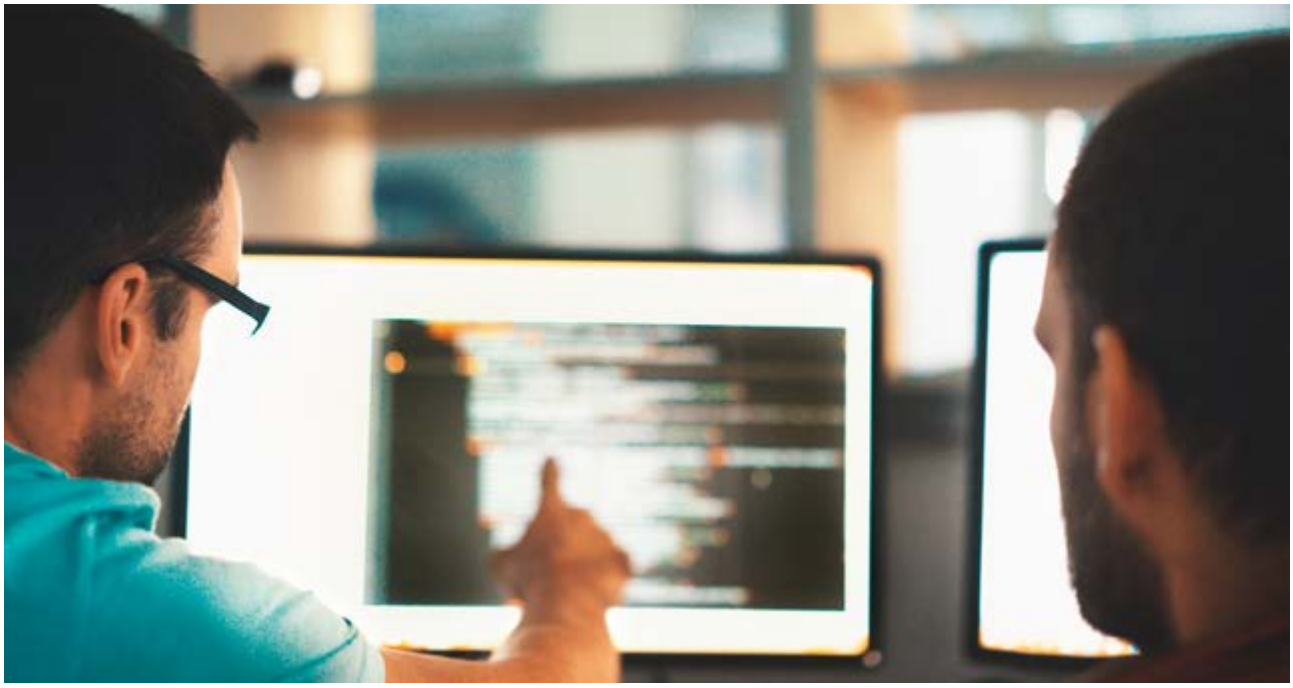
Translated article “Strikte Trennung erforderlich,“ Automoblil Elektronik 11-12/2017

Control unit software

Partitioning enables updates in the field and makes electronics safer

Where control unit software is concerned, the automotive industry increasingly relies on agile software development. With agile software development, software-controlled vehicle functions are kept current via continual updates and upgrades -- also in the field. For these updates to be installed without safety risks, the individual functions must be kept absolutely separate. At same time, the trend is towards increasing connectivity of ever more functions on central control units. Partitioning is a way of solving this contradiction. Previously, this was not technically possible with the microcontrollers in automotive control units. The Lightweight Hypervisor from ETAS is the answer.



For car buyers, agile software development with updates in the field is good news as it can offer real added-value. Instead of having to make sure all their needs and wishes are covered at the time of purchase, it will now be possible to gradually add or change functions at a later stage. Examples include multimedia packages, navigation software, or a software-controlled throttle of engine power for when teenagers who have just passed their driver's license are at the wheel.

