



**AUTOMOTIVE
LINUX SUMMIT**



OPEN SOURCE SUMMIT
JAPAN

THE LINUX FOUNDATION

Lightweight and Easy to Integration Into Embedded System Using Linux Container

Harunobu Kurokawa / Renesas Electronics Corporation

2022/Dec/5

#ossummit @autolinuxjp



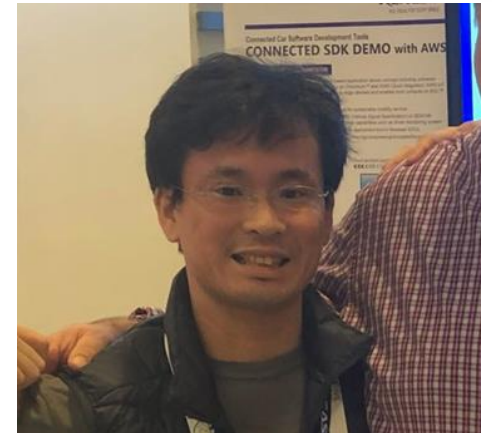
Today's topics

1. Introduction :Background AGL and Instrument Cluster
2. About Linux Container
3. Feasibility study using Container and AGL
4. Issue and Result
5. Conclusion

Introduction to Who I Am

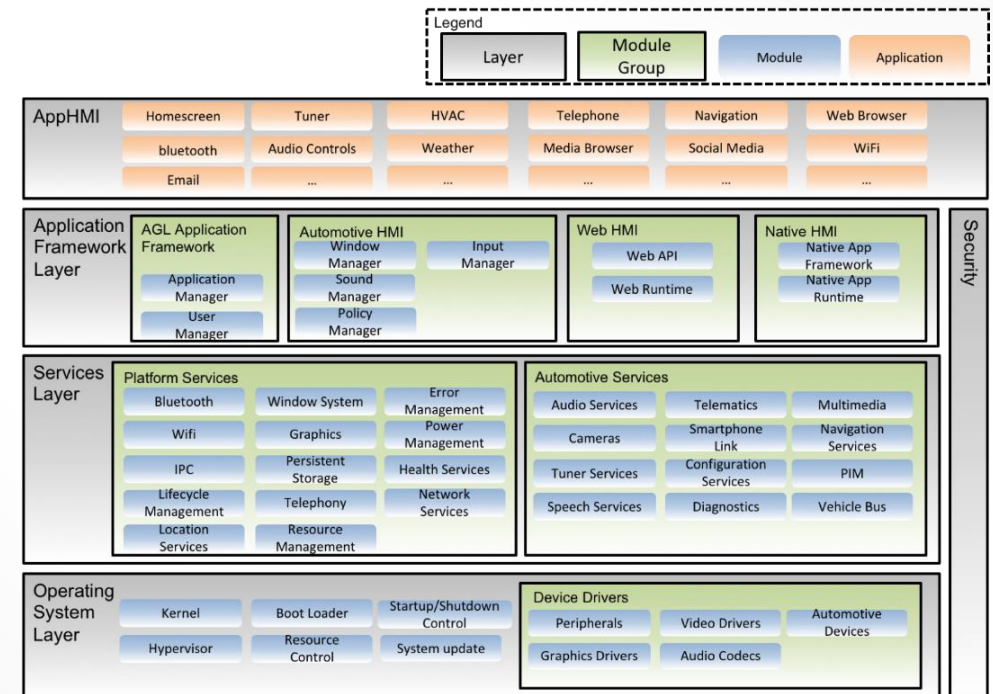
Name: Harunobu Kurokawa

- I'm working in Renesas Electronics Corporation. for 10+ years.
 - Embedded Software Engineer 2007,
 - Linux and OSS for automotive development (2013 ~)
- My experience of Open Source Community
 - Start AGL development (2016/Dec ~)
 - AGL BSP developer, reviewer and SAT member
 - Joined Instrument Cluster Expert Gr. (2019 ~)



Background of AGL / Instrument-Cluster Expert Gr.

- AGL's development policy
 - Collaborate and aim to rapid development for Automotive
 - Code first, Upstream first, and **“Unified Code Base”**
 - Product ready-ness development
- All components are Open Source
 - App/HMI layer
 - Framework layer
 - Service layer
 - OS layer: Kernel, device driver



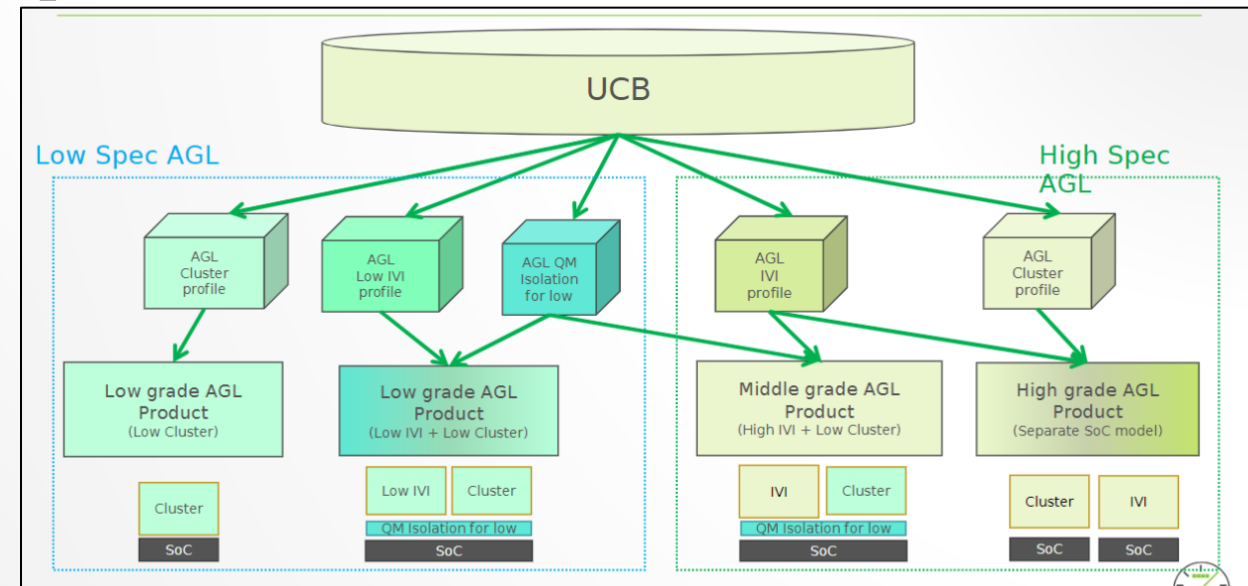
AGL Spec.

