Functional safety and Quality Management issues in AGL Instrument Cluster Expert Group

Open Source Software in Safety-Critical Systems Summit
October 31, 2019
Lyon, France

Naoto Yamaguchi
Software Fundamental Technology Group
Software Development Department I
Electronics Division
AISIN AW CO.,LTD.

Introduction to Who I Am

- Name: Naoto Yamaguchi
- Company: AISIN AW CO., LTD.

- Career
  - Received Doctor of Informatics in 2007 (Shizuoka-University).
  - Automotive RTOS platform software engineer since 2007.
  - Automotive Linux platform software engineer since 2011.
- My history of Open Source
  - Joined to AGL in 2013.
  - Member of AGL Instrument Cluster Expert Group since 2019.
  - Joined to ELISA in 2019.

Outline

- AGL Instrument Cluster EG
- Concept of Instrument Cluster EG
- Collaboration proposal from AGL
  - Function safety
    - Example use case : telltale
  - QM Isolation
    - What solution we need ?
- Conclusion

Outline

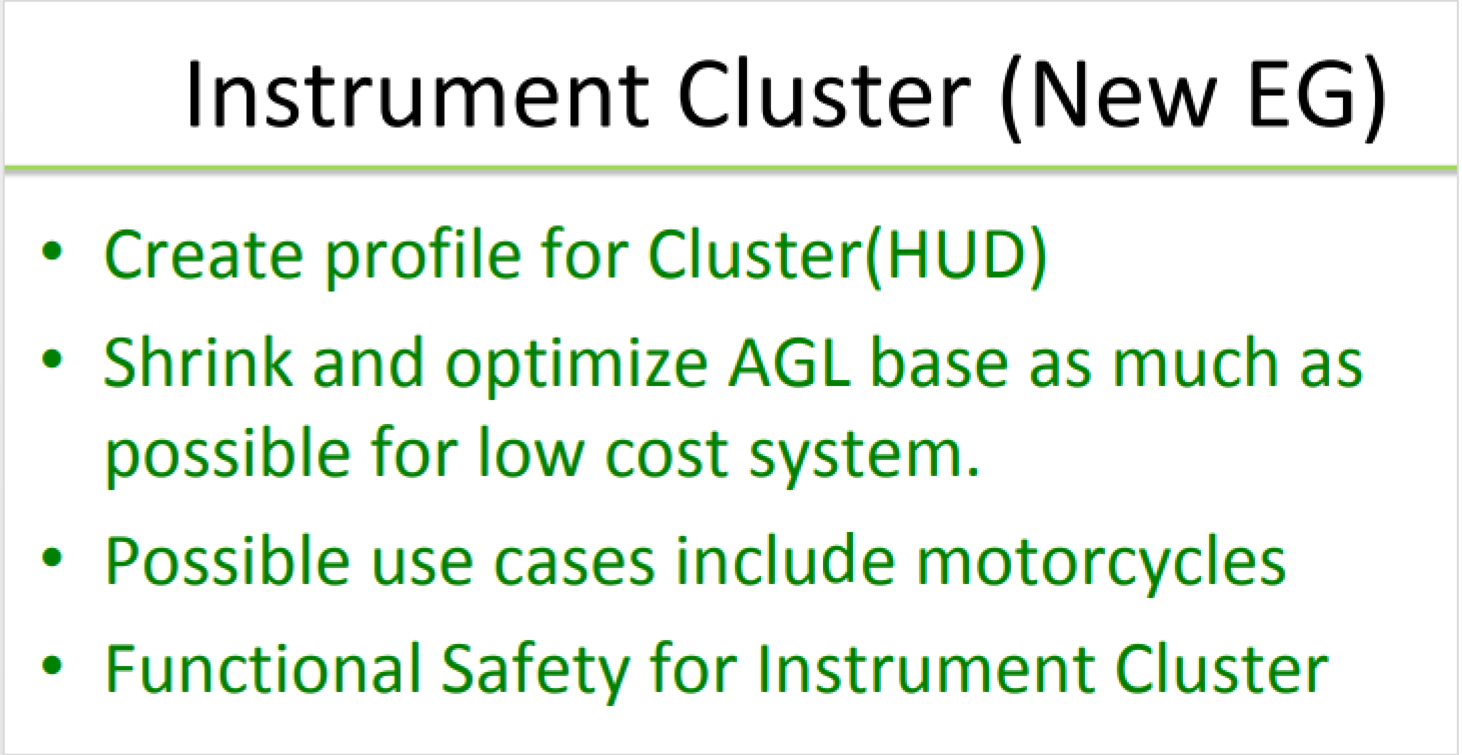
- **AGL Instrument Cluster EG**
- Concept of Instrument Cluster EG
- Collaboration proposal from AGL
  - Function safety
    - Example use case : telltale
  - QM Isolation
    - What solution we need ?
- Conclusion

What is AGL?

- Ref.
  - Offical website
    - https://www.automotivelinux.org/
  - Latest presentation
    - https://events19.linuxfoundation.org/events/agl-amm-eu-2019/program/schedule/
    - Introduction to Automotive Grade Linux - Walt Miner

What is IC(Instrument Cluster) EG?

- IC EG have started from March 2019



Instrument Cluster (New EG)

- Create profile for Cluster(HUD)
- Shrink and optimize AGL base as much as possible for low cost system.
- Possible use cases include motorcycles
- Functional Safety for Instrument Cluster

