

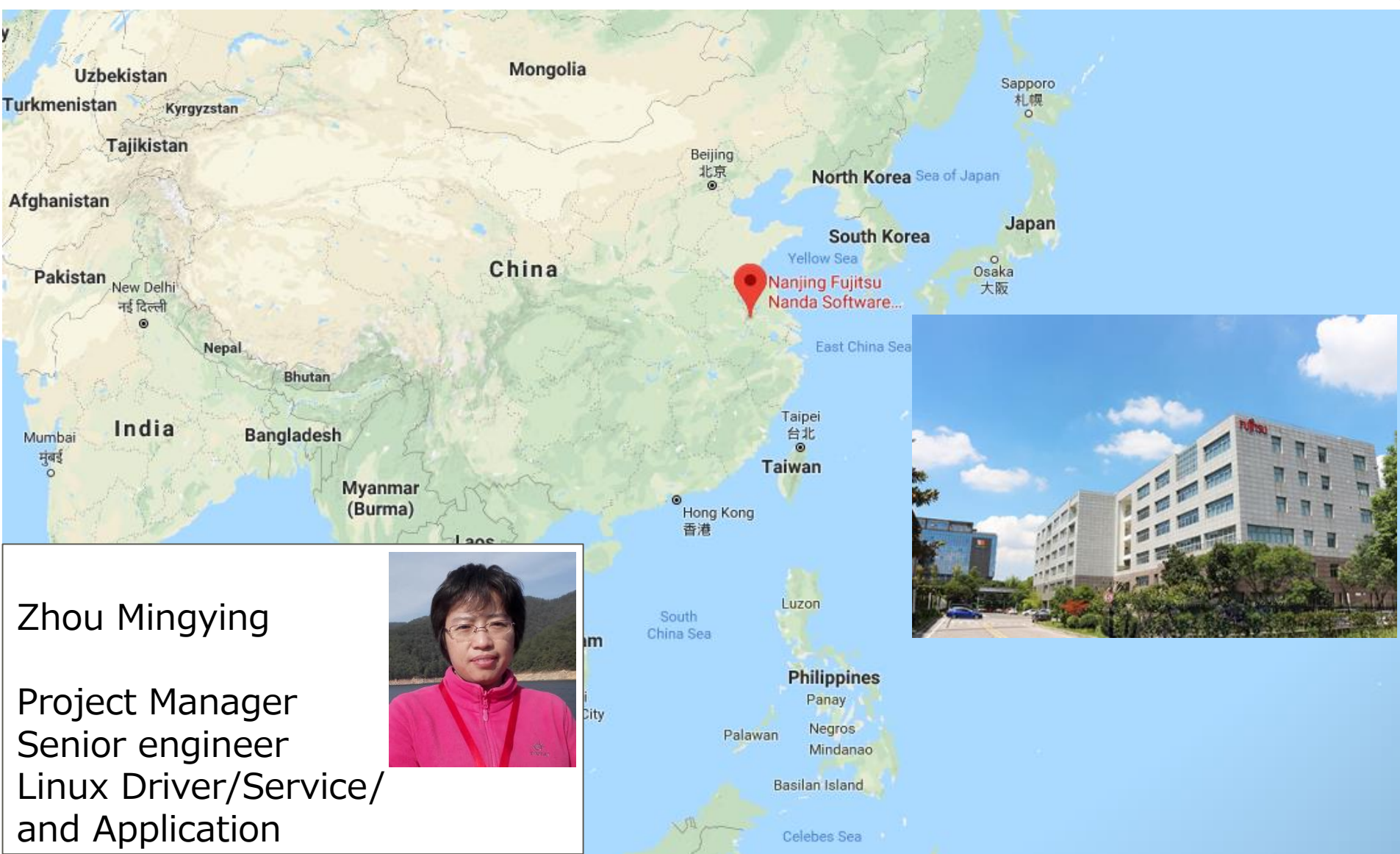
Minimize AGL For Low Profile

July 18 2019
Automotive Linux Submit, Tokyo

Zhou Mingying
Nanjing Fujitsu Nanda Software Tech. Co., Ltd.(FNST)



Self Introduction



Zhou Mingying
Project Manager
Senior engineer
Linux Driver/Service/
and Application



intelligent vehicle

FNST Automotive Business



AI and Etc.
Advanced Tech.