For the manufacturers of vehicles, control units and control unit software, however, subsequent function upgrades and continual (over-the-air) updates pose new challenges and carry new risks. They need a guarantee that modifications will not under any circumstances affect other possibly safety-relevant software functions that are running on the same control unit. The trend towards concentrating ever more connected functions on ever fewer central control units does not make the task any easier. And the following question arises: How can we make it economically possible under the given circumstances to carry out validation and verification tests beforehand so as to guarantee the functional safety of the overall system following upgrades and updates?

Practicable partitioning is needed

Often, different companies provide software functions for one and the same control unit. It is important that their respective software modules do not interfere with one another and one manufacturer's problems do not have repercussions for the others. Ideally, each company's patch should be clearly delimited and separated from the others so that every company is only responsible for the smooth operation of its own software – including all the tests specified in the applicable safety standards. In case of post-production changes, responsibility would also be limited to those areas. This kind of "freedom of interference" is specified by the applicable safety standards but is difficult to verify with conventional control unit architecture.

Key data

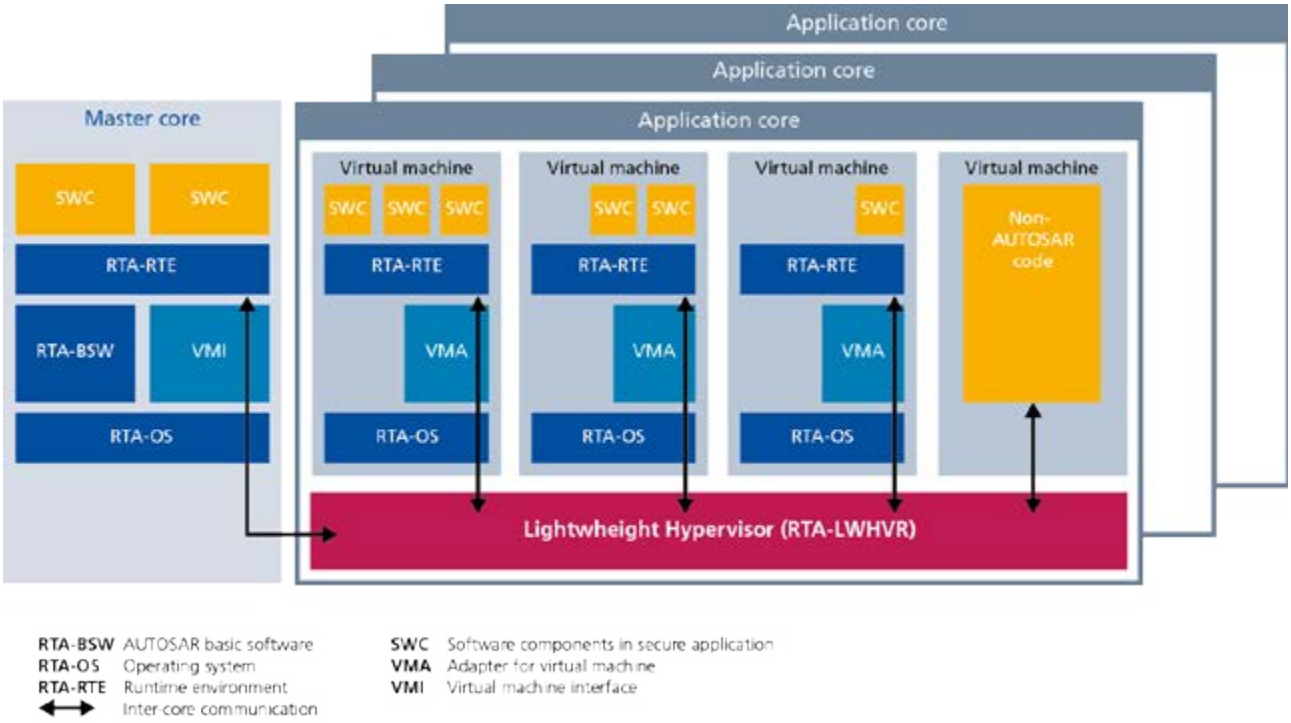
Agile software development enables us to keep software-controlled vehicle functions up to date via a series of updates and upgrades – also in the field. To be able to install these updates without compromising security, we need a strict separation of individual functions – which goes against the trend towards increasing connectivity of ever more functions in central control units. One way to resolve this conflict is through partitioning – which used to be technically impossible with the microcontrollers in automotive ECUs. Now, ETAS' lightweight hypervisor is able to solve the problem.

