Instrument Cluster EG
Software Architecture Overview
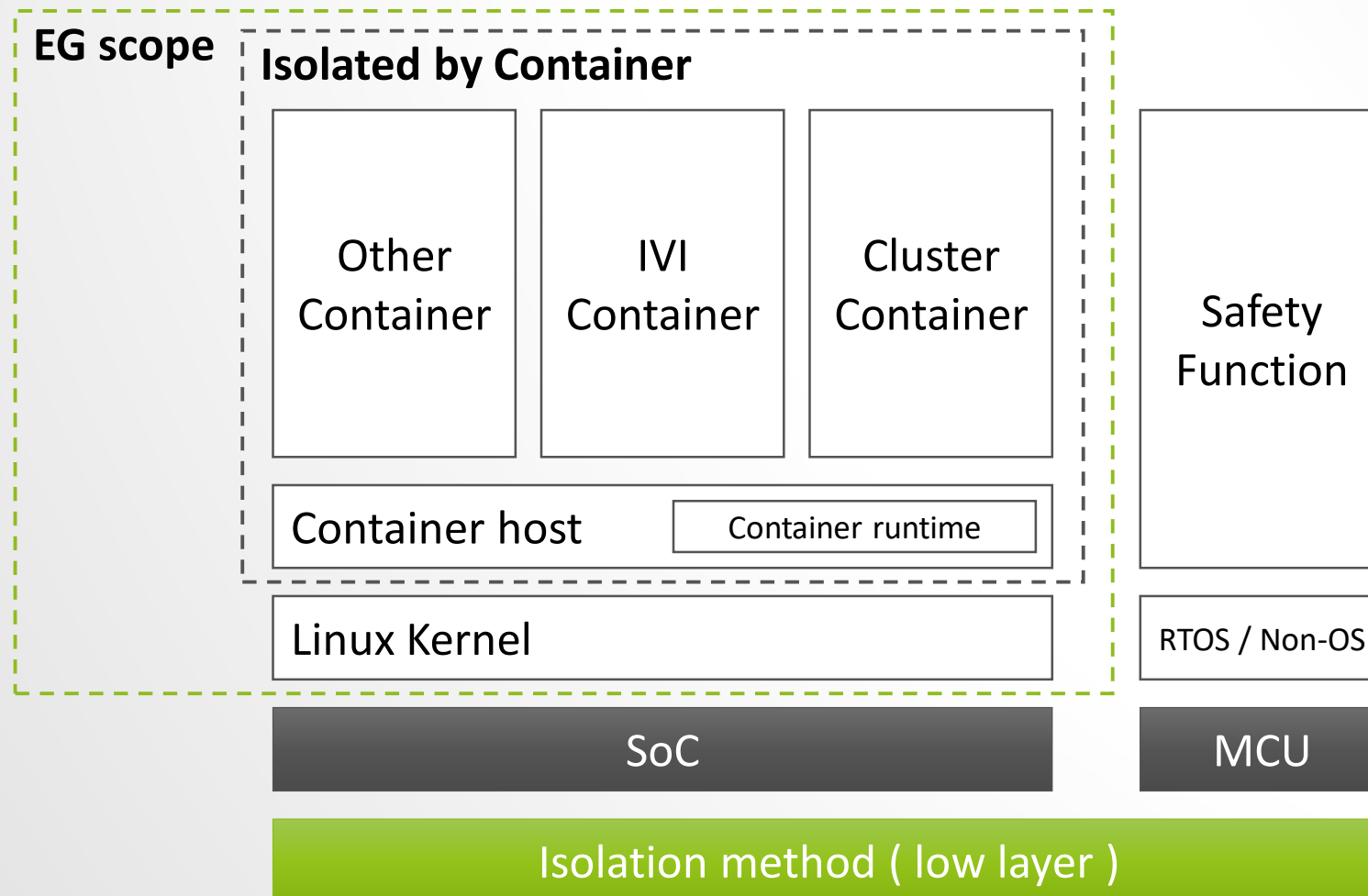
IC-EG Japan development team
29.June.2020

Contents

Container Architecture – Overview

Cluster Function and IVI Function shall be separated by Linux Container Technology in order to achieve QM isolation.

