

オープン・サーバー・プラットフォームに必要なハードウェアとソフトウェアの転換

ソフトウェア定義車両の最大の強みは、進化することだ。車両は一定の機能を備えて工場から出荷されるが、OEMは使用データを活用し、無線アップデートによって車両のライフサイクル全体にわたってこれらの機能を拡張することができる。これにより、より優れた走行性能を実現し、車内のユーザー・エクスペリエンス（UX）を向上させ、老朽化した車両を毎日新しく感じられるようにすることができる。

しかし、ハードウェアは進化しない。今まで車両が工場から出荷されると、ハードウェアが変わることはない。

この二分はやがて、Software-Defined Vehicleに難題をもたらす。コンピュータを集中化しサーバー化することは開発負荷を共有し、リソースを再配分し、その有用性を拡張するためのコスト効率の高いソリューションとなり得る。しかし、ある時点でコンピュータ・ハードウェアは、より高速な処理、より多くのメモリ、より大きなストレージ容量を必要とする新しいソフトウェア機能に対応するには古い仕様になってしまう。

OEMは、適切なソフトウェアとハードウェアのアーキテクチャを用いることで、このような制限に対処することができる。このアーキテクチャは、リアルタイムでリソースを動的に再配分できる柔軟性を持つだけでなく、将来にわたり最も必要とされる特定のコンピュータコンポーネントだけを物理的にアップグレードすることも可能だ。このアプローチが正しく行われれば、独立したソフトウェアとハードウェアのエコシステムを構築することが出来、同時に自動車の寿命を大幅に延ばすことができる。

脳の手術

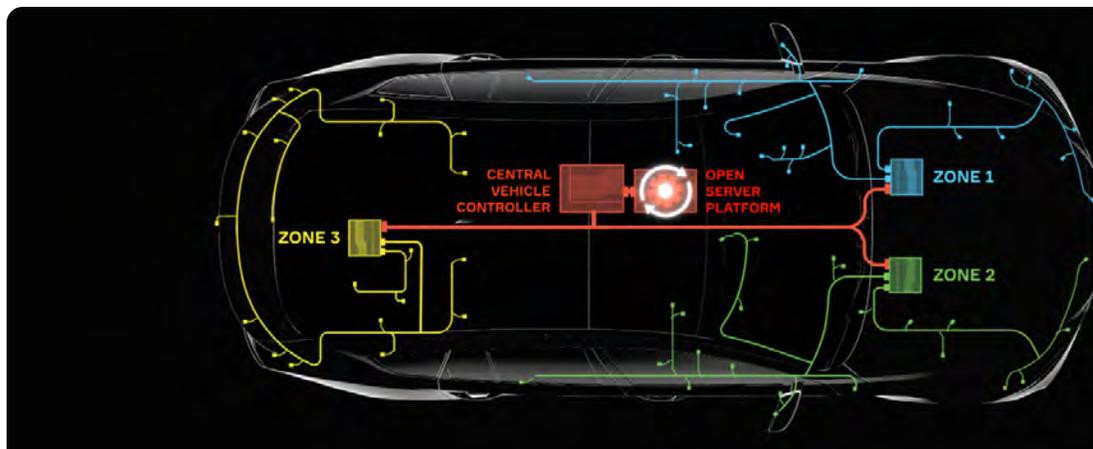
我々は自動車のコンピュータを“脳”と考える。人間の脳が情報を取り込み、その入力に基づいて行動するように、車両のコンピュータ・プラットフォームはセンサーやその他の外部ソースからデータを取り込み、何をすべきか、何を伝えるべきかを決定する。

しかし、人間の脳は時間とともに変化し、成長する。赤ちゃんは子どもへと成長し、子どもは大人へと成長する。脳が物理的に変化するにつれて、彼らはより深く理解し、より複雑な推論ができるようになる。

これとは対照的に、コンピューターは静的であり、処理の限界も決まっている。ソフトウェア開発者は、より高度で高性能なアプリケーションを開発するにつれ、その限界を押し広げ、開発した機能を実現するために次世代ハードウェアへの移行が必要になる。

