

クラウドから車載エッジまで一貫した開発環境が ソフトウェア定義型自動車を加速

Girish Shirasat

Arm オートモーティブ部門ソフトウェア戦略 & アーキテクチャ担当ディレクター

Stefano Marzani

AWS 自動運転車部門プリンシパルスベジスタソリューションズアーキテクト



ホワイトペーパー

ビジョン

2025 年、大手自動車部品メーカーで働くソフトウェアエンジニアの Judith は、在宅勤務で ACC（アダプティブ・クルーズ・コントロール）機能に関する顧客のフィードバックに対応しています。彼女は問題のある挙動に関連する大量のデータにクラウドワークスペースでアクセスし、それを利用して制御アルゴリズムを修正します。次に、車両ダイナミクスシミュレーターを備えたソフトウェア・イン・ザ・ループ・テストベンチで修正後の性能を評価/検証し、パッケージに「準備完了」のマークを入れます。同じ頃、何千キロも離れた別の大陸でデザイン会社に勤務する Mark は、来週、車が発売される地域向けにローカライズ中の UI ソフトウェアに最後の仕上げを加えています。彼は、OEM が提供したクラウドパネルで仮想エージェントを使用したマッシュプリー・パラレル・テストスイートを実行し、修正を加えます。これが終われば UI パッケージに「準備完了」とマーク。一方、DevOps バリデーションエンジニアの Kay は OEM の本社で働いています。2 つのソフトウェアパッケージを確認し、最終的なハードウェア・イン・ザ・ループ・テストスイートでコンテンツを完全に検証/承認した後、翌週以降に販売される本番車両すべてに導入します。

はじめに

自動車業界は「ソフトウェア定義」の未来に向かっていきます。上記のビジョンのように、多くの OEM が機敏かつ柔軟にソフトウェアを開発し、自動車の機能性を継続的に高めることで、妥協のない品質と安全性を提供します。

現代的なデジタルサービスや使いやすい車載アプリケーションを含むソフトウェア中心のエコシステムを作るには、機能安全のコンセプトやリアルタイム実行の点で車載技術特有の性質を維持できるクラウドネイティブのアプローチ^[1]が不可欠です。このアプローチなら、革新的で効率的なワークフローを駆使し、多くの開発者が開発プロセスに関与できます。自動車メーカーも開発期間を短縮し、現代の消費者の期待に合わせて機能を柔軟かつスピーディーに改善/更新できます。

自動車向けクラウドネイティブ開発パイプラインの技術的な実現に向けてまず必要なのは、クラウド環境と、最終的にワークロードを運用する組み込み車載エッジプラットフォームの**一貫した開発環境**です。最先端の情報アーキテクト、Kevin Hoffman は次のように述べています^[2]。

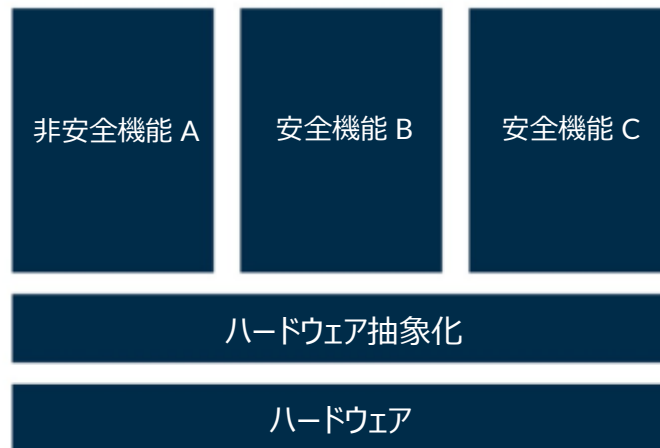
*「一貫した開発環境に極めて厳しく取り組む目的は、アプリケーションが**どこでも機能する**という安心感をチームと組織全体に与えることです」*

この一貫性を達成すれば、開発者のデスク上にある組み込みハードウェアに関係なく、開発と検証をクラウドで実行できます。車載バリューチェーン全体でソリューションの製品化期間が大幅に短縮されるとともに、アプリケーションが移植可能となり、開発ワークフローの将来も保証されます。

