Version 1.0

Automotive Grade Linux Requirements Specification

May 28, 2015

www.automotivelinux.org

www.linuxfoundation.org

LINUX FOUNDATION

May 28, 2015

Table of Contents

1 Automotive Grade Linux *5*
1.1 Overview 5
1.2 Document Scope 6
1.3 Glossary of Terms 6
2 Architecture *7*
3 App/HMI Layer *8*
3.1 Home Screen 9
3.1.1 Layout 9
3.1.2 System UI Parts 10
3.1.3 Application Management 10
3.1.4 Application Switch 10
3.1.5 Application History 10
3.1.6 Application Stack 12
3.1.7 Role of Home Screen 14
3.1.8 Requirements 16
4 Application Framework Layer *20*
4.1 AGL Application Framework 20
4.1.1 Application Manager 21
4.1.2 Window Manager 21
4.1.3 Policy Manager 27
4.1.4 Sound Manager 59
4.1.5 Input Manager 63
4.1.6 User Manager 65
4.2 Web HMI 71
4.2.1 Web API 71
4.2.2 Web Runtime 75
4.3 Native HMI 76
4.3.1 Native App Runtime 76
4.3.2 Native Application Framework 77
5 Services Layer *77*
5.1 Platform Services 78
5.1.1 Bluetooth 78
5.1.2 Error Management 92
5.1.3 Graphics 98
5.1.4 Location Services 98

5.1.5 Health Monitoring 100
5.1.6 IPC 100
5.1.7 Lifecycle Management 100
5.1.8 Network Services 100
5.1.9 Persistent Storage 100
5.1.10 Power Management 100
5.1.11 Resource Management 102
5.1.12 Telephony Services 102
5.1.13 Wi-Fi 103
5.1.14 Window System 110

5.2 Automotive Services 111
5.2.1 Audio Services 111
5.2.2 Camera Services 111
5.2.3 Configuration Services 111
5.2.4 Diagnostic Services 111
5.2.5 Multimedia Services 111
5.2.6 Navigation Services 116
5.2.7 PIM 117
5.2.8 Smartphone Link 117
5.2.9 Speech Services 125
5.2.10 Tuner Services 125
5.2.11 Vehicle Bus / Vehicle Info Control 132
5.2.12 Telematics Services 141
5.2.13 Window System 142

6 Security Services 143

6.1 Access Control 143
6.1.1 Requirements 144

7 Operating System Layer 144

7.1 Kernel 144
7.1.1 Linux Kernel 144

7.2 Boot Loader 145

7.3 Hypervisor 145
7.3.1 Requirements 145

7.4 Operating System 145
7.4.1 File Systems 145
7.4.2 Resource Control 150
7.4.3 Startup/Shutdown Control 153
7.4.4 Database 155
7.4.5 System Update 156

7.5 Device Drivers 157

7.5.1 Peripherals 157
7.5.2 Graphics Drivers 158
7.5.3 Video Drivers 158
7.5.4 Audio Codecs 158
7.5.5 Automotive Devices 158

8 Notices....................159

1 Automotive Grade Linux

1.1 Overview

Automotive Grade Linux (AGL) is a Linux Foundation Workgroup dedicated to creating open source software solutions for automotive applications. Although the initial target for AGL is In-Vehicle-Infotainment (IVI) systems, additional use cases such as instrument clusters and and telematics systems will eventually be supported. AGL has participants from the Automotive, Communications, and Semiconductor Industries and welcomes contributions from individual developers.

By leveraging the over $10B of investment made in the Linux kernel and other open source software projects, the AGL Workgroup:

- Enables rapid software innovation for automotive suppliers to keep up with the demand from consumers for better IVI experiences
- Utilizes the talents of thousands of open source software developers dedicated to maintaining the core software in areas like the Linux kernel, networking, and connectivity, used in systems across numerous industries

The goals of the Automotive Grade Linux Workgroup are to provide:

- An automotive-focused core Linux operating system stack that meets common and shared requirements of the automotive ecosystem with a broad community of support that includes individual developers, academic organizations and companies.
- A transparent, collaborative, and open environment for Automotive OEMs, Tier One suppliers, and their semiconductor and software vendors to create amazing in-vehicle software.
- A collective voice for working with other open source projects and developing new open source solutions.
- An embedded Linux distribution that enables rapid prototyping for developers new to Linux or teams with prior open source experience

This results in faster time to market by jump-starting product teams with reference applications running on multiple hardware platforms.

1.2 Document Scope

The scope of this document is to define the architecture of the Automotive Grade Linux software platform. The requirements are broken up into an overview of the Architecture and a description of each of the layers in the architecture followed by the requirements for each module in the various layers. The Architecture Diagram and the layout of the specification take into consideration all of the components that would be needed for an IVI system; however the are missing requirements for individual modules. As the spec continues to evolve those sections will continue to be filled in.

The main goal of this document is to define the core software platform from which applications can be built. As such, this document does not define application requirements except in a single case (Home Screen). Application requirements will be developed by various projects that use the AGL platform. Those application requirements can be used to drive new or revised requirements into the platform.

At this time there is no plan to use this specification to create a compliance or certification program. The specification is used as blueprint to guide the overall work of AGL and to derive work packages for companies and individuals to complete in order to attain the goals of the AGL Workgroup.

