Analysis and Refinement of Architecture for Realizing AGL Instrument Cluster

- AGL All Member Meeting Spring 2021 -

18.March.2021

Masanori Maruyama
Nippon Seiki Co.,Ltd.

Naoto Yamaguchi
AISIN AW CO.,LTD.

Today's Presenters

Name :
Masanori Maruyama

Company :
Nippon Seiki Co.,Ltd

Career :
Automotive software engineer since 2003. (Cluster, HUD)

Name :
Naoto Yamaguchi

Company :
AISIN AW CO.,LTD.

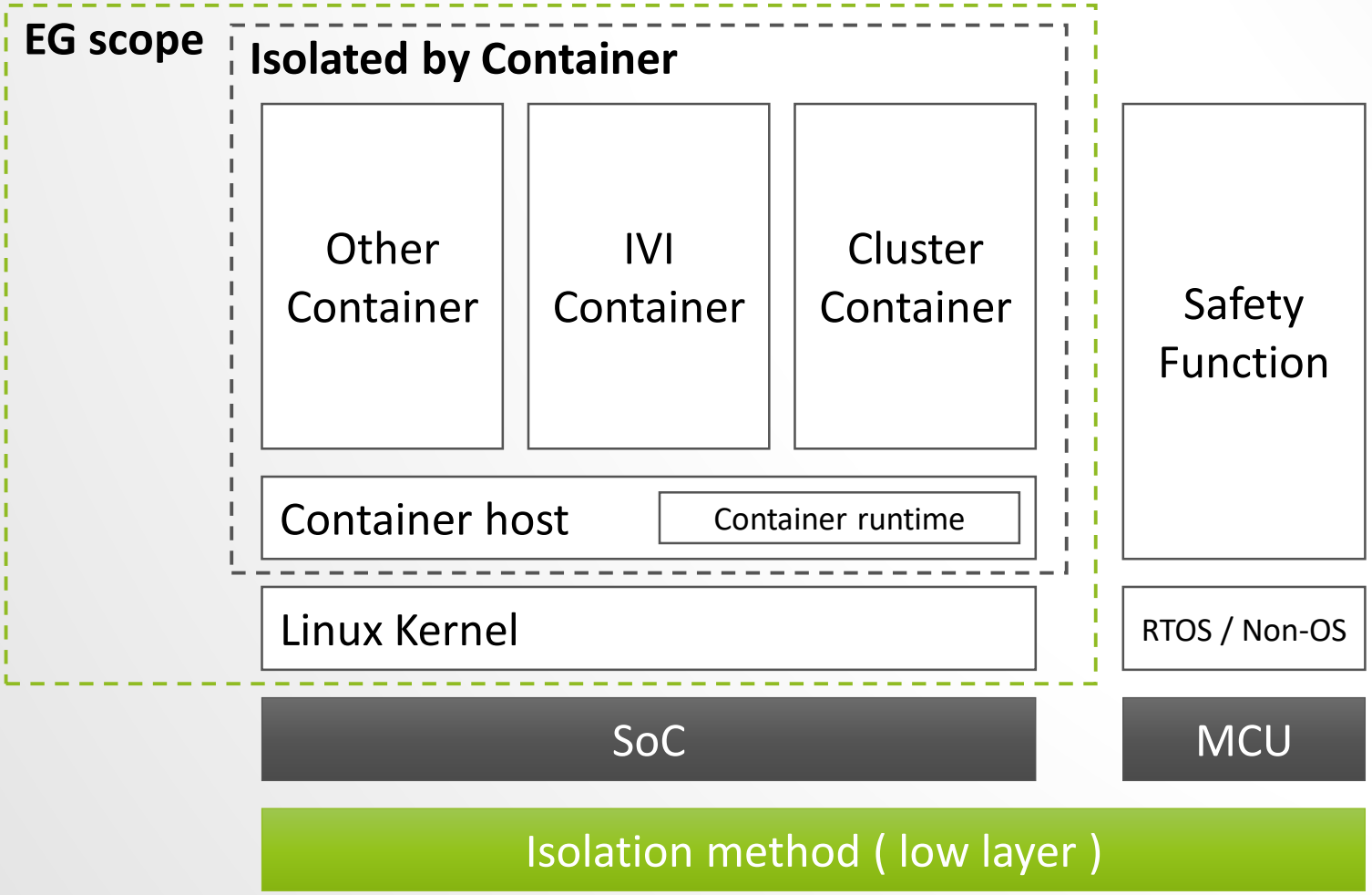
Career :
Automotive platform software engineer since 2007.

