OTA の更新には、柔軟なアーキテクチャーが必要

車両は、ドライバーがガレージに戻ってから次に出発するまで、一晩中そこで静かに停車します。しかし、この車両はただ停車しているのではなく、機能を拡張し、安全性とインテリジェンスを高め、同日に組立ラインから出荷される新しい車両と同等機能を発揮する為、機能更新を行います。

これを可能にするのが、Over-the-Air (OTA) ソフトウェア更新です。OTA を車両アーキテクチャと統合することで、メーカーは、車両が工場出荷から長い時間経過した後も、アクティブ セーフティ、インフォテインメント等様々な領域を、低リスクで費用効果が高い、洗礼されたスケーラブルな方法で進化させることができます。

一元化されたコンピューター処理と、OTA 対応に最適化された設計がアーキテクチャーに組み込まれていれば、アプリケーション、OEM 戦略、ユーザーの好みに応じた必要な頻度で、ソフトウェアとファームウェアのアップデートをスムーズかつ確実に行うことができます。OTA は ソフトウェア・ディファインド・ビークルの構築に不可欠なだけでなく、革新的な新サービス、機能提供する新しい方法、新ビジネスモデルの可能性を切り開き、かつてない柔軟性と拡張性を実現する技術です。

ソフトウェア・ディファインド・ビークルの実現

主要な OEM 各社が、後から機能の追加・拡張を行うことを想定したソフトウェア・ディファインド・ビークルを製品化しています。機能の追加・拡張は、携帯電話や Wi-Fi のネットワークを使用したソフトウェアおよびファームウェアの OTA アップデートによって行われます。複雑で安全性が重視されるシステム内にますます多くの機能が搭載されるようになったため、アップデートは安全性、信頼性、確実性の高い方法で行うことが不可欠になっています。

OTA はどこでも見かける機能になってきたとはいえ、簡単に後付けできるものではありません。設計段階から OTA を念頭に置いた車両アーキテクチャーが求められます。 OTA に対する最適化の鍵は、車両のコンピューティング機能の一元化です。一元化されていれば、 1 箇所にアップデートをダウンロードするだけで済み、車両内のあちこちにある複数のシステムを別々にアップデートする必要がありません。

幸いにも、車両の電子的アーキテクチャーについてはすでに一元化を指向する流れができています。たとえば、 Aptiv のサテライトアーキテクチャーには、レーダー、カメラ、その他のセンサーから情報を収集して強力なドメイン コントローラーで一元管理する仕組みがあります。このアプローチでは小型・軽量のセンサーを使用できるため、パッケージングと設計の柔軟性が高まるだけでなく、メーカーの放熱対策もしやすくなります。同時に、サテライトアーキテクチャーではドメイン コントローラーに高度な機能を持たせることができます。たとえば、センサー フュージョンを使用してさまざまなセンサーからの入力を統合し、全体像を 1 つの環境モデルとして認識できます。

レーダーやカメラに搭載されるソフトウェアやファームウェアは一部残りますが、車両の稼働現場でそうしたコードのアップデート処理を行う必要はまずありません。さまざまなソフトウェアの機能やコンポーネントは、一元化されたコンピューターに組み込まれます。例として、車両の周囲にある物体を識別して動きを追跡するトラッカー ソフトウェアが挙げられます。

このアプローチでは、ソフトウェアの変更確認に必要なエンジニアリング リソースが少なくて済み、コストを抑えることができます。

車両に適用する (「フラッシュ」する) ソフトウェア アップデートの提供前には、すべての対象コンポーネントに新しい厳格なテスト条件を適用し、適切に機能することを確認しなければなりません。たとえば個々のスマート センサーにトラッカーが搭載されている場合、アップデートの適用前にすべてのセンサーの再検証を行う必要があります。一方、アップデートの影響を受けるソフトウェア コンポーネント (この例ではトラッカー) がドメイン コントローラーのみに搭載されている場合、必要な再検証作業の対象はドメイン コントローラーのみです。したがってプロセスが簡素化され、少ないリソースと短い時間で対応できます。

OTA アップデートでは、再検証以外にも、クラウドへのソフトウェア アップロード、クラウド管理、暗号化、ダウンロード、通信時間など、ある程度の関連コストが毎回発生します。アップデートをできる限り簡素化すれば、そうしたコストを抑えることができます。

