Instrument Cluster EG
Software Architecture Overview
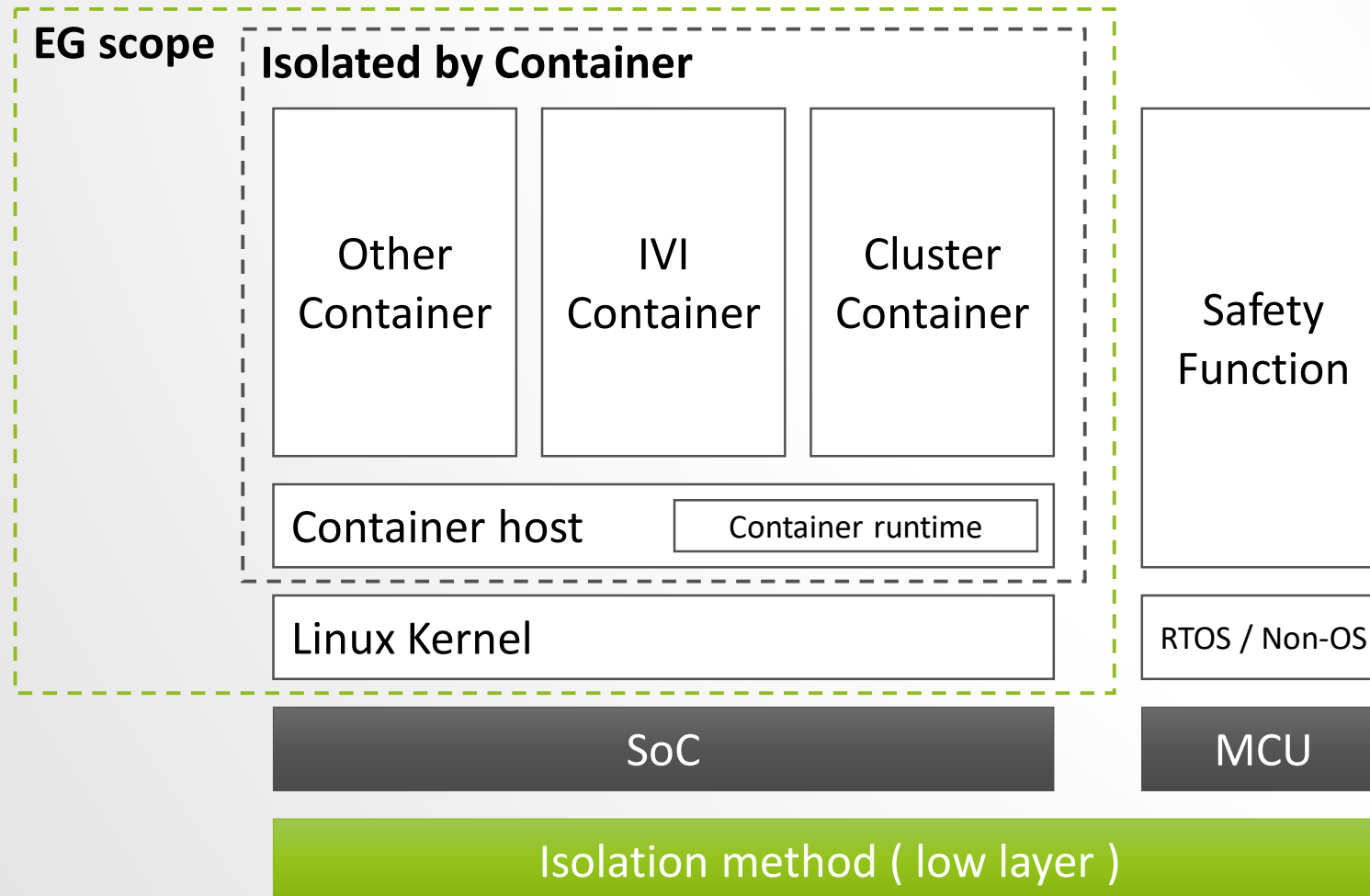