AUTOMOTIVE GRADE LINUX

Contents

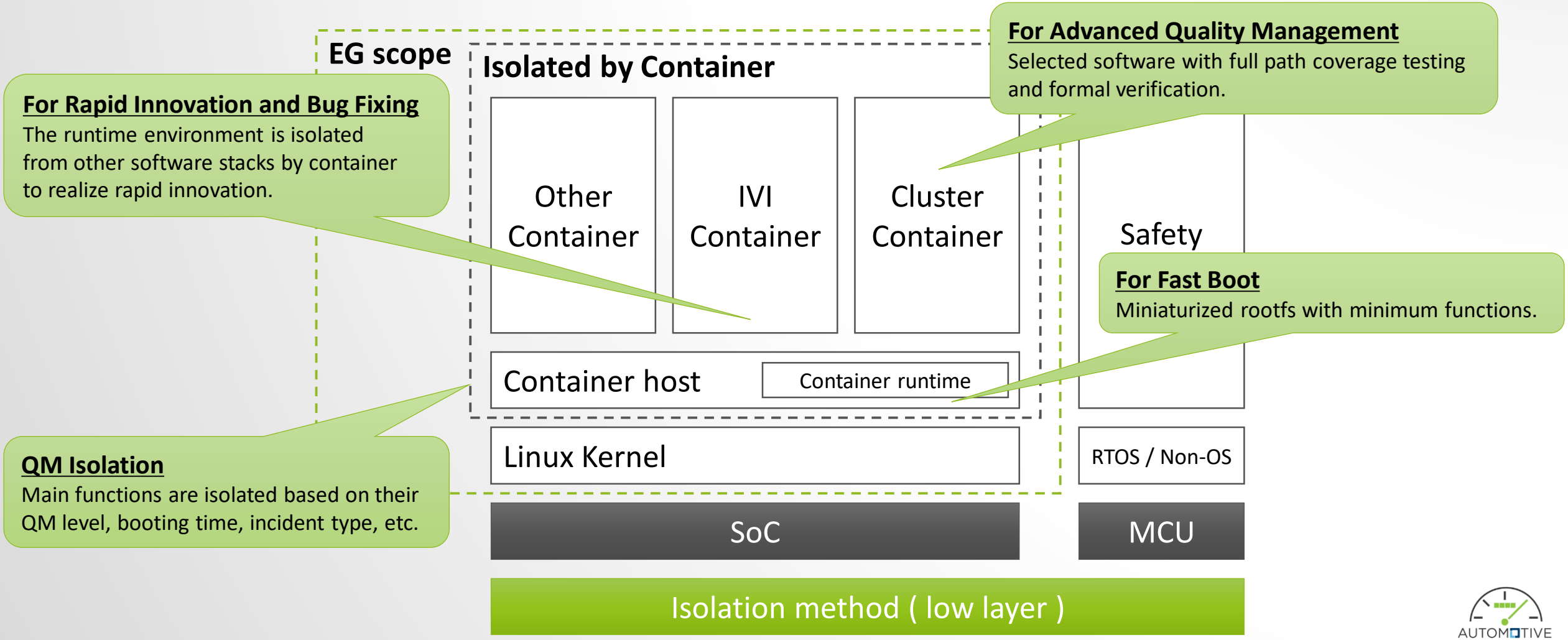
Container Architecture – Overview

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



Container Architecture – Overview

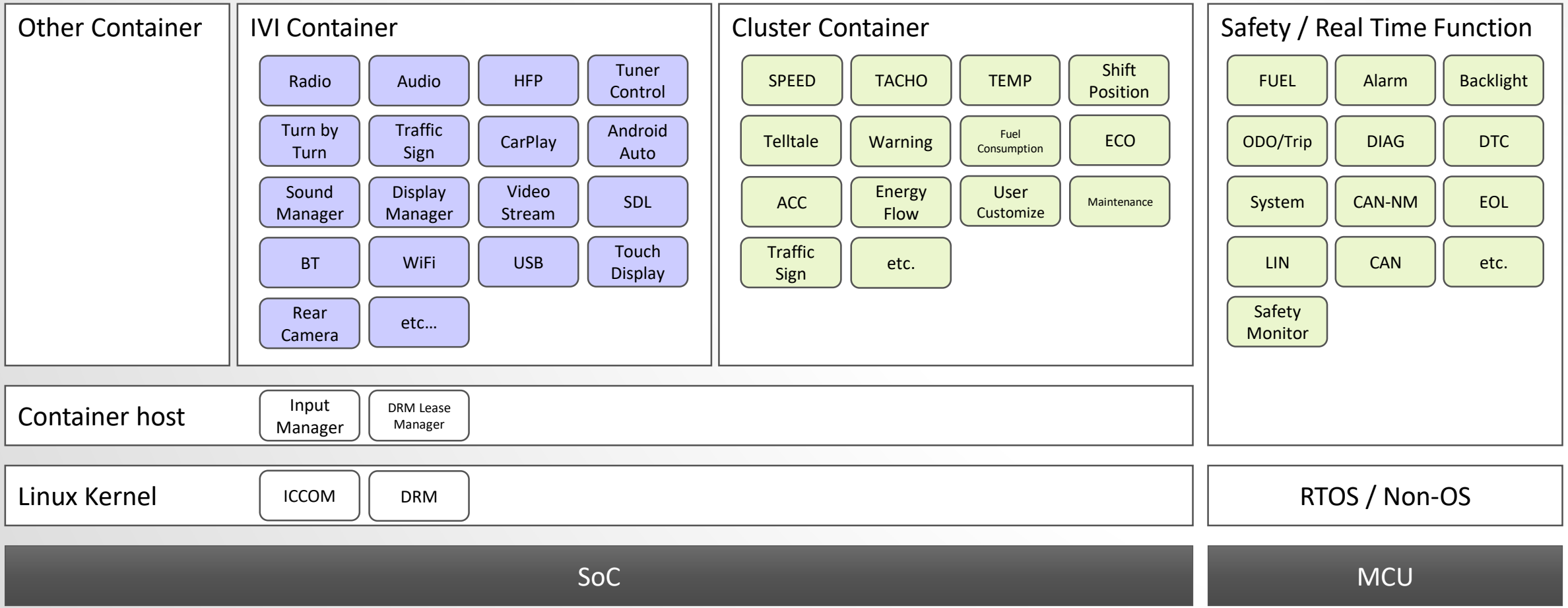
Cluster Function and IVI Function shall be separated by Linux Container Technology in order to achieve QM isolation.



Function Block Assignment

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

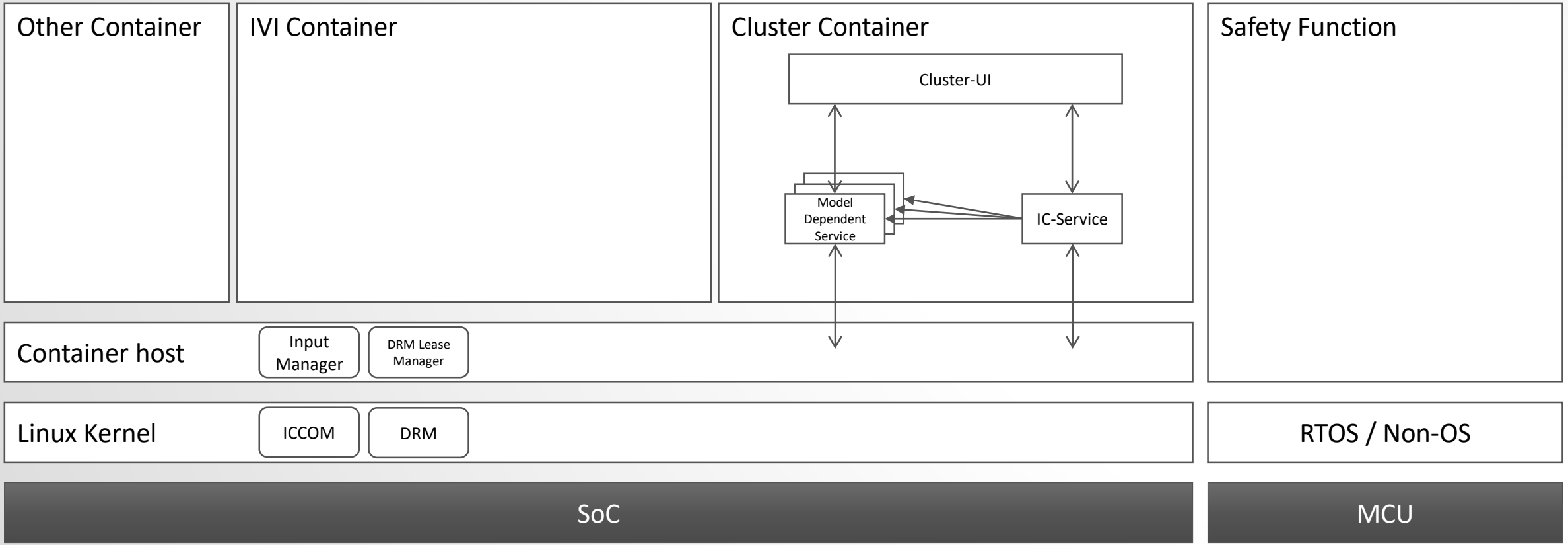
- All of the other cluster function shall be assigned onto the cluster container.



Cluster Container – Overview

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

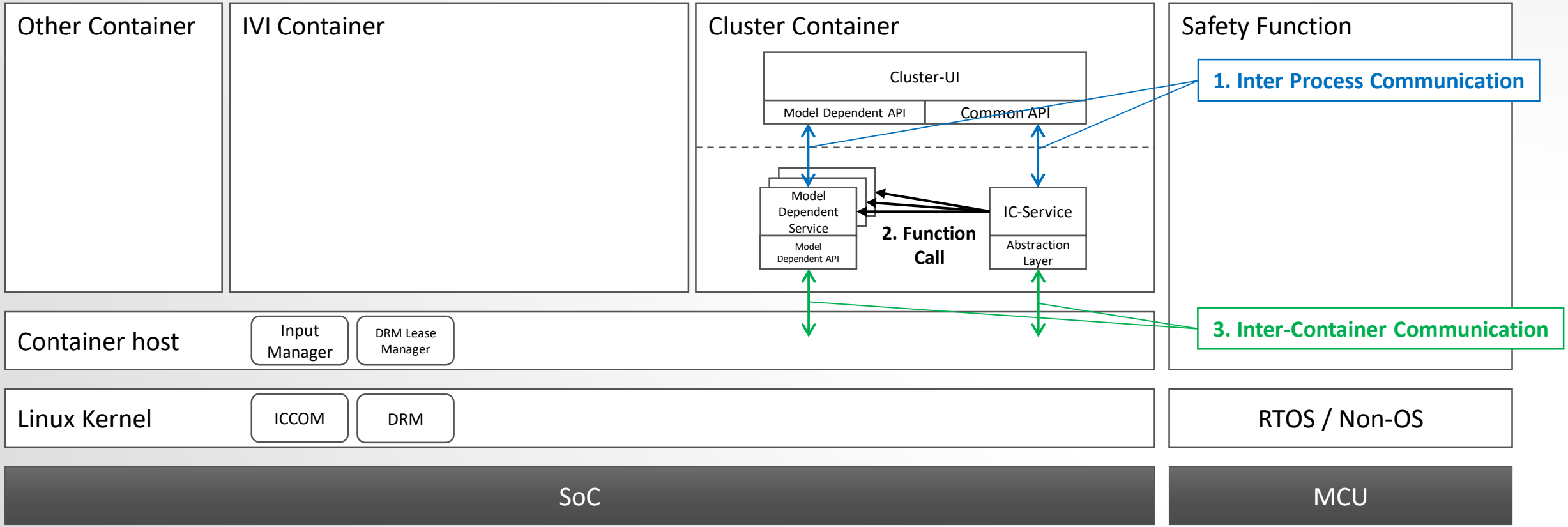
- IC-Service shall consist of a function logic.
- Cluster-UI shall consist of an UI state machine and assets.
- IC-Service shall be separated by a model dependency.



