Automotive Containerization Architecture for the Linux Based Instrument Cluster

Naoto Yamaguchi
Doctor of Informatics, Researcher
Software Fundamental Technology Group
Application Design Department
Connected & Sharing Solutions Division
AISIN CORPORATION

Introduction to Who I Am

- Name:
  - Naoto Yamaguchi
- Company:
  - AISIN CORPORATION
- Career
  - Received Doctor of Informatics in 2007 (Shizuoka-University).
  - Automotive RTOS platform software engineer since 2007.
  - Automotive Linux platform software engineer since 2011.
- My history of Open Source Community
  - Member of AGL Instrument Cluster Expert Group since 2019.
  - Joined to ELISA in 2019.

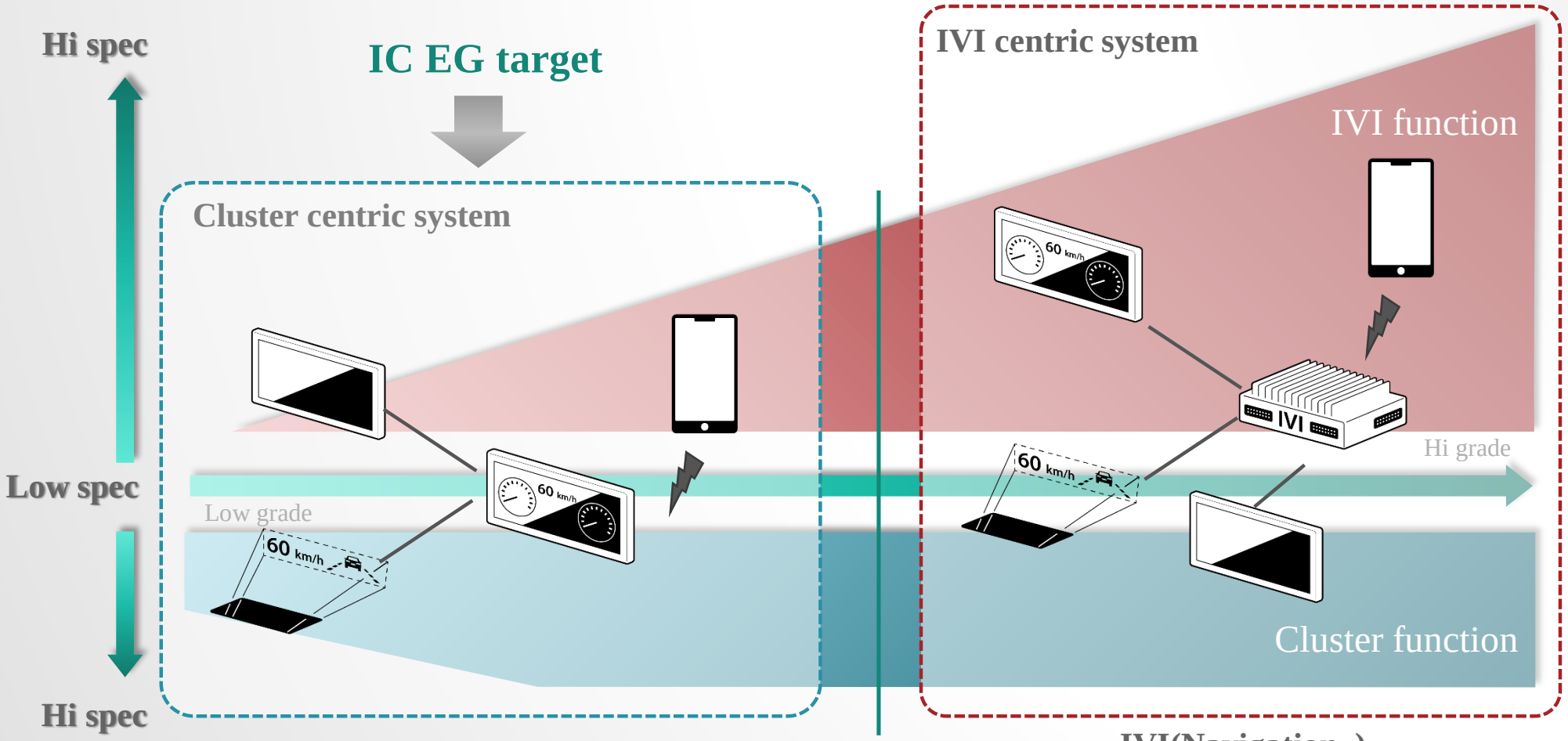
Outline

- Background
- QM Isolation Architecture
- AGL Instrument Cluster
- Conclusion

Outline

- **Background**
- QM Isolation Architecture
- AGL Instrument Cluster
- Conclusion

EG scope and system image?



Cluster(HUD)
+
Smart phone connection(SDL..)
for small IVI function

IVI(Navigation..)
+
Cluster(HUD)
+
Smart phone connection(SDL..)

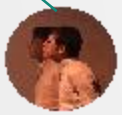
AUTOMOTIVE GRADE LINUX

Motivation

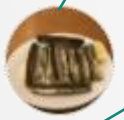
- Create a base platform for Instrument Cluster, not a platform based on conventional IVI.
  - There are different system requirements between IVI and Cluster.
    - e.g.)Functional safety, boot time etc…
  - This platform is build from open source technology.
    - Need to independent from existing proprietary software component.
    - Need to be able to choice open source software component or proprietary software component.

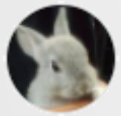
Member

Toshihisa Haraki:
EG leader

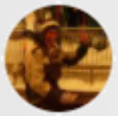


Naoto Yamaguchi:
Architecture design, Integration and OSS assessment.

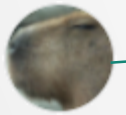


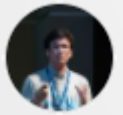


Hiroyuki Ishii:
Architecture design, Integration and Ref-hw support.



Kenji Hosokawa:
Graphics expert.





Harunobu Kurokawa:
Linux kernel and BSP expert.

Naohiro Nishiguchi:
Audio expert.

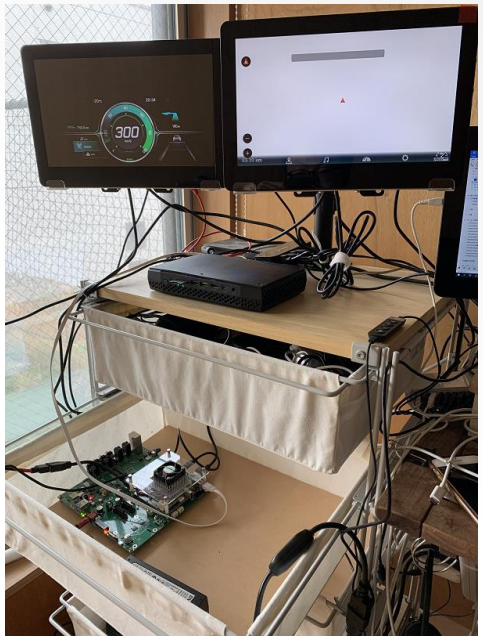
Jan-Simon Moeller:
Release management and tools.

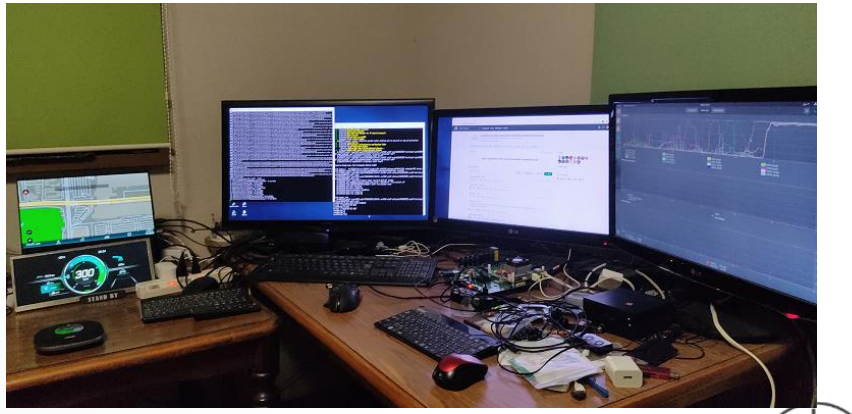
Keiya Nobuta:
Com. Stack expert

Scott Murray:
Yocto expert.

And more members.

Dev. Lab.





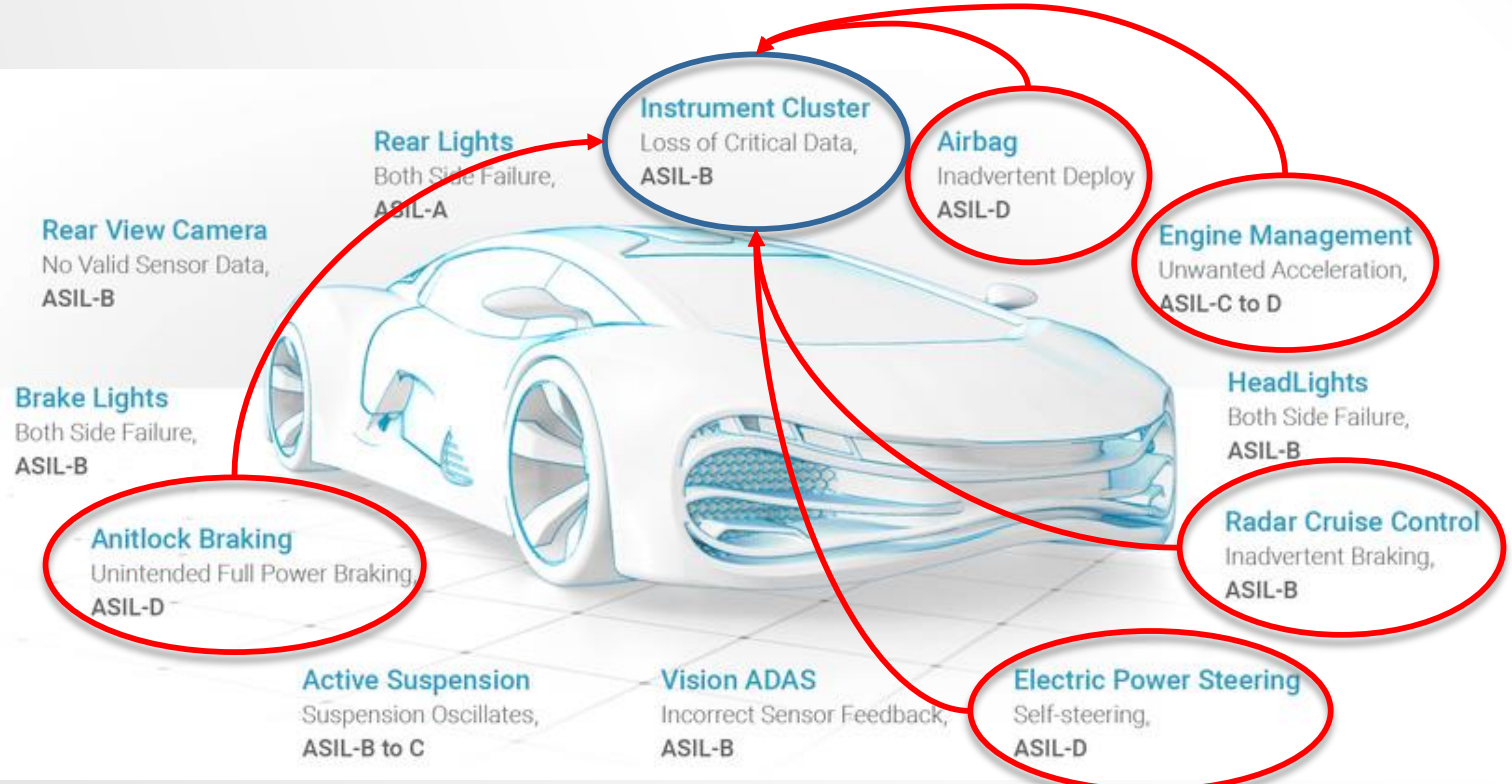
AUTOMOTIVE GRADE LINUX

What is the product development issues?