https://wiki.automotivelinux.org/_media/agl_roadmap_tokyo_2019_amm.pdf

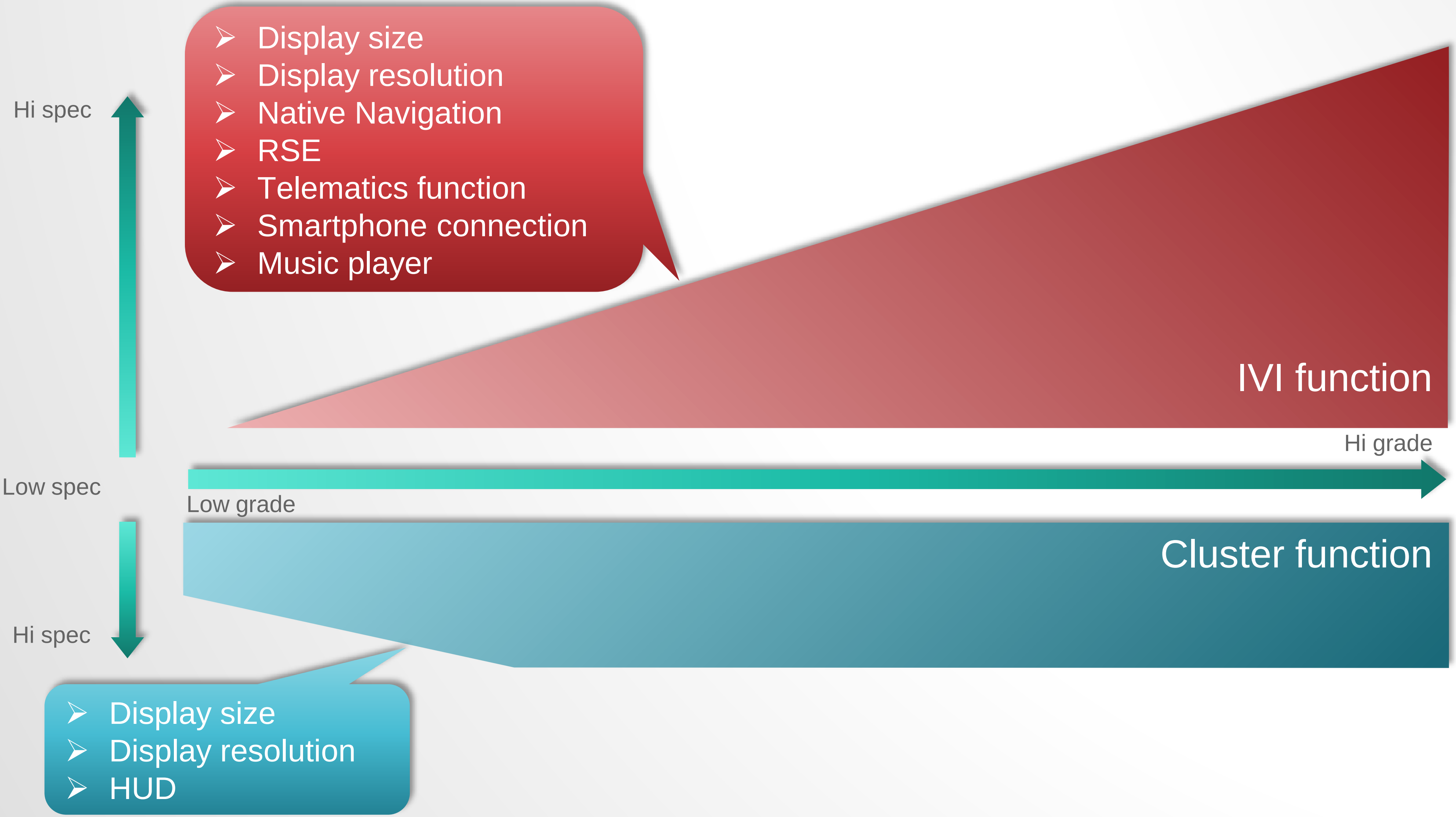
Member of IC expert group

Toyota, Honda, Mazda, Suzuki

ADIT, Denso, Panasonic, Continental, Bosch, Nipponseiki, Denso Ten, Aisin AW

Member of the ELISA project

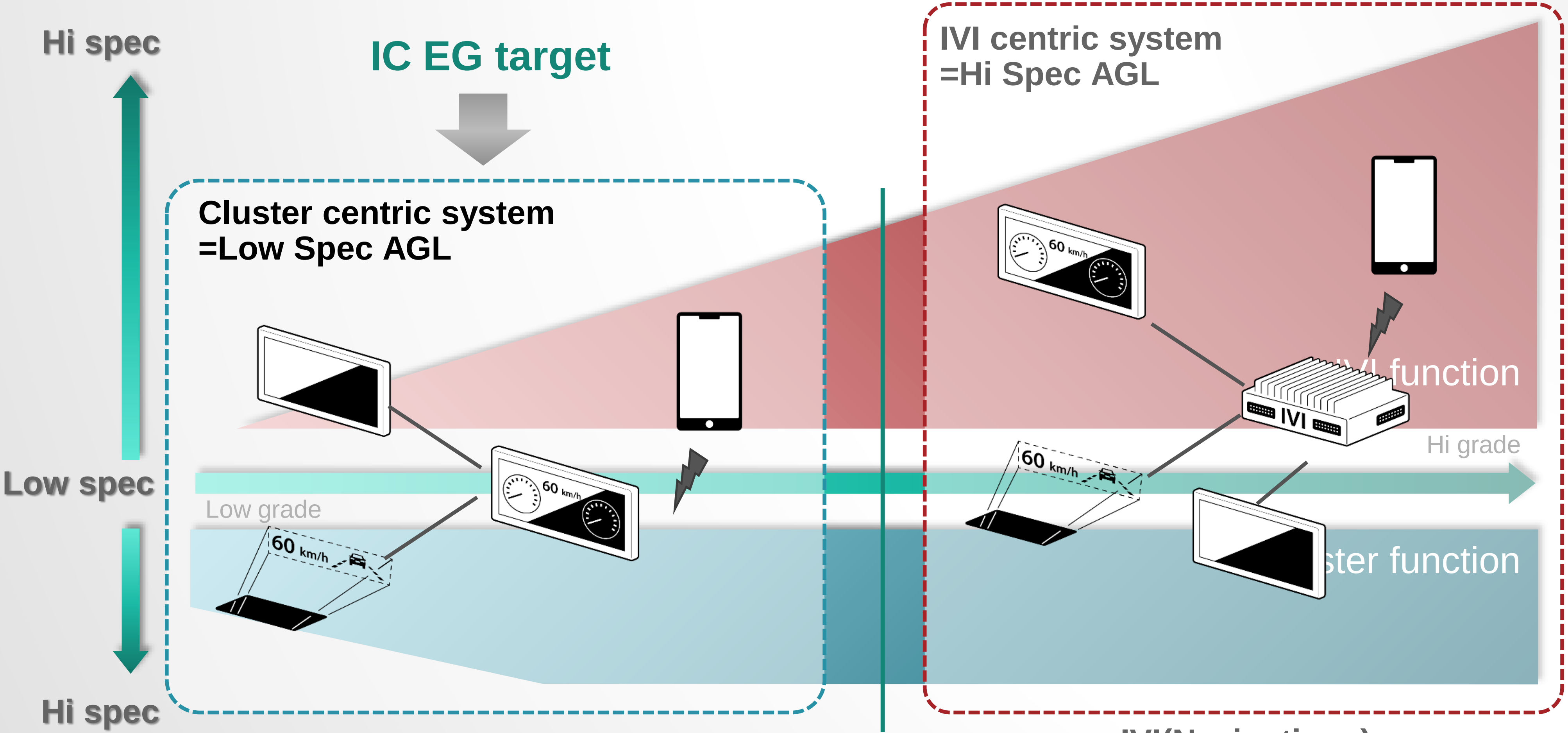
EG scope and system image



- Display size
- Display resolution
- Native Navigation
- RSE
- Telematics function
- Smartphone connection
- Music player

- Display size
- Display resolution
- HUD

EG scope and system image



IC EG target

Cluster centric system
=Low Spec AGL

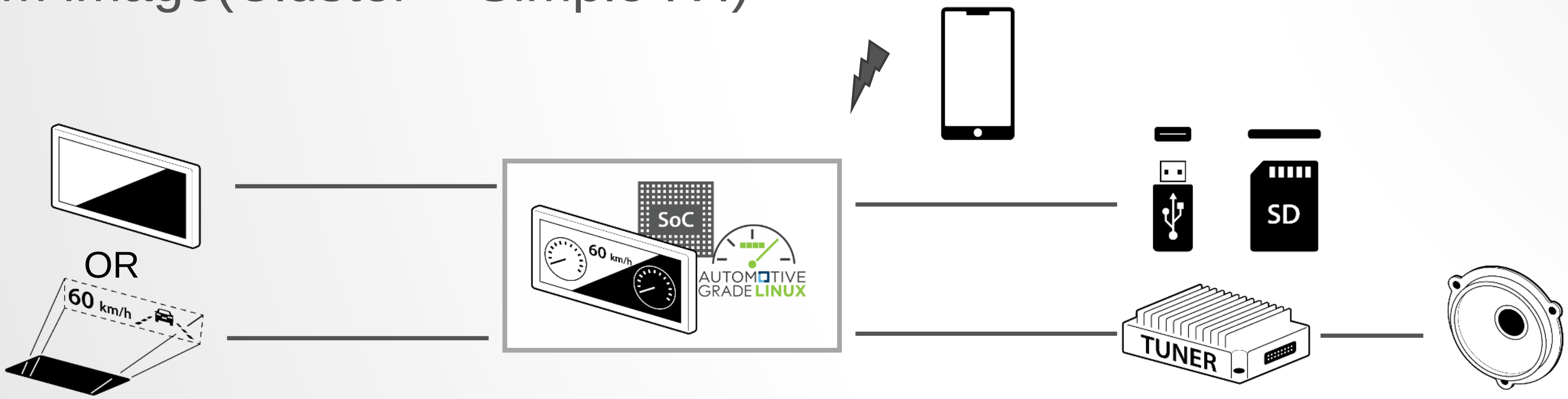
IVI centric system
=Hi Spec AGL

Cluster(HUD)
+
Smart phone connection(SDL..)
for small IVI function

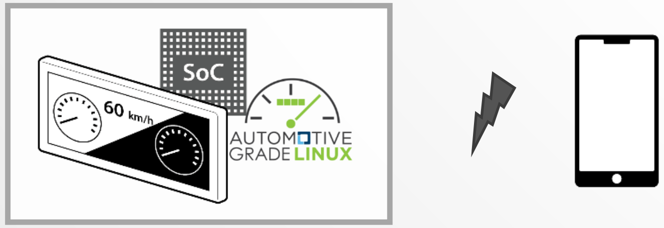
IVI(Navigation..)
+
Cluster(HUD)
+
Smart phone connection(SDL..)

What does IC EG aim?

- System image(Cluster + Simple IVI)



- Minimalize system image
  e.g.)Motorcycle or cluster alone use case



Motivation

- Create Cluster centric platform(Low Spec AGL)
  - We want to create a base platform for Cluster, not a platform based on conventional IVI.
  - There are different system requirements between IVI and Cluster.
  - e.g.)Functional safety, boot time etc…

Outline

- AGL Instrument Cluster EG
- **Concept of Instrument Cluster EG**
- Collaboration proposal from AGL
  - Function safety
    - Example use case : telltale
  - QM Isolation
    - What solution we need ?
- Conclusion

What are the product development issues?

1. **Quality and Robustness**
   - Functional safety is required.
   - Quality management is required.

Today presentation focus

2. **Lightweight**
   - Constraints on boot time are severe.
   - Current AGL stack is heavyweight.