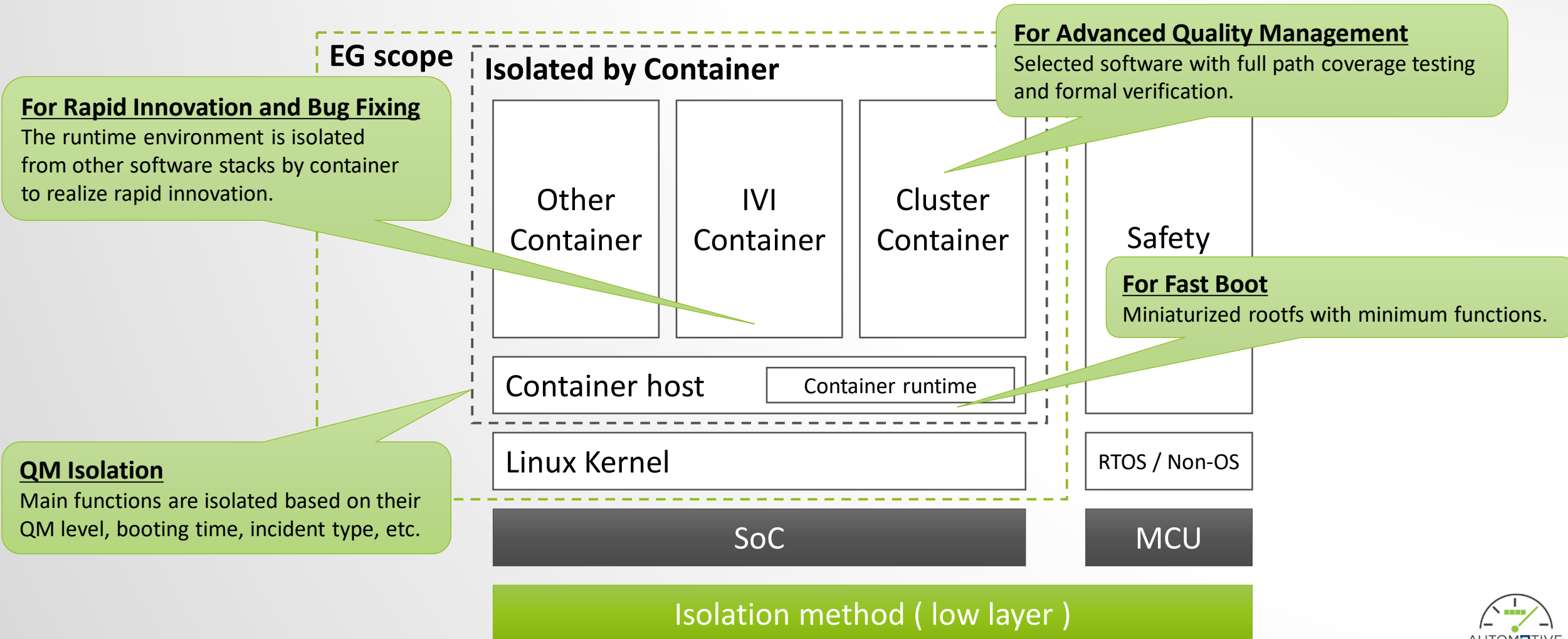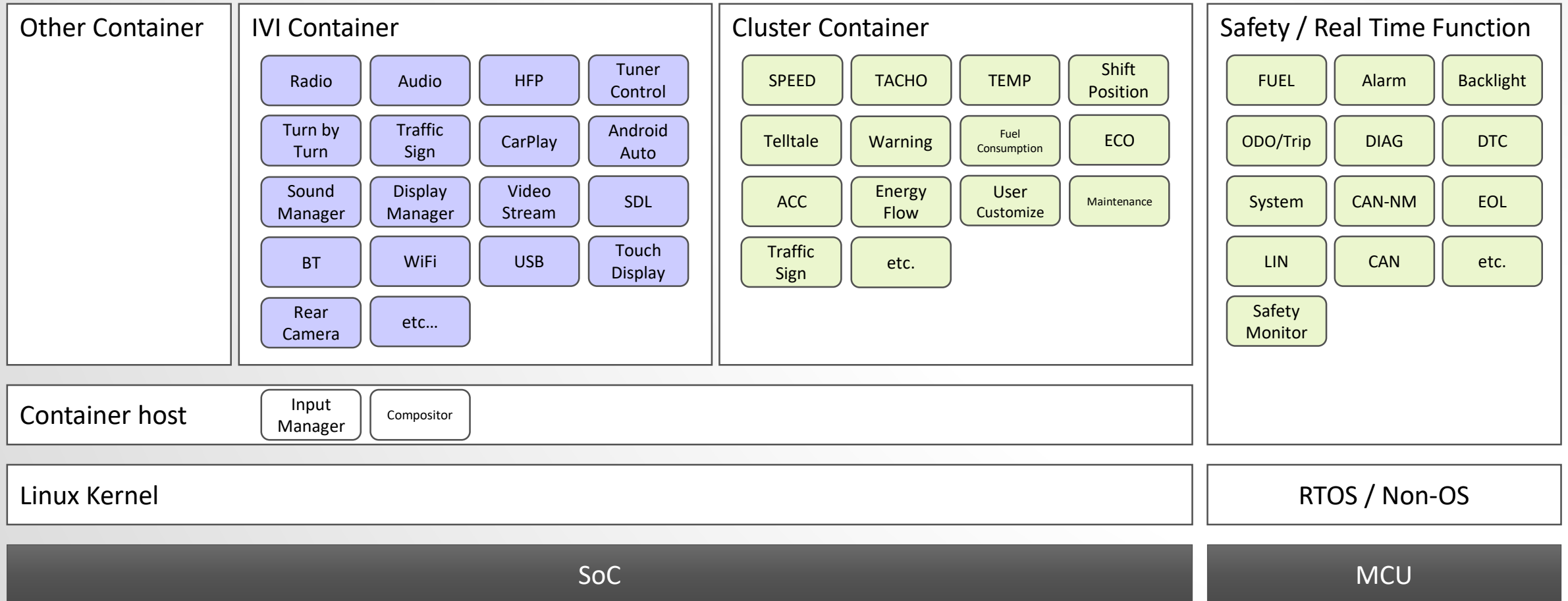IC-EG Japan development team