- Quality and Robustness
  - Functional safety (ASIL-B)
    - Instrument cluster has telltale function.
      - It show critical frailer information to driver.
  - Quality management
    - There have separate quality requirement between Instrument Cluster and IVI.

Why ASIL-B is required Instrument Cluster.

- Typically instrument cluster assigned ASIL-B.
  - Includes telltale function that is assigned ASIL-B.
    - ASIL-B was decomposed from other units.
  - Existing instrument cluster does not have ASIL from own functions.



Ref. https://www.synopsys.com/automotive/what-is-asil.html

Our approach

Main function is the very function of our system

- Requires advanced quality management.
- Requires open innovation.
- Requires cyber security.
- Requires fast boot.
- Requires various functions.
- …

Main target of IC-EG

Safety function ensures vehicle safety

- What function does it include?
- Which OS do you use?
- Which communication method do you use?

Collaborate ELISA to find a solution.

Main function

Safety function

Isolation method

Functional safety will be discussed in the ELISA Project.

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Main function and safety function are isolated by isolation method.

- Hardware separation? Using hypervisor?

Collaborate ELISA to find a solution.

AUTOMOTIVE GRADE LINUX

Puzzles in automotive quality management

- There are many puzzles in the automotive system (main function).

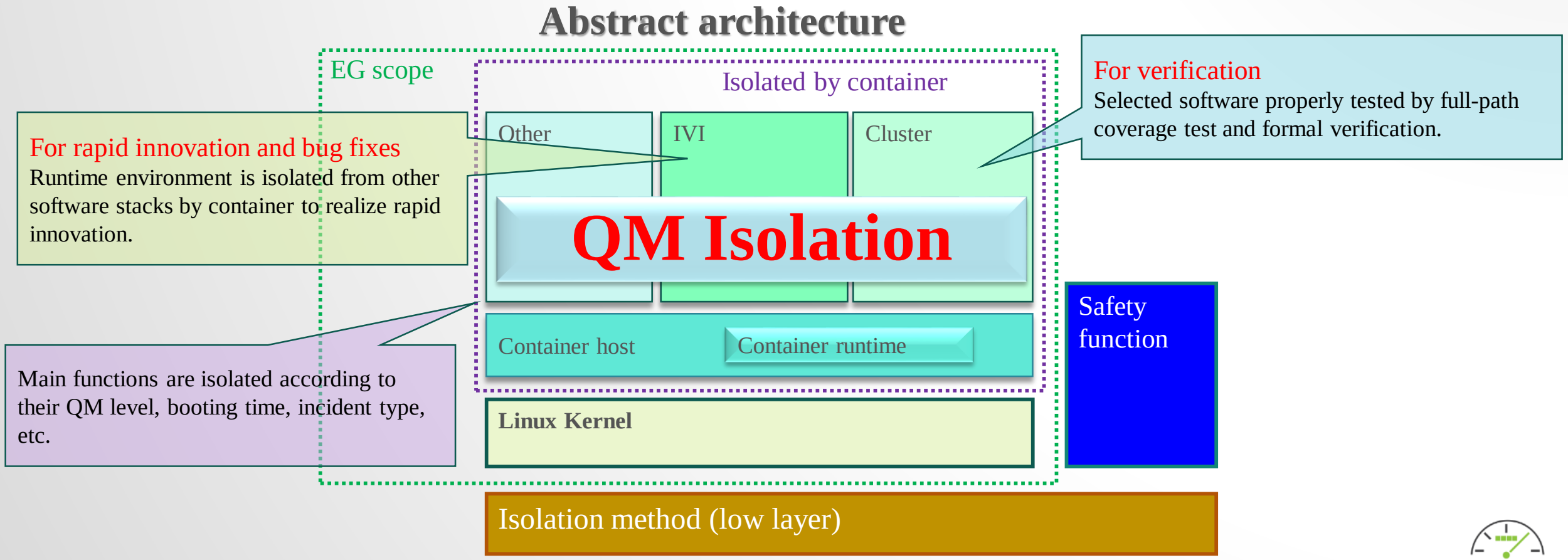
IVI	Instrument Cluster

• Rapid innovation
  • New features are added
  • Short-term development
  • Rapid bug fixes
	Puzzle
	• Advanced quality management
	  • Full path coverage testing
	  • Formal verification
	  • Careful bug fixes

• Various functions
  • Many pre-installed applications
  • Applications installed from store
	Puzzle
	• Selected functions
	  • Combinational verification
	  • Fast boot-up

Our approach

- Our answer to the puzzle issues is “one more isolation method" which takes one-more layer to isolate the functions by using Linux container technology.



Outline

- Background
- **QM Isolation Architecture**
- AGL Instrument Cluster
- Conclusion

What is Linux Container

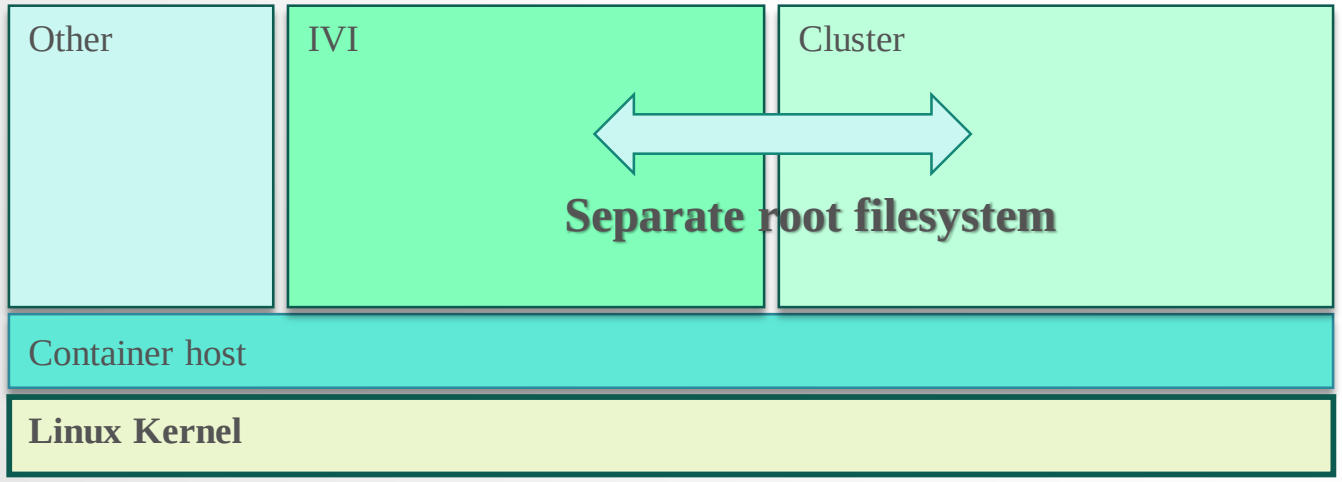
- Operating system level virtualization method.
  - Running multiple isolated Linux systems (containers) on a host using a single Linux kernel.
    - Isolate root filesystems on Linux kernel by using chroot.
    - Control resource (such as cpu, memory) by using cgroups.
    - Hide resources from other containers by using namespace.
  - Easy to use in embedded environment.
    - Only require to Linux BSP.
    - No need additional virtualization driver development.
    - Can integrate container based system over hypervisor.

QM Isolation

- Software stack isolation
  - We think so should separate root filesystem between Instrument Cluster and IVI.

  - **Instrument Cluster:**
    - Build by highly quality assurance software.
    - Will not change after SOP without critical bug fix.
  - **IVI:**
    - Build by standard software. Will change after SOP to upgrade function.

  - Linux container realize root filesystem isolation.

