Public Report

Project Identity

- **Project Title:** Integrate Onboard Onboard API for React Native Applications
- **Registration Number (Diarienummer):** 2024-03984
- **Coordinator:** Volvo Technology AB
- **Project Partner:** Diffo Solutions
- **Project Period:** 2024-11-26 to 2025-10-31

1. Executive Summary

This project shows that a third-party developer can build, test, and validate a vehicle-integrated application without needing continuous access to physical ECUs or test vehicles. The key enabler is a cloud-based development environment combined with Volvo's Onboard API — a set of interfaces available in the Service and Entertainment Module (SEM), Volvo Trucks' Android Automotive based in-cab headunit.

Diffo Solutions, a Finnish startup specializing in driver performance and transport profitability analytics, integrated their existing React Native application with Volvo's Onboard API. The integration was done in two phases:

1. **Phase 1:** GPS location and fuel consumption data via the API, replacing the higher-latency Volvo Connect cloud path.
2. **Phase 2:** Trigger and push behavior for real-time data events.

Previously, Diffo received vehicle data through Volvo Connect (a cloud-to-cloud integration). That path introduces latency that is less ideal for real-time driver coaching and trip analysis - according to Diffo Solutions. By moving data access onboard — directly from the vehicle's ECUs through the API — the project eliminated those delays and enabled use cases such as automatic trip detection, real-time fuel efficiency, and loading/unloading event recognition.

The project also evaluated the use of a cloud-based virtual development environment to allow third parties to test their apps against a simulated SEM headunit without physical vehicle access. This approach shortens development lead time, improves software quality through broader test coverage, and lowers the investment needed for service development — all critical for growing the ecosystem and innovative third-party services on Volvo Trucks' platform.

This report presents the findings organized around five key decisions, each backed by at least ten external references from standards bodies, regulators, industry analysts, and technology providers.

2. Public Project Summary

The project verified that an external partner (Diffo Solutions) can develop and test an in-vehicle service against Volvo's Onboard API in a simulated development setup, without requiring physical aftermarket hardware or dedicated truck bench access. Integration between Diffo's React Native app and the vehicle-facing API on the SEM headunit demonstrated low-latency data access — a major improvement over the cloud path through Volvo Connect.

The delivery followed an agile, two-phase approach:

1. Baseline integration for GPS and fuel telemetry via API.
2. Implementation of trigger and push behavior for real-time data events.

Volvo provided the API library, the test environment, and technical support. Diffo adapted their cross-platform React Native application to use the onboard API. The collaboration model proved operationally effective for OEM–third-party co-development and supports a scalable ecosystem model where partners can self-serve using documentation and cloud-based test environments.

3. Why This Matters for Software Defined Vehicles (SDV)

3.1 The shift to software-first truck development

Modern commercial vehicles are becoming Software Defined Vehicles (SDV). This means that a growing share of the vehicle's value comes from software features delivered after the truck leaves the factory — through over-the-air (OTA) updates, service bundles, and data-driven features. According to a Deloitte study of 160+ automotive executives across 8 countries, **81% of OEM fleets are expected to be software-defined by 2030**, with OEMs investing up to **US$3 billion each** in SDV research and development [D1-10].

McKinsey estimates the global automotive software and electronics market will grow to **$519 billion by 2035** at a 4.5% compound annual growth rate, with operating system and middleware segments growing even faster at ~16% CAGR [D1-5].

3.2 Regulatory drivers

This shift is reinforced by new regulations:

- **UNECE Regulation No. 156** (2021) requires a Software Update Management System (SUMS) for vehicle type approval. OEMs must demonstrate they can manage OTA updates securely throughout the vehicle's lifecycle [D1-1].
- **UNECE Regulation No. 155** (2021) requires a Cybersecurity Management System (CSMS), linking cyber risk management to vehicle operations and making continuous vulnerability handling mandatory [D1-2].

- **ISO 24089:2023** specifies requirements for software update engineering for road vehicles, covering infrastructure and deployment of software update packages [D1-3].

These regulations mean that controlled, frequent, and traceable software iteration is not optional — it is a compliance requirement.

3.3 The problem for third-party developers

For third-party developers like Diffo Solutions, the historical barriers to truck-integrated development are real:

- Physical access to test trucks or rigs is limited and only available at specific locations.
- Evaluating partner solutions requires Volvo engineers on-site, creating low transparency for the partner.
- The traditional process is not scalable — it cannot support a growing ecosystem of third-party app providers.
- Partners often default to aftermarket hardware (separate screens - phones or tablets), which are less safe for drivers and harder to maintain.

