Instrument Cluster EG
Software Architecture Overview

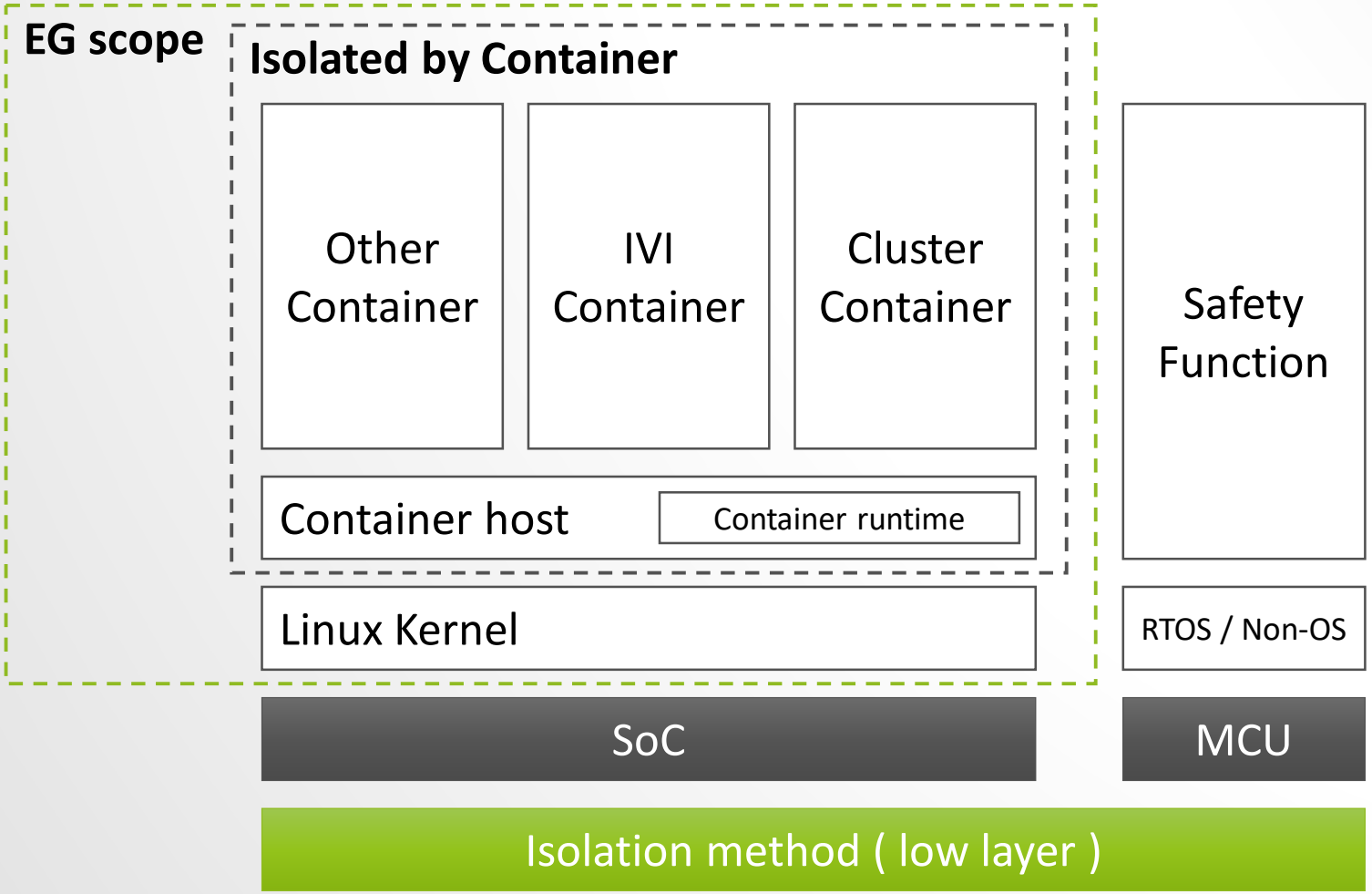
IC-EG Japan development team
29.June.2020

Contents

1. Container Architecture – Overview
2. Function Block Assignment
3. Cluster Container – Overview
4. IC-Service Interface
5. IC-EG Scope
6. Data Flow – Overview
    1. ICCOM
    2. Input Manager
    3. Window Manager
    4. Sound Manager