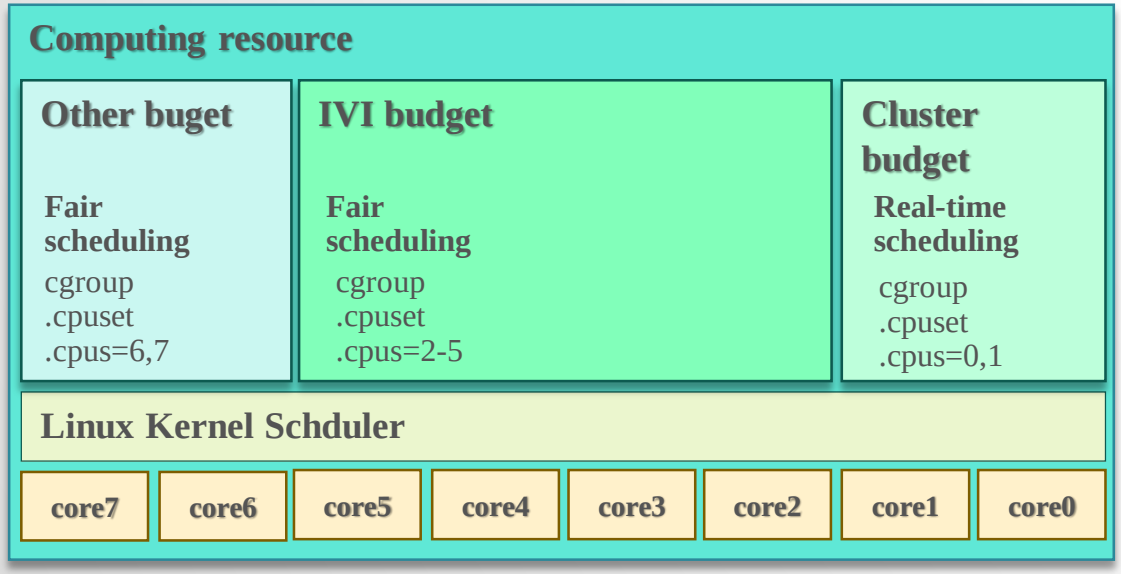


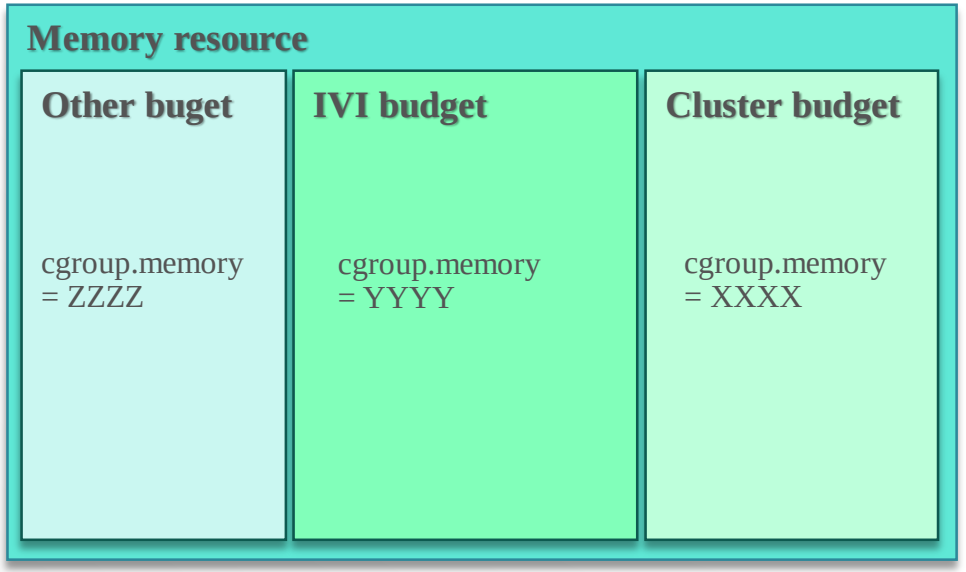
QM Isolation

- Resource isolation
  - Computing resource
    - Quickly response is required in instrument cluster container. It require to real-time scheduling.
      - When all of system build by real-time scheduling, must protect instrument cluster computing budget from IVI.
      - These context priority design is too complexly. Difficult to rapid innovative development in IVI container.
  - Memory resource
    - When all of memory was allocated, new memory allocation request cause forced process kill or process deadlock in typical linux system.
      - In this case, must protect instrument cluster memory budget from IVI.
      - These memory allocation issues need to be closed in the IVI.

QM Isolation

- Computing resource isolation
  - CPU shielding based isolation using cgrpup cpuset.
    - It realize isolated scheduling class for instrument cluster container. We can use real-time scheduling isolated from other container.
- Memory resource isolation
  - Memory budget guarantee using cgroup memory
- Easy to re balancing, it's strong point against hypervisor.

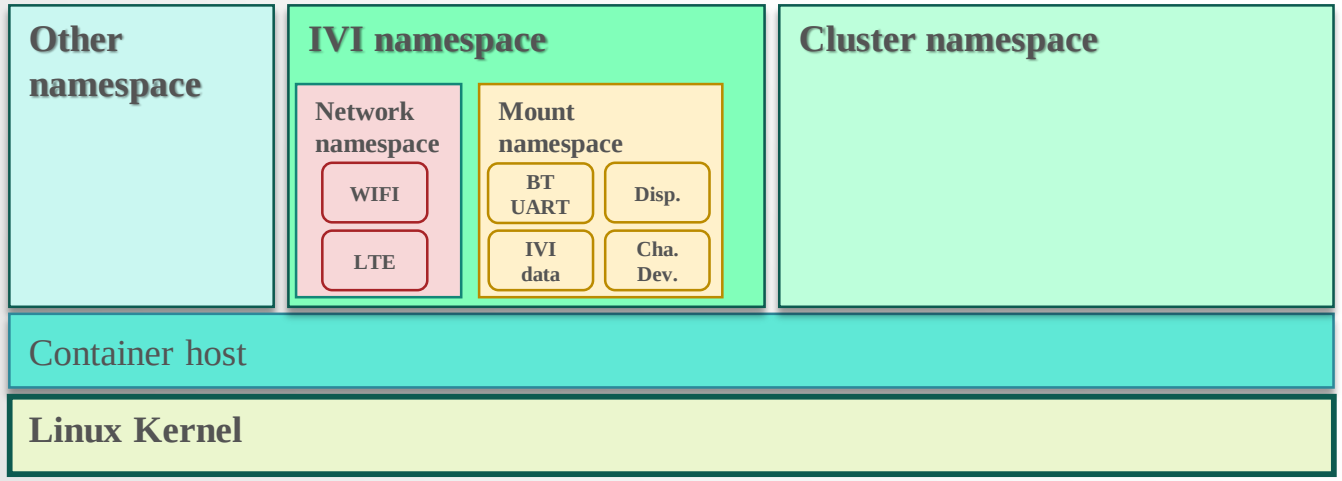




AUTOMOTIVE GRADE LINUX

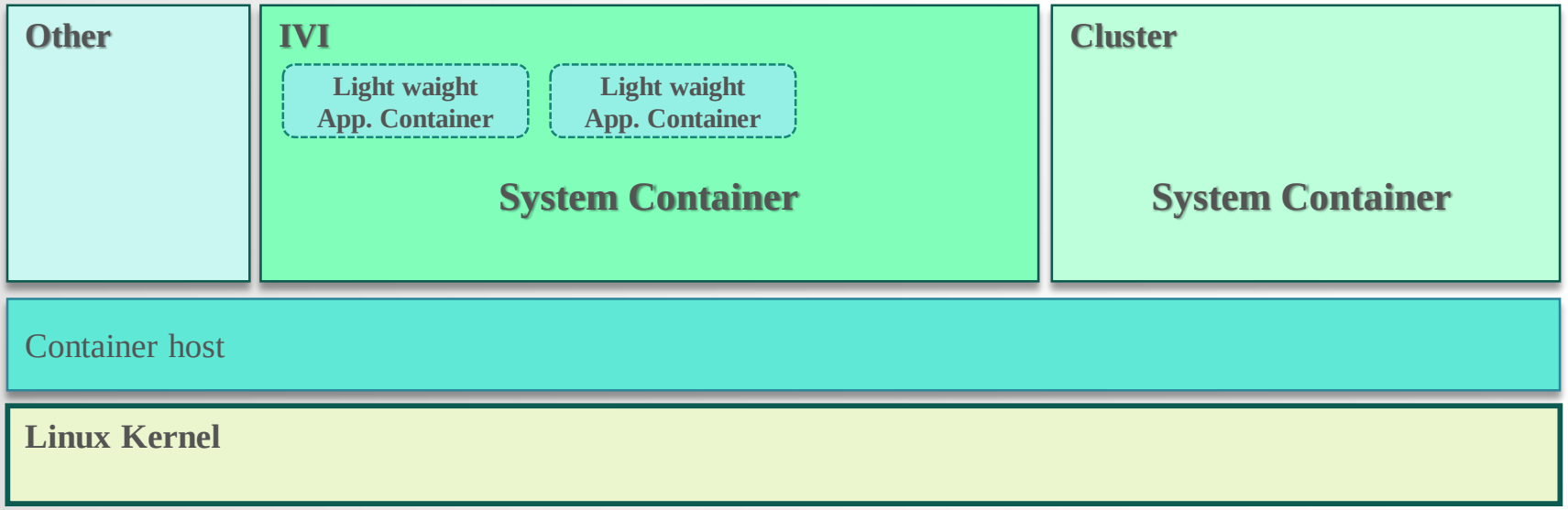
QM Isolation

- Hide resources from other containers
  - IVI container has connected and connectivity device such as Blutooth, wifi, LTE and more.
    - In cyber security point of view, these device should insert into container and hide to other container.
  - Device hiding is most easy ways to protect from illegal device access.
    - Network name space realize network device hiding.
    - Mount namespace realize character and block device hiding.



QM Isolation

- System container vs application container
  - System Container
    - It realize to isolated linux system.
  - Application Container
    - It realize to isolated application runtime environment.
- Choice a system container
  - To realize container local lifecycle management.
  - To realize light weight application container inside IVI container.



QM Isolation

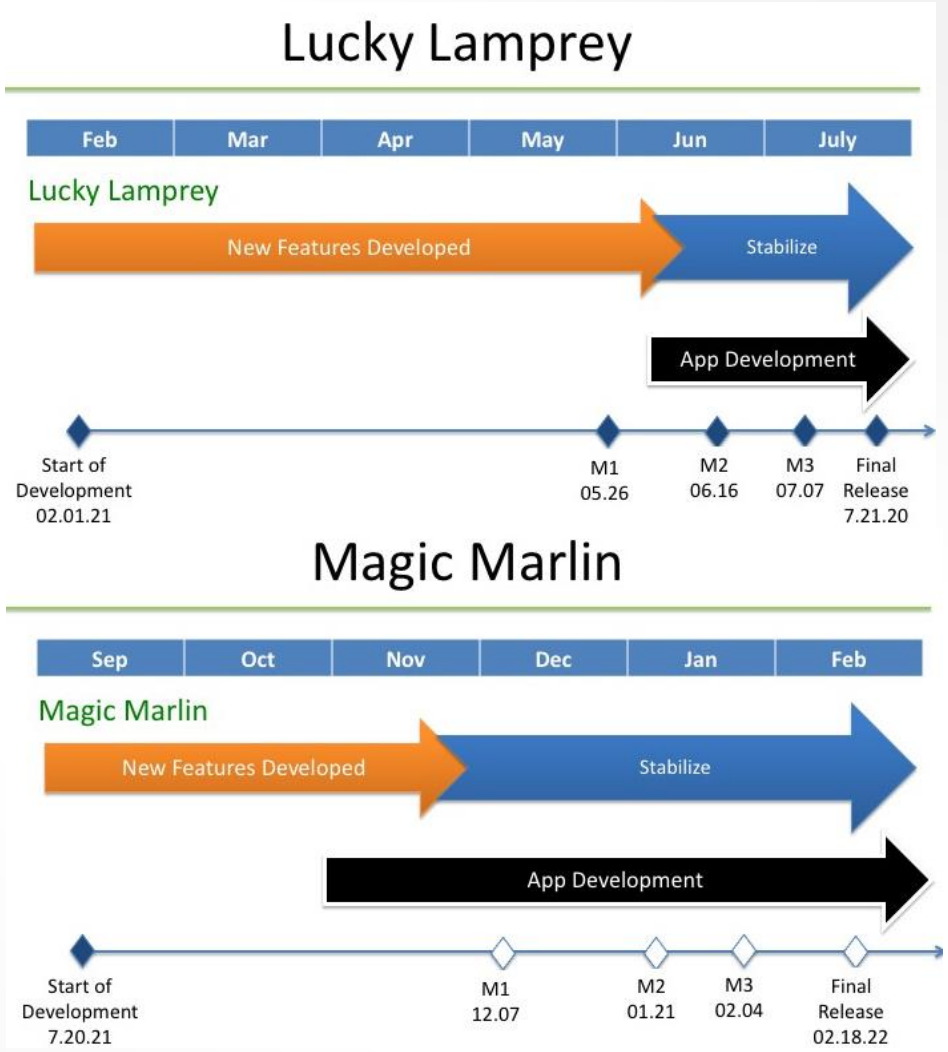
- Summary for QM Isolation
  - Software type based isolation using Linux container technology.
    - Software binary isolation using chroot.
    - Common resource isolation using cgroup.
    - Device isolation using namespace based hiding.
  - Each container build by system container.
    - To support container local lifecycle.

