

Proposal(Updated) for AGL HMI-Framework

AGL All-Member Meeting @ TOKYO
February 2018

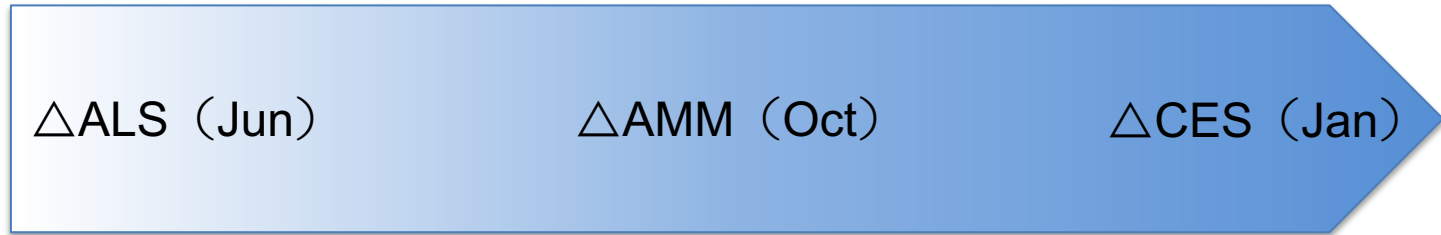
Toshikazu Oiwa
toshikazu_ohiwa@mail.toyota.co.jp

TOYOTA MOTOR CORPORATION

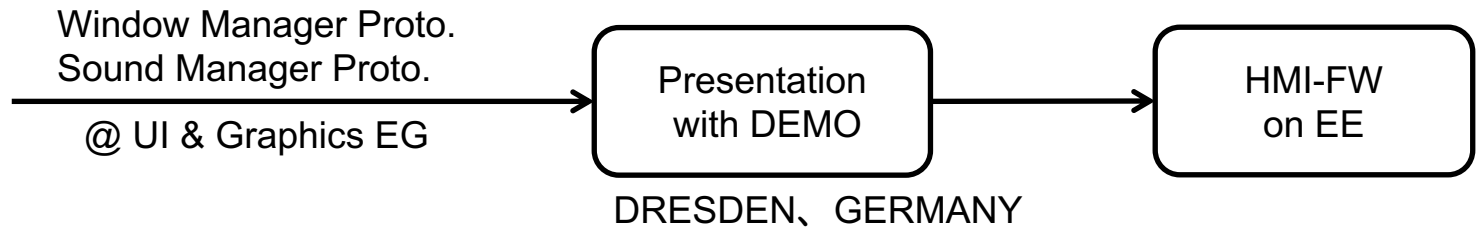
Who is Oiwa?

- Software engineer, expert in in-vehicle infotainment.
- Have been developing software for in-vehicle infotainment system such as apps, services since 1994.
- In charge of HMI-Framework.

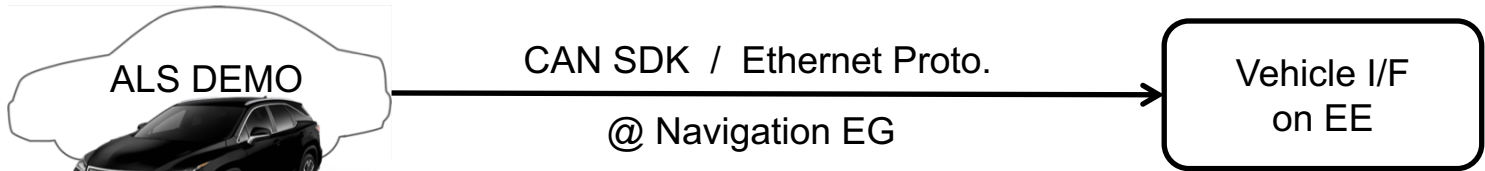
2017 AGL Development Plan of TOYOTA



HMI-FW



Vehicle I/F



Reference Hardware



Table of Contents

- Background of HMI-Framework
- Proposal for HMI-Framework
 - OEM can choose GUI
 - OEM can choose HMI-Manager
 - HMI-Framework can arbitrate HMI Resources
- Demonstration
- Schedule
- Conclusion

Background of HMI-Framework

Background for in-vehicle HMI #2

- Apps compete to output information in in-vehicle system HMI

Running at the same time in in-vehicle system.

Compete for getting resources(screen,speaker) to output information.



Wants to display map



Wants to display control of HVAC



can't appropriately place on screen

⇒Lack of proper management results in driver distraction.

In addition, resources management would be different by system configuration such as low-end systems or high-end systems.

Requirements for HMI-Framework

- ① OEM can choose GUI
- ② OEM can choose HMI-Manager
- ③ HMI-FW can arbitrate HMI resources

Why requirement① ?