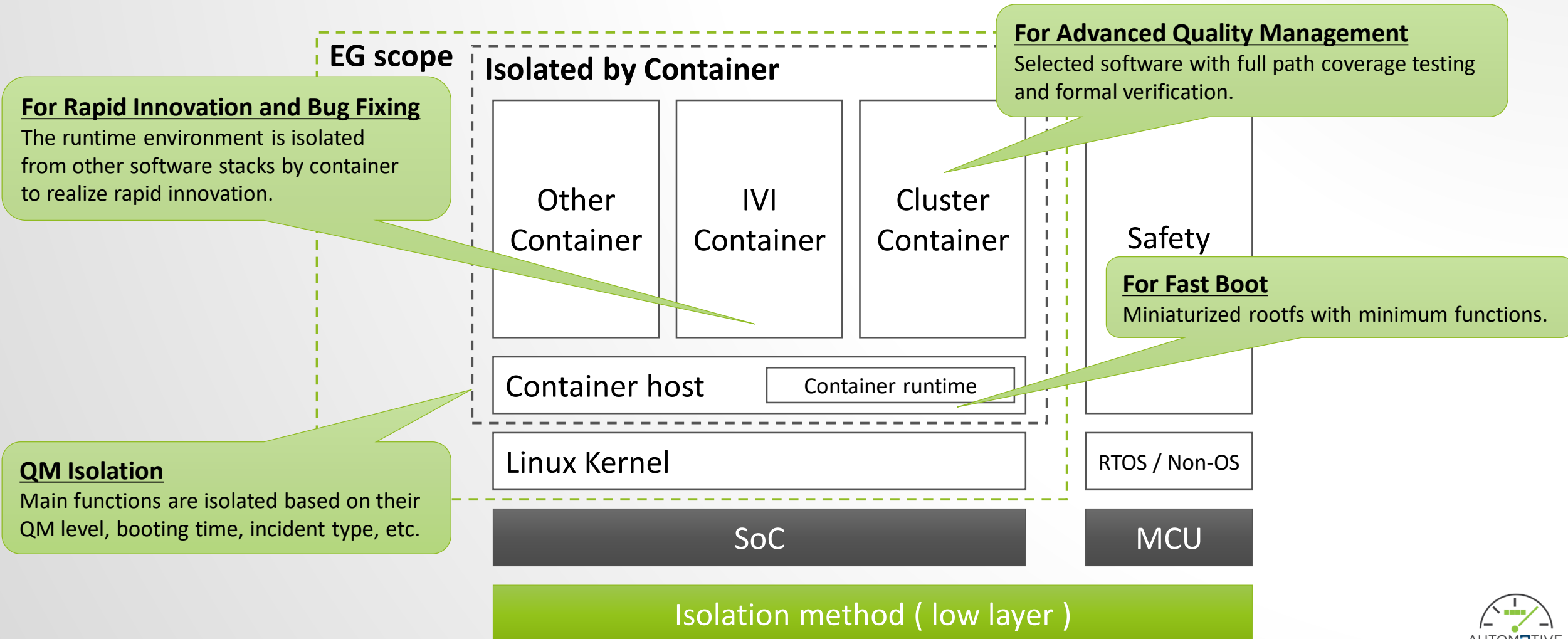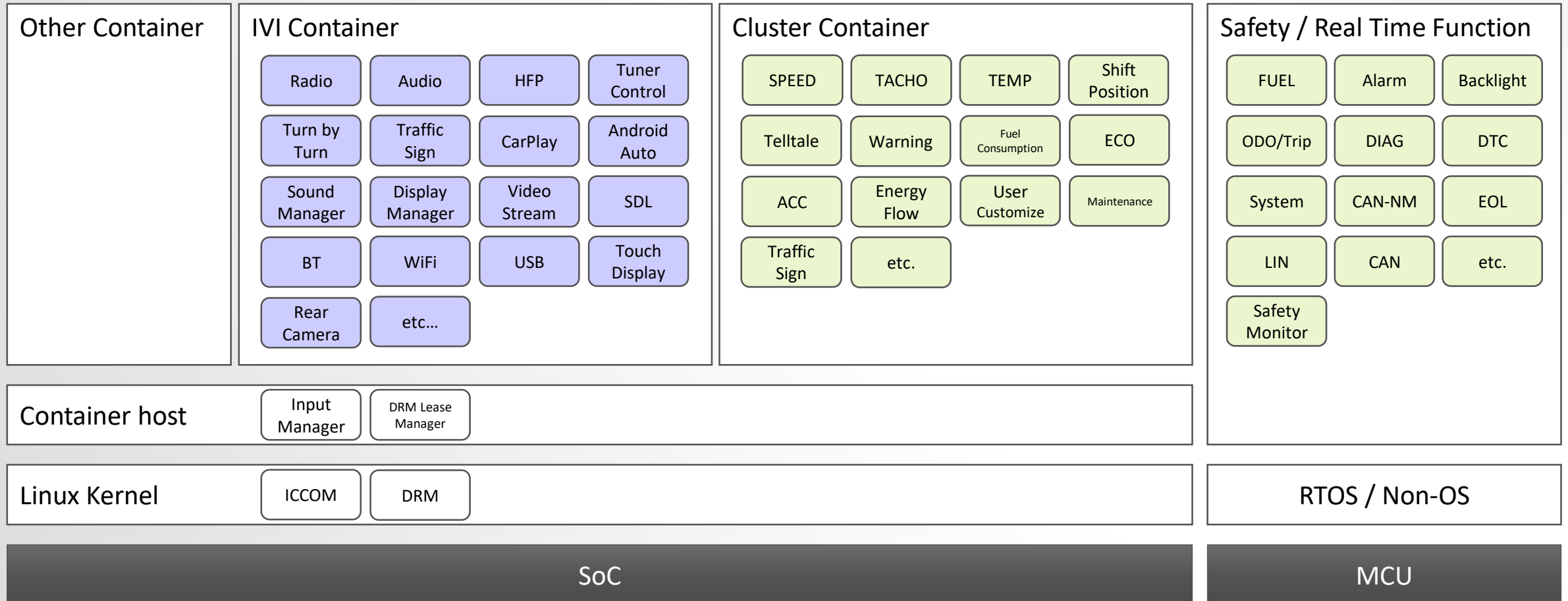# Container Architecture – Overview

Cluster Function and IVI Function shall be separated by Linux Container Technology in order to achieve QM isolation.

For Advanced Quality Management
Selected software with full path coverage testing and formal verification.

For Rapid Innovation and Bug Fixing
The runtime environment is isolated from other software stacks by container to realize rapid innovation.

EG scope

Isolated by Container

Other Container	IVI Container	Cluster Container

Safety

For Fast Boot
Miniaturized rootfs with minimum functions.

Container host   Container runtime

Linux Kernel

RTOS / Non-OS

QM Isolation
Main functions are isolated based on their QM level, booting time, incident type, etc.

SoC

MCU

Isolation method (low layer)

Function Block Assignment

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.

Other Container

IVI Container

Radio	Audio	HFP	Tuner Control
Turn by Turn	Traffic Sign	CarPlay	Android Auto
Sound Manager	Display Manager	Video Stream	SDL
BT	WiFi	USB	Touch Display
Rear Camera	etc…		

Cluster Container

SPEED	TACHO	TEMP	Shift Position
Telltale	Warning	Fuel Consumption	ECO
ACC	Energy Flow	User Customize	Maintenance
Traffic Sign	etc.		

Safety / Real Time Function

FUEL	Alarm	Backlight
ODO/Trip	DIAG	DTC
System	CAN-NM	EOL
LIN	CAN	etc.
Safety Monitor		

Container host

| Input Manager | DRM Lease Manager |

Linux Kernel

| ICCOM | DRM |

RTOS / Non-OS

SoC

MCU

AUTOMOTIVE GRADE LINUX

Cluster Container – Overview

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.
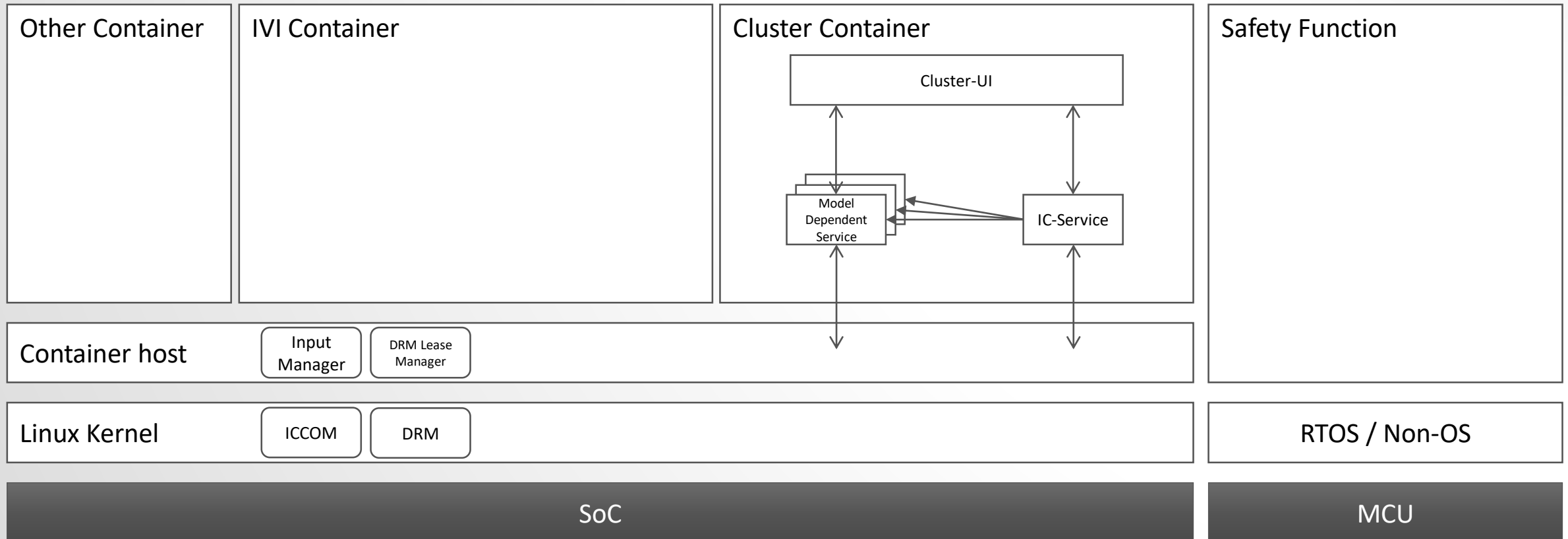
IC-Service Interface

