How to Balance Safety and Hardware Decoupling in SoDeV

Automotive Grade Linux All Member Meeting Japan
2026 May 13-14, Tokyo, Japan

Naoto Yamaguchi
AISIN CORPORATION

Speaker Biography

- Name: Naoto Yamaguchi
  - Doctor of Informatics
  - Working in AISIN CORPORATION. for 15+ years.
  - Embedded Software Engineer since 2007,
  - Linux and OSS for automotive development (2011 ~)

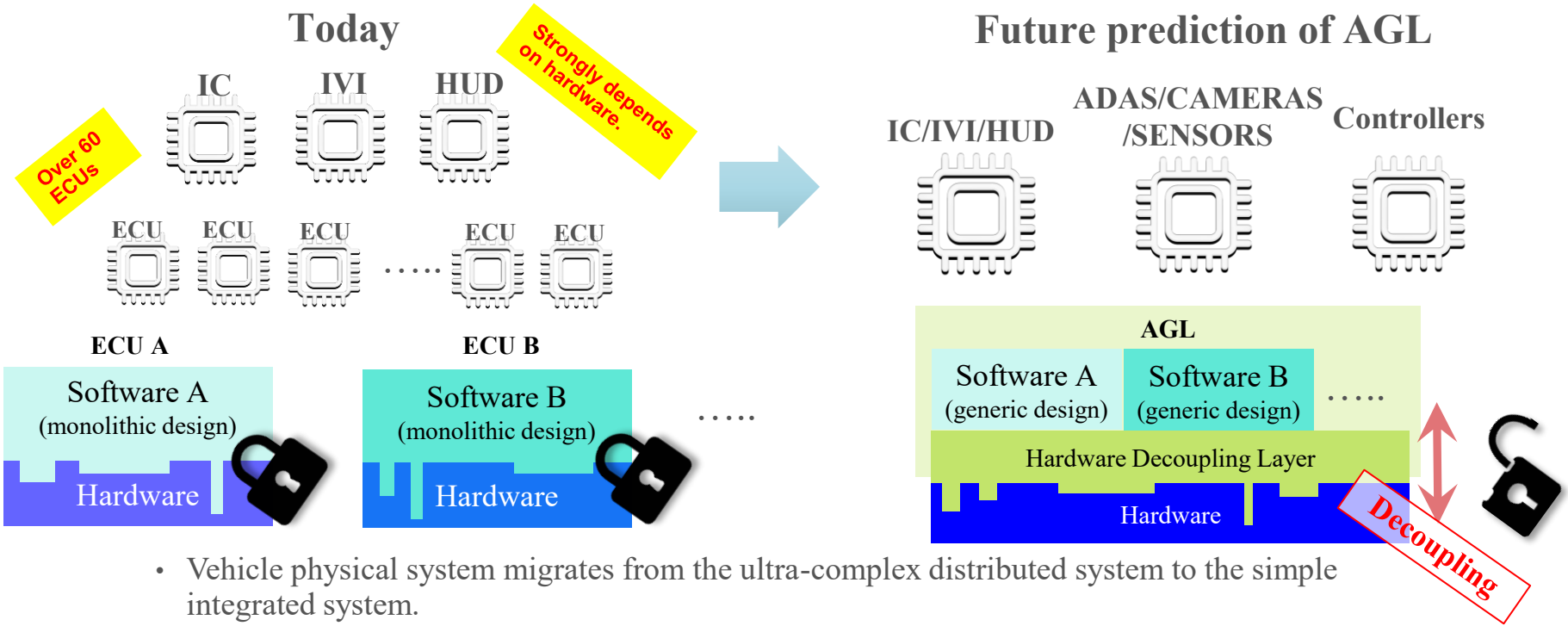
- My experience of Open-Source Community
  - Join to AGL (2014~)
    - Instrument Cluster Expert Gr. (2019 ~)
    - AGL Steering Committee/Board member (2020 ~)
    - SoDeV Development (2025 ~)

Overview

- SoDeV Review
  - Sharing for SoDeV architecture.
- Existing work for AGL System
  - Sharing for safety for telltale feature.
- Consideration for Telltale on SoDeV
  - SoDeV design and discussion

SoDeV Review

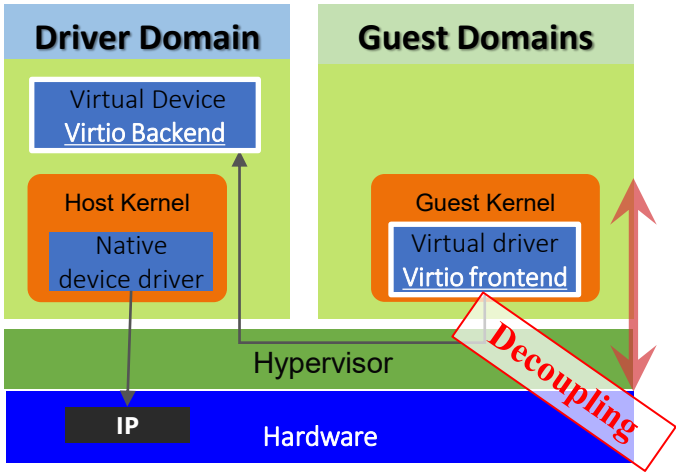
Evolution of Vehicle Software



- Vehicle physical system migrates from the ultra-complex distributed system to the simple integrated system.
- Hardware decoupling is the most important thing. Reduce complexity, use the same software, easy to upgrade and more.

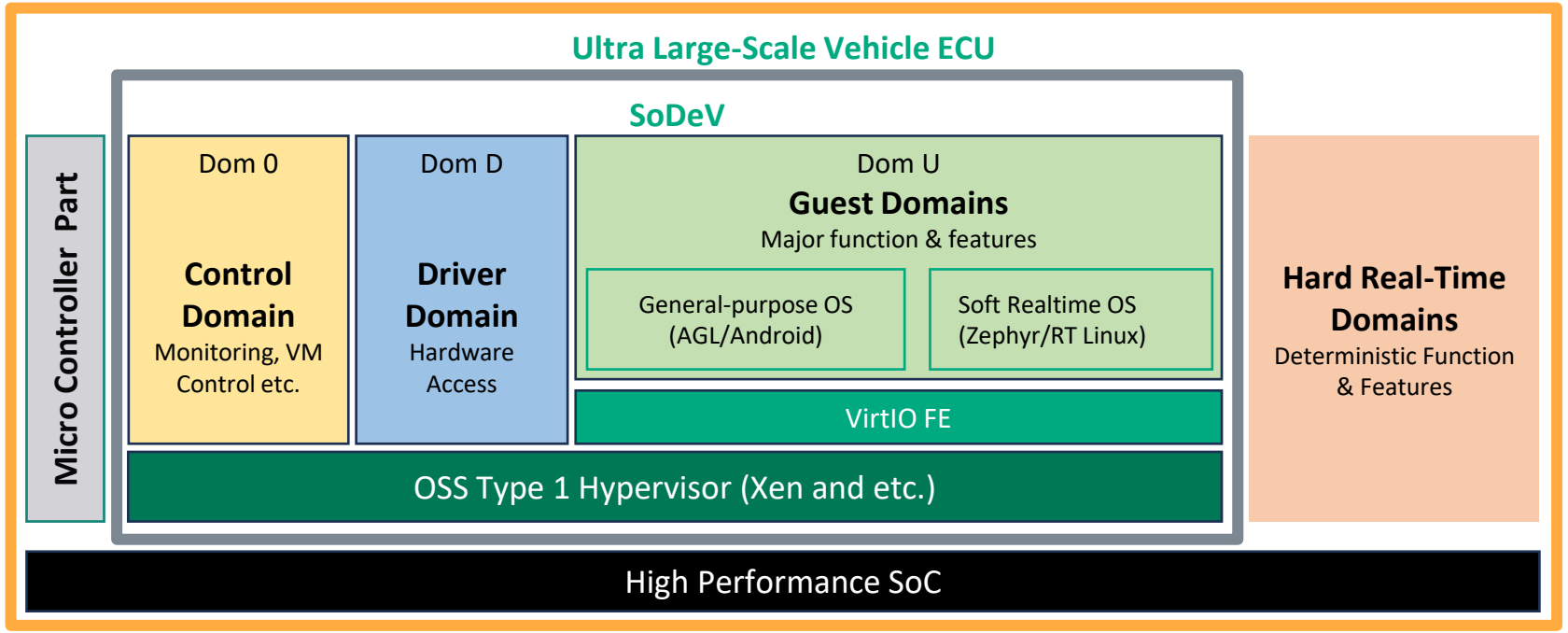
Hardware decupling Layer : VirtIO

- AGL focuses on achieving hardware decoupling.
- Our proposal is
  - 'VirtIO is a common and generic hardware interface in automotive systems'.
- Keypoints:
  - VirtIO is a standardized interface for efficient I/O (input/output) in virtualized environments.
    - It makes fast communication between the Guest VM and host.
    - The guest sends requests to the host through virtio instead of using real devices.
  - It's an open standard. Some open-source software supported it.
    - Linux Kernel, Xen, KVM, etc...
- It achieves software reusability between automotive systems.