自動車の世界では進化する安全規制を満たしたり、最新のサイバーセキュリティ対策を採用し続けたりするために、より新しいハードウェアが必要になることもある。

消費者はこのようなアップグレードサイクルには他製品で馴染みがあるはずだ。例えば携帯電話では最新のアプリや機能、サイバーセキュリティ保護を適切にサポートできるプロセッサを手に入れるために多くの人が2~3年ごとに新しいデバイスに切り替える。もちろん、重要な違いは、携帯電話全体を交換する費用は、自動車全体を交換する費用よりもはるかに少ないということである。また、車両は明らかにはるかに複雑であり、数百ものデバイスが分散して配置され、中央のコンピュータ・ユニットに接続されることが多くなっている。



OEMは、適切なソフトウェアとハードウェアのアーキテクチャを用いることで、このような制限に対処することができる。このアーキテクチャは、リアルタイムでリソースを動的に再配分できる柔軟性を持つだけでなく、将来にわたって最も必要とされる特定のコンピュータ・コンポーネントだけを物理的にアップグレードすることもできる。このアプローチが正しく行われれば、独立したソフトウェアとハードウェアのエコシステムを構築しながら、車両の寿命を大幅に延ばすことができる。

ハードウェア・アーキテクチャ

自動車における解決策は、OEMがより高度な機能を実行するために必要なコンピュートだけをアップグレードできるように、ソフトウェアとハードウェアのアーキテクチャを構成することである。

ハードウェア面では、OEMはすでにゾーン・アーキテクチャに移行しており、車両内の多くのデバイスが最も近いゾーン・コントローラに接続し、そのゾーン・コントローラがそれらのデバイスからのデータを集約してセントラルビークルコントローラ（CVC）と通信する。CVCは、ゾーン・コントローラと連携してすべてのデータ通信プロトコルとデバイスとのシグナリングを処理するほか、車体制御機能、データ・ストレージ、車両アクセス、外部との通信、そして潜在的には推進力とシャーシ制御を処理する。

言い換えれば、CVCとゾーンコントローラは、車両を走らせるのに必要な低レベルの機能をすべて管理する。このアーキテクチャにより、より高度な機能を別のコンピュート・プラットフォーム（オープン・サーバー・プラットフォーム、OSP）で

サポートすることが可能になる。

OSPは車両の頭脳であり、複雑でデータ指向の高い計算能力を必要とする機能を実行する。主に先進運転支援システム（ADAS）や車内UXに関連する機能だ。実際には、OSPが脳であるのに対し、CVCは小脳と捉えられる。なぜならCVCはOSPからの指示を車両の動きに繋げるからだ。

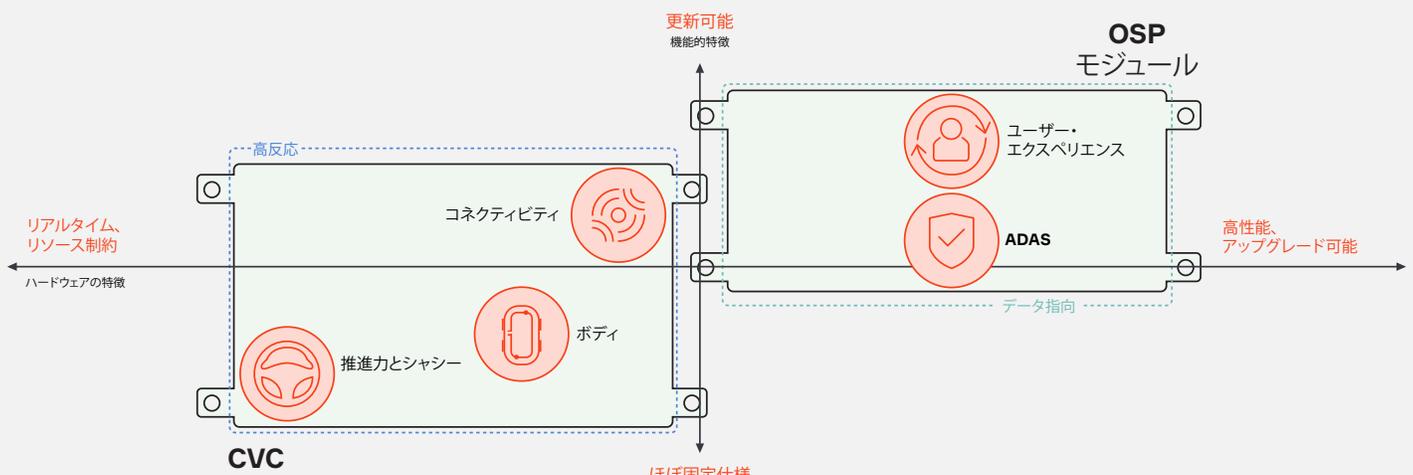
この2つをアーキテクチャ的に分離することは、非常に理にかなっている。低レベルの機能と車両の入出力との通信は、車両の寿命が尽きるまであまり変わらない。しかし、より高度な機能は、開発者がより多くの使用例に関わったり、より革新的なUX機能を開発したりするにつれて、絶えず進化していく。これらの機能には、より強力な処理能力、より多くのメモリ、グラフィック・プロセッシング・ユニットが必要になる。

