

## **ECU consolidation; learning from other industries**

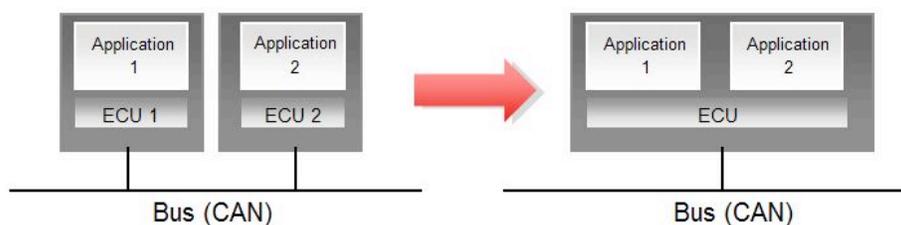
*By Georg Doll, general manager of automotive solutions at Wind River, an Intel company*

The number of ECUs you'll find in even the most standard of car models is staggering. There has long been a drive to consolidate ECUs but it is not just the number of them that is the challenge. The complexity of new functions such as adaptive cruise control, blind spot detection and hill start assist further complicate the software development and testing process. There is also an increasing appreciation and adoption of safety standards for the functional aspects of software used for some of these functions. ECUs used for different functions may have different Automotive Safety Integrity Levels (ASIL) and ISO 26262 requirements compared to other ECUs so this can be a major consideration when reviewing the potential of consolidating two or more ECUs. Having a separate ECU for each function also allows it to maintain its own criteria regarding dynamic behavior, security and functional safety requirements. Needless to say testing and debugging more complex ECU environments will bring certification challenges.

For the automotive manufacturer there are also potential business issues to consider. Traditionally ECUs have been purchased from Tier 1 suppliers for dedicated functions. Consolidating them will potentially require far more collaboration and design effort from a single supplier and it will also mean placing more emphasis, or risk, with a reduced number of suppliers.

However, the automotive market is not the only one where consolidation of systems has been achieved. There is a lot to learn from the aerospace industry where integrated avionic systems are becoming the norm.

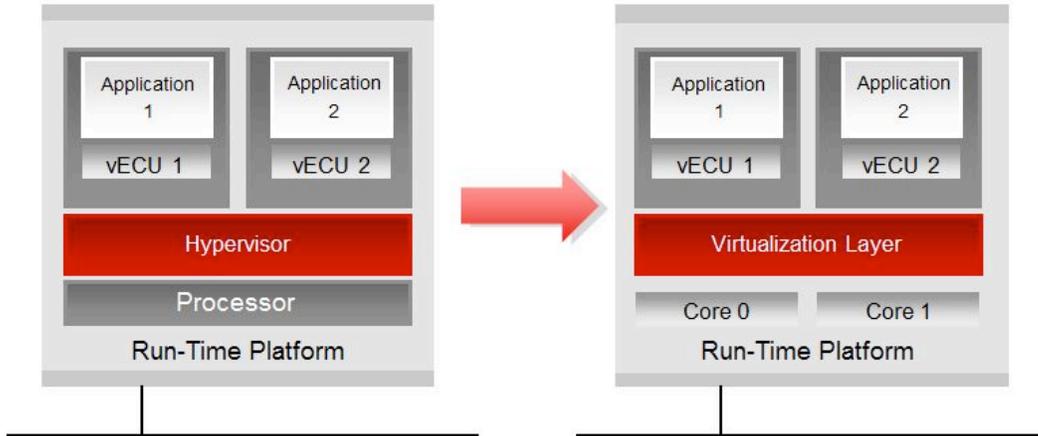
So how can consolidation be achieved? One approach, shown in figure 1, focuses on software integration. A single CAN-based ECU uses a common operating system such as OSEK or AutoSAR to run multiple applications. While possible there are a number of concerns with this approach. Namely that applications need to be optimized for the operating system and that a shared memory model may cause errors to spread and there is no simple way to isolate faults. The amount of testing required would also be significant.



**Figure 1.** ECU consolidation via software integration.

Another approach, recently taken by a number of vendors, is to virtualize the ECU. This is achieved by using a virtualization layer to run multiple ECU operating systems on a single processor. While this does make it easier to isolate faults there is the need for an additional integration step which could slow down the overall development. However, the logical next step in the virtualized ECU approach is to add multiple cores. See Figure 2.

The basic concept is centralizing compute power into function-oriented regions, decoupling software functionality from the underlying hardware using virtualization technology, and deploying virtual ECUs on multi-core processors so there is little interference between them, as shown below.



**Figure 2.** A better option: Virtual ECU + multi-core.

This approach allows the consolidation of many software-driven functions to a smaller number of more compute intensive hardware platforms. Most importantly, from the function safety aspects of separation each application is logically separately from others but each can still have dynamically allocation CPU resources in order to meet function-specific performance requirements. This method also aids the project development cycle by moving integration and test to earlier stages of the project. There are additional benefits too, one of which being that security for the ECU can be built as a single virtual function rather than having separate functions for each ECU. In this way it is easier to manage and secure the consolidated ECU platform.

+++Ends