OTA Updates Require Flexible Architecture

A vehicle sits quietly overnight in a garage, unmoving until its driver returns and takes it on its next trip. But instead of simply waiting, this vehicle is expanding its capabilities, becoming safer, increasing its intelligence and learning to perform functions just as well as newer vehicles rolling off assembly lines that same day.

Over-the-air (OTA) software updates make this possible. When integrated with the vehicle architecture, OTA allows manufacturers to evolve their vehicles' active safety, infotainment and other domains in an elegant, scalable, low-risk and cost-effective way, long after they leave the factory.

An architecture designed with centralized compute and optimized for OTA can help ensure that software and firmware updates happen smoothly and securely, at whatever frequency is required by the application, the OEM strategy and consumers' preferences. Not only is OTA essential to build software-defined vehicles, but it also opens up the potential for innovative new services, new ways to deliver features and new business models — providing flexibility and extensibility never before possible.

ENABLING THE SOFTWARE-DEFINED VEHICLE

Major OEMs are releasing vehicles that are software-defined with the intent to deliver more features at a later date, using cellular or Wi-Fi networks to deliver OTA updates to software and firmware. Increasingly, the features reside in complex, safety-critical systems, so it is imperative that these updates are done safely, reliably and securely.

While OTA is becoming ubiquitous, it is not a capability that can just be bolted on; rather, the vehicle architecture should be designed with OTA in mind. Key to optimizing for OTA is centralizing the compute power within a vehicle, so that updates only have to be downloaded to that central location, rather than distributed to systems throughout a vehicle.

Fortunately, vehicle electronic architectures are already moving toward centralization. Aptiv's Satellite Architecture, for example, takes the intelligence of radars, cameras, and other sensors and centralizes it in a powerful domain controller. This approach keeps the sensors small and light, which improves packaging and design flexibility, while also allowing manufacturers to better manage heat dissipation. Simultaneously, Satellite Architecture enables advanced capabilities at the domain controller, such as using sensor fusion to unite the inputs from various sensors into a cohesive environmental model.

Some software or firmware content remains with the radar or camera, but that code rarely requires changes in the field. Software features and components are pulled into the central compute — for example, the tracking software, which identifies objects around the vehicle and tracks their movements.

This approach keeps costs down, because it requires fewer engineering resources to verify any software changes.

Before any software is updated — or "flashed" — every component that is updated must go through a new, rigorous set of testing conditions to ensure that it works properly. If the tracker, for example, resides in smart sensors, then every sensor would have to be reverified before the update is deployed. However, if the tracker — or other software components affected by an update — resides in the domain controller, only the domain controller needs reverification. That simplifies the process and requires fewer resources and less time.

Every OTA update will have certain costs associated with it beyond reverification, including the costs of uploading the software to the cloud, cloud management, encryption, downloading and airtime usage. Simplifying those updates as much as possible can help keep costs down.

If the updates cannot be completely centralized, it still makes sense to designate one master to control updates for every other component in the vehicle, ensuring that they are all flashed correctly and compatible with one another. In Aptiv's Smart Vehicle Architecture™, the Central Vehicle Controller is the master. This approach ensures that the system avoids a mismatch where, for example, one component is expecting data in a certain format from another but does not receive it because of an incomplete or corrupted update.

Beyond OTA, centralization of software has other benefits as well. With software applications containerized on a common platform, performing interoperability tests and applying cybersecurity for those applications becomes much easier.

For all of these reasons and more, centralization continues to gain momentum in the industry.

FASTER AND MORE SECURE

To achieve the safest and most reliable OTA updates, each processor in a domain controller — and processors in sensors if necessary — must have the memory capacity to hold both the old software image and a new one. Only when new software images are fully downloaded, decrypted and verified would the system switch over, and all affected processors would do so simultaneously. This ensures that every component continues to be rebootable at all times.

Three methods have been used to facilitate OTA updates. Each method is progressively faster, and more manufacturers are turning to the faster methods to ensure safety and flexibility.