- Comfortable GUI "GUI for better UX" (background #1)
 - Intuitively easy to understand (3 D)
 - Animation (Graphic Effects)Wants to be able to use the most suitable GUI.

Why requirement②③ ?

- Safe HMI(background #2)
 - Manage resources(Window-Resources/Sound-Resources)
 - Arbitration of Window-Resources/Sound-ResourcesWants to be able to use the most suitable software to system.
Wants to be able to arbitrate resources.

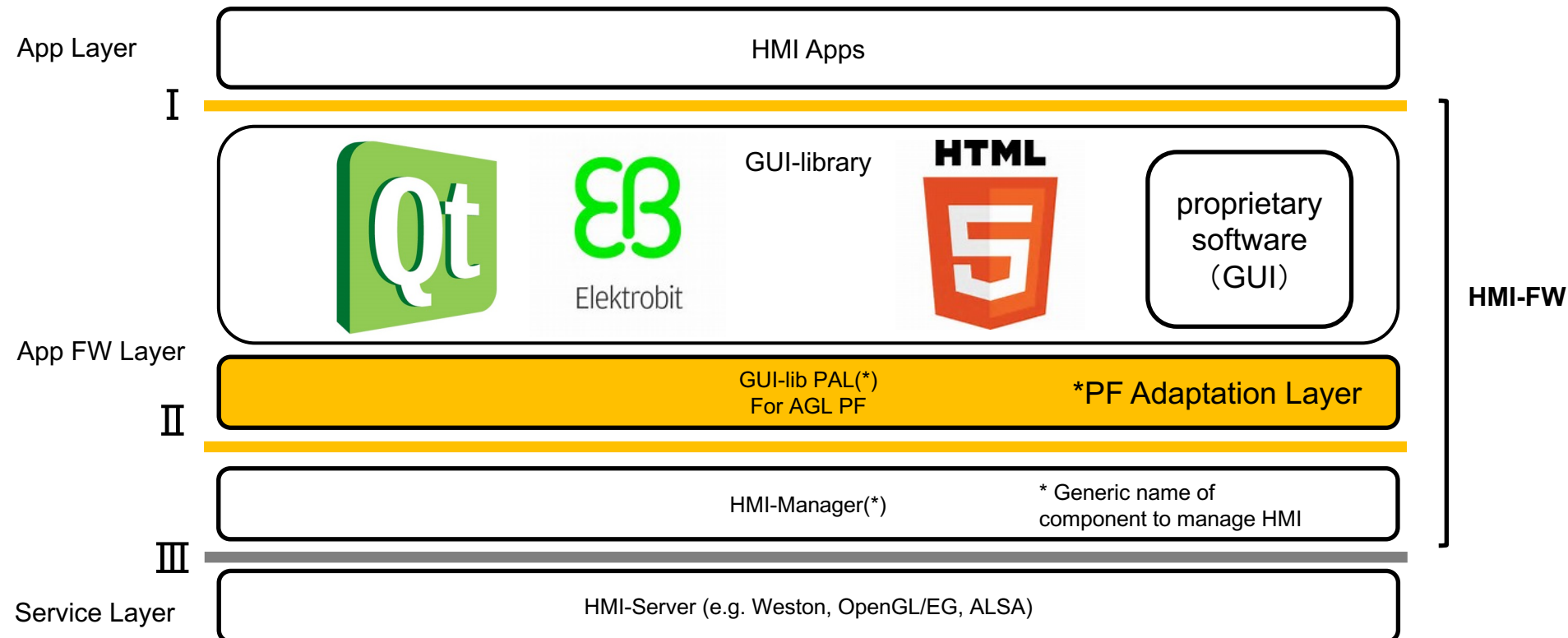
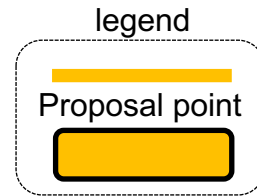
Proposal of HMI-Framework

- OEM can choose GUI
- OEM can choose HMI-Manager
- HMI-FW can arbitrate HMI resources