一貫した開発環境を実現し、車載ソフトウェアに対する需要の増大^[3]に対応するには、十分にデモンストラクション化されたマルチコアの組み込みシステムが自動車内に必要です。本書では、車載システム開発にクラウドネイティブアプローチを適用する方法に注目します。特に、クラウド内と車載エッジの実行環境の一貫性に重点を置き、ソフトウェア定義型自動車の製品化期間を短縮する可能性を検討します。

ソフトウェア定義型自動車（SDV）とは？

図 1 – 抽象化ハードウェアの上にある車載機能



ソフトウェア定義型自動車の「ソフトウェア定義」とは、車の機能をソフトウェア機能やサービスと見なし、実装することです。そして個々の物理的な ECU 制御ユニットなどに実装するのではなく、共有型または集中型のコンピューターで実行します。さらに「ソフトウェア定義」とは、製造前、製造後を含め、車載システム開発のライフサイクル全体において、ソフトウェア定義の機能を柔軟に開発し、導入することも意味します。図 1 のように、理想的に言えばソフトウェアサービスは、ハードウェアやベンダーを問わず、特定の機能を果たすためにデータを提供する（センサー）、データを処理する（アプリケーションロジック）、データを消費する（アクチュエータ）サービスと解釈されます。このレベルの柔軟性があれば、OEM だけでなく、専門性の高いソフトウェア会社も特定の用途向けの新しい機能を開発可能です。製品の販売後に新しい機能を追加することは、スマートフォンなどの業界で 10 年以上前から一般に行われていますが、同じアプローチが今では車載システムなど従来の組み込み市場でも強く求められています。

SDV とは、以下の特徴を持つシステムと説明できます。

- 車がネットワーク接続機能を持ち、「ビッグデータ」の観点で継続的にデータを送信するとともにソフトウェア更新を無線で受信できる^[5]。
- ハードウェアの上のレベルでソフトウェアが抽象化されている。
- 車の機能や能力がソフトウェアで実現され、車のライフサイクルを通じて更新や管理が可能。
- DevOps（開発運用）の観点で、クラウドから車載エッジまでハードウェアプラットフォーム全体でクラウドネイティブの設計概念を採用する。
- 車載アプリケーションのミックスドクリティカル管理により、障害時の安全をさまざまなレベルで保証するワークロードを実現する^[6]。

SDV における「クラウドネイティブ」

「ソフトウェア定義」は新しい概念ではありません。上記のスマートフォンの例に加え、電気通信やデータセンターには、ソフトウェア定義ネットワーク、ソフトウェア定義ストレージ、ソフトウェア定義のコンピューター処理デザインがすでに存在し、企業アプリケーションの分野にも普及しています。このような分野ではクラウドネイティブの概念を有効に利用し、多数の設計パターンやエコシステムツールを通じてソフトウェア定義システムを運用しています^[12]。したがって、クラウドネイティブの概念を車載システムにも応用し、車載関連の膨大な技術的、商業的エコシステムを革新の有効性やスピード向上に役立てることを検討するのは極めて論理的です。