Edge
Computing

Image
Recognition

Operation
Prediction

Chart
Robot



In-Vehicle Embedded

In Vehicle
Device

Drive Record

Digital·Tachograph

DCM/TCU/T-BOX

Adaptive
Cruise control

Multimedia

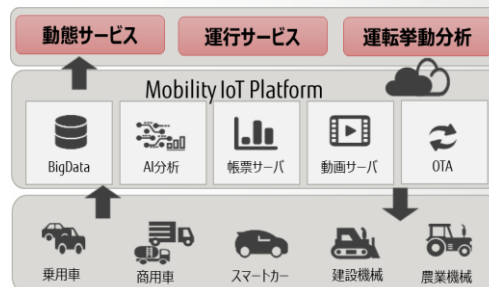
Multimedia
Service

Navigation

Infotainment Device



Mobility as a Service



Cloud

Error Analysis

Runtime Manage
System

MaaS Platform



Automotive
Electronics

Vehicle
Info

Engine

Power Train
Control

Engine cooling
System

Power Train
Control

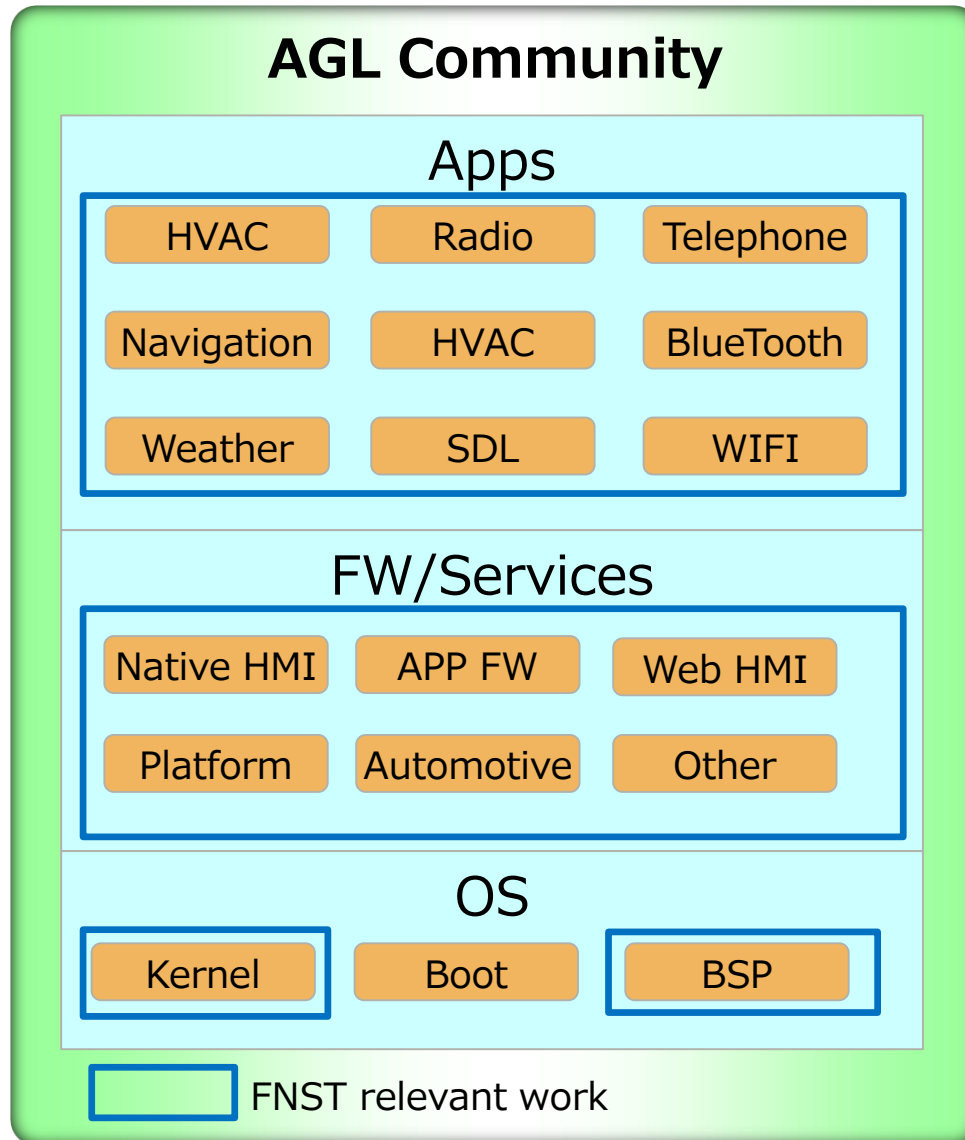
Engine ECU

Control
ECU

Transmission ECU

fastECU

Intelligent Vehicle



Intelligent Vehicle

Agenda

- Why minimize AGL
- How to minimize AGL
- The effect of AGL minimization
- What to do next