Requirement① OEM can choose GUI

I. No change in API which Apps refer to
Apps can use API of each GUI-lib.

II. Each GUI-lib has layer to adapt to PF
Can have many different GUI-lib without change of PF.



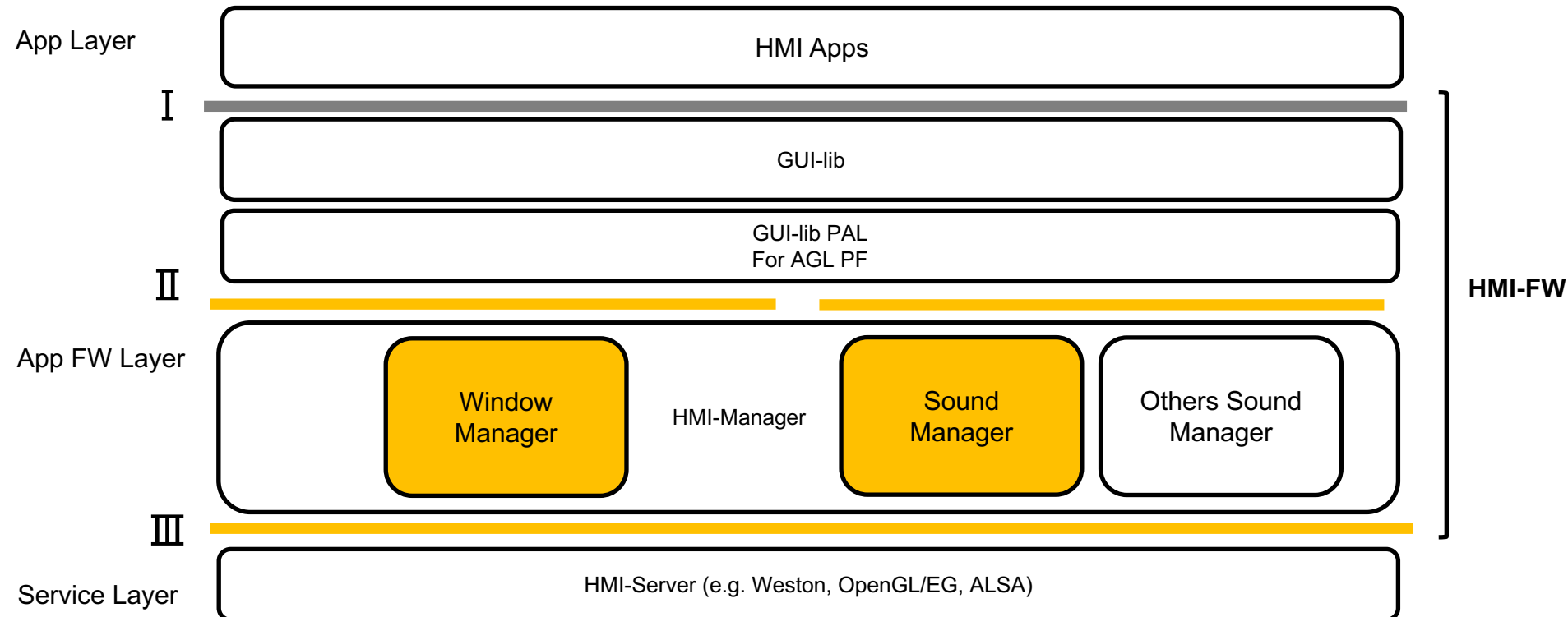
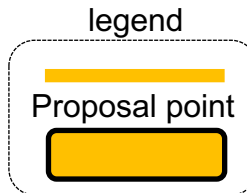
Requirement② OEM can choose HMI-Manager(*)

* Generic name of component to manage HMI

* Decides the optimum layout and controls(e.g. screen), based request from apps.