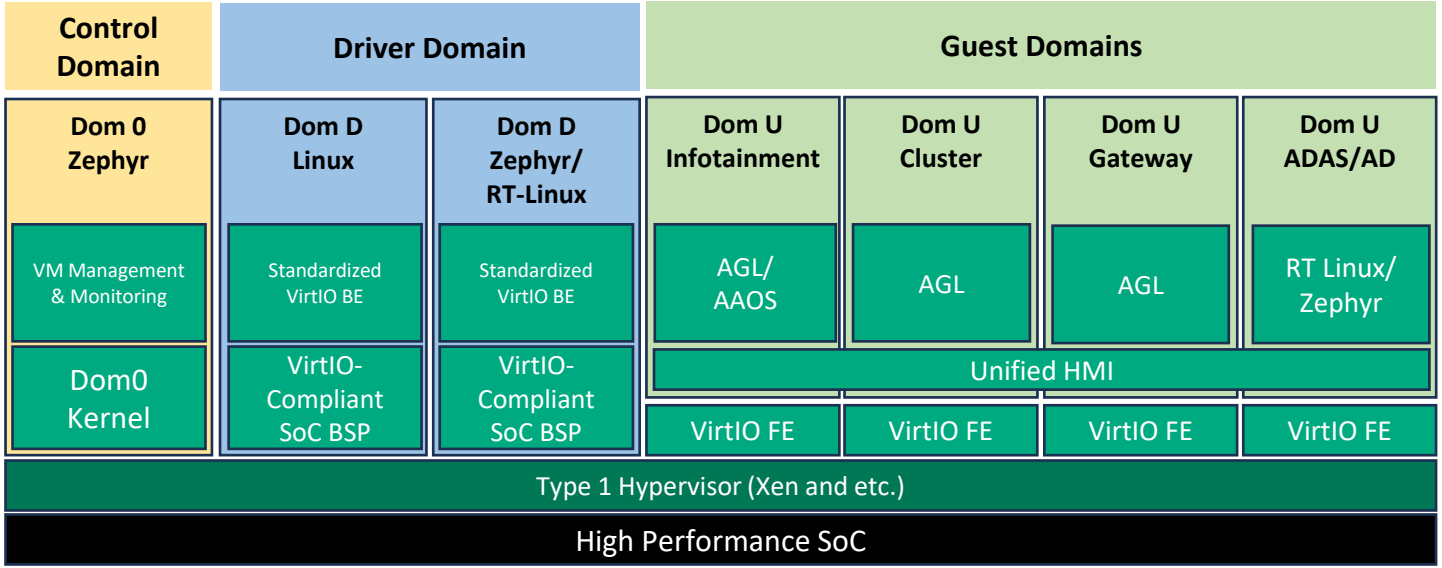
SoDeV Development Scope

- SoDeV covers wide functional domain of Ultra Large-Scale Vehicle ECU.
  - Inside/outside microcontroller and hard real-time function is out of scope.
  - SoDeV development focuses to hypervisor-based infrastructure and that guest examples.



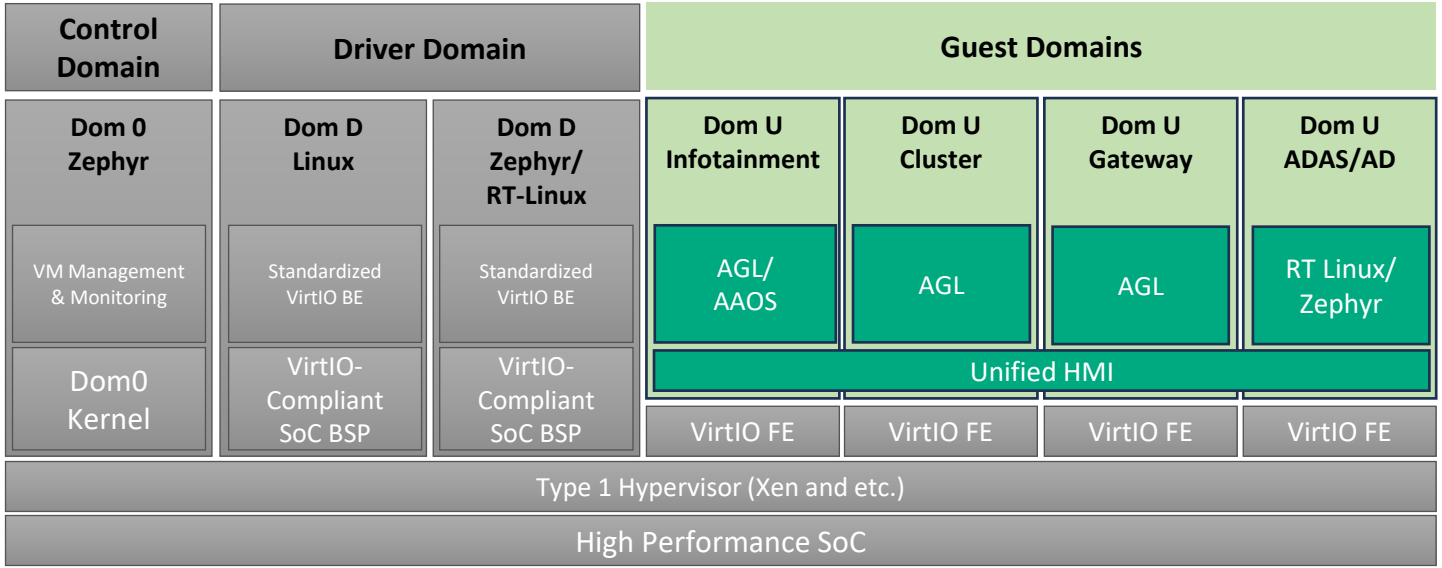
SoDeV Architecture Overview

- SoDeV development focuses on hypervisor-based infrastructure and guest examples.
  - This infrastructure is built on a Type 1 hypervisor. The 1st reference uses Xen.
  - It has a device-specific domain isolated from the hardware-decoupled domain.



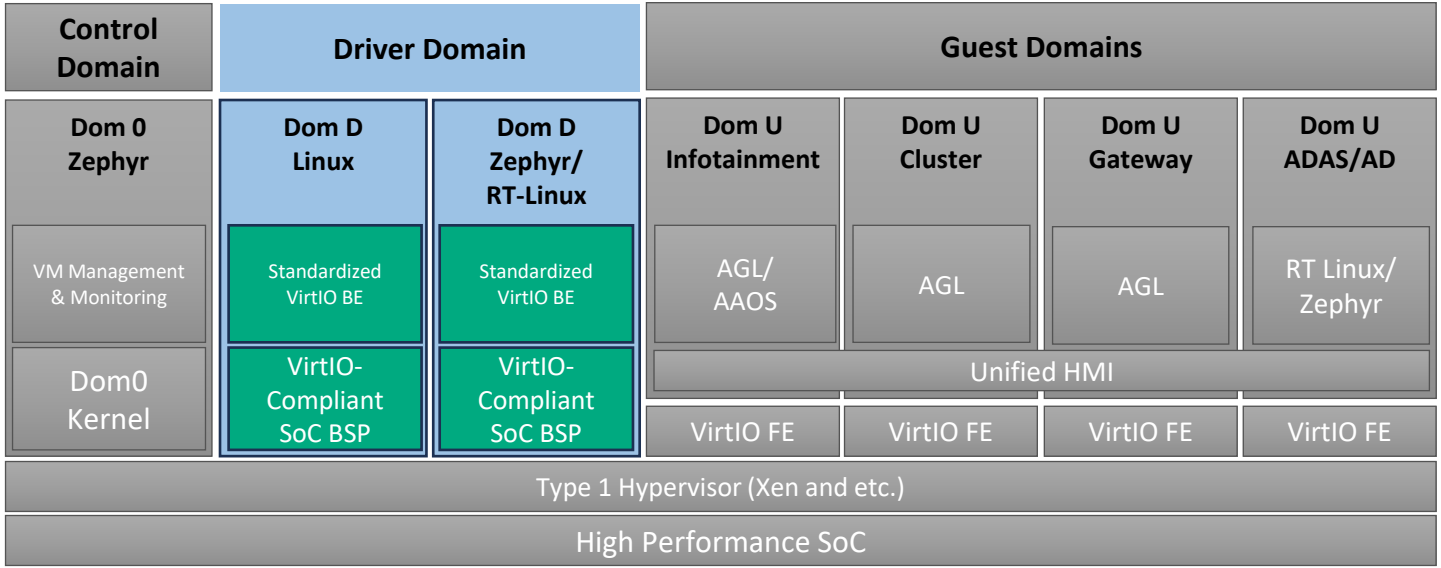
SoDeV Architecture

- Supporting use cases.
  - Infotainment systems, such as Android Automotive OS (AAOS) and AGL IVI, are supported.
  - As well as AGL Instrument Cluster.
  - Architectural design aims to follow other use cases such as AD/ADAS and Connected Gateway.



SoDeV Architecture

- Point: Two isolated device-specific domains.
  - One is the flexible domain. It follows various devices to support various use cases. It is built on Linux.
  - Another one is the real-time domain. It follows limited devices to support real-time use cases. It is built on RTOS or RT-Linux.



SoDeV Recap

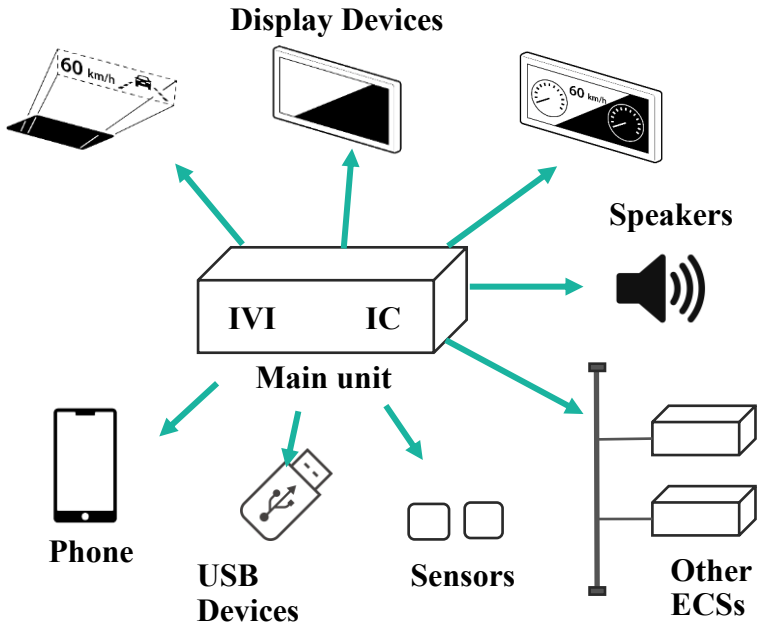
- SoDeV is a reference SDV platform that is accelerated by open-source technology.
- SoDeV focuses on
  - Hardware decoupling to realize software reusability.
    - It's accelerated by VirtIO infrastructure.
  - VM-based architecture to realize system scalability.
    - It's accelerated by hypervisor infrastructure.
  - Integrate deterministic and non-deterministic software into one system using VMs.
    - Remote rendering capability for deterministic domain accelerated by Unified HMI.
    - Accelerate some SDV trends such as cloud development and others.
- These are the research points.