<https://confluence.automotivelinux.org/download/attachments/5472286/Diagram.png?version=2&modificationDate=1556702778000&api=v2>



Background of Instrument-Cluster Expert Gr.

- Motivation
 - Current SW components are based on infotainment (IVI)
 - We would like to create a new Profile next to IVI.
- Develop Cluster platform based on Linux.
 - Discussion for Cluster SW architecture.
 - Focus for both High end and “Low Spec” AGL
 - Create various profiles from UCB
- IC-EG was launched in 2019

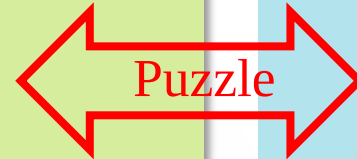


Instrument Cluster profile

There are many puzzles in the automotive system between IVI and Cluster.

IVI

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



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



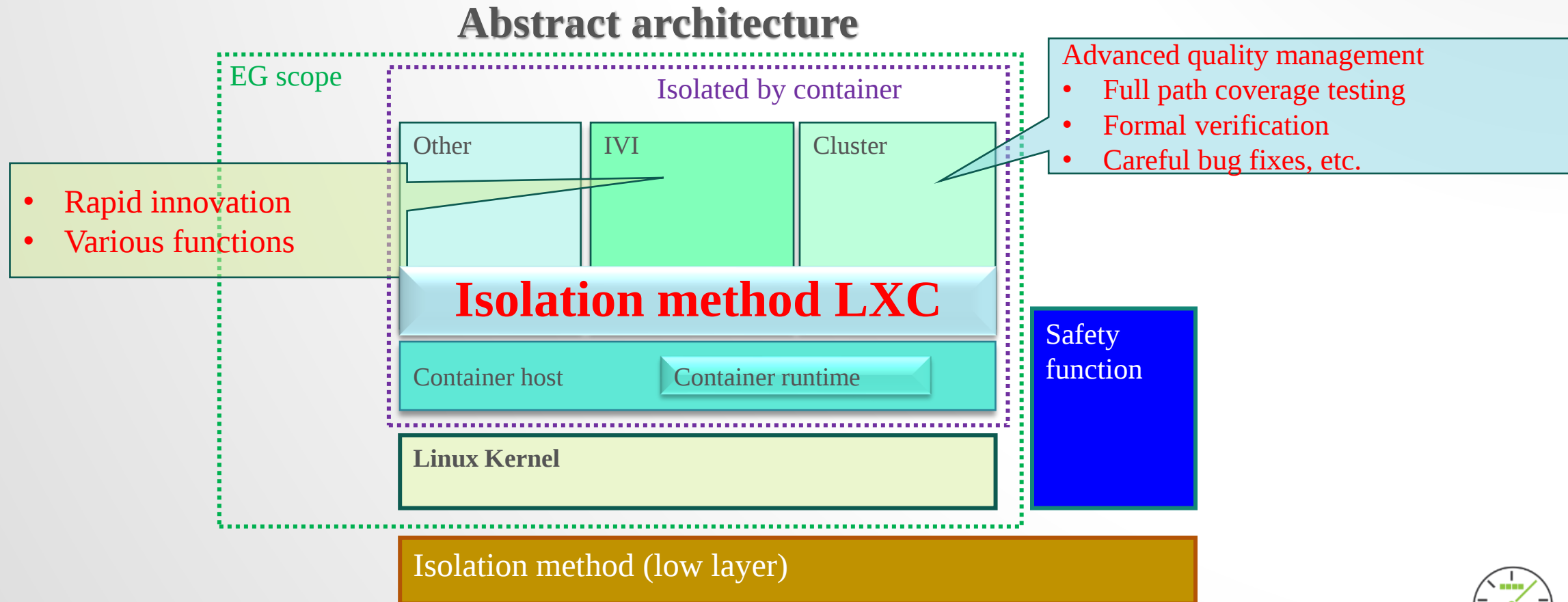
Instrument Cluster

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

- Selected functions
 - Combinational verification
 - Fast boot-up

Our approach

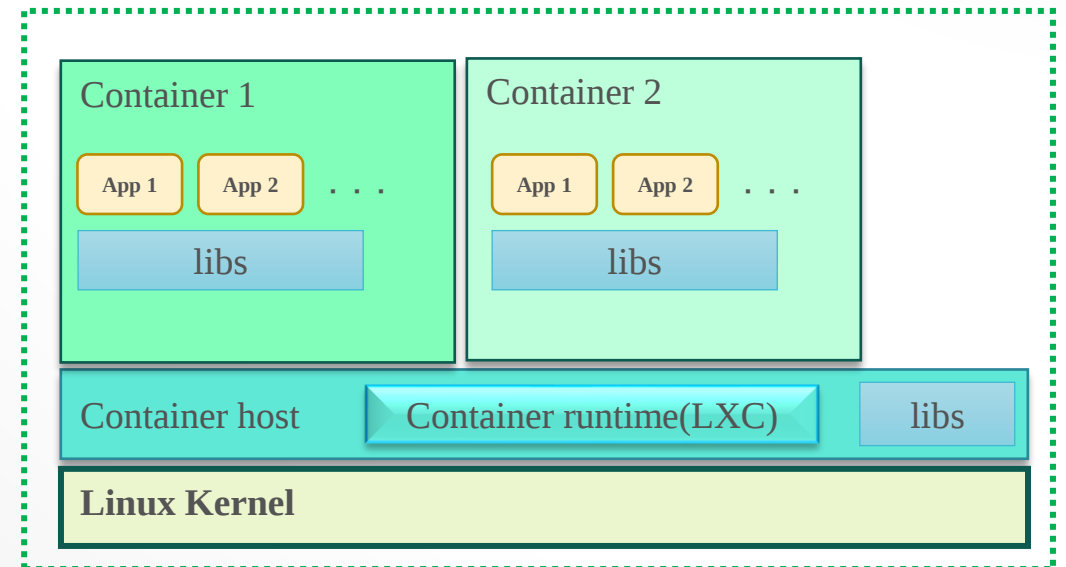
- In order to resolve the puzzle issues,
 - isolation method which to separate the functions by Linux container technology (LXC)



-
1. Introduction :Background AGL and Instrument Cluster
 2. **About Linux Container**
 3. Feasibility study using Container and AGL
 4. Issue and Result
 5. Conclusion

About LinuX Container (LXC)