II. No change in API which Apps and GUI-lib refer to
 Apps and GUI-lib are available to use unique API of HMI-Manager.

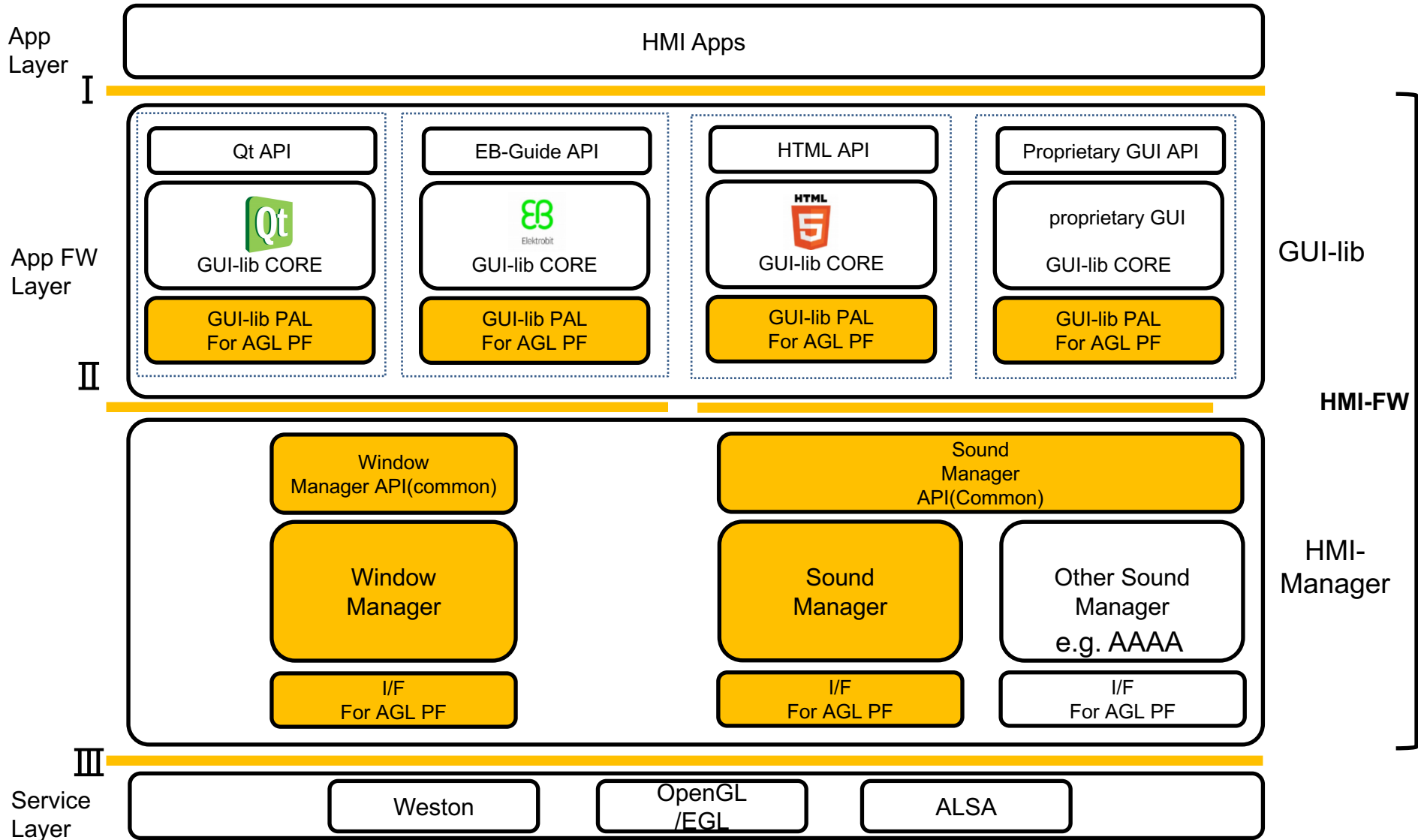
III. Each HMI-Manager has interface to adapt to service layer
 Can have different HMI-Manager without change in PF.