1.3 Glossary of Terms

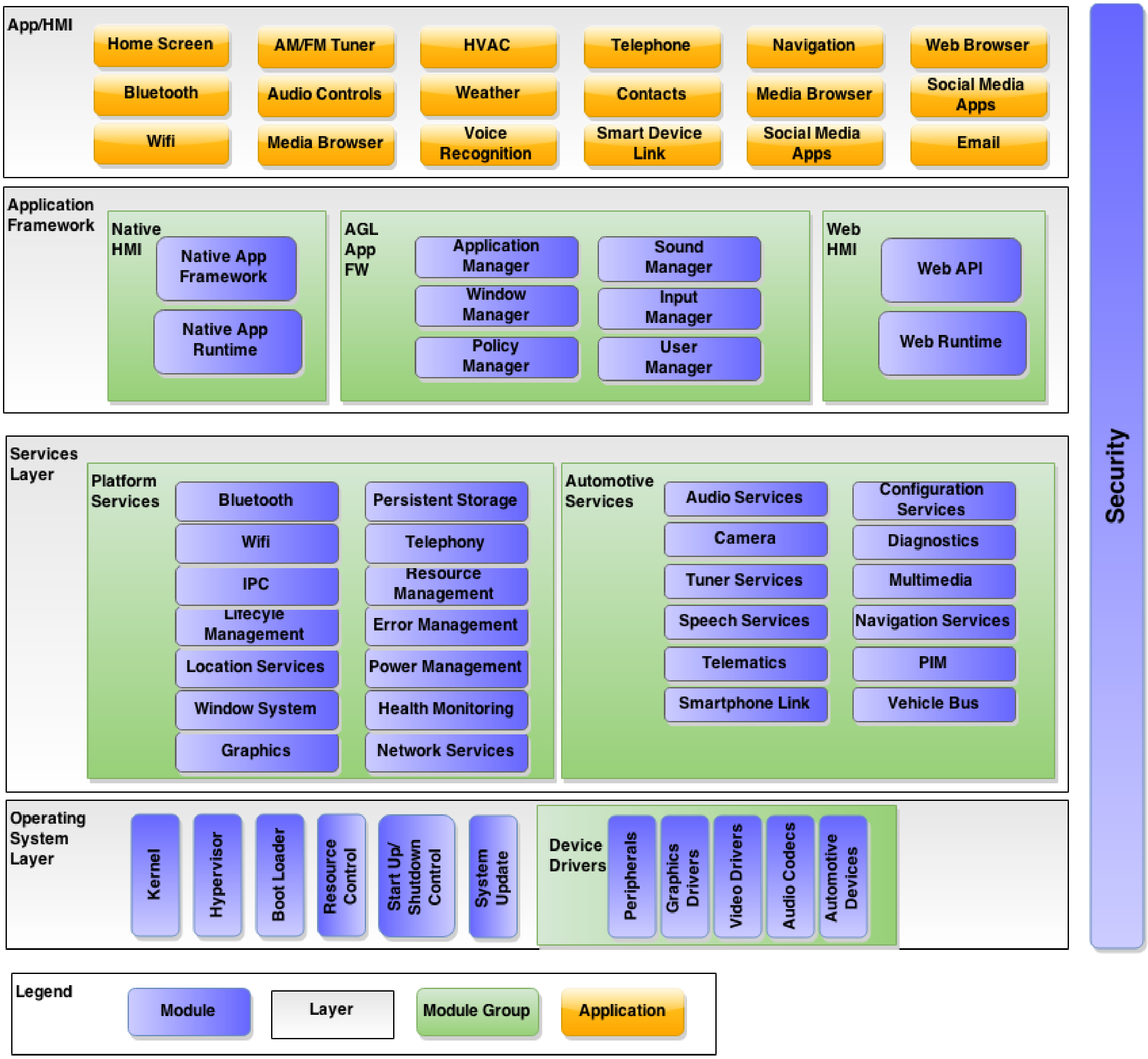
Term	Definition
A2DP	Advanced Audio Distribution Profile
AGL	Automotive Grade Linux
AVRCP	Audio Video Remote Control Profile
FS	File System
GPS	Global Positioning System
GPU	Graphical Processing Unit

HFP	Hands Free Profile
IBOC	In-Band On Channel
LTSI	Long Term Support Initiative
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OS	Operating System
OSS	Open Source Software
SDL	Smart Device Link
STT	Speech to Text
TTS	Text to Speech

2 Architecture

The Automotive Grade Linux Software Architecture diagram is below. The architecture consists of five layers. The App/HMI layer contains applications with their associated business logic and HMI. Generally applications are out of scope for this document since they are product specific for the OEM that is developing a system based on AGL.

The Application Framework layer provides the APIs for creating both managing and running applications on an AGL system. The Services layer contains user space services that all applications can access. The Operating System (OS) layer provides the Linux kernel and device drivers along with standard OS utilities.



3 App/HMI Layer

Applications may use a web based framework or a native framework. A system may include applications that use different frameworks. Coordination of applications between frameworks is performed by the AGL App Framework. The diagram represents possible applications that could appear in a given system, but is not all inclusive. Reference applications may be provided by AGL to demonstrate the capabilities of the platform.

3.1 Home Screen

Home Screen provides the Home User Interface (Home UI) of the system which meets the following requirements:

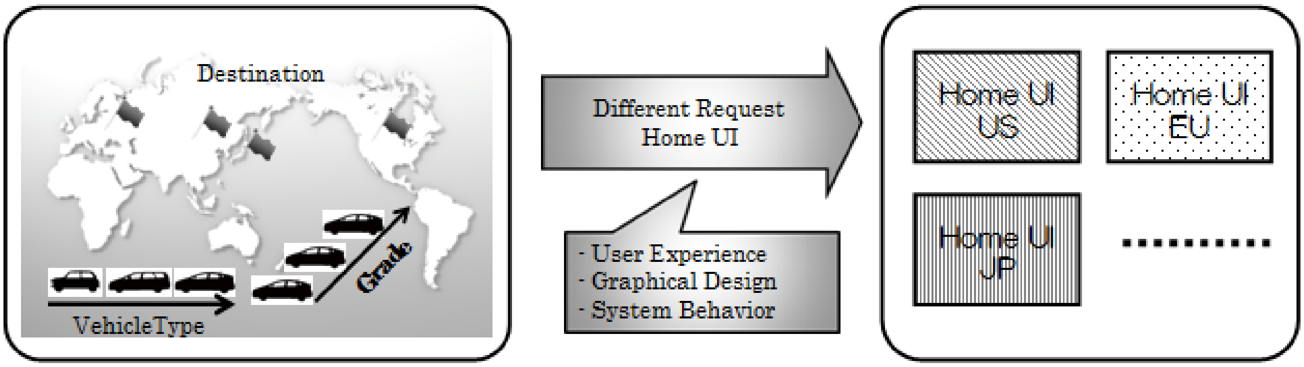
- Rich User Experience (Rich UX)
- Driver Distraction mitigation
- Variations support

Rich UX covers requirements such as usability and user satisfaction. Driver Distraction mitigation covers requirements on display control and user operation behavior while vehicle is in motion to minimize driver distraction. Variations support covers requirements to support customization of design and behavior of the system to meet the different needs of vehicle type, destination and grade.