Functional safety

Main function is the very function of our system

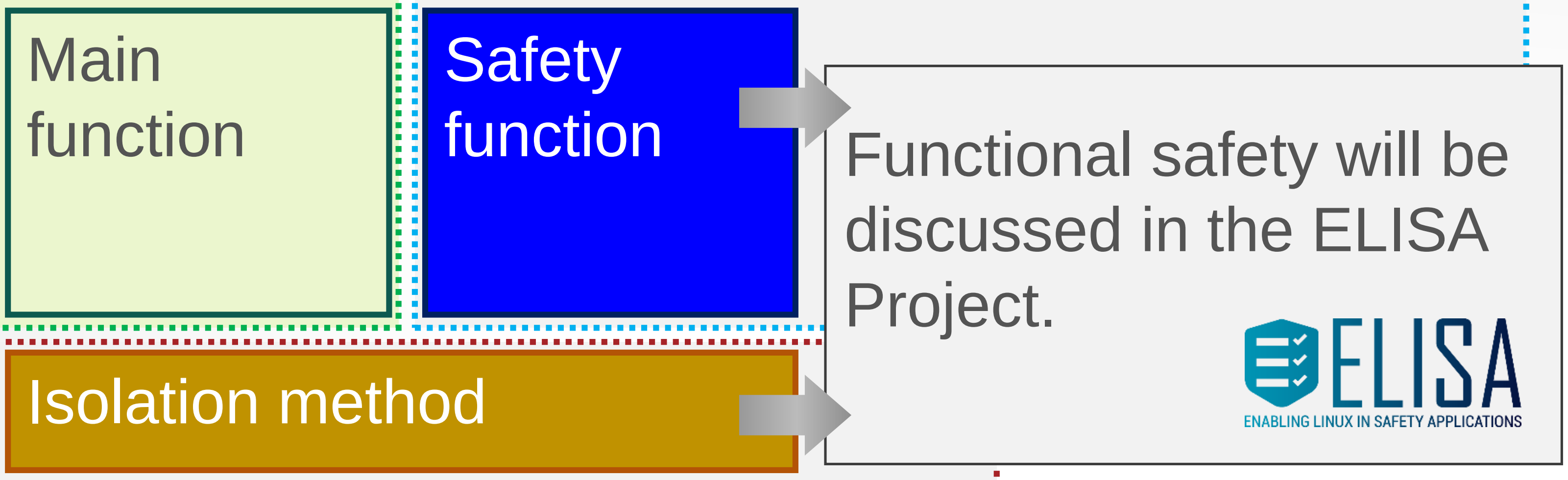
- Requires advanced quality management.
- Requires open innovation.
- Requires cyber security.
- Requires fast boot.
- Requires various functions.
- …

Main target of IC-EG EG

Safety function ensures vehicle safety

- What function does it include?
- Which OS do you use?
- Which communication method do you use?

Collaborate ELISA to find a solution.



Functional safety will be discussed in the ELISA Project.

ELISA
ENABLING LINUX IN SAFETY APPLICATIONS

Main function and safety function are isolated by isolation method.

- Hardware separation? Using hypervisor?

Collaborate ELISA to find a solution.

AUTOMOTIVE GRADE LINUX

What are the product development issues?

1. Quality and Robustness

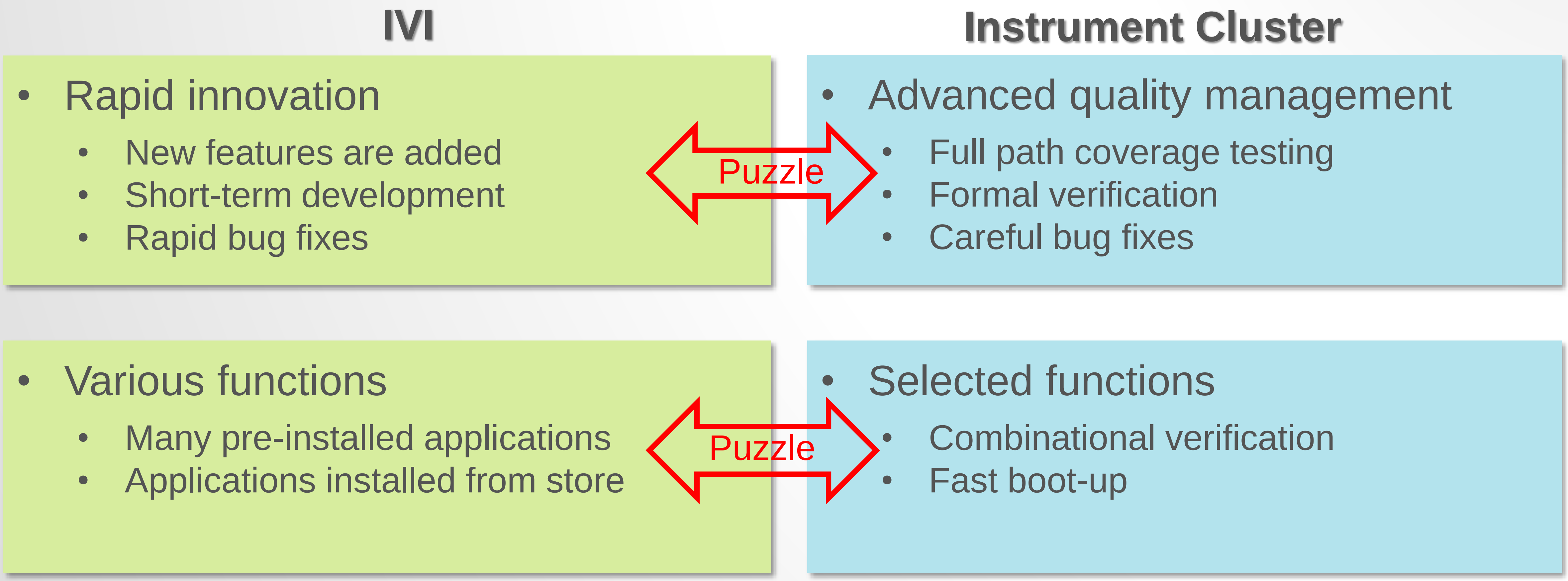
- Functional safety is required.
  - **Collaborate with ELISA Project**
- Quality management is required.

2. Lightweight

- Constraints on boot time are severe.
- Current AGL stack is heavyweight.

Puzzles in automotive quality management

- There are many puzzles in the automotive system (main function).



IVI

- Rapid innovation
  - New features are added
  - Short-term development
  - Rapid bug fixes

Puzzle

- Various functions
  - Many pre-installed applications
  - Applications installed from store

Puzzle

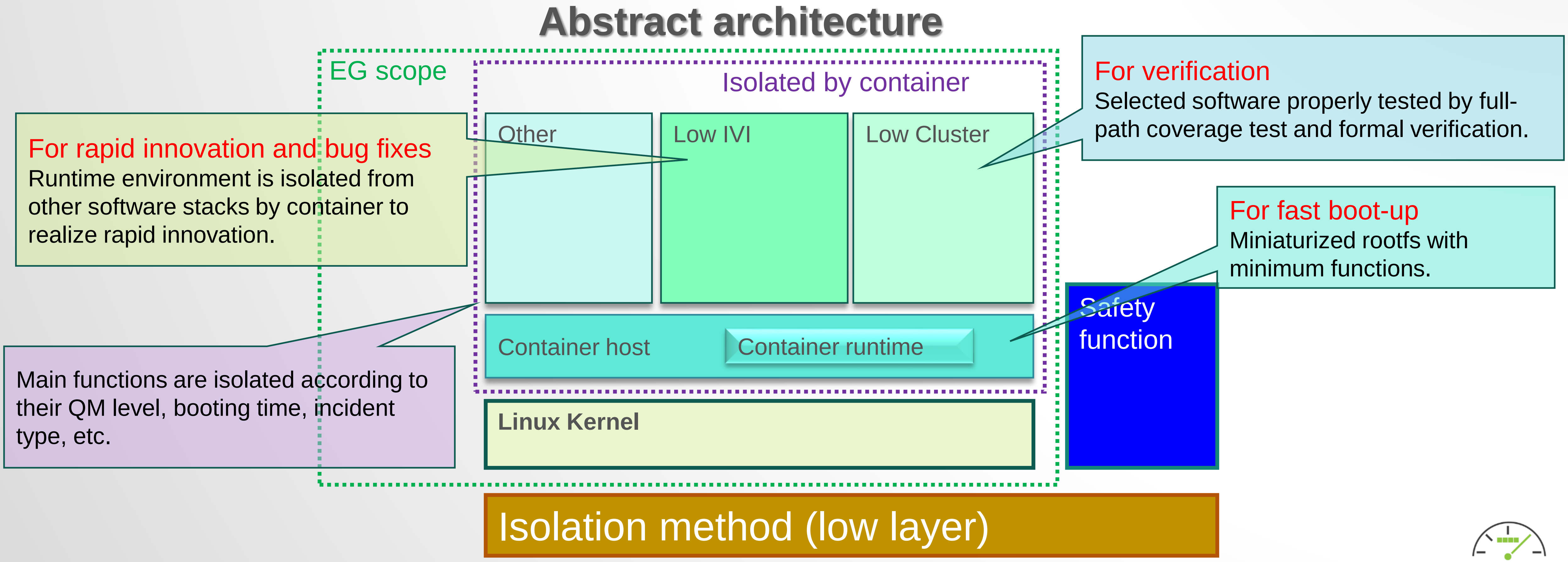
Instrument Cluster

- Advanced quality management
  - Full path coverage testing
  - Formal verification
  - Careful bug fixes

- Selected functions
  - Combinational verification
  - Fast boot-up

QM Isolation

- Our answer to the puzzle issues is “one more isolation method" which takes one-more layer to isolate the functions by using Linux container technology.



For rapid innovation and bug fixes
Runtime environment is isolated from other software stacks by container to realize rapid innovation.

Main functions are isolated according to their QM level, booting time, incident type, etc.

For verification
Selected software properly tested by full-path coverage test and formal verification.

For fast boot-up
Miniaturized rootfs with minimum functions.

What are the product development issues?