These problems are directly described in the project's Statement of Work and confirmed during the project.

3.4 Volvo Trucks' in-vehicle platform: SEM and Android Automotive OS

Volvo Trucks' in-vehicle application platform is built on the **Service and Entertainment Module (SEM)** — an Android Automotive OS (AAOS) based headunit integrated into the truck cab. The SEM comes in two hardware versions:

- **SEM 2.0:** Intel x86_64-based, running Android Automotive OS.
- **SEM 2.5:** Qualcomm ARM64-based, running Android Automotive OS.

Android Automotive OS (AAOS) is a full-stack operating system designed by Google specifically for vehicles, running directly on the in-vehicle hardware. Unlike Android Auto (which mirrors a phone), AAOS runs natively in the truck and supports third-party apps including media, communication, navigation, and specialized fleet management applications [D3-4].

The Volvo Trucks Developer Portal (https://developer.volvotrucks.com) provides guides for third-party developers to make their Android apps compatible with the SEM platform, architectural requirements, and testing guidance [D3-7].

4. Cloud-Based High-Fidelity Environment: Technical Background

4.1 Core concept

The central idea is simple: instead of requiring a physical truck or bench rig to test an app, developers use a cloud-hosted virtual representation of the SEM headunit and its connected ECUs. This virtual environment reproduces the key behaviors a third-party app needs to integrate against:

- **API behavior:** The API response format, error codes, authentication, and signal values behave like the real onboard system.

"High fidelity" in this context does not mean full vehicle dynamics simulation. It means the virtual environment is good enough for a third-party app developer to confidently validate their integration before ever touching a physical truck.

4.2 Why this matters for Volvo Trucks' ecosystem

It is difficult and time consuming to test on the real HW or rigs (test trucks). Physical access to the test environment is currently only available in Sweden. The current process "is not scalable to grow the ecosystem and does not support lifecycle management for partners.

A cloud-based environment solves this by:

1. **Removing geographic barriers:** A developer in Finland (like Diffo) can run integration tests against a virtual SEM without traveling to Sweden.
2. **Enabling parallel development:** Multiple third-party partners can develop simultaneously without competing for the same physical truck.
3. **Supporting CI/CD pipelines:** Automated tests can run on every code commit, catching regressions early.
4. **Lowering entry costs:** Partners do not need to invest in their own test hardware.

The project evaluated **Amazon Virtual Engineering Workbench** as a cloud-hosted environment for this purpose.

4.3 Industry context: virtual ECU and cloud simulation tools

The approach taken in this project aligns with mainstream automotive practice. Leading tool providers offer cloud-scalable virtual ECU and simulation-in-the-loop (SiL) environments:

- **Google Cuttlefish** is a configurable virtual Android device that runs in the cloud, enabling Android Automotive development and testing without physical hardware [D2-8].

For AAOS specifically, the Android Emulator supports generic testing of the graphical interface (Volvo Trucks Developer Portal recommends a 9-inch, 1280x720 landscape configuration matching the SEM side display). However, as the Developer Portal notes: "It is not possible to

test any vehicle dependent functionality using the Android Emulator." This is precisely the gap that a high-fidelity cloud simulation of the SEM and its connected ECUs (Onboard API) would fill.

4.4 Architecture pattern

The development and test architecture used in this context follows this pattern:

1. **Cloud developer workspace:** Reproducible build/test environment for React Native, Android SDK, and integration harnesses.
2. **Virtual SEM/API layer:** Simulated Onboard API behavior approximating the real onboard data services (GPS, fuel, triggers).
3. **Contract and integration testing:** API shape validation, trigger behavior verification, and negative-path testing.
4. **Quality gates:** Automated pass/fail checks in the CI pipeline.
5. **Selective physical validation:** Final confidence checks on a real truck for scenarios that cannot be fully simulated.

This pattern reduces dependence on scarce physical trucks while preserving a path to physical confirmation where needed.

5. Project Execution and Outcomes

5.1 Execution model

The project used agile coordination between Volvo GTT and Diffo Solutions with weekly synchronization and a clear responsibility split:

- **Volvo:** Onboard API library, test environment access, platform support, and technical guidance.
- **Diffo:** App adaptation to use the onboard Onboard API, integration implementation, and iterative validation in the Android Automotive context.

5.2 Delivered scope

- Integrated vehicle data access (GPS position and fuel consumption) from the React Native app context via the onboard Onboard API, replacing the higher-latency Volvo Connect cloud path.
- Implemented trigger and notification logic to support real-time driver-facing events (trip start/end detection, fuel efficiency alerts).
- Verified the application in an Android Automotive-aligned environment compatible with the SEM headunit.
- Delivered a PUGH matrix evaluating the onboard Onboard API integration against aftermarket alternatives.