3.1.1 Layout

The following use cases are considered for Layout.

- Home Screen developer changes the Home UI by using a customizable layout definition.



3.1.2 System UI Parts

The use case assumed about System UI Parts is as follows.

- An application or System uses status bar and on-screen in order to notify information to a user.
- User uses the system setting UI in order to change settings.
- User uses software keyboard in order to input characters.

3.1.3 Application Management

The use case assumed about Application Management is as follows.

- A user downloads and installs or updates the delivery application from application store.
- A user uninstalls the delivery application.
- A user launches the installed delivery application or the pre-installed application.
- Also a user terminates those applications.

3.1.4 Application Switch

The use case assumed about Application Switch is as follows.

- User switches application via application history or application stack.
- The system switches application according to Driving Mode status.

3.1.5 Application History

Application switching by application history is assumed as follows.

- The system records the order of the applications in the order in which the application is displayed.
- The order of application that is recorded is updated each time the display of the application is switched.
- Screen of the application is displayed in the order in which they are recorded in the history at the time of switching applications.

□ Specification of operation

- User runs a swipe from the edge of the application screen area.

□ Specification of action

- The order of the screen is managed order management list (application history).

- List order update opportunity(Update has determined a display of the application)

- Application starts or stops.

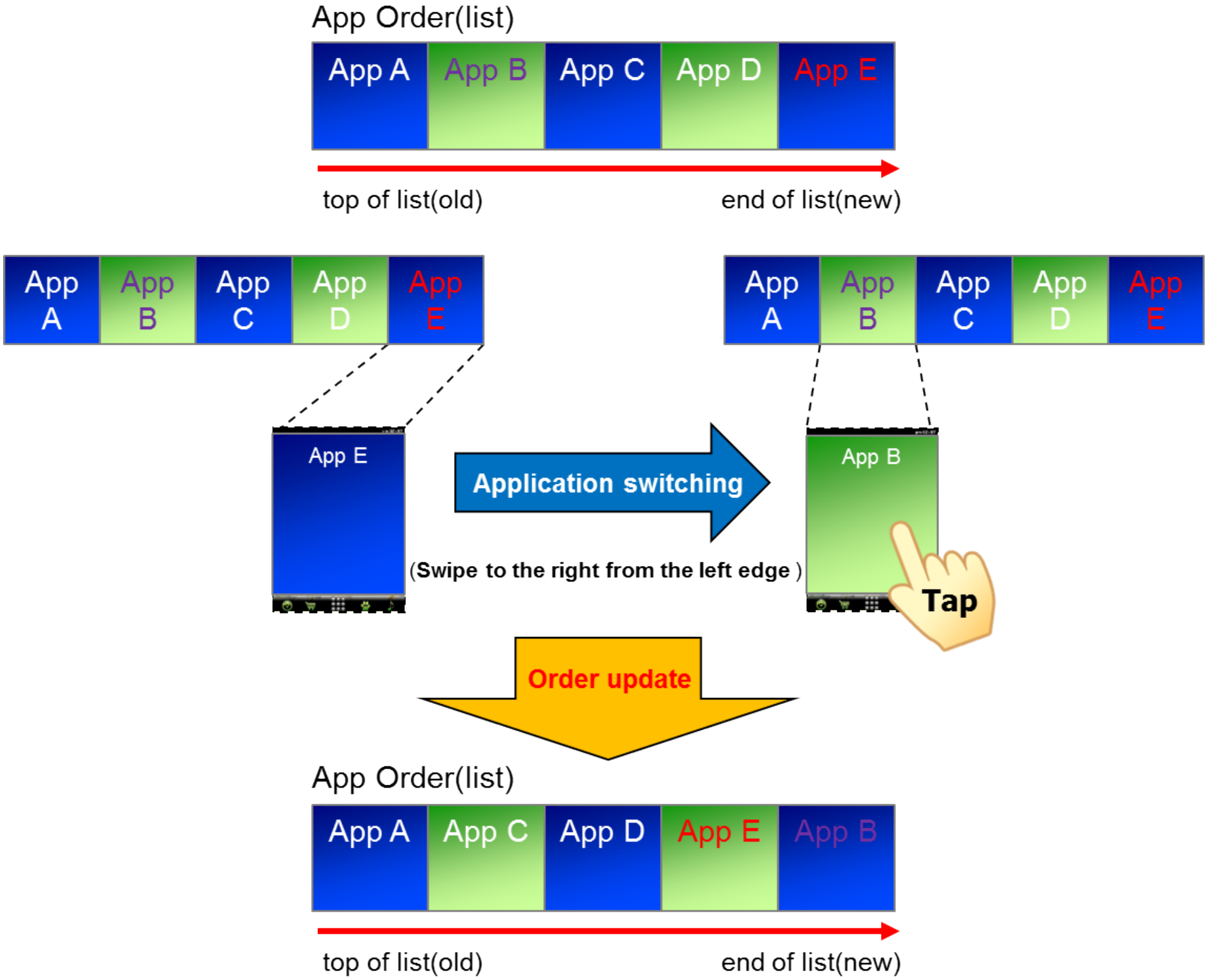
- Allowed to stand between the screen N seconds after the swipe.

□"N seconds"□User defines the value of any.

- User to operate the screen after you swipe.