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)

AUTOMOTIVE GRADE LINUX
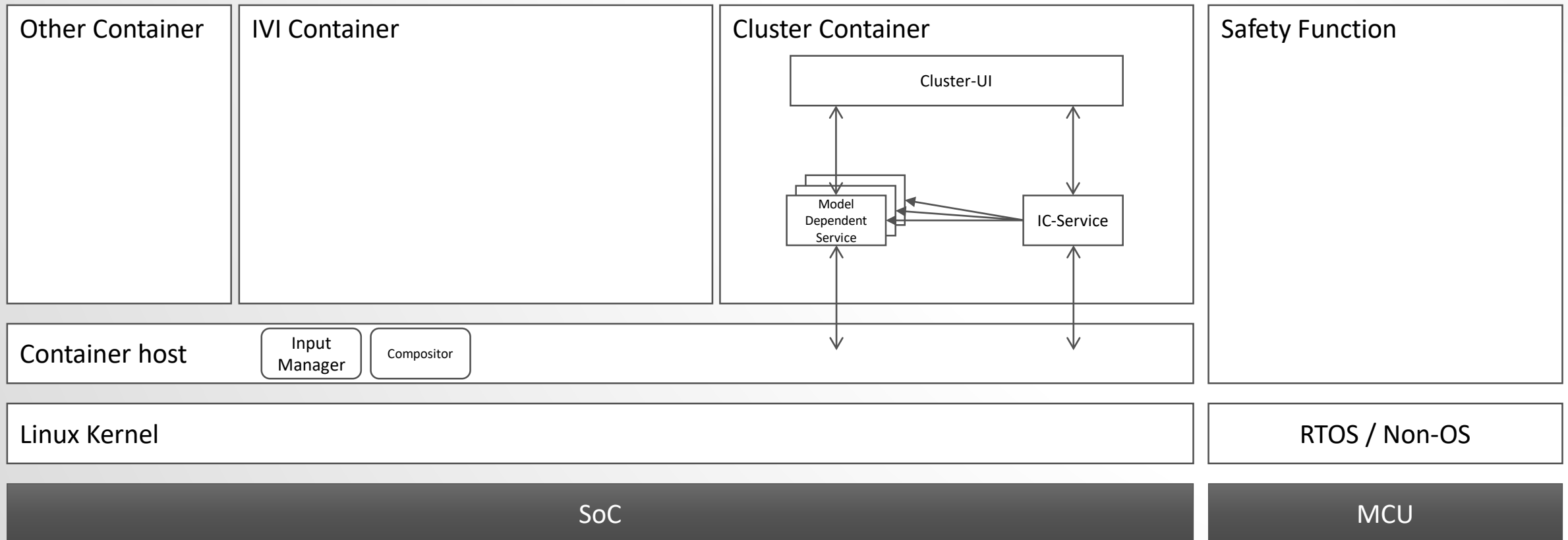
Function Block Assignment