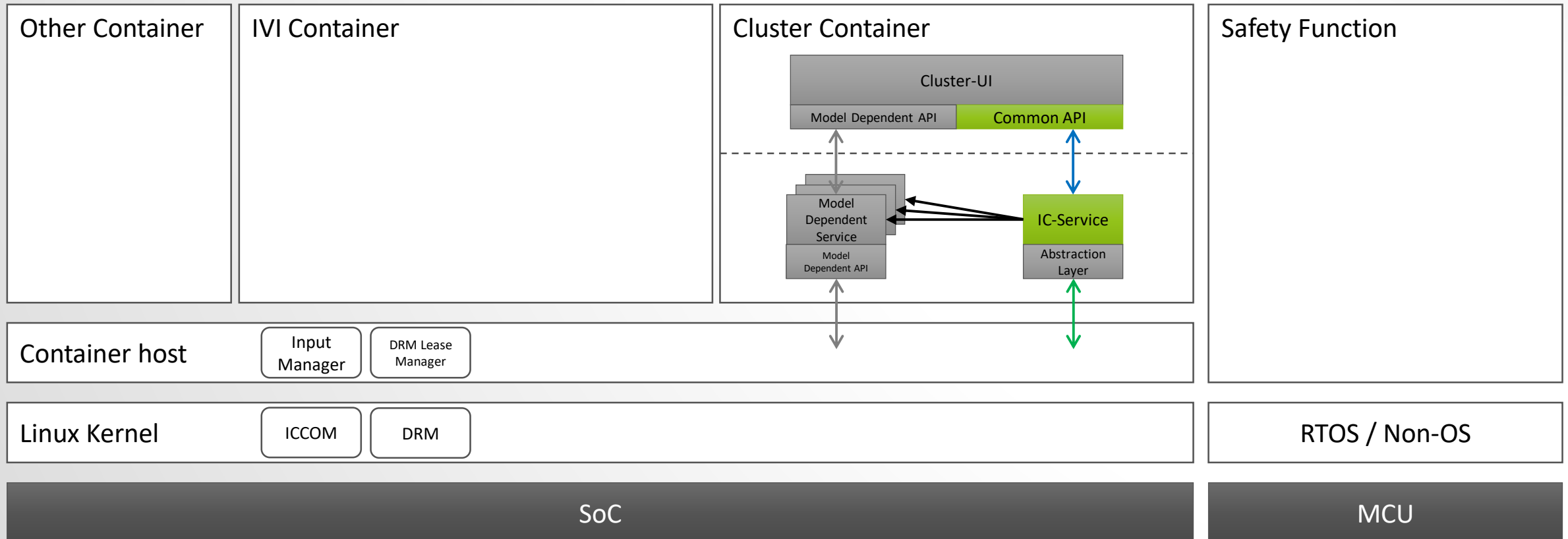IC-Service shall consist of the following three interface.

1. Cluster-UI shall be defined a separated process. → **Inter Process Communication**
2. Model dependent service shall be called from IC-Service as a common interface. → **Function Call**
3. IC-Service shall communicate with another container or container host. → **Inter Container Communication**
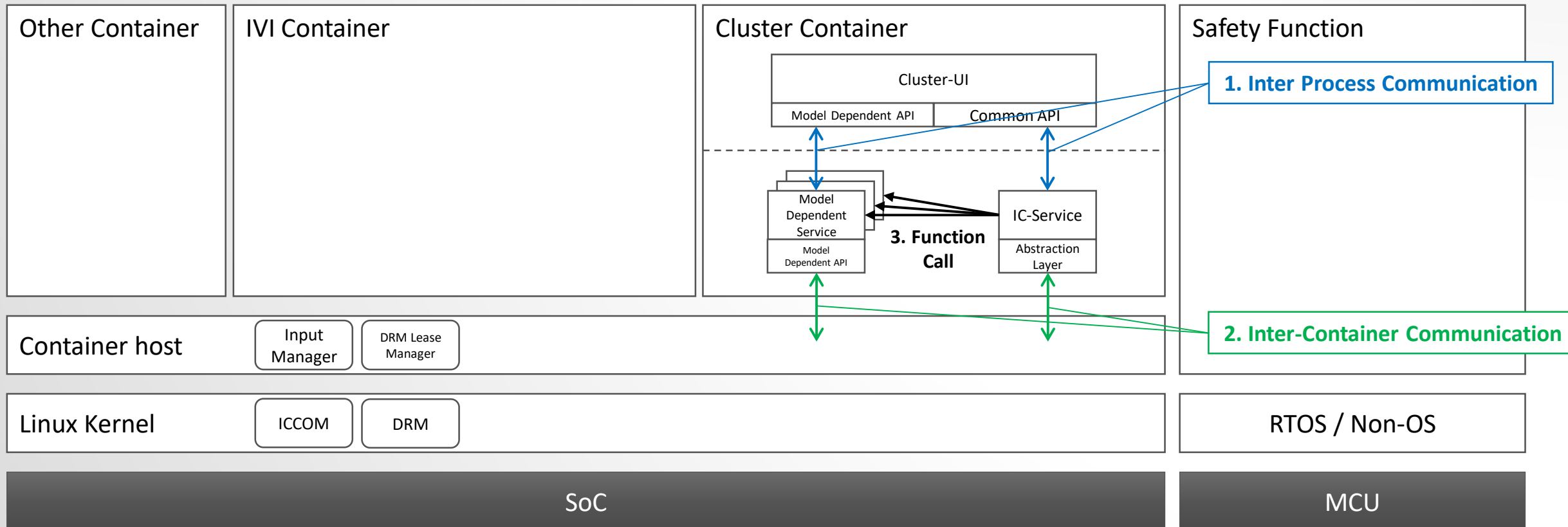
IC-EG Scope

IC-Service logic and common API shall be fully provided by IC-EG.

- The others shall be prepared as for a reference model by IC-EG.