ソフトウェア・アーキテクチャ

ソフトウェア・アーキテクチャもまた、分離をサポートするために適切な構造になっていなければならない。重要な要素は、抽象化とインターフェースの

分離と相違性

車両のソフトウェア・ドメインによってニーズが異なるため、ハードウェア・アーキテクチャはその違いに最適に対応するような設計でなければならない。つまりCVCまたはOSPのどちらかで、必要に応じて機能を実行することになる。



標準化である。つまり、上位レベルのソフトウェアに標準的で一貫性のあるインターフェースを提示することで、上位レベルが下位レベルの動作方法の詳細にこだわる必要がなくなる。

最も低いレベルにあるのがデバイス抽象化レイヤー（DAL）だ。デバイスには、センサー（レーダー、カメラ、サーマルセンサーなど）からアクチュエーター（シートコントロール、ドアロック、ウィンドウリフターなど）まで、あらゆるものが含まれる。ゾーン・コントローラーはデバイスとのすべての直接通信を処理し、DALはマイクロサービスを通じて制御と診断のための上位レベルのソフトウェアにアプリケーション・プログラミング・インターフェース（API）の標準セットを提示する。

例えば、上位のソフトウェアは、DAL APIを通じて窓を下げるよう要求することができるが、そのモーターとどのように会話するかを知る必要はない。代わりに、ゾーンコントローラーは、ローカル相互接続ネットワークバスを通じて適切な信号をフォーマットし、窓モーターに送信する。センサーデータのストリーミングも標準化されるだろう。

次のレベルは車両抽象化レイヤー（VAL）で、CVCが管理する複雑なボディ制御機能を抽象化する。高レベルのソフトウェアは、VALのAPIと相互作用してデータを取得したり、アクションを起こしたりする。

ウィンドウリフターの例を続けると、VALレベルの抽象化には、ユーザーコントロールやウィンドウリフターのモーターなど、ウィンドウ操作のあらゆる側面を管理するサービスを含めることができる。車内のUX機能が、インフォテインメントシステムを通じて、ユーザーからすべての窓を50%下げる要求を受けるとするなど。

コンテナ化

ソフトウェア・アーキテクチャのもうひとつの重要な要素は、コンテナ化だ。ソフトウェア・コンテナは、ファイルやライブラリのすべてを、それらを必要とするアプリケーションとともにパッケージ化し、コードをホスト環境から比較的独立させる。このアプローチにより、サービス指向のアーキテク

チャを構築することができ、各サービスはそれぞれのコンテナに関連付けられ、他のソフトウェアに影響を与えることなく個別に更新することができる。

コンテナ化はまた、ソフトウェア開発と、その上で実行されるハードウェアを切り離すのにも役立つ。つまり、コンテナ化されたサービスは、あるコンピ

ソフトウェア・コンテナは、ファイルやライブラリのすべてを、それらを必要とするアプリケーションとともにパッケージ化し、コードをホスト環境に対して独立させる。

ュート・プラットフォームから別のプラットフォームに移動できる可能性がある。また、コンテナ化されたアプリケーションは実行するハードウェアに縛られないため、OSPを構成するハードウェアを、例えばあるシステムオンチップから別のシステムオンチップへと変更することもできる。