Outline

- Background
- QM Isolation Architecture
- **AGL Instrument Cluster**
- Conclusion

AGL Instrument Cluster

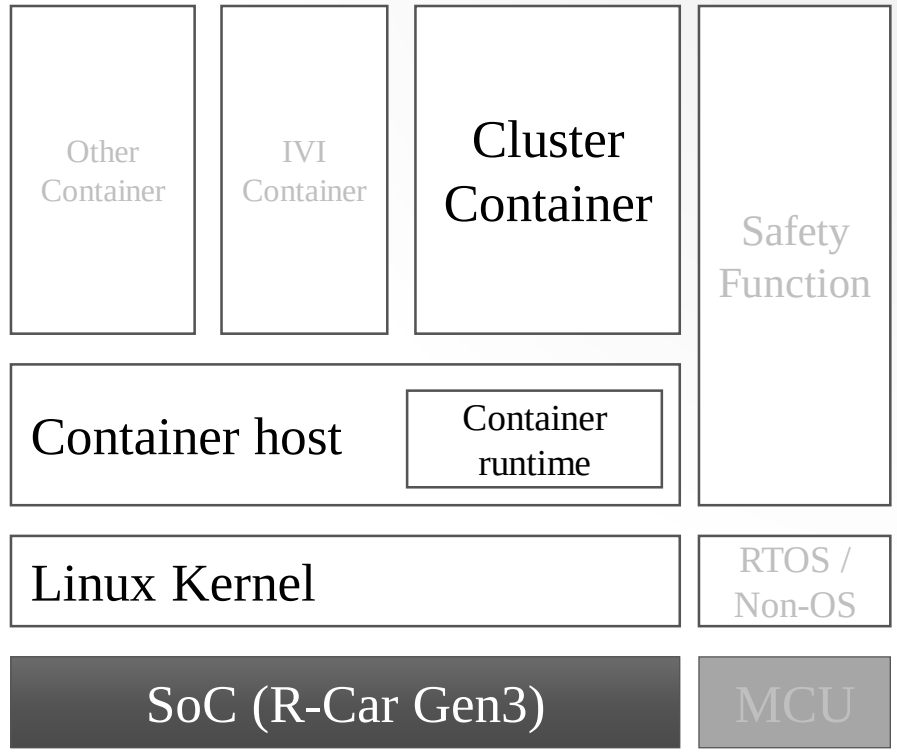
- AGL Lamprey released in July.
  - Upstreaming start.
  - Most basic integration feature was merged.
  - This release support container integration using Yocto.
- AGL Marlin will release in Feb.
  - Some common software will merge.
  - Some development issue will fix.
  - Some demo software will merge to evaluate AGL Instrument Cluster architecture.



AUTOMOTIVE GRADE LINUX

Status for Lamprey release

- Most basic integration feature was merged.
  - Limited container host and cluster container are available.
  - Display isolation feature is integrated. But it support single display only.
  - Demo GUI is available. It support standalone demo.
  - Support container image build using Yocto.
  - R-Car H3/M3 starter kit without kingfisher board support only.

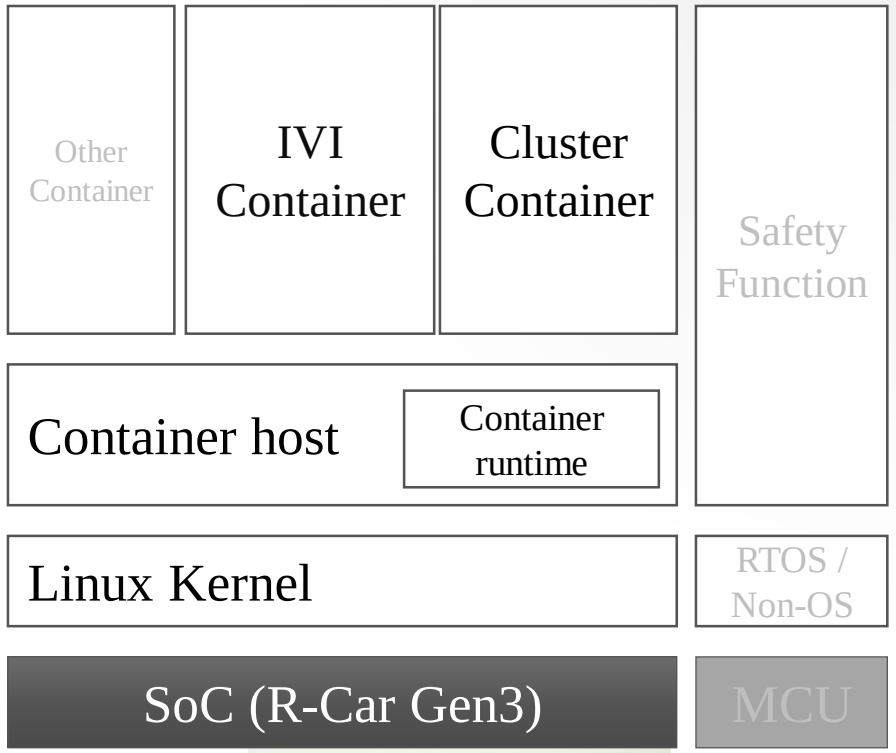


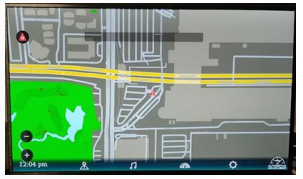
Demo Display 1920x1080

R-CarH3

Plan for Marlin release

- Some common software will merge.
  - IC service framework.
  - ICCOM.
  - Multiple display support.
  - Guest networking support.
  - Sound support.
- Some development issue will fix.
  - Firmware downloading.
  - Input device support at compositor in container.
- Some demo software will merge.
  - IC service for demo.
  - IVI demo application.
  - Ref-hw support.





Infotainment Display 1920x1080

R-CarH3+Kingfisher

Cluster Display 1920x720

AUTOMOTIVE GRADE LINUX

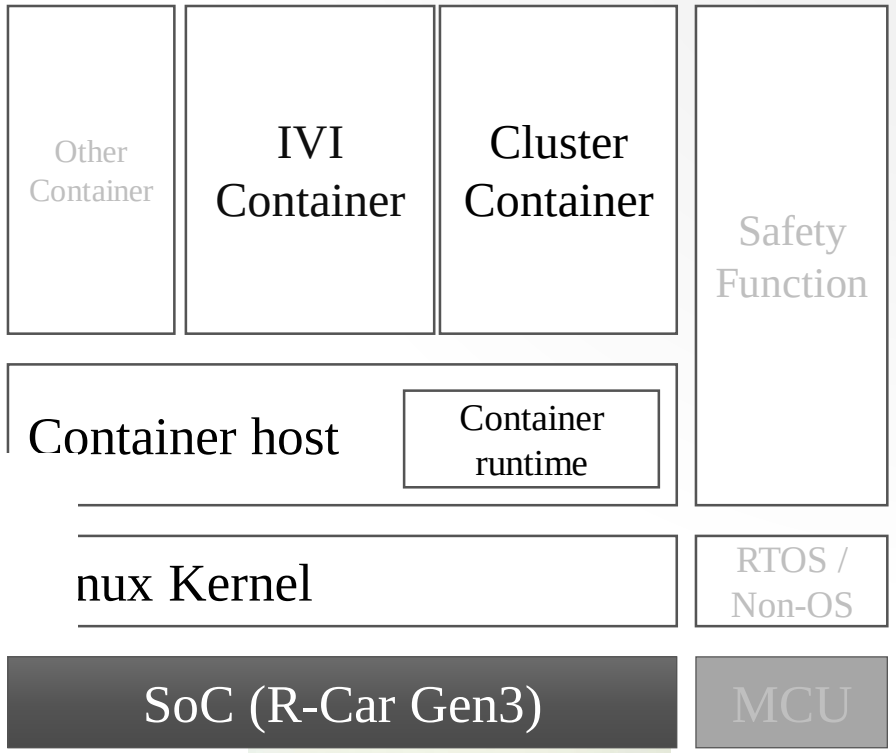
Plan for Marlin release

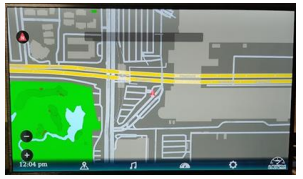
- Some common software will merge.
  - **IC service framework.**
  - ICCOM.
  - **Multiple display support.**
  - Guest networking support.
  - Sound support.

We will merge some new features into AGL Marlne release.
In this topic, we share a part of new features design.

- Some
  - Firr
  - Input device support at compositor in container.

- Some demo software will merge.
  - IC service for demo.
  - IVI demo application.
  - Ref-hw support.