Why minimize AGL

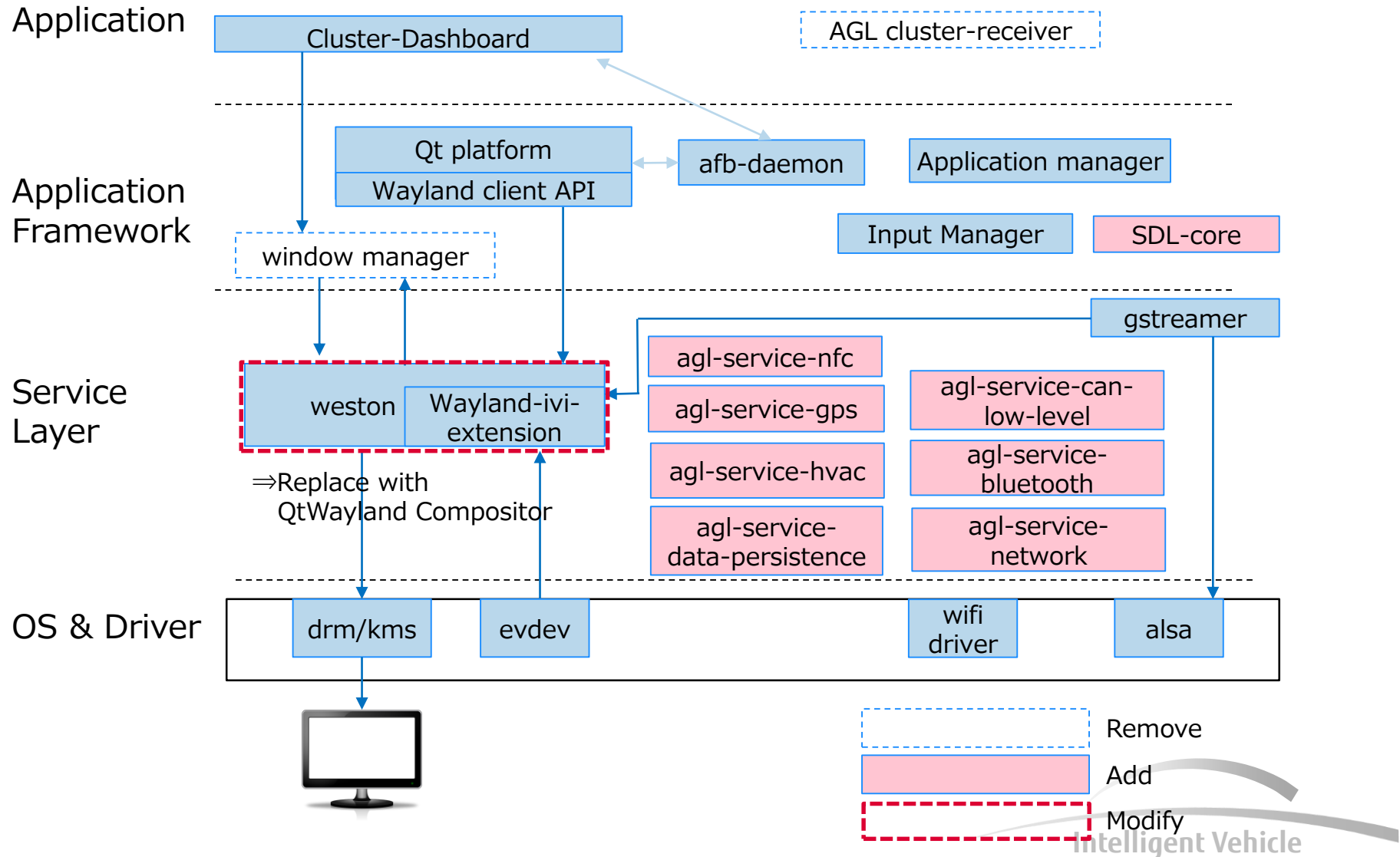
- Low-end SoC
- Profile Architecture



- The status of AGL now (stable AGL7.0.3)
 - High-end is main now, such as R-Car H3,M3
 - Action on low-end SoC is slow because of a lot of Apps and Services
- What we want to do
 - low-end , such as E3,D3 for low-profile in emerging countries
 - Less applications, just vehicle information and smart phone connection, such as
 - agl-service-can-low-level
 - agl-service-hvac
 - SDL(SmartDeviceLink)



Profile Architecture



How to minimize AGL

- Add a new image option
- Remove apps and add services
- Replace compositor
- Simulate low-end SoC with M3



Add a new image option

■ Add “agl-demo-minimal”

- add bb file to meta-agl-cluster-demo

```
$ ls meta-agl-cluster-demo/recipes-platform/images/  
agl-demo-minimal-crosssdk.bb  agl-demo-minimal.bb  agl-demo-minimal.inc
```

