Introduction
In the past, automotive audio systems were characterized by simple functionality, a simple tuner, minimal user interface, and primitive two-way communication channels. They were also closed, in the sense that all of the software was loaded pre-sale by the manufacturer, and normally remained unchanged for the lifetime of the device. The amount of software was small.

Most in-vehicle infotainment (IVI) systems today have sophisticated user interfaces, consisting of input keys, possibly a touch screen, rear-view cameras, audio and high-resolution video outputs. They combine many communication, productivity and media functions. They also support different wireless communication modes, including: Wi-Fi, Bluetooth® and infrared and they allow users to load data and access application programs (apps). The total amount of software running on today’s devices measures into the millions of lines of code.

Need for Virtualization in Automotive Domain
The gap between consumer electronics and automotive electronics started growing wider in the early 2010s due to the rapid consumer adoption of smartphones. Consumers expect to do everything in their car that they would on mobile devices. The automotive industry has been conservative during the past, but now it must try to bridge the gap between consumers' expectations for infotainment features and the current performance of today’s automotive electronics.

The key problems faced by automakers with respect to IVI electronics are Firmware updates and in-vehicle security. To address these problems, virtualization is emerging as a key solution.

Virtualization
The concept of virtualization originated in mainframes and is used extensively in the personal computer domain. It has recently picked up in the embedded electronics world.

There are three different types of virtualization techniques:

1. Virtual Machines (VMs): Hypervisors run on top of the host operating system (e.g. VMware products).

<table>
<thead>
<tr>
<th>guest application</th>
<th>guest application</th>
<th>guest application</th>
</tr>
</thead>
<tbody>
<tr>
<td>guest operating system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>virtual-machine monitor (VMM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>host operating system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>host hardware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Para-Virtualization/Bare-Metal Virtualization: Hypervisors run directly on the host’s hardware, controlling the hardware and managing guest operating systems (e.g.
COQOS from OpenSynergy). This is most widely used on automotive embedded platforms.

3. Containers Virtualization: The kernel of an operating system allows for multiple isolated user-space instances, not just one.

---

**Virtualization Benefits in the Automotive Domain**

**Firmware Update**

IVI architecture can be separated into critical and non-critical partitions. The critical partition concerns vehicle network communication and other software that requires a Real Time Operating System (RTOS) or thin Operating System (OS). User interface-related features are placed in the non-critical partition, which runs on a high-end, consumer-based OS, like Linux or Android. This architecture benefits vehicle manufacturers as it makes it possible for the firmware of one partition (e.g. non-critical) to be updated without affecting the other.

**In-Vehicle Security**

Virtualization can significantly enhance security. A virtual machine encapsulates a subsystem, so that its failure cannot interfere with other subsystems. In an IVI, the in-vehicle Controller Area Network (CAN) communication stack is of critical importance. If the CAN communication stack were subverted by an attacker, the IVI could interfere with other critical CAN nodes in the vehicle network. Similarly, an encryption subsystem needs to be strongly shielded from potential threats so that sensitive user and vehicle information can be protected from hackers. This is a significant challenge for a system running millions of security-critical lines of code. The high-level OS is particularly vulnerable to attacks in open systems, and is large enough to contain thousands of bugs. With virtualization, the high-level OS is unable to interfere with data belonging to other subsystems and access to the processor is limited significantly.
Limits of Virtualization
Virtualization by its very nature is intricate. It therefore presents, like most technology-based platforms, its own potential problems and limitations.

Software complexity
The isolation provided by virtualization creates a complete machine for each subsystem. This means that each virtual machine has to run its own operating system, making it relatively “heavy” when it comes to software code. Increasing the number of virtual machines could create serious performance issues - significantly increasing the amount of code, increasing memory size and introducing more potential points of failure.

Integration
The subsystems of an embedded system are not independent. They need to be designed to cooperate closely in order to achieve the intended functionality of the system.

Security policies
Many embedded systems must meet critical security requirements. Virtualization alone does not necessarily help address all security requirements.

Trusted computing base
Virtualization does not support minimizing the Trusted Computing Base (TCB). These are the essential components to the system’s security (hardware, software, and firmware). The TCB frequently has high performance requirements, meaning that it must be able to communicate efficiently with the rest of the system - which Virtualization does not serve.

Conclusion
Para-virtualization is generally seen as the best option in the embedded domain. A thin, para-virtualized hypervisor can continue to serve as the foundation for embedded IVI systems, performing resource management (memory, devices, energy, and global scheduling) and facilitating secure communication and resource sharing among guest operating systems.

On the other hand, some cores also come with hypervisor mode, which enables hardware-assisted virtualization. Focusing on this strategy, para-virtualization suppliers are also extending their hypervisor mode to accommodate these processor capabilities, including hardware-assisted virtualization and 40-bit addressing. Hardware-assisted virtualization reduces the maintenance overhead of para-virtualization, reducing the changes needed in the guest operating system. These types of virtualization architectures are poised to provide the best options for designing world-class software for tomorrow’s connected vehicles.

Reference

Author:
Swaminathan G, Visteon Electronics, India
Saminathan V, Visteon Electronics Innovation Head, India.
Author:
Swaminathan G, Visteon Electronics, India
Saminathan V, Visteon Electronics Innovation Head, India.