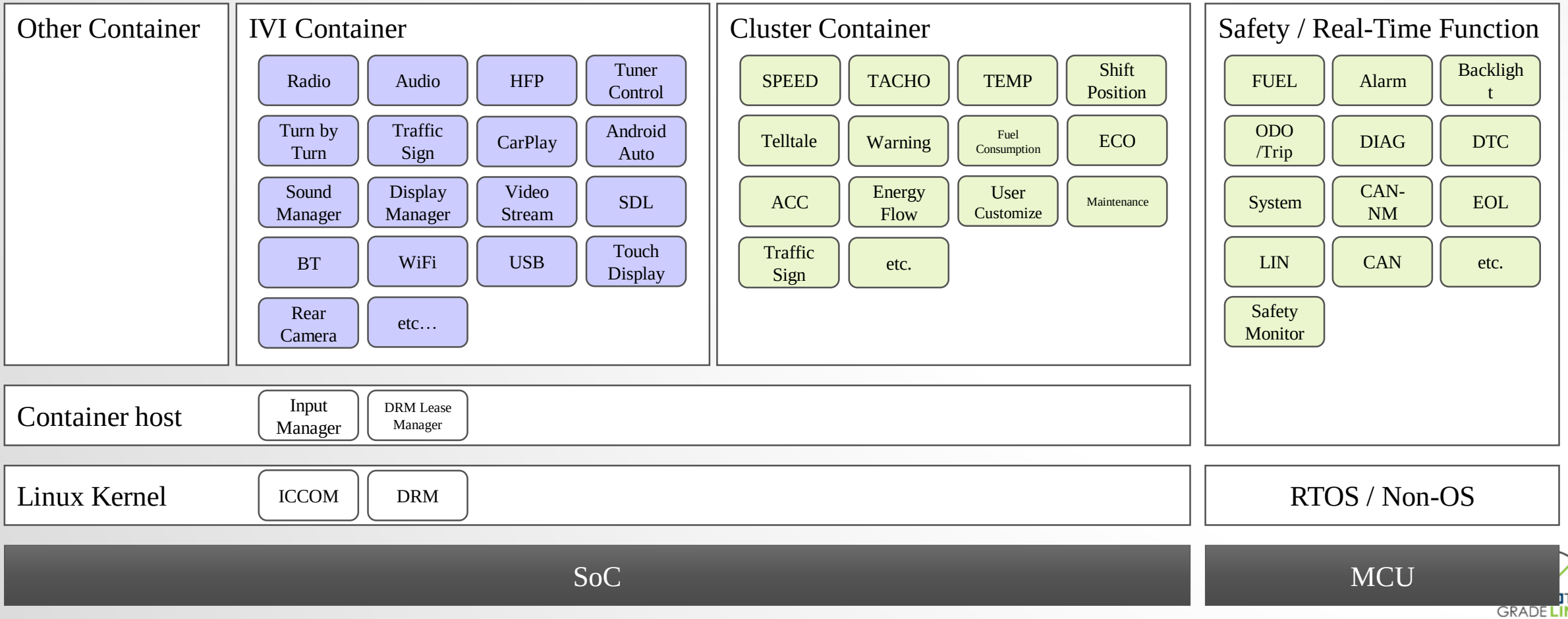
Infotainment Display 1920x1080

R-CarH3+Kingfisher

Cluster Display 1920x720

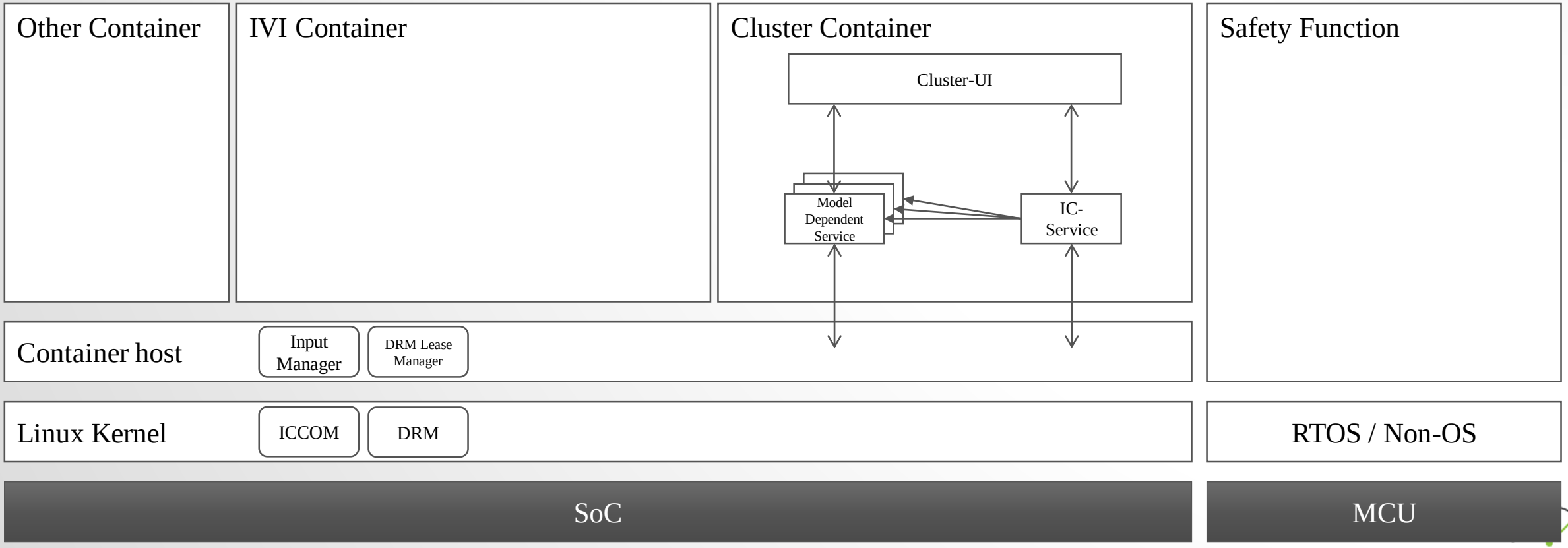
Function Block Assignment Definition

- Safety monitoring and real-time function which includes device access shall be assigned outside of AGL.
- All of the other cluster function shall be assigned onto the cluster container.



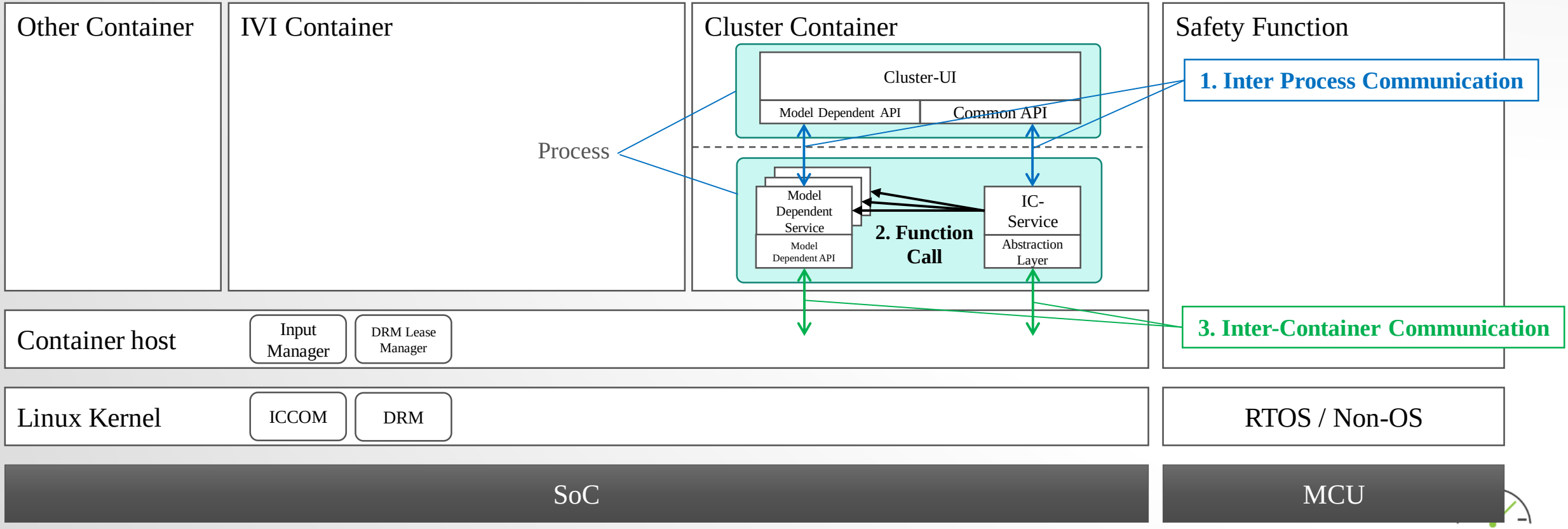
Cluster Service and UI

- Cluster container shall consist of IC-Service and Cluster-UI component.
  - IC-Service shall consist of a function logic.
  - Cluster-UI shall consist of an UI state machine and assets.
  - IC-Service shall be separated by a model dependency.



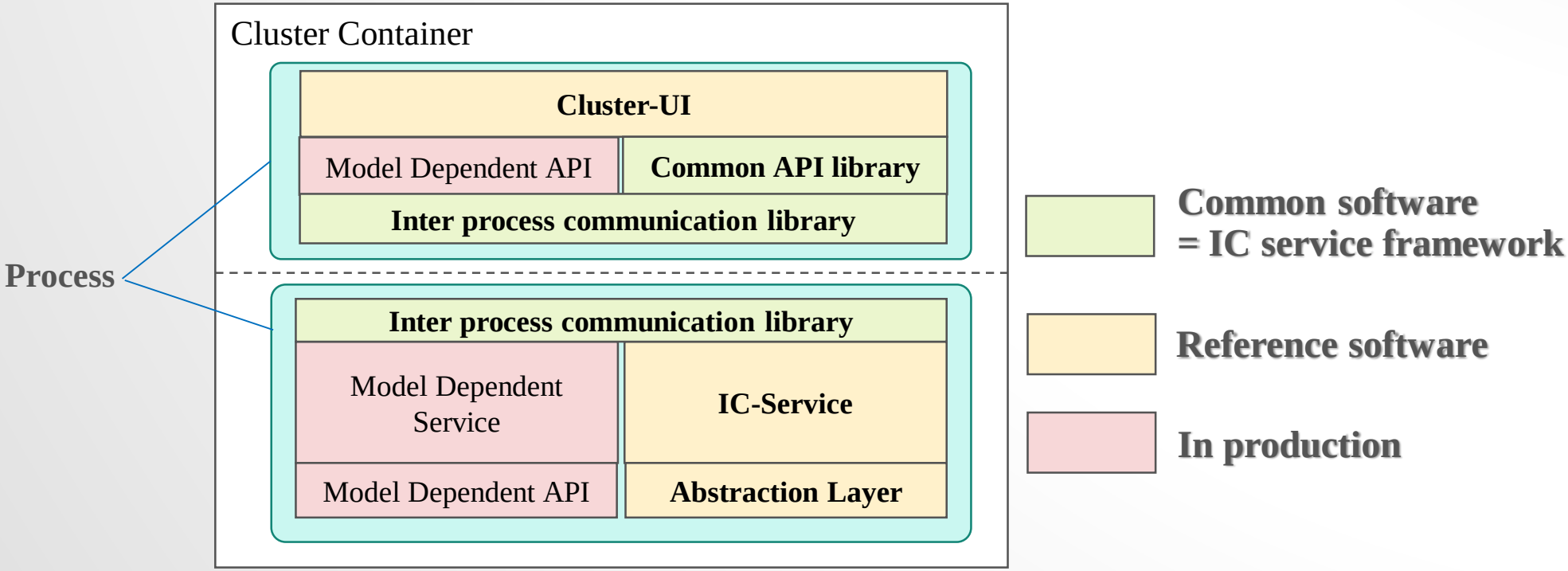
IC-Service Interfaces

- IC-Service shall consist of the three type of interface.
  - Cluster-UI shall be defined a separated process.
  - Model dependent service shall be called from IC-Service as a common interface.
  - IC-Service shall communicate with outside of container.