IC-Service Interface

IC-Service shall consist of the following three interface.

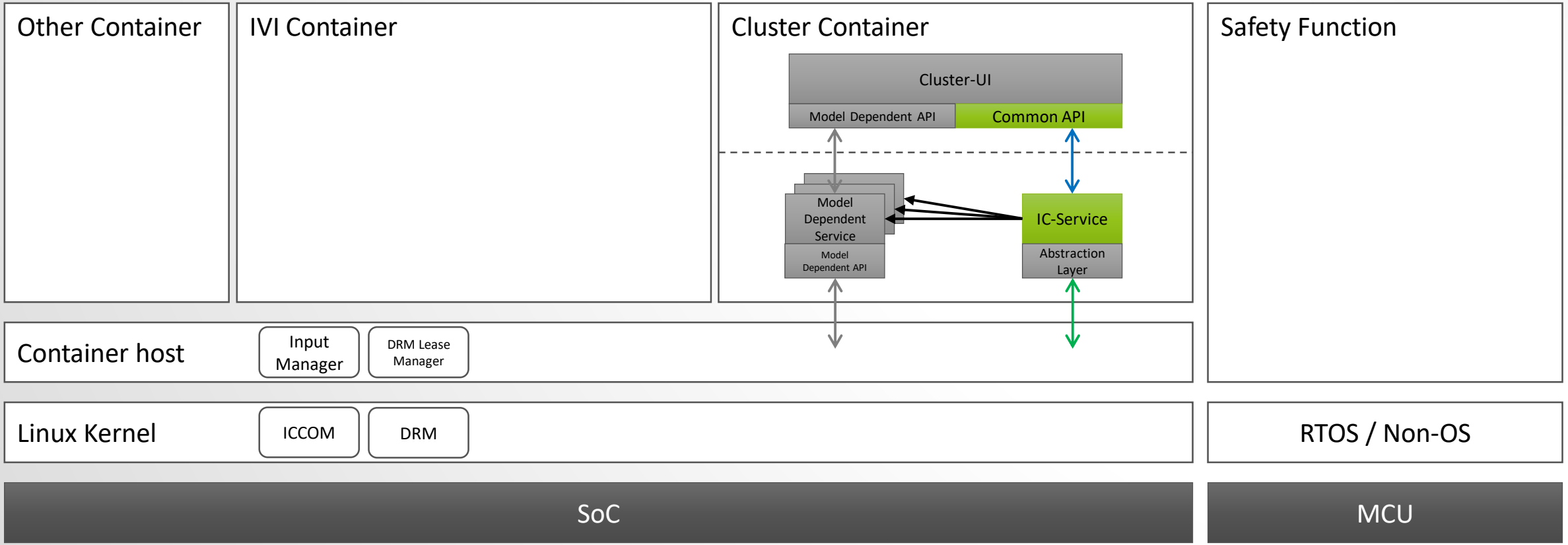
1. Cluster-UI shall be defined a separated process. → **Inter Process Communication**
2. Model dependent service shall be called from IC-Service as a common interface. → **Function Call**
3. IC-Service shall communicate with another container or container host. → **Inter Container Communication**



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.

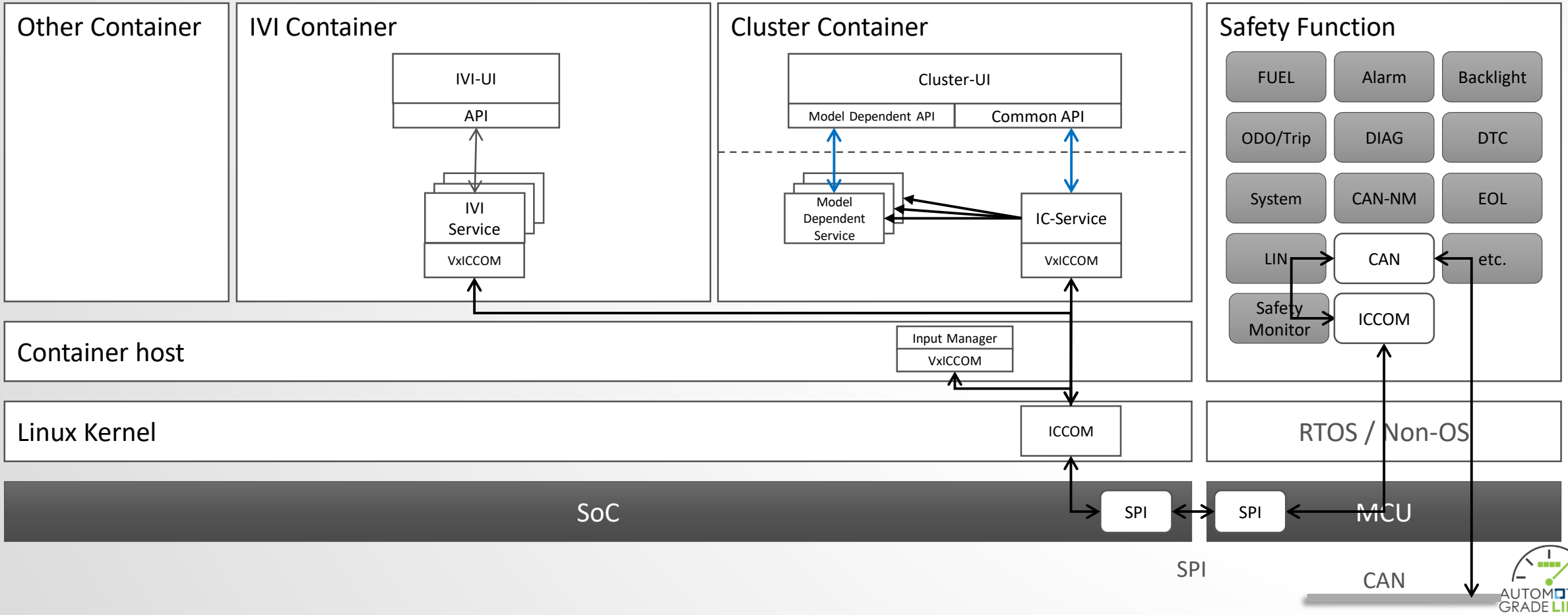


Contents

Data Flow – 1. ICCOM

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

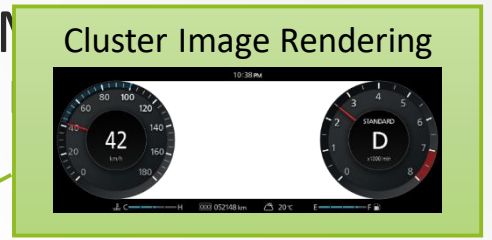
- ICCOM socket shall be directly opened in application container, and not in the container host.
- Keeping advantage of peer to peer communication shall reduce latency and complexity.