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	Compositor

Linux Kernel	RTOS / Non-OS

SoC	MCU

AUTOMOTIVE GRADE LINUX

Cluster Container – Overview

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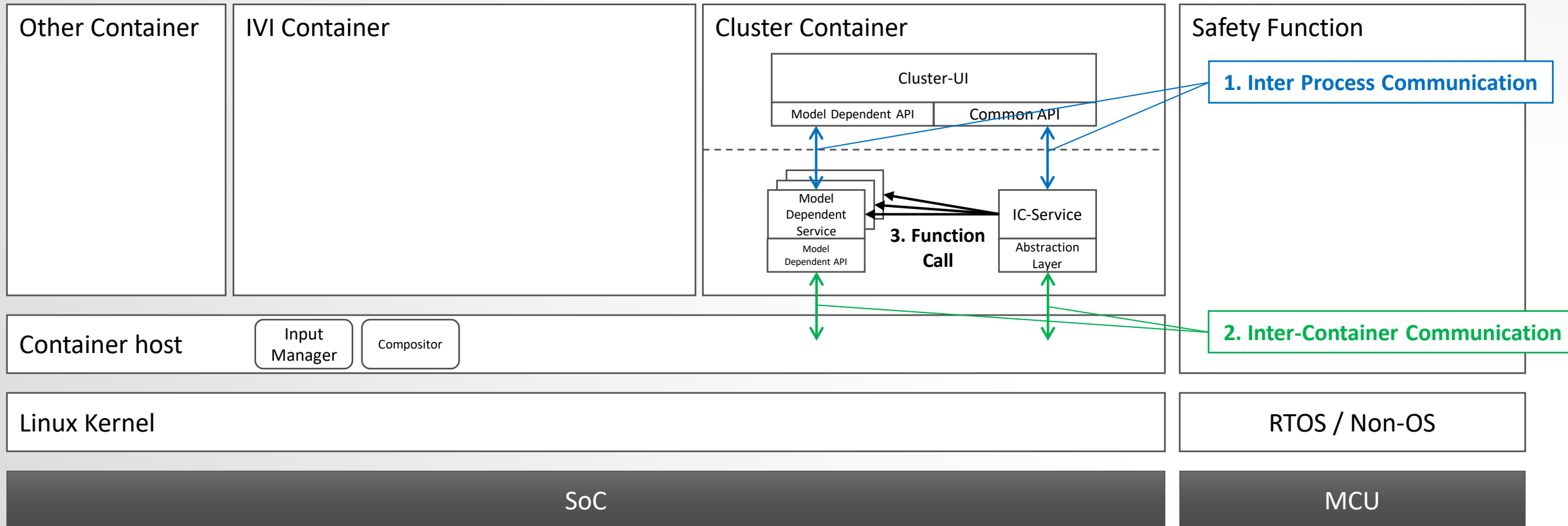
- IC-Service shall consist of a function logic.
- Cluster-UI shall consist of an UI state machine and assets.
- IC-Service shall be separated by a model dependency.