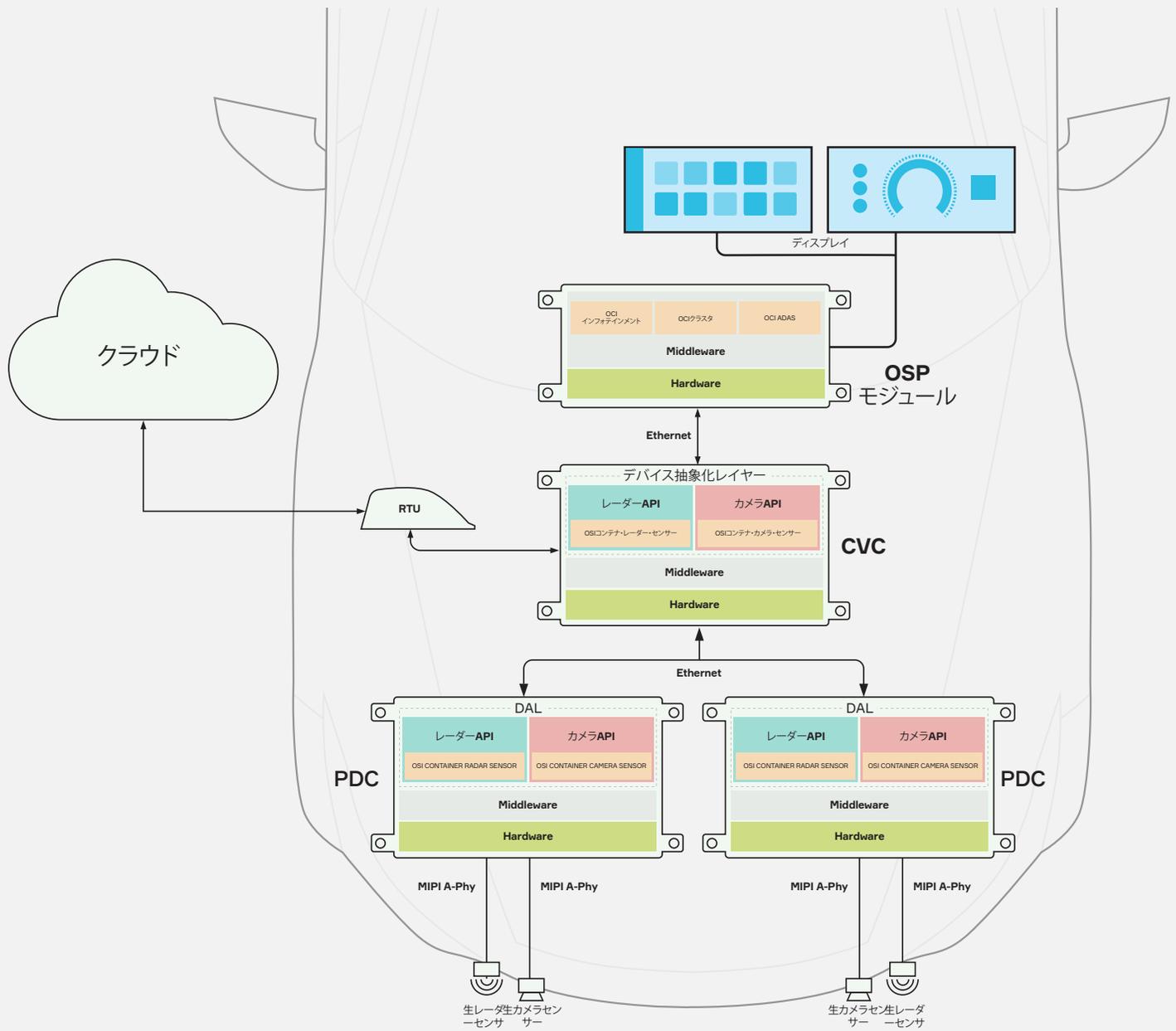
ひっくり返った台本

これらの主要なアーキテクチャ要素が整えば、OSPは交換可能になる。ハードウェアは、上位機能がOSP上で分離された構造になっている。ソフトウェアは、OSP上のあらゆるアプリケーションが標準インターフェースを通じて車両と通信するように構成されている。また、ソフトウェアのコンテナ化によって、OSP上のコンピュータへのアプリケーションのアクセスを抽象化する。OSPが古くなり、最新のアプリケーションを実行できなくなったり、最新のOSアップデートを受けられなくなったりした場合、オーナーはディーラーにOSPを持ち込んで、より新しく高性能なOSPと交換することができる。

