Red Hat and DXC

Mixed-criticality workloads for modern automotive platforms

Key concepts

ACC - Adaptive cruise control

ASIL - Automotive Safety Integrity Level

AUTOSAR - Automotive Open System Architecture

CARLA - CAR Learning to Act

ECU - Electronic control unit

FFI - Freedom from Interference

HMI - Human-machine interface

HPC - High-performance computing

IC - Instrument cluster

ROS2 - Robot Operating System

SDV - Software-defined vehicle

As part of their ongoing collaboration, Red Hat and DXC have developed a joint demonstrator that integrates Red Hat® In-Vehicle Operating System (OS) with DXC's expertise in advanced automotive applications. This demo can be used as a proof-of-concept to show how container-based automotive software architectures allow safety-critical and nonsafety applications to coexist without interfering with each other. Though it is not intended to be a production-ready solution, it does provide the foundational structure to more efficiently use resources, provide lightweight isolation, and streamline deployment.

Built on Qualcomm's SA8775P platform, the system runs on Red Hat In-Vehicle OS—a safety-certified open source Linux® platform for software-defined vehicles (SDVs) that supports the entire automotive development lifecycle, from early-stage development through to production. It allows safety-critical (up to Automotive Safety Integrity Level (ASIL-B) and nonsafety applications to share a single, ASIL-certified Linux kernel. Native Linux features, including cgroups and namespaces combined with tailored configurations, enforce Freedom from Interference (FFI) requirements. This eliminates the need for separate guest OS instances and significantly reduces startup time and memory footprint, compared to virtual machine (VM)-based approaches which require additional hypervisor overhead and resource allocation.

Containerization further simplifies system configuration, maintenance, and lifecycle management, delivering a scalable and efficient deployment model for in-vehicle applications.

The demonstrator shows how applications such as adaptive cruise control (ACC) and the instrument cluster (IC) can reliably operate alongside nonsafety workloads. DXC has also integrated the Robot Operating System (ROS2). ROS2 supports data handling and orchestration, and demonstrates how 2 open source technologies—ROS2 and Red Hat In-Vehicle OS—can operate together in a containerized, safety-aligned environment.

This demo also serves as a reference architecture to help original equipment manufacturers (OEMs) explore container-native approaches for deploying mixed-criticality workloads in modern (and future) vehicles.

Industry challenge

Automotive software architectures have evolved significantly, transitioning from distributed electronic control units (ECUs) toward centralized high-performance computing (HPC). This shift enables the consolidation of workloads and offers greater computational power, but also introduces new challenges. Making optimal use of shared resources when running mixed workloads is one such challenge, and includes safety-critical (ASIL) and nonsafety (QM) applications on shared hardware.

Traditionally, virtual machines (VMs) have been used to isolate mixed workloads and consolidate legacy systems. While effective, VMs can introduce integration complexity, often requiring large, specialized teams to continuously integrate, test, and troubleshoot across multiple system layers and technologies.

To address these challenges, modern automotive architectures are increasingly adopting container-based approaches. Containers are a lightweight and modular alternative to VMs, enabling more efficient scaling and use of resources, as well as streamlined deployment.

Red Hat In-Vehicle OS takes advantage of both container technologies and Linux kernel capabilities to offer Freedom from Interference (FFI) between safety-critical and non-safety workloads—helping OEMs meet safety goals while maximizing the efficiency of shared HPC platforms.

Demonstration of isolated safety and non-safety workloads

The following applications are included in the demo:

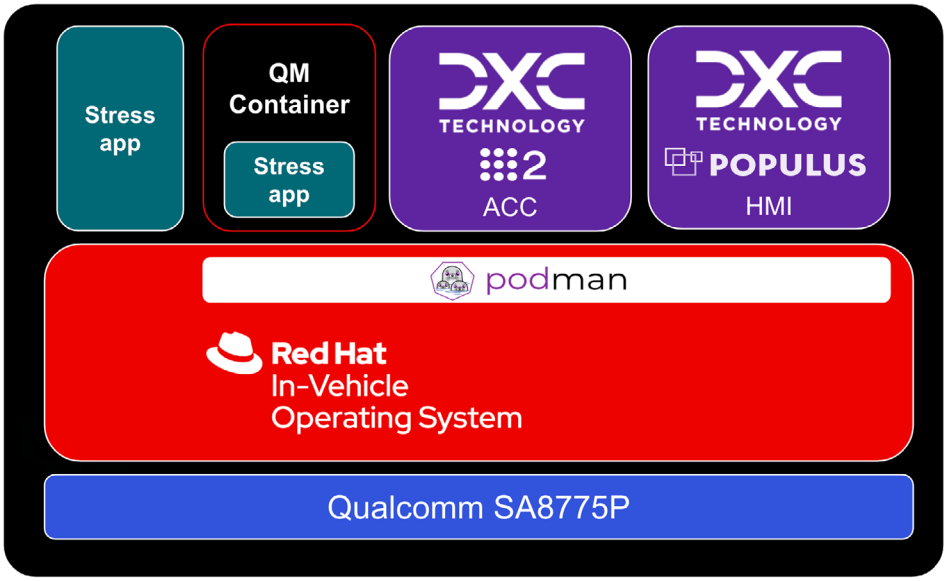

Figure 1. Architecture diagram of demo setup.

