Mikael Klein (Ericsson)
- Contributor – Mohamed ElGamal (Ericsson)
- Contributor – Takurou Sakai (KDDI)
- Contributor – Inma Carrion (Charter Communications)
- Supervisor – Ryokichi Onishi (Toyota)
- Supervisor – Christer Boberg (Ericsson)