Existing work for AGL System

AGL Instrument Cluster Expert Group

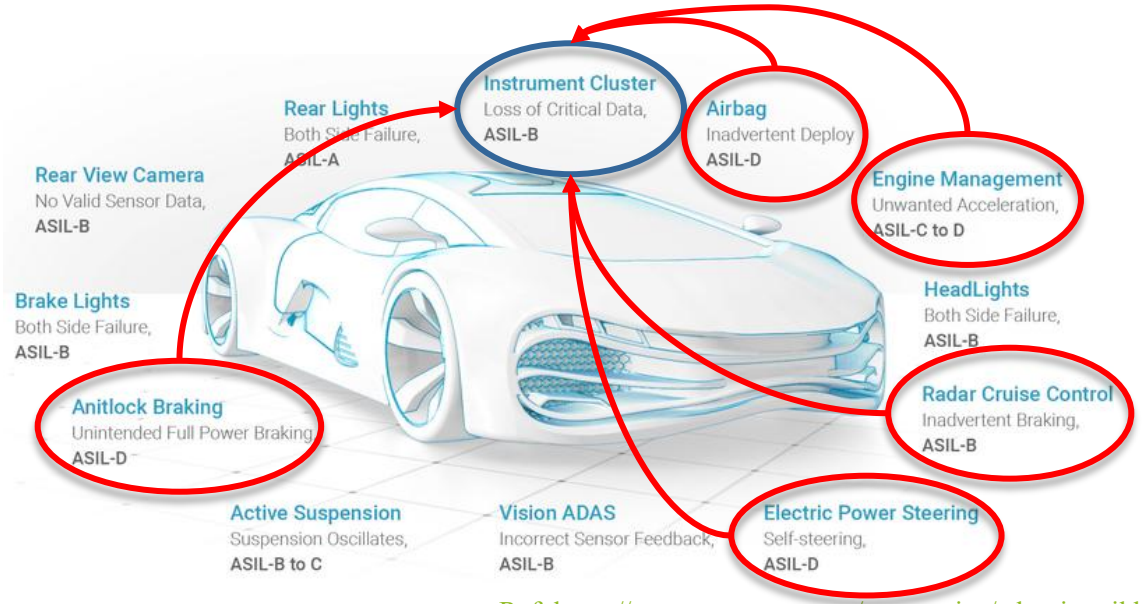
- AGL Instrument Cluster Expert Group launched at 2019

- Target
  - Instrument cluster and IVI into one system.
  - Platform architecture aims to support functional safety.
  - Create a base platform built from open-source technology.
    - It is not supporting functional safety because it deepens on final system design.



Safety for Instrument Cluster

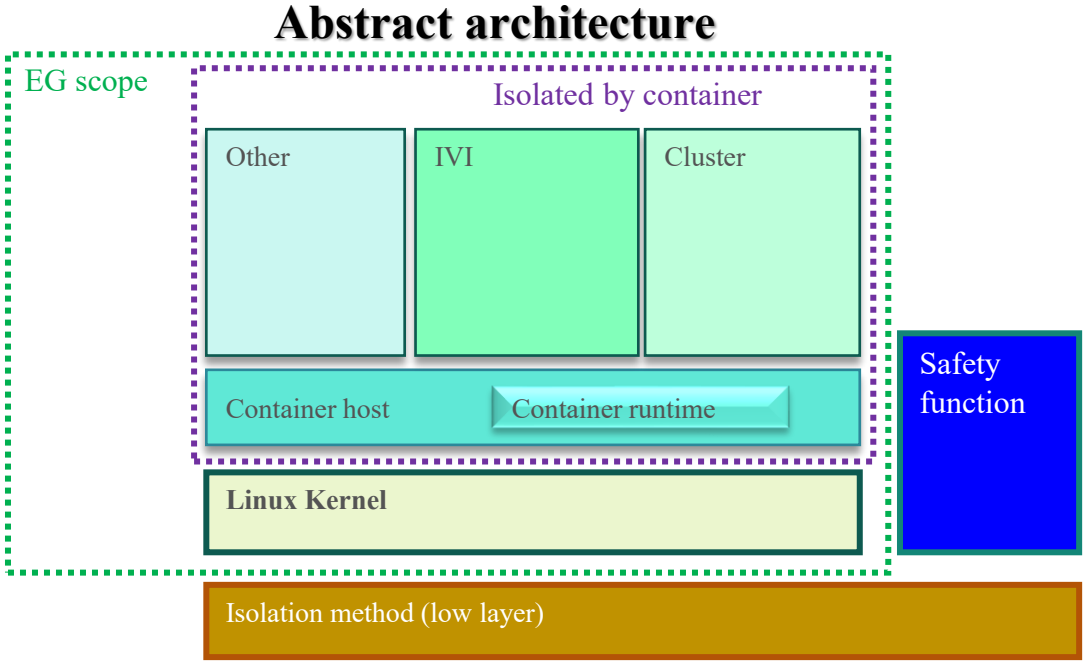
- Typically instrument cluster assigned ASIL-B.
  - Includes telltale function that is assigned ASIL-B.
    - ASIL-B was decomposed from other units.
  - Existing instrument cluster does not have ASIL from own functions.



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

Instrument Cluster Expert Group Architecture

- One more isolation method from safety island architecture
  - Safety function is built on Safety island.
  - Main function is built on Linux with Linux container.



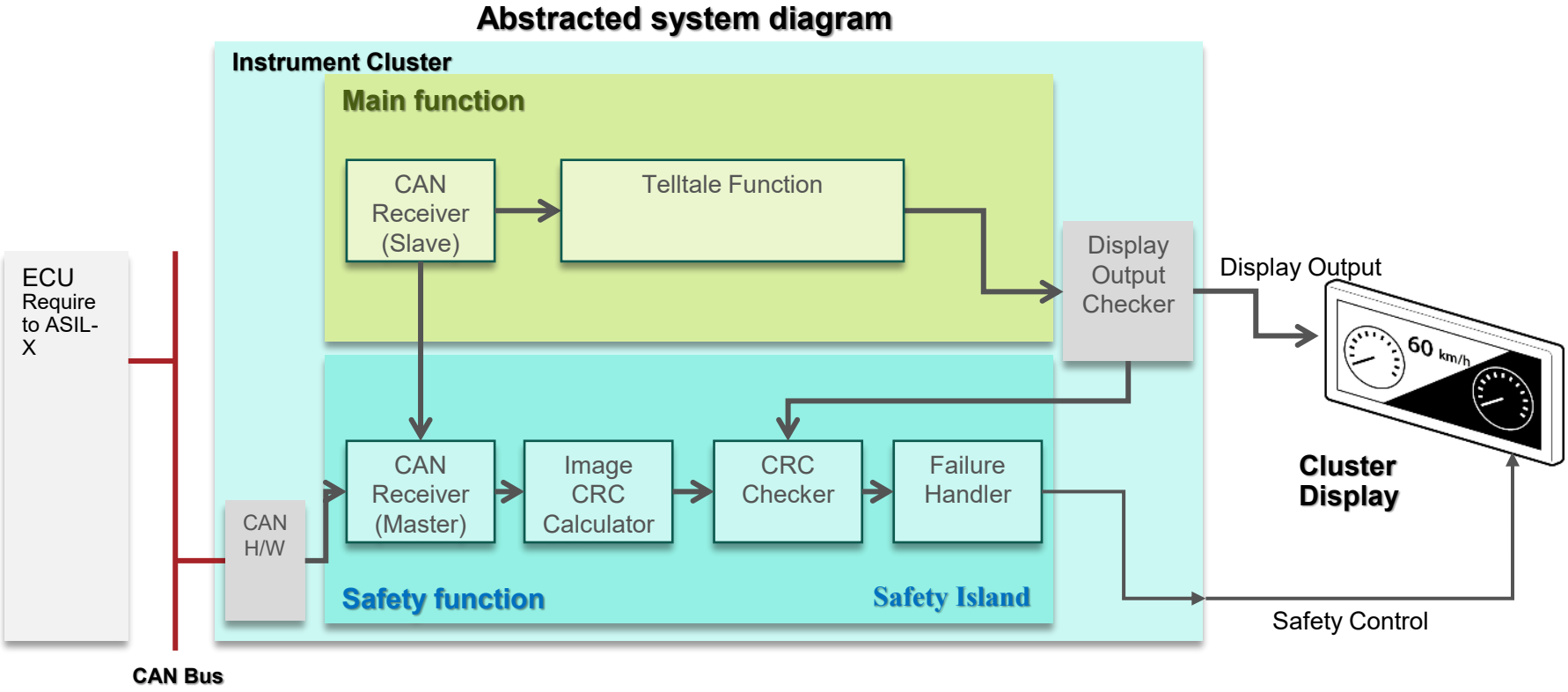
Example : Telltale

- Example system
  - ASIL-X ECU and Instrument Cluster are connected by CAN.
  - This system must avoid miss notification by telltale. It assigns ASIL-B.
  - Cluster outputs Safety Control signal separately from output to Display.
  - When safety control is enabled, Cluster display show the failure information.

