International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

The Role of Cloud Technologies in the Development of the Connected Car Ecosystem and the Internet of Things

Madugula Abhiram*

Software Developer, Saanavi Technologies,Bloomington, Illinois, US
Email: abhirammadugula93@gmail.com

Abstract

The article examines how cloud technologies shape the connected car ecosystem and its interaction with the Internet of Things. Relevance follows from the rapid growth of software-defined vehicles, continuous telemetry streams, and service models based on remote computation, storage, and lifecycle analytics. Novelty is associated with an integrated synthesis that links in-vehicle data acquisition, edge/fog mediation, and cloud-native backends to concrete engineering decisions in automotive software production, with emphasis on Google Cloud Platform–oriented stacks and Java-centric enterprise integration. The work aims to systematize architectural patterns that enable scalable ingestion, secure communication, fleet-level management, and data-driven services across vehicles, roadside infrastructure, and IoT platforms. The study applies analytical review, comparative reasoning, and structured source analysis to examine peer-reviewed research and provider architectures. The conclusion formulates technology implications for platform design, data governance, and deployment practices in automobile software, highlighting trade-offs among latency, resilience, cost, and security. The article targets researchers and practitioners in automotive IT, cloud engineering, and applied AI.

Keywords: connected car; Internet of Things; cloud computing; edge computing; fog computing; vehicle telemetry; cloud-native architecture; Google Cloud Platform; automotive software; software-defined vehicle.

Received: 2/10/2026
Accepted: 4/10/2026
Published: 4/17/2026

---

* *Corresponding author.*

1. Introduction

Connected vehicles increasingly operate as networked computing nodes that exchange data with external services, roadside infrastructure, and device ecosystems, which transforms automotive software from an embedded-only product into a distributed system spanning vehicle, edge, and cloud layers. Cloud platforms support persistent telemetry ingestion, large-scale storage, analytics pipelines, and service delivery channels that extend vehicle functionality beyond production time. At the same time, IoT integration expands the operational perimeter to smart cities and enterprise environments. Real-world connected-vehicle platforms reported in the literature emphasize end-to-end data paths from vehicles to backend systems, as well as the practical constraints imposed by throughput, data heterogeneity, and governance requirements in production deployments.

The article aims to explain and enable the connected car ecosystem and IoT convergence through repeatable architectural patterns and engineering practices aligned with modern automotive software development, with particular attention to Java-based enterprise integration and to deployment logic using the Google Cloud Platform. The objectives are:

1) to describe cloud-centric functional building blocks that support connected-car services across ingestion, processing, storage, and exposure layers;

2) to analyze edge/fog mediation strategies that address latency and mobility constraints in vehicle-centric IoT;

3) to compare implementation implications for platform design, security posture, and operational scalability in automotive software delivery. Novelty follows from consolidating peer-reviewed evidence on connected-vehicle data platforms with provider-grade cloud and IoT reference architectures, then mapping them to concrete design choices relevant to software teams working in automotive domains.

2. Materials and Methods

Materials were selected to represent complementary layers of the connected-car ecosystem: in-vehicle data collection and communication, edge/fog intermediates, cloud backends, and production-grade connectivity patterns. M.F.A. Abdullah surveys edge computing for V2X and frames a three-layer division across vehicle network, edge cloud, and centralized cloud that supports latency-sensitive services near vehicles and batch analytics in the cloud [1]. Amazon Web Services describes MQTT-centered connected-vehicle communication patterns and design guidance for extensible, scalable telemetry exchange between vehicles and cloud backends Reference [2]. K. Behravan reviews fog computing in vehicular networks and systematizes architectures, technical issues, and performance metrics relevant to vehicular offloading and proximity processing [3]. Google Cloud documents connected-device architectures for ingestion and analytics on Google Cloud, positioning IoT backend patterns as decision guides for fleet-scale connectivity and analysis [4]. D.H. Ku proposes a V2X-car edge cloud architecture grounded in cloud-native automation concepts and multi-cluster orchestration for distributed environments [5]. N. Keshari analyzes vehicular fog computing as a paradigm that uses moving and parked vehicles as fog nodes and discusses issues induced by high mobility and dynamic topology [6]. S. Lu and W. Shi articulate vehicle computing as a mobile computing platform concept and discuss enabling technologies and open challenges, supporting interpretation of vehicle–edge–cloud cooperation [7]. A. Mostefaoui and coauthors report a production connected-car big-data platform and describe technologies used to gather, store, process, and exploit automotive data at scale [8]. D. Rocha and coauthors detail a modular in-vehicle C-ITS architecture for sensor data acquisition and cloud connectivity through vehicle networks and interfaces such as CAN/OBD-II [9]. L. Zhu and coauthors study vehicular edge cloud computing with content caching optimization driven by prediction and deep reinforcement learning to reduce access costs under mobility dynamics [10].