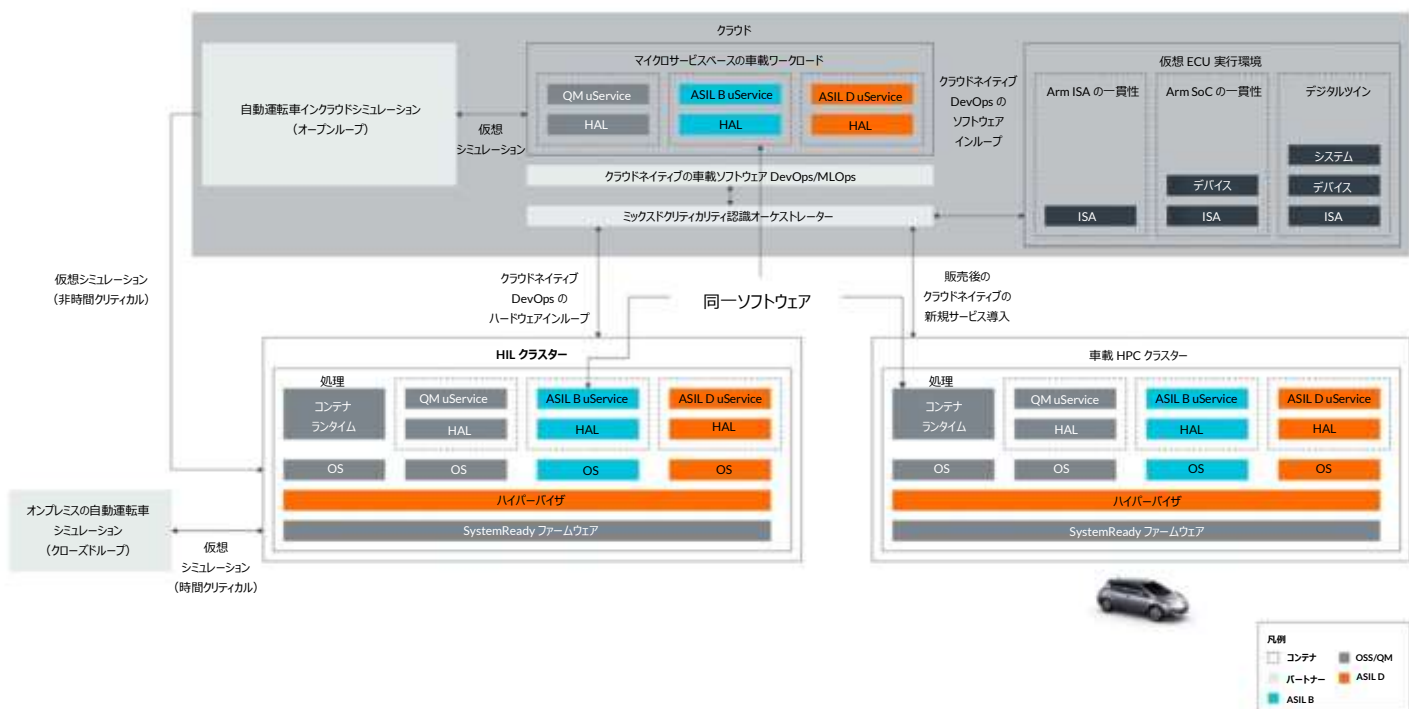


図 2 – 車載システム開発におけるクラウドネイティブの応用

図 2 を見れば、開発者がノート PC を持ってカフェに座り、クラウドと車載エッジの**一貫した開発環境**を前提としてクラウドに接続し、ACC（アダプティブ・クルーズ・コントロール）やラテラル・ライン・コントロールなどのインフォテイメントや ADAS（先進運転支援システム）アプリケーションの開発、コードのコミット、ビルドと統合サイクルのトリガーを、物理的な車載ターゲットシステムを綿密に表現したクラウド内の仮想実行環境で実行する姿を容易に想像できます。クラウドと車載エッジの一貫した開発環境のレベルとシミュレーションの速度は、開発者の作業効率に直接的な影響を与えます。

車載システムのソフトウェアの複雑性が Boeing 787 Dreamliner に搭載されているソフトウェアを上回る現在^[7]、車載システムの開発者は、サービス志向のアーキテクチャ/マイクロサービスをベースとした実証済みのソフトウェア設計パターンを選択することが予想されます。ここではコンテナがワークロードの移動性と複雑性を達成する基本的な技術となります^[11,12]。車載アプリケーションの開発には、コンテナが一般的に利用されている企業用やスマートフォン用のアプリケーション開発では検討されてこなかった特有の課題があります。つまり車載ワークロードは、安全性とリアルタイムの要件によってはミックスドクリティカルの性質を持ちます。ワークロードを形成するマイクロサービスが、対応の安全性分解に加え、特定の空間的、時間的要件に制約を受けるためです。このようなマイクロサービスには用途に応じて、品質管理（QM）されている、ISO 26262 仕様の定める ASIL-B/Avcv-D のインテグリティレベルに達しているなどの条件が課される場合があります。このような場合は、安全認証を取得したコンパイラやツールをクラウドベースのツールフレームワークに組み込む必要があります。また、車載アプリケーションを運用するハードウェアは従来から非常に分散し（現代の自動車は 100 個以上の電子制御ユニット [ECU] を搭載）、性質も多岐にわたるため、ソフトウェアとハードウェアの分割が困難です。最後に、既存のクラウドネイティブインフラは、エンタープライズクラスのプラットフォーム認識機能によって最適なハードウェアノードにマイクロサービスを導入するオーケストレーターを備えていますが、それを車載分野に特有の組み込みハードウェアシステムの能力を理解し、最適な形でマイクロサービスを導入できるよう拡張する必要があります。

