ACRN Case Study for Software Defined Cockpit

Whitepaper

October 2019

Table of Contents

1.0 Terms and Abbreviations

Term	Definition
IoT	Internet of Things
IVI	In Vehicle Infotainment
IC	Instrument Cluster
OEM	Original Equipment Manufacturer
ECU	Electronic Control Unit
RSE	Rear Seat Entertainment
ADAS	Advanced Driver Assistance System
SDC	Software Defined Cockpit
HMI	Human Machine Interface
HUD	Head Unit Display
SoC	System On Chip

2.0 ACRN hypervisor

2.1 Challenges for IoT

IoT developers face mounting demands as connected devices are increasingly expected to support a range of hardware resources, operating systems, and software tools/applications. As we've seen in other market segments, virtualization is key to meeting these broad needs.

However, existing solutions don't offer the right size and flexibility for IoT. Data center hypervisor code is too big, doesn't offer safety-critical capabilities, and requires too much overhead for embedded development. Proprietary solutions are typically expensive and make it difficult to deliver long-term product support.

Meet Project ACRN, an open-source hypervisor catering to the specific requirements of the embedded, IoT and Edge segments.

2.2 ACRN advantages

ACRN is *intentionally* different from anything currently in market. It provides a flexible, lightweight hypervisor, built with real-time and safety-criticality in mind, optimized to streamline embedded development through an open source, scalable reference platform.

The biggest news about ACRN's hypervisor is that it's **slim**. With around 30,000 lines of code, this solution offers a lightweight code base designed for resource constrained devices. Equally important, it targets complex embedded systems requiring various levels of safety-criticality.

Rich virtualization of I/O mediators allows resource sharing to maximize the potential for IoT devices. ACRN is also built to address the need for real-time responses, with low latency and high adaptability. Together, these features encourage SoC consolidation to help reduce both development and materials cost. ACRN intends to enhance safety considerations by isolating safety-critical workloads from other tasks.

Intel's experience and leadership in virtualization technology was key to ACRN development. The project, under the auspices of the Linux Foundation, is fully open source under a **flexible license** (BSD-3). The code is publicly available on GitHub at https://github.com/projectACRN/acrn-hypervisor. The development is done in the open via the project mailing lists. We believe this approach fosters collaboration among Key players in embedded hypervisors industry, accelerating feature development and speeding the maturity of this important technology. Open source has the added benefits of helping ensure code transparency and compatibility and providing a high-quality reference stack. These in turn can allow faster development and time to market, in addition to enhancing long-term support and maintenance cost savings.

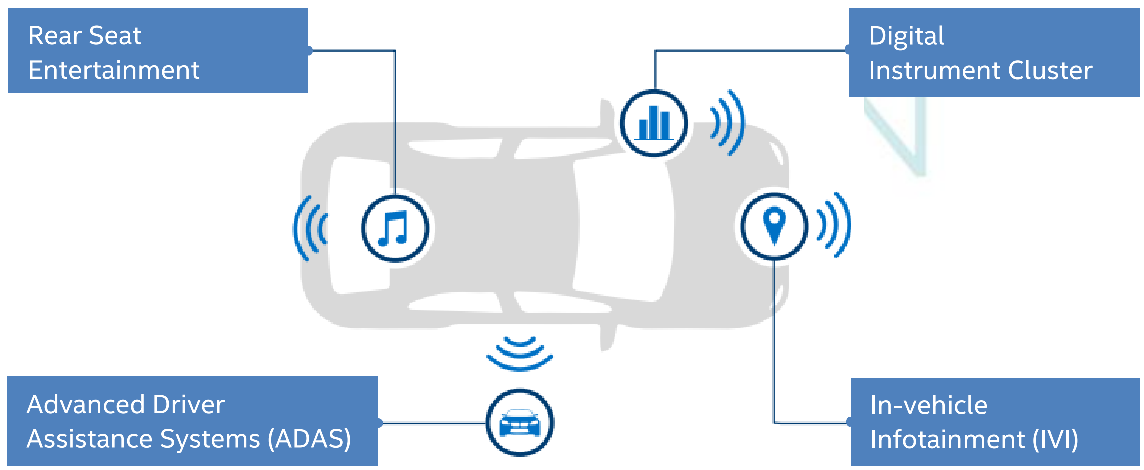
ACRN is an open source, slim hypervisor that supports resource sharing. However, it also caters to more demanding segments requiring strict system isolation and real-time behaviors. ACRN is ideal for a broad range of IoT uses, such as automotive SDC, industrial, and retail.

Refer to ACRN's website to know more about "ACRN: A Big Little Hypervisor for IoT Development": https://projectacrn.org/

3.0 Software Defined Cockpit (SDC)

Automotive functions like Infotainment, Instrument Cluster and ADAS form the "Software Defined Cockpit" (SDC). The SDC provides a context for understanding how to deliver technologies that help the OEM deliver more/bigger/better car experiences.