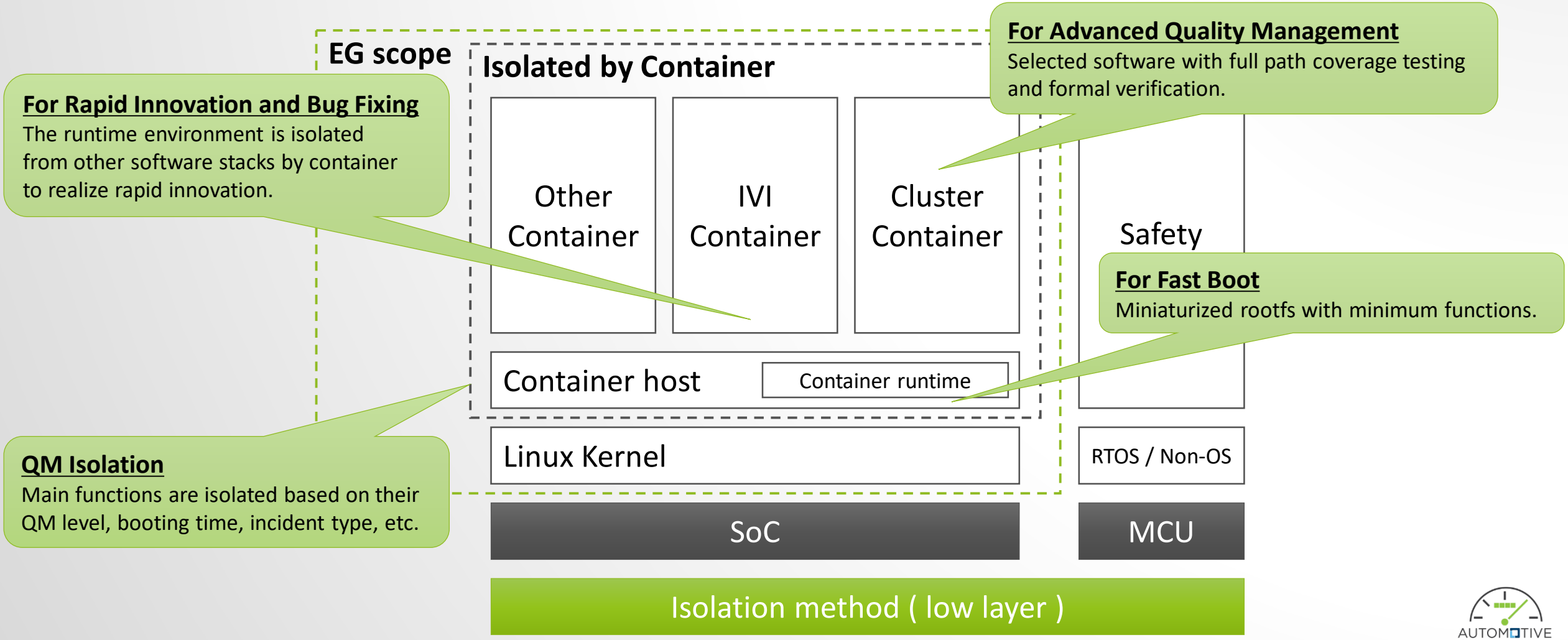
Container Architecture – Overview

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



Container Architecture – Overview

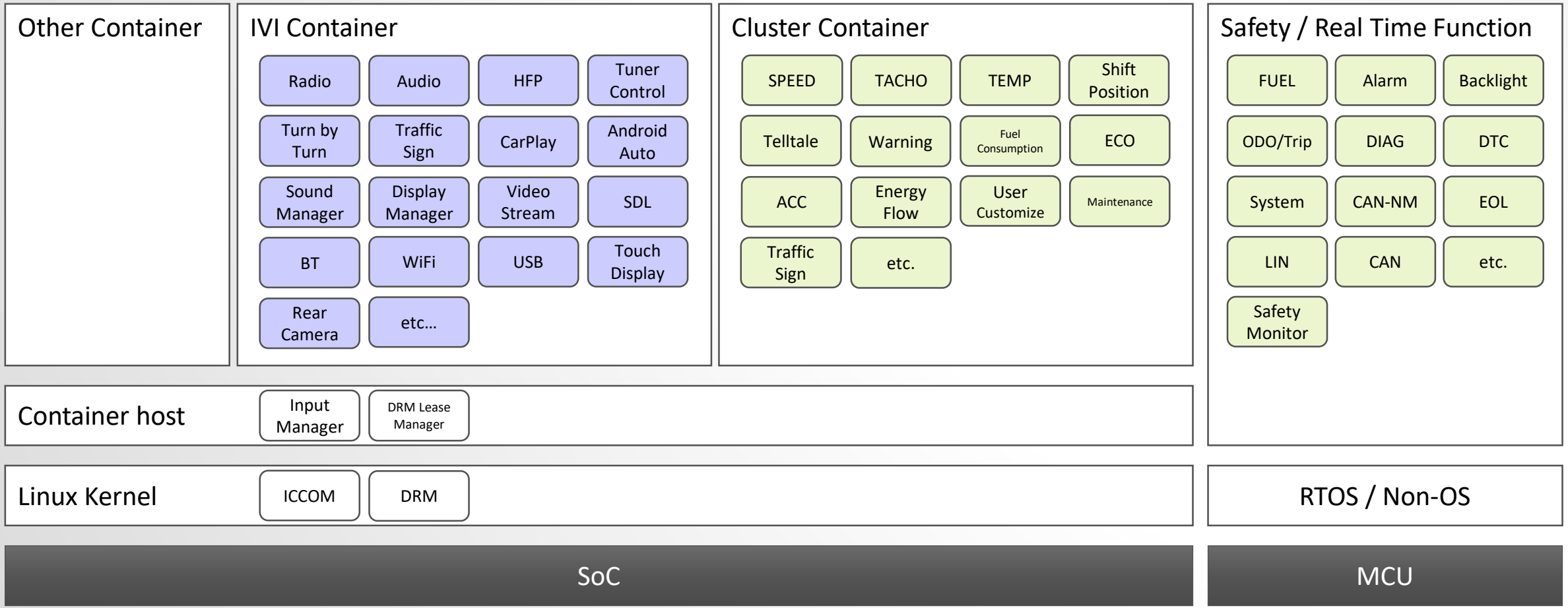
Cluster Function and IVI Function shall be separated by Linux Container Technology in order to achieve QM isolation.



Function Block Assignment

Safety monitoring and real time function which includes device access shall be assigned outside of AGL.

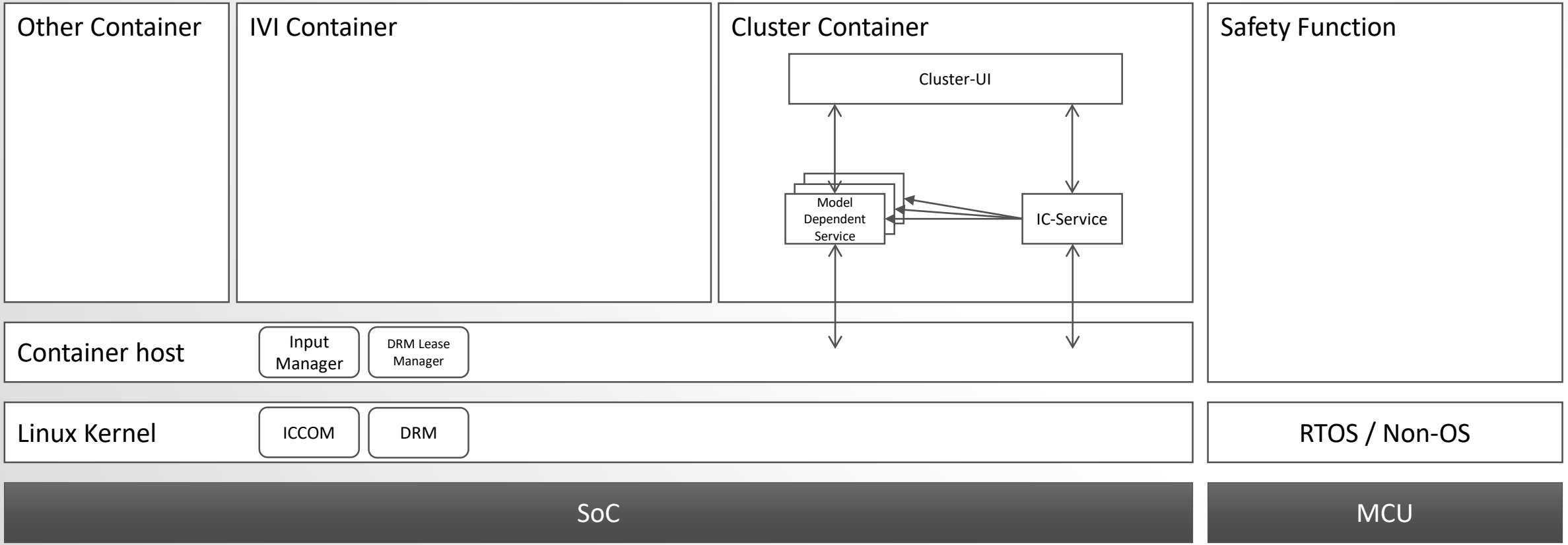
- All of the other cluster function shall be assigned onto the cluster container.