5.3 Main outcomes

1. **Technical feasibility confirmed:** A React Native (non-native Android) application can access the onboard Onboard API within the AAOS context on the SEM. This was uncertain at the start of the project.
2. **Latency eliminated:** Moving data access from Volvo Connect (cloud) to onboard (Onboard API via onboard Telematics unit) eliminated the data delay that made real-time driver coaching impossible.
3. **Lead time validated:** The integration was achievable within the 2-month Phase 1 target, confirming that with the right library and test environment, third-party developers can integrate quickly.
4. **Scalable model established:** The project demonstrated that an external partner can integrate with high self-sufficiency, supporting a scalable onboarding model for the Volvo Trucks ecosystem.
5. **Cloud development feasibility assessed:** Amazon Virtual Engineering Workbench was evaluated as a path to enable cloud-based development and testing for future partners.

6. Decision Analysis and Evidence

Each of the five decisions below represents a strategic choice made during the project.

Decision 1 — SDV Requires Faster Software Iteration and OTA Lifecycle

Decision statement: The project should prioritize software delivery speed and controlled updateability as central value drivers, not only integration completion.

Why this matters: SDV platforms compete on how fast they can deliver new software features and updates to vehicles in the field. Regulations now mandate controlled, traceable OTA update processes. This means development environments must support frequent, automated, and auditable software iterations.

What the evidence says:

- UNECE R156 mandates a Software Update Management System for vehicle type approval [D1-1].
- UNECE R155 requires continuous cybersecurity lifecycle management, including secure OTA channels [D1-2].
- ISO 24089:2023 specifies engineering requirements for road vehicle software updates [D1-3].
- AUTOSAR Adaptive Platform enables dynamic, updateable vehicle software architectures [D1-4].

- McKinsey estimates the automotive software market at $519B by 2035, driven by OTA-enabled features [D1-5].
- Google's Android Automotive platform explicitly describes itself as "evolving to support the SDV" with "a more centralized and updatable software foundation" [D1-6].
- NHTSA recommends secure OTA update processes and continuous vulnerability remediation [D1-7].
- The Uptane framework provides secure OTA update signing architecture specifically for automotive ECUs [D1-8].
- ISO 24089 Amendment 1 (2024) provides updated guidance on software update deployment [D1-9].
- Deloitte's 2024 study of 160+ executives found 81% of fleets expected to be software-defined by 2030, with OEMs investing up to $3B each in SDV R&D [D1-10].

Implication for this project: The Onboard API integration and the cloud-based test environment must support repeatable, automated integration loops so that new features can be validated and deployed quickly — consistent with an OTA-enabled lifecycle.

Decision 2 — Cloud High-Fidelity Simulation of the SEM Headunit Is a Core Enabler

Decision statement: Use cloud-based high-fidelity simulation of the SEM (Android Automotive headunit) and its onboard ECU interfaces to allow third-party development without physical truck access.

Why this matters: The main bottleneck for growing Volvo Trucks' third-party ecosystem is physical asset scarcity. Test trucks and bench rigs are expensive, located in specific countries, and have limited availability. Cloud simulation lets developers like Diffo iterate on their integration from anywhere, at any time.

What "high-fidelity" enabled in this project:

- Reliable validation of React Native app behavior against the Onboard API without a physical Telematics unit.
- Early exposure to edge cases (connectivity loss, stale signals, trigger timing).
- Repeatable, deterministic regression tests on every code change.
- Faster issue isolation compared to ad-hoc physical testing sessions.

What the evidence says:

- NVIDIA DRIVE Sim provides high-fidelity virtual environments that reduce physical fleet testing costs [D2-1].

- Synopsys Silver enables virtual ECU development "ahead of and independent of hardware test rigs" with deterministic SiL execution [D2-2].
- dSPACE offers cloud-distributed virtual ECU simulation, stating "SIL is all you need to validate production software in a purely virtual environment" [D2-3].
- ETAS virtual tooling supports ECU software integration before full hardware maturity [D2-4].
- AWS positions cloud simulation to "accelerate automotive time-to-market" with OEM supplier examples [D2-5].
- Siemens' PAVE360 portfolio centers on system-level virtual platform testing for SDV [D2-6].
- MathWorks reports OEMs like Rivian "reduced need for physical prototypes" by 66% using virtual simulation [D2-7].
- Google Cuttlefish provides a configurable virtual Android device running in the cloud, applicable to Android Automotive targets [D2-8].
- The Android Automotive emulator supports GUI testing with OEM emulator images, though vehicle-dependent functionality requires higher-fidelity simulation [D2-9].
- The Eclipse SDV initiative provides open-source building blocks for virtual ECU development [D2-10].