Contents

AUTOMOTIVE
GRADE LINUX

Data Flow – 1. ICCOM

ICCOM is responsible for vehicle signal handling which transferred from MCU. (i.e. CAN)

- ICCOM socket shall be directly opened in application container, and not in the container host.
- Keeping advantage of peer to peer communication shall reduce latency and complexity.
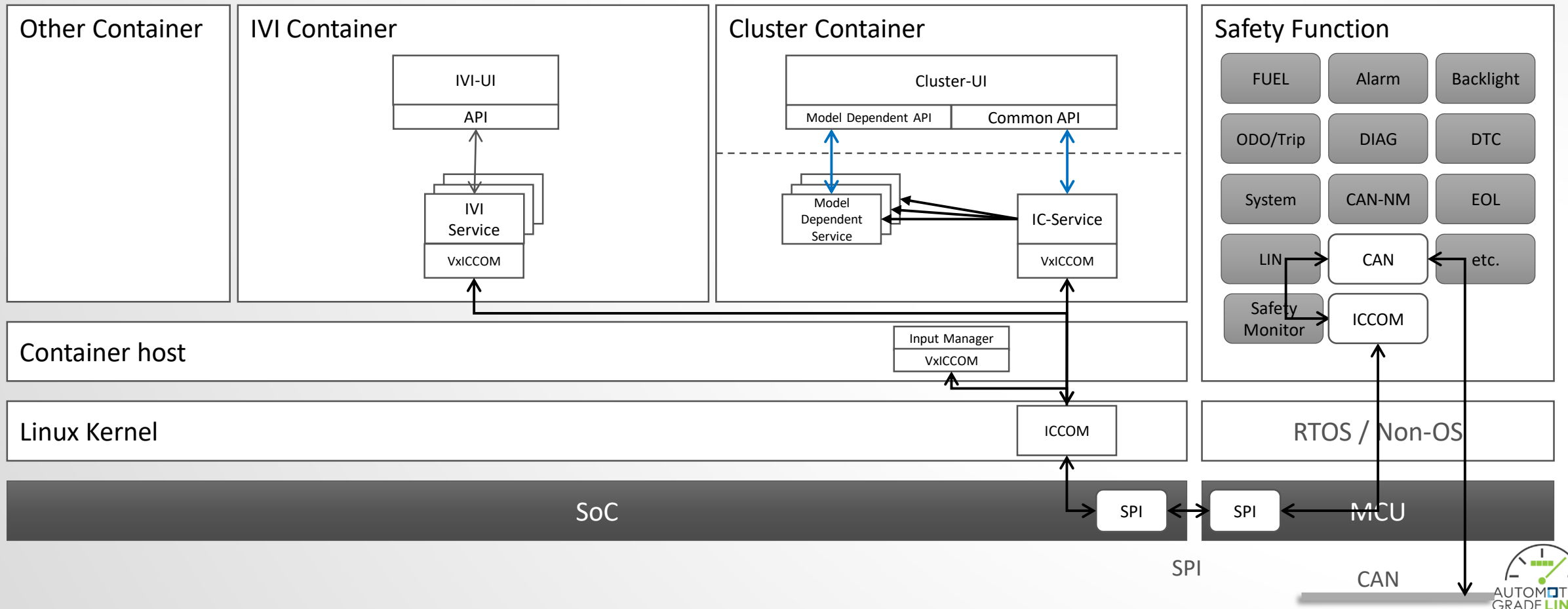
1. ICCOM – e.g. HMI Speed Meter

ICCOM is responsible for vehicle signal handling which transferred from M
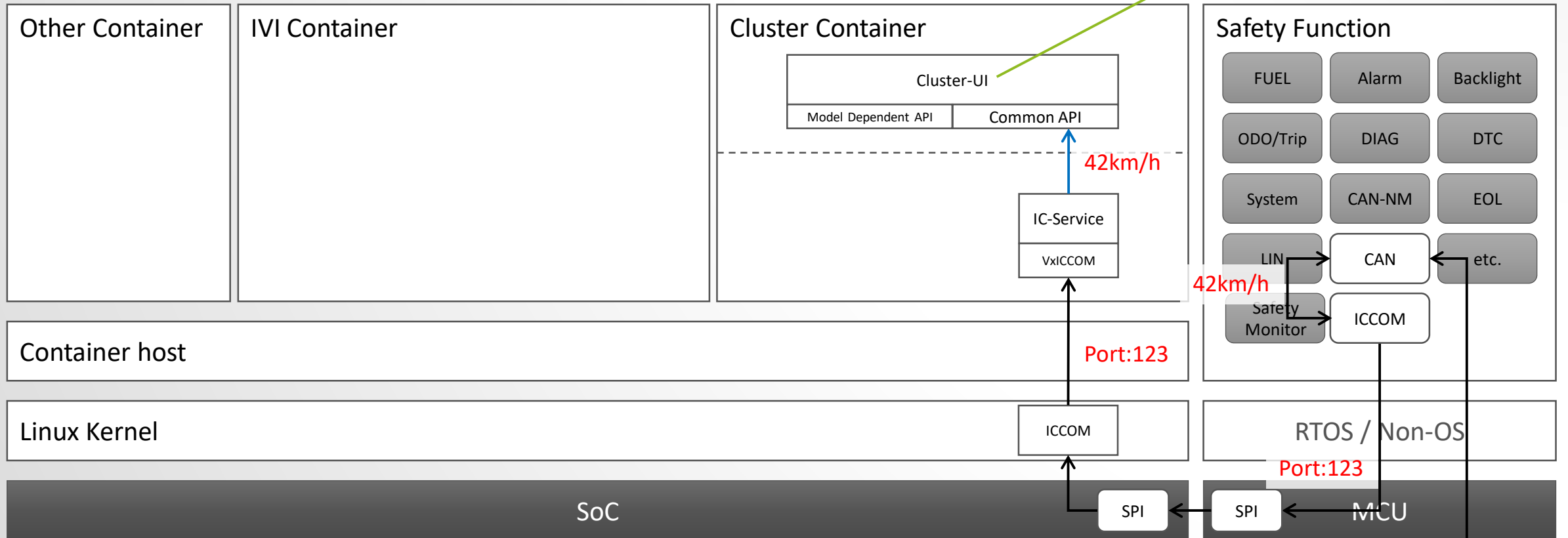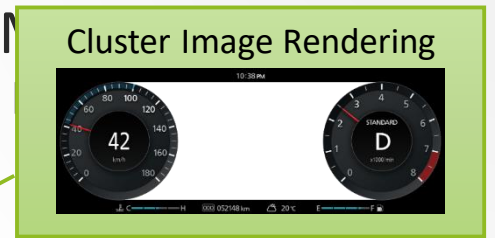- ICCOM socket shall be directly opened in application container, and not in the container
- Keeping advantage of peer to peer communication shall reduce latency and complexity.

Cluster Image Rendering

Other Container	IVI Container	Cluster Container	Safety Function

Cluster Container

Cluster-UI

Model Dependent API	Common API

42km/h

IC-Service

VxICCOM

Safety Function

FUEL	Alarm	Backlight
ODO/Trip	DIAG	DTC
System	CAN-NM	EOL
LIN	CAN	etc.
Safety Monitor	ICCOM	

42km/h

Container host

Port:123

Linux Kernel

ICCOM

RTOS / Non-OS

Port:123

SoC

SPI

SPI

MCU

CAN

Data Flow – 2. Input Manager

Input Manager is responsible for event data handling such as physical input device.

- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered, the event shall be handled by pre-defined priority.
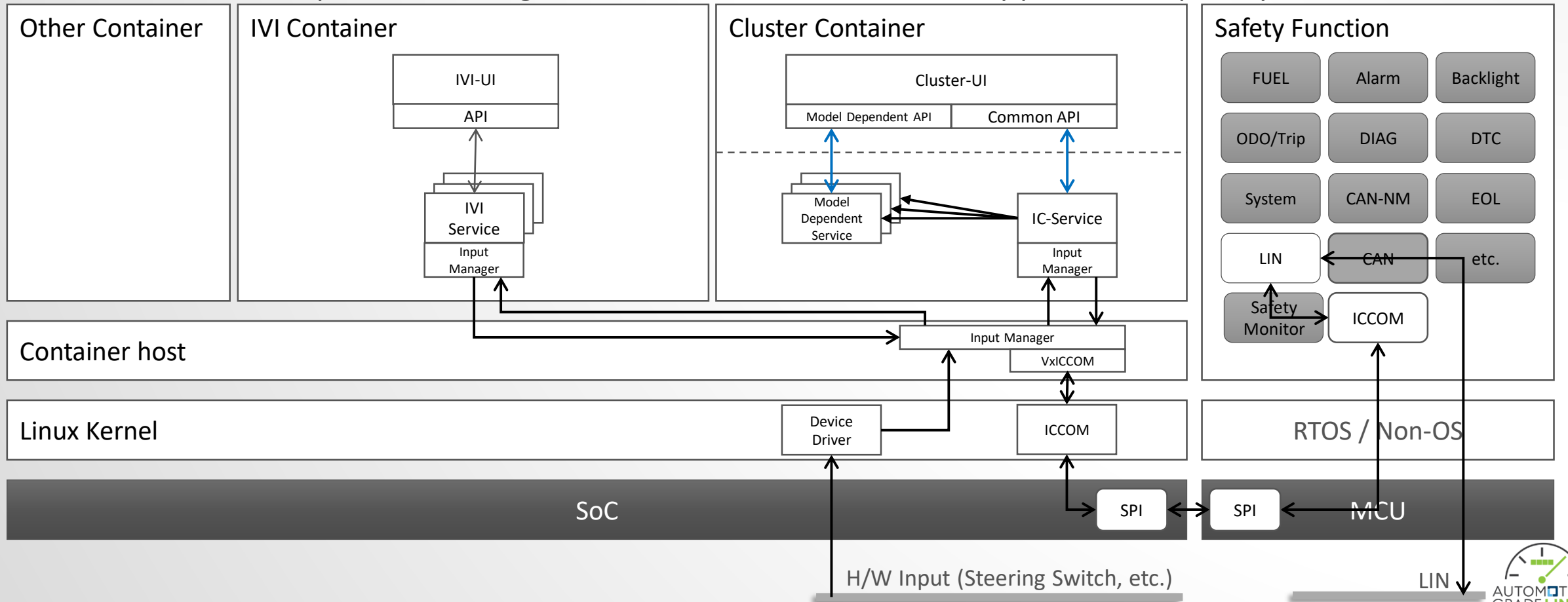
2. Input Manager – e.g. Steering Switch

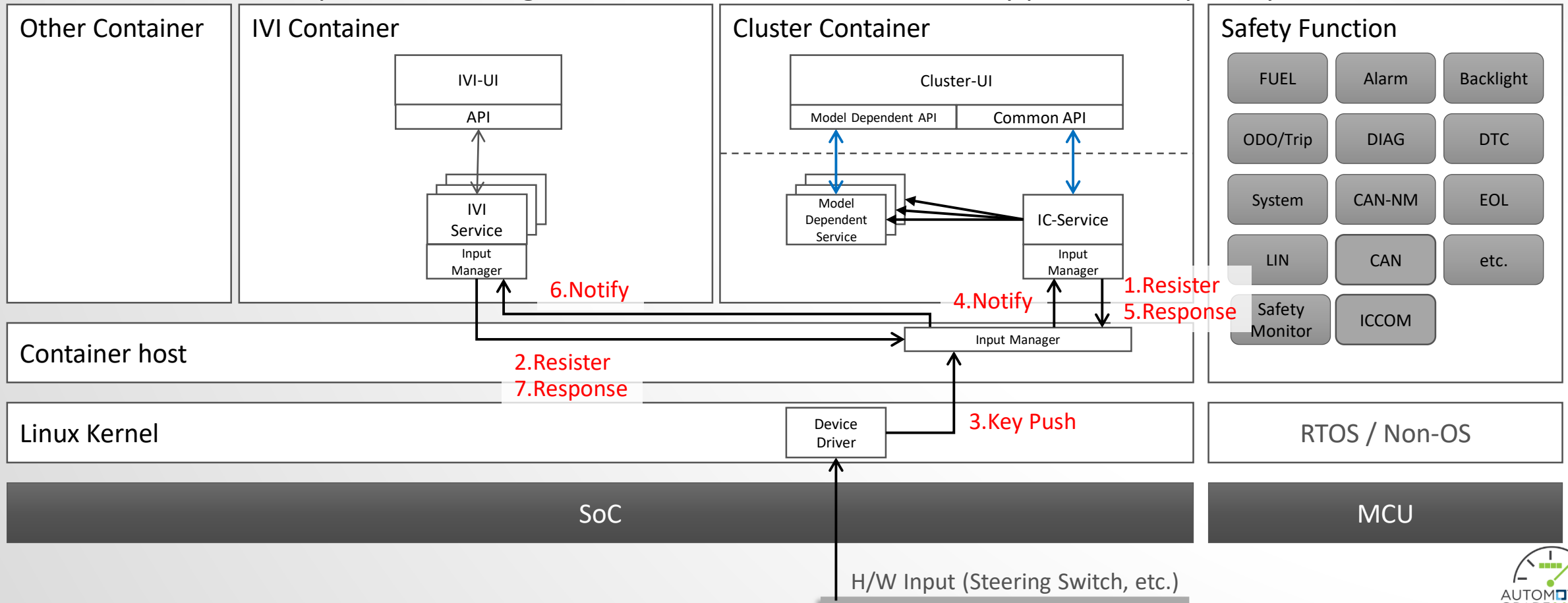Input Manager is responsible for event data handling such as physical input device.

- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered, the event shall be handled by pre-defined priority.

Other Container	IVI Container	Cluster Container	Safety Function

IVI Container
- IVI-UI
- API
- IVI Service
- Input Manager

Cluster Container
- Cluster-UI
- Model Dependent API
- Common API
- Model Dependent Service
- IC-Service
- Input Manager

Safety Function

FUEL	Alarm	Backlight
ODO/Trip	DIAG	DTC
System	CAN-NM	EOL
LIN	CAN	etc.
Safety Monitor	ICCOM	

Container host
- Input Manager

6.Notify
4.Notify
1.Resister
5.Response
2.Resister
7.Response

Linux Kernel
- Device Driver

3.Key Push

RTOS / Non-OS

SoC

MCU

H/W Input (Steering Switch, etc.)