これらのすべての要件や問題を解決して初めて、開発者はクラウドでシミュレーションを安全に実行し、拡張性や柔軟性などの根本的なメリットを活用できます。たとえば開発者は DevOps インフラの一環として完全なソフトウェア・イン・ザ・ループ (SiL) 検証を実行できることを期待します。つまり、クラウド上のシミュレーターを使用してシミュレートした各種の運用設計ドメイン (ODD) を実行するとともに、膨大なシミュレーションデータをテスト対象のソフトウェアに入力して結果を速やかに検証します。このような膨大なリアル/シミュレーション環境では何千、何万ものシナリオを同時に検証し、何千ものコア上で並列実行する場合があります。ハードウェア・イン・ザ・ループ (HiL) リグの組み込みシステムだけで、このような規模を得ることはできません。

Arm は、お客様との事前調査に基づき、クラウドからエッジまで一貫した開発環境があれば、現在 HiL リグで実行しているテストの約 70%をクラウドベースの SiL 環境に移行し、クラウドの拡張性を活用できると推定しています。

一部の機能（入出力のテストなど）は、組み込みシステムの物理的な性質に関連するため常にハードウェアでの検証が必要です。しかしクラウドから車載エッジへの移行に関してシステム開発の観点から言えば、クラウドで検証されたソフトウェアを、クラウドネイティブのオーケストレーションを通じて物理的な ECU に導入し、ラボ内で HiL 検証を実行した後、販売済みの車が実際に道路を走っている状態で導入することが期待されます。

クラウドから車載エッジまで一貫した開発環境によるクラウドネイティブ実現

クラウドネイティブの車載システム開発における重要なインフラ要件の 1 つは、開発、統合、検証を目的として、**物理的環境と非常に類似した**、そして可能な限り車載エッジとの一貫した開発を実現するさまざまな仮想システム実行環境です。高いレベルの一貫性は開発者のフィードバックループに直接影響を与え、開発の有効性を高めます^[8]。次に一貫した開発環境について、各種の一貫性やペルソナの見直しによるクラウドでの実現方法について、さまざまな開発者の観点から掘り下げてみましょう。

図 3 – クラウドから車載エッジまで一貫した開発環境



アプリケーション開発者

車載システム開発者がクラウドベースの実行環境に最初に期待するのは、クラウド環境と車載エッジ環境における命令セットアーキテクチャ (ISA) と CPU アーキテクチャの一貫性です。たとえば Arm ベースの車載コンピューティングプラットフォームでは、[AWS Graviton](#) が提供する新しい Arm ベースのクラウドインスタンスでクラウド環境をホスト可能です。クラウドから車載エッジへ、エンドツーエンドで Arm ベースのコンピューティングがつながるのは画期的なことです。これが理想的な開発環境と考えられるのには複数の理由があります。

- 開発者の効率**：開発有効性の専門家である Tim Cochran が述べているように^[9]、開発者のフィードバックループの削減は、企業収益を左右する開発効率の向上にとって極めて重要です。Arm 上で Arm 用のコードを開発する主なメリットは以下のとおりです。
 - クロスコンパイルの回避。あるアーキテクチャの開発システム上で、異なるアーキテクチャのターゲット上で実行するコードをコンパイルすることは、根本的に開発/導入/テストの手順を増やすことであり、開発者のフィードバックループを延長します。
 - クロスコンパイルは、余計なエラーをソフトウェアに持ち込む可能性があります。
 - クラウド内で Arm CPU モデルをホストすることは、ISA の一貫性が不十分で CPU アーキテクチャの一貫性が必要な場合に有効です。異なるアーキテクチャ上でのエミュレーションは速度と効率性に劣る場合がありますが、Arm-on-Arm 環境では Arm ハイパーバイザ拡張を直接使用し、モデルの実行速度を高めることができます。

-
- **パフォーマンスの最適化**：コンパイラは必要なすべての最適化に対応すると一般に思われていますが、複数のパートナーやサードパーティのエコシステムプロバイダーの例を見ても、ライブラリや機能ブロックを本当に最適化するには、まだ開発者が手作業で特定アーキテクチャ向けの最適化を実行し、重要なセグメントの処理を限界まで速めなければならない場合があります。クラウドとエッジの一貫した開発環境があれば、このような最適化を開発から導入へと「シームレス」に移行できます。