For agl-demo-minimal-
crosssdk option

For agl-demo-minimal
option

Include file

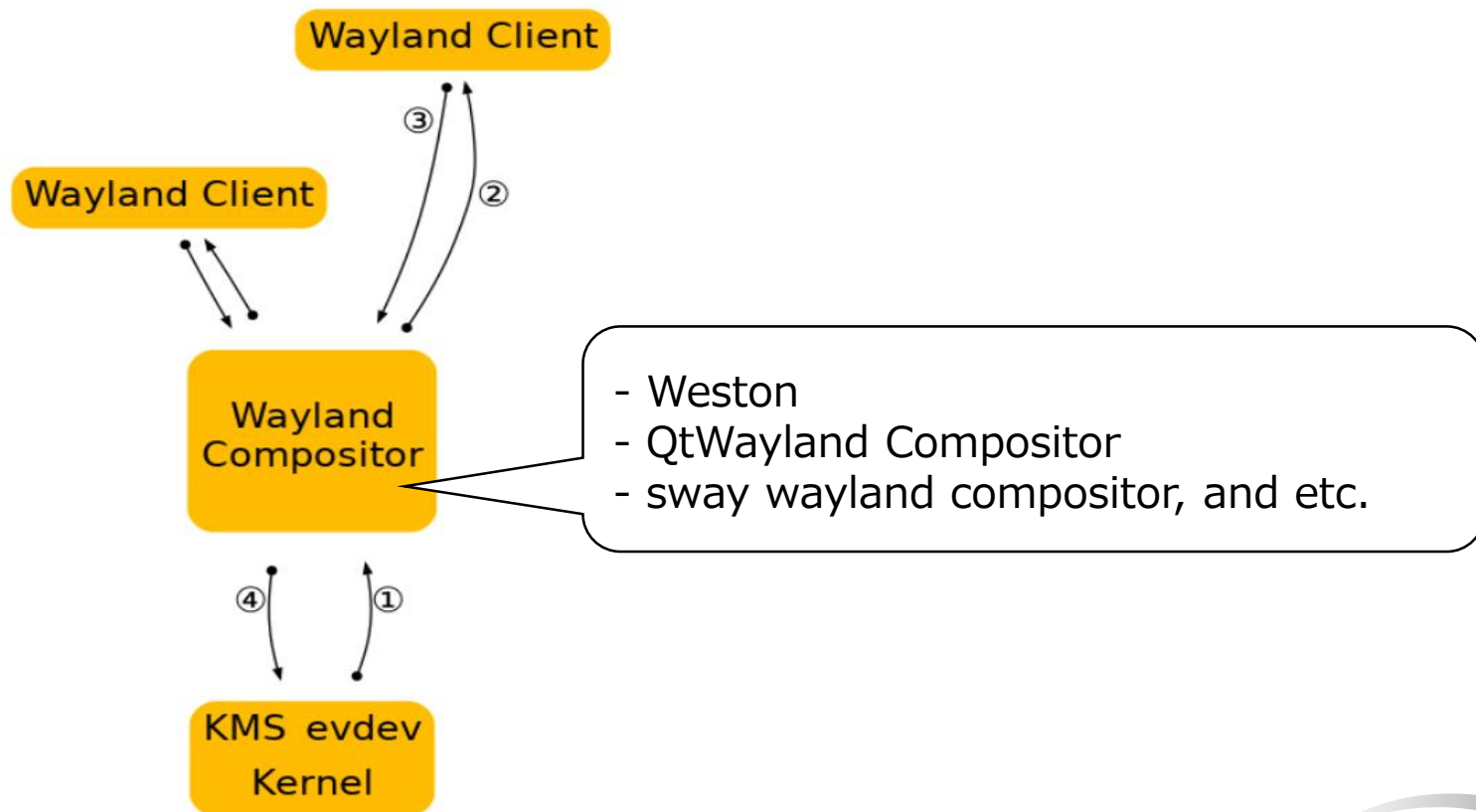
Though add the new option, no effect to other existing images

- Add bb file for agl-demo-minimal image
 - recipes-platform¥packagegroups¥packagegroup-agl-demo-minimal.bb
 - No effect to existing images
 - Define necessary components
 - Other components will be excluded
 - Qt Lite
 - When remove apps, the dependency Qt components will be excluded
 - Remove charts, virtual board and xml components, etc.
 - Remove unused language packages
 - Chinese
 - Japanese
 - Patch path (Under review):
 - <https://gerrit.automotivelinux.org/gerrit/c/AGL/meta-agl-cluster-demo/+21880>

Replace compositor(1)

■ What's Wayland compositor

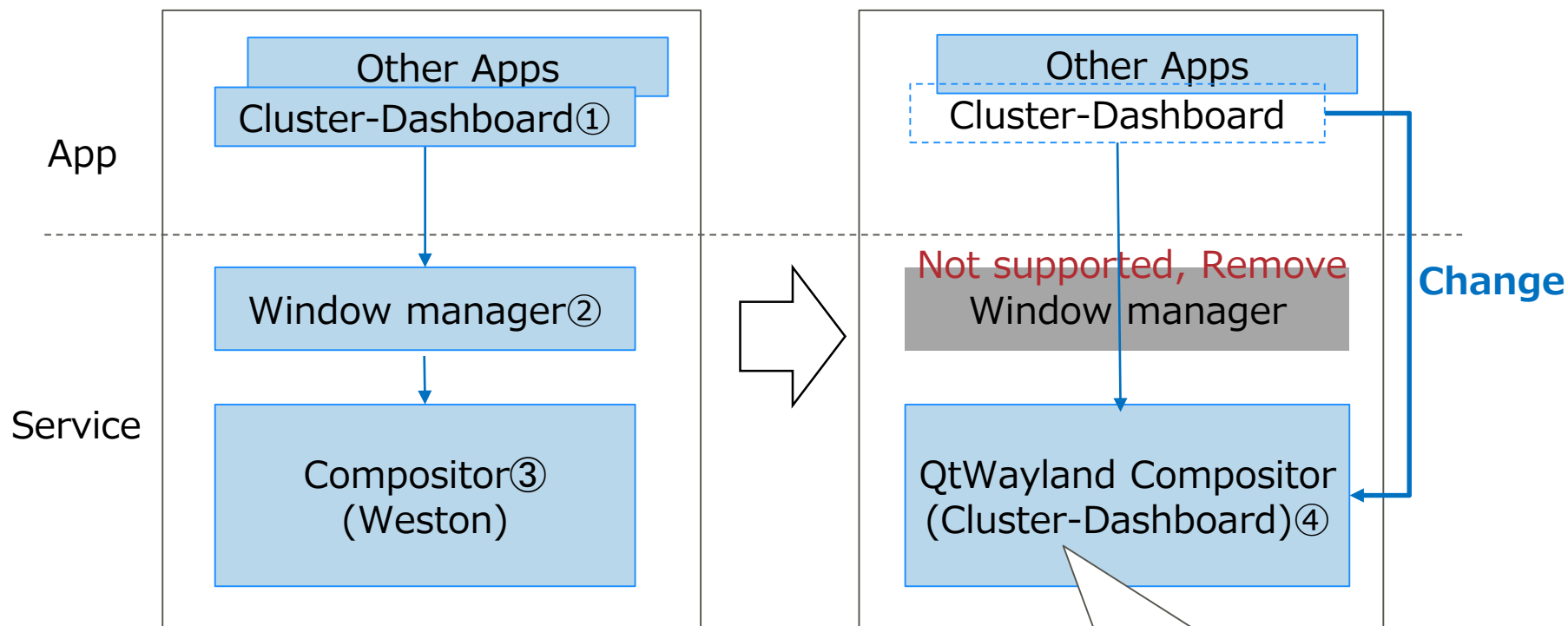
- Wayland is a display server and compositing protocol
- Wayland compositor is the implementation of the protocol