1. **Update from external storage.** In this scenario, new software images are buffered in an external gateway that communicates with the cloud through a cellular modem. This method is cost effective, as it does not require additional memory in the domain controller or in sensors. It also allows any current component to be OTA-ready. However, there is no ability to revert back to previous images if a failure occurs. A typical update for an advanced driver-assistance system (ADAS) domain controller could take up to 2 minutes.

2. **Update from local storage.** In this scenario, new software images are downloaded to local storage within the electronic control unit. An update process copies the new image from local storage to the active flash memory. Using this technique to update a radar sensor, for example, would take about 14 seconds.

3. **Update from doubled flash memory.** In this scenario, the processor contains enough memory to hold both the old software image and the new image in an A/B configuration. Once the download is complete and verified, switching to the new image is nearly instantaneous. The switchover has no impact on system availability. If a failure occurs, reverting to the previous image is also instantaneous.

The timing for when all of this happens is up to the OEM. One approach would be to download a software image while the vehicle is driving, and then install the new image the next time the vehicle starts up. Another approach would be to download and install updates when the ignition is off, perhaps overnight. The latter method requires that the system be powered during that time, so power management must be taken into account.

HOW OFTEN?

How often an application is updated through OTA is also up to the OEM and will likely depend on the type of application.

For example, a mature feature intended to meet compliance requirements may not need OTA updates very often or at all. But a configuration that focuses on comfort features might benefit from more regular updates, and a premium function might benefit from near-continuous updates to ensure that consumers have the very latest features available in their vehicles at all times. Naturally, this also means that any bug fixes would be rolled out as soon as the situation requires them.

Frequent updates dovetail well with the practice of continuous integration / continuous deployment (CI/CD), an approach to development and operations that began in IT and is rapidly being adopted in the automotive industry. With CI/CD, developers use automated tools to make software updates more frequently and get those changes into the field as soon as they are ready, and OTA technology is evolving to enable that.

"Key to optimizing for OTA is centralizing the compute power within a vehicle, so that updates only have to be downloaded to that central location, rather than distributed to systems throughout a vehicle."

THE FUTURE: PIECEMEAL UPDATES

Today, most OTA updates require reprogramming an entire image. However, systems are quickly evolving to allow smaller portions of software to be updated without a full image swap — just as we are accustomed to smartphones allowing updates to individual apps. In fact, as Android and other operating systems move to automotive platforms, they are already beginning to drive app updates through OTA.

The need to allow such updates increases as the amount of software in a vehicle continues to grow. While many vehicles today operate at automated driving Levels 0 through 2, Levels 3 and beyond will have much more software, and updating a full image every time could be prohibitive.