□"operation"□Screen tap. Menu display. Other.

Figure 5□2 represents a sample Home Screen depicting the above mentioned use cases.



3.1.6 Application Stack

Application switching by application stack is assumed as follows.

- The user specifies the type of any order. The system records the order of the application to the rule as of the specified type.
- Examples of the types of any order
- Application start-up order
- Screen of the application is displayed in the order in which they are recorded in the stack

when switching applications.

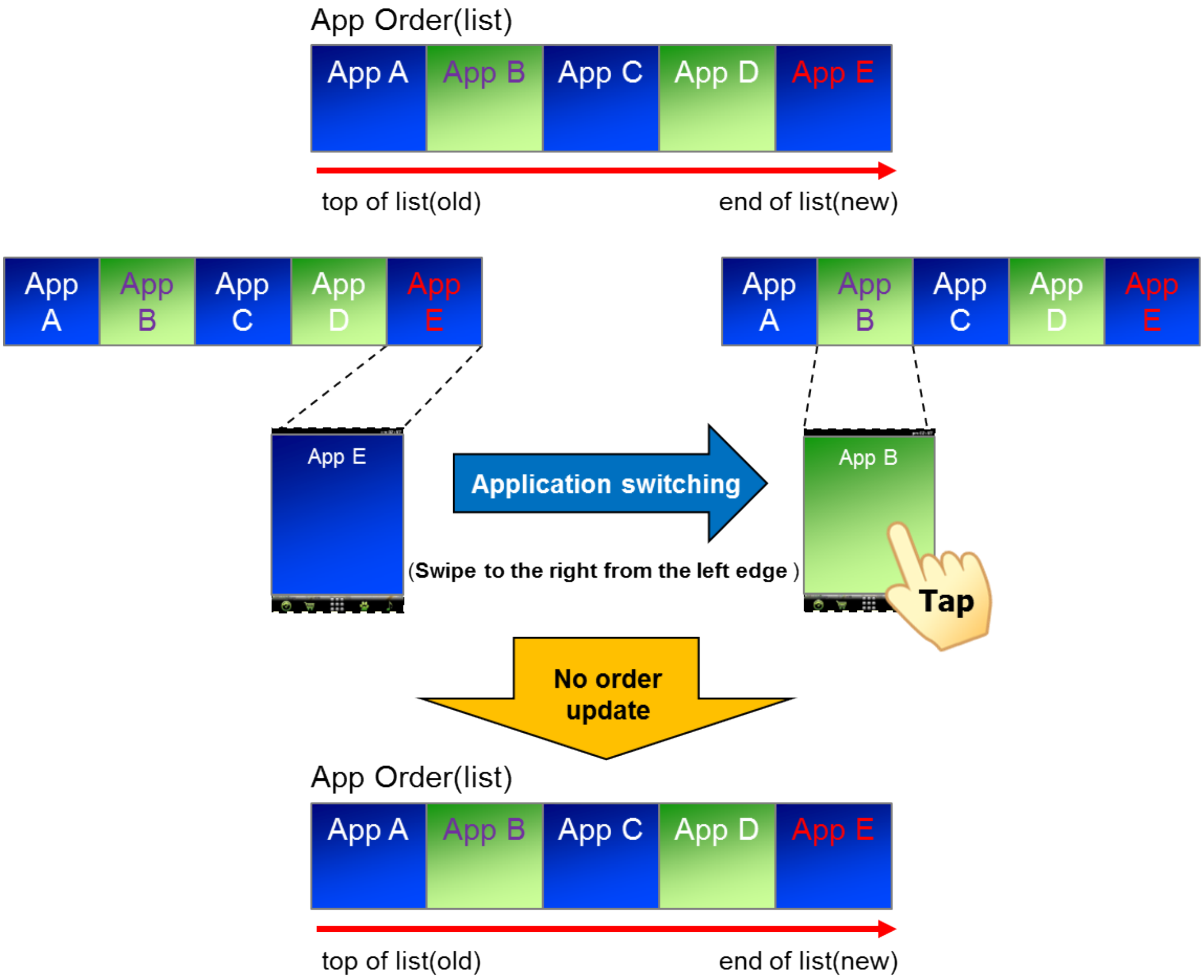
□ Specification of operation

- User runs a swipe from the edge of the application screen area.

□ Specification of action

- The order of the screen is managed order management list (application stack).
- List order update opportunity.(Application start-up order as an example)
- Application that started at the end of the list when the application is started is added.
- Application that has stopped from the list when the application is stopped will be deleted.

Figure 5-3 represents the switching example depicting the application of the above switching.



3.1.7 Role of Home Screen

Table 5-1 describes the role of the Home Screen to satisfy the purpose and use cases mentioned above.

No	Use Case	Role	Description
1-1	Layout	GUI Layout definition	Function to define a customizable GUI Layout definition.

1-2		Change Layout	Function to apply the customized GUI layout definition.
2-1	System UI Parts	Status Bar	Function to display the information from application or system. Function to quickly access and set certain system settings.
2-2		On-screen	Function to display a popup window such as alert messages.
2-3		System Setting	Function to display system settings menu regarding GUI, such as locale and network.
2-4		Software Keyboard	Function to display software keyboard.
3-1	Application Management	Application Management	Function to download applications from application store. Function to install, uninstall and update the downloaded applications.
3-2		Application Launcher	Function to launch/terminate applications.
4-1	Application Switch	Application List	Function to switch applications by installed application list.
4-2		Application History	Function which switches application in order by applications history.
4-3		Application Stack	Function to switch application in any order.

Table 5-2: Relevance of the Role and Purpose

No.	Role	Rich UX	Driver Distraction mitigation	Variations support
1-1	GUI Layout definition	□	□	□
1-2	Change Layout	□	□	□
2-1	Status Bar	□		□
2-2	On-screen	□		□
2-3	System Setting	□		□
2-4	Software Keyboard	□		□
3-1	Application Management	□	□	
3-2	Application Launcher	□	□	
4-1	Application List	□	□	
4-2	Application History	□	□	
4-3	Application Stack	□	□	

3.1.8 Requirements

3.1.8.1 Layout

Home Screen must provide a mechanism for customizable GUI layout definition by each vehicle type, each destination and each grade.