アップデートを完全に一元化できない場合も、車両内の特定の 1 箇所をマスターにして、すべてのコンポーネントに関するアップデートをそこで管理することは有意義です。各所へのフラッシュ処理やコンポーネント間の互換性維持を確実に行えるからです。 Aptiv のスマート ビークル アーキテクチャー™ では、セントラル ビークル コントローラーがマスターの役割を果たします。もしアップデートに不備や破損があると、あるコンポーネントから別のコンポーネントに送られるデータの形式が合わず受信できないといった不適合が発生しかねませんが、このアプローチであればシステム レベルで不適合問題を回避できます。

ソフトウェアの一元化には、 OTA 以外にもメリットがあります。共通のプラットフォーム上でソフトウェア アプリケーションをコンテナ化すれば、アプリケーションの相互運用性テストやサイバーセキュリティ対策の適用が容易になります。

以上のような理由から、一元化の流れは業界内で加速し続けています。

より速く、より確実に

OTA アップデートの安全性と信頼性を最大限に高めるには、ドメイン コントローラーの各プロセッサーと、場合によってはセンサー上のプロセッサーに、古いソフトウェア イメージと新しいソフトウェア イメージの両方を保存できる容量のメモリが必要になります。システムの切り替えは、新しいソフトウェア イメージ全体のダウンロード、複合化、検証が済んだとき初めて可能になります。また、切り替えはアップデートの影響を受ける全プロセッサーで同時に行われなくてはなりません。これは、いずれのコンポーネントをいつ再起動してもよい状態を常に維持するために必要な仕組みです。

OTA アップデートの実施方法としては以下の 3 種類が採用されています。どの方法も徐々に速くなっています。安全性と柔軟性を確保するために、メーカーはより短時間で済む方法への切り替えを進めています。

1. **外部ストレージからのアップデート:** 外部のゲートウェイが携帯電話モデムを使用してクラウドと通信し、新しいソフトウェア イメージをバッファーに保存します。ドメイン コントローラーやセンサーに追加のメモリを搭載する必要がないため費用効果の高い方法であり、既存のコンポーネントを OTA 対応にすることも可能です。しかし、障害が発生した場合に以前のイメージを復元する機能は実現できません。先進運転支援システム (ADAS) のドメイン コントローラーをこの方法でアップデートする場合に必要な時間の目安は、最大 2 分程度です。

2. **ローカル ストレージからのアップデート:** 電子制御ユニット内のローカル ストレージに新しいソフトウェア イメージをダウンロードします。アップデート プロセスを実行すると、ローカル ストレージ上の新しいイメージが稼働用のフラッシュ メモリにコピーされます。レーダー センサーをこの方法でアップデートする場合に必要な時間の目安は 14 秒程度です。

3. **二重化フラッシュ メモリからのアップデート:** 古いソフトウェア イメージと新しいソフトウェア イメージの両方を A/B 構成で保持するのに十分なメモリがプロセッサーに搭載されていることが前提となります。新しいイメージへの切り替えは、ダウンロードと検証が完了した後、ほぼ瞬間的に行われます。切り替えによってシステムの可用性が低下することはありません。また、障害が発生した場合に以前のイメージを復元する操作も瞬間的に完了します。

これらの処理を行うタイミングは OEM の判断で決定されます。たとえば、車両の運転中にソフトウェア イメージをダウンロードしておき、次回の始動時に新しいイメージをインストールするといった方法が考えられます。また、夜間など、イグニッションがオフになっているときにアップデートのダウンロードとインストールを行う方法もあります。後者の方法をとるには、夜間などにシステムの電源をオンにする必要があるため、電源管理の方法もあわせて検討することが必要です。

アップデートの頻度

OTA を使用してアプリケーションをアップデートする頻度も OEM の判断によりますが、アプリケーションの種類に応じて決まる場合が多いと考えられます。

たとえば、コンプライアンスのために搭載されている安定した機能などについては、 OTA アップデートの頻度は低くてよいか、まったく不要な場合もあると考えられます。一方で、快適性のための機能を重視した車両構成では、頻繁なアップデートのメリットが出やすい可能性があります。また、プレミアム機能については、アップデートをほぼ継続的に随時行い、ユーザーにいつも最新機能を提供することがメリットになる場合もあるでしょう。当然、そのような提供方法では、バグ修正が必要な状況においても即座にバグ修正のアップデートを適用できます。