しかし、その影響はOSPにとどまらない。これらの

抽象化のレイヤー

このアーキテクチャの各デバイスは、より高度な機能を実現するためにソフトウェアの抽象化を使用している。パワーデータセンター（PDC）のゾーンコントローラーは、デバイスとの通信を処理し、セントラルビークルコントローラー（CVC）にデバイス抽象化レイヤー（DAL）を提示し、中央車両コントローラーは、オープン・サーバー・プラットフォーム（OSP）上で動作するアプリケーションにより複雑な機能を提供する。



要素を総合すると、独立したソフトウェアとハードウェアのエコシステムが可能になる。

- マシン・エコシステム：ハードウェアは、アプリケーション・ソフトウェアとは無関係に開発し、認証することができる。性能は、処理速度やメモリ容量などの標準的なベンチマークで評価できる。開発サイクルは18ヶ月以内。
- デバイスのエコシステム：車両に搭載されるすべてのデバイスは、独自のハードウェアとソフトウェアを組み合わせるが、標準的なAPIに合わせる。これらは標準的なバスで接続される。それらの開発は、特定のマシンやアプリケーションに接続されることはない。
- アプリケーションのエコシステム：コンテナ化されたアプリケーションは、ソフトウェアを変更することなく、あるハードウェア・プラットフォームから別のプラットフォームへ移行することが可能。明確に定義されたAPIを備えた標準的なUXとADASのスタックは、スケーリングに役立つ。また、すべてのアプリケーションをクラウド環境で開発できる。

これらのエコシステムは独立しているため、ベンダーロックインは時代遅れになる。デバイス、アプリケーション、マシンはすべて、他のエコシ

テムを邪魔することなく交換することができる。

交換可能なOSPを持つことで、OEMはモデルやトリムレベルを簡単に差別化することもできる。2つのモデルで同じデバイスとCVCを搭載した場合、ソフトウェア機能に関連するハードウェアで異なるOSPを接続するだけで、異なるADASまたはUX機能を持つことができます。

SVA™のビジョン

Aptivのスマート・ビークル・アーキテクチャー (SVA)™テクノロジーは、これらの要素をすべて考慮しながら、将来の機能強化のための基礎を築いている。

ハードウェア面では、当社のOSPモジュールはスタンドアロン、またはCVCの上に直接設置する設計が可能で、間に液冷プレートを挟むことで温度を抑えられる。この構成により、ハードウェアコストがさらに最適化され、アップグレードが必要な場合にOSPの交換が容易になる。

車載イーサネットと PCI Express は、OSP を CVC に接続する。ADAS に特化したコンピュータ・ソリューションの現在のバージョンは、一部のセンサーに直接接続されているが、将来のバージョンで



このようなハードウェアとソフトウェアの基礎要素で車両が構築されれば、ソフトウェア定義の車両の真の可能性を実現する道は開け、今後何年にもわたって最新の機能を確保することができる。

は、すべてのセンサー接続がゾーン・コントローラと CVC を経路するようになる。しかし、UX に特化した OSP 実装の中には、アーキテクチャのコスト制約内で高帯域幅要件をサポートするために、高解像度ディスプレイに直接接続し続けるものもある。

ソフトウェア面では、Signal-to-Serviceの抽象化を進めており、ウインドリバー社の買収により、同社のVxWorks®リアルタイムOSを利用でき、セーフティクリティカルな環境でOpen Container Initiative準拠のコンテナをサポートする。