This kind of separation is not only desirable under the aspect of liability and functional safety. It also simplifies the workflows in software development, which often takes place in teams all over the world. There are also advantages from a security standpoint: when individual functions on a control unit are clearly separated, it increases protection against attacks from cyber criminals. Even if hackers were able to gain access to a function, it would be difficult for them to access other vehicle control functions. This would discourage criminals because their aim is to cause as much damage as possible to car manufacturers.

Separation makes sense – but how?

Architecture based on AUTOSAR 4.x already defines fundamental mechanisms to allow for the various software modules on a control unit to be developed independently from one another and it also offers basic elements to separate them from one another (e.g. memory protection). However, in order to realize the concepts envisaged by AUTOSAR, a hypervisor is needed which partitions

Fig. 1: Schematic diagram of a control unit with the Lightweight Hypervisor ETAS RTA-LWHVR.



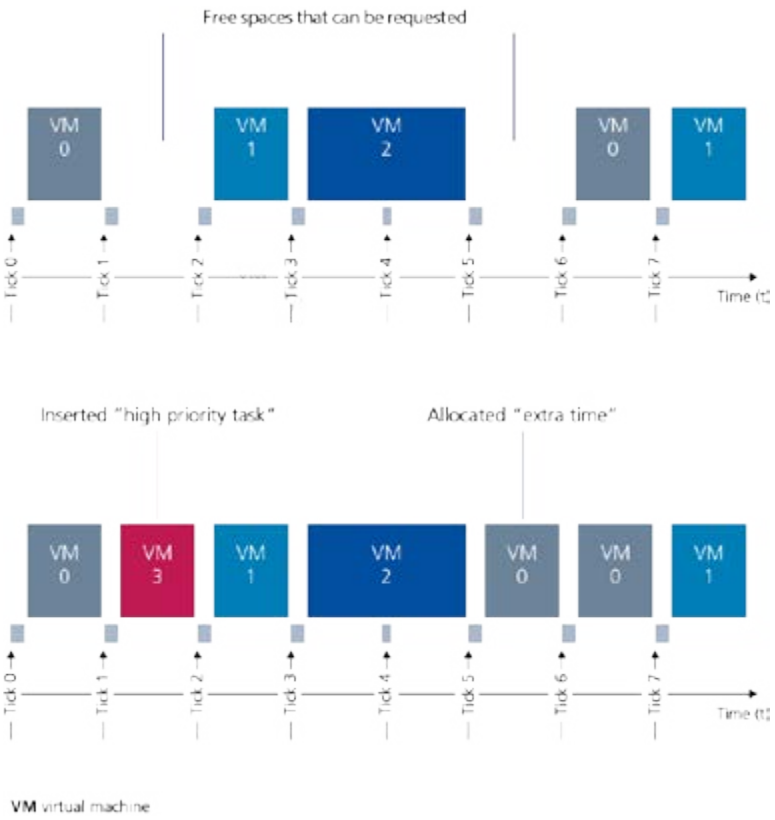


Fig. 2: Time management in an application core. Hypervisor slots additional requested runtime in the free spaces.

the individual control unit into different virtual machines (VM). This solution ensures that updates and upgrades of individual functions can be carried out without negatively influencing the other functions on the control unit. This means that modification of individual functions would therefore not have to be preceded by an extensive re-validation of all software on the control unit.

Partitioning enables the respective software to operate as if each function was running on its own hardware. The functions are so clearly separated that they can be individually modified when updates and upgrades take place without having to carry out a complete revalidation. And not only that. Early on in the development stage of the control unit, the software companies involved can work entirely independently from one another thanks to partitioning. Software errors or malicious hacking attempts are contained locally on the individual VMs. The degree of separation is so high that software with different automotive safety integrity levels (ASIL) from the lowest level "QM" up to the highest " ASIL D", can be operated on one and the same control unit – without compromising the safety and security of the overall system.