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

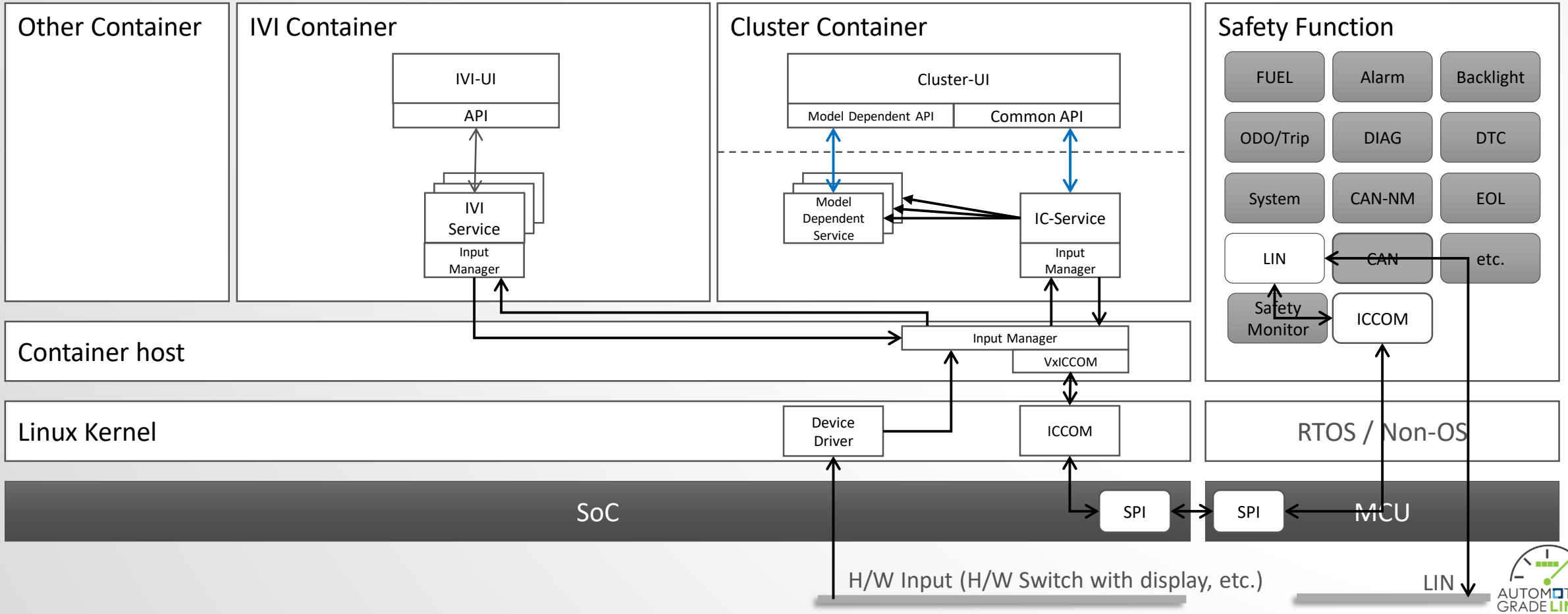
CAN

AUTOMOTIVE GRADE LINUX

Data Flow – 2. Input Manager

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

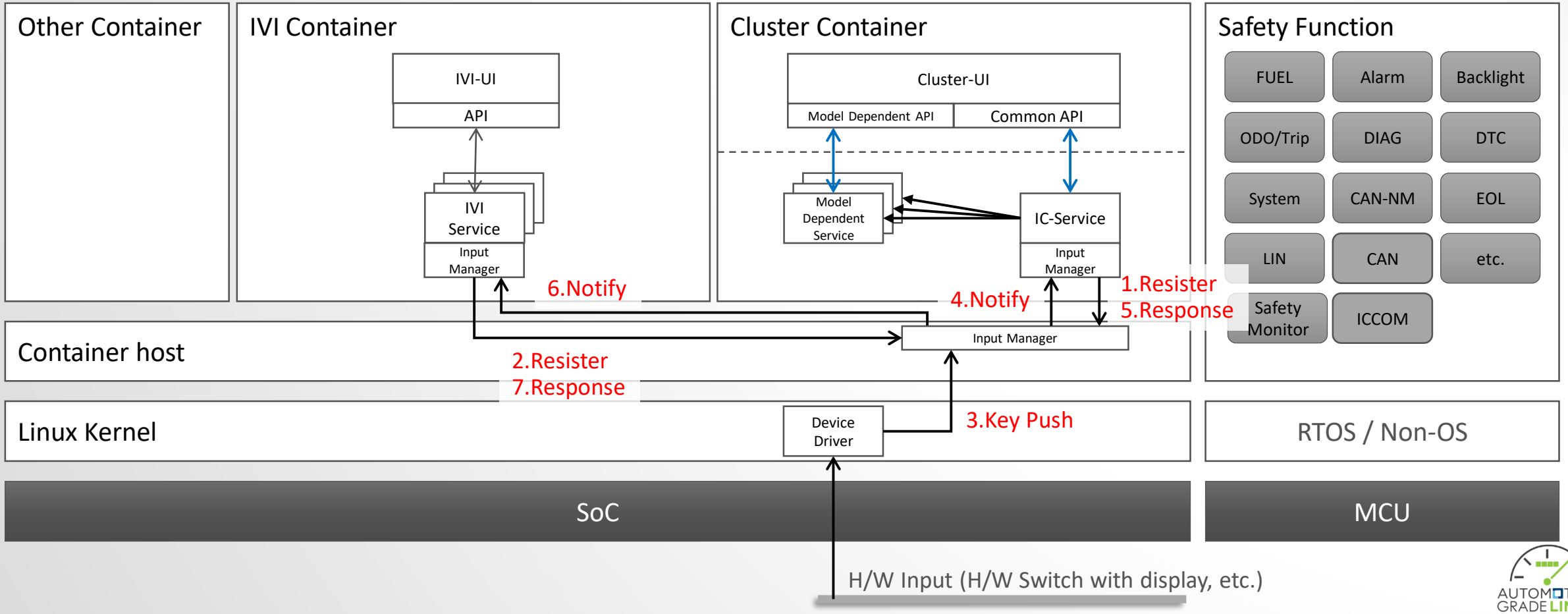
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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



2. Input Manager – e.g. H/W switch

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

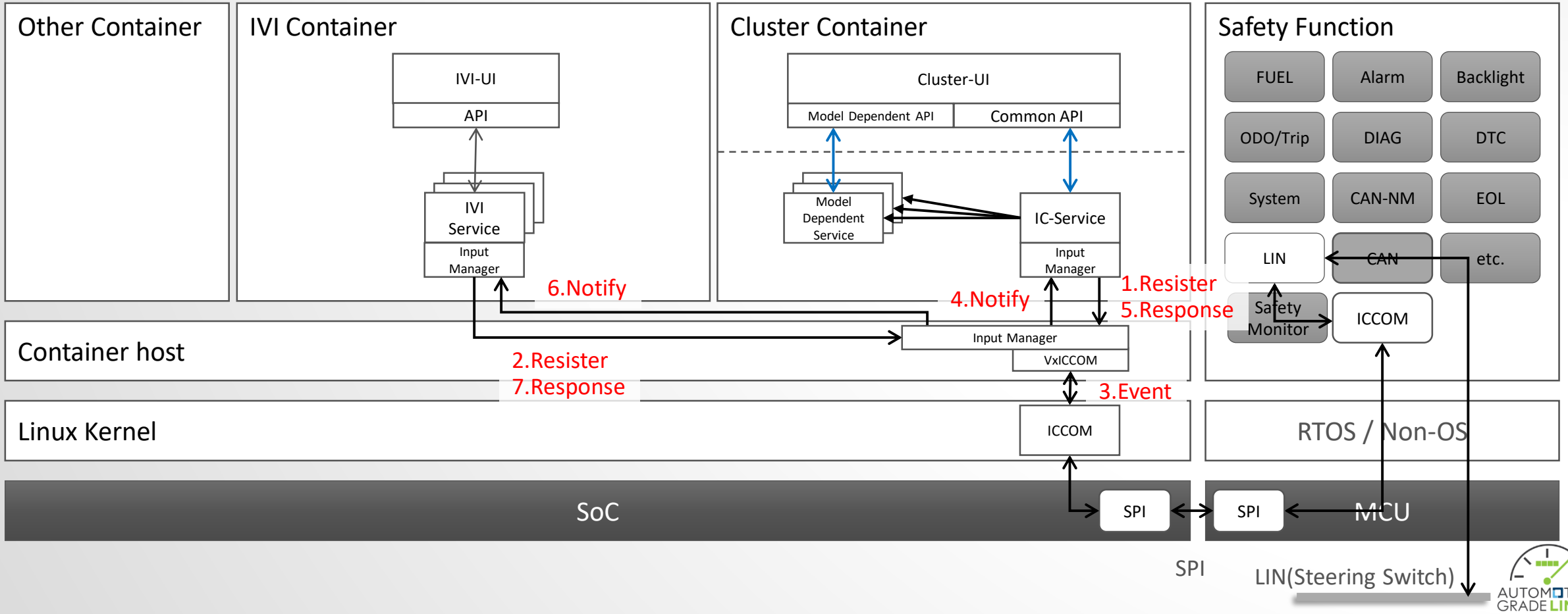
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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