Home Screen must provide a mechanism for a customizable GUI layout definition for different vehicle type, destination and grade.

GUI layout definitioncan be definedsuch as the following items:

(In addition, items that can be defined is not limited to the following.)

- screen resource (Display, Layer Type, Area)

- sound resource (Zone, Sound Type)
- input resource (Device, Event Type)
- UI Component to be used in the entire system
- transition effect (Animation effect)
- Background image

Home Screen must provide a mechanism to apply customized GUI layout definition.

3.1.8.2 System UI Parts

Home Screen must provide a mechanism to display two or more information simultaneously to the status notification area.

Home Screen must provide a mechanism to displaying status to status notification area.

- Current Time: Displaying clock capability
- Icons of Status: Displaying icons for notify information from applications
- Status Message: Displaying text for notify information from applications
- Communication Status: Status of mobile communication and wireless communications (Wi-Fi, Bluetooth, etc.)

Home screen must provide an interface to retrieve information from application for notification.

Home Screen must provide a mechanism to show popup window into on-screen window.

Home Screen must provide GUI method to hide on-screen window by user operation.

Home Screen must provide a mechanism to hide on-screen window within a specified duration.

Home Screen must provide an interface for applications to request to show popups.

Home Screen must provide an interface for applications to cancel the previously requested popup.

Home Screen must provide a mechanism to show text information, draw images and show software switch like button in the on-screen window.

Home Screen must provide a mechanism to specify attributes such as position and size of On-screen window.

Home Screen must support a mechanism to specify other window display effect when the On-screen window is displayed. (e.g. tone down)

Home Screen must provide system setting menu regarding GUI, such as locale and network.

Home Screen must provide a mechanism to change current date and time setting.

Home Screen must provide a mechanism to change timezone setting.

- The platform must set up the date, time and timezone according to a current position automatically.
- Home Screen must provide a mechanism to set up turning on and off of the automatic date/time/timezone setup.

Home Screen must provide a mechanism to change language setting.

Home Screen must provide a mechanism to change wireless communications (Wi-Fi, Bluetooth, etc.) setting.

- Enable/Disable
- Connect/Disconnect
- Search the devices
- Display the list of available and/or registered devices

Home Screen must provide a mechanism to change mobile communication setting.

- Enable/Disable

- A setup and change of various attributes
- Display the list of registered devices and select device

HomeScreen must support to change the appearance of a screen to a user's liking.

These are as follows.

- Tone of a screen.
- Appearance of a window frame.
- Animation effect when screen transition was occurred.

Home Screen must support a mechanism to set or change master audio volume.

Home Screen must support a mechanism to set or change display brightness.

Home Screen must provide a mechanism to show software keyboard.

Home Screen must provide a mechanism to apply default settings (e.g. theme, local, wallpaper) to a new user, when a user is added by the User Manager.

3.1.8.3 Application Management

Home Screen must provide a mechanism to manage downloaded application package.

- Display downloaded application list from application store.
- Download the application
- Install the downloaded application
- Uninstall the downloaded application
- Update the downloaded application

Home Screen must provide a mechanism to launch the application.

Home Screen must provide a mechanism to terminate the application.

3.1.8.4 Application Switch

Home Screen must provide a mechanism to show the list of installed applications.

Examples of assumed application list

- list of application name
- list of application's icon
- list of live thumbnail for all the running applications

Home Screen must provide a mechanism for switching display application in order by application history.

Home Screen must provide a mechanism for the application stack in any order. For example, such as launch order or display order.

Home Screen must provide a mechanism for the system to switch applications.

For example, when Driving Mode changes, system must be able to switch application based on policy.

4 Application Framework Layer

The Application Framework layer provides the methods needed to create software applications and their user interfaces. The platform can support multiple application frameworks any of which may be built into an SDK or product build. The application framework contains any code specifically written for that framework as well the bindings to the Services and Operating Systems layers that the application framework provides for its applications.

4.1 AGL Application Framework

The AGL Application Framework provides basic services to all applications regardless of the framework they are implemented in so that there is a standard method providing the services.

4.1.1 Application Manager

Application Manager describes requirements for AGL application lifecycle function. Application lifecycle contains application installation/removal and launch/hide/resume/kill.

4.1.1.1 Requirements

AGL System must support application lifecycle (install/uninstall, launch/kill, suspend/resume) based on appid/pid via launcher.

AGL System must support a database to store application metadata (appid, exec path etc.).

AGL System must provide an interface to get a list of installed applications.

AGL System must provide an interface to get the state of an application.

AGL System must provide application privilege control.

4.1.2 Window Manager

A window system is a software component that facilitates an implementation of graphical user interface. A window system is responsible for managing display devices, Graphics Processing Units (GPUs), input devices, and graphics memory. A window system works with the software component named window manager that is responsible for a layout management of windows, and a routing of user interactions.

A window manager is as software component that is responsible for a layout management of windows.

Window manager of automotive middleware layer makes up for traditional window management system to be satisfied IVI’s complex requirements, typically requested from Policy Manager. Also, AGL aims to provide well-portability among various hardware platforms.

4.1.2.1 Use Case

Please refer "screen resource control" of Policy Manger section.

4.1.2.2 Role

Table 7-148 describes the role of window manager to be satisfied above purpose and use cases.

Table 7-16 : Role of Resource Control

No.	Role	Description
1	Window drawing	Provide capability to draw a window to any place and any size and any scale. Also provide capability to change visibility of the window.
2	Overlay of multiple windows	Provide capability to overlay two or more windows with any z-order. Also provide capability to use hardware layer efficiently.
3	Visual effect	Provide capability to adapt visual effect as below. · Animation effect to change visibility · Animation effect to transit between two or more windows · Visual effect for a window, such as gray-out and transparent.
4	Frame rate control	Provide capability to control dynamic frame rate change. This is useful if system resource was shortage.
5	Multiple hardware layer support	Provide capability to use hardware layer efficiently if hardware supports two or more hardware layers.