Implication for this project: The Amazon Virtual Engineering Workbench evaluation done in this project is a concrete step toward making cloud-based SEM simulation available to all partners via the Volvo Trucks Developer Portal.

Decision 3 — OEM APIs and Third-Party Access Increase Innovation Value

Decision statement: Use governed, standard-oriented API access patterns to lower integration friction and unlock scalable external innovation on the Volvo Trucks platform.

Why this matters: Innovation in fleet services depends on reducing the cost and time for external developers to build integrated solutions. Standardized APIs and clear onboarding processes mean less custom work and faster time-to-first-value for every new partner.

Volvo Trucks' approach: The Volvo Trucks Developer Portal (https://developer.volvotrucks.com) provides three integration paths [D3-7]:

1. **Cloud-to-Cloud APIs** following the rFMS (remote Fleet Management System) standard for fleet, vehicle, and driver data.
2. **Volvo Connect web SDK** for building service integrations within the Volvo Connect portal.
3. **In-vehicle apps on the SEM** using Android Automotive OS, with a Developer Guide, APK Screener, and testing resources.

This project focused on the third path — direct onboard integration via the Onboard API.

What the evidence says:

- The **COVESA Vehicle Signal Specification (VSS)** is an open data model with 61 contributors including BMW, Volvo Cars, JLR, and Bosch, creating a common language for vehicle signals across OEMs [D3-1].
- The **W3C VISSv2** specification defines a standardized vehicle data API that has been "implemented and deployed on production vehicles," with roles explicitly defined for third-party apps [D3-2].
- The **EU Data Act** (Regulation 2023/2854, applicable since September 2025) legally requires connected device manufacturers to enable user data sharing with third parties, fostering competition and innovation [D3-3].
- Android Automotive OS supports multiple third-party app categories (media, communication, navigation, IoT) and has a growing developer ecosystem [D3-4].
- McKinsey estimates **$250–400 billion** in annual value from connected-car data monetization by 2030, stating that ecosystem partnerships and open APIs are prerequisites [D3-5].
- **ISO 20078** defines the international standard for "Extended Vehicle" web services providing a normative framework for third-party access to vehicle data via REST APIs [D3-6].
- The **Volvo Trucks Developer Portal** invites partners to "leverage connected vehicle data to create powerful solutions for the transport industry" [D3-7].
- The **rFMS standard** is a multi-OEM initiative by Volvo, Scania, Daimler, MAN, DAF, and Iveco providing a standardized REST API for fleet management data [D3-8].
- **Mercedes-Benz Developer Portal** demonstrates real third-party innovation spanning smart EV charging, road safety, insurance, and smart city applications [D3-9].
- Deloitte's SDV study finds that cloud-native APIs, microservices, and DevOps approaches "enable modular design, faster updates, and adaptability" [D3-10].

Implication for this project: The Onboard API integration validates the onboard API access path. The experience directly informs what documentation, library, and test environment Volvo should make available on the Developer Portal for future partners.

Decision 4 — Onboard Data Paths Improve Latency and Resilience

Decision statement: For near-real-time use cases like Diffo's driver coaching and trip profitability analysis, design for onboard data access rather than cloud-only roundtrips.

Why this matters: Diffo's core product requires knowing in real-time when a trip starts, when the driver is at a fuel station, and what the instantaneous fuel consumption is. The data latency from Volvo Connect (cloud-to-cloud path) was too high for those use cases. Moving to onboard data access via the Onboard API eliminates the cloud roundtrip entirely.

What the evidence says:

- **ETSI Multi-access Edge Computing (MEC)** standards define ultra-low latency and high bandwidth as core edge benefits for V2X use cases [D4-1].
- **Azure IoT Edge** enables local analytics and offline decision-making at the edge, reducing dependence on cloud connectivity [D4-2].
- **AWS IoT Greengrass** provides local execution with offline resilience for environments with intermittent connectivity [D4-3].
- The **W3C VISSv2** specification includes handling for temporary connectivity loss and local data history capture [D4-4].
- **5GAA's MEC evaluation methodology** (2025) provides automotive-specific support for edge latency and reliability requirements [D4-5].
- **C-V2X frameworks** emphasize local/near-edge messaging where latency and availability are critical [D4-6].
- Android Automotive OS runs natively on in-vehicle hardware, enabling local app logic independent of cloud roundtrips [D4-7].
- The **Automotive Edge Computing Consortium (AECC)** has published 15+ white papers on automotive edge computing, arguing that vehicles generate massive data volumes requiring local processing [D4-8].
- Research by **Shi et al. (2016)** in IEEE Internet of Things Journal established edge computing as necessary for latency-sensitive IoT applications where cloud-only introduces unacceptable delays [D4-9].
- The project's SOW states: "Higher latency for the data available on Volvo Connect in comparison to onboard platform (not suitable for solutions that require more frequent and real-time data)" [D4-10].