Components for Requirement①②

legend

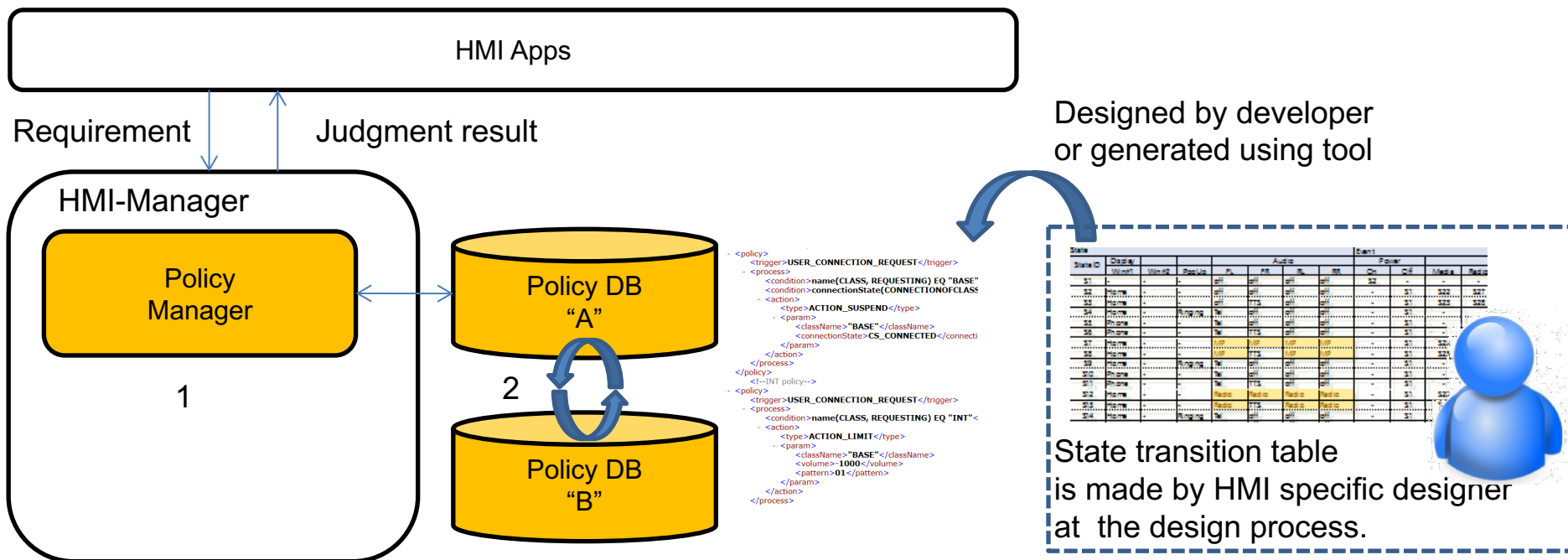
Proposal point



Requirement③ HMI-FW can arbitrate HMI resources

1. Judges use of resources for request from Apps(Policy Manager)
Upon request (Event) from Apps,
Policy Manager decides which App can allocate resources.

2. Has a rule of arbitration as Policy DB
OEM can easily change Policy DB without changing HMI Manager.
Policy DB is described in a format such as XML.



Structure of HMI-Manager including Policy Manager

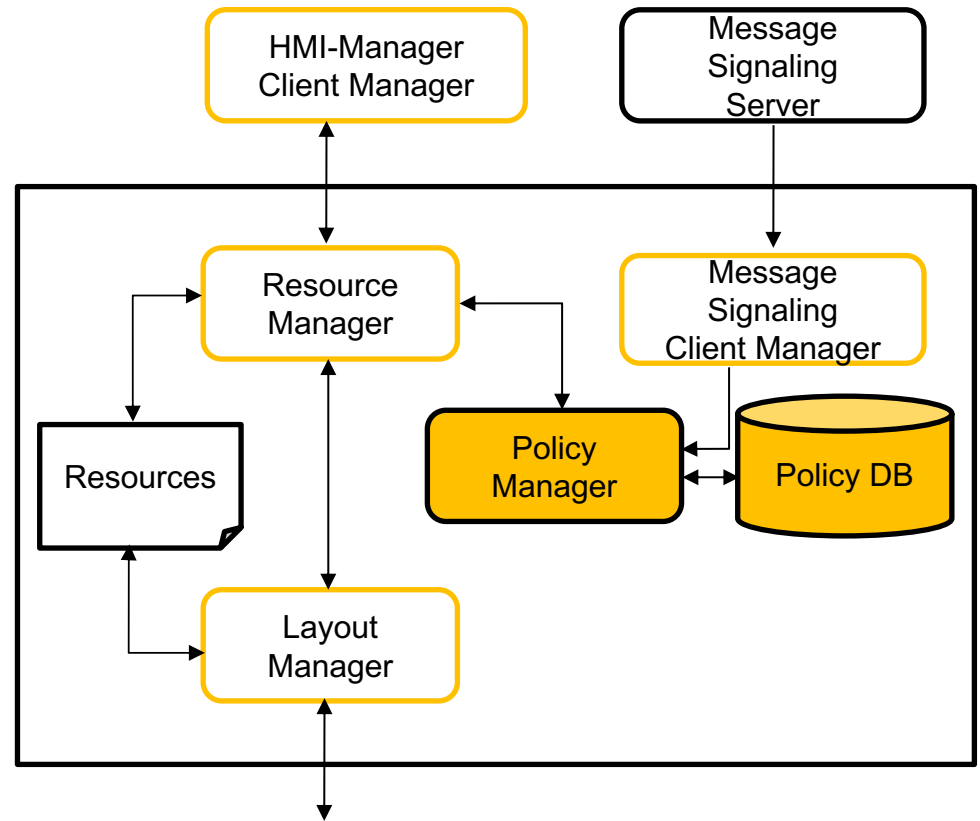
Decides the optimum screen layout or speaker layout

and controls screen or speaker, based on request from Apps.

Window Manager/Sound Manager in HMI manager are the same in structure.

[Main components]

- Policy Manager
- Resource Manager
Manages resource information such as display, speaker.
- Layout Manager
Manages layout according to judgment result by Policy Manager.