1. Quality and Robustness

- Functional safety is required.
  - **Collaborate with ELISA Project**
- Quality management is required.
  - **QM Isolation**

2. Lightweight

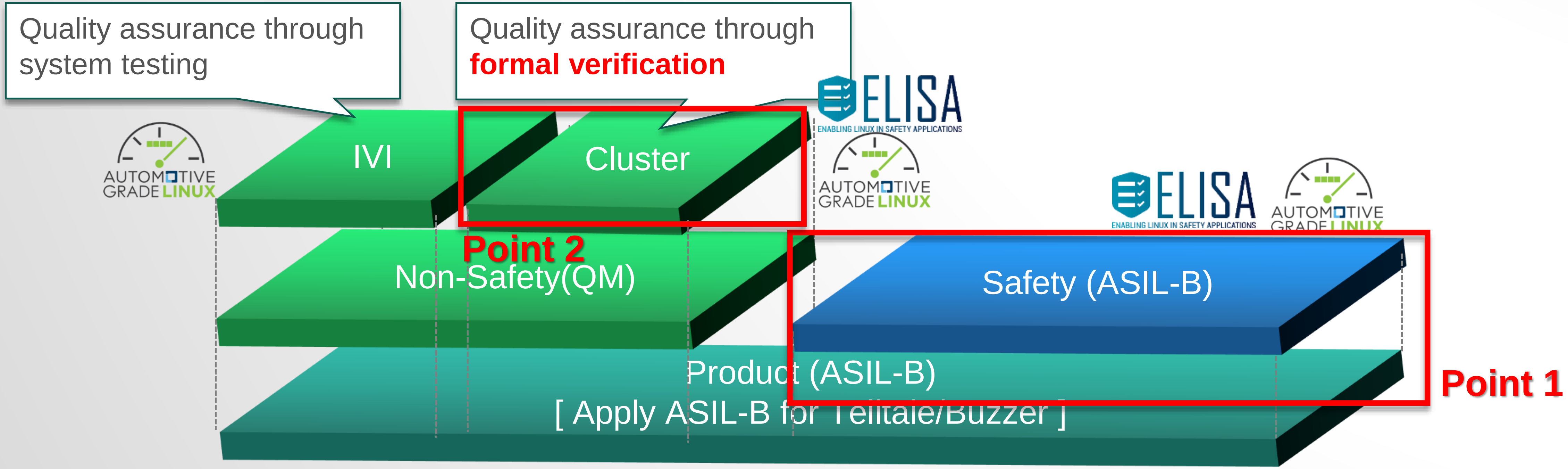
- Constraints on boot time are severe.
- Current AGL stack is heavyweight.

Outline

- AGL Instrument Cluster EG
- Concept of Instrument Cluster EG
- **Collaboration proposal from AGL**
  - Function safety
    - Example use case : telltale
  - QM Isolation
    - What solution we need ?
- Conclusion

Collaboration proposal from AGL

- AGL IC-EG want to collaborate on two points with ELISA.
  - Point 1
    - How to realize safety.
  - Point 2
    - How to create verifiable open source software stack.

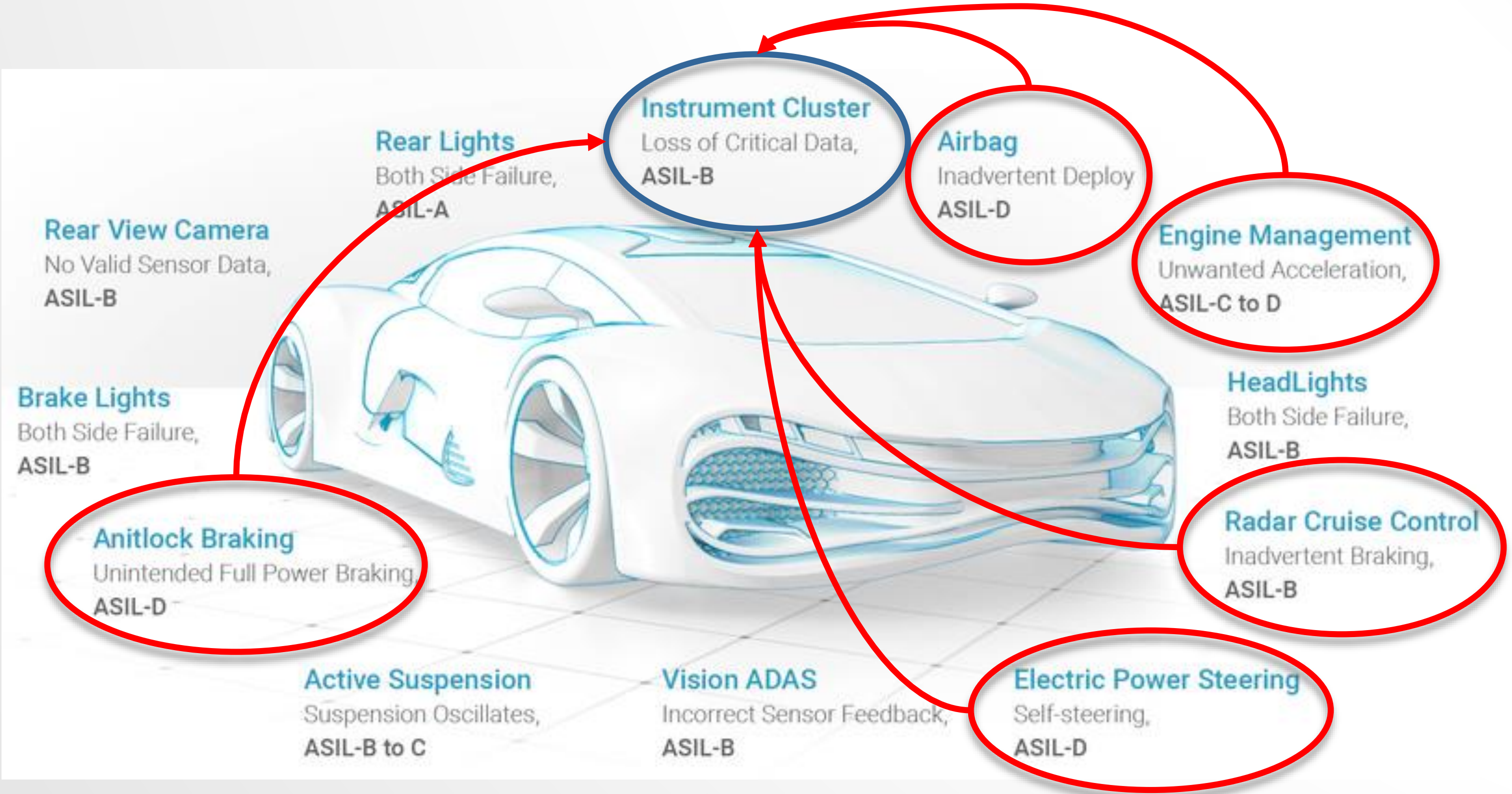

Quality assurance through system testing
Quality assurance through **formal verification**
IVI
Cluster
Point 2
Non-Safety(QM)
Safety (ASIL-B)
Product (ASIL-B)
[Apply ASIL-B for Telltale/Buzzer]
Point 1
ELISA ENABLING LINUX IN SAFETY APPLICATIONS
AUTOMOTIVE GRADE LINUX

Outline

- AGL Instrument Cluster EG
- Concept of Instrument Cluster EG
- **Collaboration proposal from AGL**
  - **Function safety**
    - **Example use case : telltale**
  - QM Isolation
    - What solution we need ?
- Conclusion

Why ASIL-B is required 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.