Technology advances are helping the automotive market to evolve into a digital era that could have only been imagined just a few years ago. At the same time, technology is helping to drive down the cost of safety, improving efficiencies and integrating the driving experience into the connected world around it.



- **Digital Instrument cluster (IC)**: presents the vital driving information for the driver, like speed, fuel level, trip miles, etc. It also optionally includes a Heads-Up Display (HUD) component that projects information on the windshield, with alerts for low fuel or tire pressure for example.

- **In-Vehicle Infotainment (IVI):** offers functions like navigation systems, radios and Entertainment systems. It can expand to allow connection to mobile devices for calls, music and applications via voice recognition. On most recent cars, IVI includes the display of back-up / Surround-view cameras, the parking assist function and gesture recognition / Touch (HMI).

- **Rear Seat Entertainment (RSE):** offers the luxury of entertainment system for back seaters. It also allows further functionalities like virtual office access or connecting to IVI front system and mobile devices (cloud connectivity).

- **Advanced Driver Assistance Systems (ADAS)**: offer driving assistance functionalities ranging from blind spot monitoring, adaptive cruise control, lane departure warning to much more evolved aids like brake assistance, collision avoidance, self-parking systems and driver Monitoring.

The traditional setup for In Vehicle cockpit system has one physical Electronic Control Unit (ECU) per feature. It is not uncommon to find as many as 100+ ECUs in a car, which ultimately increases the complexity, cost and maintainability.

4.0 ACRN and SDC consolidation

4.1 SDC Consolidation: The Concept

The number of ECU's in the car has doubled in the last 10 years. Each ECU has its own infrastructure of power supply, connectivity, software and safety requirements. With virtualization technologies, most of these features can be consolidated into one system. Only demanding applications, like automated driving, need their own ECU.

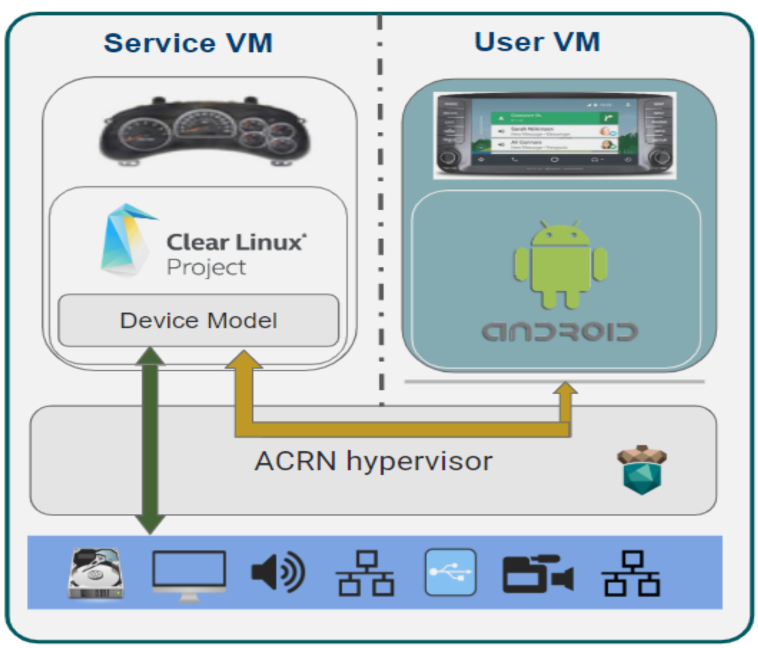
Each of these features have different requirements for safety and openness. Infotainment and RSE are less critical but also benefit the most from supporting third-party applications, possibly via an application store. The Instrument Cluster (IC), on the other hand, must not crash, and some essential information must be displayed. Safe driving depends on driving information presented by the Instrument Cluster.

Virtualization provides the best solution today to accommodate these diverse systems in one system (ECU) while reducing costs and platform complexity. It provides isolation at the Operating System level. The Instrument Cluster can run its own Safety Critical operating system. The Infotainment and Rear Seat Entertainment can run in a less restricting OS like Linux or Android. Each System being completely isolated and independent from each other while still using one single SoC.

4.2 SDC Consolidation with ACRN

The main ingredient for a virtualized system is the hypervisor, which plays the role of moderator between the different virtual machines running on the same hardware platform (ECU). With its small footprint, real-time behavior and OS-level isolation built-in features, the ACRN hypervisor drives the transformation of automotive experience.

Below is a diagram representing the concept of a consolidated SDC system, featuring Intel® Apollo Lake, running ACRN hypervisor with two distinct virtual machines: one for the Instrument Cluster (Service VM) and one for the Infotainment (User VM):



The electronic Instrument Cluster requires the highest level of safety, security, and real-time performance. It must also guarantee a display refresh rate of 60 frames per second. The car's infotainment system supports the latest Android 9.0, and integrates a variety of cool, smart cockpit applications, including enhanced navigation, map navigation, voice control, and AI-assisted driving, which becomes some key differentiators.