Note: Picture is from <https://wayland.freedesktop.org/docs/html/ch01.html#sect-Compositing-manager-display-server>

Replace compositor(2)

■ Replace Weston with QtWayland compositor



- 3 processes(①②③) => 1 process(④)
- Customized compositor is more flexible

Customized compositor.
But protocol ivi-
controller and ivi-wm
aren't supported

Replace compositor(3)

■ Action items

- Build plugin
- Create QtWayland compositor sample
 - Cluster Demo (Cluster-Dashboard)
- Add configure file kms.conf
- Create QtWayland compositor service

Build plugin

■ Compile qtbase with eglfs, gbm and kms (AGL7.0.3)

- Step 1: modify qtbase_%.bbappend

Add

```
PACKAGECONFIG_GL = "${@bb.utils.contains('DISTRO_FEATURES', 'opengl', 'gles2_eglfs kms gbm', '', d)}"
```

- Step 2: build

```
$source meta-agl/scripts/aglsetup.sh -m m3ulcb -b build agl-cluster-demo  
$bitbake qtbase
```

or

```
$source meta-agl/scripts/aglsetup.sh -m m3ulcb -b build agl-cluster-demo  
$bitbake agl-demo-minimal
```

■ Plugin libqeglfs-kms-integration.so

- After building, the plugin libqeglfs-kms-integration.so is included by qtbase-plugins-5.9.7+git0+81b29a44d2-r0.aarch64.rpm
- Install plugin
 - Install the qtbase-plugins-5.9.7+git0+81b29a44d2-r0.aarch64.rpm
 - Rewrite image file agl-demo-minimal

Create QtWayland compositor

■ Create a new sample compositor Cluster-Dashboard

- Delete window manager relevant code
- Create bb file
 - recipes-demo-hmi¥cluster-dashboard-sample¥cluster-dashboard-sample_git.bb
- Compile and install

■ Stop Weston

```
#killall weston
```

- If have removed Weston and window manager, “killall Weston” isn’t needed

■ Start QtWayland compositor

- Cluster-Dashboard-sample

If create a service, it can be started when the OS start

```
#!/var/local/lib/afm/applications/cluster-gauges/1.0-923279f-dirty/bin/cluster-gauges --platform eglfs
```

Before starting QtWayland compositor, should make a kms.conf file

Add configure file kms.conf

■ kms.conf (example)

```
m3ulcb:~# cat /etc/kms.conf
{
  "device": "/dev/dri/card0",
  "outputs": [
    {
      "name": "LVDS1",
      "mode": "off"
    }
  ]
}
```

You can get some sample from the bellow address:

<https://doc.qt.io/qt-5/embedded-linux.html>

You can define the kms.conf file as you need.



Create QtWayland compositor service

■ Add detail to bb file

- recipes-demo-hmi¥cluster-dashboard-sample¥cluster-dashboard-sample_git.bb

```
inherit systemd
SRC_URI_append = " ¥
    file://0001-Modify-cluster-dashboard-as-a-qtwayland-compositor.patch ¥
    file://0002-Add-kms-conf-file.patch ¥
    file://qtwaylandcompositor.service ¥
"

do_install_append() {
    install -d ${D}/usr/bin
    install -m 755 ${WORKDIR}/build/app/cluster-gauges ${D}/usr/bin
    install -d ${D}/etc
    install -m 644 ${S}/kms.conf ${D}/etc
    install -d ${D}${systemd_unitdir}/system
    install -m 644 ${WORKDIR}/qtwaylandcompositor.service ${D}${systemd_unitdir}/system/qtwaylandcompositor.service
}

SYSTEMD_SERVICE_${PN} = "qtwaylandcompositor.service"
```

Remove window manager relevant code

Add kms.conf file

Add compositor service file

Add the above code to bb file, the service will be installed at the first start of AGL. And can be started from next OS start

Simulate low-end SoC with M3

- We don't have Low SoC, use M3 to instead of low SOC
- CPU&Memory limit set to simulate Low SoC

- Bootargs modify

⇒ editenv bootargs

```
edit: console=ttySC0,115200 root=/dev/mmcblk1p1 rootwait ro  
rootfstype=ext4 maxcpus=1 mem=896M
```

- DTS file modify (example)