AUTOMOTIVE GRADE LINUX

2. Input Manager – e.g. LIN

Input Manager is responsible for event data handling such as physical input device.
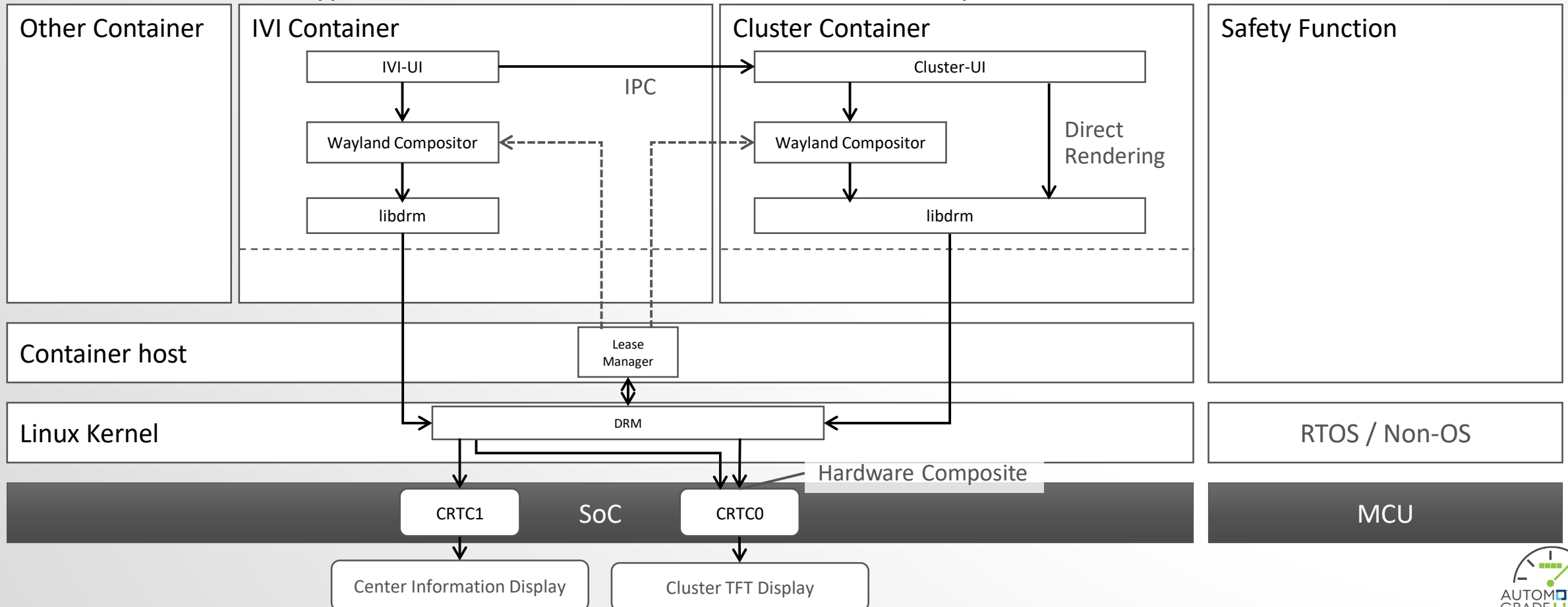
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered, the event shall be handled by pre-defined priority.
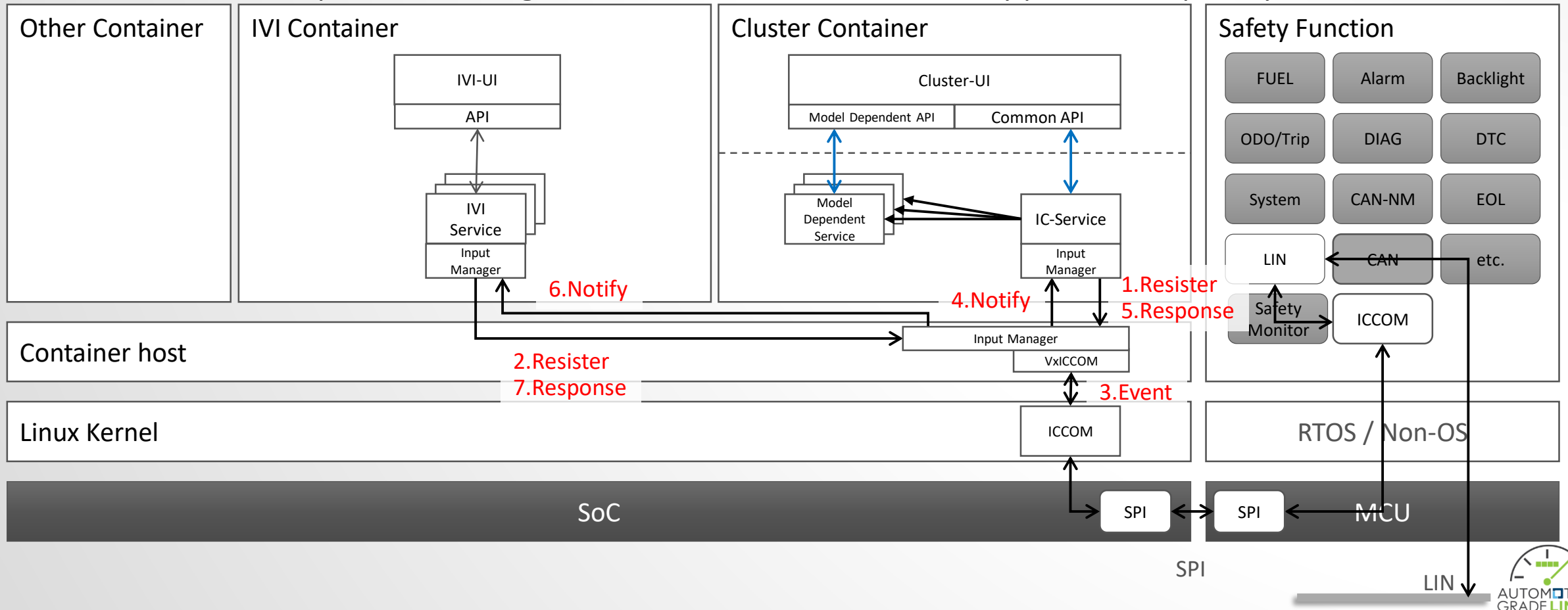
Data Flow – 3. Window Manager