Implication for this project: The Onboard API integration proves the viability of the onboard path. For Diffo, this means real-time trip detection and fuel coaching are now possible — features that were not achievable with the cloud-only data path.

Decision 5 — Cloud-Based SDV Development Improves Quality, Lowers Investment, and Increases Innovation

Decision statement: Adopt cloud-based virtual development and testing practices to improve software quality, reduce the investment needed for each integration project, and enable more innovation experiments across the ecosystem.

Why this matters: If every third-party integration requires physical truck access, the ecosystem can only grow as fast as Volvo can provide test trucks and rigs. Cloud simulation removes this bottleneck.

Lead-time effect

Fewer waits for physical truck slots and lab setups. The project's SOW set a 2-month target for Phase 1 integration. With cloud simulation, this target can be met or beaten because developers don't spend days waiting for truck access.

Quality effect

Cloud simulation enables broader scenario coverage:

- McKinsey reports that CI/CD practices deliver **+40% productivity and −60% defect rates** in automotive software development [D5-4].
- DORA's multi-year research program shows that speed and stability are not tradeoffs — teams that deploy more frequently also have lower failure rates [D5-1].
- Volvo Cars reported achieving **7× more tests per day** by moving to cloud-based SiL testing on AWS [D5-7].

Investment/ROI effect

- Deloitte's 2024 study found that reducing software versions and standardization can save OEMs up to **US$1 billion in R&D**, with centralized approaches yielding up to **23% R&D cost savings** [D5-8].
- McKinsey reports that digital twins can deliver **50% faster time-to-market** and **25% quality improvement**, with test acceleration factors of up to **10,000×** [D5-9].
- For third-party developers specifically, cloud simulation means no investment in their own test hardware — lowering the entry barrier for building on Volvo's platform.

Innovation throughput effect

If tested concepts cost less and take less time to validate, more experiments can run per period. This directly increases the number of innovations that reach the market.

What the evidence says:

- The **DORA research program** (Google Cloud) empirically links engineering capabilities to software delivery performance [D5-1].
- The **DORA ROI of DevOps study** directly addresses investment returns from software delivery transformation [D5-2].
- Earlier DORA research found elite performers **24× more likely to use cloud capabilities** effectively [D5-3].

- McKinsey's analysis found top-quartile automotive software teams achieve **3× productivity and 6× quality** vs bottom quartile; CI/CD yields **+40% productivity, −60% defects** [D5-4].
- McKinsey shows 30–50% of development effort wasted on integration; platforms reduce this overhead [D5-5].
- McKinsey describes 12% CAGR software growth with architecture reuse payoff [D5-6].
- **Volvo Cars + AWS** case demonstrates 7× increase in daily test throughput via virtualized SiL [D5-7].
- **Deloitte SDV Study** quantifies $1B savings potential and 23% R&D cost reductions [D5-8].
- McKinsey digital twin analysis reports 50% faster time-to-market and 25% quality improvement [D5-9].
- **EE Times/NXP** coverage shows shift-left testing with virtualization enables hardware-free testing [D5-10].

Implication for this project: The project validates that cloud-based development is the path to scaling Volvo Trucks' partner ecosystem. Making the Onboard API library and virtual SEM environment available through the Developer Portal would allow many partners to develop simultaneously.

---

7. How Cloud SEM Simulation Shortens Lead Time

7.1 Baseline physical-only bottlenecks

Without cloud simulation, a third-party developer like Diffo must:

- Travel to Sweden to access test trucks or rigs.
- Coordinate integration windows with Volvo platform engineers.
- Wait for environment setup and reset between test sessions.
- Struggle to reproduce intermittent issues in non-deterministic physical sessions.

These waits dominate total elapsed lead time, even when the actual coding effort is small. "It is difficult and time consuming to test on the real HW or rigs (test trucks). Physical access to the test environment is limited."

7.2 Simulation-first acceleration mechanisms

A cloud-hosted virtual SEM environment shortens lead time by:

1. **Removing travel and scheduling:** Test from anywhere, at any time.
2. **Enabling parallel work:** Multiple developers and partners work concurrently.
3. **Providing reproducibility:** Deterministic replay of failures reduces diagnosis time.