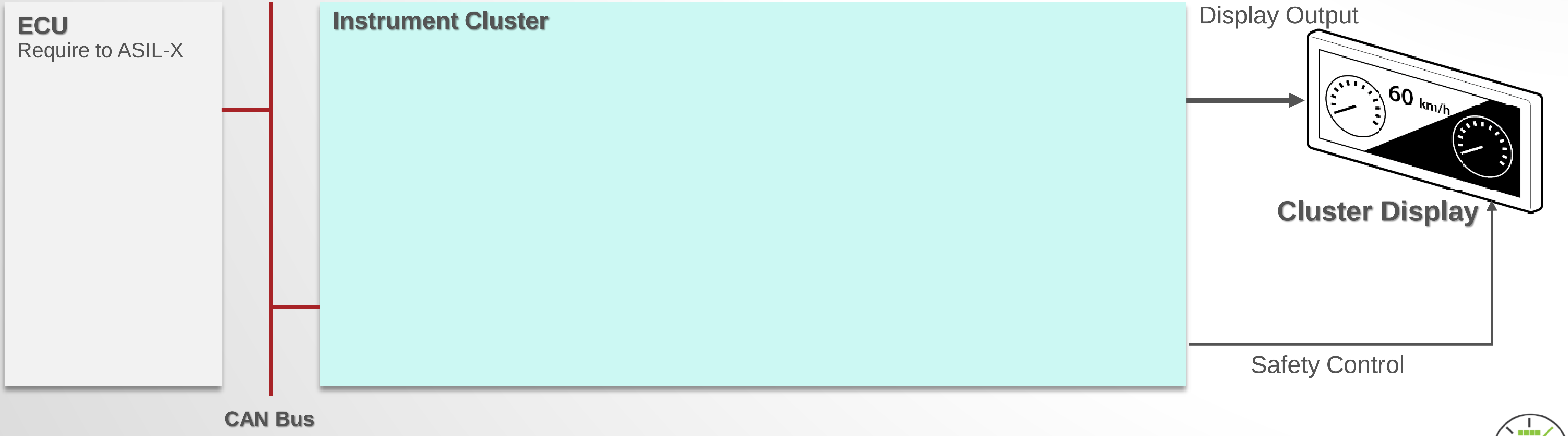


Ref. https://www.synopsys.com/automotive/what-is-asil.html

Case study : Telltale

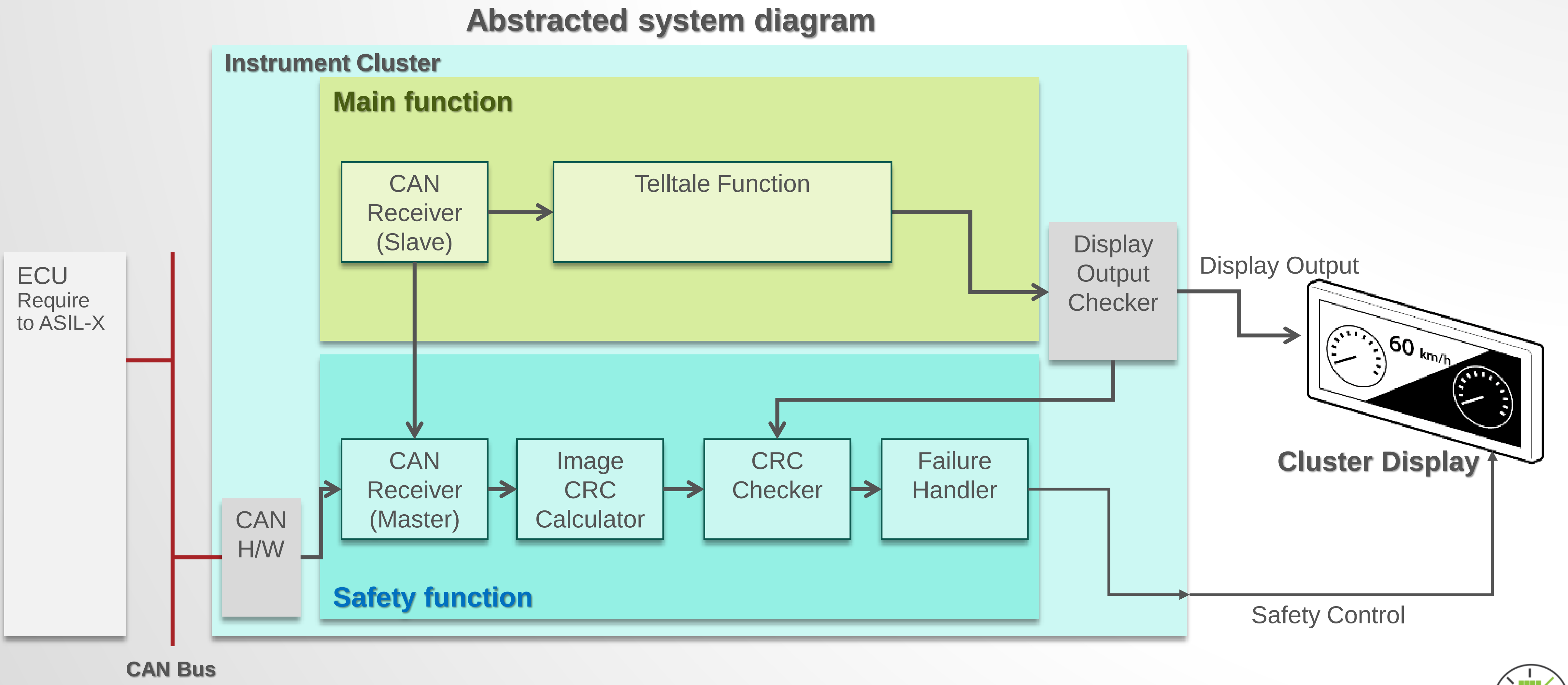
- Typical system block diagram
  - ASIL-X ECU and Instrument Cluster are connected by CAN.
  - Cluster outputs Safety Control signal separately from output to Display.
  - When safety control is enabled, Cluster display show the failure information.