IC-Service Interface

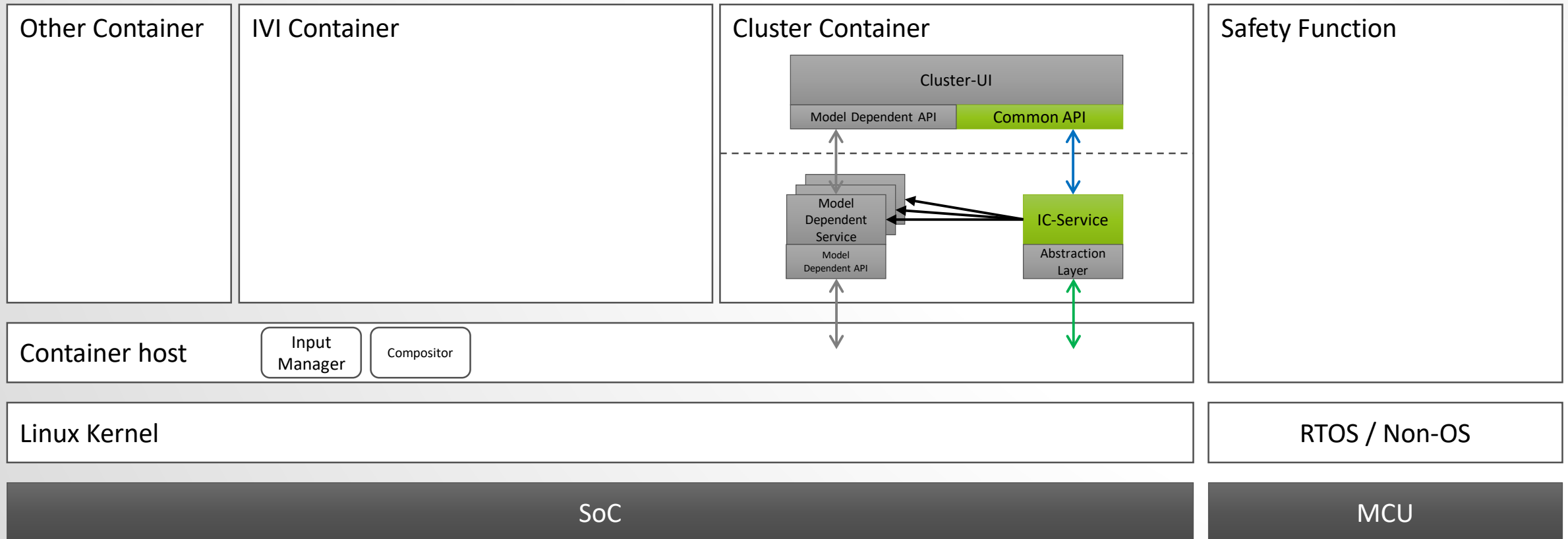IC-Service shall consist of the following three interface.

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

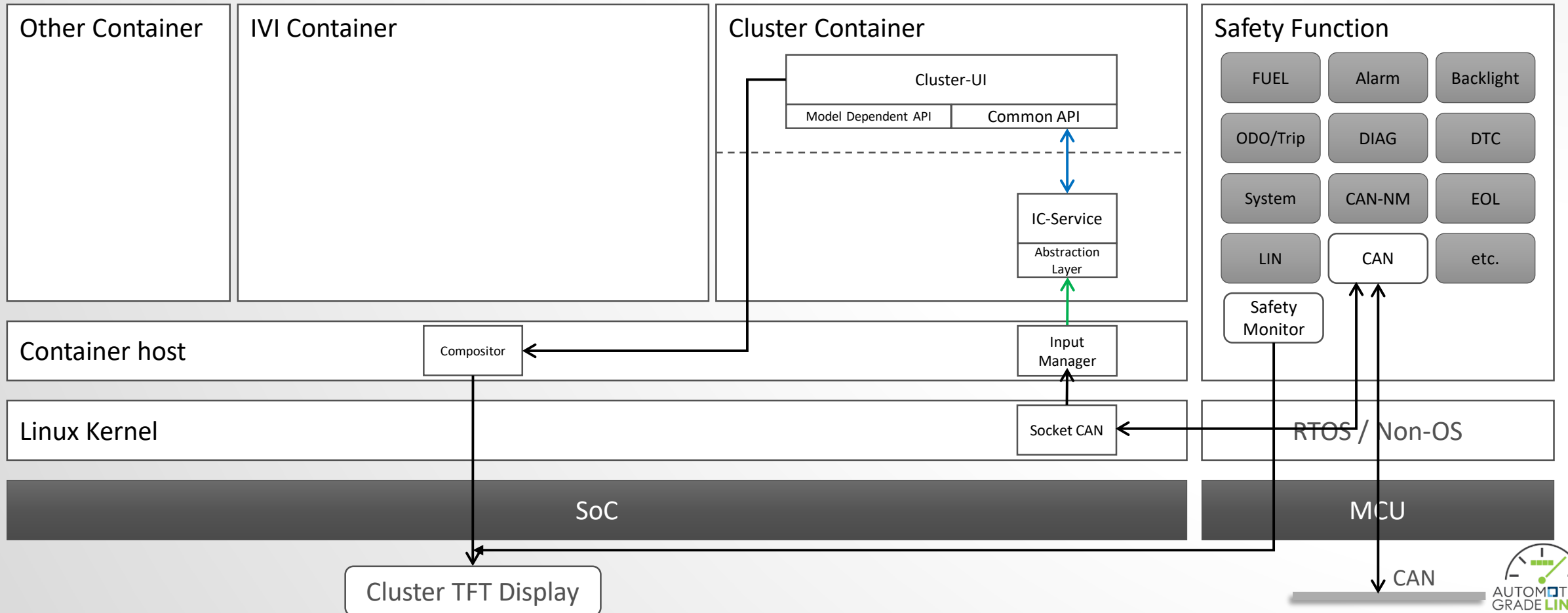IC-Service logic and common API shall be fully provided by IC-EG.