Methods: The article relies on analytical literature review, source-based synthesis, and comparative analysis of architectural patterns. In the narrative, comparative reasoning is used to relate the responsibilities of vehicles, edges, fog/cloud responsibilities across the surveyed materials. In contrast, structured source analysis is used to extract recurring components (identity and connectivity, ingestion buses, processing stages, storage tiers, governance controls, and operational automation) and align them with automotive software production constraints (fleet management, Over-The-Air integration surfaces, observability, and lifecycle data governance).

3. Results

The examined sources converge on a layered interpretation of the connected car ecosystem in which a vehicle produces heterogeneous signals, an intermediate proximity tier reduces latency and bandwidth pressure, and cloud backends deliver scalable storage, analytics, and service orchestration. Edge-oriented surveys describe the edge as the contact zone between vehicles and the cloud, where computation and storage close to vehicles support low-latency awareness and local broadcasting, while the centralized cloud supports large-scale processing and analytics [1]. Vehicle-centric computing research extends this view by arguing that connected vehicles themselves behave as mobile computing platforms that collaborate with roadside or cellular edge systems and cloud services, shifting part of the computation to the most suitable tier based on resource and time constraints [7].

A practical connected-car platform requires an acquisition and normalization step that bridges in-vehicle networks with external communication interfaces. A modular Cooperative Intelligent Transport Systems (C-ITS) in-vehicle architecture illustrates how sensor data can be collected from vehicle buses (such as CAN via OBD-II) and auxiliary devices, then forwarded through communication mechanisms toward cloud connectivity [9]. In production big-data platforms, the same acquisition stage expands into a multi-component pipeline that gathers, stores, processes, and exploits automotive data, with explicit attention to technology composition across ingestion, storage, and processing subsystems in real deployments [8]. The synthesis from these sources supports an interpretation that data heterogeneity (signals, diagnostics, events, streams) pushes platform design toward standardized schemas, late-binding enrichment, and multi-tenant governance models that can evolve with fleet growth and new services.

Connectivity patterns form the backbone of the connected-car ecosystem because telemetry exchange, command delivery, and fleet management depend on reliable, scalable messaging. Provider architecture guidance focuses on publish/subscribe communication as a common organizing principle, emphasizing extensible connectivity for telemetry and bi-directional interaction between vehicles and backend services [2]. When aligned with IoT fleet scale, cloud documentation frames multiple ingestion and backend patterns (for example, device-to-broker or device-to-streaming endpoints) and links them to downstream analytics and operations, which supports a design approach where the ingestion layer is explicitly separated from analytics and service backends to improve evolvability and operational isolation [4]. This separation is directly relevant to automotive software teams building Java-centric microservices, because it encourages stable contract boundaries between message ingestion, stream processing, and service APIs, limiting ripple effects when protocols or device firmware evolve.

Latency and mobility constraints are recurring drivers of edge/fog participation. Fog computing surveys emphasize that vehicular networks face stringent delay requirements, heterogeneity, and resource management constraints, motivating the placement of computation and storage closer to vehicles. Vehicular fog computing work notes that high mobility and dynamic topology complicate resource allocation and data retrieval, thereby increasing the value of adaptive scheduling and locality-aware offloading strategies [6]. Cloud-native edge architectures provide operational automation and distributed orchestration (e.g., multi-cluster management logic) to maintain secure deployment and monitoring across geographically dispersed nodes [5]. Taken together, these sources support a design implication: cloud technologies do not replace proximity tiers, but instead coordinate them through policy, observability, and lifecycle management, while latency-sensitive processing and fast control loops remain nearer to vehicles and roadside infrastructure.

Resource efficiency under mobility conditions extends beyond computation offloading to include data placement and decision-making, especially for services that require repeated access to data, models, or task inputs. Vehicular edge cloud research shows that caching strategies benefit from predicting environment dynamics and coordinating decisions across time scales, aiming to reduce access cost under high mobility and fluctuating demand [10]. This connects to the connected-car service layer because many user-visible features (navigation intelligence, perception updates, context-aware infotainment) repeatedly reuse data artifacts, and cloud-centric designs can treat these artifacts as managed assets with controlled distribution and versioning, while edge tiers handle locality and response time constraints. The implication for IoT convergence is that caching and content-placement policies should be unified across the vehicle, roadside, and cloud layers to avoid fragmented behavior and inconsistent service quality.

Figure 1 integrates the reviewed architectural logic into a single conceptual model that reflects how vehicles, edge/fog tiers, and cloud services cooperate in connected-car ecosystems aligned with IoT backends. The diagram is constructed as an original synthesis grounded in the layered descriptions and pipeline components reported across the sources [1–4, 7–10].