- System-level virtualization software
 - Container includes shared libraries
 - Multiple application running in container
- User-space interface for the Linux kernel containment
 - Multiple containers are running on single Linux kernel
 - copy of an operating system running on the same kernel
 - Container don't have own kernel.
- Feature
 - Isolate root filesystems (chroot)
 - Control resource (cgroups)
 - Hide resources (namespace)



About LinuX Container (LXC)

- Easy to use for embedded environment.
 - Only require to Linux BSP (kernel)
 - No need additional virtualization driver development
 - Just enable the necessary kernel config.
 - Yocto Recipe is provided in meta-virtualization
 - Add the following description in host side only

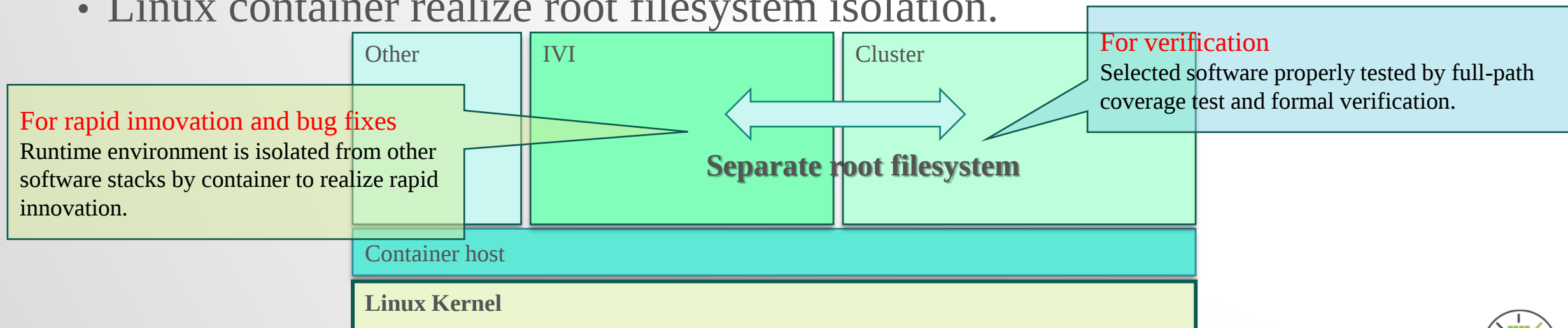
```
<conf/local.conf>  
DISTRO_FEATURES_append = " virtualization"  
IMAGE_INSTALL += " lxc"
```

```
<conf/bblayer.conf>  
${TOPDIR}/../meta-virtualization ¥
```

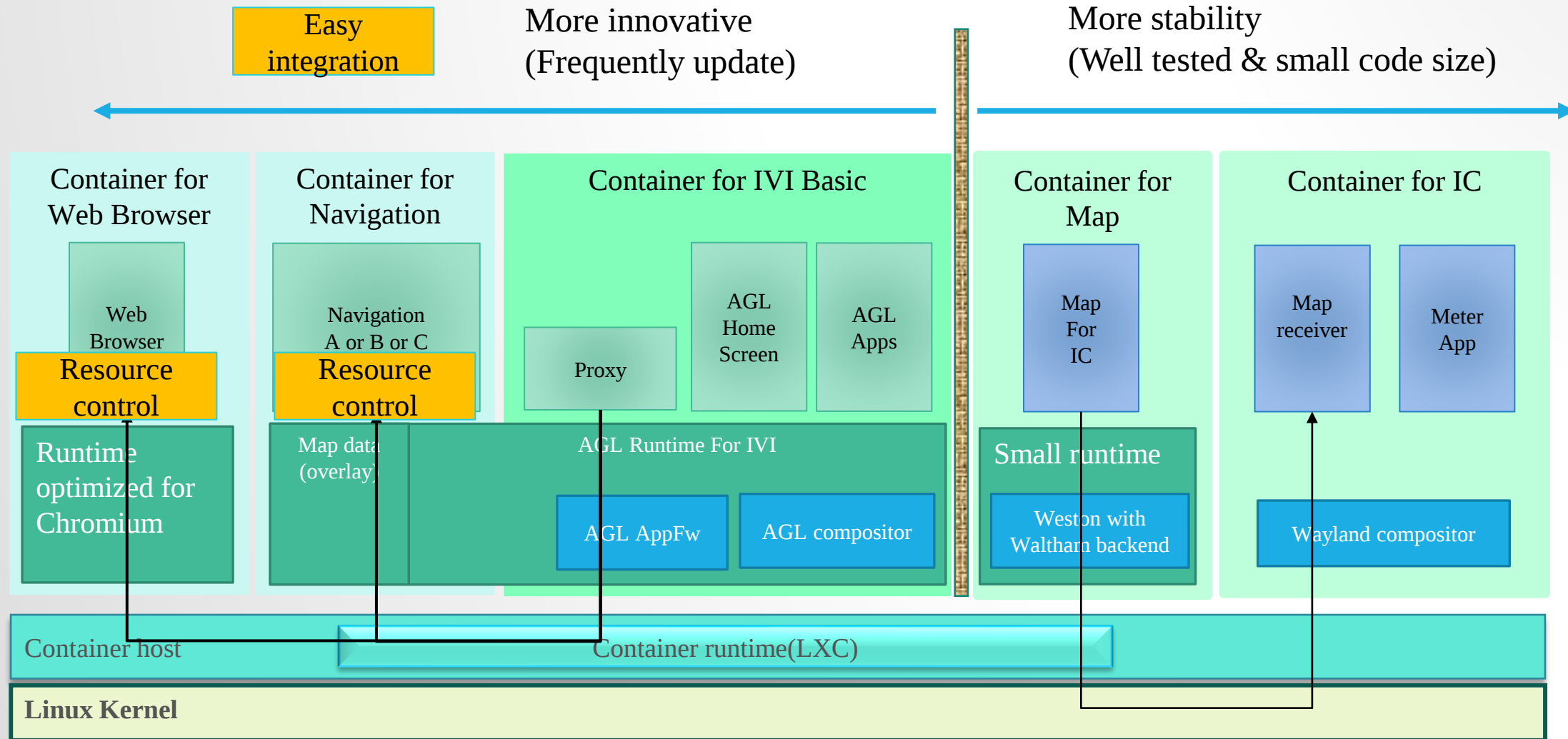
- License
 - LGPL-2.1+

Isolation root filesystem for Cluster

- Separate root filesystem between Instrument Cluster and IVI.
 - **IVI:**
 - Build by standard software.
 - Change after product release to upgrade function.
 - **Instrument Cluster:**
 - Build by highly quality assurance software.
 - NOT change after SOP without critical bug fix.
- Linux container realize root filesystem isolation.