Abstracted system diagram



Case study : Telltale

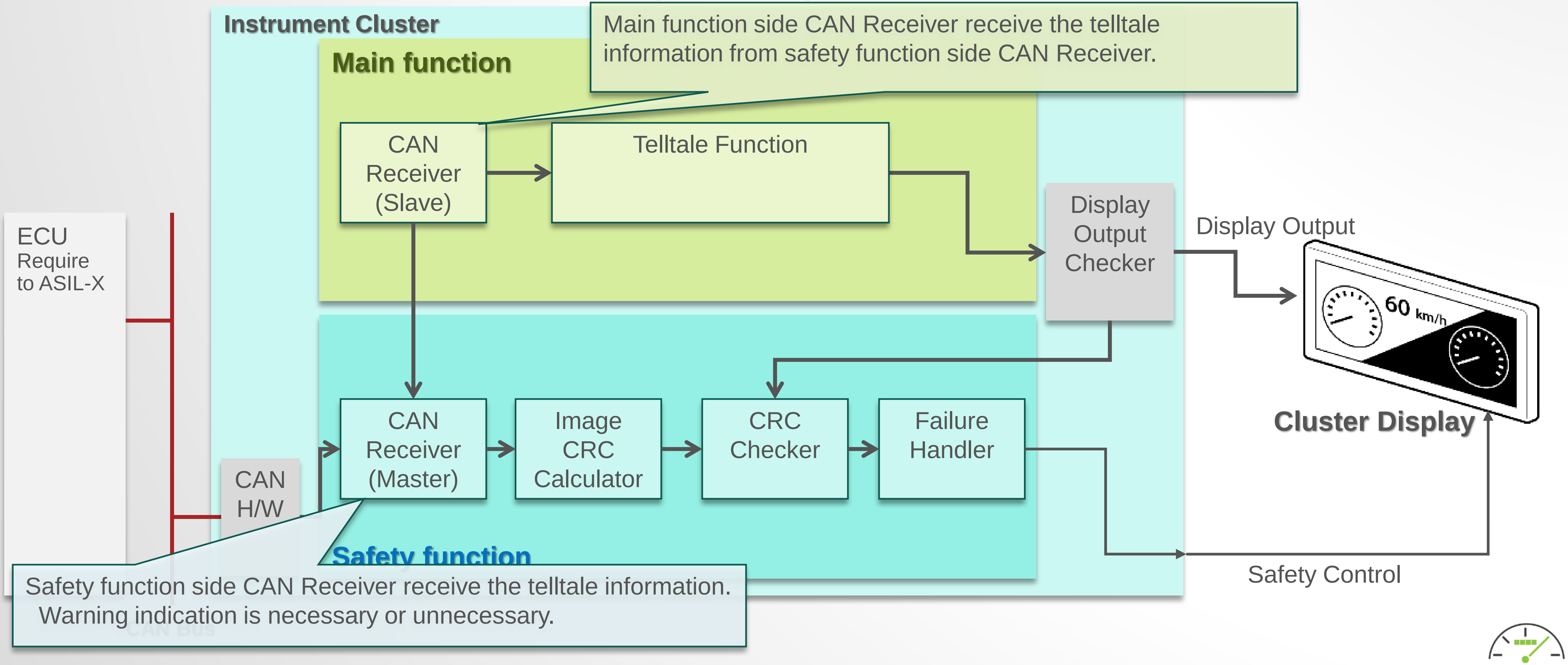
- More detail of system block diagram



Case study : Telltale

- More detail of system block diagram

Abstracted system diagram

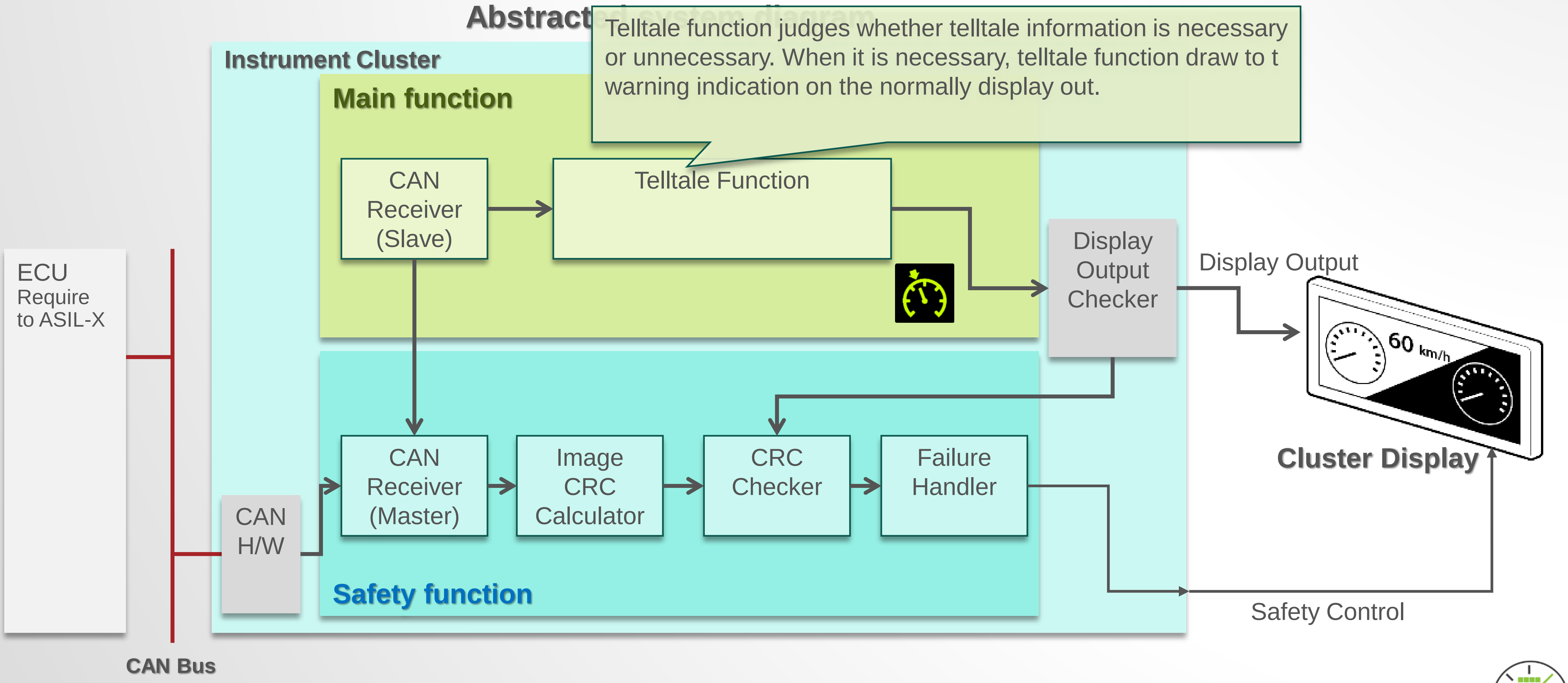


Main function side CAN Receiver receive the telltale information from safety function side CAN Receiver.

Safety function side CAN Receiver receive the telltale information.
Warning indication is necessary or unnecessary.

Case study : Telltale

- More detail of system block diagram

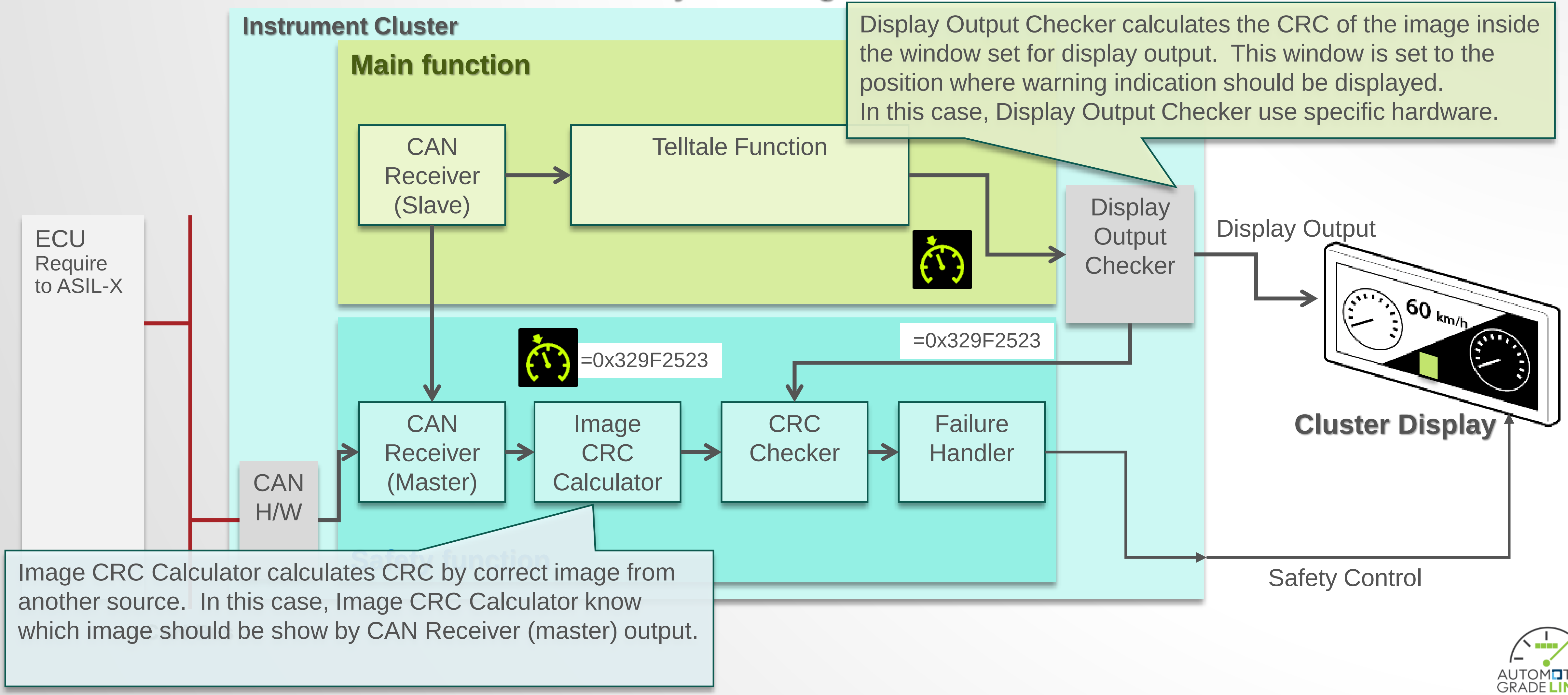


Telltale function judges whether telltale information is necessary or unnecessary. When it is necessary, telltale function draw to t warning indication on the normally display out.

Case study : Telltale

- More detail of system block diagram

Abstracted system diagram



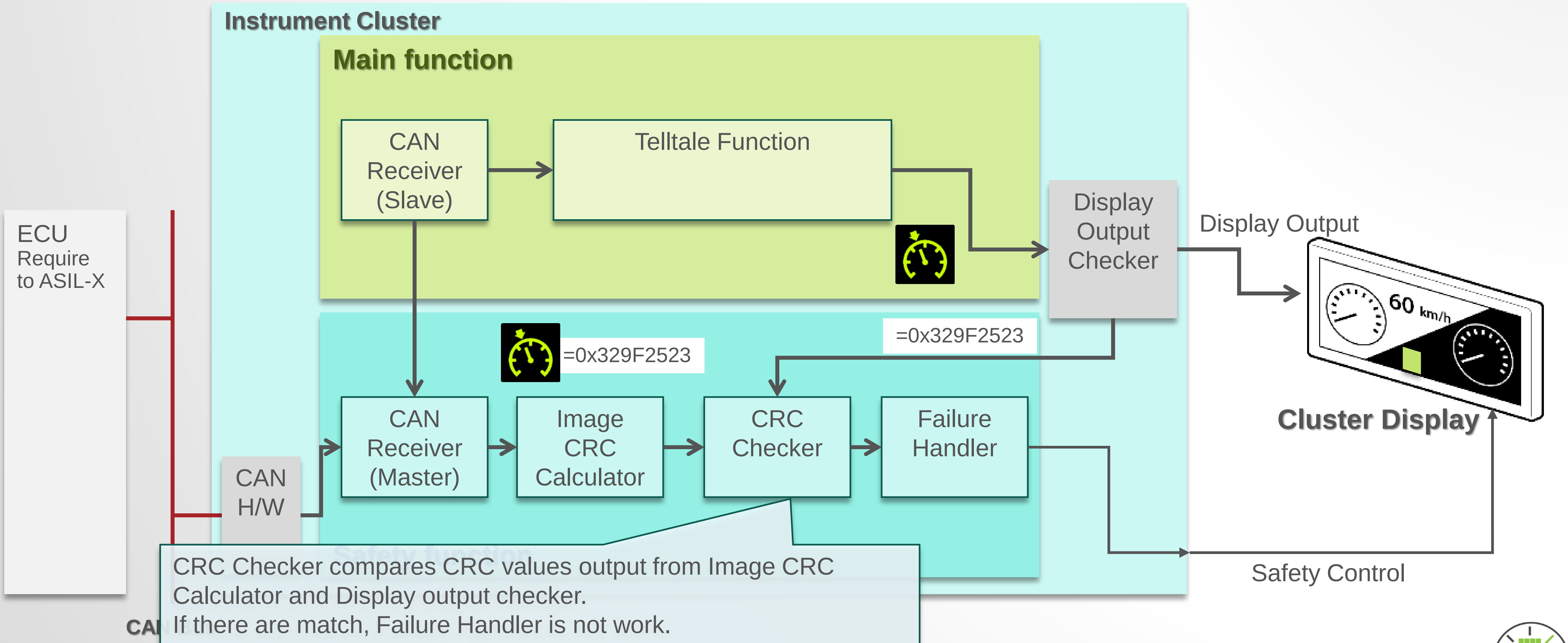
Display Output Checker calculates the CRC of the image inside the window set for display output. This window is set to the position where warning indication should be displayed.
In this case, Display Output Checker use specific hardware.

Image CRC Calculator calculates CRC by correct image from another source. In this case, Image CRC Calculator know which image should be show by CAN Receiver (master) output.