Cluster Container – Overview

Cluster container shall consist of IC-Service and Cluster-UI component.

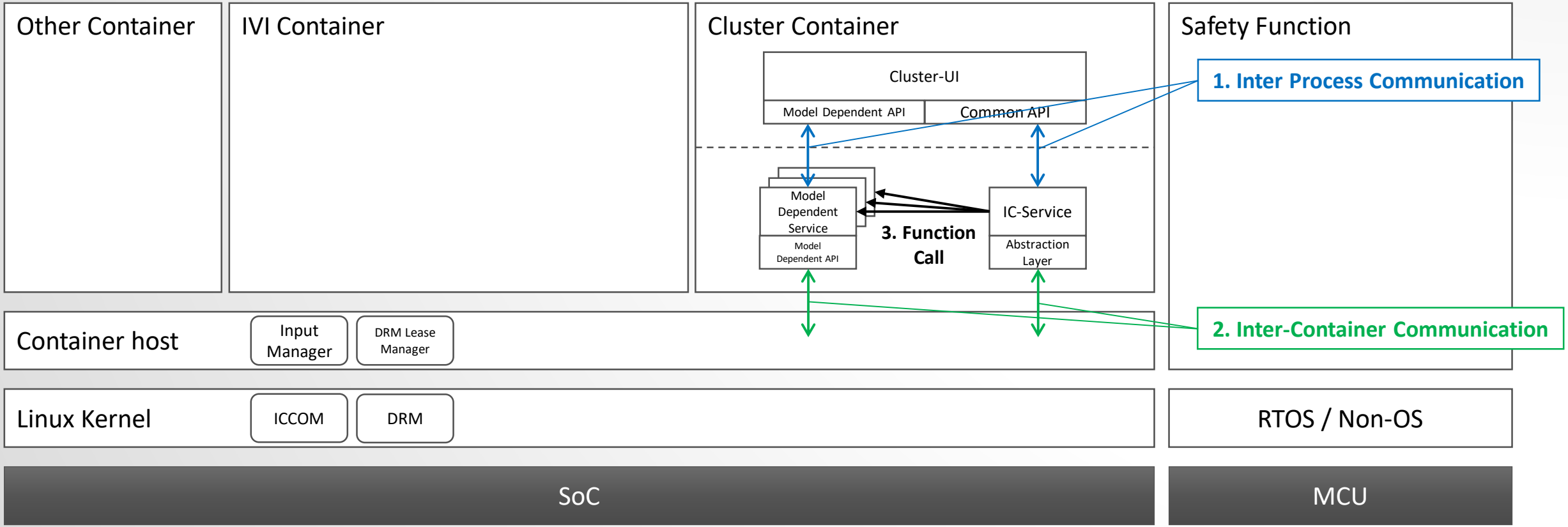
- IC-Service shall consist of a function logic.
- Cluster-UI shall consist of an UI state machine and assets.
- IC-Service shall be separated by a model dependency.



IC-Service Interface

IC-Service shall consist of the following three interface.

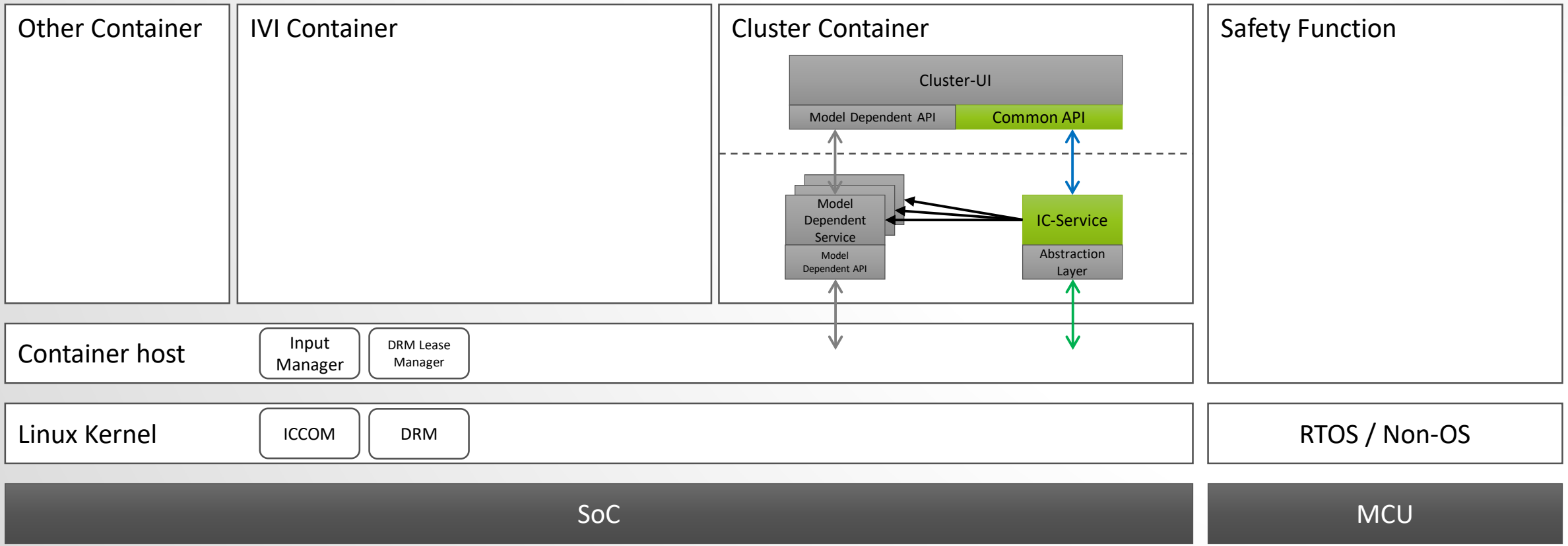
1. Cluster-UI shall be defined a separated process. → **Inter Process Communication**
2. Model dependent service shall be called from IC-Service as a common interface. → **Function Call**
3. IC-Service shall communicate with another container or container host. → **Inter Container Communication**



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.