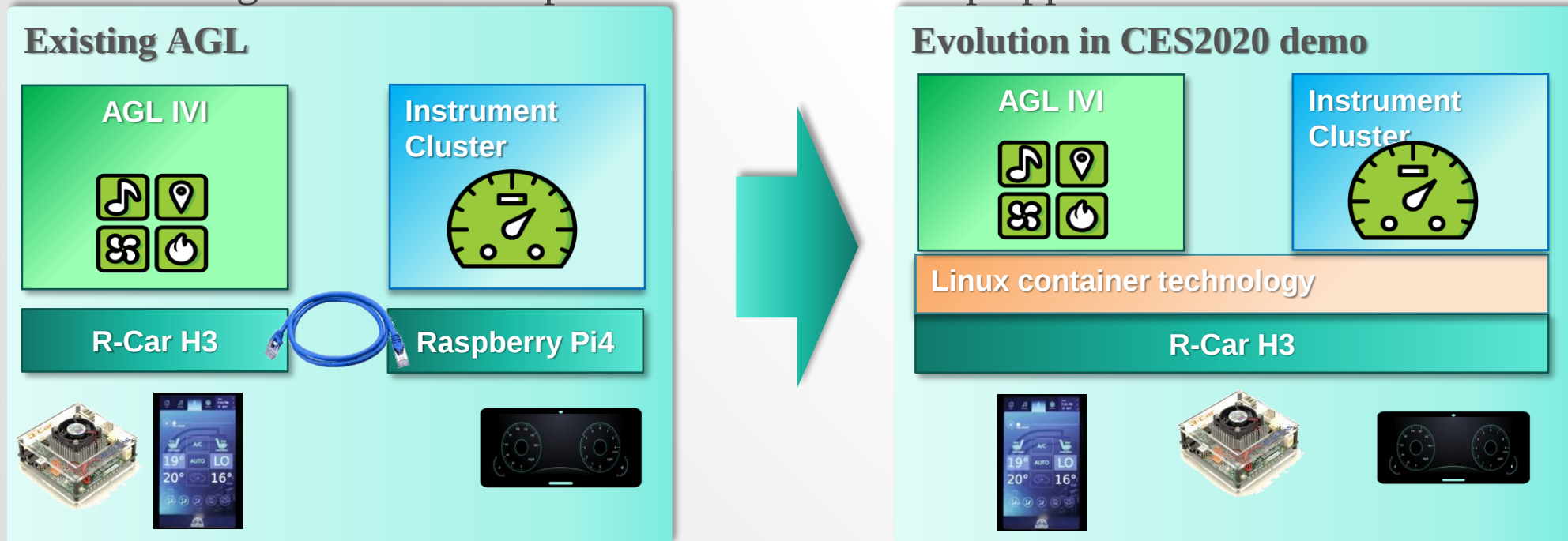
Example: Multiple Container use-case for Automotive



-
1. Introduction :Background AGL and Instrument Cluster
 2. About Linux Container
 3. **Feasibility study using Container and AGL**
 4. Issue and Result
 5. Conclusion

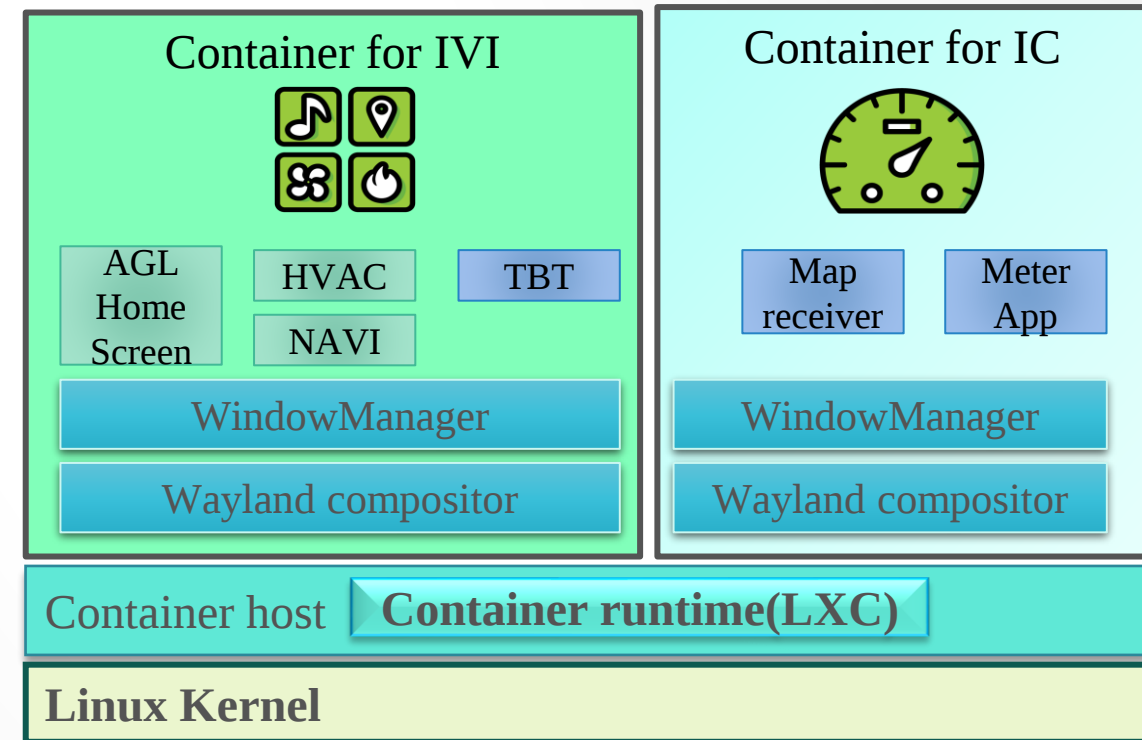
Feasibility study

- We start investigation using existing AGL profile (based CES2020 demonstration)
 - Existing AGL used two systems and two different SoCs
 - R-Car H3 (IVI) and Raspberry Pi 4(Cluster)
- Unify the two system and run on one SoC (R-Car H3)
 - IVI : agl-demo-platform with TBT application
 - Cluster : agl-cluster-demo-platform with TBT/map application



Feasibility study : Good point

- Easy to create rootfs
 - Install LXC into host filesystem
 - Includes related settings.
 - Can create minimal host filesystem
 - Mount guest rootfs using LXC config
eg) `lxc.rootfs.path = dir:/lxc/guest`
- Don't need to modify apps or settings in both container guests
 - Each Guests are reused existing AGL profile
 - IVI : agl-demo-platform
 - Cluster : agl-cluster-demo-platform
 - AGL has specific parameter in Apps.
 - App ID, AppFW, layer manager, any parameter
 - as well as yocto etc.