プラットフォーム開発者

アプリケーション開発に加え、現代の自動車に存在するもう 1 つのソフトウェアの層が、ファームウェア、OS/RTOS、デバイスドライバ、関連コンポーネントを含むベースプラットフォームシステムソフトウェアです。プラットフォーム開発者には、CPU コアおよび特定の機能を満たすための特長を備えた SoC（システムオンチップ）をシミュレートする仮想環境、さらには適切なデバイスインタフェースと対応の SoC システムアーキテクチャが必要です。これは、仮想 SoC モデルをクラウドでホストし、車載ソフトウェアの CI/CD 全体に統合することで可能となります。ここでの最大の問題は、プラットフォームに応じた修正をほとんどせずに市販の OS イメージをブートするための基本的なシステムアーキテクチャの標準化です。Arm SystemReady 認証プログラムが幅広い業界サポートを得て解決しようとしているのは、まさにこの課題です。

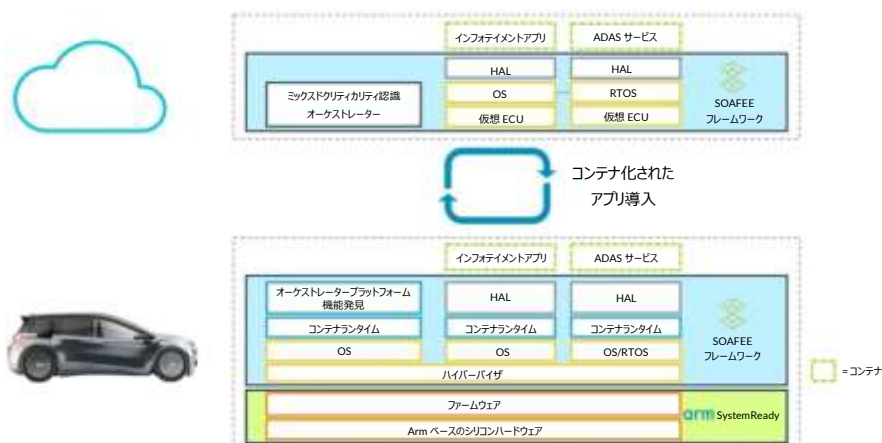
システム開発者

本書ですでに述べたように、車載システムで実行されるワークロードは本質的にミックスドクリティカルで分散型です。このようなワークロードを開発および検証するには、システムの機能的、非機能的な面（リアルタイム特性、安全性、パフォーマンスなど）をシミュレートする車載のデジタルツインをクラウド内でホストする必要があります。またこのデジタルツインは、物理的な環境に見られるさまざまな運用設計ドメインもクラウドでの実行時にモデリングする必要があります。車載デジタルツイン技術をすでに開発中の大手エコシステムパートナーは、一貫した開発環境の溝を埋める点で大きな進歩を遂げています。

SOAFEE – 車載分野のクラウドネイティブ化を推進する業界主導の取り組み

カフェで働く ADAS 開発者のビジョンを実現するには、バリューチェーン全体にわたるパートナーエコシステムが協力する必要があります。Arm は、AWS などの主要パートナーとともに、SOAFEE (Scalable Open Architecture for Embedded Edge) プロジェクトを発表しました。SOAFEE は、車載アプリケーションのミックスドクリティカリティに対応するクラウドネイティブのアーキテクチャであり、商用、非商用製品に利用できるオープンソースのリファレンス実装も提供します。SOAFEE を牽引する業界主導の SIG (special interest group) は、オープン標準に基づいて SOAFEE アーキテクチャを定義し、オープンソース実装を開発しています。取り組みの詳細は、[SOAFEE のウェブサイト](#)をご覧ください。