The Hypervisor solution – implementation is all-important

In principle, the Hypervisor approach creates the necessary conditions for agile software development for control units. But implementation is key here. Adaptation to the specific requirements of the automotive control unit in question is absolutely necessary. Normally, Hypervisor needs its own memory and access rights management. The classic Hypervisor Privilege Mode has three levels: the Hypervisor itself, basic software and the application functions. These requirements alone are pushing the limits in the automotive sector. Neither memory management nor the three-level Hypervisor priority mode of today's vehicle microcontrollers are fully supported. This problem needs to be solved before Hypervisor technology can be rolled out for widespread use in the automotive sector.

To this end, ETAS has developed the Lightweight Hypervisor (RTA-LWHVR) for one of Germany's large car manufacturers. Thanks to new architecture and more flexible communication structures, this Hypervisor for deployment in vehicles requires only five kilobytes (kB) of memory and access times have been shortened by a factor of four to five. In this specific project a central body control unit was partitioned into eleven virtual machines, each reserved for the software of a particular supplier. The ASIL classification ranged from QM to B.

Lightweight Hypervisor overcomes the limits of AUTOSAR

Despite the sheer amount and diversity of software functions, in this project it was possible to operate them in an encapsulated way without encountering any problems. This was due among other things to the fact that the computer cores in the Lightweight Hypervisor are divided up into a master core and various application cores. The separation ensures that the master core assumes the hardware management, the operation of the centralized basic software, and a handful of software applications, while the application cores host only the separate virtual machines (Fig. 1). For each individual control unit, there is also the option of choosing between distributed runtime environments according to AUTOSAR with an AUTOSAR runtime environment (RTE) or non-AUTOSAR-compliant software.

A further important success factor: the virtual machines have access to a shared memory, and there are clear rules governing this. The execution times on the core are also clearly defined. To enable

fast reactions, functions can request further time budgets in addition to their guaranteed execution time without compromising the execution of other functions (Fig. 2). The real-time requirements are therefore always guaranteed.

This flexible access allows the virtual machines, when they require a great deal more runtime, to query the Hypervisor to find out whether additional runtime capacities beyond the capacity allotted can temporarily be accessed. The Lightweight Hypervisor prioritizes these requirements and ensures that no other functions are compromised during the temporary access to the runtime buffer.

Market-ready, future-proof solution for agile software development

Because we have succeeded in reducing the memory requirements of the overhead to 5 kB and performance requirements to only five percent of the available core capacity, the Lightweight Hypervisor, ETAS RTA-LWHVR, is easily integrated into the technical framework conditions for embedded vehicle systems. Which means that it has succeeded where the classic hypervisor approach in vehicles has so far failed. Due to its low demand on resources, it is suitable for flexible use in control units of a diverse array of vehicle domains. And it is also already available for many of the microcontrollers on the market.

The Lightweight Hypervisor solution enables reliable and effective partitioning of production control units, paving the way for the safe and secure updating and upgrading of vehicle functions. Providers of control unit software can work just as freely and independently at the development phase as they can after production has started and they are developing software-controlled functions which are installed at the service shop or over-the-air. Moreover, by encapsulating functions, the level of safety is also increased even though software of different safety classes is operating on the same partition – regardless of whether it conforms to AUTOSAR or not.

This flexibility will not only simplify workflows in software developments, it will also lead to a marked reduction in the need for testing, especially where agile software and function development are concerned. On account of the intelligent inter-core communication and encapsulation, functions can be modified without having to fully revalidate the entire system. This fulfils an essential requirement for safe and secure function updates and upgrades in vehicles.

Customers will benefit from this in two ways. In the future, when buying a vehicle, they can start off with a lower-priced basic version which they gradually customize, adding new functions by means of upgrades. And when the vehicle is in operation, the reliable partitioning increases safety because manufacturers are able to carry out safety updates tailored to vehicle needs at any time – without worrying about compromising other functions. In the battle against cybercrime and its constantly changing attack strategies, it is also a major advantage to be able to quickly implement updates.

Authors
Dr. Alexander Leonhardi, senior manager for RTA Solutions and Consulting at ETAS in Stuttgart
Dr. Gary Morgan, senior consultant at ETAS in York (GB)
James Dickie, product manager for RTA products at ETAS in York (GB)