Feasibility study : Good point

- Reboot is very fast
- We compare system-reboot and Container reboot
 - System reboot : over 30 seconds
 - Container reboot : 5 seconds
 - No any optimization
 - Due to no need to restart kernel.

System reboot



IVI container reboot



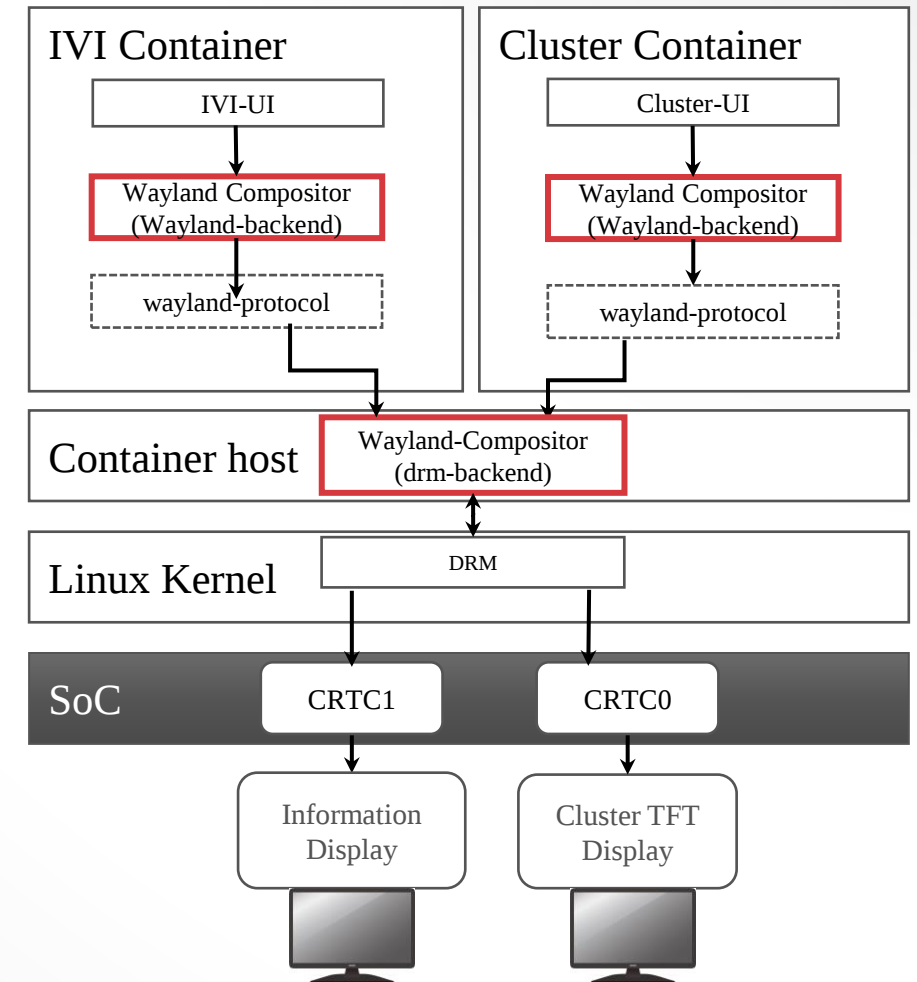
-
1. Introduction :Background AGL and Instrument Cluster
 2. About Linux Container
 3. Feasibility study using Container and AGL
 4. **Issue and Result**
 5. Conclusion

Feasibility study : Issue1

- Issue 1 : DRM device access
- Container use-case (Isolate display)
 - Display(s) connection for each containers
 - Compositors inside a both containers
- The drm device is controlled by only one process
 - “drm master” is usually assumed by single compositor.
 - Secondly compositor cannot access drm device.
- Hack: Use wayland-backend as nested- compositor

But,

- Host need wayland-compositor and libraries.
 - Cluster rendering is affected by Host-side compositor or IVI rendering
 - Cluster requirement: keep rendering fps, etc.
- Cluster compositor should wait until host-side weston started.



Feasibility study : Issue1

- Issue 1 : DRM device access



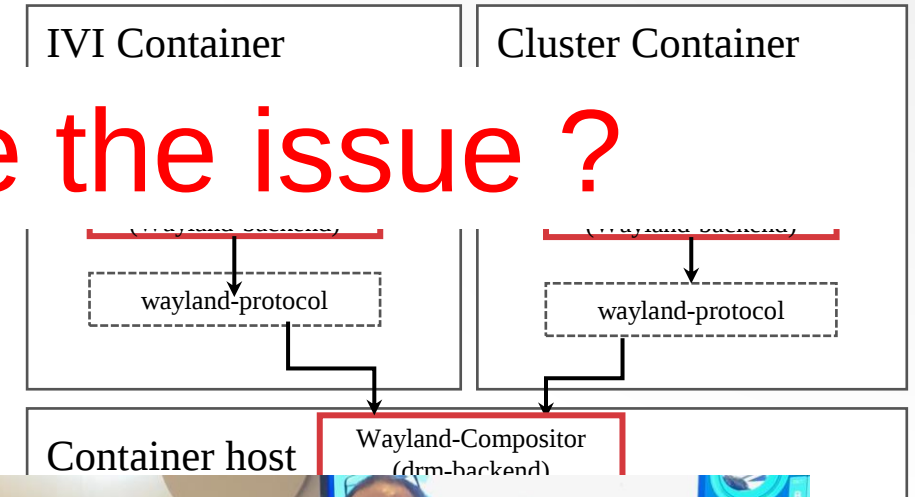
- Cluster rendering (Isolate display)

How to resolve the issue ?

