Software Configuration As A Service (SCAAS): Scalable Runtime Tuning For Mass-Customized Vehicles

Ronak Indrasinh Kosamia

Atlanta, GA
ronak.kosamia@medtronic.com
0009-0004-4997-4225

Abstract:
Software-defined vehicles, in general, and mass customization in the automotive industry, in particular, have encouraged the need to implement dynamic, scalable software configuration methodologies. This paper presents a new framework, called Software Configuration as a Service (SCaaS), which provides scalability run time tuning and configuration provisioning with lifecycle management of a large vehicle fleet. SCaaS application of traditional DevOps concepts into the automotive sphere enables Original Equipment Manufacturers (OEMs), and Tier-1 suppliers to inject, modify, or retire software configuration at run time, per-vehicle. SCaaS, by decoupling feature enablement and monolithic software updates, enables real-time targeted deployment of vehicle-specific configurations to minimize the friction and accelerate the rollout of features. In addition, SCaaS opens new revenue streams with the Feature on Demand (FOD), which can phrase customer personalization and OEM profitability. The suggested architecture guarantees secure runtime-safe configuration management, which makes SCaaS the critical driver of the mass-customized, software-centric automotive ecosystem in the future.

Keywords: Software-defined vehicles, runtime configuration, mass customization, Feature on Demand, DevOps in automotive.

INTRODUCTION

The automotive industry is also rapidly transforming towards software-defined vehicles, as more capabilities become controlled by software, and not predefined hardware components. The current automobiles come with complex software stacks including infotainment systems, advanced driver assistance features, connectivity modules, and the powertrain control systems. A similar increase in demand with the evolution of these systems is mass customization, where particular vehicles in a fleet can be customized in feature sets or configurations to meet the customer-specific needs or market conditions.

Traditional software update schemes in automobiles leverage over-the-air (OTA) systems that, in the context of updating firmware or software in multiple systems, can be successful, but are not flexible or granular enough to update specific features. Moreover, traditional configurations cannot promptly adapt to market conditions, customer reviews, or vehicle operational data information using the capabilities of OEMs.

This paper describes Software Configuration as a Service (SCaaS), a scalable, DevOps-style, time-tested runtime configuration management system, to address such issues on mass-customized vehicles. SCaaS applies continuous integration, deployment, and configuration management beyond traditional IT infrastructure by applying it to the automotive realm. By leveraging cloud-native control layers and vehicle-resident runtime interpreters, SCaaS facilitates the secure and dynamic injection, modification, and retirement of software configurations across fleet-sized deployments.

Unlike conventional OTA systems that distribute full software builds, SCaaS enables lightweight, targeted configuration changes without interrupting vehicle operation. This approach reduces the deployment overhead, minimizes downtime, and enhances the capability to deliver differentiated features to specific vehicle IDs in real time. Additionally, SCaaS supports new business models such as Feature on Demand (FOD), where advanced functionalities can be activated post-purchase, contributing to ongoing revenue streams for OEMs and Tier-1 suppliers.

The remainder of this paper outlines the state of current vehicle software configuration methods, details the SCaaS architecture, explores technical mechanisms for secure runtime tuning, and discusses the operational and commercial benefits of deploying SCaaS in future automotive platforms.

SCAAS: CONCEPT AND ARCHITECTURE

Software Configuration as a Service (SCaaS) introduces a scalable, secure, and lifecycle-managed approach for runtime software configuration within mass-customized vehicles. The SCaaS will enable OEMs and Tier-1 suppliers to dynamically control vehicle capabilities, operational configurations, and software behaviors by dynamically injecting particular configurations (independent of systematic upgrade) with concepts of DevOps and cloud-native system architecture [1].

A. Concept Overview

SCaaS builds on historic vehicle software administration to separate feature commissioning and configurational regulation, such as monolithic firmware or software releases. SCaaS is used to distribute lightweight, structured configuration packages, instead of complex software builds, which can be securely distributed to individual vehicles based on unique identifiers, operational conditions, or customer requiremets [2].