頻繁なアップデートは、継続的インテグレーション/継続的デプロイ (CI/CD) 型の開発・運用アプローチと非常によく馴染みます。これは IT 業界で生まれた考え方ですが、自動車業界にも急速に取り入れられつつあります。 CI/CD においては、開発者のもとで自動化ツールによって頻繁にソフトウェア アップデートが作成され、運用現場側の準備ができしだいアップデートが適用されます。 OTA テクノロジーは、こうした運用方法を実現可能にする方向で進化しています。

「OTA に対する最適化の鍵は、車両のコンピューティング機能の一元化です。一元化されていれば、1 箇所にアップデートをダウンロードするだけで済み、車両内のあちこちにある複数のシステムを別々にアップデートする必要がありません」

将来の展望: 部分的なアップデート

現在の OTA アップデートの多くはイメージ全体を入れ替える形でしか適用できない仕組みになっていますが、システムの急速な進化により、イメージの完全な入れ替えを行わずソフトウェアのごく一部のみをアップデートすることも可能になりつつあります。これは私たちが慣れ親しんでいる、スマートフォンで個々のアプリをアップデートする方法と似ています。実際、Android やその他のオペレーティング システムが車載用プラットフォームに進出しており、OTA を使用したアプリのアップデート提供がすでに始まっています。

このようなアップデートの必要性が高まっている背景には、車両に搭載されるソフトウェアが増え続けている状況があります。現在、多くの車両の自動運転レベルは 0 ～ 2 ですが、3 以上のレベルになると搭載ソフトウェアの量が大幅に増加するため、イメージの完全なアップデートを毎回行うのは現実的ではない可能性があります。

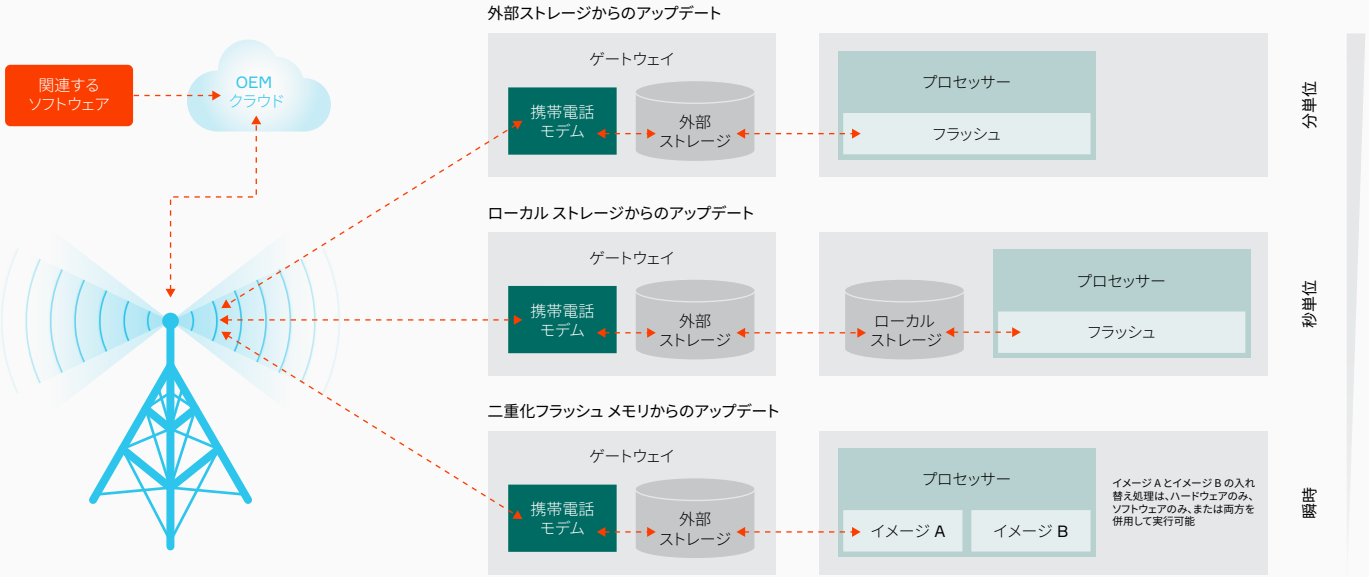
同時に、ソフトウェア定義で実現される機能が増えることは、ソフトウェアというパズルのピースとなるコードの行数やソースの量が増えることを意味します。OEM の皆様としては、トラッカーや定速走行・車間距離制御機能など、車両に搭載されているソフトウェアの一部のみを単一のソースからアップデートでき、パズルの他のピースには手を加えないで済むのが理想的でしょう。アップデートを展開して適切な場所に配置し、システムを再起動する処理は、専用のソフトウェアによって実行されるのがよいと考えられます。これが、ソフトウェア アーキテクチャーがきわめて重要である理由です。コンテナ化によってそれぞれのピースを個別に保持し、増分アップデートを可能にする必要があります。