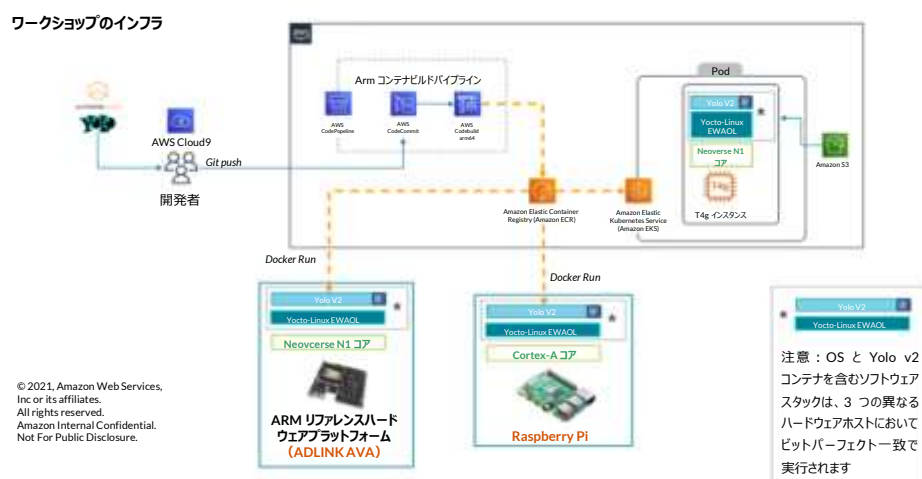
図 5 – SOAFEE の
ハイレベルリファレンス
アーキテクチャ



Arm DevSummit ワークショップ – 初の SOAFEE ベース実装を披露

Arm DevSummit 2021 でのライブワークショップ^[9]では、SOAFEE の初のリファレンス実装が発表されました。

図 6 クラウドからエッジまで一貫した開発環境を利用した車載システム開発



アーキテクチャに記載のとおり、本ワークショップでは新しい車載ネイティブのソフトウェア開発インフラを導入しました。このインフラでは、コンテナ化された同じワークロードを（一貫した開発環境により）ターゲットとする複数の処理環境（[AWS EC2 Graviton2](#) のインスタンス、[Raspberry Pi](#)、[AVA 開発プラットフォーム](#)）で実行できます。

このアーキテクチャでは AWS サービスを使用して CI/CD のパイプラインを作り、認識ネットワーク「YOLO」をビルド、コンテナ化、評価、導入します（クラウドと組み込みデバイスで大規模に）。そしてこのネットワークを車載アプリケーションワークロードの代役として使用し、設計パラダイムを実証します。ワークショップで使用した YOLO は「YOLOv2-Tiny」バージョンで、Ubuntu Linux 20.04 上で実行しました。

完全なテスト対象システム（SUT）スタックに、ネイティブプロパティでクラウド内で動作する組み込みオペレーティングシステム（Yocto-Linux ディストリビューション）を含めるのは、これが初めてでした。これも SOAFEE アーキテクチャのリファレンス実装である Arm の [Edge Workload Abstraction and Orchestration Layer \(EWAOL\)](#) のおかげです。

図 7 テスト対象システム：OS からのフルスタック



OS より上のすべての SUT コンポーネントは、すべてのターゲットに共通の基本アーキテクチャである Aarch64 を活用し、[命令セットの一貫性](#)によって実行されました。Arm の EWAOL は、複数の組み込みプラットフォーム上でのアプリケーション運用とオーケストレーションに対応する標準ベースのフレームワークをユーザーに提供します。この機能の組み合わせにより以下が可能となります。

- 開発中のソフトウェアの単位（この場合は YOLOv2-Tiny 認識モジュール自体）だけでなく、組み込み OS より上の[組み込みソフトウェアスタック全体のクラウド実行](#)。
- SUT のクラウドから組み込みエッジへの[シームレスなポータビリティ](#)。クロスコンパイルやエミュレーションは不要（コンパイルエラーやパフォーマンス低下などの問題もなし）。

実はこのアプローチにより、クラウド内での[組み込みコード](#)の記述およびテストの開始、組み込み開発ワークフローの[シフトレフト](#)、クラウドの拡張性を利用したテストカバレッジの大幅な拡大が可能となります（この場合は AWS Batch を使用して複数の SUT 実行を並列起動）。

結論と将来の活動

結論として、車載エコシステム内のさまざまな例に実証されるように、ソフトウェア定義型自動車は近未来のものではなく、明らかにすでに進行中のトレンドです。未来にはソフトウェアが競合する OEM の主要差別化要因となり、車載システム開発の未来においてはソフトウェア開発者が最も重要な役割を担うようになるでしょう。業界がこの転換を遂げるには、既存の車載向けクラウドネイティブ技術を利用して解決できる、そして解決すべき課題がいくつかあります。これを踏まえ、SOAFEE などの取り組みが、車載分野におけるクラウドネイティブ技術の導入と促進に重要な役割を果たしています。