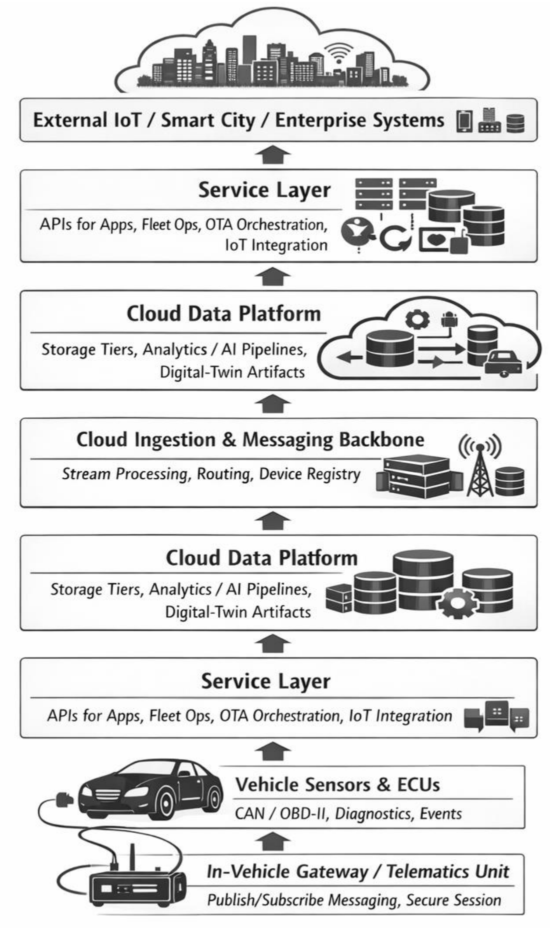

Figure 1: Cloud–Edge–Vehicle reference flow for the connected car ecosystem integrated with IoT backends (synthesized from sources [1–4, 7–10])

Across the reviewed evidence, the cloud tier supports three dominant value-creation domains. First, it enables lifecycle analytics and fleet-scale learning by consolidating data across vehicles and time, which is consistent with production connected-car big-data platforms designed to gather and process massive automotive data volumes Reference [8]. Second, it enables continuous improvement loops in software-defined vehicles by coordinating artifacts (data, configurations, models, software packages) with governance and versioning logic that multiple services can consume and stakeholders can consume system integration because IoT backends and enterprise systems increasingly require standardized interfaces, reliable ingestion patterns, and scalable analytics entry points, which are central themes in cloud IoT architecture guidance.

4. Discussion

The results highlight a consistent tension between centralized scalability and proximity-driven responsiveness. Surveys of vehicular fog and edge computing emphasize that vehicular services often face strict deadlines and mobility-driven instability, motivating the placement of storage closer to vehicles [3, 6]. At the same time, production-grade connected-car platforms emphasize that long-horizon analytics, fleet governance, and platform evolution require centralized data capabilities with explicit technology composition across ingestion, storage, and processing layers [8]. A coherent connected-car strategy treats cloud and proximity tiers as coordinated components rather than competing alternatives, with the cloud orchestrating policy, identity, and lifecycle management while edge/fog tiers handle locality and fast interaction loops (see Table 1).

Table 1: Functional partitioning across Vehicle–Edge/Fog–Cloud layers (compiled from sources [1, 3, 6–8, 10])

Layer	**Dominant responsibilities**	**Engineering implications for automotive software**
Vehicle	Signal generation, local sensing, first-hop preprocessing, safety-critical logic	Stable in-vehicle interfaces; gateway patterns for telemetry export; separation between safety-critical and service telemetry domains
Edge/Fog	Low-latency aggregation, locality-aware offloading, dynamic resource allocation, and caching under mobility	Orchestration across distributed nodes; locality-aware scheduling; resilience under topology changes; observability spanning heterogeneous environments
Cloud	Fleet-scale ingestion, storage, analytics/AI pipelines, cross-vehicle learning, service orchestration	Governance and schema management; scalable stream processing; artifact versioning; multi-tenant security controls; integration APIs for IoT/enterprise

Google Cloud architecture guidance frames connected-device ingestion and backend choices as patterns tied to analytics value extraction and fleet scale [4]. AWS guidance frames vehicle communication around an MQTT publish/subscribe backbone and highlights extensibility and scalability for vehicle telemetry exchange [2]. In practice, both perspectives converge on a decoupled ingestion backbone plus independently deployable services, which suits Java ecosystems because stream consumers, enrichment services, and API gateways can evolve independently while retaining stable message contracts. Such separation is reinforced by in-vehicle modular architectures that treat acquisition as an explicit subsystem, thereby reducing coupling between sensor interfaces and backend analytics pipelines [9] (see Table 2).