6	Reduced dependency of hardware	Provide well-defined interface to reduce dependency of hardware. Well-defined interface also makes it possible to increase the effect of portability and development cost.
7	Multi window / multi display	Support multi window management and multi display.
8	Compatibility	From the compatibility point of view, AGL should use public API, and shall not rely on hardware specific API.

4.1.2.3 Requirements

4.1.2.3.1 Window Drawing

System must provide a mechanism to manage surfaces, such as create, delete, make visible and make invisible.

System must provide a mechanism to create and delete surface.

When surface is created or deleted, system must notify status change to GUI resource.

This notification mechanism makes possible to assign surface to proper area by GUI resource.

System must provide a mechanism to change visibility per each surface.

And, provide an interface to change visibility.

All the surfaces must be set to invisible for initial state.

Surface will be visible only if GUI resource issues to change visibility.

System must provide a mechanism to move surface’s area. If area size was different between previous area and new one, then system must support to fit into new area by VIC.4.1.4.

System must provide a mechanism to fit surface into area. Because, size of area may different from size of surface.

If resize was happened, system must notify to surface's owner application.

If size of surface and size of area was different, system must provide a mechanism to fit surface into area by squeeze.

If size of surface and size of area was different, system must provide a mechanism to fit surface into area by using combination of scaling and trimming function.

That means, system must provide a mechanism to fit surface into area keeping original aspect ratio. This makes it possible to fit by "pan & scan".

If size of surface and size of area was different, system must provide a mechanism to fit surface into area by using combination of scaling and background color.

That means, system must provide a mechanism to fit surface into area keeping original aspect ratio. System also provides a mechanism to fill background color into redundant pixels. This mechanism makes it possible to do "letterbox" method.

4.1.2.3.2 Overlay of Multiple Windows

System must provide a mechanism to create and delete a layer.

Layer must have a concept of z-order. That means, display order for each layer is decided by their z-order attribute.

Z-order attribute is fixed value. So, if application wants to change display order of surfaces, then, attached layer must be changed.

System must provide a mechanism to create and delete "area" to display surface.

Area is a concept which defines where to display in specific layer.

System must provide a mechanism to attach surface to any layer.

Also, system must be able to change attached layer.

And, provide an interface to attach and change.

System must provide a mechanism to assign surface to any area in a layer.

And, provide an interface to assign surface to any area.

System must provide a mechanism to change visibility per each layer.

That means all the surfaces belonging to same layer will be changed visible or invisible at the same time.

And, provide an interface to change visibility per layer.

Initial state must be set to invisible.

System must provide a mechanism to enable superimposed display based on z-order of each layer, and disposition of surfaces.

4.1.2.3.3 Visual Affect

System must provide a mechanism to apply animation effect when visibility change was happened.

Per each animation, system must provide a mechanism to apply below attributes.

- Duration

Animation type

System must provide typical animation effects, such as slide-in, slide-out, zoom-in and zoom-out.

Also, system must provide a mechanism to add, delete and change animation effect easily by plug-in architecture.

System must provide a mechanism to apply animation effect when move surface was happened.

Per each animation, system must provide a mechanism to apply below attributes.

- Duration

Animation type

System must provide typical animation effects, such as slide-in, slide-out, zoom-in and zoom-out.

Also, system must provide a mechanism to add, delete and change animation effect easily by plug-in architecture.

System must provide a mechanism to make effect to surface.

And, provide an interface to set effect type from application and other software components.

System must provide a mechanism to make specific surface to gray-out.

System must provide a mechanism to make specific surface to low brightness

System must provide a mechanism to add, delete and change effect for surface easily by plug-in architecture.

4.1.2.3.4 Frame Rate Control

System must provide a mechanism to reduce frame rate independent from refresh interval of application.

System also provides a mechanism to set frame rate as 0fps, independent from refresh interval of application.

This function is useful to keep whole system quality even if high load status, such as live thumbnail and moving surface.

4.1.2.3.5 Multiple Hardware Layer Support

If hardware supports two or more hardware layers, system must provide a mechanism to use hardware layers efficiently.

- Never use software overlay when superimposing two or more hardware layers

Assign hardware layer for graphical high load function, such as video playback

4.1.2.3.6 Reduced Dependency of Hardware

Window Manager must be able to retrieve system structure regarding displays and layers of each display. And system must provide a mechanism to adapt any structure without re-build, such as by using re-configuration.

4.1.2.3.7 Multi Window

AGL specifies that automotive grade Linux shall manage multiple windows owned by multiple processes on a display.

AGL specifies that automotive grade Linux shall support multi-headed display.

4.1.2.3.8 Compatibility

AGL specifies that automotive grade Linux shall have a window manager that uses only public APIs provided by Window System and OpenGL/ES 2.0 for rendering and user interaction.

AGL specifies that automotive grade Linux shall have a window manager that relies on a standard rendering API such as OpenGL/ES 2.0 only. The window manager shall not rely on any hardware specific API.

A window system and OpenGL/ES 2.0 API are responsible for a hardware abstraction.

4.1.3 Policy Manager

4.1.3.1 Overview

4.1.3.1.1 Purpose

Policy Manager collects information and makes decisions based on them. To do that, Policy Manager collects lots of status, such as user operation and application status, then issue Vehicle Info Control or Resource Control to provide information. Policy Manager controls two types of resource, one is called "GUI resources" such as screen and sound, and other one is called

"System resources" such as CPU and memory.

4.1.3.1.2 GUI Resources

(1) Definition

· About Control of GUI Resources

AGL is supposed the following devices in this feature. For example, display with touch panel, speaker, and microphone. And AGL defines that "GUI resources" are resources that provide user or is provided by user on those devices, such as windows, sound streams and input events.