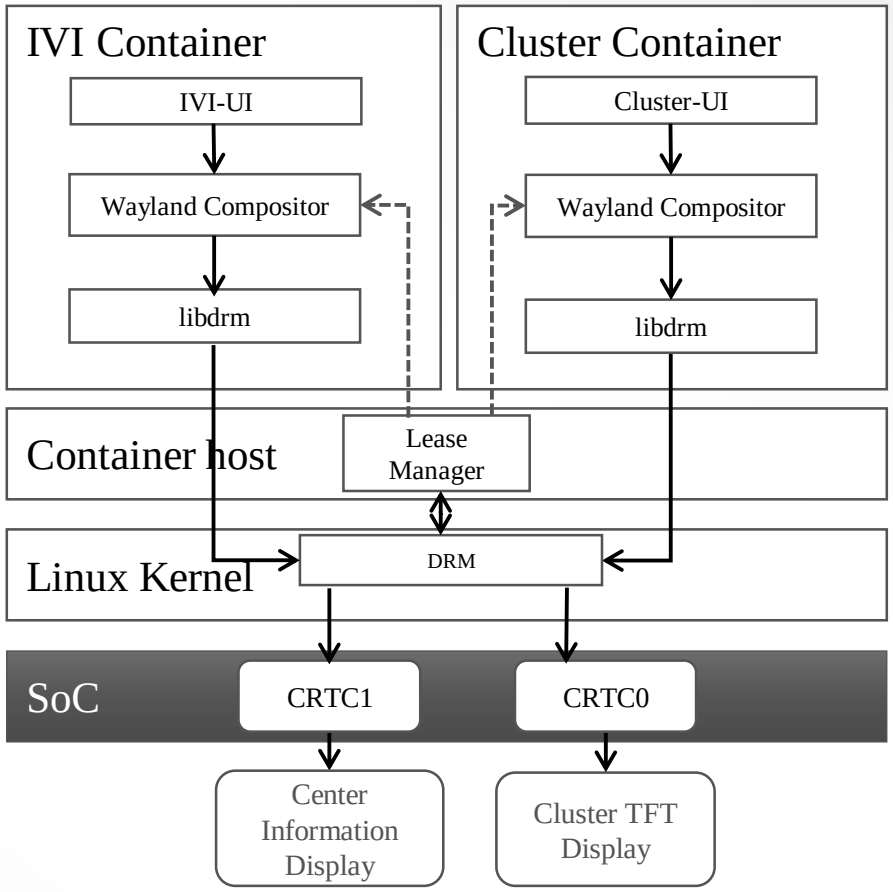
IC Service Framework

- IC service framework consist from Common API library and Inter process communication library.
  - It designed for instrument cluster use case.
  - Already merged in AGL master branch.
  - Design document and API specification published in AGL confluence.



Multiple Display Support

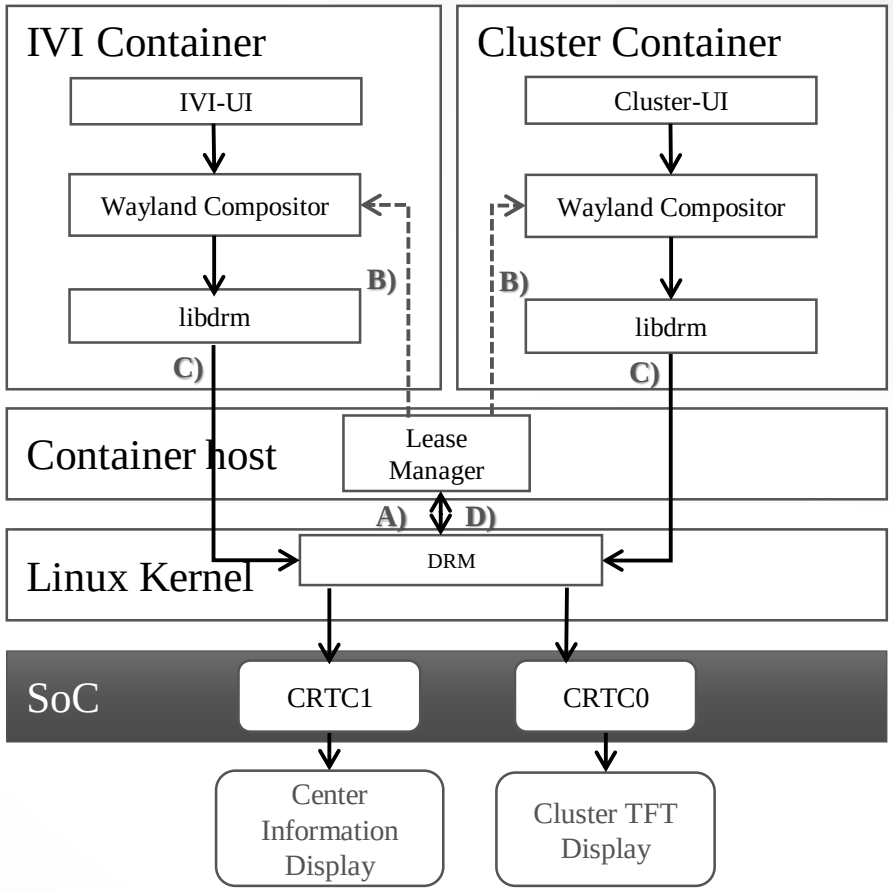
- Basic design
  - Cluster and IVI have one own display.
  - Each container have compositor inside a container.
- Issue
  - One drm display device has one or more displays.
    - ex. card0-HDMI-A-1, card0-HDMI-A-2, card0-DVI-D-1
    - That device shall manage by only one process, it call drm master. The drm master is usually assumed by compositor.
  - What does this mean?
    - One display device can manage by only one compositor.
- Our solution
  - Create drm lease manager to realize multiple compositor using drm lease kernel feature.
  - The drm lease kernel feature is introduced in 4.15.



Drm Lease Manager

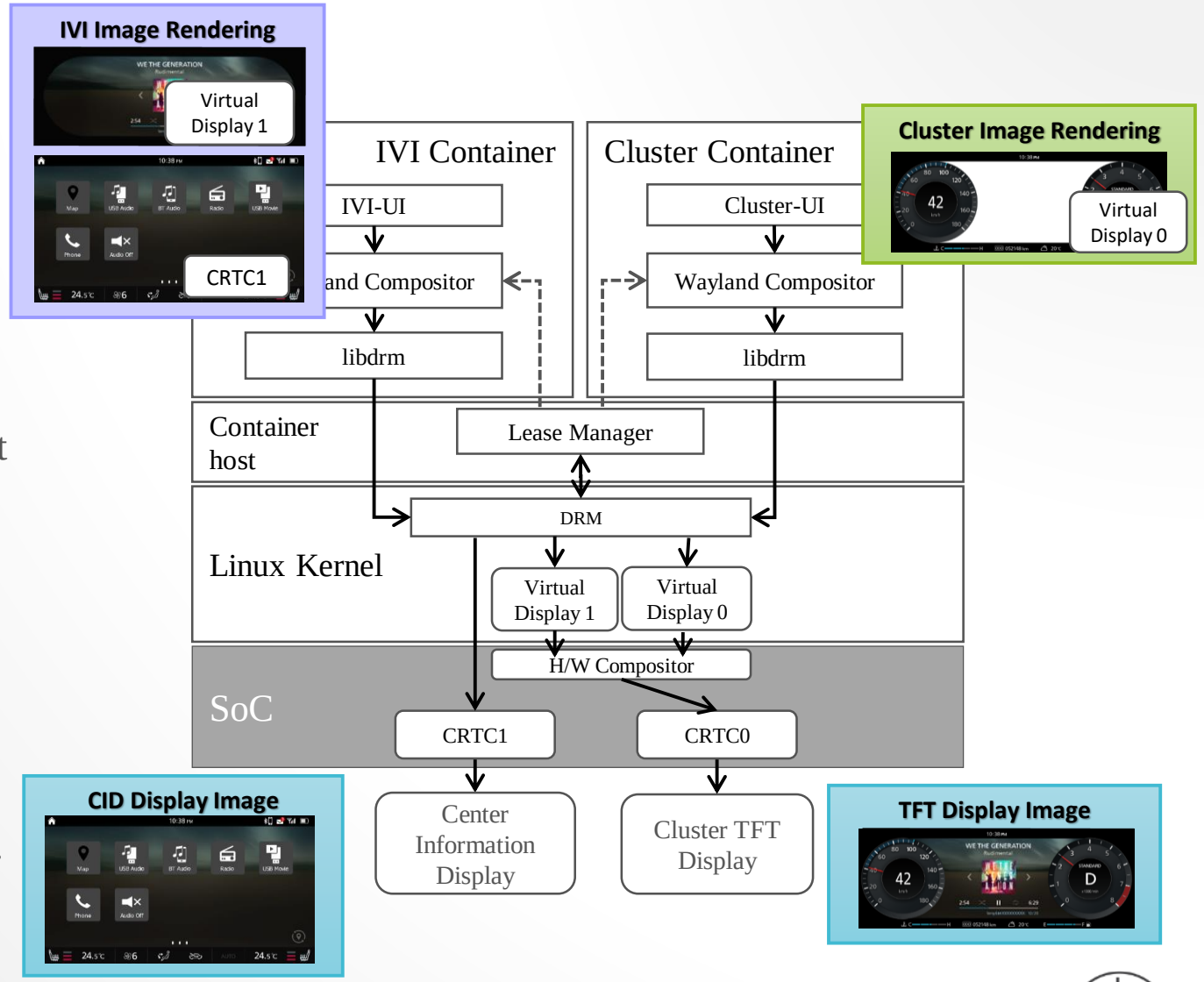
- Basic design
  - The traditional drm master (fd), created by opening /dev/dri/cardX, which can create and lease out resources to other DRM Masters. The drm lease manager takes on this role.
  - Each compositor controls resources leased from drm lease manager.