Table 2: Cloud platform patterns relevant to connected cars and IoT backends (synthesized from sources [2, 4, 8, 9])

Pattern	**What the pattern achieves**	**Source basis**
Publish/subscribe telemetry backbone	Decouples producers (vehicles) from consumers (services), supports scalable fan-out and evolvable consumers	Vehicle communication guidance
Fleet-scale connected-device backend	Provides decision patterns for device connectivity and analytics entry points at scale	Connected-device architectures
Production data platform for connected vehicles	Operationalizes end-to-end data handling for automotive groups, spanning ingestion to exploitation	Connected-car big data platform report
Modular in-vehicle acquisition with cloud connectivity	Establishes structured first-hop data capture and forwarding from vehicle networks	Modular in-vehicle architecture

The interpretation of edge cloud automation work suggests that the operational model increasingly resembles cloud-native, distributed software rather than isolated embedded deployments. Cloud-native edge proposals emphasize secure deployment automation, dynamic orchestration, and real-time monitoring across distributed nodes [5]. When combined with research on caching optimization under mobility, the evidence supports the view that performance depends on coordinated policy and adaptive placement rather than static offloading rules [10]. For automotive software teams, this shifts competency requirements toward platform engineering, including CI/CD discipline, distributed systems, unified observability, configuration governance, and data lifecycle controls spanning the vehicle and edge layers.

5. Conclusion

Cloud technologies support the growth of the connected-car ecosystem through scalable ingestion backends, fleet-level backends, and service orchestration that extend vehicle functionality throughout its lifecycle. Edge and fog tiers remain structurally linked to connected-car value because mobility and latency constraints demand proximity processing, dynamic resource allocation, and locality-aware caching. The reviewed evidence supports a design stance in which in-vehicle modular acquisition, publish/subscribe telemetry, and decoupled cloud services form a stable foundation for IoT integration and software-defined vehicle evolution. For Java-centric teams operating on Google Cloud Platform, these findings translate into concrete engineering priorities: contract-first messaging, scalable stream consumers, governance-aware data modeling, and operational automation that treats vehicle–edge–cloud as one distributed system rather than separate technology silos.

References

[1] Abdullah, M. F. A., Yogarayan, S., Abdul Razak, S. F., Azman, A., Muhamad Amin, A. H., & Salleh, M. (2023). Edge computing for vehicle to everything: A short review. *F1000Research, 10*, 1104. https://doi.org/10.12688/f1000research.73269.4

[2] Amazon Web Services. (2024, January 12). *Designing next-generation vehicle communication with AWS IoT Core and MQTT* [White paper]. https://docs.aws.amazon.com/pdfs/whitepapers/latest/designing-next-generation-vehicle-communication-aws-iot/designing-next-generation-vehicle-communication-aws-iot.pdf

[3] Behravan, K., Farzaneh, N., Jahanshahi, M., & Hosseini Seno, S. A. (2023). A comprehensive survey on using fog computing in vehicular networks. *Vehicular Communications, 42*, 100604. https://doi.org/10.1016/j.vehcom.2023.100604

[4] Google Cloud. (2024, September 9). *Connected device architectures on Google Cloud*. https://docs.cloud.google.com/architecture/connected-devices

[5] Ku, D., Zang, H., Yusupov, A., Park, S., & Kim, J. (2025). Vehicle-to-everything-car edge cloud management with development, security, and operations automation framework. *Electronics, 14*(3), 478. https://doi.org/10.3390/electronics14030478

[6] Keshari, N., Singh, D., & Maurya, A. K. (2022). A survey on vehicular fog computing: Current state-of-the-art and future directions. *Vehicular Communications, 38*, 100512. https://doi.org/10.1016/j.vehcom.2022.100512

[7] Lu, S., & Shi, W. (2023). Vehicle computing: Vision and challenges. *Journal of Information and Intelligence, 1*(1), 23–35. https://doi.org/10.1016/j.jiixd.2022.10.001

[8] Mostefaoui, A., Merzoug, M. A., Haroun, A., Nassar, A., & Dessables, F. (2022). Big data architecture for connected vehicles: Feedback and application examples from an automotive group. *Future Generation Computer Systems, 134*, 374–387. https://doi.org/10.1016/j.future.2022.04.020

[9] Rocha, D., Teixeira, G., Vieira, E., Almeida, J., & Ferreira, J. (2023). A modular in-vehicle C-ITS architecture for sensor data collection, vehicular communications and cloud connectivity. *Sensors, 23*(3), 1724. https://doi.org/10.3390/s23031724

[10] Zhu, L., Li, B., & Tan, L. (2024). Vehicular edge cloud computing content caching optimization solution based on content prediction and deep reinforcement learning. *Ad Hoc Networks, 165*, 103643. https://doi.org/10.1016/j.adhoc.2024.103643