Case study : Telltale

- More detail of system block diagram

Abstracted system diagram

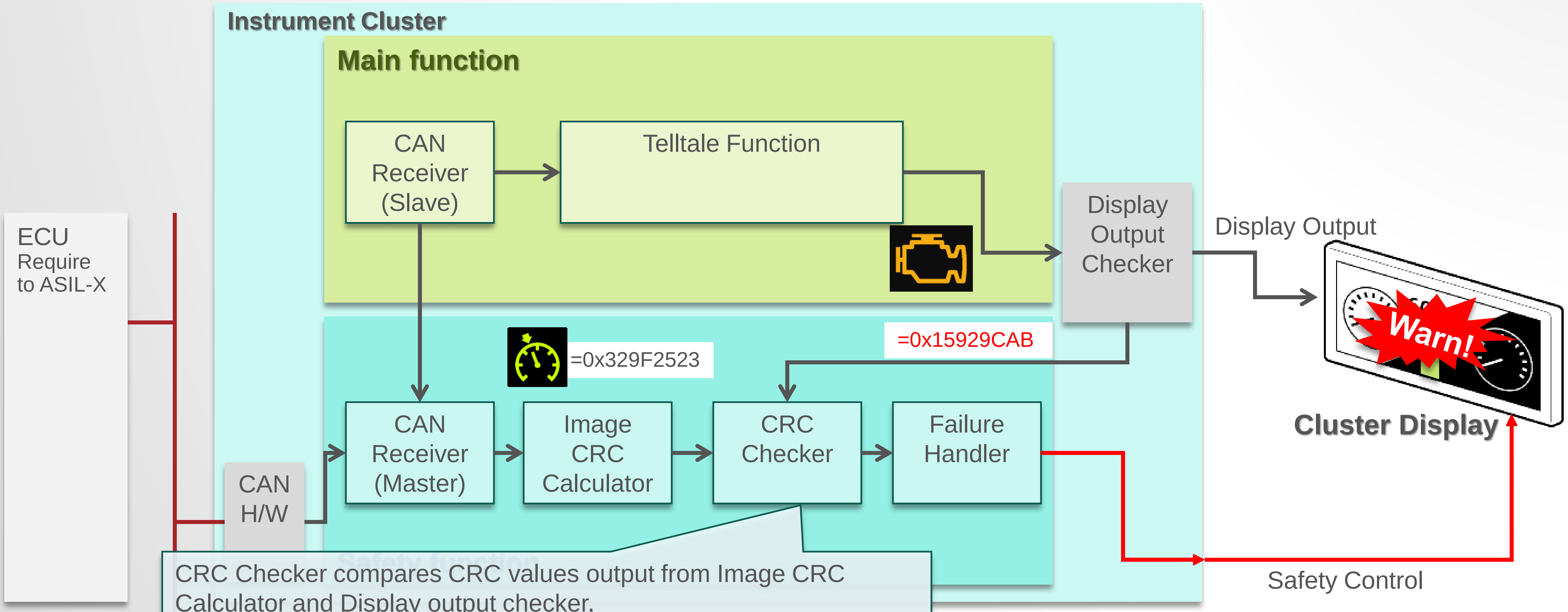


CRC Checker compares CRC values output from Image CRC Calculator and Display output checker.
If there are match, Failure Handler is not work.

Case study : Telltale

- More detail of system block diagram

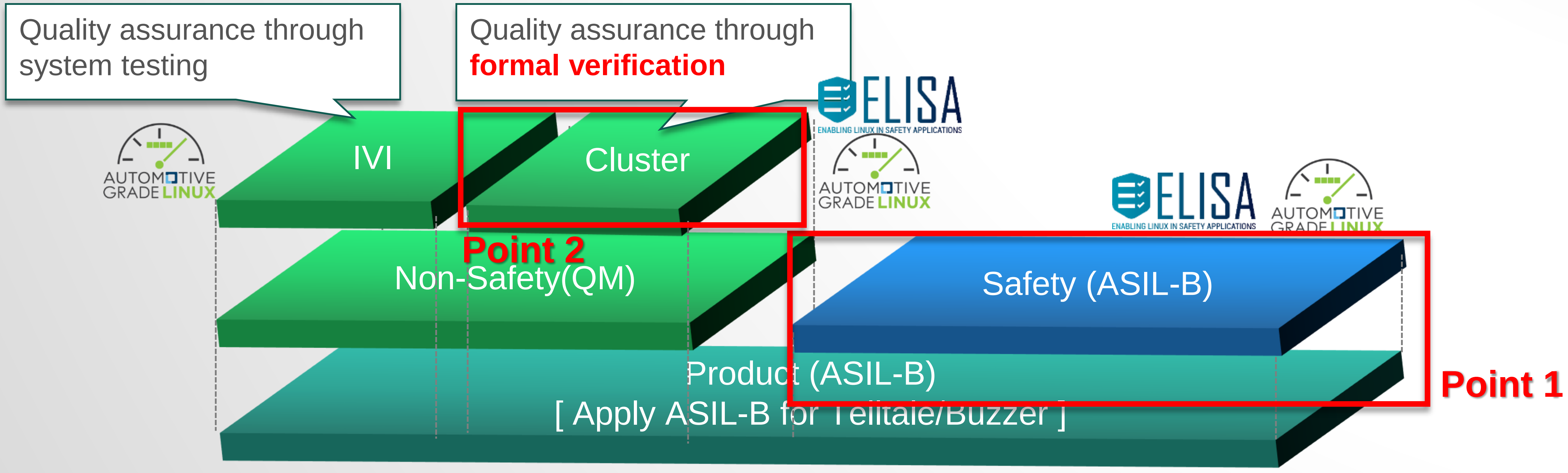
Abstracted system diagram



CRC Checker compares CRC values output from Image CRC Calculator and Display output checker.
If there are match, Failure Handler is not work.
If there are not match, Failure Handler enable Safety Control.

What we want in collaboration Point 1

- We want to create a base platform for Instrument Cluster in AGL community.
- We know some examples of functional safety requirements for Instrument Cluster. But we can't analyze and upstream that case to share knowledge.
- We want to find a generic solution with ELISA project to realize opensource base safety systems.



Quality assurance through system testing

Quality assurance through **formal verification**

IVI

Cluster

Point 2

Non-Safety(QM)

Safety (ASIL-B)

Point 1

Product (ASIL-B)
[Apply ASIL-B for Telltale/Buzzer]

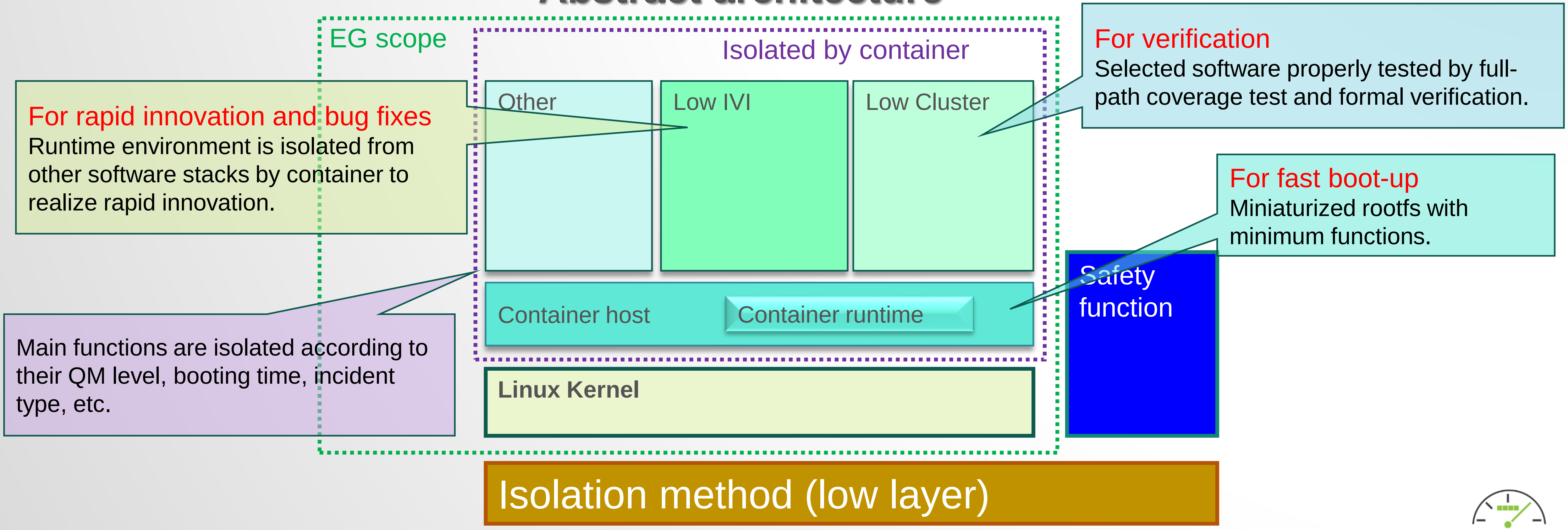
Outline

- AGL Instrument Cluster EG
- Concept of Instrument Cluster EG
- **Collaboration proposal from AGL**
  - Function safety
    - Example use case : telltale
  - **QM Isolation**
    - **What solution we need ?**
- Conclusion

Remind of QM Isolation

- Realize to provide the best software stack for each.
- See here for more details of architecture.
  - https://events19.linuxfoundation.org/events/agl-amm-eu-2019/program/schedule/
  - At “Joint presentation: Container Based Architecture for AGL”

Abstract architecture



For rapid innovation and bug fixes
Runtime environment is isolated from other software stacks by container to realize rapid innovation.