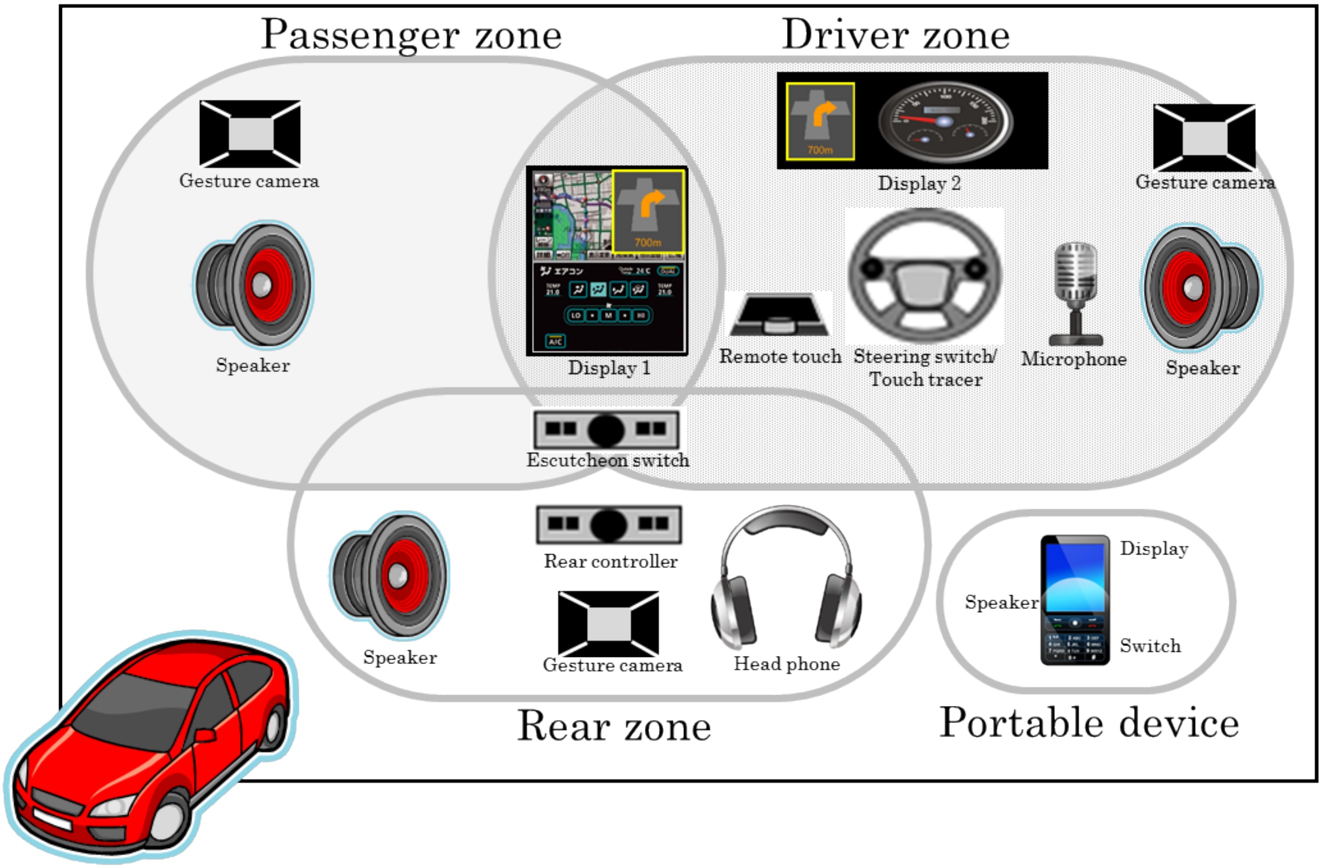


Figure 7-1: GUI resources

Policy Manager controls GUI resources according to external conditions. For example, Policy Manager limits the information of GUI resources while the vehicle is driving, because, the too much information distracts the attention of driver from driving operations.

- Associated Software Architecture

The software architecture of Policy Manager and related components regarding GUI resources control is as below.