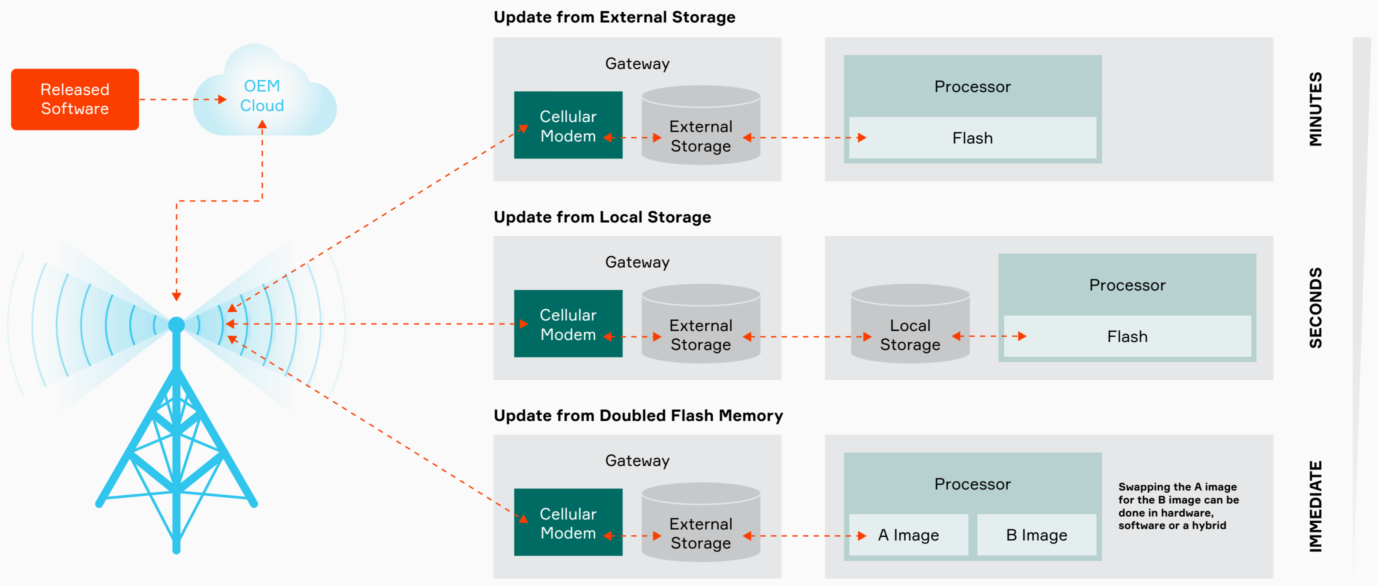
At the same time, the growing functionality defined by software means more lines of code and more sources providing pieces of the software puzzle. Ideally, OEMs would want to have the ability to update just the portion of the software on a vehicle from a single source — for example, the tracker, or the adaptive cruise control function – and leave the other pieces intact. Dedicated software would unpack those updates, put them in the relevant locations and reboot the system. This is why the software architecture is extremely important; it must enable the containerization that keeps the pieces separate and allows these incremental updates.

With any OTA update, security is the top concern at every step in the process. Before the update is deployed, engineers must analyze the code thoroughly, not only to ensure that it works well with other software in the vehicle, but also to check for threats and common vulnerabilities. Best practices are laid out in the ISO/SAE 21434 automotive cybersecurity standard, but essentially the systems themselves must be designed with an “assume harm” mentality — that is, to assume that any new software could potentially cause harm, either intentionally (if there is a cyberattack) or unintentionally — and to take proactive and reactive measures to contain that harm.

“The success of the software-defined vehicle depends on OTA for the continuous evolution of features that consumers have come to expect in other products.”

Reflashing scenarios

As software is released to the cloud, vehicles download the updates to centralized external storage. From that point, there are several ways to propagate those updates to individual processors.



NEW BUSINESS MODELS

The possibilities for continuous improvements to vehicle features are promising, but equally interesting is the opportunity for new business models that leverage OTA.

One model might be to offer features through a subscription service, charging for the features through recurring billing. Or a driver might want to activate a highway driving assist just long enough for a weekend getaway. OTA would allow the features to be downloaded or enabled for only the time that the subscription is active.

When a vehicle has enough memory and compute capacity, coupled with an array of radars, cameras and other sensors, all that's needed for all-new features is the software, delivered through OTA.

OTA is a critical piece of any next-generation electrical and electronic architecture, and Aptiv has designed our Smart Vehicle Architecture™ approach and our next-generation ADAS platform to ensure that they are optimized to receive OTA updates as efficiently and securely as possible.

The success of the software-defined vehicle depends on OTA for the continuous evolution of features that consumers have come to expect in other products. OTA is a key ingredient in fostering an ecosystem of software providers, tapping into their innovation and allowing that innovation to flourish over the life of a vehicle.

ABOUT THE AUTHOR

Nabeel Bitar
Vehicle Systems Architect – Advanced Safety Start Center

Nabeel Bitar is responsible for global OEM technical discussions and designs for new business opportunities in Advanced Safety. Nabeel has been with Aptiv for more than 25 years in software and systems engineering roles and has been developing ADAS and autonomous driving systems since 2003. Previously, Nabeel designed anti-lock brake and traction control software, as well as electric power steering systems at Delphi Saginaw Steering (now Nexteer).

LEARN MORE AT APTIV.COM/CONNECTIVITY-SECURITY →