Functional Safety Architecture for linux-based systems

Tadao Tanikawa, Automotive and Industry Company, Panasonic Corporation

Panasonic

Historical view on Functional Safety

We have achieved reliable systems by separating hardware modules.

And we have employed reliable software modules (including OS).

Fortunately they have been relatively small, such as RTOS and some simple applications.

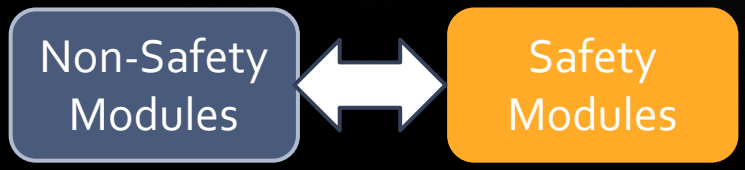
In our next step, we are considering to use general purpose OS (such as Linux) and more complex applications in which use OSS libraries or frameworks.

Currently we have two technical directions.

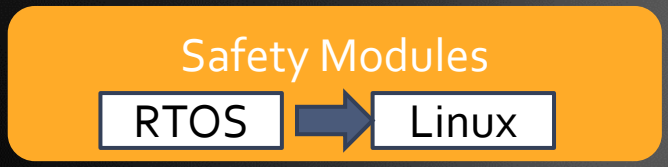
- Separating software modules into two parts:

  safety modules and non-safety modules

- Begin to ensure that the general purpose OS can support functional safety.



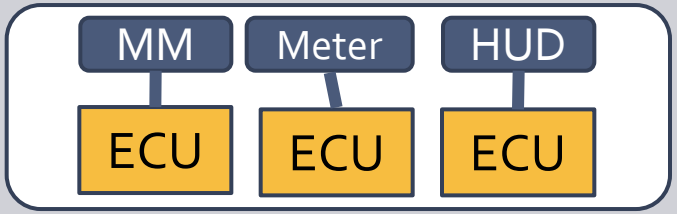
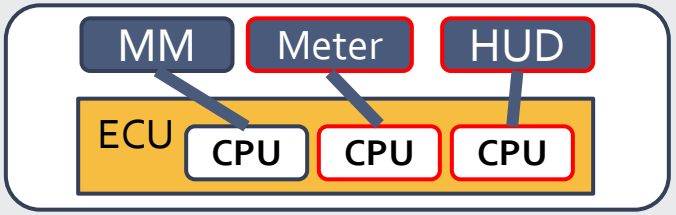
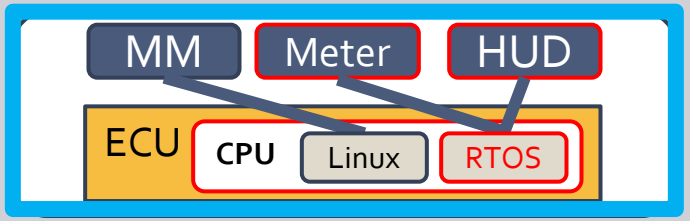
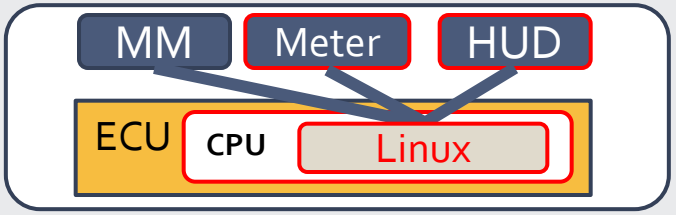
Separation



Moving to High Level

Panasonic

There are four types of partitioning

partitioning	Types	Functional Integrity	ASIL	Architecture
Hardware	**A-1** Separated Units	Low	High	
	A-2 Dedicated Multi-Core	Low	Med.	
Software	**B-1** Hypervisor	High	Low	
	B-2 Linux	Very High	Low	