The Instrument Cluster runs in Clear Linux*, while the infotainment system uses the latest Android* 9.0. In the past, such a scenario required the use of at least two distinct SoC chips. But now, a single powerful Intel(R) Apollo Lake processor can easily handle this use case by including a hypervisor to support two or more operating systems.

4.3 ACRN features

Safety and Security Isolation

By leveraging Intel silicon's Virtualization Technology (VT), ACRN can provide secure isolation of each Guest OS. In the event that one of Guest OS is compromised, it won't impact other Guest VMs and their workload functionality.

Extensive Sharing Capabilities

ACRN provides rich I/O mediation for device sharing, including storage, network, USB, audio, camera, GPIO, GPU sharing, etc.

Multiple OS Support

Not only can ACRN support multiple Guest VM instances but ACRN can also support different types of Guest OS. For example, Clear Linux*, Android, Automotive Grade Linux* (AGL), Yocto*, Ubuntu, etc.

Fast Boot

Used with the optimized Slim Boot Loader (SBL) firmware (see https://slimbootloader.github.io), Service VM kernel and ACRN embedded hypervisor, the Instrument Cluster can be launched within 2 seconds from power on to the screen being displayed.

A Tutorial on how to use SBL on UP2 boards with ACRN is available on the ACRN project home page: https://projectacrn.github.io/latest/tutorials/using_sbl_on_up2.html

MISRA-C Compliance

MISRA-C is a coding guideline recommended by ISO 26262. ACRN hypervisor uses a coding guideline derived from MISRA-C:2012, as well as coding style rules as required by the standard. The MISRA-C compliance is enforced by code review and static checkers in ACRN development process.

Add-on differentiations

A lot of interesting features can be developed in the context of software defined cockpit and virtualization technology. For example, projecting road navigation map from IVI to IC. This is called "display surface sharing via HyperDMA" in ACRN. Surface sharing is one typical automotive use case which requires that the Service VM accesses an individual surface or a set of surfaces from the User VM without having to access the entire frame buffer of the User VM. It leverages hyper_DMABUF, a Linux kernel driver running on multiple VMs and expands DMA-BUFFER sharing capability to inter-VM.

5.0 ACRN in automotive: a success story

In September of 2019 a Chinese automobile manufacturer launched its SUV model based on the ACRN hypervisor.

It features:
- An eye-catching dual 12.3-inch floating LCD screen, a newly designed electronic Instrument Cluster,
- A facial recognition system that automatically adjusts the interior settings customized for the driver,
- An AR-enhanced real-life navigation system that gets you where you need to go.

It also supports online payment, intelligent interactive voice recognition capability, and many other amenities. These applications are made possible by adopting Intel's Apollo Lake processor and ACRN™ virtualization solution.

5.1 Intel Apollo Lake SoC - Two birds one stone

The electronic Instrument Cluster used by this SUV model requires the highest level of safety, security, and real-time performance. It must also guarantee a display refresh rate of 60 frames per second.

The car's infotainment system supports a variety of cool, smart cockpit applications, including Baidu AR enhanced navigation, map navigation, voice control, and AI-assisted driving.

The Instrument Cluster System runs a Clear Linux* OS, while the infotainment system runs the latest Android* 9.0.

In the past, such a scenario would have required the use of at least two distinct SoCs. But by using ACRN, only one, powerful Intel Apollo Lake processor can easily handle this use case by supporting two or more Operating Systems.

5.2 ACRN Hypervisor - Hero behind the scene

Once the hardware virtualization function of Intel Apollo Lake processor is enabled, the ACRN hypervisor can support multiple operating systems simultaneously while ensuring the safety isolation of each operating system without affecting the other.

ACRN builds a virtual layer on top of the underlying hardware processor, achieving multi-OS workload consolidation to ensure isolation and non-interference between the electronic Instrument Cluster and the Infotainment system. ACRN supports fast start-up and enables rapid display of IC dashboards on start-up. ACRN also supports the underlying hardware resource sharing, where a single, physical graphics card can simultaneously display the User Interface (UI) of the Instrument Cluster dashboard and the IVI system. It also supports shared storage devices, shared network interface controllers, and shared sound cards.

5.3 Android 9.0 - Tailored for Automotive

Android 9.0 is the first automotive version of the Android Operating System released by Google. The operation interface of Android 9.0 is more in line with automotive usage scenarios.

A series of designs ensures safety from hardware, through the kernel, to the application framework. The improved automotive Hardware Abstraction Layer (HAL) and the Car Service allow users to control the car's air conditioning, lighting, volume and other settings more easily and in more friendly way. These features make it easy for developers to use their imaginations to build more applications.