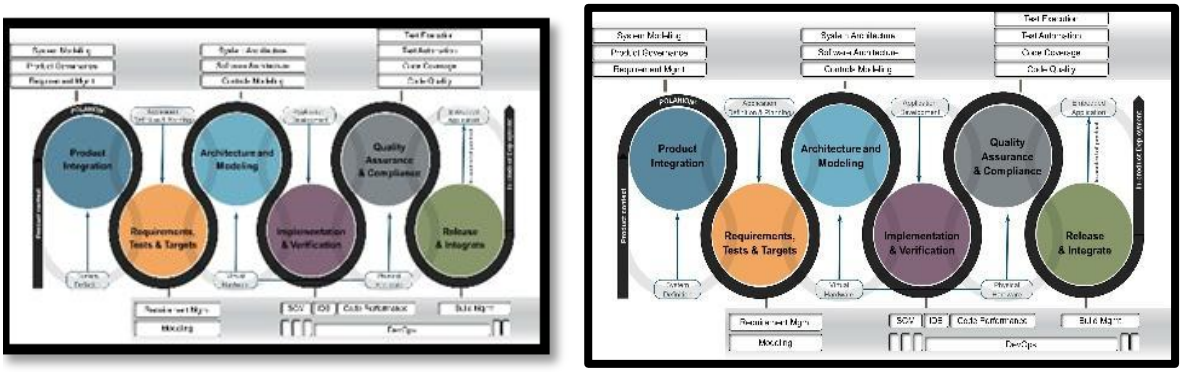

Fig 1: Coordinating Automotive

With SCaaS, OEMs have the potential to:

- Runtime updates or inject software configurations based on vehicle ID
- Withdraw or decommission configurations per operational feedback or policy changes
- Easily manage configuration lifecycles in fleet-scale deployments with minimal disruption
- Assist in revenue collection via Feature on Demand (FOD) by acquiring silent functionality after sale.

This model fits both the increased market demand to personalize software-adjusted vehicle experiences and the operational requirements to ensure the integrity of the system and dependable regulatory compliance.

B. System Architecture

The SCaaS architecture comprises three major layers, namely the Cloud Control Layer, the Vehicle Runtime Layer, and the Secure Communication Interface, conceptually depicted below.

1) Cloud Control Layer

The Cloud Control Layer serves as the central configuration management system and is involved in the following operations:

- Configuration package storage and versioning
- Configuring lifecycle policies (deploy, monitor, retire)
- Configuration of mapping to individual vehicle IDs or fleet segments
- Offering administrative interfaces to OEM and Tier-1 engineering departments [3].
- Tracking the deployment status and vehicle telemetry data

This layer can be integrated with existing OEM backend systems (such as production databases, FOD platforms, and operational analytics tools) to ensure continued alignment with the remaining software development and deployment pipelines.

2) Vehicle Runtime Layer

The Vehicle Runtime Layer is embedded in every vehicle and manages:

- Safely receiving configuration files in the Cloud Control Layer
- Runtime interpretation and application of configuration parameters
- Verification that configurations are compatible with new software modules
- Providing failsafe against incorrect or incompatible configurations
- Enabling rollback or retirement of configurations without vehicle shutdown

The layer acts independently of the core safety-critical systems to maintain system integrity and provide runtime flexibility. It can resort to lightweight containerization or sandbox environments to separate configuration modification and primary control modules.

3) Secure Communication Interface

The communication interface between the Cloud Control Layer and Vehicle Runtime Layer is secure and robust, with encryption, authentication, and integrity mechanisms in place to guarantee:

- End-to-end confidentiality and integrity of configuration packages
- Vehicle identity validation and cloud service authorization
- Strength to withstand malicious manipulation or unauthorized access

It employs industry-standard protocols, including Transport Layer Security (TLS) and Public Key Infrastructure (PKI), to ensure security in transit and at rest [4].