Reference

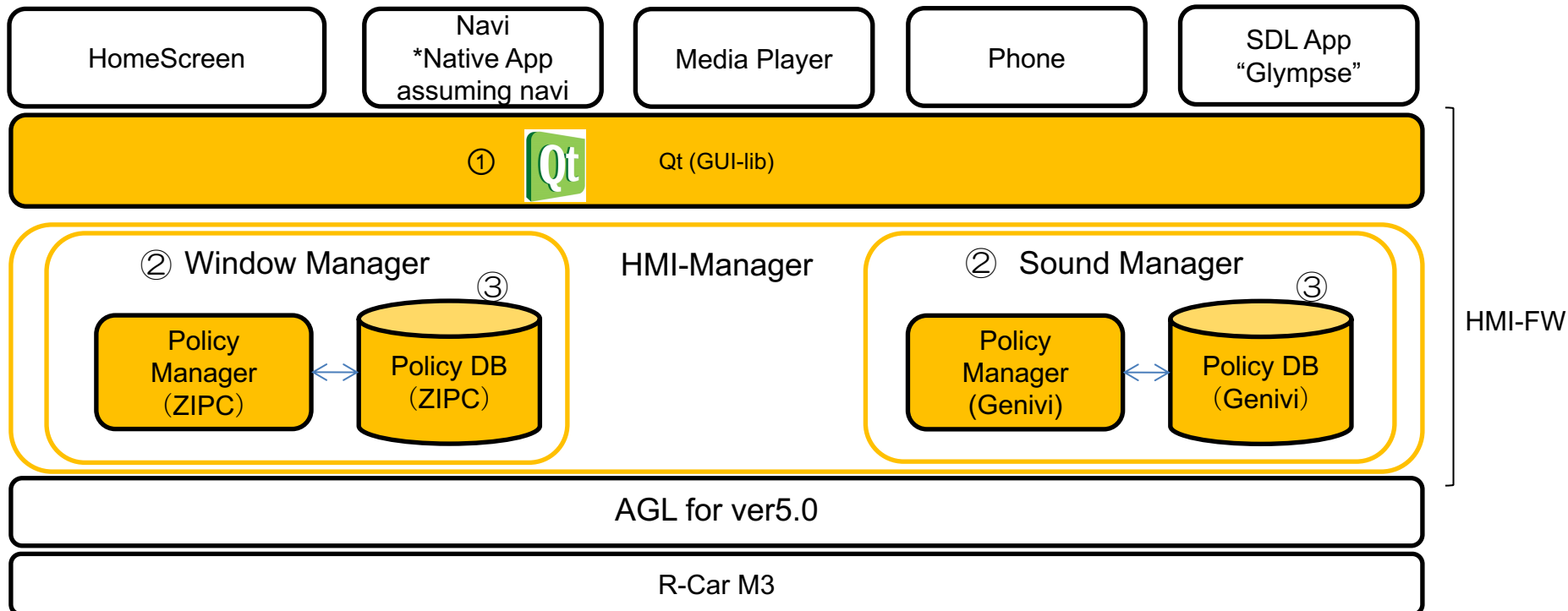
<https://wiki.automotivelinux.org/hmiframework>

General Information “HMI-Framework Architecture”

Demonstration

Structure of demonstration

- Built software for AGL 5.0 on M3
- Demonstration's Apps are based on Apps of CES2018
- HMI-Framework supports AFB
- Choose Qt for Req.①
- Choose Audio Manager(Genivi) for Req.②, Window Manager and Sound Manager are updated
- Window Policy DB is ZIPC format, Sound Policy DB is Genivi format for Req.③

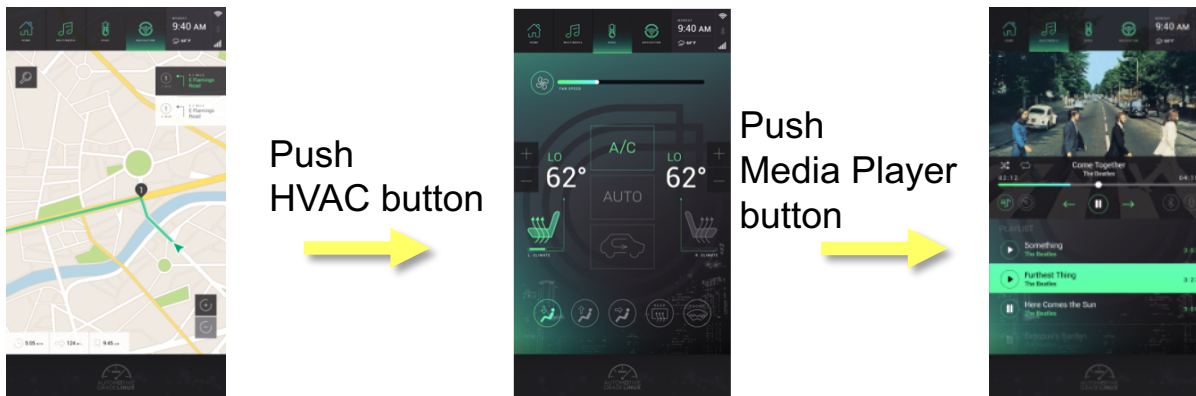


Demo① Changing the behavior of HMI

➤ Use case:

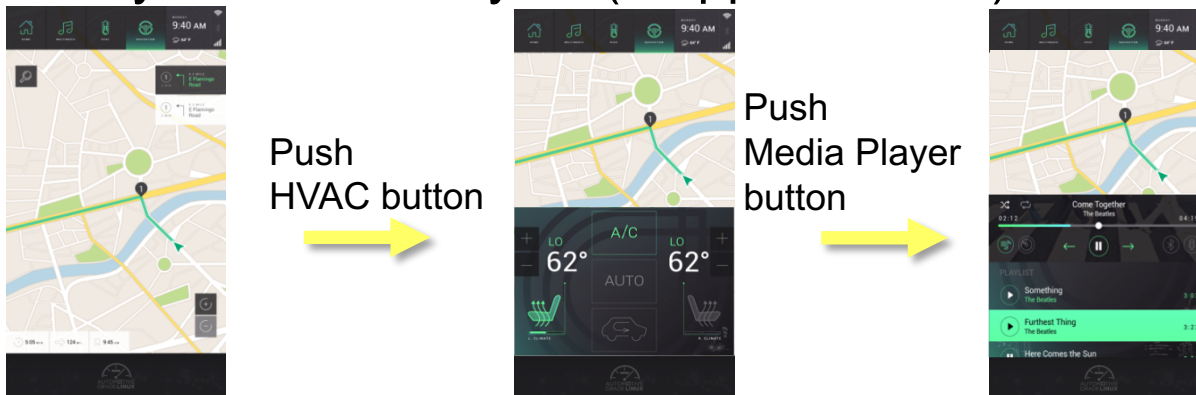
Changes the behavior of HMI according to destination or grade.

Policy A : Screen layout (1 App: 1 screen)



Step1:
screen changes
according to Policy A.

Policy B : Screen layout (2 App: 1 screen)



Step2:
Changed from Policy A to Policy B.
And restart the system.

Step3:
screen changes
according to Policy B.

※In this demonstration, “simple-egl” is used instead of “map” of native app.

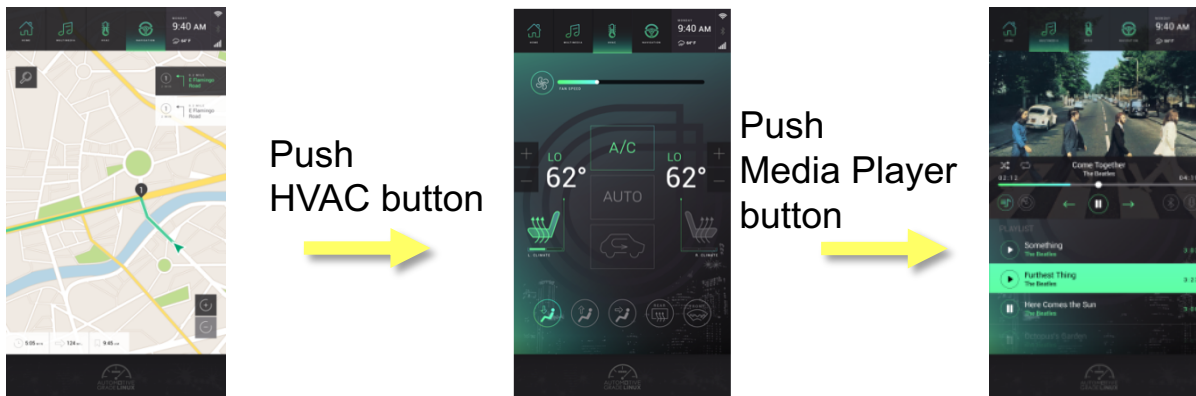
© TOYOTA MOTOR CORPORATION All Rights Reserved.

Demo① Changing the behavior of HMI

➤ Use case:

Changes the behavior of HMI according to destination or grade.

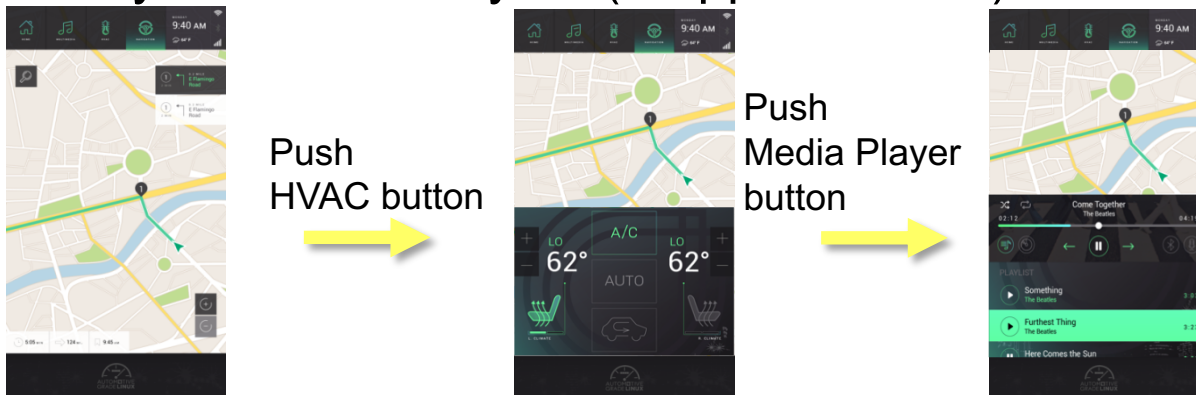
Policy A : Screen layout (1 App: 1 screen)



Step1:
screen changes
according to Policy A.