4. **Supporting early integration:** Test Onboard API contracts before waiting for truck access.
5. **Automating verification:** Every code commit triggers automated integration checks.

7.3 Lead-time model

Total lead time can be approximated as:

$T_{total} = T_{coding} + T_{wait} + T_{validation} + T_{rework}$

Cloud simulation primarily reduces T_{wait} (no physical queue) and T_{rework} (earlier defect discovery). The project's 2-month Phase 1 target demonstrates that with the right tools, integration can be fast.

8. How Quality Improves

8.1 Earlier defect discovery (shift-left)

When integration and failure-path tests run in a cloud environment before physical deployment, defects are found where the fix cost is lowest. McKinsey data suggests CI/CD practices reduce defect rates by up to 60% [D5-4].

8.2 Broader test coverage

Cloud simulation enables larger test matrices:

- Timing variants (different API update rates).
- Connectivity interruptions (simulated network drops).
- State transitions (ignition on/off).
- Fault injection (stale signals, missing data frames).

These are hard to execute repeatedly on a limited number of physical trucks.

8.3 Deterministic replay

When a bug occurs in a virtual environment, the exact sequence of events can be replayed. This removes guesswork and increases confidence in fixes.

8.4 Security and access validation

Onboard API access controls, token behavior, and scope restrictions can be tested systematically — including denied access, token expiry, and permission mismatches — improving robustness before production deployment.

9. How Investment Decreases and ROI Improves

9.1 Cost structure shift

Cloud simulation shifts part of the development cost from fixed physical infrastructure (test trucks, bench rigs, travel) to elastic cloud consumption. This does not eliminate all physical testing, but removes the most expensive bottleneck in the early and middle phases of integration.

9.2 Lower entry barriers for third parties

Diffo did not need to purchase or maintain test hardware. With a cloud-based SEM simulation available through the Developer Portal, future partners could start integration on day one — before any physical contact with a truck. This directly reduces the minimum investment needed to build a service on the Volvo Trucks platform.

9.3 ROI at the ecosystem level

- **Lower cost per validated concept** — cloud simulation is cheaper than truck bench time.
- **More concepts tested per period** — parallel development removes physical bottlenecks.
- **Faster time-to-revenue** — partners reach production integration sooner.
- **Lower maintenance cost** — better initial quality means fewer production issues.

Deloitte's research suggests standardization and virtualization can save up to $1B in R&D costs for OEMs, with centralized approaches yielding 23% R&D cost reductions [D5-8].

9.4 Innovation throughput

If the number of validated partner experiments per quarter rises while quality remains high, the Volvo Trucks ecosystem can commercialize more third-party services. This is the core business case for making cloud SEM simulation available through the Developer Portal.

10. Expected Long-Term Effects

- **OEM platform attractiveness:** Volvo Trucks' position as "the easiest platform to integrate with from a 3rd party perspective" is advanced by cloud simulation and documented APIs.
- **Ecosystem growth:** More developers can onboard and deliver integrations with lower initial investment and shorter lead times.
- **Fleet operator value:** Better operational insights via real-time in-vehicle services — Diffo's case demonstrates automatic trip analysis, fuel efficiency reporting, and work time management.

- **Driver safety:** Built-in applications reduce the need for aftermarket screens that the SOW identifies as "lower safety for driver and road users."
- **Sustainability and efficiency:** Real-time fuel efficiency coaching and transport mission profitability analysis can contribute to reduced fuel consumption and more efficient operations.

11. Risk and Governance Considerations

- **Simulation–reality gap risk:** Not all vehicle behaviors can be perfectly simulated. Mitigated by a staged validation strategy: cloud simulation first, followed by targeted physical confirmation on actual trucks for edge cases and safety-relevant scenarios.
- **API governance risk:** As the partner ecosystem grows, API versioning, deprecation, and access governance become critical. Mitigated by using the Developer Portal as a single source of truth for API documentation, with clear partner terms and access policies.
- **Security and compliance risk:** Onboard app access to vehicle data must comply with UNECE R155 cybersecurity requirements and data privacy regulations. Mitigated by the API's access-control model and Volvo's existing TARA (Threat Analysis and Risk Assessment) and Data Privacy process.
- **Vendor lock-in risk:** Mitigated by building on standards (AAOS, rFMS, COVESA VSS) and using portable development toolchains (React Native, standard Android SDK).