C. Scalability Considerations

SCaaS is configured to deploy at a fleet scale, serving millions of vehicles in multiple geographical areas and network conditions. Scalability is implemented by:

- Regional deployment control using distributed cloud infrastructure
- Technologies to control configuration propagation rates using incremental rollout mechanisms
- Operational visibility with real-time monitoring dashboards
- Telemetry feedback loops to measure the effectiveness of configurations and identify anomalies

In addition, the SCaaS architecture supports heterogeneity in vehicle platforms, software versions, and hardware configurations by abstracting configuration definitions to modular, platform-free schema.

3.4 Configuration Lifecycle Management

One of the primary innovations of SCaaS is the lifecycle-controlled configuration process, which incorporates:

- **Injection:** Safe, on-demand deployment of new settings to chosen vehicles
- **Validation:** Automated compatibility analysis and online safety checks
- **Monitoring:** Ongoing analysis of vehicle actions after configuration
- **Retirement:** Inactivation or deletion of configurations according to feedback or policy
- **Rollback:** Restoration of a previous configuration upon failure

Such a formalized process is done to make sure that configuration modifications supplement functionalities without jeopardizing safety or functionality.

SCaaS can therefore offer a strong research base to scale, runtime-safe, and revenue-producing configuration management in next-gen, software-intensive motorized systems.

TECHNICAL MECHANISMS AND RUNTIME TUNING DETAILS

Software Configuration as a Service (SCaaS) requires a high-reliability, secure, and efficient suite of technical facilities to support real-time configuration of vehicle functions without compromising system stability or exceeding safety limits. The main technical processes behind SCaaS, such as configuration injection, runtime

application, security mechanisms, and real-world examples of tunable vehicle functions, are presented here [5].

A. Configuration Injection per Vehicle ID

SCaaS uses individual vehicle identification to provide highly specific package configurations. Individual vehicles in a fleet share a globally unique vehicle identifier (GUID), typically using existing telematics implementations or embedded hardware security modules.

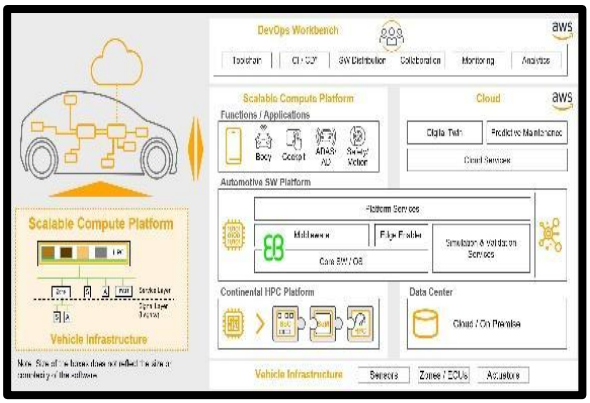

Fig 2: Scalable computer platform

The configuration injection mechanism works as follows:

I. **Identification and Target Mapping:** The Cloud Control Layer identifies configuration packages and sets them to particular vehicle IDs by criteria feature eligibility, hardware capabilities, regulatory constraints, or customer subscriptions.

II. **Package Generation:** Configuration data is encapsulated in secure, signed packages containing:

- Parameter definitions and permissible value ranges
- Versioning metadata
- Integrity verification checksums
- Failsafe rollback triggers

III. **Secure Transmission:** The information is sent via encrypted channels to the intended vehicle using mutual authentication protocols to avoid unauthorized opening or impersonation.

IV. **Reception and Staging:** The Vehicle Runtime Layer receives and validates the integrity of the package, and stages it to be run.

This specialized injection system lets each vehicle have a granular level of configuration, allowing mass customization at the unit level.

B. Runtime-Safe Configuration Application

Executing configurations on-the-fly in a vehicle demands strong guardrails to preserve operational stability, especially in systems that interface with safety-critical devices. SCaaS integrates multiple runtime security precautions:

- **Compatibility Validation:** With a configuration package loaded, the Vehicle Runtime layer validates parameter definitions against the installed software stack, hardware profile, and regulatory limitations in the vehicle.
- **Sandboxed Execution:** Configurations are first tested in isolated runtime environments to determine their implications prior to making changes to operational vehicle systems.
- **Progressive Activation:** SCaaS enables progressive configuration rollout of complex features or performance changes, using change in a cumulative fashion to ensure unintended side effects.
- **Fallback Mechanisms:** The system applies fallback mechanisms that ensure the execution rollbacks to a safe configuration in the case of anomalies, instability, or incompatibility.