Multiple container DRM sharing shall be done by introducing DRM Lease Manager.
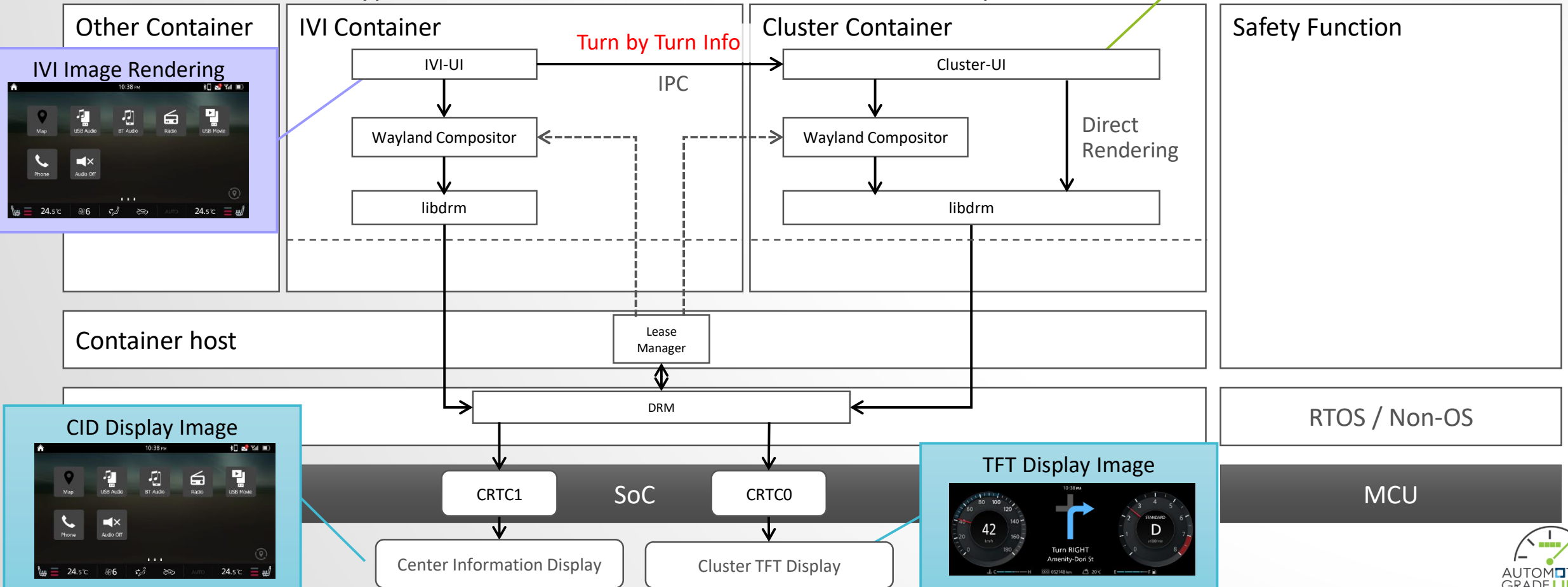
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.

3. Window Manager – e.g. IPC

Multiple container DRM sharing shall be done by introducing DRM Lease Manager.
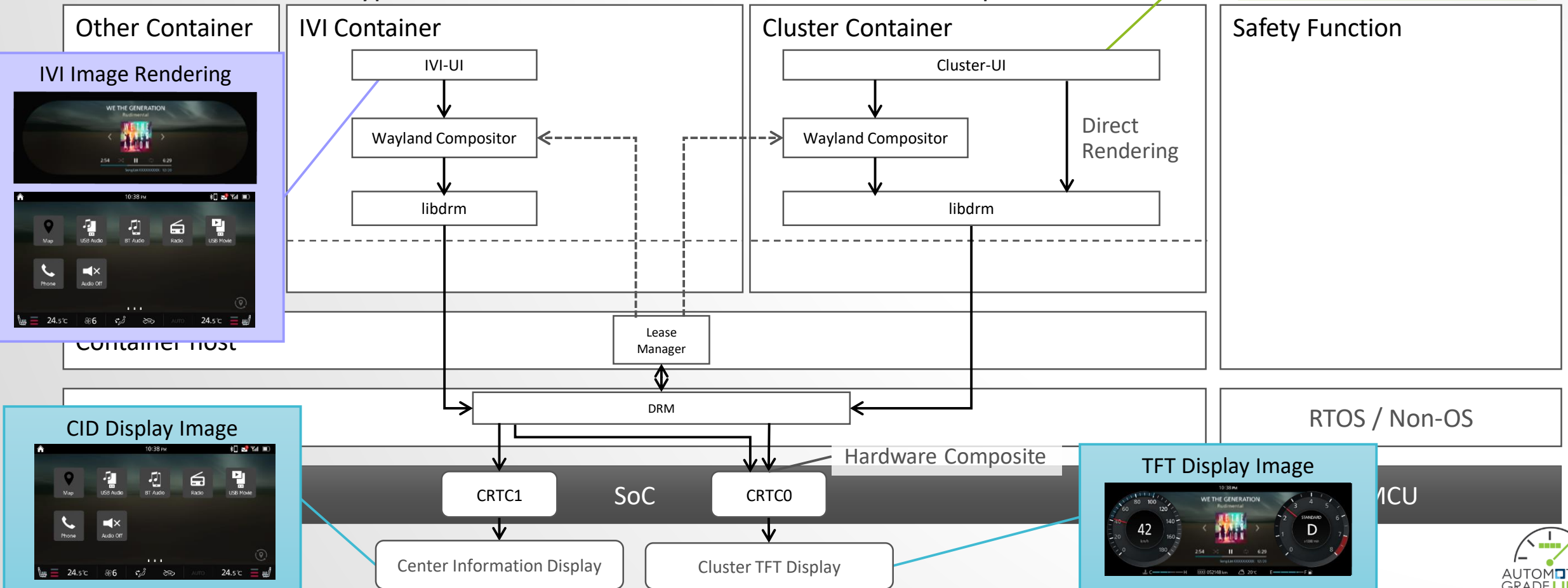
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.

Cluster Image Rendering

Other Container

IVI Image Rendering

IVI Container

IVI-UI → Turn by Turn Info → **Cluster Container** → Cluster-UI

IPC

Wayland Compositor ← --- → Wayland Compositor

Direct Rendering

libdrm → libdrm

Safety Function

Container host

Lease Manager

DRM

RTOS / Non-OS

CID Display Image

CRTC1 — SoC — CRTC0

TFT Display Image

MCU

Center Information Display

Cluster TFT Display

AUTOMOTIVE GRADE LINUX

3. Window Manager – e.g. H/W Composite

Multiple container DRM sharing shall be done by introducing DRM Lease Manager.
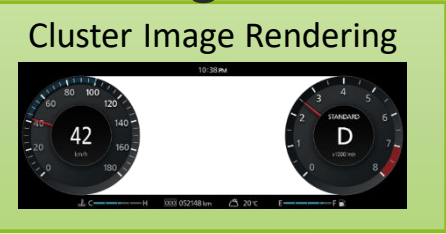
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.