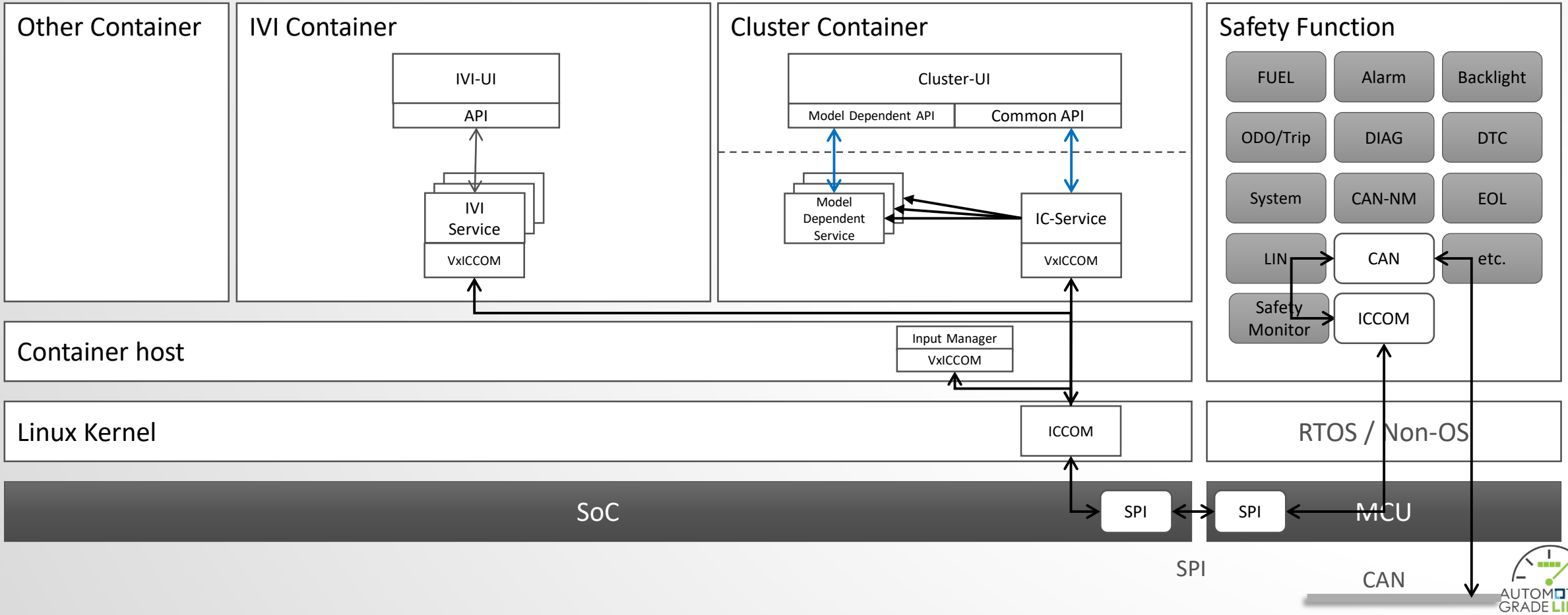
Contents

1. Container Architecture – Overview
2. Function Block Assignment
3. Cluster Container – Overview
4. IC-Service Interface
5. IC-EG Scope
6. Data Flow – Overview
    1. ICCOM
    2. Input Manager
    3. Window Manager
    4. Sound Manager

Data Flow – 1. ICCOM

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

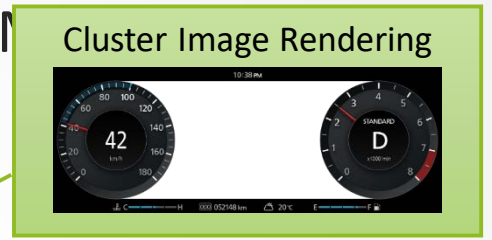
- ICCOM socket shall be directly opened in application container, and not in the container host.
- Keeping advantage of peer to peer communication shall reduce latency and complexity.



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.



Other Container

IVI Container

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.