Abstracted system diagram

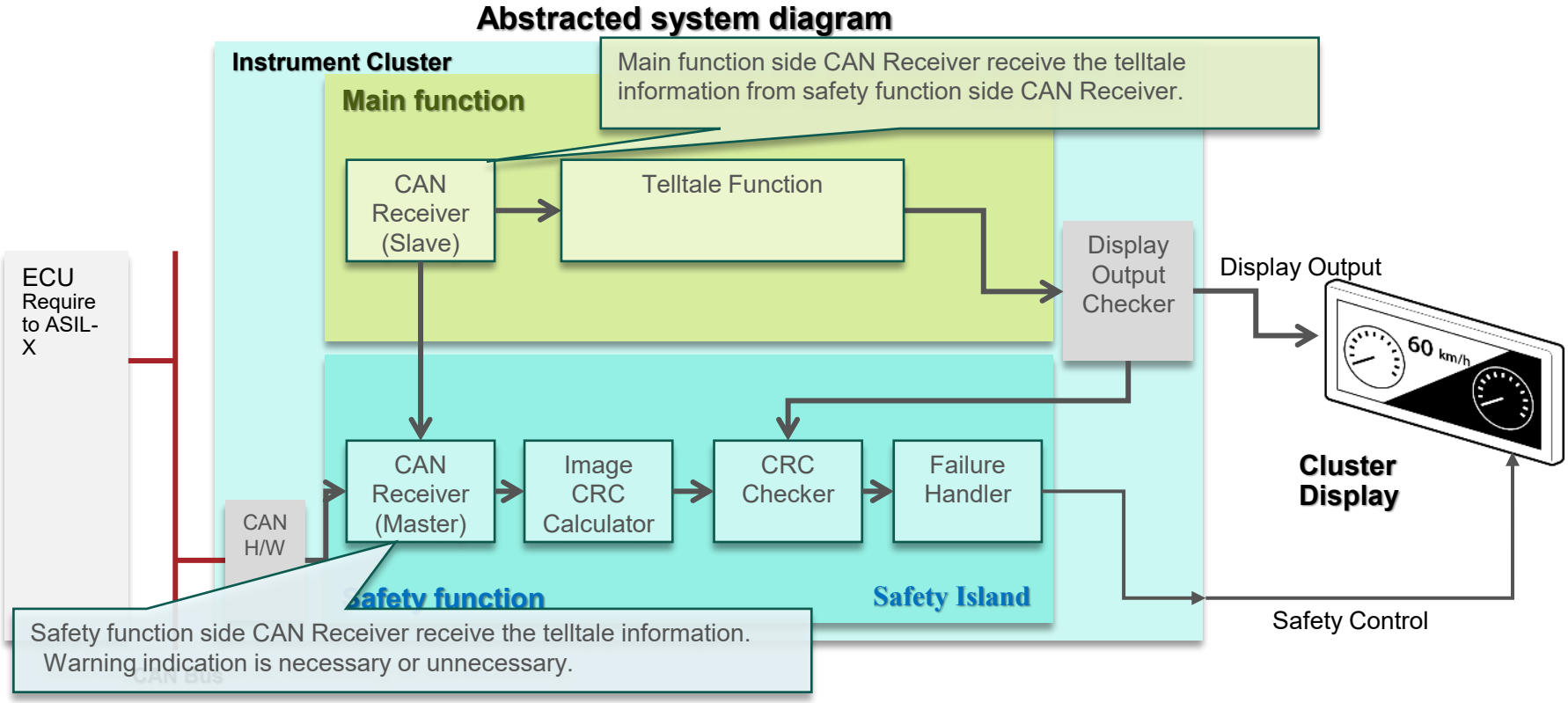
Example : Telltale

- More detail of system block diagram



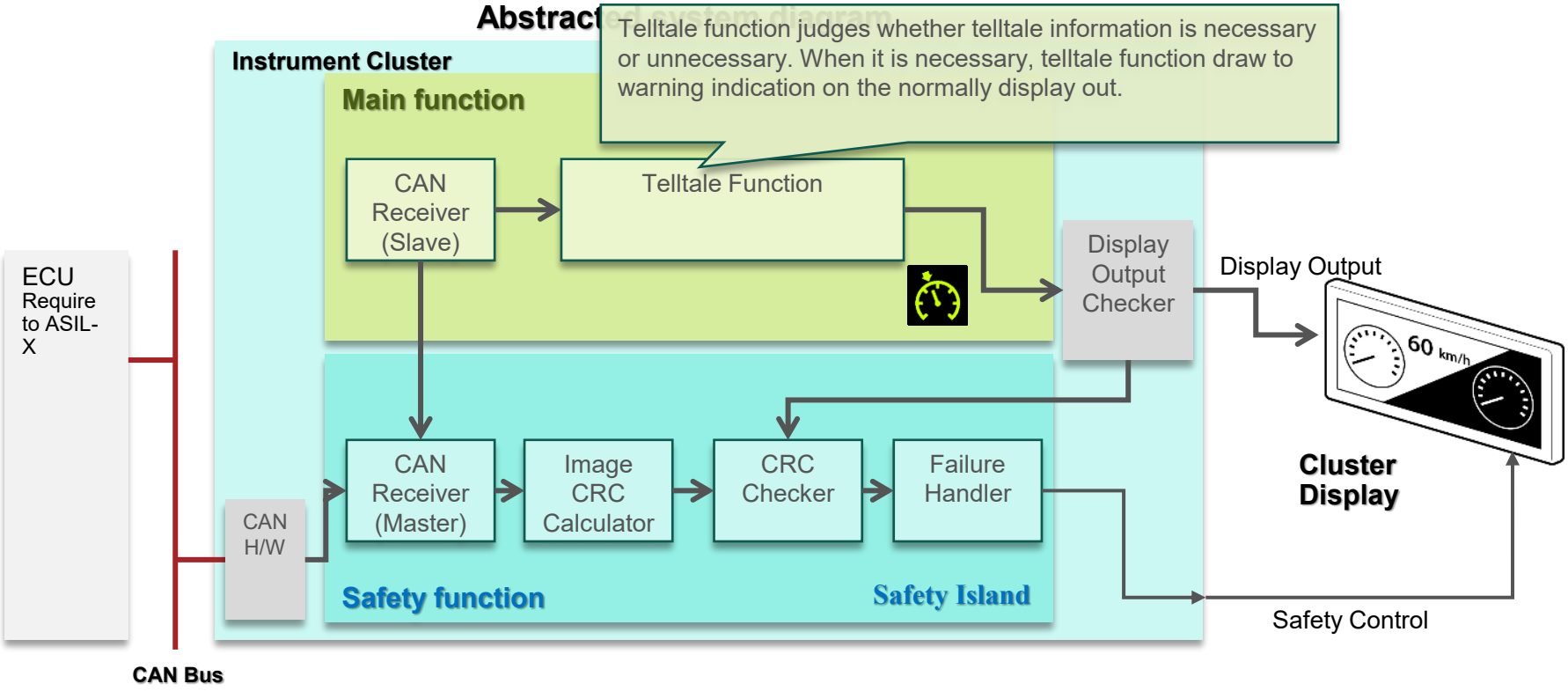
Example : Telltale

- More detail of system block diagram



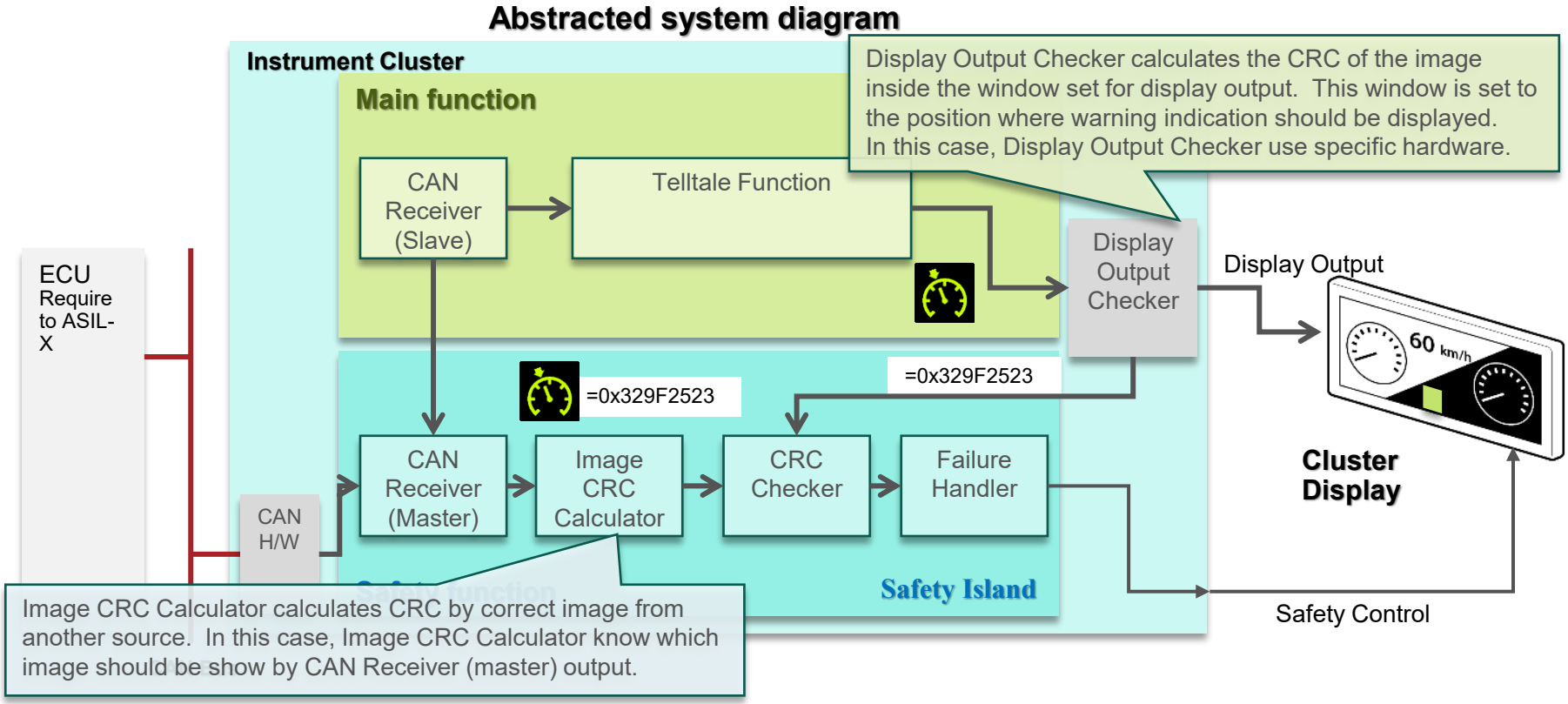
Example : Telltale

- More detail of system block diagram



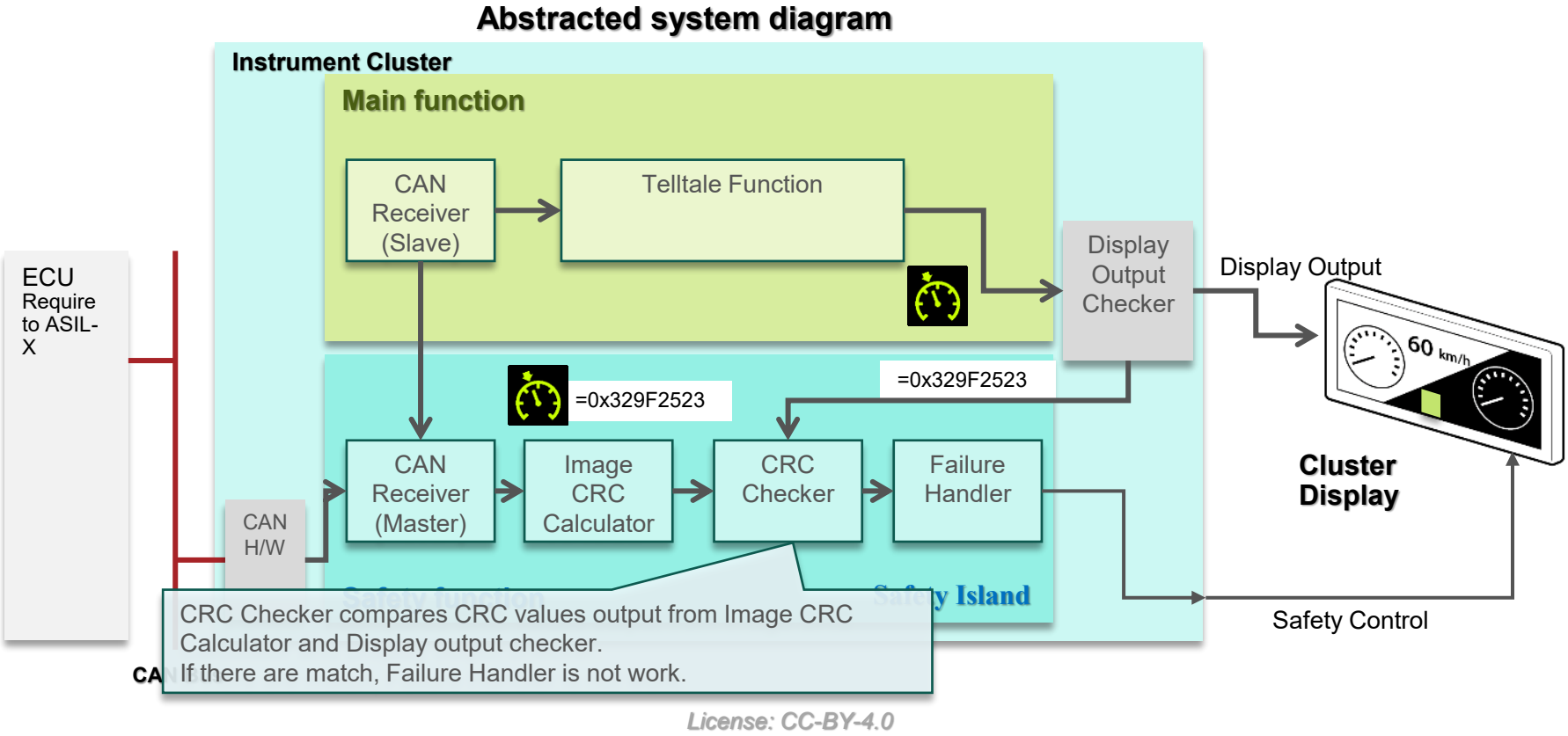
Example : Telltale

- More detail of system block diagram



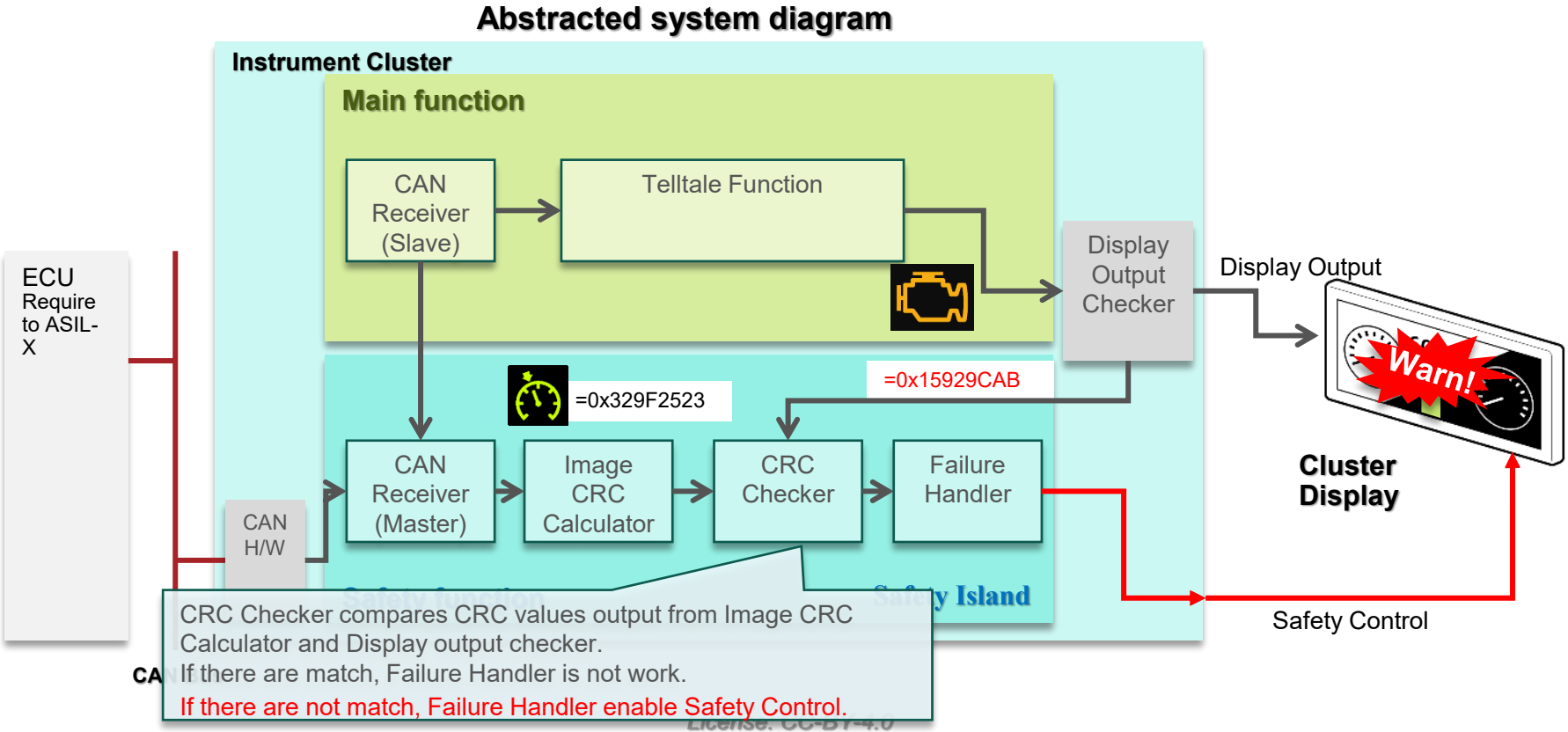
Example : Telltale

- More detail of system block diagram



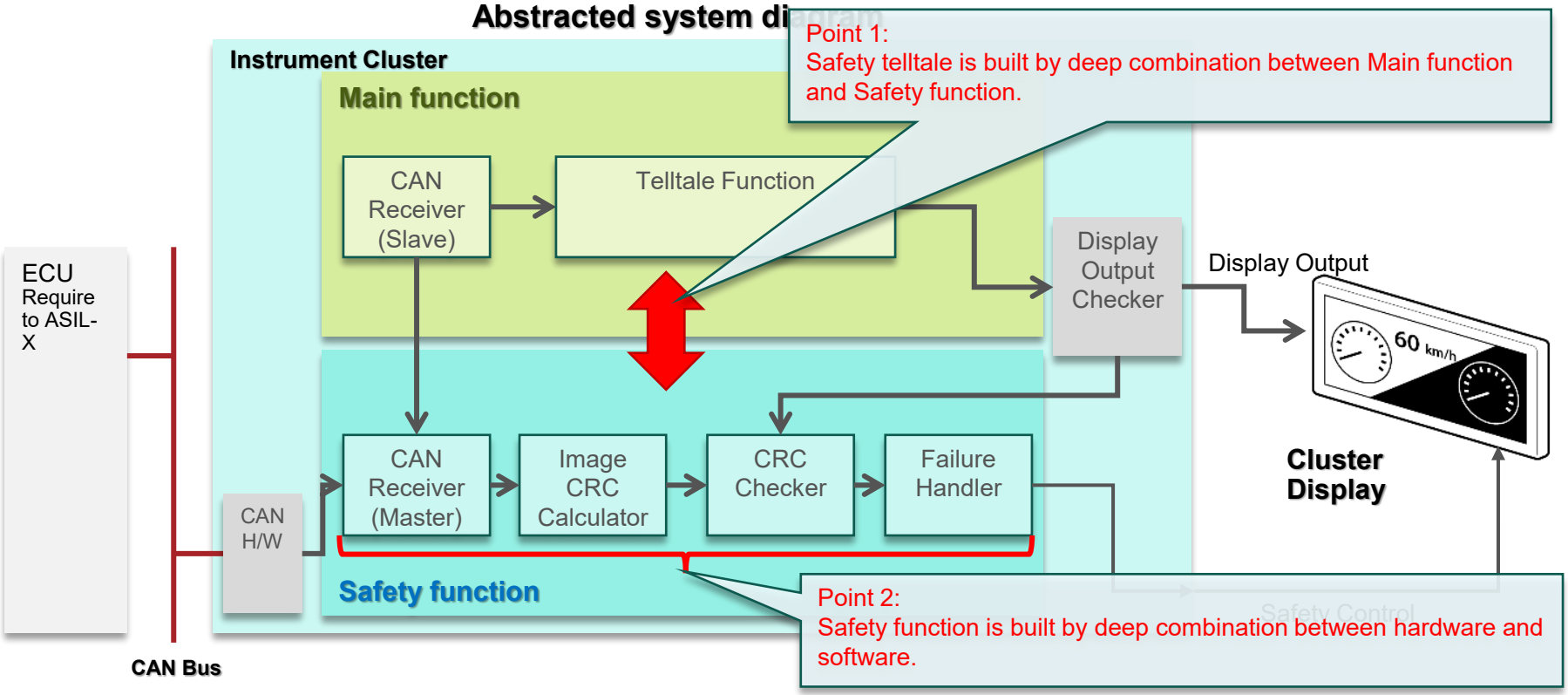
Example : Telltale

- More detail of system block diagram



Example : Telltale

- More detail of system block diagram



Instrument Cluster Recap

- Instrument Cluster EG defined a two-layer isolation architecture.
  - One approach is an isolation between QM blocks.
  - The functional safety approach is a safety island architecture.