Step2:
Changed from Policy A to Policy B.
And restart the system.

Policy B : Screen layout (2 App: 1 screen)



Step3:
screen changes
according to Policy B.

※In this demonstration, “simple-egl” is used instead of “map” of native app.

Demo② Arbitration of HMI Resources

➤ Use case :

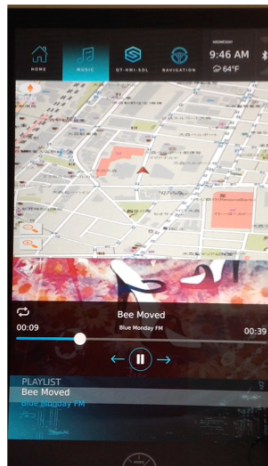
When executing another application during navigation map display on running, split the display screen and display multiple applications.

NEW

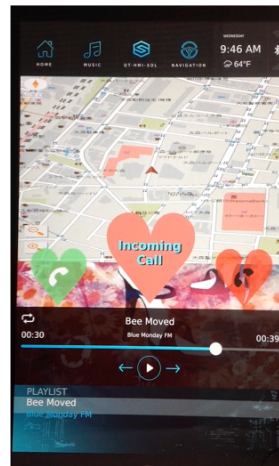
- Screen transitions mixed application's normal and half size display
- Non-rectangular or rectangular on-screen display



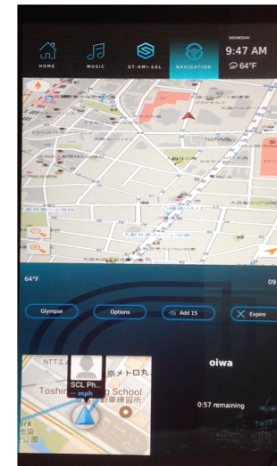
Normal size display



Half size display



On-Screen display



Screen transition of half size app

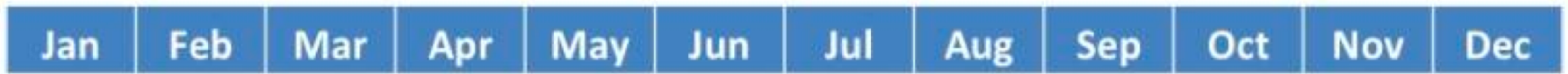


Screen transition to normal size app

Schedule

Schedule for HMI-Framework development

◇ Release



Electric Eel

Patch Updates

HMI-FW has already been released in UCB5.0(EE)

- Screen layout (1 App: 1 screen)
- Active source change

Funky Flounder

Features Developed

Stabilize

Patch Updates

New features(Add)

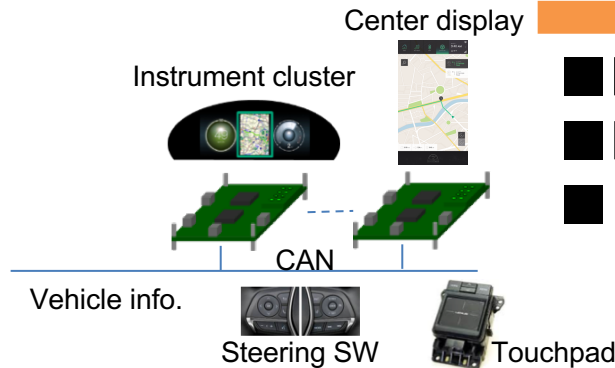
- Screen layout (2App: 1 screen)
- On-Screen

Grumpy Guppy

Features Developed

Stabilize

- Multi ECU/Display
- Policy Manager(Update)
- Input Manager



Assumed system

Conclusion

Conclusion

Proposed three requirements for HMI-Framework.

- ① OEM can choose GUI
 - ⇒ Enables compelling application development.
- ② OEM can choose HMI-Manager
 - ⇒ Enables flexible development for OEM needs.
- ③ HMI-Framework can arbitrate HMI resources
 - ⇒ Each OEM can easily develop each OEM's specification.

By our proposal, we hope that OEM will promote product development using AGL and that AGL will become more active.

Thank you once again for taking the time to join today's presentation.

toshikazu_ohiwa@mail.toyota.co.jp