42km/h

Safety Monitor

ICCOM

Container host

Port:123

Linux Kernel

ICCOM

RTOS / Non-OS

Port:123

SoC

SPI

SPI

MCU

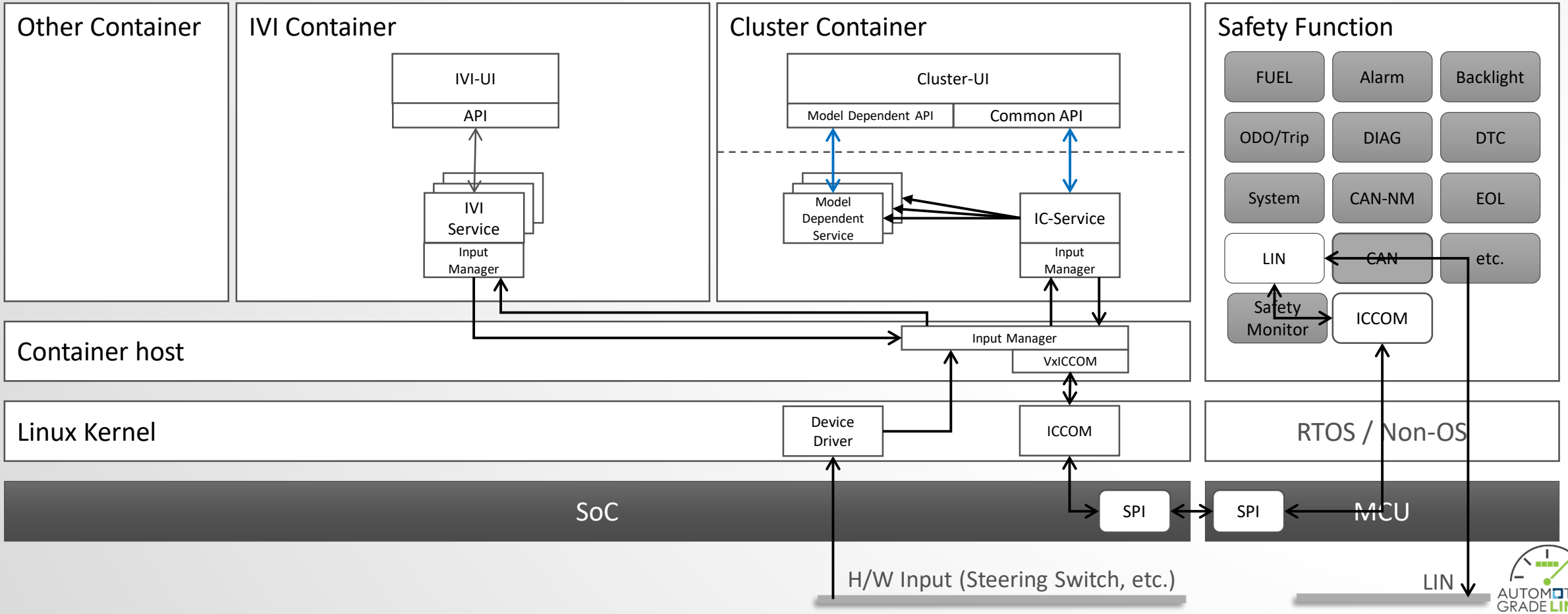
CAN

AUTOMOTIVE GRADE LINUX

Data Flow – 2. Input Manager

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

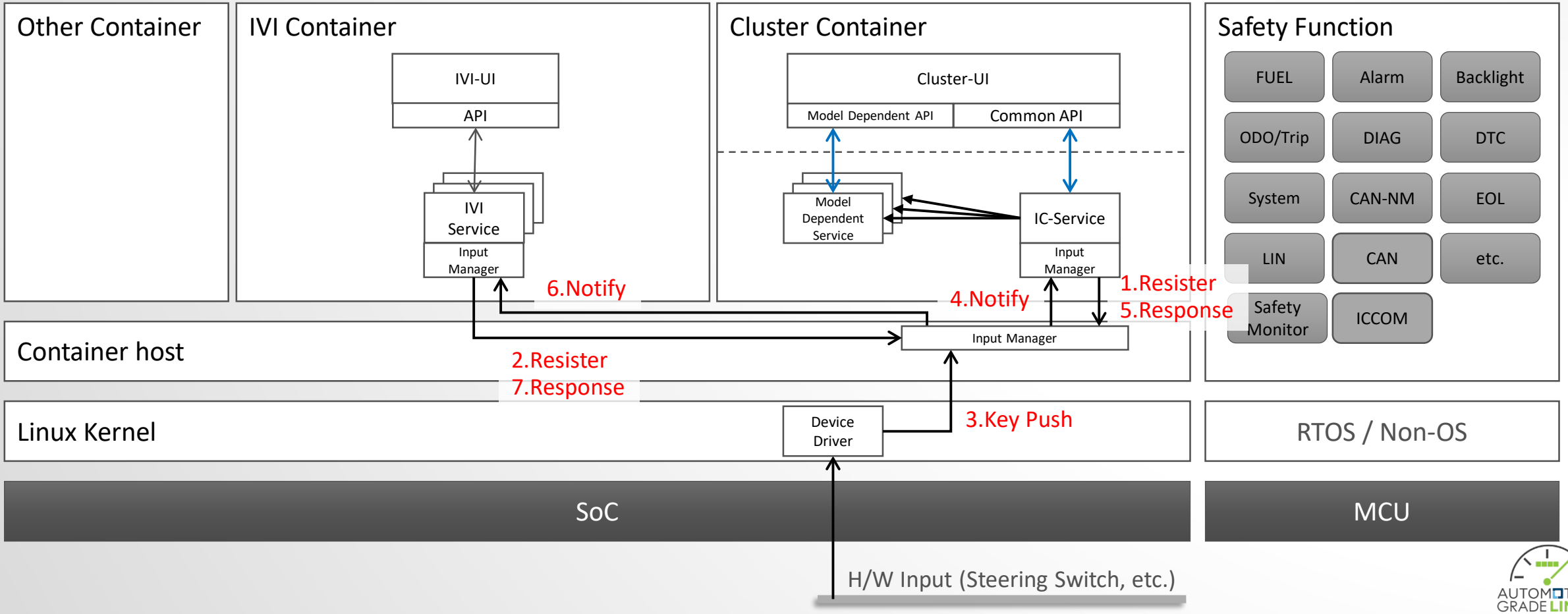
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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