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

Data Flow – 1. Common Function

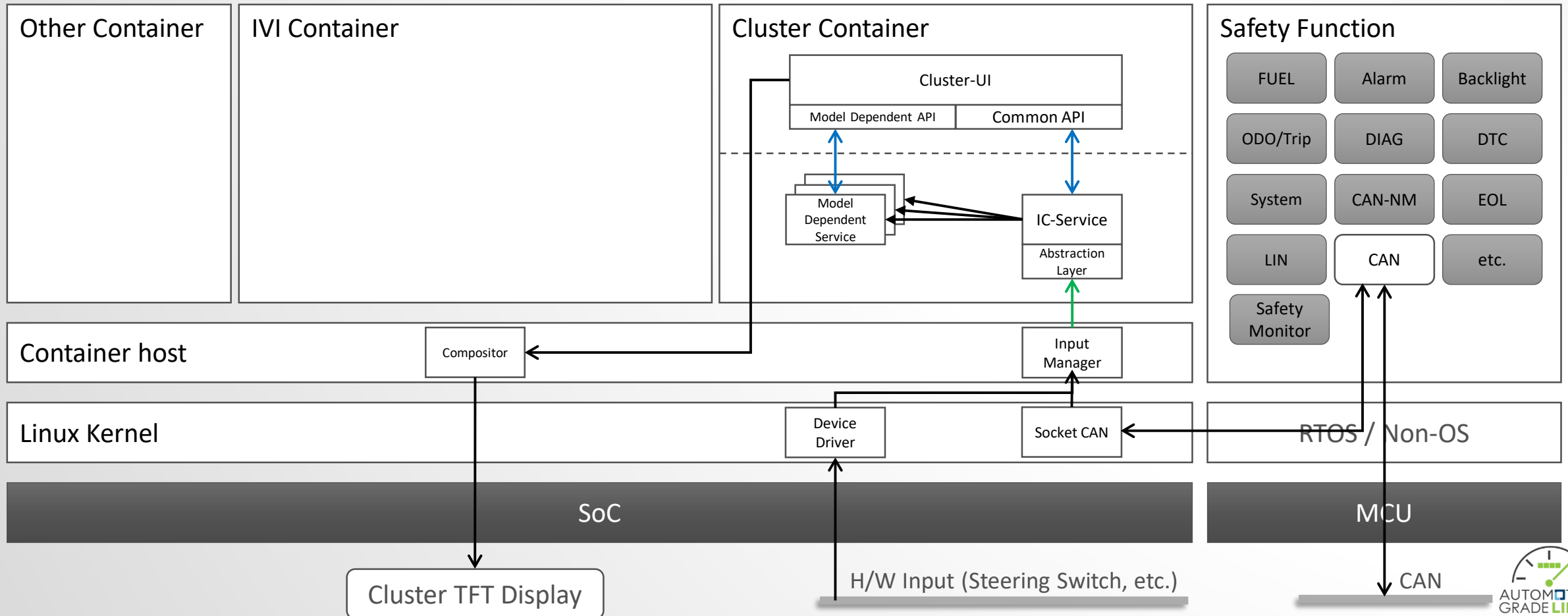Data flow example for OEM/Tier-1 common function.