- Control flow
  - A) DRM Lease Manager(daemon) opens DRM device and creates lessee DRM Masters for each client.
  - B) Sends lessee file descriptor to clients (via lease client library) when requested
  - C) Clients render directly to DRM device using lessee
  - D) DRM Lease Manager revokes lessee when the client is finished with it.



For product use case

- In product use case
  - The cluster screen show a part of the IVI information such as map, music track and A/C.
- Our proposed design
  - IVI container render two display images.
  - One is CID image. It will be rendered directly to the center information display.
  - Another one is a shared image for cluster display. It shall be rendered onto the virtual display.
  - Cluster image is rendered another channel of the virtual display.
  - Both image shall be composite by H/W.
  - And then, final image shall be displayed onto the cluster display.
- This implementation is highly depend on BSP.
  - These are TODO task.

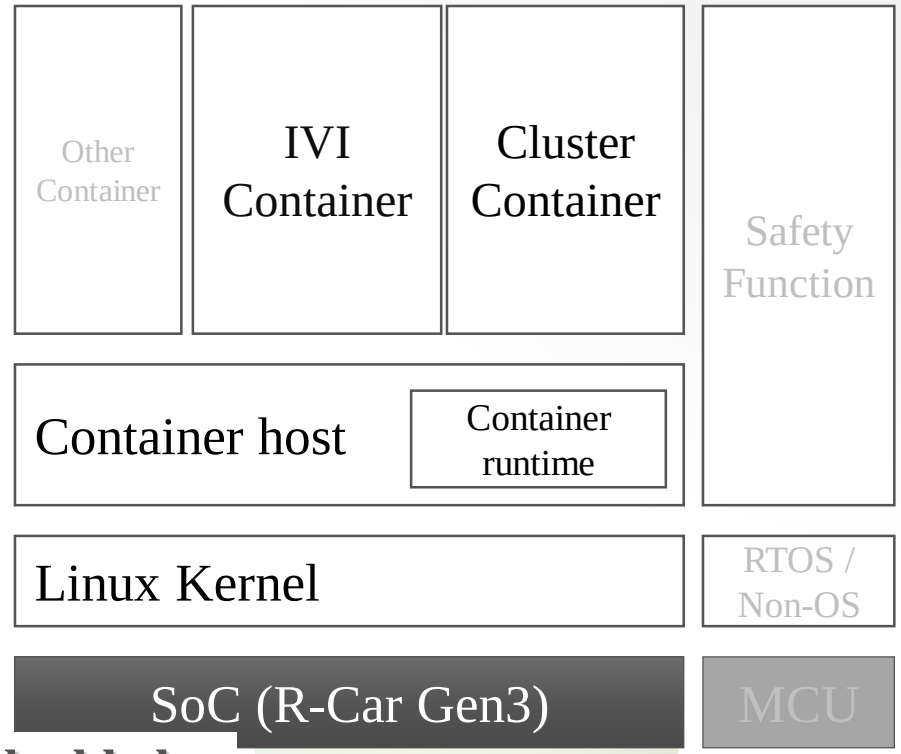


Plan for Marlin release

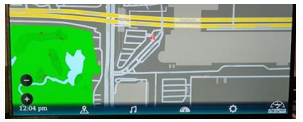
- Some common software will merge.
  - IC service framework.
  - ICCOM.
  - Multiple display support.
  - Guest networking support.
  - Sound support.

- **Some development issue will fix.**
  - **Firmware downloading.**
  - **Input device support at compositor in container.**

- Some de
  - IC servi
  - IVI demo application.
  - Ref-hw support.



We have solved various integration issues for embedded container through this development.
In this topic, we share what issue how to fixed.



Infotainment Display 1920x1080

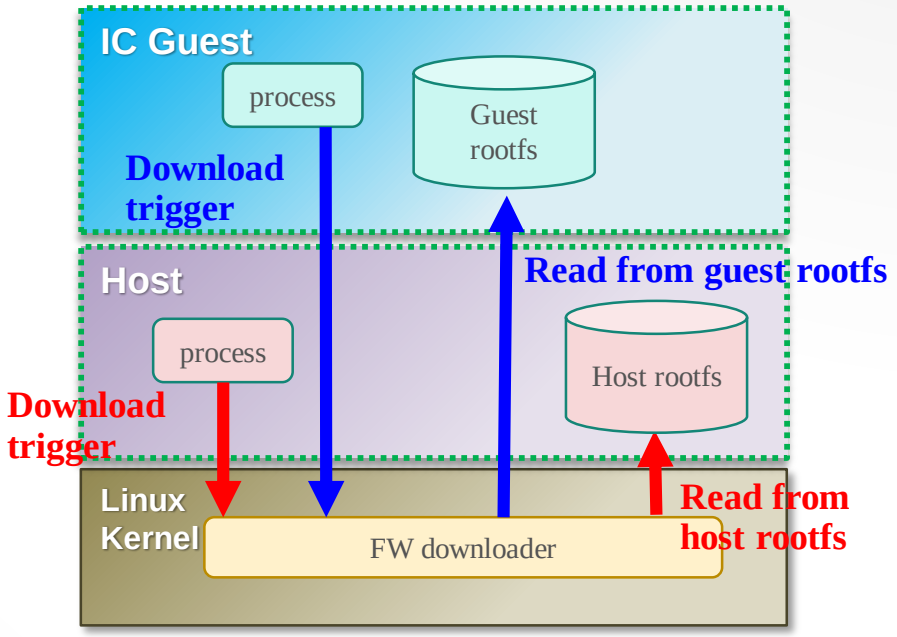
R-CarH3+Kingfisher

Cluster Display 1920x720

AUTOMOTIVE GRADE LINUX

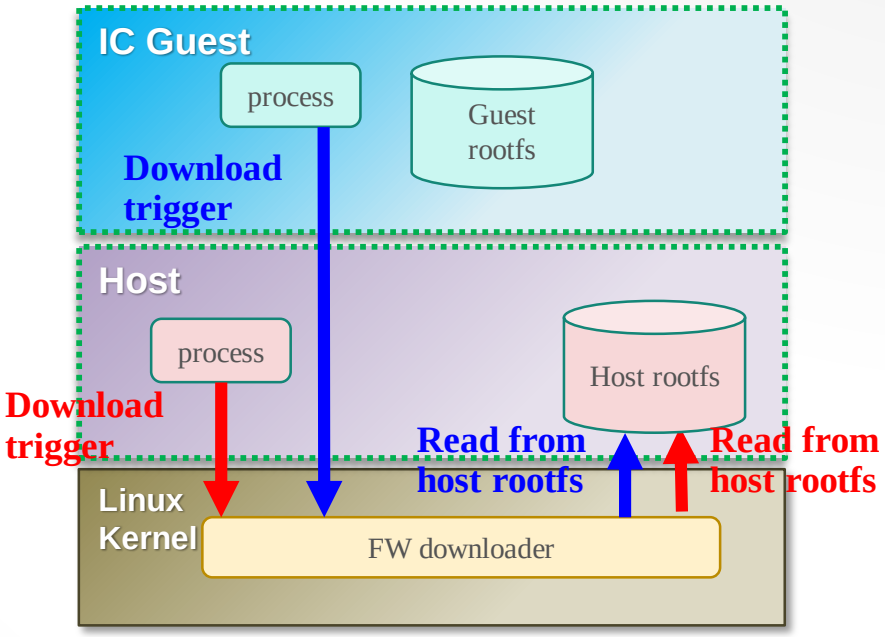
Firmware downloading

- Typical behavior for firmware downloading.
  - When driver need firmware, firmware downloader read firmware file from root file system. When that firmware downloading is triggered, it's depend on driver implementations.
  - At kernel module loading.
    - Kernel modules are only loaded by host triggers. In this case, it is not having a issue.
  - At 1'st device file open.
    - Device file are opened by both host and guest. In this case, it is having a issue.
- What is issue
  - Firmware downloader inherit mount namespace.
    - When triggered from host, firmware read from host root file system by firmware downloader.
    - When triggered from guest, firmware read from guest root file system by firmware downloader.
      - In this case, firmware download will fail.



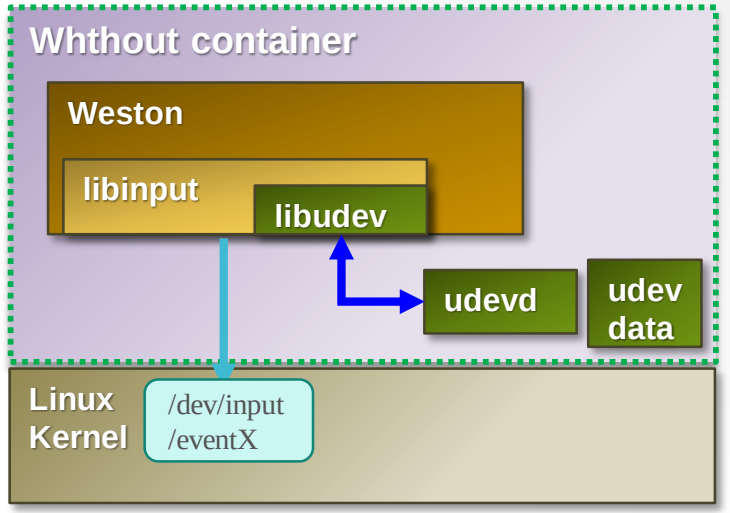
Firmware downloading

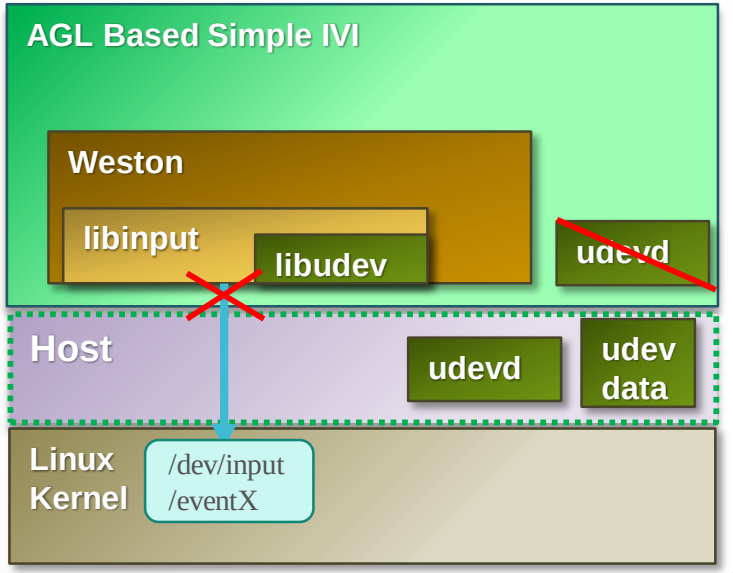
- In kernel upstream, firmware downloader is improved in 5.7.
  - Firmware downloader doesn‘t inherit mount name space. In this case we can controlling which firmware is downloading.
  - Patch
    - https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=901cff7cb96140a658a848a568b606ba764239bc&h=linux-5.7.y
  - Renesas R-Car Gen3 boards migrate kernel version from 5.4 to kernel 5.10 in current master branch.
    - This fix already merged.
  - The default kernel version of yocto dunfell is kernel 5.4. This version is not merged this patch.
    - We tried to back port this patch, it's successfully. No needed to additional modify from original patch.