- Cluster rendering is controlled by only one process
 - "drm master" is usually assumed by single compositor.
 - Secondly compositor cannot access drm device.
- Hack: Use wayland-backend as nested- compositor

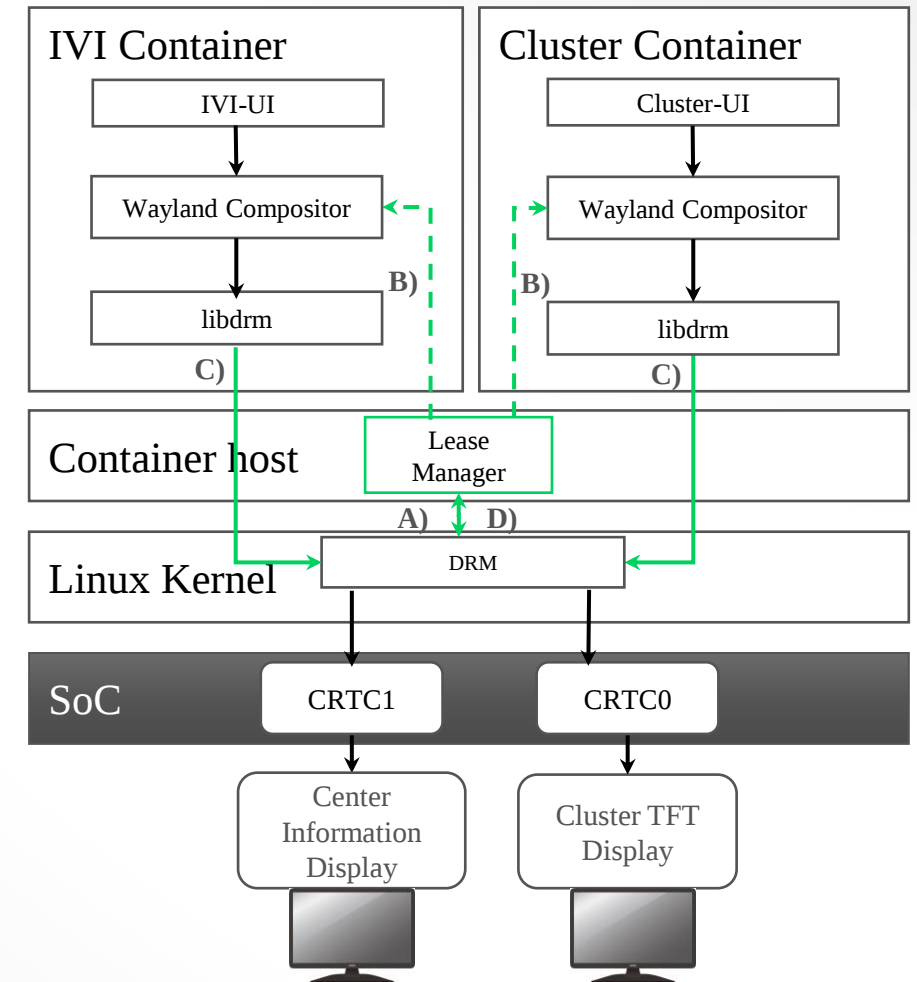
But

DRM Lease!!



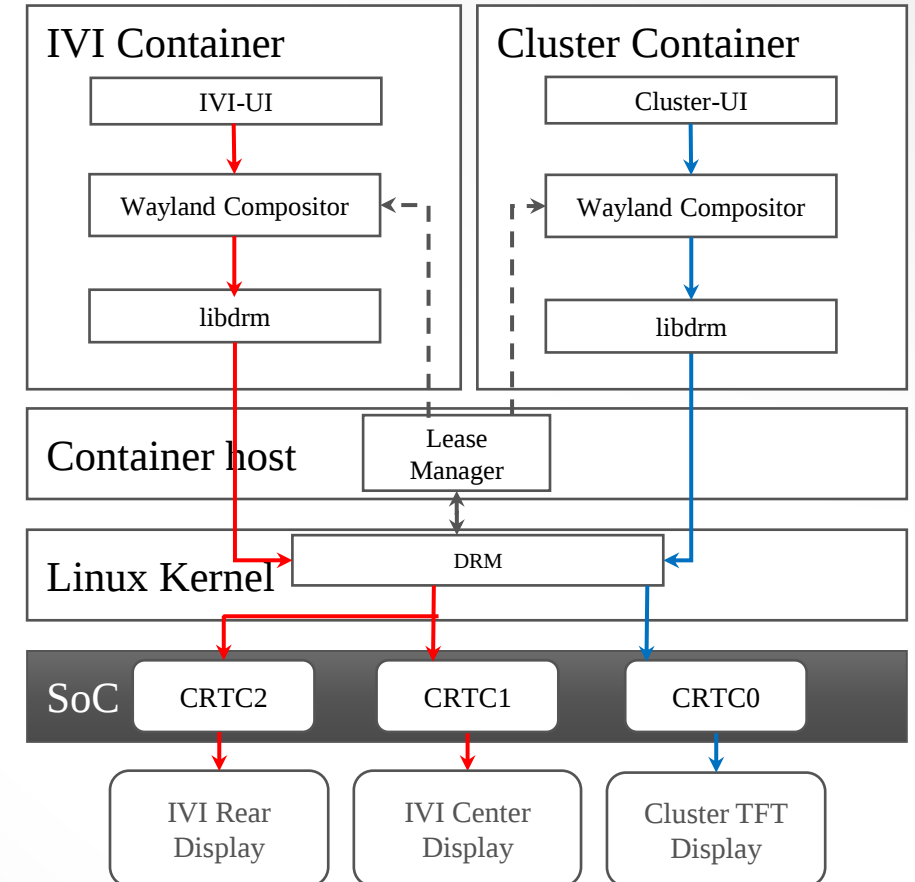
Drm Lease Manager

- Create **drm lease manager** to realize multiple compositor using **drm lease** kernel feature.
- Basic design
 - The drm master (fd) which can create and lease out resources to other DRM Masters. The drm lease manager takes on this role.
 - Each compositor controls resources leased from drm lease manager.
- Control flow
 - DRM Lease Manager(daemon) opens DRM device and creates lessee DRM Masters for each client.
 - Sends lessee file descriptor to clients (via lease client library) when requested
 - Clients render directly to DRM device using lessee
 - DRM Lease Manager revokes lessee when the client is finished with it.