    - It achieves FFI(Freedom From Interference) that supports isolation between the ASIL part and the QM part.
    - On the other hand, FFI does not realize the hardware decoupling.
- The telltale design is built by two software blocks that depend on the hardware design.
  - In case of the IC EG architecture, it is not intended for full hardware decoupling.
    - It can achieve the telltale.
  - SoDeV architecture aims to achieve full hardware decoupling.
    - It's a big gap.

Consideration for Telltale on SoDeV

What is issue

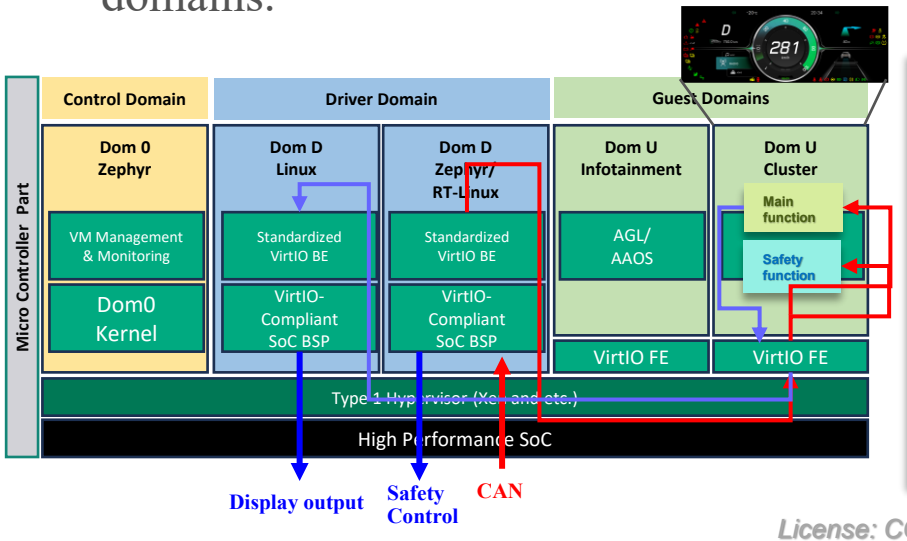
- SoDev aims to create a reference SDV platform.
  - That key feature is a hardware decoupling for guest domains using VirtIO infrastructures.
  - Some guest domain features require to safety such as the instrument cluster
    - Includes a telltale function.
- Consideration point.
  - Existing telltale design couple between software and hardware.
  - How to balance safety and hardware decoupling?