クラウドネイティブの車載ソフトウェア開発を実現する最初の主要技術は、クラウドベースの開発環境と車載エッジの一貫性です。これにより車載ソフトウェア開発者はソフトウェアをクラウドで開発および検証し、エッジで運用を開始できます。そして車載システム開発を効率化するシフトレフト、スケールアウト、販売後導入のモデルが実現し、革新的なビジネスモデルが自動車業界を変えることになるでしょう。Arm ベースの車載ターゲットシステムと AWS Graviton のインスタンスは、現在提供されている一貫した開発環境の例であり、Arm DevSummit 2021 ^[9]ではエンドツーエンドのデモが披露されました。

最後に、新しい技術やコンセプトを開発するには、コンセプトを実証するコードを書き、革新やエコシステムの出発点となるプラットフォームを開発者に提供することが不可欠です。DevSummit ワークショップでの AWS とのデモンストレーションに加え、Arm は Autoware Foundation (AWF) と協力し、Open AD Kit ^[13]の開発を発表しました。これは、リファレンス自動運転ソフトウェアスタックとしてオートウェアを AWS Graviton インスタンス上でネイティブに開発し、クラウドネイティブのオーケストレーターを使用して車載エッジプラットフォームにシームレスに導入するための完全なエンドツーエンドの開発キットです。

これは自動車業界にとって大きな変革の始まりです。Arm、AWS、SOAFEE SIG のメンバーは今後も協力し、車載開発のクラウドネイティブ化を推進する予定です。このビジョンに賛同される方は、ぜひ SOAFEE SIG にご参加ください。詳細は soafee.io/をご覧ください。

参考文献

- [1] Girish Shirasat, [The Cloud-native Approach to the Software Defined Car \(ソフトウェア定義型自動車に向けたクラウドネイティブアプローチ\)](#)、2021年9月
- [2] Kevin Hoffman, [Beyond the Twelve-Factor App \(Twelve-Factor アプリを超えて\)](#)、2016年4月、O'Reilly Media, Inc.
- [3] Robert N. Charette, [How Software Is Eating the Car \(ソフトウェアが車を食い尽くす\)](#)、2021年6月
- [4] Robert Day, [The Software-Defined Vehicle Needs Hardware That Goes the Distance \(ソフトウェア定義型自動車には最後まで行けるハードウェアが必要\)](#)、2021年6月
- [5] Constantin Gillies, [The Big Loop: Artificial Intelligence and Machine Learning \(ビッグループ：人工知能と機械学習\)](#)、2021年7月
- [6] Alan Burns and Robert I. Davis, [Mixed Criticality Systems – A Review \(ミックスドクリティシティシステム – 総括\)](#)、2019年3月
- [7] [Information is Beautiful – Million Lines of Code \(情報は美しい – 百万行のコード\)](#)
- [8] 開発の有効性に関する Martin Fowler の[記事](#)
- [9] “Getting Started with Cloud-Native Automotive Software Development (クラウドネイティブ車載ソフトウェア開発の基礎)”ワークショップ、録画は[こちら](#)、手順を追ったチュートリアルは[こちら](#)
- [10] [Netflix Move to Cloud Native Architecture \(Netflix のクラウドネイティブアーキテクチャへの移行\)](#)
- [11] Computer Weekly – [How Containerisation Helps VW Develop Car Software \(コンテナ化がVWのカーソフトウェアに貢献\)](#)
- [12] [DevOps at Jaguar Land Rover \(Jaguar Land Rover における DevOps\)](#)
- [13] The Autoware Foundation – [The Autoware Foundation Releases Quick-starter Kit to Accelerate Cloud-native Autonomous Development \(Autoware Foundation がクラウドネイティブの自動運転開発を促進するクイックスターキットを発表\)](#)
- [14] [Arm Scalable Open Architecture for Embedded Edge Soafee \(SOAFEE アーキテクチャによるクラウドネイティブな開発の促進：ミックスドクリティカルな車載システムへ\)](#)



All brand names or product names are the property of their respective holders. Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given in good faith. All warranties implied or expressed, including but not limited to implied warranties of satisfactory quality or fitness for purpose are excluded. This document is intended only to provide information to the reader about the product. To the extent permitted by local laws Arm shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information.

© Arm Ltd. 2022