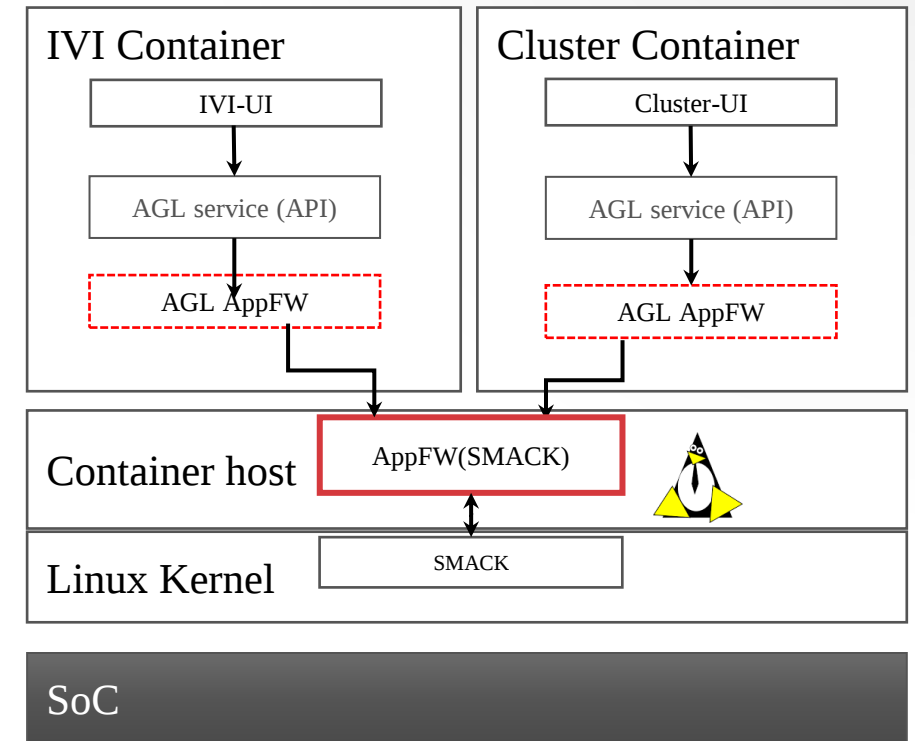
Current status

- DRM-lease manager is supported from Lamprey.
 - Two or more drm clients (e.g. weston) can launch at same time
 - Dynamic transition feature the ownership of a leases resources can be transferred from one client to another.
 - Support two or more DRM resources (connectors) to one client
 - The configuration file allows the user to specify the mapping of DRM connectors to DRM leases.
 - Easy to set the customized lease name instead of default drm name (e.g. “ivi” lease instead of “card0-HDMI-A-1”)
 - Usage: Add “agl-drm-lease” feature when user aglsetup.sh
 - See also:
https://docs.automotivelinux.org/en/marlin/#5_Component_Documentation/4_drm-leasemanager/
- These feature is available after Needlefish release.



Feasibility study : Issue2

- Issue 2 : AppFW (application framework) dependency
 - AGL API is provided service library
 - AGL service use AppFW
 - AppFW works closely with LSM (SMACK) to protect from uncertain Apps
 - Should install SMACK into Host
 - When I use agl-demo-platfrom binary as guest, the smack database is needed at least.
 - Hack : Host should have set SMACK labels in advance both IVI and Cluster
 - We should mount smack label to guest.
 - Both of guest smack's database should be stored
- ```
lxc.mount.entry = /etc/smack etc/smack none bind,optional,create=dir
```
- But,
  - Host should have correct smack label for all guest.
  - Difficult to upgrade Container



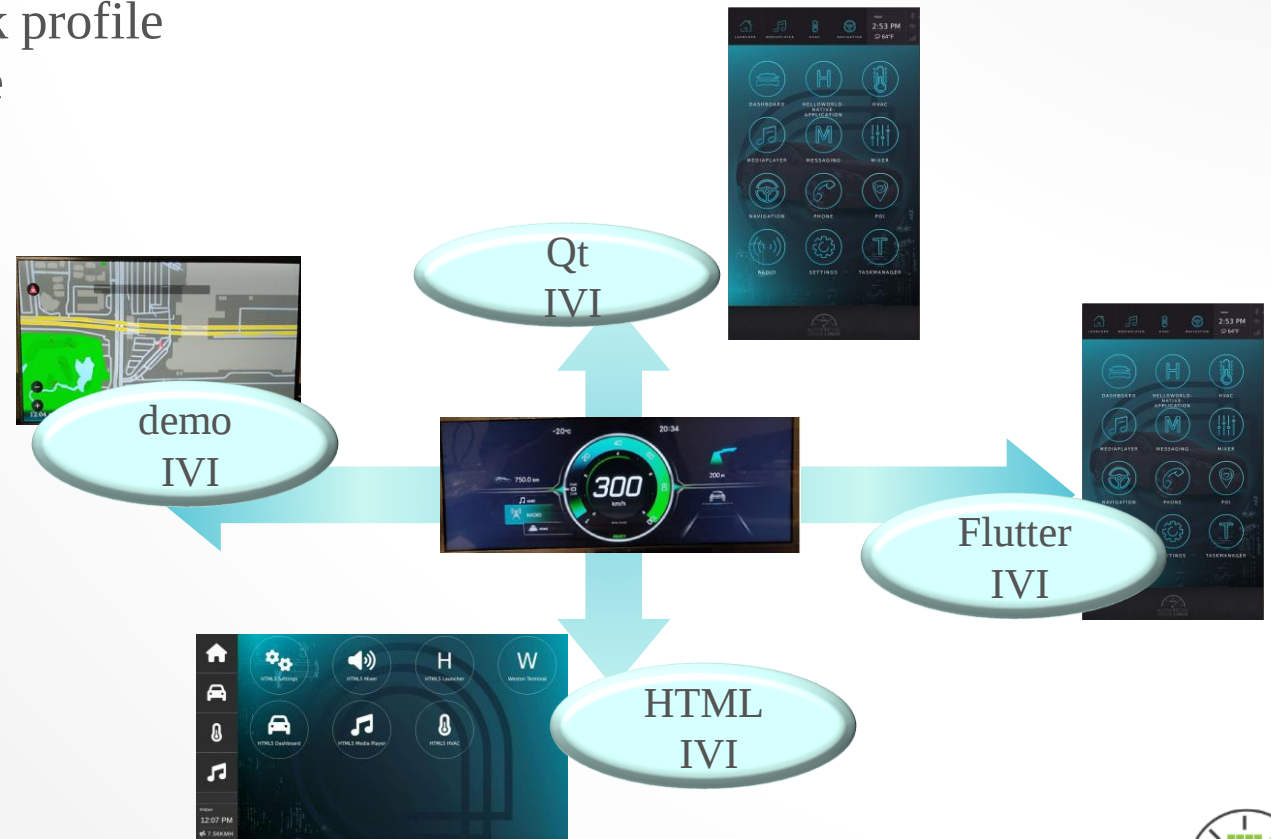
# Confirm issue2 in latest AGL

---

- AGL changed AppFW and LSM
  - Selinux is used instead of SMACK in 2022
  - We should confirm SeLinux using Latest AGL
- Confirm using AGL 14.0.1 ( agl-demo-platform and agl-ic-container-demo-host)
  - bitbake agl-cluster-demo-lxc-host  
add “agl-selinux” feature
  - bitbake agl-demo-platform  
add “agl-drm-lease” feature  
add "tar.bz2" in local.conf  
AGL\_DEFAULT\_IMAGE\_FSTYPES:append = " tar.bz2"
- Add only selinux parameter in /var/lib/lxc/ivi-demo/config
  - lxc.mount.entry = /sys/fs/selinux sys/fs/selinux none bind 0 0

# Result

- Current AGL supports simple demo-ivi only.
- Integrate 4 of AGL profiles for IVI.
  - demo-ivi: simple demo app for cluster demo
  - qt-ivi: Legacy AGL demo platform
  - Html5-ivi : Chromium based HMI framework profile
  - flutter-ivi : newly ivi HML framework profile
- Integrate all AGL profile with LXC.