12. Recommendations for Scale-Up

1. **Publish the API library and virtual SEM development environment on the Developer Portal** so that any approved partner can start integration independently.
2. **Standardize the partner onboarding blueprint:** API access scopes, authentication, test data packs, and self-service documentation.
3. **Invest in cloud SEM simulation fidelity:** Go beyond GUI-level Android Emulator testing to include API behavior simulation, and network condition injection.
4. **Define a hybrid validation policy:** Cloud simulation first for all integration testing, with targeted physical truck confirmation for specific scenarios.
5. **Expand ecosystem enablement** through reference apps, developer workshops, and sandbox environments.

13. Alignment with Vinnova Public Reporting Requirements

This report follows Vinnova's external publication requirements: project summary, implementation description, outcomes, expected effects, and utilization perspective. Confidential implementation details are intentionally omitted.

The project was conducted within Volvo CampX, with Vinnova co-funding under FFI (Fordonsstrategisk Forskning och Innovation). The results are intended to inform both Volvo's internal platform strategy and the broader community of SDV ecosystem practitioners.

14. References

SDV Requires Faster Software Iteration and OTA Lifecycle

- [D1-1] UNECE Regulation No. 156 — Software Update Management System (SUMS). UNECE WP.29, 2021. https://unece.org/sites/default/files/2021-03/R156e.pdf
- [D1-2] UNECE Regulation No. 155 — Cybersecurity Management System (CSMS). UNECE WP.29, 2021. https://unece.org/sites/default/files/2021-03/R155e.pdf
- [D1-3] ISO 24089:2023 — Road vehicles — Software update engineering. ISO, Feb 2023. https://www.iso.org/standard/77796.html
- [D1-4] AUTOSAR Adaptive Platform. AUTOSAR, current release R25-11. https://www.autosar.org/standards/adaptive-platform
- [D1-5] "The automotive software and electronics market through 2035." McKinsey, Jan 2026. https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/mapping-the-automotive-software-and-electronics-landscape
- [D1-6] "What is Android Automotive?" Android Open Source Project (AOSP), Google. https://source.android.com/docs/automotive/start/what_automotive
- [D1-7] "Cybersecurity Best Practices for the Safety of Modern Vehicles." NHTSA, 2022. https://www.nhtsa.gov/technology-innovation/vehicle-cybersecurity
- [D1-8] Uptane Framework — Secure Software Update for Automobiles. Linux Foundation. https://uptane.org/
- [D1-9] ISO 24089:2023/Amd 1:2024 — Amendment to software update engineering. ISO, 2024. https://www.iso.org/standard/87522.html
- [D1-10] "Software Defined Vehicle: Global Manufacturer Readiness Study." Deloitte, Oct 2024. https://www.deloitte.com/global/en/Industries/automotive/analysis/software-defined-vehicles.html

Cloud High-Fidelity Simulation / Virtual ECU

- [D2-1] NVIDIA DRIVE Sim — Autonomous Vehicle Simulation. NVIDIA. https://developer.nvidia.com/drive/drive-sim
- [D2-2] Synopsys Silver — Software-in-the-Loop for Virtual ECUs. Synopsys. https://www.synopsys.com/verification/virtual-prototyping/silver.html
- [D2-3] dSPACE VEOS — Virtual ECU and SiL Testing. dSPACE. https://www.dspace.com/en/pub/home/products/sw/simulation_software/veos.cfm
- [D2-4] ETAS — Virtual ECU and Software Tooling. ETAS (Bosch Group). https://www.etas.com/
- [D2-5] "Product Engineering & Design for Automotive." AWS. https://aws.amazon.com/automotive/product-engineering/

- [D2-6] Siemens EDA — PAVE360 Automotive Verification Portfolio. Siemens. https://eda.sw.siemens.com/
- [D2-7] "Building Virtual Vehicles with Simulink." MathWorks. https://www.mathworks.com/solutions/automotive/virtual-vehicle.html
- [D2-8] Google Cuttlefish — Configurable Virtual Android Device. AOSP. https://source.android.com/docs/devices/cuttlefish
- [D2-9] Android Automotive Emulator — Android Developers. Google. https://developer.android.com/studio/run/emulator
- [D2-10] Eclipse SDV — Software Defined Vehicle Working Group. Eclipse Foundation. https://www.eclipse.org/sdv/