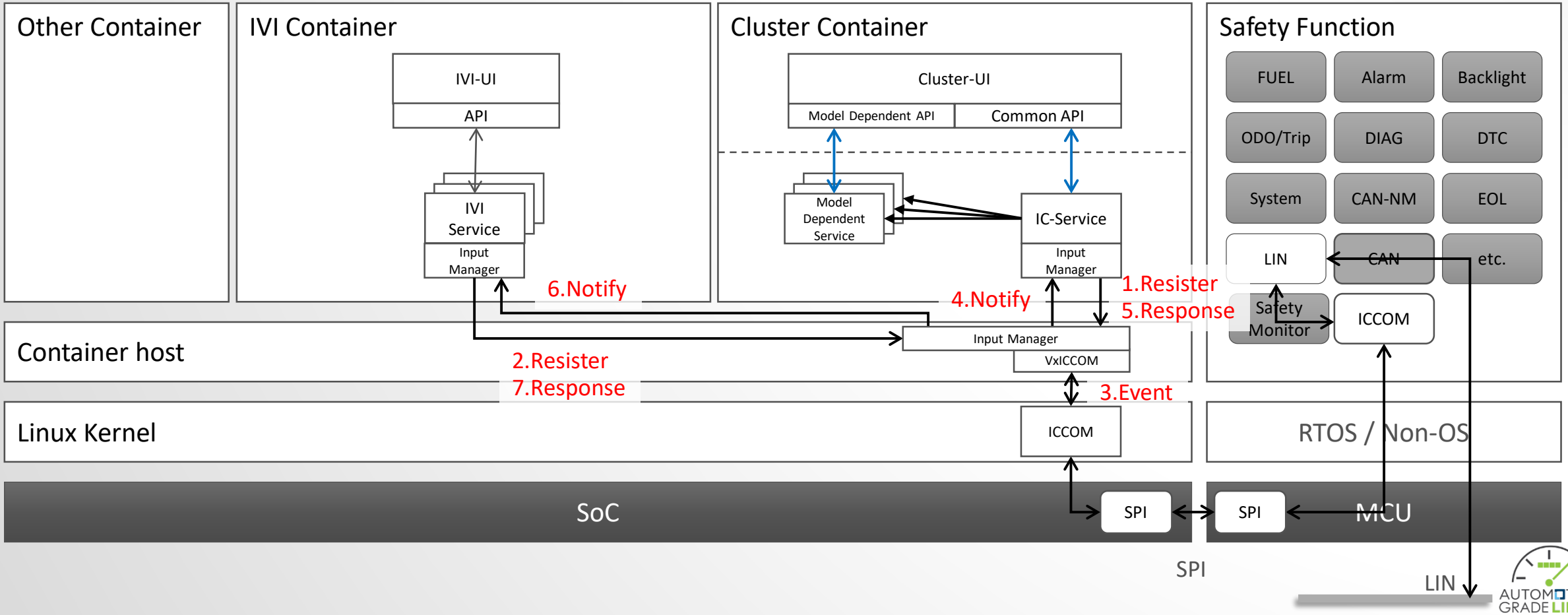
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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



2. Input Manager – e.g. LIN

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

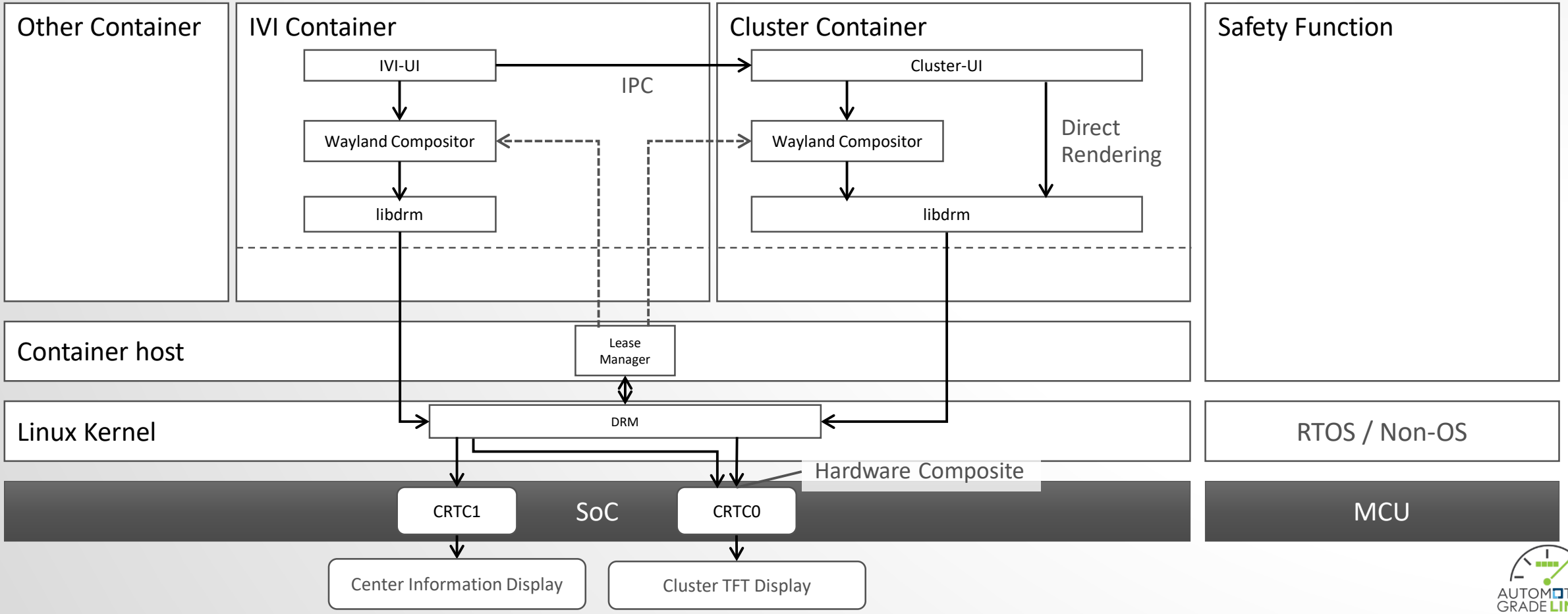
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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