The diagram shows a DTS file with four sections highlighted in red boxes, each with a corresponding callout explaining the modification:

- Section 1:** `/* device specific region for Lossy Decompression */`
`/*lossy_decompress: linux,lossy_decompress {`
`no-map;`
`reg = <0x00000000 0x54000000 0x0 0x03000000>;`
`};*/`
Callout: we can Comment Out lossy_decompress
- Section 2:** `/* For Audio DSP */`
`/* adsp_reserved: linux,adsp {`
`compatible = "shared-dma-pool";`
`reusable;`
`reg = <0x00000000 0x57000000 0x0 0x01000000>;`
`}; */`
Callout: we can Comment Out Audio DSP
Callout: we also need to comment out the &adsp section as caution1
- Section 3:** `/* global autoconfigured region for contiguous allocations */`
`linux,cma {`
`compatible = "shared-dma-pool";`
`reusable;`
`reg = <0x00000000 0x57000000 0x0 0x10000000>;`
`linux,cma-default;`
`};`
Callout: Modify Default Size to a little value
Default: 0x18000000(384MB)
Example: 0x10000000(256MB)
0x04000000 (64MB) is too small and will App will not be run because of short of Buffer such as "rcar-du feb00000.display: failed to allocate buffer with size 6475776""
- Section 4:** `/* device specific region for contiguous allocations */`
`mmp_reserved: linux,multimedia {`
`compatible = "shared-dma-pool";`
`reusable;`
`reg = <0x00000000 0x67000000 0x0 0x04000000>;`
`};`
Callout: Modify Default Size to a little value
Default: 0x10000000(256MB)
Example: 0x04000000(64MB)
Ebisu's Dts about Multimedia Dma-pool is set to 64MB as default value