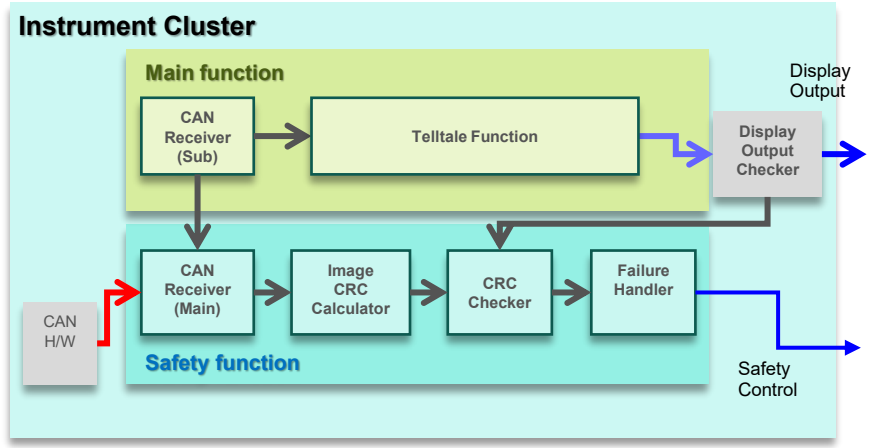
Attention

- Xen hypervisor and Zephyr RTOS is now working to safety certification.
  - When Xen and/or Zephyr get an ASIL-B certification, is a telltale feature on SoDeV safe and aligned with ASIL-B?
  - This answer is No.
    - It supports a part of ASIL-B safety requirements of instrument cluster, not all.
- If SoDeV based system need to align ASIL-B, we shall design a system to align ASIL-B.
- In this discussion, assume to;
  - Xen achieves an ASIL-B aligned FFI.
  - Zephyr achieves an ASIL-B aligned runtime environment.

Design No.1

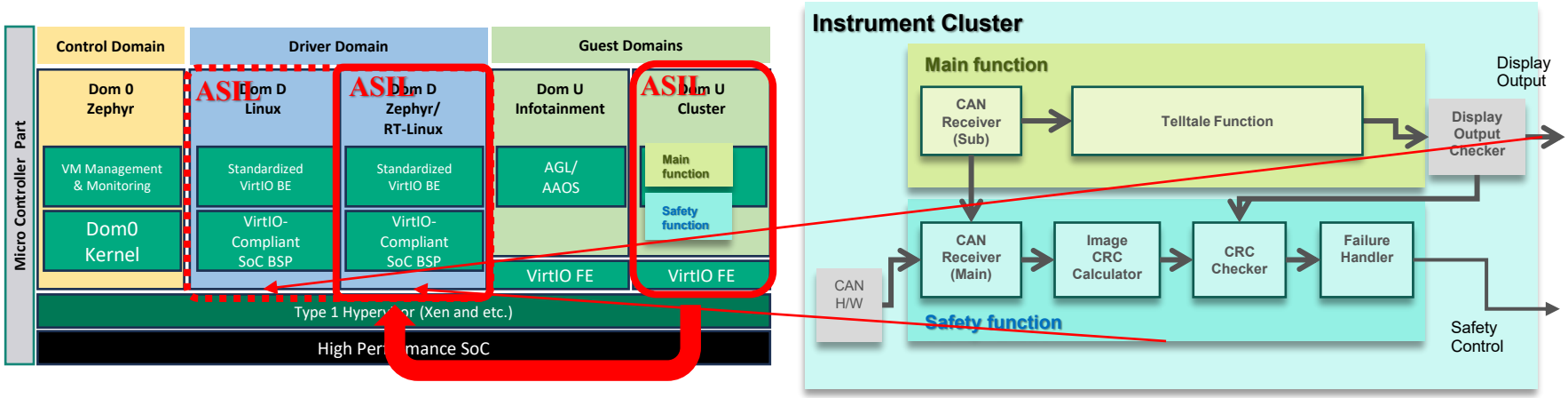
- All of the instrument cluster functions stay on one DomU.
  - It is isolated from other DomU.
  - The safety function is built on virtualization layers.
    - The safety function depends on the DomD safety.
  - Do not need telltale special interface between the instrument cluster DomU and other domains.