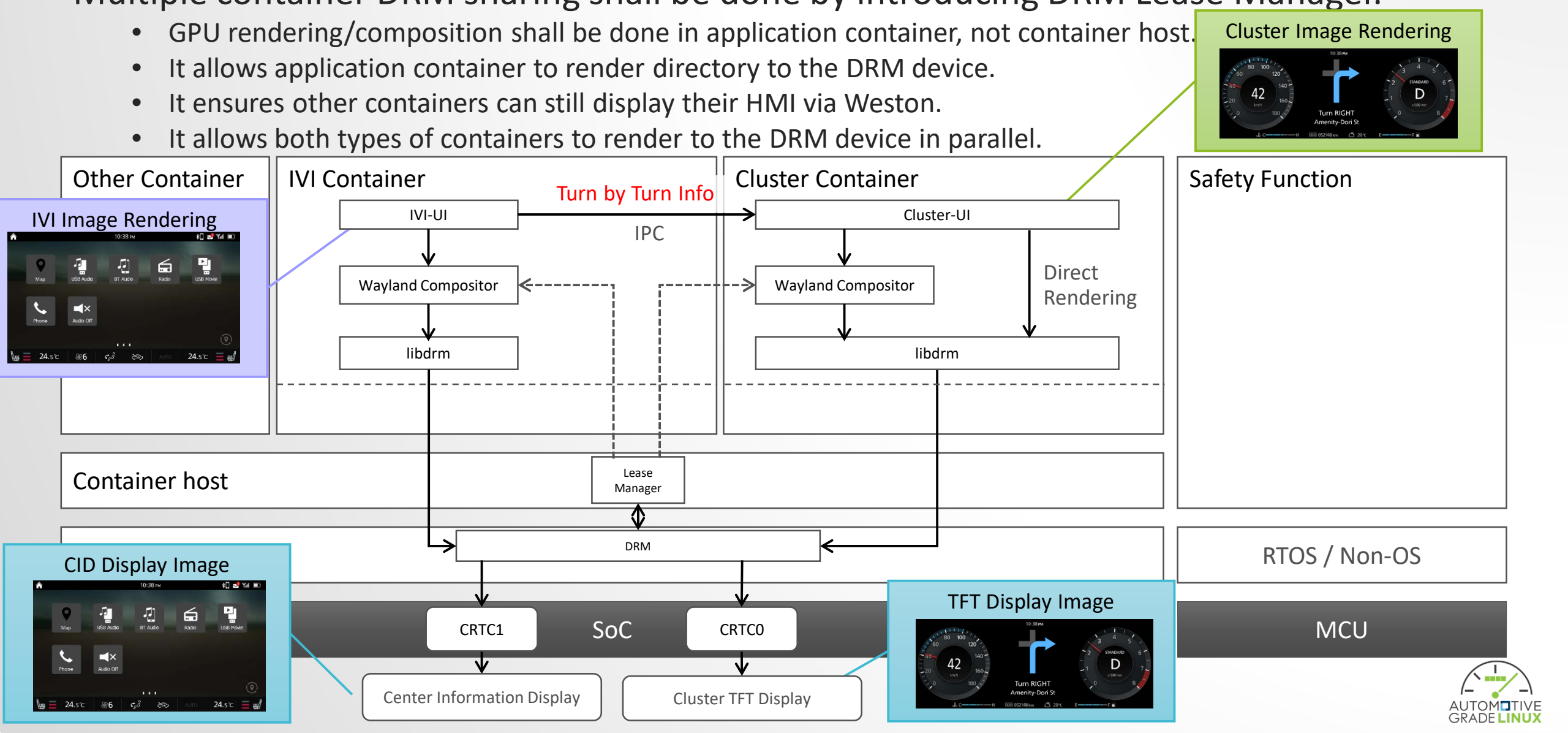
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.



3. Window Manager – e.g. IPC

Multiple container DRM sharing shall be done by introducing DRM Lease Manager.

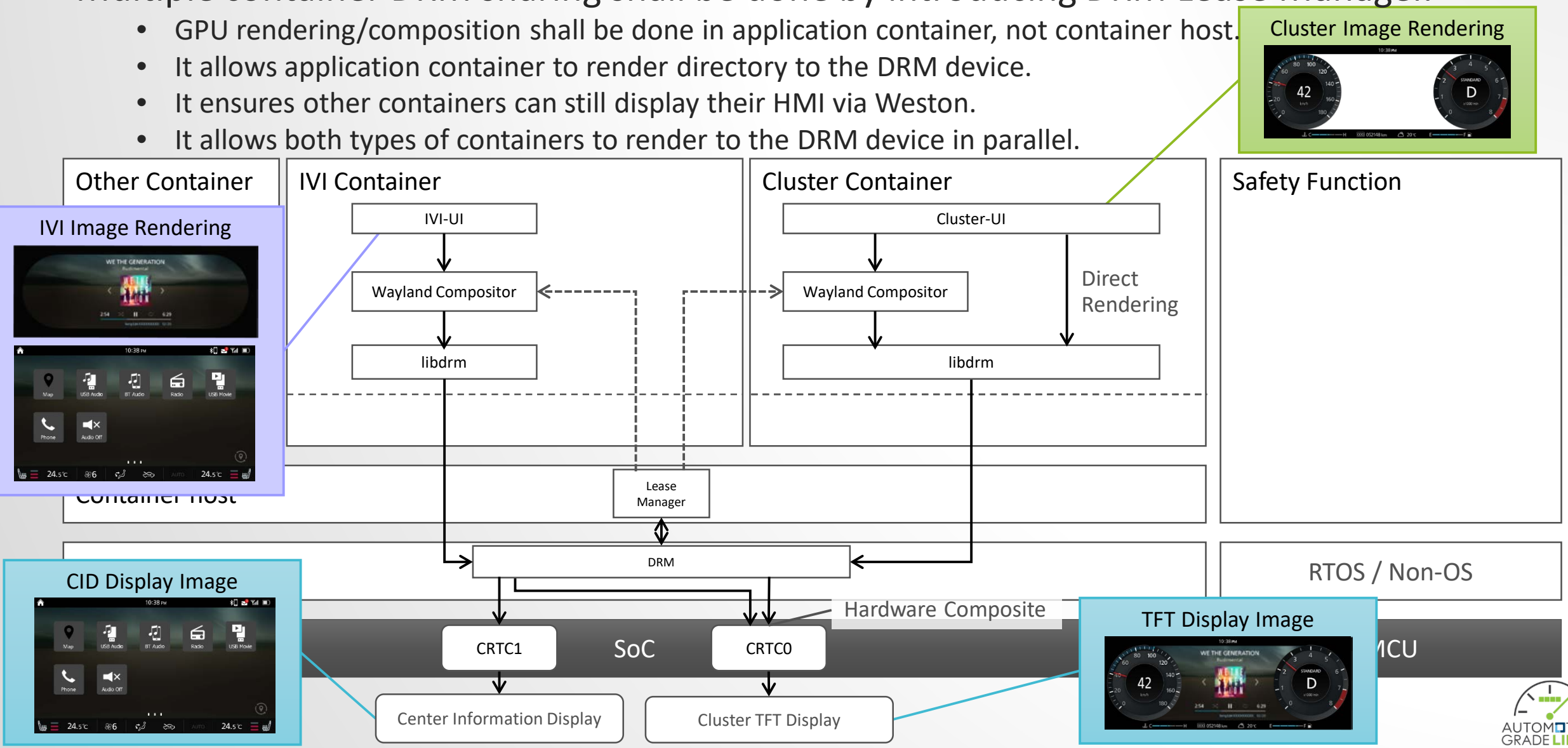
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directory to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.