These mechanisms provide that such configuration changes can be made without jeopardizing system integrity or user safety.

C. Security and Authentication Framework

To protect configuration processes, SCaaS incorporates stringent security measures:

- **End-to-End Encryption:** Encrypted communications are used between the Cloud Control Layer and Vehicle Runtime Layer, using high standards in cryptographic algorithms to resist eavesdroppers or manipulation.
- **Digital Signatures and Package Integrity:** Authorized OEM or Tier-1 parties will digitally sign configuration packages, allowing recipient vehicles to validate authenticity and integrity before applying configuration.
- **Mutual Authentication:** Vehicle and cloud endpoints authenticate each other via PKI-based credentials, and only trusted parties are capable of triggering or gaining configuration updates.
- **Tamper Detection:** Down in the vehicle, embedded sensors and software detect any attempt to change configurations improperly, raising alerts or entering failsafe modes where warranted.

This multi-layered approach to security helps reduce cyber threats and guarantees that configuration changes are done only in controlled and authorized situations.

D. Tunable Vehicle Features and Runtime Adjustments

SCaaS enables runtime variability in a wide range of vehicle functions, including operations efficiencies and user-devoted customization. Examples of applications:

- **Performance Modes:** It is the dynamic adaptation of powertrain response curves, throttle maps, or suspension tuning to driving conditions or driver preference.
- **Infotainment Features:** Enabling or changing navigation, multimedia, or connected package suites.
- **Comfort and Convenience Systems:** Allowing functions like heated seats, automatic climate control, or ambient lighting customization.
- **Advanced Driver Assistance Systems (ADAS):** Real-time parameter adjustment of lane-keep assist, adaptive cruise control sensitivity, or collision warning thresholds.
- **Feature on Demand (FOD) Enablement:** Activation after the sale of inactive hardware features to include better lighting packages, parking assist modules, or premium connectivity services.

These examples demonstrate how SCaaS can provide a fine-grained level of control, enhance customer experience as well as generate novel revenue sources to OEMs without sacrificing high safety and security levels.

POTENTIAL BENEFITS AND BUSINESS IMPLICATIONS

Both major vehicle manufacturers and end users can receive significant advantages out of Contracting Software Configuration as a Service (SCaaS), opening new frontiers of flexibility, scalability, and profit potential in the new world of software-defined vehicles. Implementing lifecycle managed, runtime configuration control via SCaaS addresses serious technical and business issues of fleet on scale and mass customization.

A. Mass Customization at Scale

Modern car industry needs increasingly personalized car specifications and the customer wants customisation that caters to its wants or needs. Nevertheless, traditional production lines are constrained in the possibility to reflect high feature variation through hardware, logistics and hard-coded baseline software.

SCaaS removes these limitations by decoupling any physical commitment of manufacturing and differentiation of the feature. OEMs can deploy standardized platforms with latent software-controlled features instead of developing specific hardware variants per configuration set. SCaaS can then facilitate the provision to act or change these features selectively at runtime, based on vehicle ID, without physical intervention.

This methodology supports:

- Individual sets of features based on customer preferences
- Adjustments to regulatory environments specific to regions
- Postpartum customization based on the changing needs of the user
- Nimble response to changing market trends or competitive factors

The SCaaS supports mass customization using software, thereby minimizing manufacturing complexity and maximizing product differentiation.

B. Accelerated Feature Rollout and Reduced Development Friction

The common software deployment pattern in the automotive industry includes long developmental cycles, in-depth validation, and updating the entire system, which may result in delayed feature releases and higher operational expenses. SCaaS establishes a faster configuration management paradigm, in line with DevOps, enabling:

- Quick implementation of new features or efficiencies
- Real-world incremental testing and validation
- Iterative configuration of vehicle systems to more effectively accommodate the intended uses
- Effective roll back strategies in the event of performance degradation or unexpected problems

This lean process leads to more productive engineering, faster delivery of new features to market, and enhanced performance of the entire vehicle during the operation phase.