OEM APIs and Third-Party Access

- [D3-1] COVESA — Connected Vehicle Systems Alliance and Vehicle Signal Specification (VSS). https://covesa.global/
- [D3-2] W3C VISSv2 Core — Vehicle Information Service Specification version 2. W3C, Feb 2024. https://www.w3.org/TR/viss2-core/
- [D3-3] EU Data Act — Regulation (EU) 2023/2854. European Commission, applicable Sept 2025. https://digital-strategy.ec.europa.eu/en/policies/data-act
- [D3-4] "Android for Cars — Overview." Google Android Developers. https://developer.android.com/cars
- [D3-5] "Unlocking the full life-cycle value from connected-car data." McKinsey, Feb 2021. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/unlocking-the-full-life-cycle-value-from-connected-car-data
- [D3-6] ISO 20078-2:2021 — Road vehicles — Extended vehicle (ExVe) web services. ISO. https://www.iso.org/standard/80184.html
- [D3-7] Volvo Trucks Developer Portal. Volvo Group. https://developer.volvotrucks.com
- [D3-8] rFMS Standard — Remote Fleet Management System. Multi-OEM initiative (Volvo, Scania, Daimler, MAN, DAF, Iveco). https://rfms-standard.com/
- [D3-9] Mercedes-Benz Developer Portal. Mercedes-Benz AG. https://developer.mercedes-benz.com/
- [D3-10] "Software Defined Vehicle: Global Manufacturer Readiness Study." Deloitte, Oct 2024. https://www.deloitte.com/global/en/Industries/automotive/analysis/software-defined-vehicles.html

Onboard/Edge Data Paths

- [D4-1] ETSI Multi-access Edge Computing (MEC). ETSI. https://www.etsi.org/technologies/multi-access-edge-computing
- [D4-2] "What is Azure IoT Edge." Microsoft, 2026. https://learn.microsoft.com/en-us/azure/iot-edge/about-iot-edge
- [D4-3] "What is AWS IoT Greengrass?" AWS. https://docs.aws.amazon.com/greengrass/v2/developerguide/what-is-iot-greengrass.html

- [D4-4] W3C VISSv2 Core — includes in-vehicle/cloud hybrid handling. W3C, Feb 2024. https://www.w3.org/TR/viss2-core/
- [D4-5] "Mobile Edge Computing Use Cases and Evaluation Methodology Review." 5GAA, Jul 2025. https://5gaa.org/mobile-edge-computing-use-cases-initial-test-specifications-and-evaluation-methodology-review/
- [D4-6] "C-V2X Explained." 5GAA. https://5gaa.org/c-v2x-explained/
- [D4-7] "What is Android Automotive?" AOSP/Google. https://source.android.com/docs/automotive/start/what_automotive
- [D4-8] Automotive Edge Computing Consortium (AECC) — White Papers on Edge Computing for Vehicles. https://aecc.org/
- [D4-9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016. doi: 10.1109/JIOT.2016.2579198
- [D4-10] "AWS Connected Mobility and EVs." AWS. https://aws.amazon.com/automotive/connected-mobility/

Quality, ROI, and Innovation Throughput

- [D5-1] DORA Research Program — Software Delivery Performance. Google Cloud. https://dora.dev/research/
- [D5-2] "The ROI of DevOps Transformation." DORA/Google Cloud, 2020. https://dora.dev/research/2020/
- [D5-3] "Accelerate: State of DevOps" report series. DORA, 2019. https://dora.dev/research/
- [D5-4] "The case for an end-to-end automotive-software platform." McKinsey, Jan 2020. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform
- [D5-5] "The case for an end-to-end automotive-software platform." McKinsey, Jan 2020. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform
- [D5-6] "The automotive software and electronics market through 2035." McKinsey, Jan 2026. https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/mapping-the-automotive-software-and-electronics-landscape
- [D5-7] "Volvo Cars streamlines in-vehicle software testing with AWS Graviton on Amazon EKS." AWS Industries Blog. https://aws.amazon.com/blogs/industries/volvo-cars-streamlines-in-vehicle-software-testing-with-aws-graviton-on-amazon-eks/
- [D5-8] "Software Defined Vehicle: Global Manufacturer Readiness Study." Deloitte, Oct 2024. https://www.deloitte.com/global/en/Industries/automotive/analysis/software-defined-vehicles.html
- [D5-9] "Digital Twins: What could they do for your business?" McKinsey. https://www.mckinsey.com/capabilities/operations/our-insights/digital-twins

- [D5-10] "Shift-Left Testing with Virtualization and Digital Twins." EE Times / NXP coverage. https://www.eetimes.com/

Additional project references

- [P-1] Volvo Trucks Developer Portal — In-Vehicle Apps Guide. https://developer.volvotrucks.com/default/in-vehicle-apps.html
- [P-2] Vinnova FFI Program — Fordonsstrategisk Forskning och Innovation. https://www.vinnova.se/e/fordonsstrategisk-forskning-och-innovation/