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

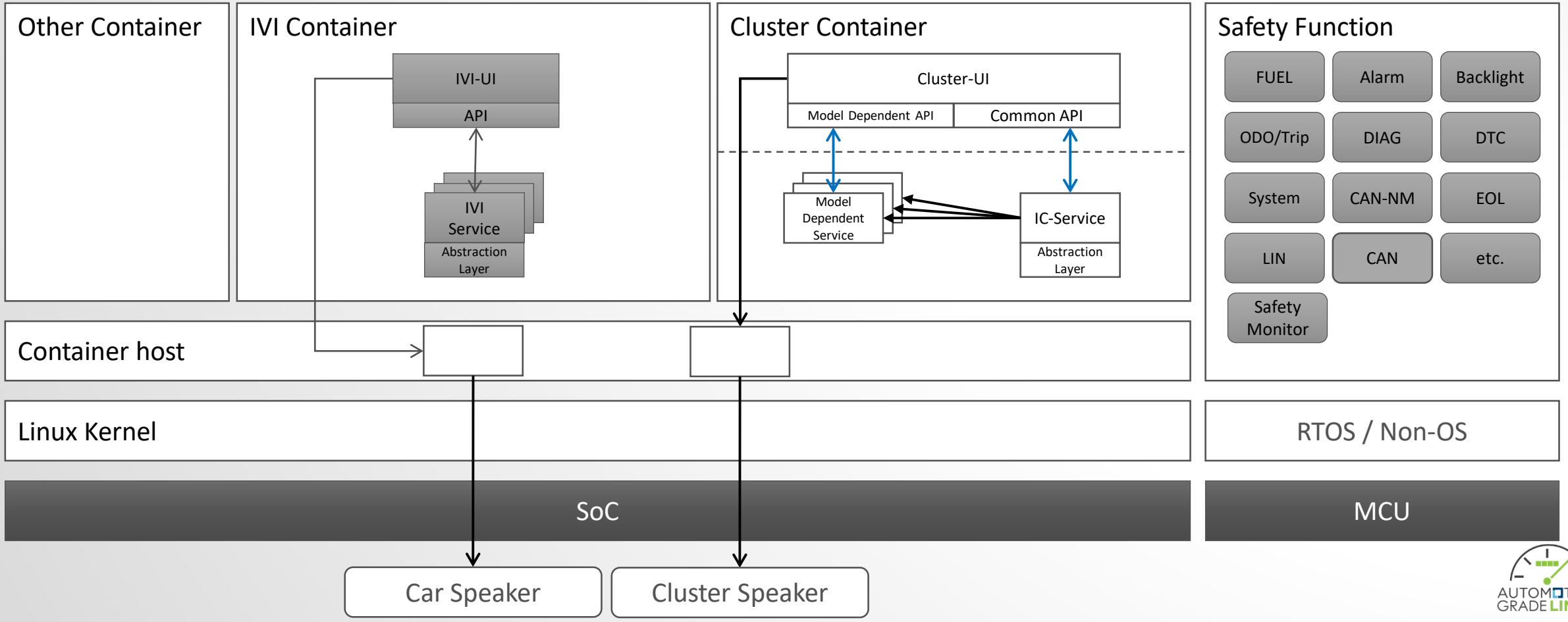
Multiple container DRM sharing shall be done by introducing DRM Lease Manager.

- 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.



Data Flow – 4. Sound Manager

Will be updated later…



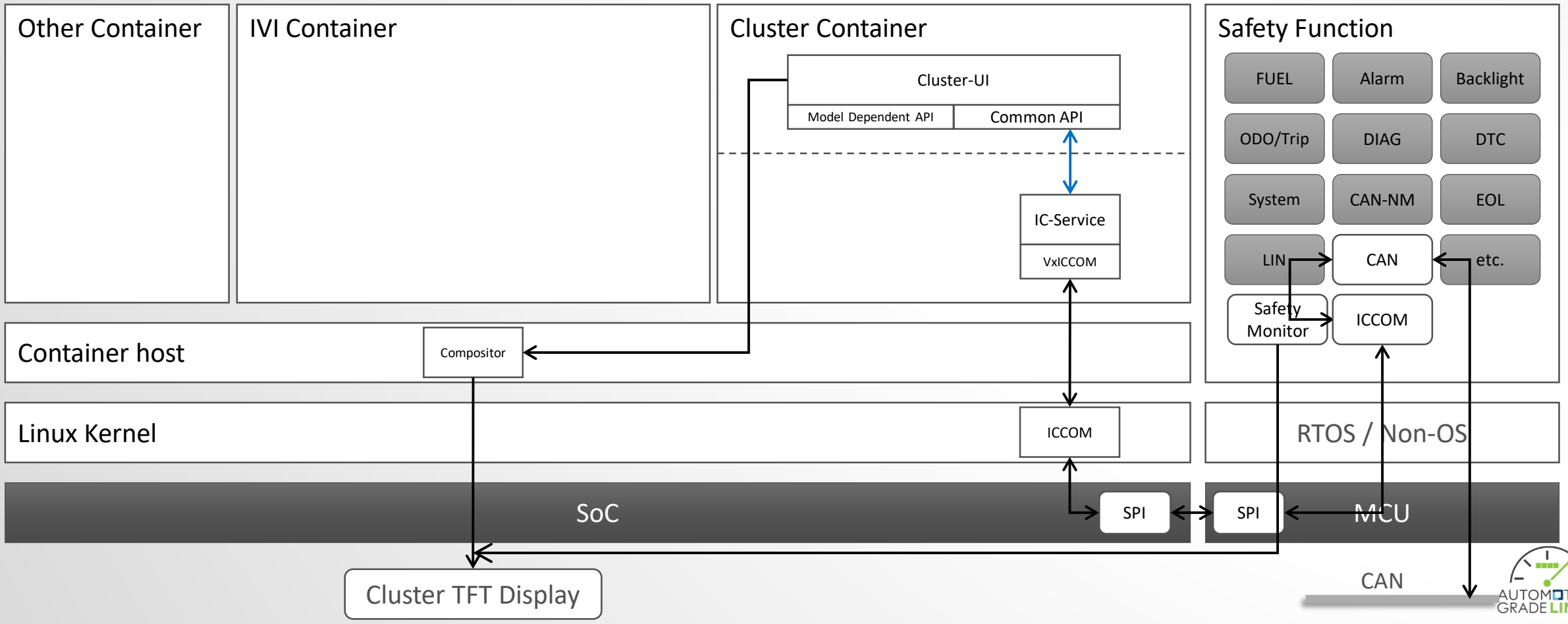
Thank you!

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