- **Adaptive cruise control application:** A safety-critical workload that processes simulated sensor data from CARLA (which stands for "CAR Learning to Act") to maintain vehicle distances. The ACC logic is implemented as a standalone C++ library, integrated into the system via a ROS2 wrapper node.
- **Instrument cluster application:** A safety-critical workload featuring DXC's Populus human-machine interface (HMI) technology displaying real-time vehicle data.
- **Non-safety stress application for testing:** To demonstrate the enforcement of temporal FFI, a central processing unit (CPU) stress application was deployed in 2 configurations, which temporarily generates a high CPU load during execution:

- Inside a container, alongside safety-critical workloads, where it had no observable temporal impact, demonstrating effective isolation.
- Directly on the host OS, bypassing container isolation, where it interfered with safety workloads, resulting in system-level unreliability and degraded ACC behavior.

This comparison underscores the role of containerization in maintaining predictable performance in mixed-criticality environments.

Red Hat In-Vehicle OS is certified for mixed-criticality environments, offering comprehensive isolation features including temporal, spatial, and resource isolation, as well as kernel-level protections. This specific demonstration primarily focuses on temporal isolation capabilities, though the system provides comprehensive mixed-criticality support.

Advantages of the pre-integrated demonstrator

This preintegrated demonstrator highlights the following strengths:

- **Faster prototyping:** Provides a sample reference architecture that accelerates early-stage exploration of containerized automotive workloads.
- **Simplified integration:** Enables OEMs and automotive Tier-1 software suppliers to evaluate container-based separation of mixed-criticality applications without starting from scratch.
- **Safety-aligned design:** Demonstrates a containerized approach aligned with temporal FFI principles, using a safety-certified Linux kernel.
- **Engineering expertise:** Demonstrates DXC's ability to integrate open source middleware, sample applications, and mixed-criticality workloads into a containerized automotive demonstrator.

ROS 2 integration and application architecture

A core part of the demonstrator is the successful integration of ROS2 with Red Hat In-Vehicle OS, despite the lack of pre-built ROS2 packages optimized for safety-certified ARM-based automotive Linux platforms. Key ROS2 components—such as rclcpp and rclpy—were manually compiled and containerized using Podman. The resulting container image, hosted in the DXC Artifactory, enables rapid deployment of simulation and control applications and serves as a foundational environment for developers.

Sample ACC application architecture

The sample ACC application was designed with portability in mind:

- Core logic is implemented as a standalone C++ library.
- ROS2 wrapper node:
  - Converts incoming ROS messages into native data structures.
  - Executes the ACC control logic.
  - Publishes results to ROS2 topics for downstream consumption.

This architecture enables:

- **Framework agnosticism:** The logic can be reused across different middleware stacks.
- **Improved certifiability:** The core control logic can be evaluated independently of ROS2.
- **Isolated testing:** Developers can validate functionality without dependency on full system integration.
- **Future migration:** The decoupled design eases porting to platforms like Adaptive Automotive Open System Architecture (AUTOSAR).

This successful integration demonstrates that, even in the absence of native ROS2 support for ARM-based platforms, open source technologies can be effectively adapted to meet the demanding requirements of modern automotive systems.

Strategic partnership: Red Hat and DXC

The combined strengths of Red Hat and DXC offer automotive OEMs a unique value proposition.

Red Hat brings:

- Proven expertise in open source technologies.
- A safety-certified Linux foundation tailored for software-defined vehicles.
- Container orchestration that scales across development pipelines and vehicle fleets.
- A certified approach to mixed-criticality workload management, enabling consistent software practices from cloud to vehicle.

DXC contributes:

- Deep automotive expertise.
- Comprehensive safety knowledge.
- Real-time software proficiency.
- Strong capabilities in automotive security and middleware solutions.
- Robust and scalable implementations that meet the evolving demands of modern vehicle architectures.

Together, they provide OEMs and Tier 1 suppliers with a practical foundation to explore container-native, safety-aligned architectures—accelerating early-stage development, improving integration efficiency, and reducing time-to-market.

Summary

The automotive industry is rapidly transitioning from distributed ECUs to centralized HPC platforms, introducing complex challenges in managing mixed-criticality workloads, ensuring safety, and optimizing system resources. Traditional virtualization methods often fall short due to integration complexity and inefficiency.

Red Hat and DXC's strategic partnership addresses these challenges by combining Red Hat In-Vehicle OS with DXC's deep expertise in automotive software, middleware, and safety systems. Their collaboration is exemplified in a joint demonstrator built on Qualcomm's SA8775P platform, showcasing how safety-critical and nonsafety workloads can reliably coexist on a shared Linux kernel while maintaining temporal FFI.

This proof of concept eliminates the need for traditional virtualization by demonstrating that temporal FFI can be enforced through container-based separation, providing OEMs with a scalable and certifiable path to deploy mixed-criticality workloads on modern automotive hardware.

Learn more

For more details about Red Hat automotive offerings or to schedule a demonstration, contact the Red Hat Automotive team.

Learn more about the mixed-criticality capabilities of Red Hat In-Vehicle OS.

Discover DXC's automotive expertise.

About Red Hat

Red Hat helps customers standardize across environments, develop cloud-native applications, and integrate, automate, secure, and manage complex environments with award-winning support, training, and consulting services.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America
1 888 REDHAT1
www.redhat.com

Europe, Middle East, and Africa
00800 7334 2835
europe@redhat.com

Asia Pacific
+65 6490 4200
apac@redhat.com

Latin America
+54 11 4329 7300
info-latam@redhat.com

redhat.com
#2398151_0725