2. Input Manager – e.g. LIN(Steering Switch)

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

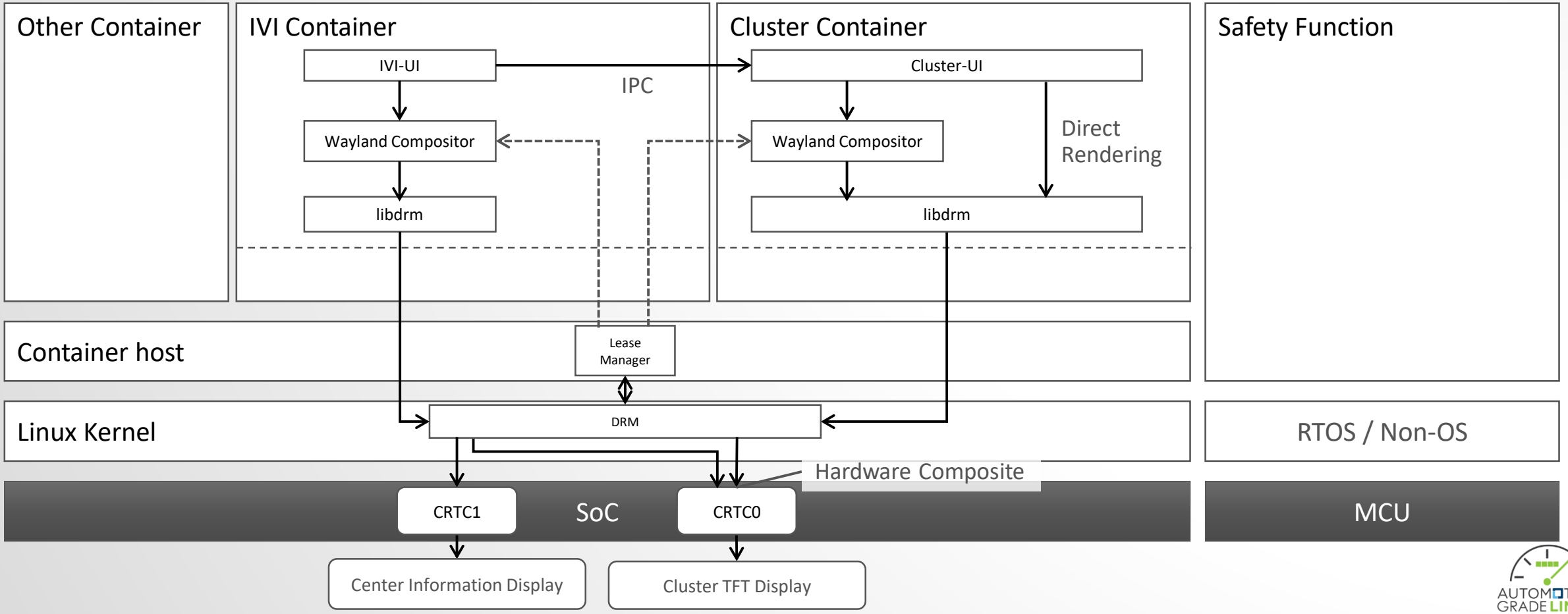
- Application container shall register to container host as a listener for specific event.
- Container host sends the event to the registered application container.
- Application container checks if it consumes the event, send back the result to the container host.
- In case of multiple container registered,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



Data Flow – 3. Window Manager

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

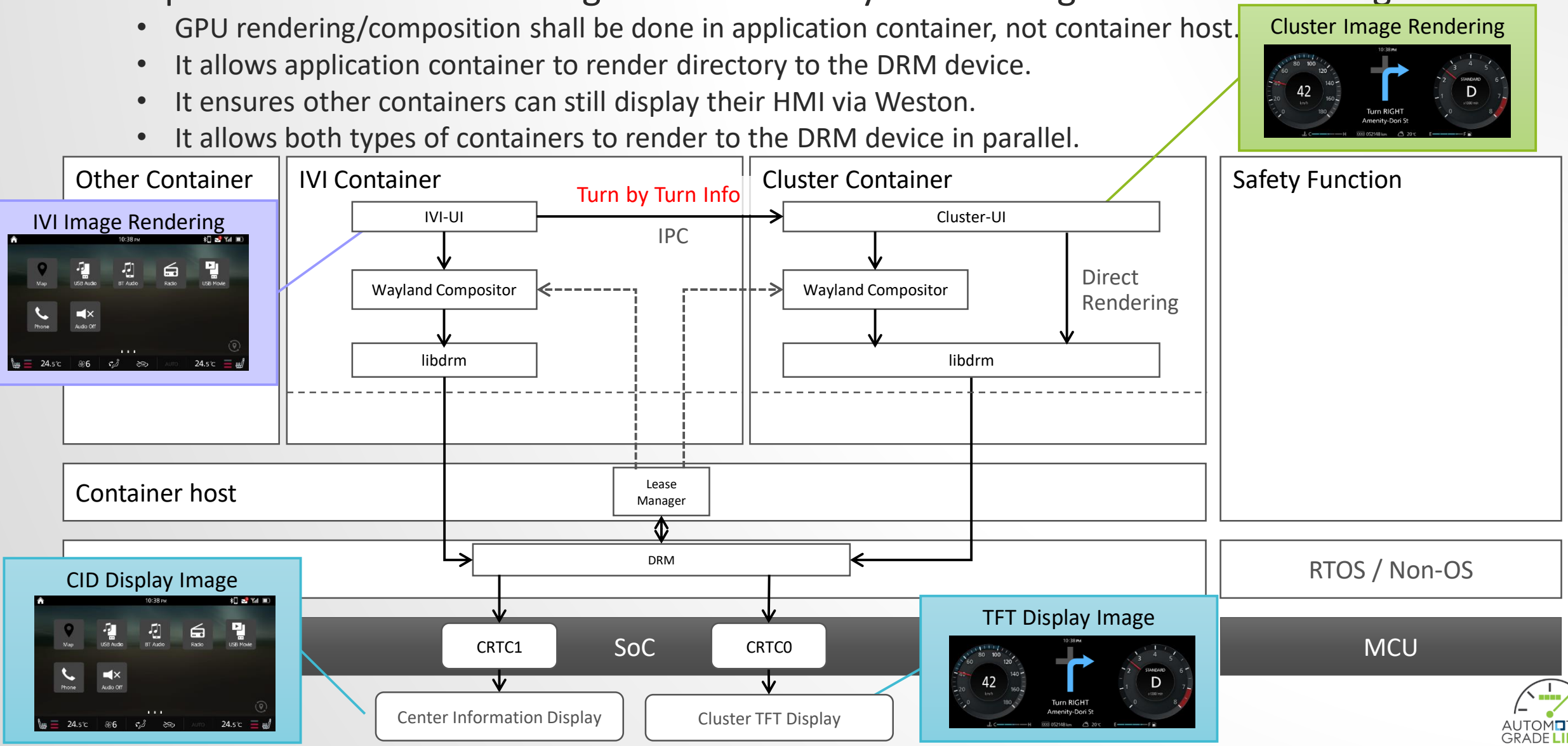
- GPU rendering/composition shall be done in application container, not container host.
- It allows application container to render directly to the DRM device.
- It ensures other containers can still display their HMI via Weston.
- It allows both types of containers to render to the DRM device in parallel.