Updated from the keynote of Mizuyama (Panasonic) in ALS 2014

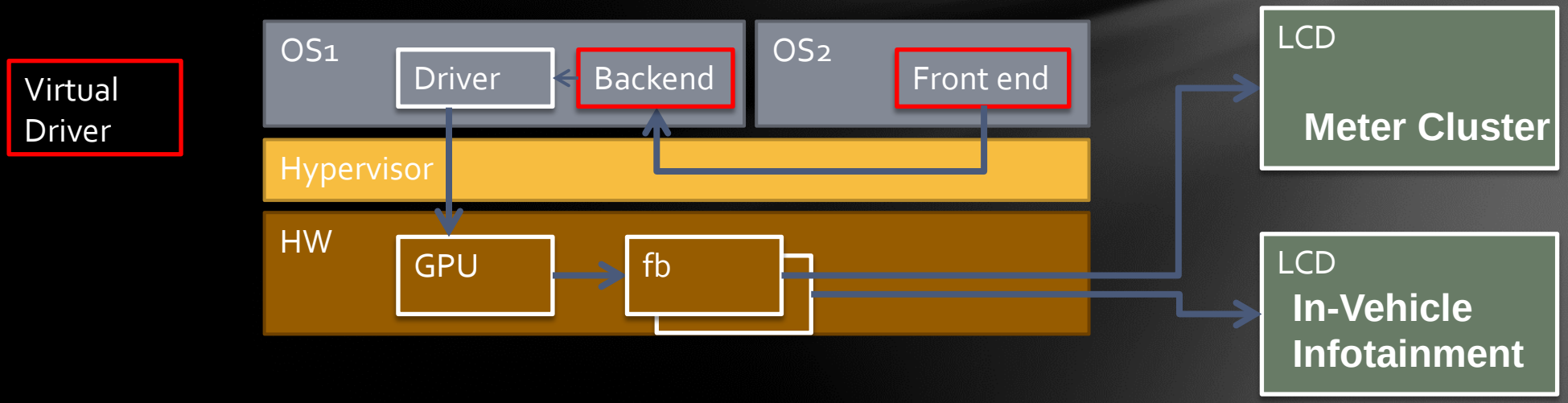
Issue of Partitioning by Virtualization

- **Performance and dependability**
- **Guarantee of real-time performance**
- **Updating software cost**
- **Obtaining certification of device drivers for peripherals.**

Panasonic

Issue of Partitioning by Virtualization

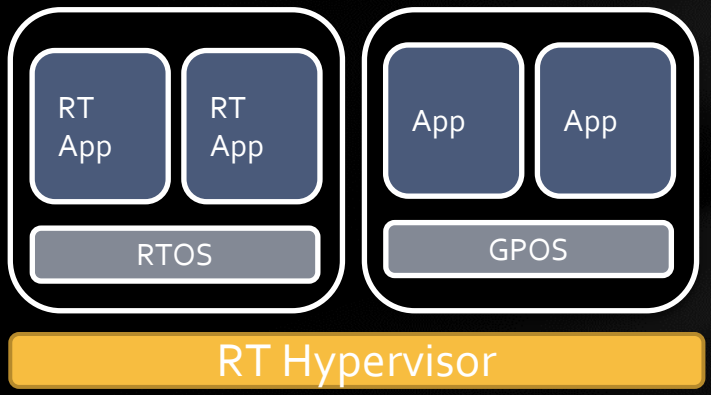
- **Performance and Dependability**
  - **To keep consistency of performance and dependability, sharing devices is important.**
    - Software partitioning needs to share devices among guest OSs.
  - We think software partitioning needs to share GPU, because there are some use cases where the guest OSs output their screens to each display, or to one composite display.



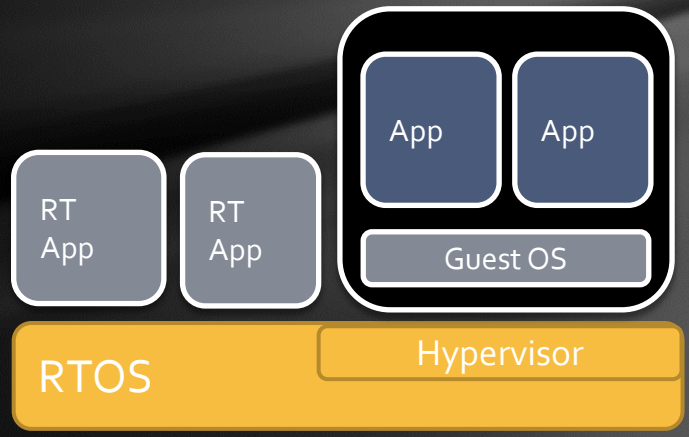
Panasonic

Issue of Partitioning by Virtualization

- **Real-Time**
  - **Guarantee partitioning software's real-time performance**
    - If a sub-system (guest OS on virtual machine) has real-time requirements, partitioning software needs to ensure the same requirements.



Type 1 Hypervisor



Type 2 Hypervisor

Panasonic

Issue of Partitioning by Virtualization

Software Updating Cost

- For example, Android Upgrade (e.g. JB to KK) has issues on virtualization.
  - Porting and verification costs much because Android is frequently updated.
  - Especially when Android upgrade has kernel changes, virtualization of the kernel is required again.
  - When it becomes necessary to modify a hypervisor, functional safety certification is needed again.

Issue of Partitioning by Virtualization

Obtaining certification of device drivers for peripherals

If supporting virtualization on several chipsets, it is important how many virtual drivers are prepared.

- Procurement of virtual drivers is an issue, because embedded systems usually use one from many type of SoCs.
- Those peripheral devices are varied, and their specifications are also different from each other.
  - Usually you have to write your own driver or modify an existing one, then you need to obtain certification of them by yourself, even if using a certificated OS.

Conclusion

We have two technical directions.

- Separating software modules into two parts

  We examined integrating general purpose OS and safety functions by software partitioning.

  Especially, when **sharing devices,** need to ensure

  - **keeping consistency of performance and dependability**
  - **being virtualized and obtaining functional safety certification**.

  We would like to discuss more about this topic.

- Begin to ensure that the general purpose OS can support functional safety

  We would like to support the challenge of Linux functional safety.

We look forward to collaboration with you.

Thank you !!

Panasonic