SVA™技術が進化するにつれて、我々のアプローチはレベル3以上の自動運転に必要なフェイルオペレー

ション冗長性を提供する。コンテナ化による柔軟性により、アプリケーションは1つのOSPから冗長性のために2つ目のOSPに移動することができ、CVCの上に設置されたOSPモジュールはそれをサポートするのに理想的だ。

このようなハードウェアとソフトウェアの基礎要素で車両が構築されれば、今後何年にもわたって最新の機能を確保することで、ソフトウェア定義の車両の真の可能性を実現する道が開けるのだ。

交換可能な構造

オープン・サーバー・プラットフォーム (OSP) モジュールは、アップグレードが必要になったときに技術者が簡単に交換できるように設計されているのが理想的です。



多機能

ADASとUXを扱うか、AI/ML、ゲートウェイイラーティングなどの追加データ加速機能を提供する。



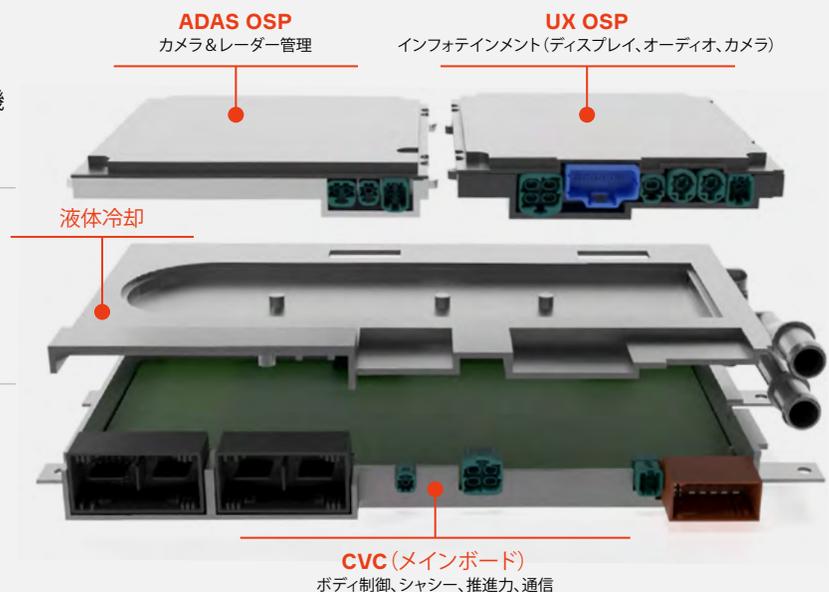
モジュール

CVCユニットによって冷却される交換可能なOSPモジュール



効率的な熱管理

車の寿命まで使えるよう設計された液冷



著者について



マルティン・ボルネマン
Vice President, Advanced Technology & Architecture,
Chief Technology Office

AptivのCTOオフィスで先端技術とアーキテクチャの開発を担当。Aptivに20年以上在籍し、イノベーション・マネジメント、プロジェクト・マネジメント、ハードウェア開発などの役職を歴任。アプティブ入社以前は、エリクソン社で通信機器の設計、ボッシュ社で無線LANの研究を行っていた。



ベンジャミン・グールド
Product Management Director, Compute
Global Product Organization

ソフトウェア・デファインド・ビークルの頭脳となるAptivの次世代コンピュータの定義を担当。2021年にチーム・カスタマー・エンジニアとしてアプティブに入社し、それ以前はモビルアイでシニア・プロジェクト・マネージャーを務めていた。インテルでキャリアをスタートし、インダストリアル・エンジニアリング、アプリケーション・エンジニアリング、テクニカル・マーケティング、プロダクト・マーケティング・マネジメントに20年間従事した。ケース・ウェスタン大学で電気工学と応用物理学の学士号を取得。



セザリー・クリマシュ
Technical Program Manager, Compute
Global Product Organization

Aptivのセントラルビークルコントローラとオープンサーバープラットフォームの設計と開発をリード。チームは高性能ハードウェアと最先端のソフトウェアソリューションを活用した次世代コンピューティングプラットフォームに取り組んでいる。2012年にハードウェアエンジニアとしてAptivに入社後、製品および技術管理でさまざまな職務に就いている。

APTIV.COM/OSP で詳細をご確認下さい。