3. Window Manager – e.g. IPC

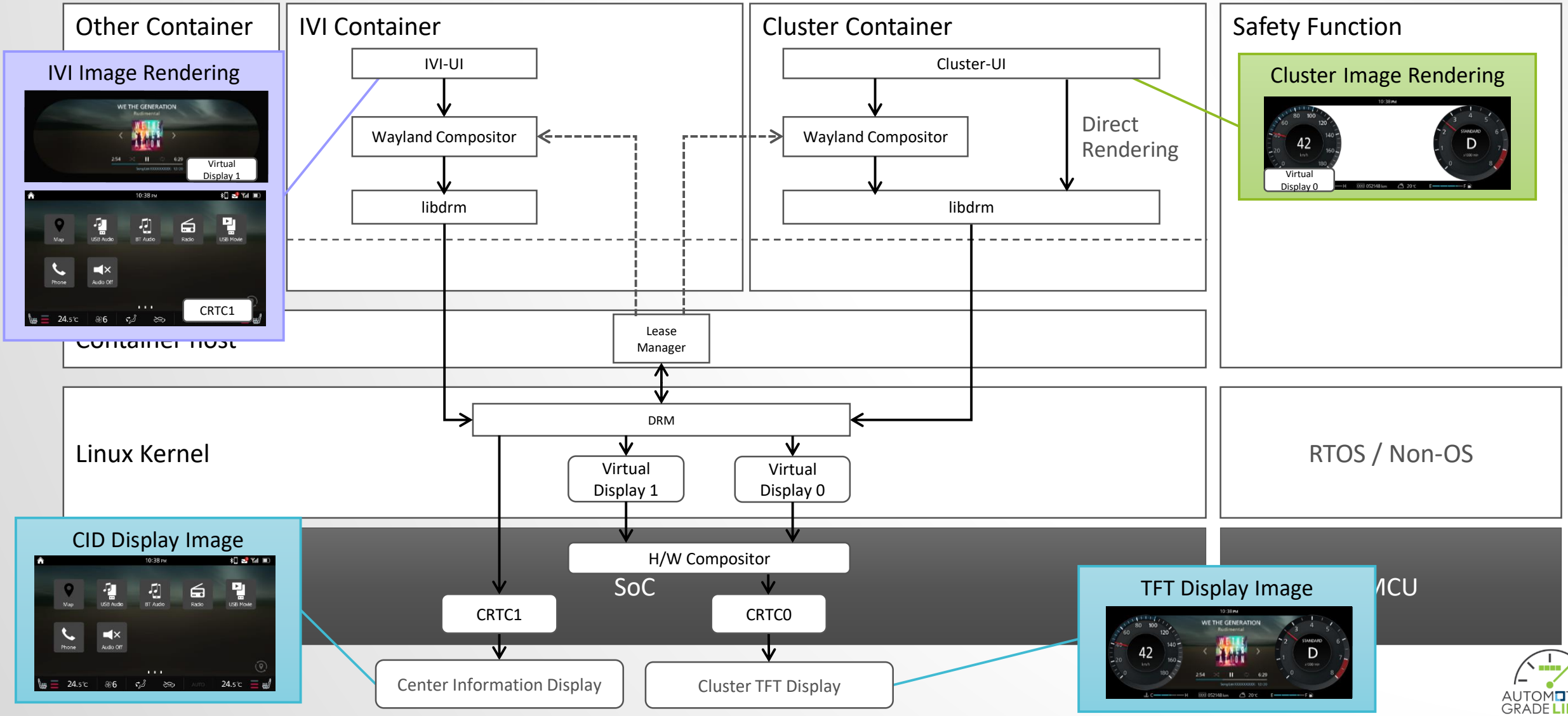
Multiple container DRM sharing shall be done by introducing DRM Lease Manager.

- 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.



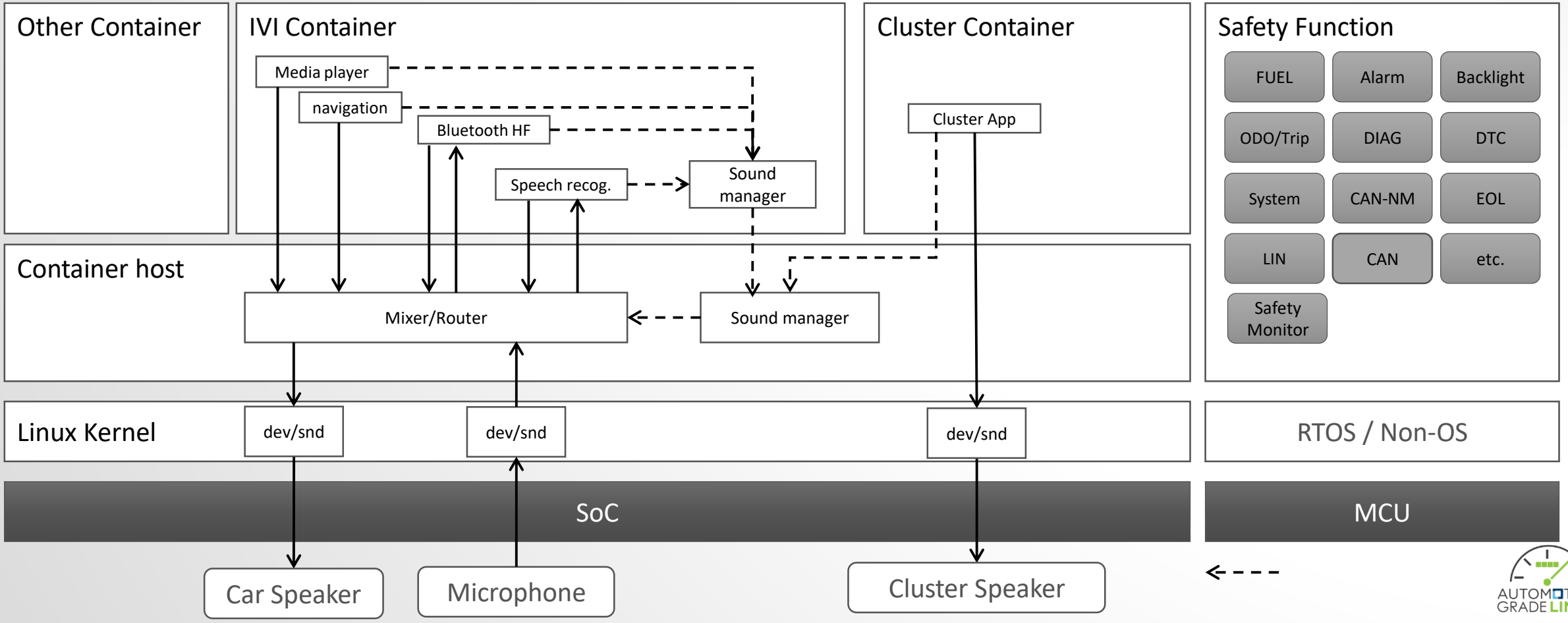
3. Window Manager – e.g. H/W Composite

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



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 manager in host shall manage system wide policy, and sound manager in IVI shall manage policy in IVI world.



Contents

Today's Presenters

Name :
Masanori Maruyama

Company :
Nippon Seiki Co.,Ltd

Career :
Automotive software engineer since 2003. (Cluster, HUD)

Name :
Naoto Yamaguchi

Company :
AISIN AW CO.,LTD.