Cluster Image Rendering

IVI Image Rendering

Other Container

IVI Container

- IVI-UI
- Wayland Compositor
- libdrm

Cluster Container

- Cluster-UI
- Wayland Compositor
- libdrm

Direct Rendering

Safety Function

Lease Manager

Container Host

DRM

CID Display Image

CRTC1   SoC   CRTC0

RTOS / Non-OS

Hardware Composite

MCU

TFT Display Image

Center Information Display

Cluster TFT Display

AUTOMOTIVE GRADE LINUX

Data Flow – 4. Sound Manager

- Data from cluster container shall not be mixed with IVI, and device for cluster shall be independent.
- Data mixing/routing shall be done by container host or external chip(like DSP).
- Sound manager in container host will control volume of IVI application by cooperating with sound manager in IVI
- Sound manger in host shall manage system wide policy, and sound manager in IVI shall manage policy in IVI world.
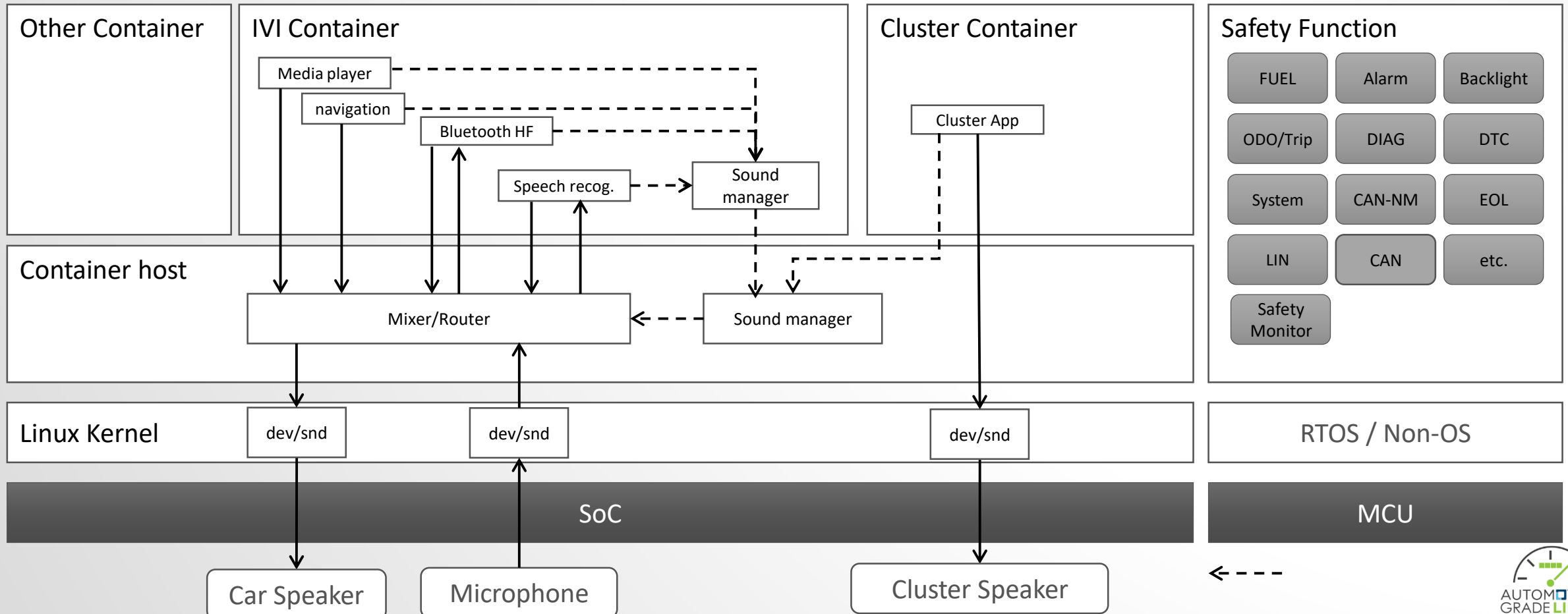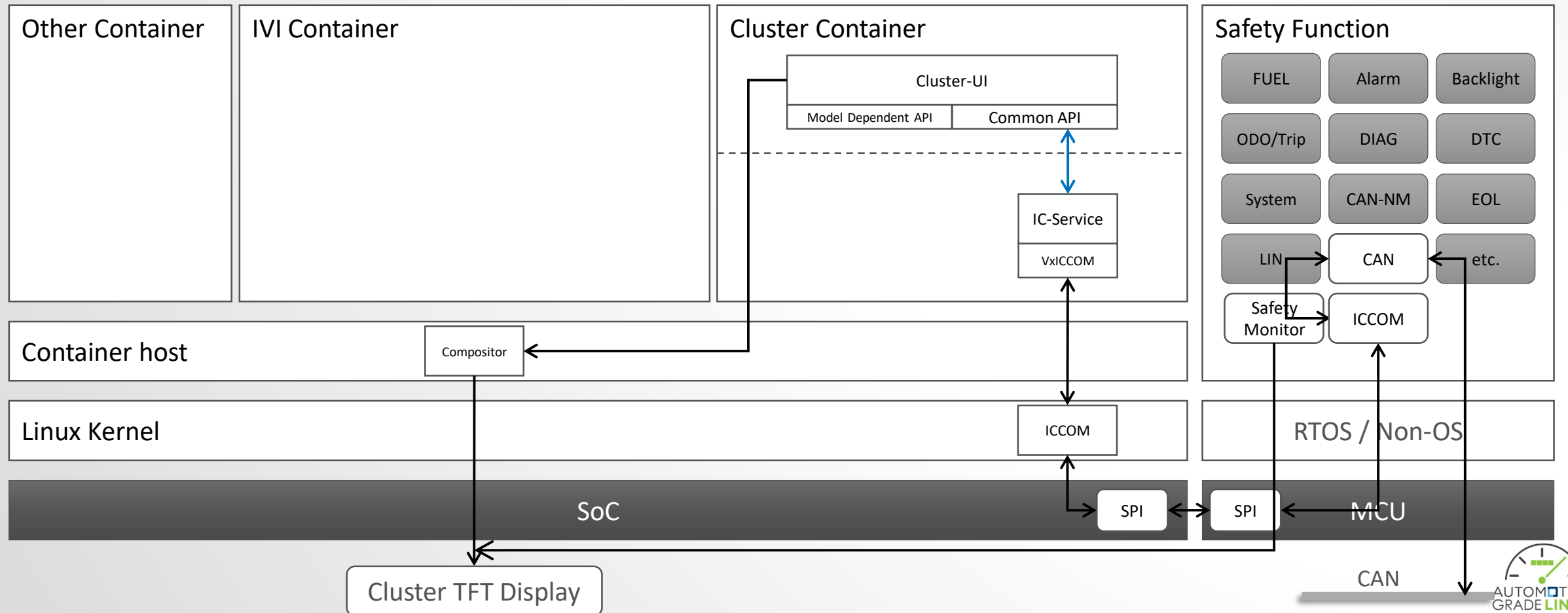
Thank you!

Appendix

IC-Service – Common Function

Data flow example for OEM/Tier-1 common function.

- e.g. | SPEED | TACHO | TEMP | Shift Position | Telltale |

IC-Service – OEM/Tier-1 Dependent

Data flow example for common function partially includes OEM/Tier logic.

- e.g.

| Fuel Consumption | ACC | Energy Flow | ECO | Warning | Traffic Sign | Turn by Turn | User Customize | Maintenance |