どのような OTA アップデートでも、プロセスのすべての段階で最大の懸念事項となるのはセキュリティです。アップデートの提供前には、エンジニアがコードの徹底的な分析を行い、車両に搭載されている他のソフトウェアとうまく連動するか、また脅威や一般的な脆弱性がないかを確認する必要があります。ベストプラクティスについては ISO/SAE 21434 自動車サイバーセキュリティ規格に記載されていますが、基本的にはシステム自体が「危害を想定」して設計されていることが必要です。つまり、新しいソフトウェアは意図的 (サイバー攻撃の場合) または非意図的に危害を引き起こす可能性があると想定し、そのような危害を阻止するための事前対応型または事後対応型の対策を講じる必要があるのです。

「ソフトウェア・ディファインド・ビークルの成功は、ユーザーが他の製品に期待するようになった継続的な機能向上を OTA で実現できるかどうかにかかっています」

再フラッシュの方法

ソフトウェアがクラウド上に提供されると、車両のシステムがアップデートをダウンロードし、一元化された外部ストレージに保存します。
その後、アップデートを個々のプロセッサーに適用する方法には複数の種類があります。



新しいビジネスモデル

車両機能の継続的改善が現実的なものになった一方で、OTA を活用した新しいビジネスモデルの実現可能性も同様に注目されています。

たとえば、サブスクリプション サービスを通じて機能を提供し、継続課金によってその機能の料金を請求するビジネスモデルが考えられます。また、週末に出かけるときだけ高速道路運転支援機能を有効化したいと考えるドライバーもいるかもしれません。OTA では、サブスクリプションが有効なときに限り該当機能のダウンロードや利用が許可される環境も実現できます。

車両に十分なメモリとコンピューティング能力があり、一連のレーダー、カメラ、その他のセンサーが装備されている場合、OTA を通じてソフトウェアを提供するだけで、まったく新しい機能を使用できるようになります。

OTA は次世代の電気的・電子的アーキテクチャーに欠かせない要素です。Aptiv のスマートビークルアーキテクチャー™ アプローチと次世代の ADAS プラットフォームは、OTA アップデートをできるだけ効率的かつ安全に受信できるように最適化されています。

ソフトウェア・ディファインド・ビークルの成功は、ユーザーが他の製品に期待するようになった継続的な機能向上を OTA で実現できるかどうかにかかっています。ソフトウェア プロバイダーのエコシステムを育て、ソフトウェア プロバイダーによるイノベーションを活用し、また、そのイノベーションの力で車両の稼働寿命が続く限り充実した利用環境を提供するうえで、OTA は鍵となる大切な要素です。

著者について

Nabeel Bitar
車両システム アーキテクト – アドバンスド セーフティ スタート センター所属

Nabeel Bitar は、高度な安全性の分野において、世界中の OEM を対象とした新しいビジネスチャンスについての技術的コミュニケーションと設計を担当しています。Aptiv では、25 年以上にわたってソフトウェアとシステム エンジニアリングの職務に従事し、2003 年から ADAS と自動運転システムの開発に携わってきました。これまでに、デルファイ・サギノー・ステアリング社 (現ネクステア社) でアンチロック ブレーキやトラクション コントロール ソフトウェア、電動パワー ステアリング システムの設計を担当しました。

詳細については、APTIV.COM/CONNECTIVITY-SECURITY をご覧ください →