# The status of AGL Cluster profile ?

- In initial release at Lamprey
  - LXC and IC app run on R-Car H3/M3 Starter Kit with single Display only.
- In Marlin release
  - IC software with demo IVI run on R-Car H3/M3 Starter Kit on Kingfisher board with two Display.
  - It already merged.
- AGL Ref-HW support merged in this year
- We will contribute the latest demo code.



R-CarH3



Demo Display



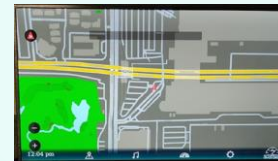
Infotainment Display



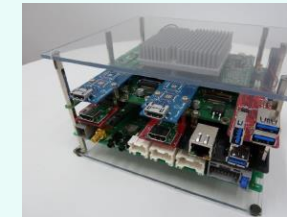
R-CarH3+Kingfisher



Cluster Display



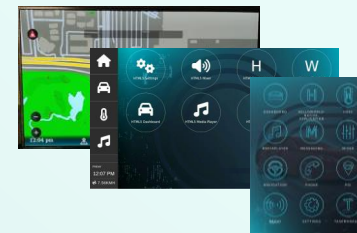
Infotainment Display



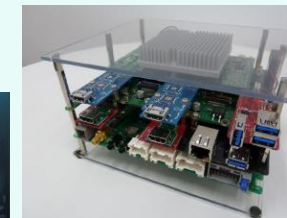
AGL Ref-HW



Cluster Display



Infotainment Display



AGL Ref-HW



Cluster Display

# How to create AGL LXC feature ?

- AGL document described build instruction.
  - Qemu, RPi4, R-CarH3/M3, Ref-HW, etc.
- There are many profile in AGL, then build procedure also busy.
  - IVI-demo, HTML, Cluster, etc.
- IC EG member create the example of cluster-demo integration feature not only build method.
  - E.g.) What's equipment required.
  - E.g.) How to connect some

[https://docs.automotivelinux.org/en/needlefish/#0Getting Started/3 Build and Boot guide Profile/1 Instrument Cluster %28IC-IVI with Container isolation%29/](https://docs.automotivelinux.org/en/needlefish/#0Getting%20Started/3%20Build%20and%20Boot%20guide%20Profile/1%20Instrument%20Cluster%20IC-IVI%20with%20Container%20isolation%29/)

The screenshot shows the 'Build and Boot AGL Instrument Cluster demo image (IC-IVI with Container isolation)' page. The left sidebar contains a navigation menu with sections like 'Getting Started', 'Hardware Support', and 'How To Contribute'. The main content area is titled 'Build and Boot AGL Instrument Cluster demo image (IC-IVI with Container isolation)' and lists 'Required Equipments' and '0. Host PC environment'. It then provides step-by-step instructions for building the image, including setting environment variables, downloading the repo tool, and cloning the AGL source files. The instructions are presented in a clean, modern layout with code blocks and numbered steps.

Home

◀ Previous Building for virtio Supported Boards Next ▶

## Build and Boot AGL Instrument Cluster demo image (IC-IVI with Container isolation)

### Required Equipments

- 1) Tested board: Starter Kit Pro/M3 + kingfisher support
- 2) MicroUSB
- 3) MicroSD card

### 0. Host PC environment

Build PC Ubuntu OS (Tested version 18.04.6 LTS)

#### 1. Define Your Top-Level Directory

```
export AGL_TOP=$HOME/AGL
mkdir -p $AGL_TOP
```

#### 2. Download the repo Tool and Set Permissions

```
mkdir -p $HOME/bin
export PATH=$HOME/bin:$PATH
curl https://storage.googleapis.com/git-repo-downloads/repo > $HOME/bin/repo
chmod a+x $HOME/bin/repo
```

#### 3. Download the AGL Source Files

- AGL Version: Nifty Needlefish (tip of branch)

```
cd $AGL_TOP
mkdir needlefish
export AGL_TOP=$HOME/AGL/needlefish
cd $AGL_TOP
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
repo init -b needlefish -u https://gerrit.automotivelinux.org/gerrit/AGL-repo
repo sync
```

- Optional: Specify the manifest file of a point release (needlefish\_14.0.1.xml) using -m option

```
repo init -b needlefish -m needlefish_14.0.1.xml -u https://gerrit.automotivelinux.org/gerrit/AGL-repo
```

- Optional: Specify the master branch

```
repo init -u https://gerrit.automotivelinux.org/gerrit/AGL-repo
```

#### 4. Downloading Proprietary Drivers

Downloading Proprietary Drivers from [Renesas-automotive-products](#) - To check the files to download

v: needlefish

- 
1. Introduction :Background AGL and Instrument Cluster
  2. About Linux Container
  3. Feasibility study using Container and AGL
  4. Issue and Result
  5. **Conclusion**

# Conclusion

---

- Select LXC for automotive system as system container.
  - Isolate both IVI and Cluster root filesystem.
  - Can separate the other profile to control resource usage.
- Easy update and exchange containers.
  - Light-weight updates filesystem instead of system.
  - Easy integrate additional IVI profile.
- Feature work
  - Will add other peripheral device control (sound, video, etc.)
  - Will develop another feature container like browse-container as high load container.
  - Will develop more systematic Isolation-system using OSS-RTOS for Automotive usecase

# One more ....

“Automotive Grade Linux JP” account is opened !!

We will help someone's question about AGL  
in Japanese Lang.

- What is AGL?
- How to join to develop ?
- What's the most interest feature?
- It is difficult to ask in English....
- Etc..

Please follow @autolinuxjp right now!

Let's Join and Develop AGL !





Thank You

@autolinuxjp



# AUTOMOTIVE LINUX SUMMIT

— @ —



OPEN SOURCE SUMMIT  
JAPAN

THE LINUX FOUNDATION