- e.g. SPEED TACHO TEMP Shift Position Telltale

Data Flow – 2. OEM/Tier-1 Dependent

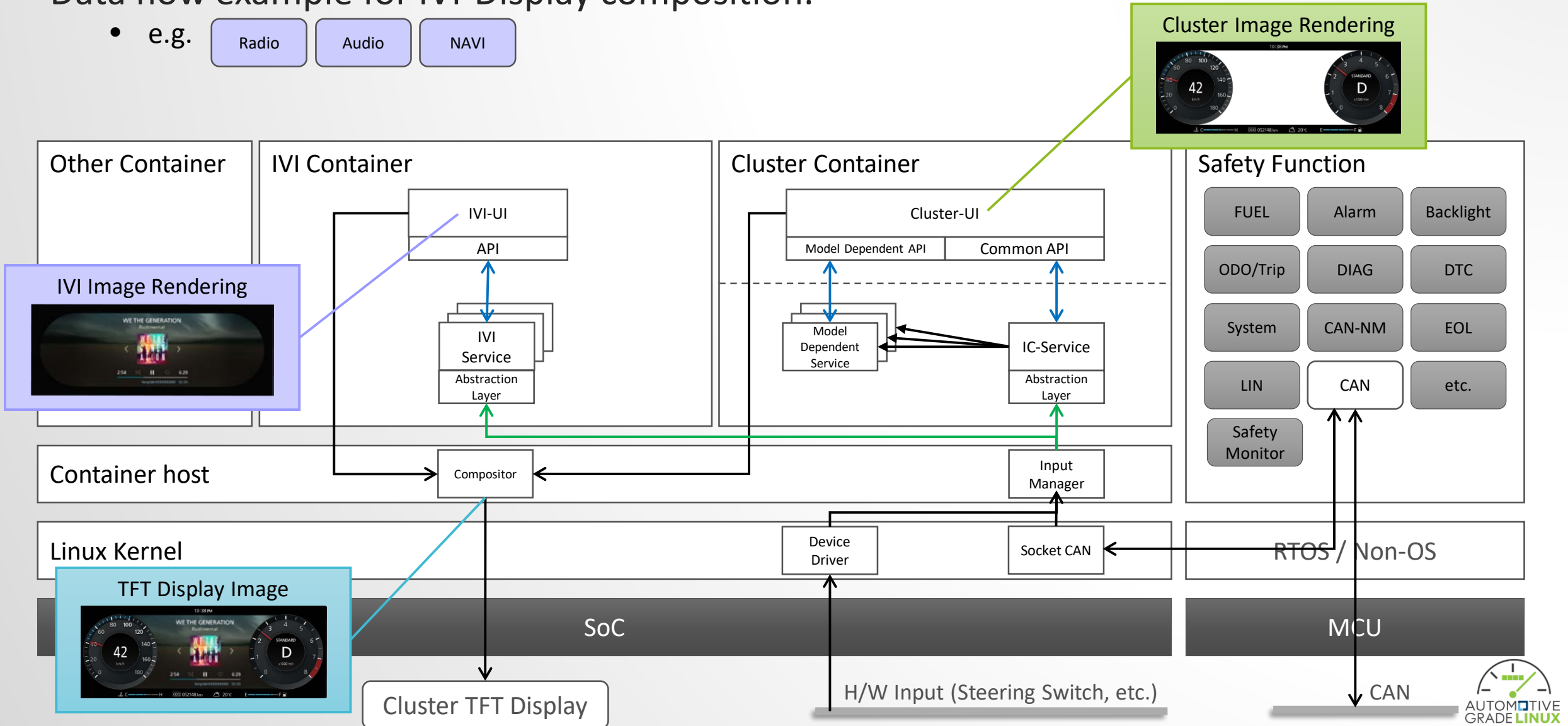Data flow example for common function partially includes OEM/Tier logic.