Input device support at compositor in container

- Current guest container is not support input device
  - Weston use libinput to detect and handle input devices.
  - The libinput depend on libudev and udev-data that is created by udevd.
    - udev-data: /var/run/udev/data/*
- What is issue
  - Libinput highly depend to udev.
  - Systemd-udev couldn't work in guest.
    - Device event pass to only host from kernel. Can't receive in guest.
    - Same situation for eudev.
  - Udev-data is not created, libinput can't find input device.
  - Weston can't handle input device in guest.

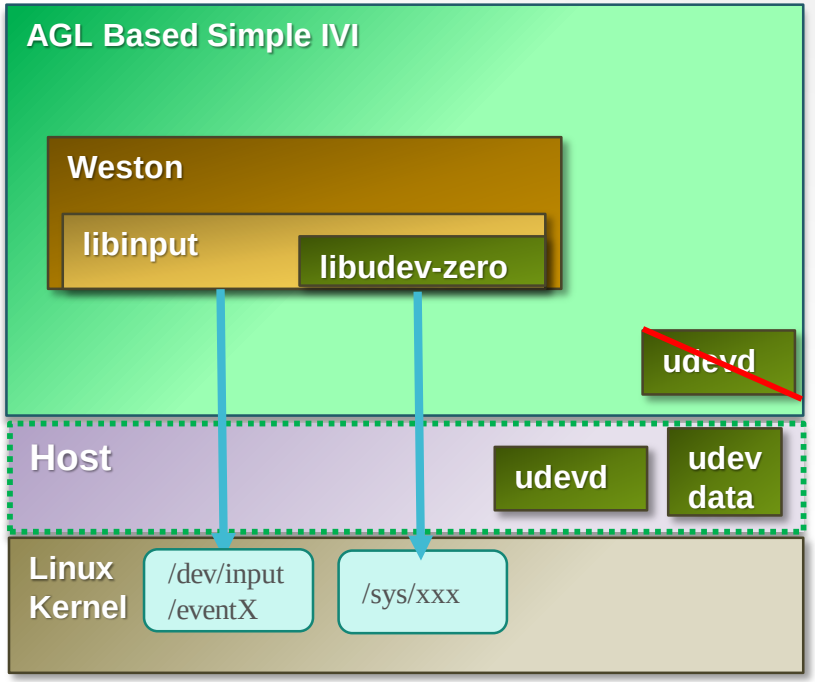




Input device support at compositor in container

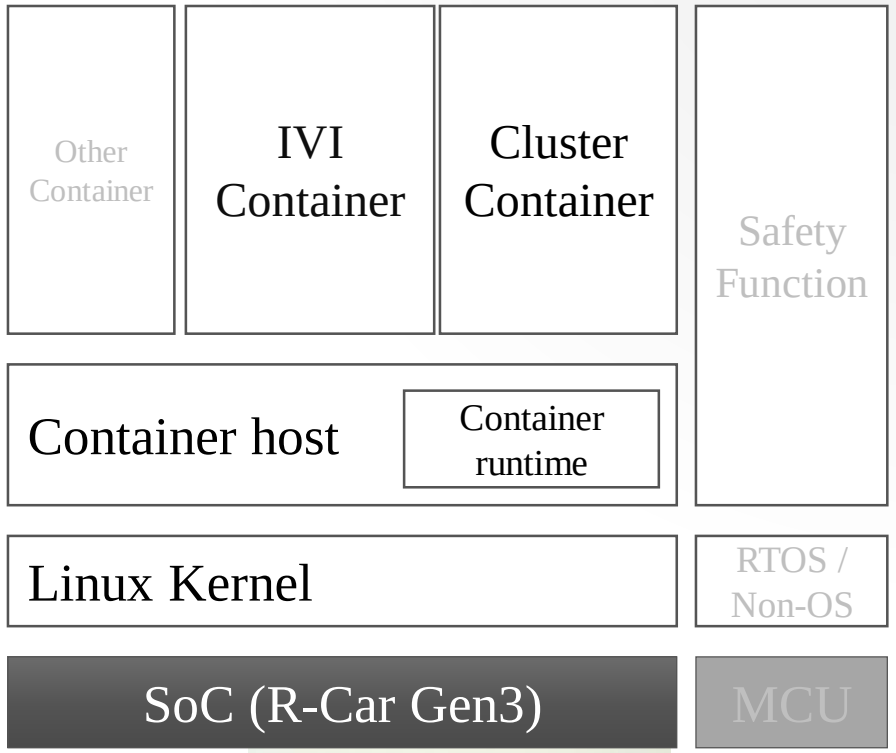
- libudev-zero
  - Daemonless udev implementation replacement for libudev.
    - https://github.com/illiliti/libudev-zero
  - libusev-zero support udevd less input device detection based on sysfs scanning.
    - When guest container mounts sysfs, libinput can detect input device informed by libudev-zero.

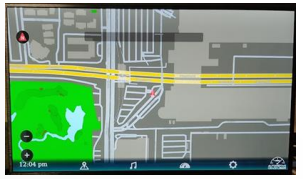
- Our proposed design
  - Libudev-zero use in guest to realize generic input device detection method.
  - Current issue:
    - When libudev replace to libudev-zero, it require to hard HACK to systemd recipe such as forces remove systemd-udevd and libudev from systemd packages.
    - We are now investigating what is more better integration such as libudev-zero will be static linking to libinput.



Plan for Marlin release

- Some common software will merge.
  - IC service framework.
  - ICCOM.
  - Multiple display support.
  - Guest networking support.
  - Sound support.
- Some development issue will fix.
  - Firmware downloading.
  - Input device support at compositor in container.
- **Some demo software will merge.**
  - **IC service for demo.**
  - **IVI demo application.**
  - **Ref-hw support.**





Infotainment Display 1920x1080

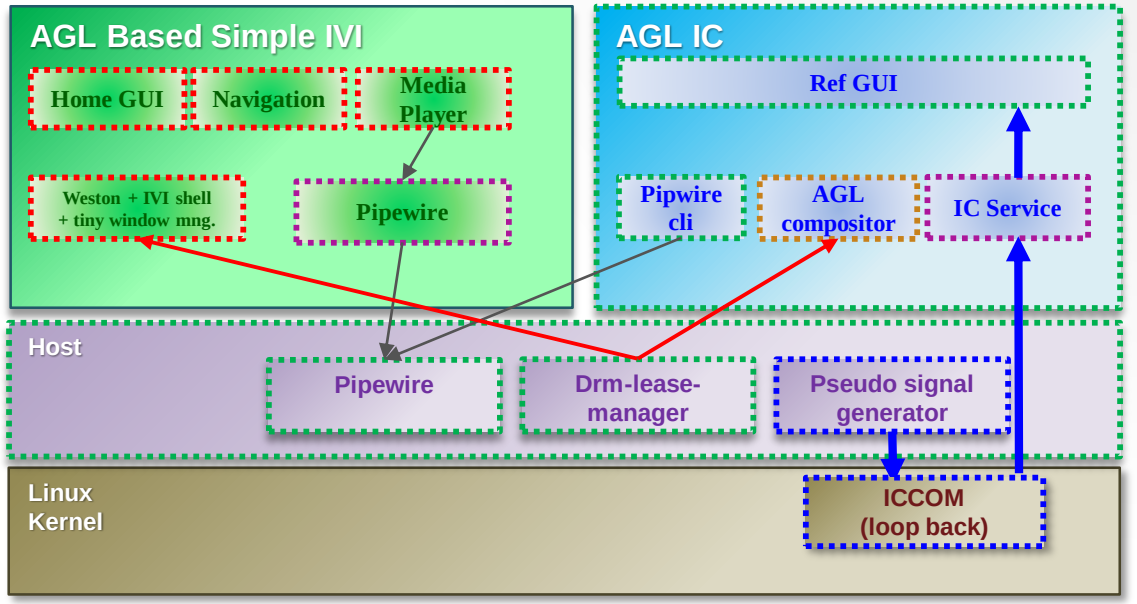
R-CarH3+Kingfisher

Cluster Display 1920x720

AUTOMOTIVE GRADE LINUX

Some demo software will merge.

- IC service framework is already merged in master branch.
  - We develop reference cluster service.
  - Pseudo signal generator passed vehicle information to IC service using ICCOM.
  - Ref-GUI draw speed meter, telltale and more vehicle information, those are received from IC service.
- IVI container is now developing.
  - Sample Navigation, Media Player, Home UI are already merged.
  - These are running on weston with IVI extension.
  - These software are only for demo and evaluation for container environment. Will replace AGL IVI profile based software in future release.



Some demo software will merge

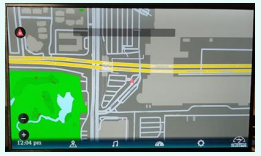
- In initial release at Lamprey, IC software stack run on R-Car H3/M3 Starter Kit with Full HD Display only.
  - It already released.

- In Marlin release, IC software stack with demo IVI run on R-Car H3/M3 Starter Kit on Kingfisher board with two Display.
  - It already merged.

- AGL Ref-HW support merged in last week.

Demo Display 1920x1080

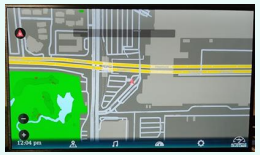
R-CarH3



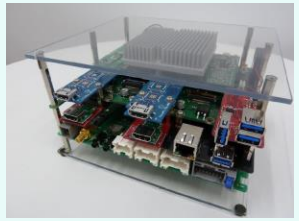
Infotainment Display 1920x1080

R-CarH3+Kingfisher

Cluster Display 1920x720



Infotainment Display 1920x1080



AGL Ref-HW

Cluster Display 1920x720

AUTOMOTIVE GRADE LINUX

Conclusion

- In this presentation, I talked about AGL Instrument Cluster expert group activity.
  - AGL Instrument Cluster expert group is a most active expert group for AGL expert group.
  - EG activities are contributed by many members around the world.

- In this year, we set mile stone that is build up initial software stack.
  - This mile stone will achieve with the Marlin release.

- In next year, we will resume OSS qualification task.
  - This activity of this year was sharing in AGL Tech Day. This activity aim to create OSS assessment standard and assessment software stack for Instrument Cluster.

Thank You