For verification
Selected software properly tested by full-path coverage test and formal verification.

For fast boot-up
Miniaturized rootfs with minimum functions.

Main functions are isolated according to their QM level, booting time, incident type, etc.

What is QM isolation?

- "One more isolation" is a method to take one-more layer to isolate the functions with Linux container technology.
- Why container?
    - Linux container technology
        - Isolate root filesystems on Linux kernel by using **chroot.**
            - Isolates software stack in accordance with their QM level.
        - Control resource (such as cpu, memory) by using **cgroups.**
            - Guarantees the resources to instrument cluster.
        - Hide resources from other containers by using **namespace.**
            - Protects cluster resources from other functions.

Issue of verification side

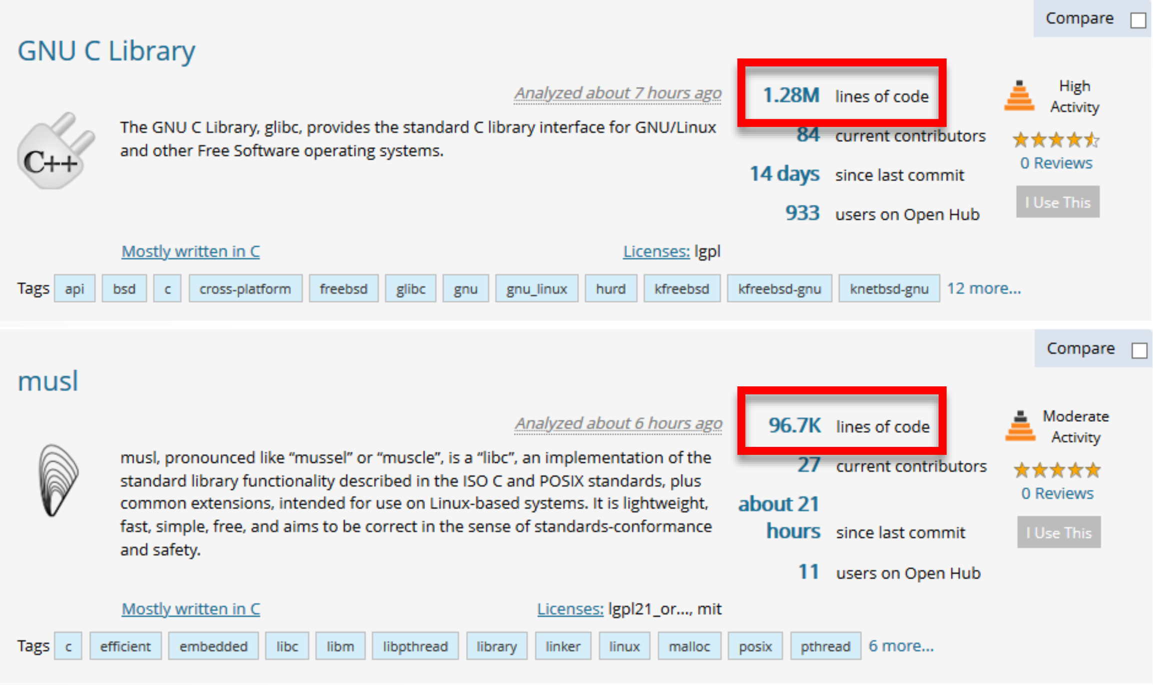
- QM isolation realize to isolate software stack according to their QM level, booting time, incident type, etc. It realize to minimize software stack that needs to be verified.
- But it still requires a lot of code verification.
  - This issue is same of certification case.

Example of issue

LSB Core Module Library	
libcrypt	libpthread
libdl	librt
libgcc_s	libssl3
libncurses	libstdcxx
libncursesw	libutil
libnspr4	libz
libnss3	libc
libpam	libm

16 library

Ref. http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Common/LSB-Common/requirements.html



GNU C Library

Analyzed about 7 hours ago

The GNU C Library, glibc, provides the standard C library interface for GNU/Linux and other Free Software operating systems.

1.28M lines of code
84 current contributors
14 days since last commit
933 users on Open Hub

Mostly written in C

Licenses: lgpl

Tags api bsd c cross-platform freebsd glibc gnu gnu_linux hurd kfreebsd kfreebsd-gnu knetbsd-gnu 12 more...

musl

Analyzed about 6 hours ago

musl, pronounced like “mussel” or “muscle”, is a “libc”, an implementation of the standard library functionality described in the ISO C and POSIX standards, plus common extensions, intended for use on Linux-based systems. It is lightweight, fast, simple, free, and aims to be correct in the sense of standards-conformance and safety.

96.7K lines of code
27 current contributors
about 21 hours since last commit
11 users on Open Hub

Mostly written in C

Licenses: lgpl21_or..., mit

Tags c efficient embedded libc libm libpthread library linker linux malloc posix pthread 6 more...

Ref. https://www.openhub.net/

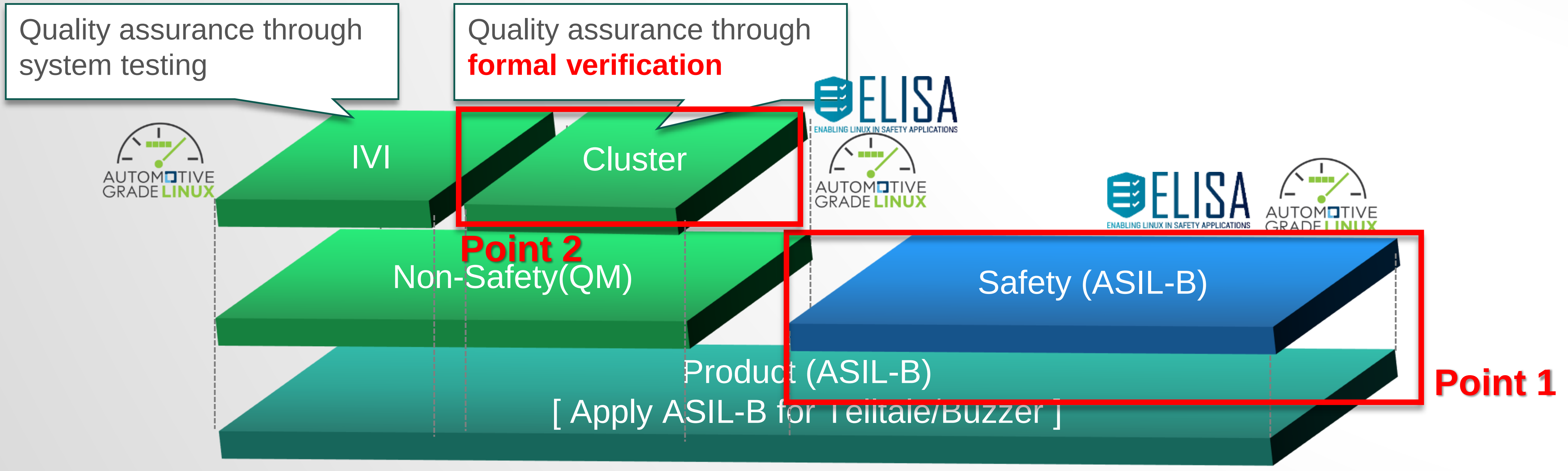
What solution we need

- ELISA community has several interesting efforts to realize functional safety.
  - One of them is an effort related to POSIX analysis and source code verification tools.

- Quality Management of main functions as a big issue.
  - But we haven’t found an answer yet.
  - We think that current ELISA method is applicable to the Quality Management side as well.

What we want in collaboration Point 2

- We want to seek and use verification methods with ELISA Project.
  - Example of activity
    - AGL community define the minimalized software stack.
    - Both community analyze software stack and develop the verification tool.
    - This activity will share in the both community.


Quality assurance through system testing
Quality assurance through **formal verification**
IVI
Cluster
Point 2
Non-Safety(QM)
Safety (ASIL-B)
Product (ASIL-B)
[Apply ASIL-B for Telltale/Buzzer]
Point 1
ELISA
ENABLING LINUX IN SAFETY APPLICATIONS
AUTOMOTIVE GRADE LINUX

Outline

- AGL Instrument Cluster EG
- Concept of Instrument Cluster EG
- Collaboration proposal from AGL
  - Function safety
    - Example use case : telltale
  - QM Isolation
    - What solution we need ?
- **Conclusion**

Conclusion

- Summary of our presentation
  - In this presentation, we shared the concept and issue of AGL Instrument Cluster EG.
  - In functional safety side, we shared the case study of telltale function and our issue.
  - In main function side, we shared our QM isolation concept detail and issue.
  - Overall, we proposed the content of the ELISA Project and AGL collaboration..
- Future agenda
  - We hope to start the discussion about the collaboration between AGL and ELISA based on this presentation.
  - For the current status, please visit the following link:
    - https://confluence.automotivelinux.org/display/IC/Instrument+Cluster+Home