- e.g.

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

Data Flow – 3. IVI Composition

Data flow example for IVI-Display composition.

- e.g.   Radio   Audio   NAVI

Cluster Image Rendering

Other Container

IVI Image Rendering

IVI Container

IVI-UI

API

IVI Service

Abstraction Layer

Cluster Container

Cluster-UI

Model Dependent API | Common API

Model Dependent Service

IC-Service

Abstraction Layer

Safety Function

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

Container host

Compositor

Input Manager

Linux Kernel

TFT Display Image

Device Driver

Socket CAN

RTOS / Non-OS

SoC

Cluster TFT Display

H/W Input (Steering Switch, etc.)

MCU

CAN

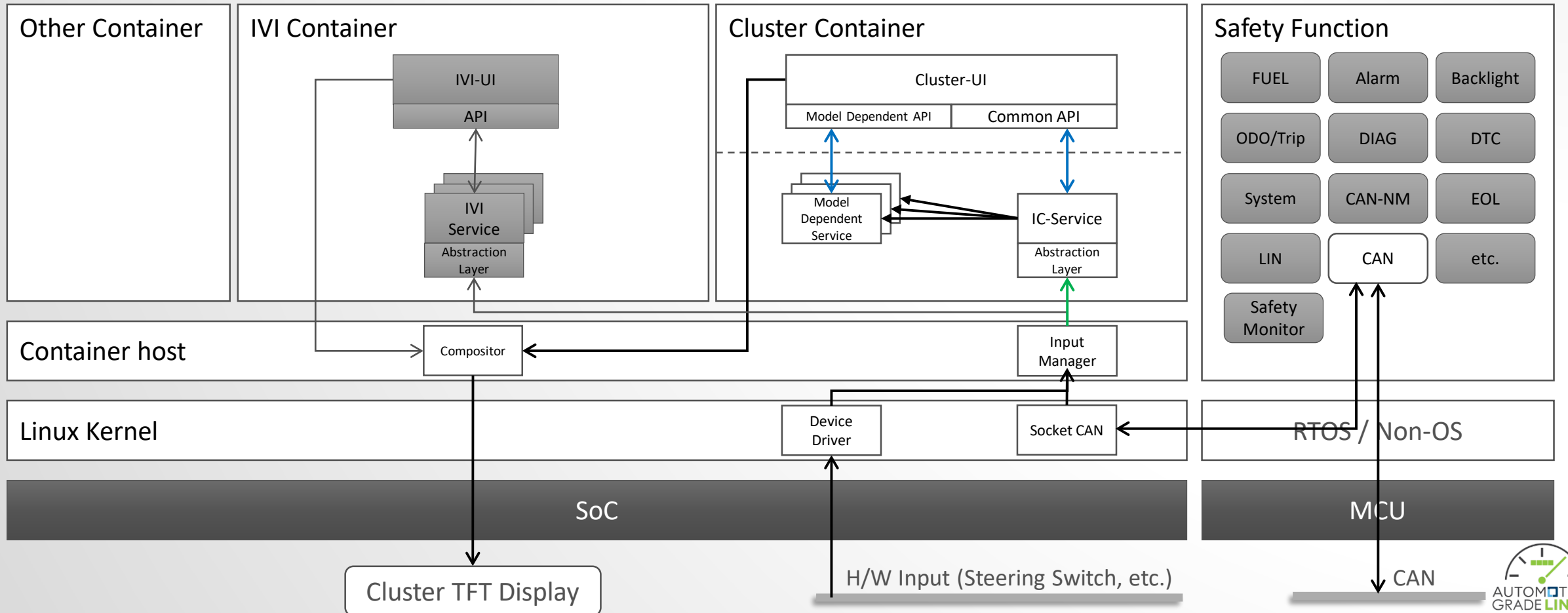AUTOMOTIVE GRADE LINUX

Implementation Model – Open Point

The following points needs to be clarified and aligned with Cluster Tier-1.

1. Implementation model of IC-Service and the other components.
2. API and Interface method of IC-Service.

→ It shall be defined with IC-EG Technical Working Group.

Thank you!