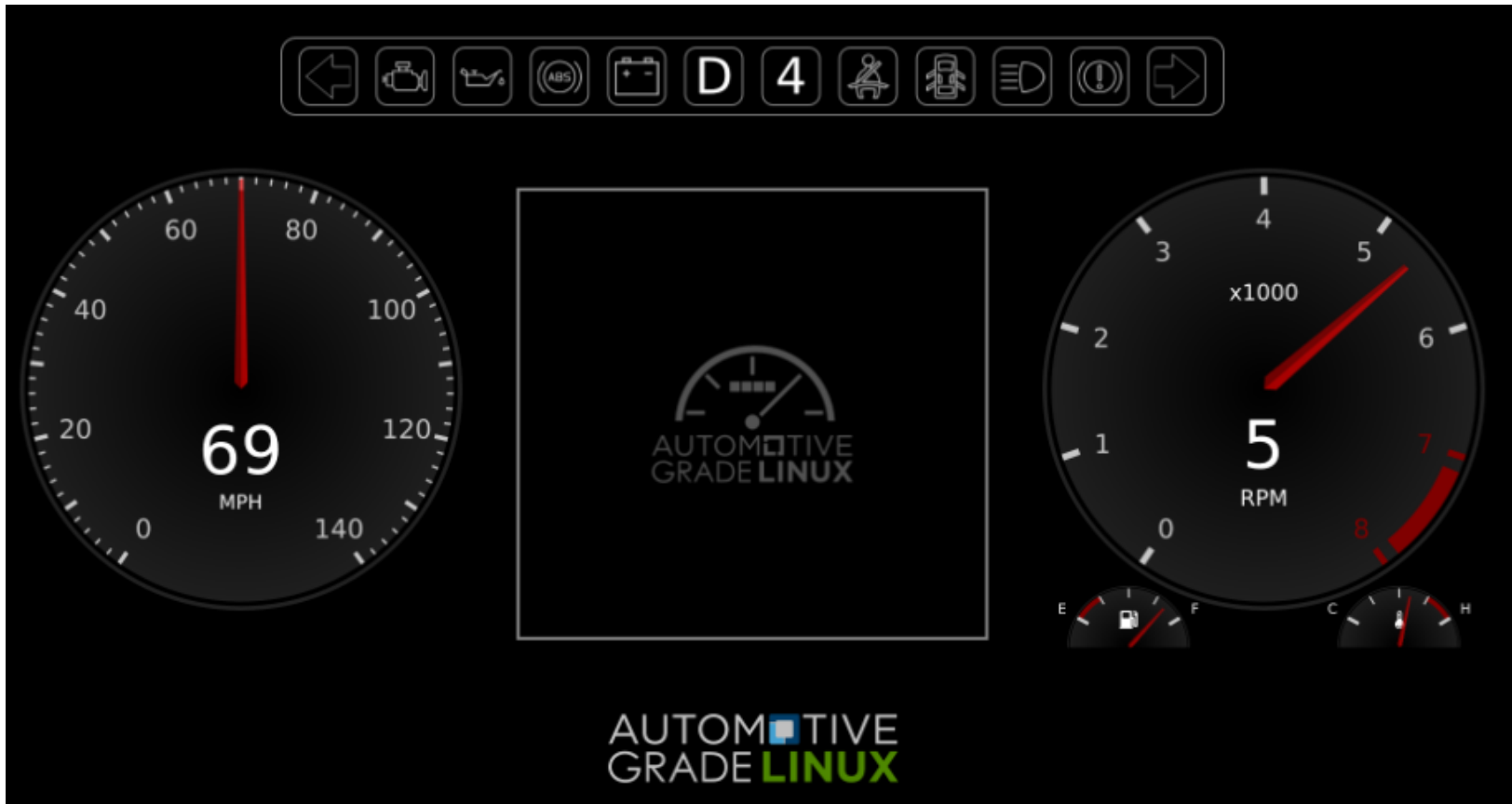
The effect of AGL minimization

- Demo screen
- Packages & ROM & RAM
- Performance



Demo Screen

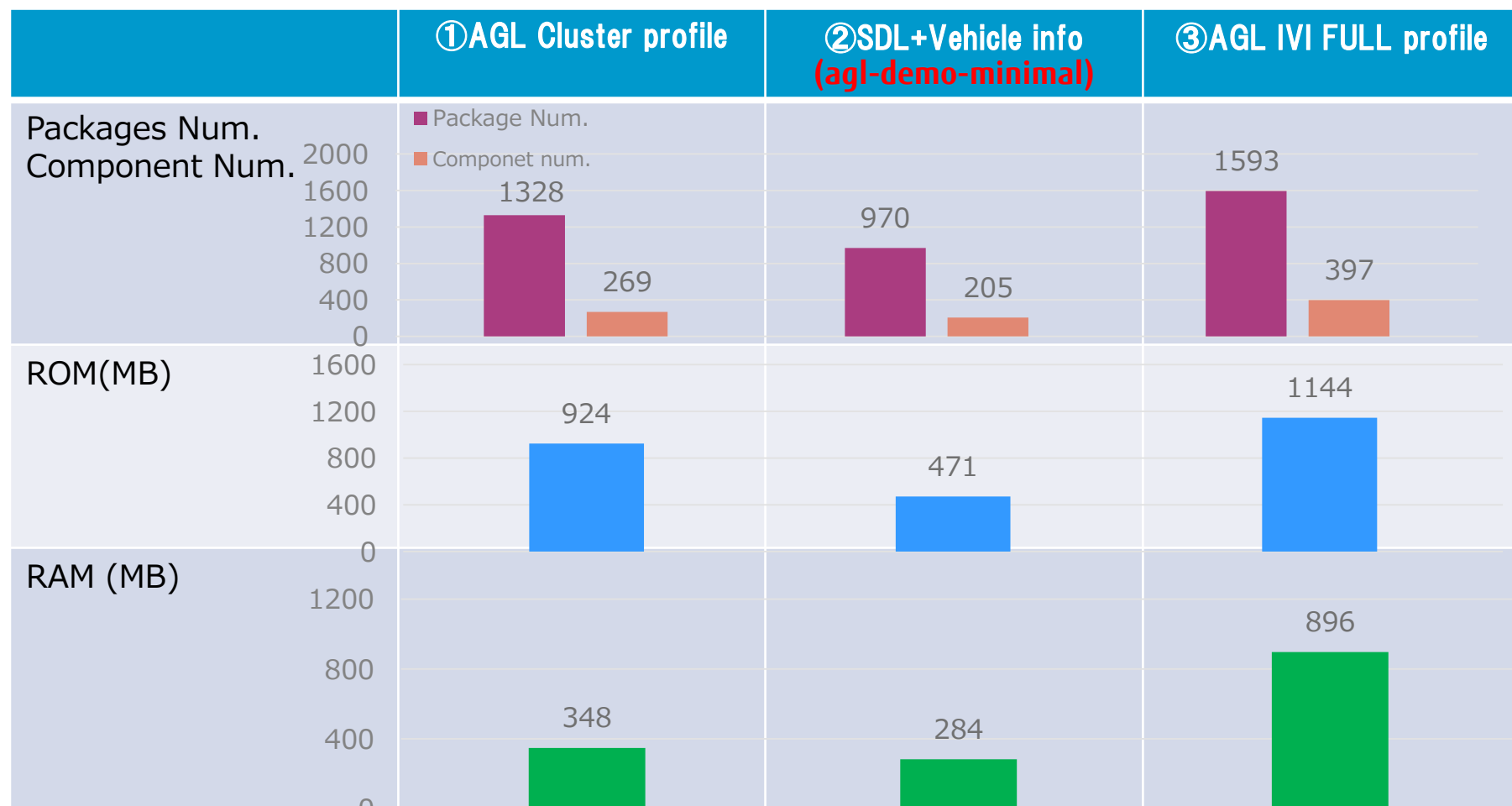
- Same as AGL's agl-cluster-demo-platform's screen