Career :
Automotive platform software engineer since 2007.

AUTOMOTIVE GRADE LINUX

Previous discussion

- Why need quality management process?

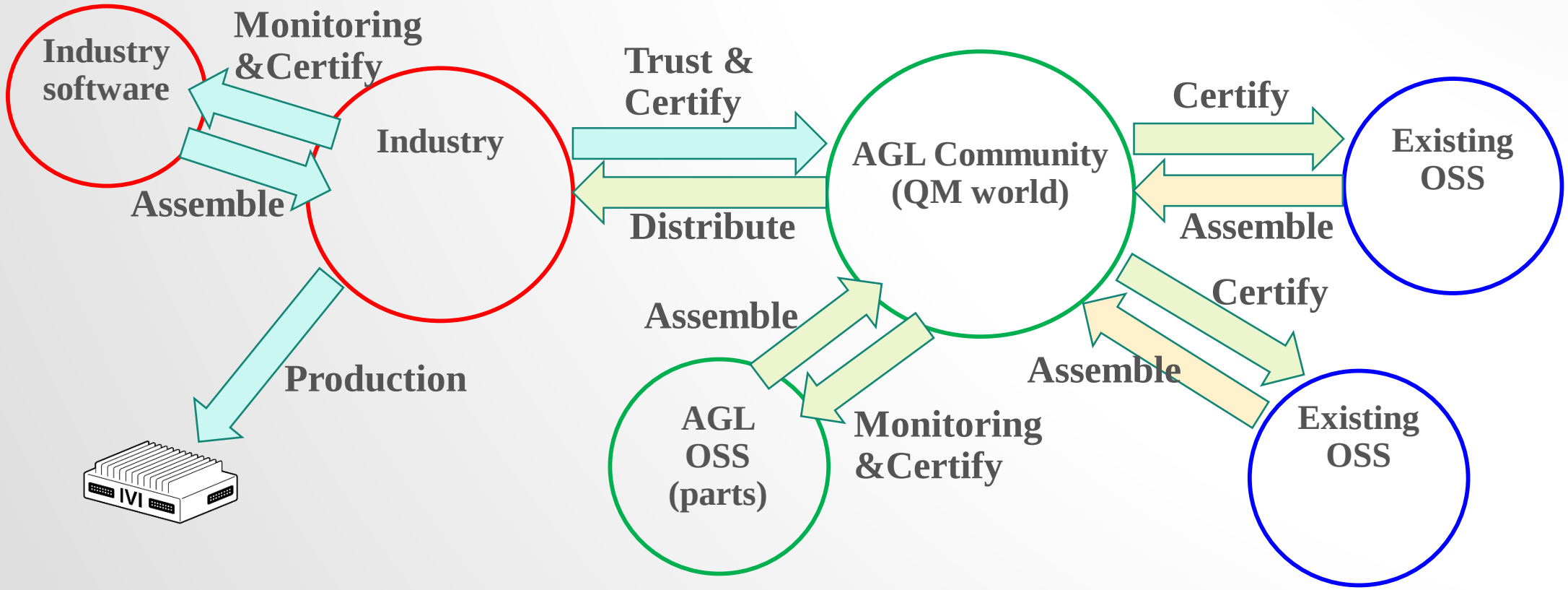
- AGL is linux based platform development project for automotive use case.
  - AGL has been developed for IVI use cases. There ware not much emphasis on software quality.
    - Any existing open source can add to AGL distribution without quality check.
    - Not defined coding standard.
    - Not defined documentation standard.

- Instrument cluster expert group started quality management process discussion since 2020.
  - We investigated to Automotive SPICE and existing open source development process.

- Our first activity already shared in last AMM.

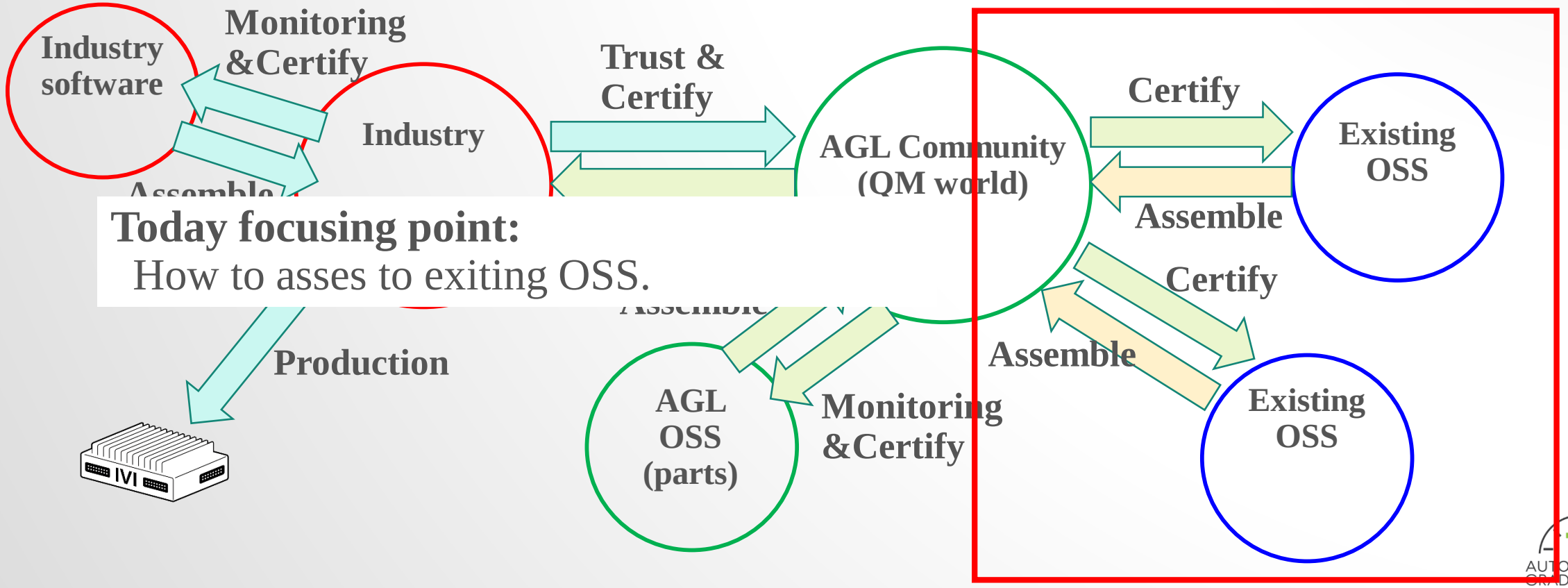
What we aim

- We want to create workflow from open source development to product development.
  - Want to be able to certify that it has quality control.
  - Want to be able to embrace by open source community and industry.



What we aim

- We want to create workflow from open source development to product development.
  - Want to be able to certify that it has quality control.
  - Want to be able to embrace by open source community and industry.



Today focusing point:
How to asses to exiting OSS.

Software quality in the production case

- In production case.
  - This software doesn't have bug in the own product use case.
    - If this product does not use wifi function of linux kernel, we can break it.
  - This software allows for many workarounds.
    - When this system has hardware bugs, this software implement workaround to recover hardware bugs.
  - This software allows for own product specific design.
    - We can change special interface from original interface to reduce own development cost in this product.

- In AGL platform case. Is that same?
  - **The answer is NO.**

What is software quality in reusing case?

- What is the difference between product development and AGL development?

- Existing OSS is developed by own use case, typically it is not fully match to automotive specific use case.
  - When AGL community are reusing existing OSS, AGL community should realize automotive specific use case by architectural design. This architectural design must not break.
  - AGL community shall share the document, that describe to why need this OSS.
    - This topic activity is shared in last presentations.

- Existing OSS is developed by own community rule.
  - Not all communities have such rules. On the other hand this community policy must be respected.
  - In this case, AGL community shall select OSS based on own policy.

How to do?

- When in case of requirements assign to existing OSS, we have to trust these OSS.
  - That means AGL community have to certify existing OSS.