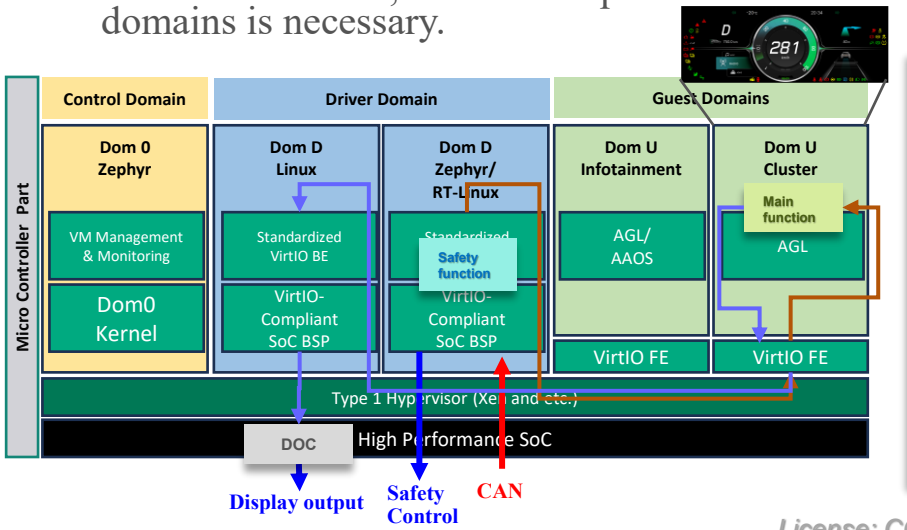
Design No.1

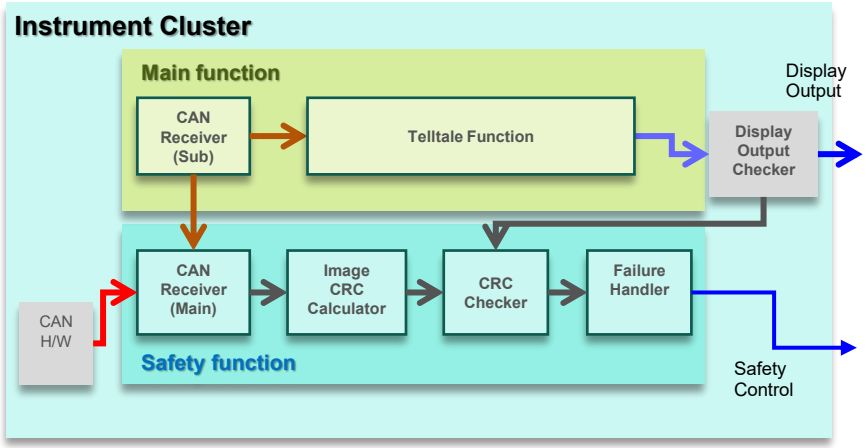
- All of the instrument cluster functions stay on one DomU.
  - Strong point:
    - All instrument cluster software will be easy to upgrade.
  - Weak point:
    - All instrument cluster software shall be aligned to ASIL-B in the worst case.
      - It includes a main function and safety function.
    - Both DomD shall be alien to ASIL.
      - It includes a route of the telltale showing and failure path.



Design No.2

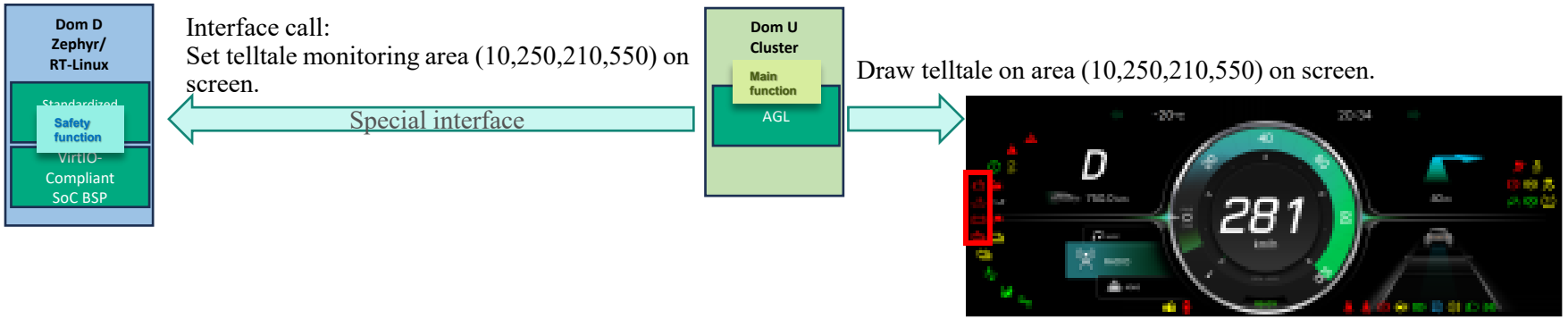
- Telltale drawing function stays on DomU.
  - It's not required ASIL in this model case.
- Telltale monitor function stays on DomD for safety use case.
  - It is isolated from other domains.
  - The safety function is not built on virtualization layers.
  - The safety feature depends on only DomD for safety.
  - On the other hand, the telltale special interface between the instrument cluster DomD and other domains is necessary.





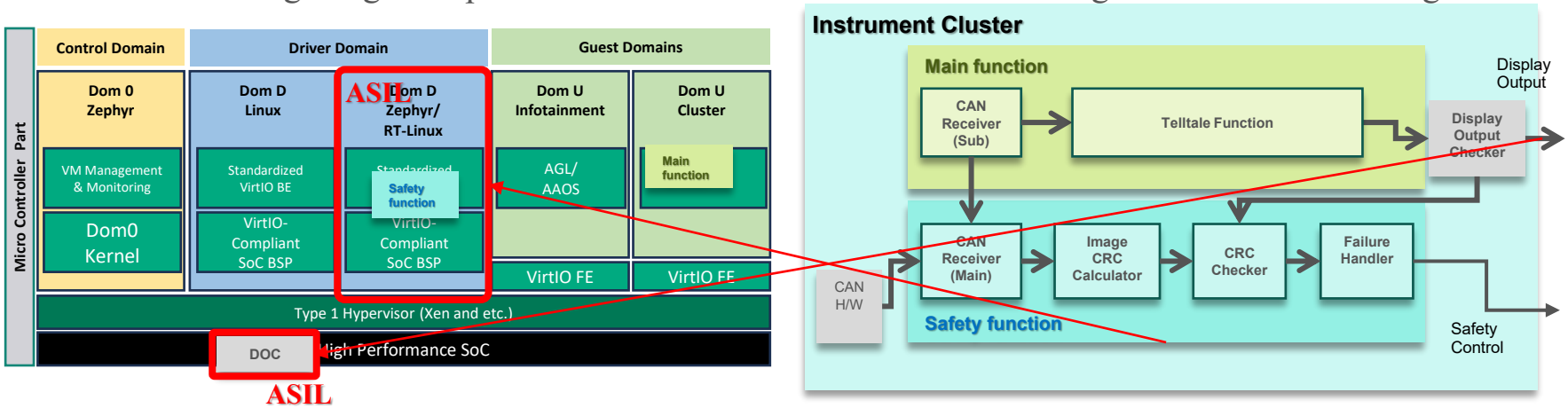
Design No.2

- Telltale drawing function stays on DomU.
  - It's not required ASIL in this model case.
- Telltale monitor function stays on DomD for safety use case.
  - It is isolated from other domains.
  - The safety function is not built on virtualization layers.
  - The safety feature depends on only DomD for safety.
  - **On the other hand, the telltale special interface between the instrument cluster DomD and other domains is necessary.**



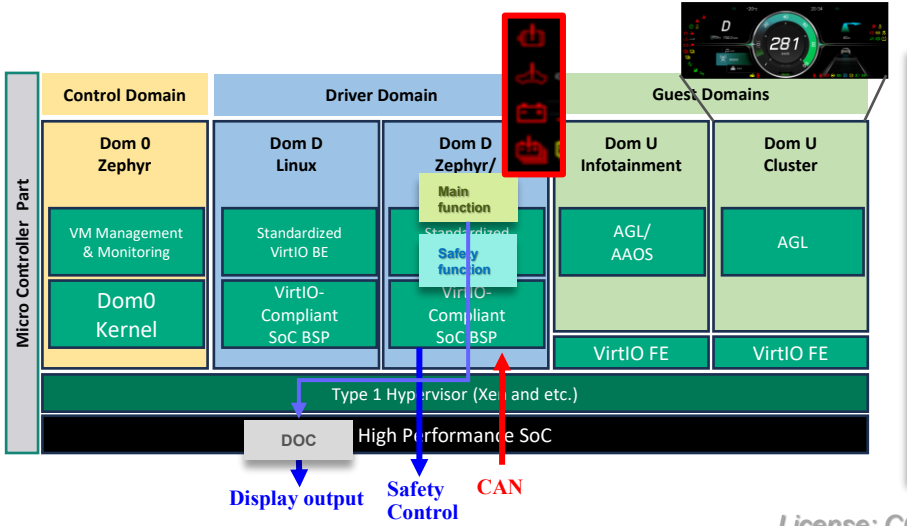
Design No.2

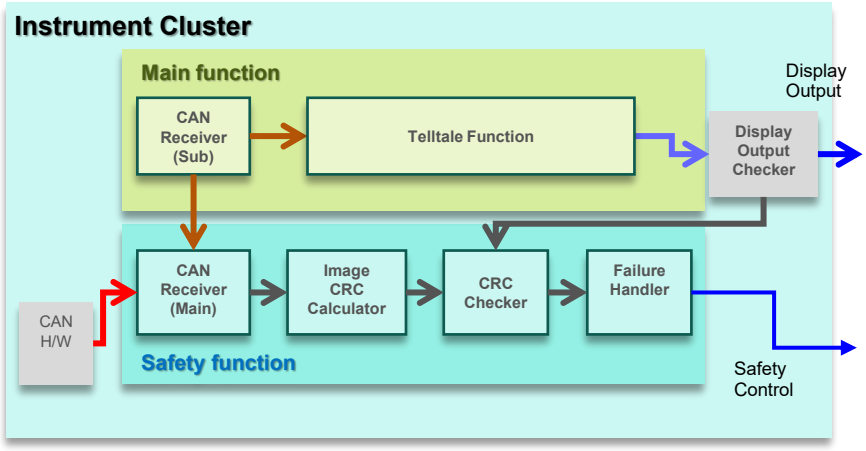
- Telltale monitor function stays on DomD for safety use case.
  - Strong point:
    - Possible to reduce the ASIL-related part from design 1.
  - Weak point:
    - Necessarily for the telltale special interface between the instrument cluster DomU and DomD for safety.
    - If we change the inside of the telltale drawing area of DomU, that effects to DomD for safety.
    - The drawing image and position must match between telltale drawing and telltale monitoring.