Packages&ROM&RAM

■ Packages&ROM&RAM is smallest on ②SDL+Vehicle

■ Remove Window Manager and Languages, Qt Lite

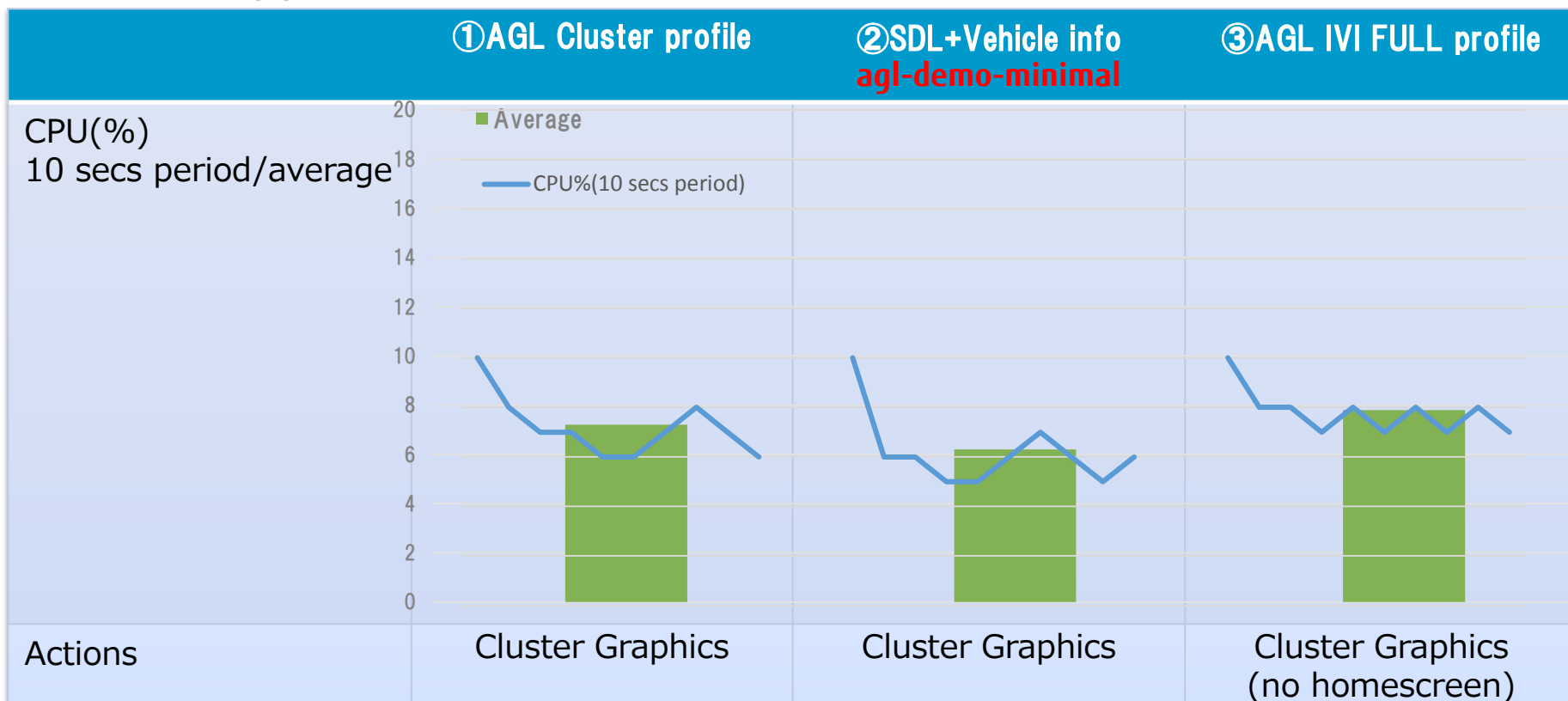


Test version AGL7.0.2

Intelligent Vehicle

■ CPU(%) is smallest on ②SDL+Vehicle

- Remove Weston&Window Manager (① ③ use window Manager)
- Less applications



- Test version AGL7.0.2
- Test after booting(Cluster Graphics), no loaded
- Boot with 2CPU on M3 (simulate E3)



What to do next?

- Help Qt to support other protocols
- Modify applications



Help Qt support other protocol

- Qt supports ivi-application only (2019.7)
- Other protocol

Weston support the bellow protocols

- ivi-controller
 - AGL used ivi-controller before 2018.10
- ivi-wm
 - From 2018.2, AGL start to use ivi-wm protocol

Make Qt Wayland Compositor to support ivi-wm,
This will help QtWayland compositor more flexible.
The existing applications don't need to modify.

■ Modify application

- The original applications depends on window manager
- After removing window manager, the application should be modified

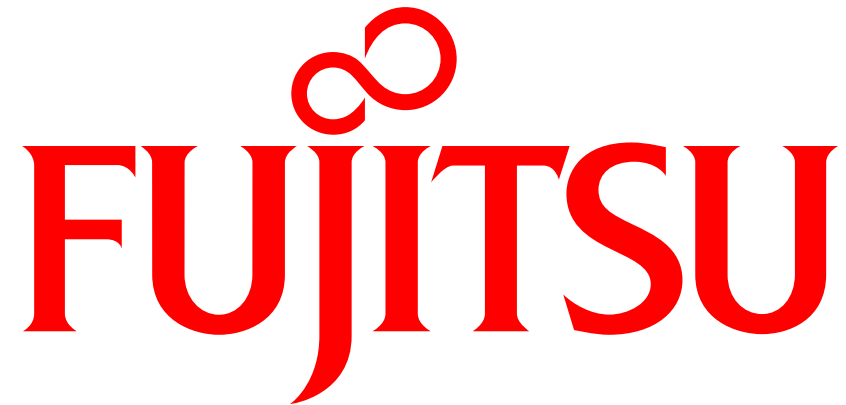
Questions?



Thank you!

zhoumy@cn.fujitsu.com





shaping tomorrow with you