- Automotive SPICE define these method in REU.2 (Reuse Program Management).
  - It’s good reference.

REU.2.	BP1	Define organizational reuse strategy.
	BP2	Identify domains for potential reuse.
	BP3	Assess domains for potential reuse.
	BP4	Assess reuse maturity.
	BP5	Evaluate reuse proposals.
	BP6	Implement the reuse program.
	BP7	Get feedback from reuse.
	BP8	Monitor reuse.

How to select existing OSS

- Automotive SPICE requires nine work products in REU.2.
- What should we do?

REU.2.	04-02	Domain architecture.	Design and describe to "which OSS will handle which functions".
	04-03	Domain model.	
	08-17	Reuse plan.	Define the assessment rule.
	09-03	Reuse policy.	
	12-03	Reuse proposal.	
	13-04	Communication record.	Record the content of the assessment review.
	15-07	Reuse evaluation report.	Review for the these documents.
	15-13	Assessment/audit report.	Describe to the assessment result based on assessment rule.
	19-05	Reuse strategy.	Define to the reusing process.

Define the assessment rule

- Instrument Cluster expert group defined assessment rule draft.
- This assessment rule consists of five items.
  - License
    - Define the acceptable OSS license.
  - Community check list
    - Define the good OSS community criteria.
  - Long Term Stable
    - To be discuss.
  - Source code assessment
    - Understand the risks in this source code.
  - Requirement matching
    - To be discuss.

License

- AGL community has been a common understanding about OSS licensing. But it is not documented.
- We documented this common understanding.

Table 1-1. Allow license list

No.	License name	License URL
1	GNU General Public License, version 2	https://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
2	GNU Lesser General Public License, version 2.1	https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html
3	Apache License 2.0	https://www.apache.org/licenses/LICENSE-2.0
4	3-clause BSD license	https://opensource.org/licenses/BSD-3-Clause
5	2-clause BSD license	https://opensource.org/licenses/BSD-2-Clause
6	MIT License	https://opensource.org/licenses/mit-license.php
7	Mozilla Public License 2.0	https://www.mozilla.org/en-US/MPL/2.0/
8	zlib/libpng License	https://opensource.org/licenses/Zlib
9	Boost Software License 1.0	https://opensource.org/licenses/BSL-1.0
10	GCC Runtime Library Exception	https://www.gnu.org/licenses/gcc-exception-3.1.en.html

Table 1-2. Deny license list

No.	License name	License URL
1	GNU General Public License, version 3	https://www.gnu.org/licenses/gpl-3.0.en.html
2	GNU Lesser General Public License, version 3	https://www.gnu.org/licenses/lgpl-3.0.en.html
3	GNU Affero General Public License version 3	https://opensource.org/licenses/AGPL-3.0

***The GPLv3 and GPLv3 like license does not allowtivoization. This is incompatible with embedded use cases.**

Table 1-3. Special allow license list

No.	License name	License URL
1	GNU General Public License, version 3	https://www.gnu.org/licenses/gpl-3.0.en.html
2	GNU Lesser General Public License, version 3	https://www.gnu.org/licenses/lgpl-3.0.en.html

***The GPLv3 and GPLv3 like license does not allow tivoization. When these software only to use debugging (not installing in final product), it's no problem.**

AUTOMOTIVE GRADE LINUX

Community check list

- Why need this checklist?
- How do you think about good code?
- We think so;
  - Described by common style.
    - It is much easier to understand a large codebase when all the code in it is in a consistent style.
  - Reviewed by many contributors.
    - This is also the common sense of open source.
  - Frequently tested.
    - It is important to use tests to find regressions.
  - Etc…
- We want to define to quality check guide line for the OSS in AGL instrument cluster.
  - It's based on common sense of open source and automotive software.

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the contribution rule	Must
3	Defining the release rule.	Must
4	Providing a change logs.	Must
5	Have a bug tracking system or other bug report and fix solution such as active mailing list, github issue, etc..	Should
6	Have and maintain a test suite.	Should
7	Used in popular distributions such as RHEL, SUSE, Ubuntu, Debian.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the contribution rule	Must
3	Defining the release rule.	Must
4	Providing a change logs.	Must
5	Have a bug tracking and fix solution such github issue, etc..	
6	Have and maintain a	
7	Used in popular distr SUSE, Ubuntu, Debi	
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

It is important to define the rules for the development of community.

If the software is developed under the common rules, the risk of the declining software quality is low.

AUTOMOTIVE GRADE LINUX

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the contribution rule	Must
3	Defining the release rule.	Must
4	Providing a change logs.	Must
5	Have a bug tra… system or other bug report and fix solution such… git…	Should
6	Ha…	
7	Us… SU…	
8	2 o…	
9	Inc… pac…	

When found the bugs such as CVE, we must know that bugs is including or not in this version. If that OSS has been released with versioning, we can check easily.
When we can get change logs, we can check to "this version up shall deriver to own product or not?".
These points are important points after SOP.

Community check list

No.	Requirement	
1	Defining the codin	
2	Defining the contri	
3	Defining the release rule.	Must
4	Providing a change l[illegible]s.	Must
5	Have a bug tracking system or other bug report and fix solution such as active mailing list, github issue, etc..	Should
6	Have and maintain a test suite.	Should
7	Used in popular distributions such as RHEL, SUSE, Ubuntu, Debian.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

When OSS community has bug tracking system, we can know "which bug is not fixed now". But this is a difficult request for small OSS community.

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the codin	
2	Defining the contri	
3	Defining the releas	
4	Providing a change	
5	Have a bug tracking system or [illegible] and fix solution such as a[illegible]ailing list, github issue, etc..	Should
6	Have and maintain a test suite.	Should
7	Used in popular distributions such as RHEL, SUSE, Ubuntu, Debian.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

When this OSS has and maintain a test suite, this OSS is tested frequently. And we can test self building binary in own environment.

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the cor	
3	Defining the rel	
4	Providing a char	
5	Have a bug tracl and fix solution such as active github issue, etc..	
6	Have and maintain a test suite.	Should
7	Used in popular distributions such as RHEL, SUSE, Ubuntu, Debian.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

When this OSS is used in major distribution, this OSS is tested by many developer in many environment.
In this case, it is more likely that bugs will be found and fixed.

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the contribution rule	Must
3	Defining the re	
4	Providing a ch	
5	Have a bug tra and fix solutio github issue, et	
6	Have and maintain a test s	Should
7	Used in popular di butions such as RHEL, SUSE, Ubuntu, Debian.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

This is very sad...
When this OSS is developed by one contributor. If he is dead, this OSS will not maintain.

Community check list

No.	Requirement	Req. level (Draft)
1	Defining the coding rule or guideline	Must
2	Defining the contribution rule	Must
3	Defining the release rule.	Must
4	Providing	
5	Have a bu and fix sc github iss	
6	Have and	
7	Used in popular distri SUSE, Ubuntu	as RHEL, an.	Should
8	2 or more active contributors.	Should
9	Including OIN(Open Invention Network) packages list	Recommend

When this OSS is including in OIN safe lists, it doesn't have patent risk.

AUTOMOTIVE GRADE LINUX

Source code assessment

- Community check list is defined how to asses to the health of community.
- Source code assessment aim to check source code quality using static analysis tool.
  - We are not aiming for a MISRA C check. MISRA C rule is now conflicting to existing OSS coding guideline such as Linux kernel coding style.
  - We are aiming for understanding the risks of the code.
- Jan-Simon Moeller is strongly support to this assessment infrastructure.
  - Thanks for his contributions.

Conclusion

- In this session, maruyama-san shared architecture of the AGL instrument cluster in 1st part. Our expert group is developing the code based on this architecture now.

- I shared activity of the quality management in the AGL instrument cluster expert group in 2nd part. This work is big challenge. Our expert group is developing the software stack based on this document. And it is evaluating to this method in parallel.

- If you are interesting this session, please join our expert group.