Design No.3

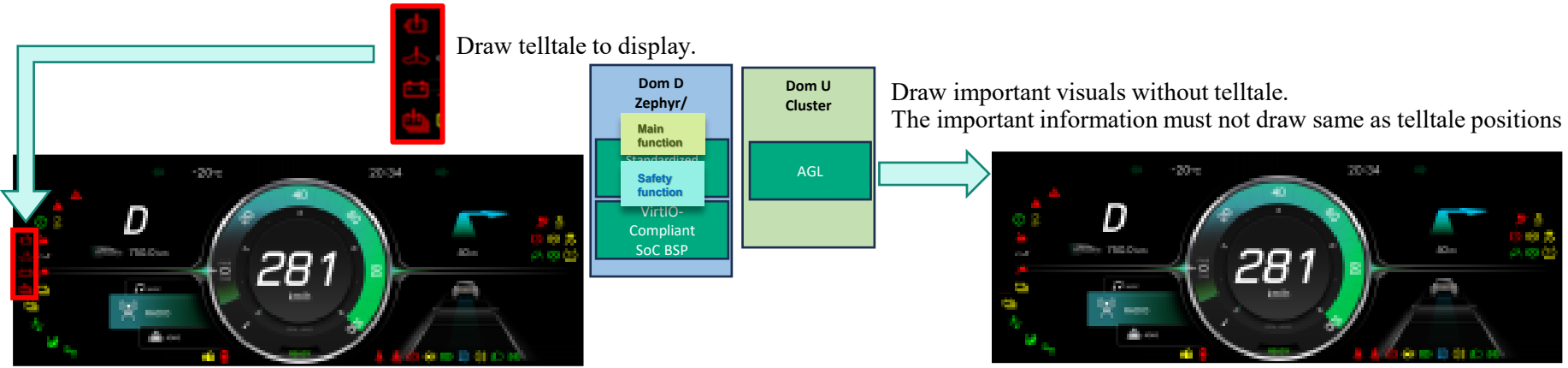
- All telltale functions stay on DomD for safety.
  - Forcefully overlay the telltale image including background on the DomU instrument cluster image.
  - Non telltale function in instrument cluster stays on DomU.
    - The telltale feature is isolated from other features and from other instrument cluster functions.
  - Instrument cluster function is decoupled from safety.
  - On the other hand, instrument cluster visual design is restricted by the telltale function on DomD for safety.





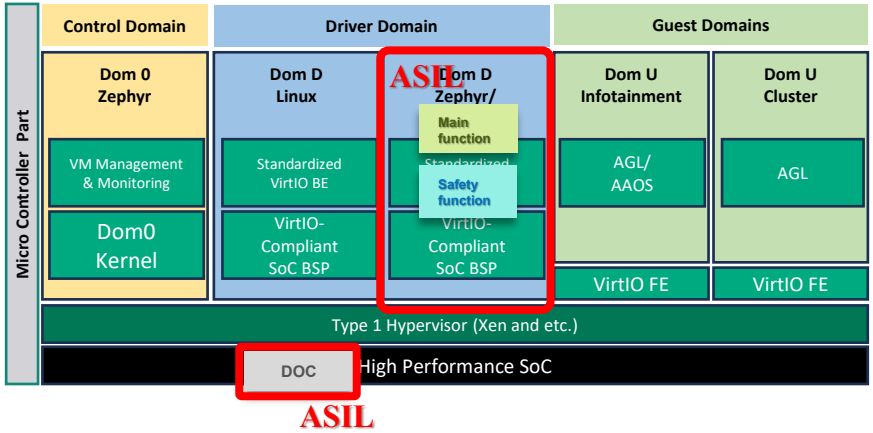
Design No.3

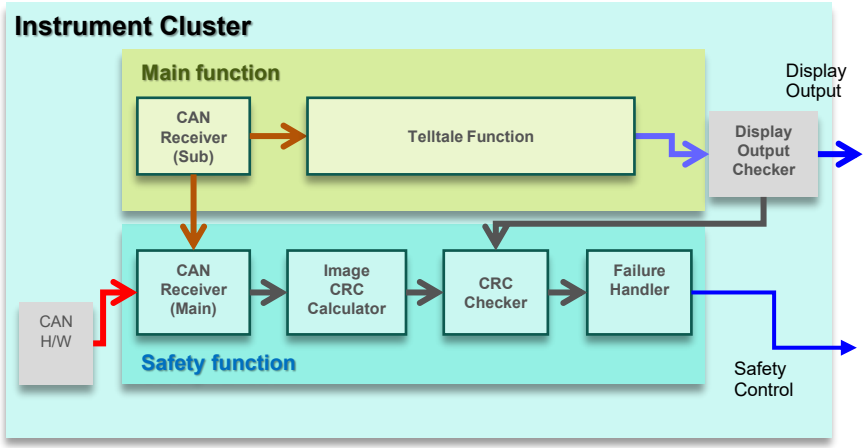
- All telltale functions stay on DomD for safety.
  - Forcefully overlay the telltale image including background on the DomU instrument cluster image.
  - Non telltale function in instrument cluster stays on DomU.
    - The telltale feature is isolated from other features and from other instrument cluster functions.
  - Instrument cluster function is decoupled from safety.
  - **On the other hand, instrument cluster visual design is restricted by the telltale function on DomD for safety.**



Design No.3

- All telltale functions stay on DomD for safety.
  - Strong point:
    - Possible to reduce the ASIL-related part from design 1.
  - Weak point:
    - The instrument cluster visual design must follow the telltale layout.
      - Get the telltale area from DomD for safety by the platform configuration interface.
      - The instrument cluster visual design on DomU is coupled with DomD safety.





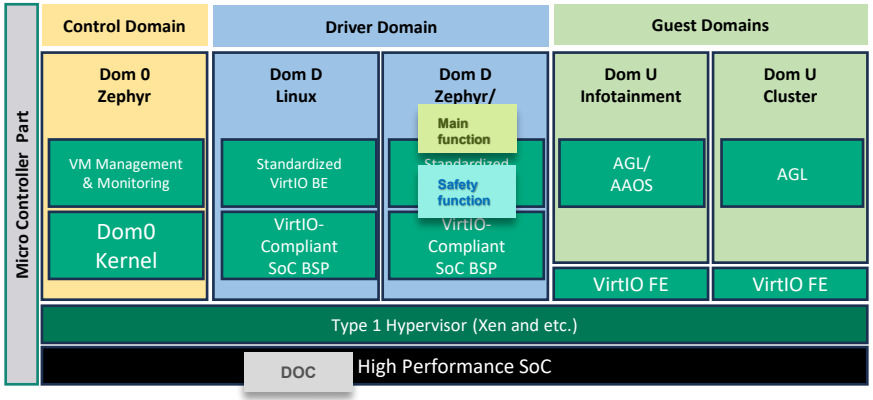
Other idea

- The telltale function stays on the microcontroller outside of SoDeV.
  - It uses a physical telltale. Example for LED.
  - It achieves full decoupling from others.

- Others??

Which Design is Better Balance?

- Which design is better balanced?
  - My answer is Design No.3, Instrument cluster function is decoupled from safety.
- Because….
  - Instrument cluster's physical display does not use the same aspect ratio and resolution.
  - When SoDeV aims to decouple hardware, these physical display variation issues shall be resolved. This issue is similar to design No3 weak point.

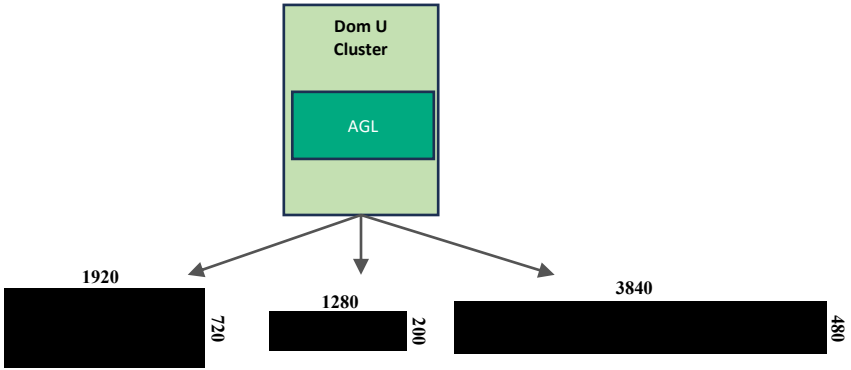


Which Design is Better Balance?

- DomU must adopt to physical display, even if it used VirtIO based hardware decoupling.
  - Instrument cluster visual must adopt every.
- If SoDeV adds to support Design No3 telltale, that extention is a part of a display variations.
  - **No big impact for SoDeV architecture to adding a telltale support.**

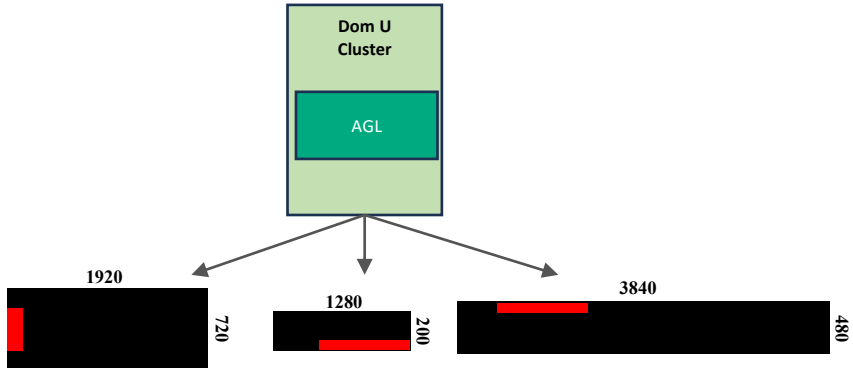
Situation of without telltale.

Need to follow display variation.



Situation of with telltale.

Need to follow display variation with telltale.



Conclusion

Conclusion

- Reviewed for SoDeV architecture and mind, recapped for AGL Instrument Cluster Expert Group activity for the telltale feature.
- These are the starting points of consideration for how to balance safety and hardware decoupling in SoDeV.
- Showed three designs to discuss safety and hardware decoupling balancing.
- Proposed the candidate design for the telltale feature.
  - It's only one use-case.
  - Need to discuss other use-cases in SoDeV.
- Will discuss other use-cases in AGL community.
- Welcome to your contributions for the SoDeV works!!

AUTOMOTIVE GRADE LINUX

Thank you!