C. Revenue Generation through Feature on Demand (FOD)

SCaaS enables Feature on Demand (FOD) business models by providing capabilities that OEMs and Tier-1 suppliers can monetize on optional or dormant vehicle capabilities after sale. Manufacturers can provide:

- Paid subscriptions to premium features
- Hardware-specific functionalities activated once only
- Promotion feature unlocks or time-limited trial Activation
- Selective activation of features according to local markets

The model offers an ongoing revenue stream beyond the sale of a vehicle, is in line with changing customer demand toward digital services, and contributes to the financial feasibility of the software-driven automotive ecosystem.

D. Operational Efficiency and Lifecycle Optimization

In addition to business advantages, SCaaS also enhances operational efficiency:

- Minimizing the necessity of huge, disruptive OTA updates
- Reducing vehicle unavailability during software updates
- Allows real-time tuning of performance using telemetry data
- Enabling predictive maintenance and system optimization with dynamic configuration changes

These advantages together make SCaaS an effective strategic initiative to improve product quality, operational resiliency, and customer satisfaction in the context of connected, software-defined vehicle ecosystems.

CONCLUSION

Incremental complexity of SDC vehicles, the need to provide mass customization and ongoing feature improvement requires new ways of managing software and configuration. Software Configuration as a Service (SCaaS) proposes a scalable, secure, and managed lifecycle of the framework of the run-time tuning and adjustments of a large vehicle fleet by enabling OEMs and Tier-1 suppliers to dynamically inject, update, and remove configurations in a per-vehicle context.

SCaaS, in contrast to conventional, monolithic software update systems, combines lightweight configuration bundles, DevOps-driven controls, and secure communications to enable on-demand, vehicle-specific modifications in real time without perturbing system stability. This methodology allows swift feature deployment, scalable business operation, and after-sale customization, as well as accommodating new business models with features on demand (FOD) capability.

SCaaS technical architecture responds to the key needs in scalability, safety of runtime operations, and cybersecurity to ensure that configuration changes can be implemented effectively and safely both at fleet scale. Additionally, the proposed system would increase engineering efficiency, minimize development friction, and offer an avenue towards future capabilities like optimization of configuration through AI.

While challenges related to safety-critical systems, dependency management, cybersecurity, and regulatory alignment remain, SCaaS represents a significant step toward fully realizing the potential of software-defined,

mass-customized automotive ecosystems. By extending proven DevOps principles into the automotive domain, SCaaS enables a more agile, customer-centric, and revenue-generating approach to vehicle configuration and feature management.

REFERENCES:

[1] H. A. Tran, D. Tran, and A. Mellouk, "State-dependent multi-constraint topology configuration for software-defined service overlay networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 1986–2001, Oct. 2022.

[2] R. H. Jhaveri, S. V. Ramani, G. Srivastava, T. R. Gadekallu, and V. Aggarwal, "Fault-resilience for bandwidth management in industrial software-defined networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3129–3139, Oct. 2021, doi: 10.1109/TNSE.2021.3097079.

[3] G. P. Espinel, J. L. C. Medina, M. J. F. Calero, and M. Urbieta, "Software configuration management in software product lines: Results of a systematic mapping study," *IEEE Latin America Transactions*, vol. 20, no. 5, pp. 718–730, May 2022, doi: 10.1109/TLA.2022.9693556.

[4] P. Laclau, S. Bonnet, B. Ducourthial, X. Li, and T. Lin, "Predictive network configuration with hierarchical spectral clustering for software-defined vehicles," in *Proc. 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy, Jun. 2023, pp. 1–5, doi: 10.1109/VTC2023Spring57469.2023.10199920.

[5] X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong, "Resource allocation with workload-time windows for cloud-based software services: A deep reinforcement learning approach," *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1871–1885, Apr. 2022, doi: 10.1109/TCC.2022.3155041.