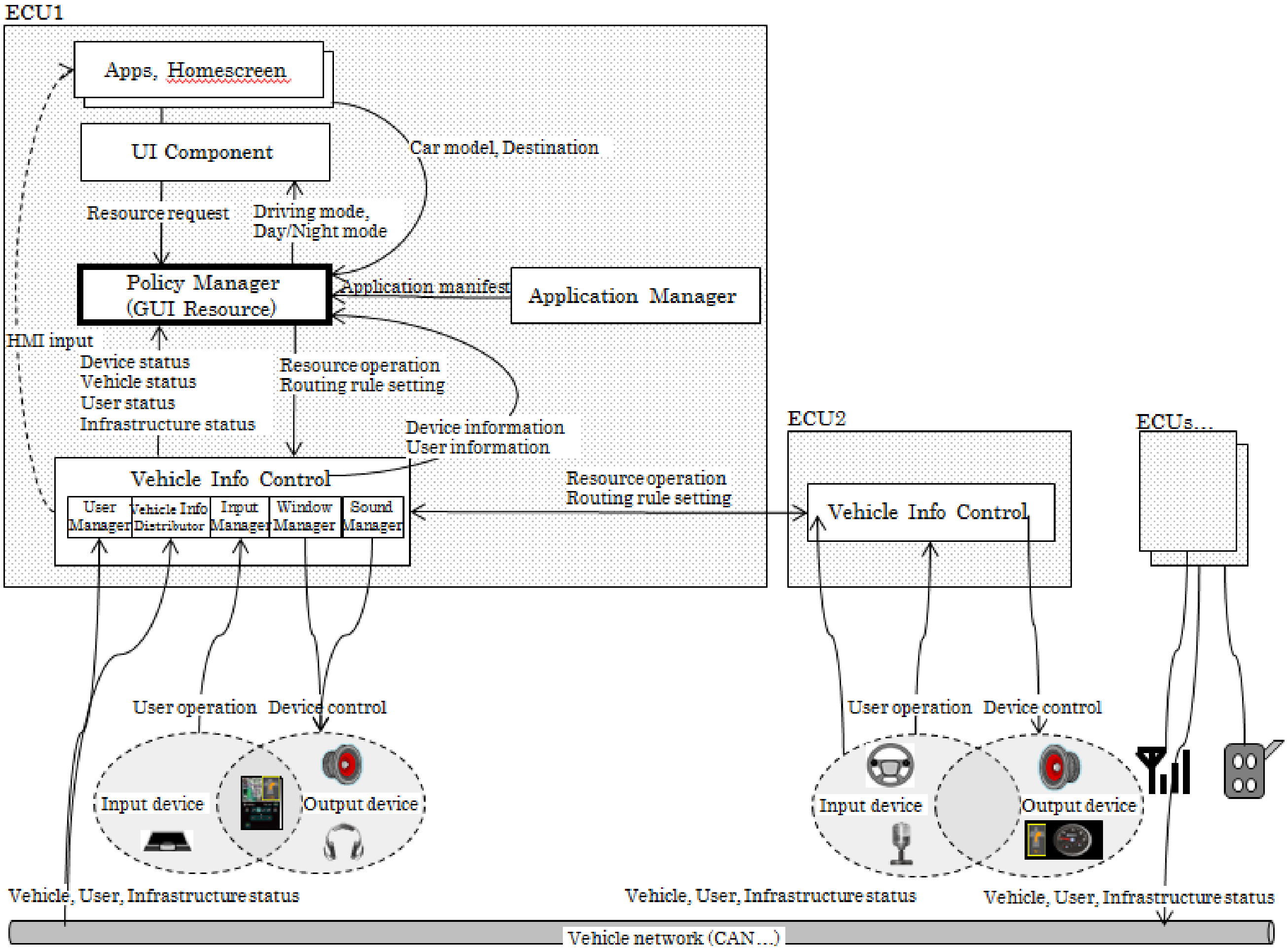


Figure 7-2: Associated Software Expected Use Case

Policy Manager is related with the below components.

Table 7-1: Related Components

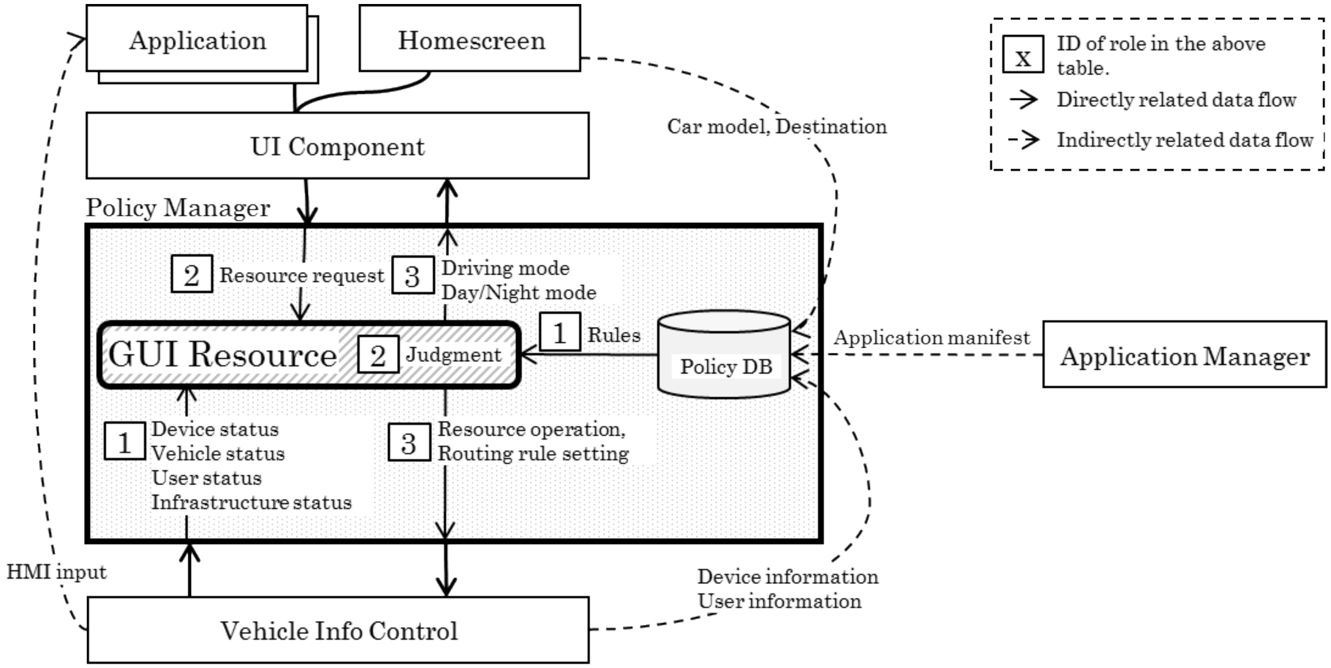
No.	Component		Description
1	Homescreen		Request to control of GUI resources.
2	Applications		Request to output or input of GUI resources.
3	UI Component		Receive driving mode and day night mode. And then provide the corresponding feature to applications UI such as input limitation and changing the theme.
4	Application Manager		Detect application installation. Then Notify the definition of GUI resources such as role by application configurations.
5-1	Vehicle Info Control	Window Manager	Control screen resource such as show/hide windows.
5-2		Sound Manager	Control sound resource such as mute/unmute sound streams.
5-3		Input Manager	Control input resource such as notify/not notify touch event on touch panel display to applications.
5-4		Vehicle Info Distributor	Provide the vehicle information from vehicle network such as CAN.
5-5		User Manager	Detect user switching. Then Notify the definition of user information such as application list of login user.

(2) Role

Policy Manager has the below role.

Table 7-2: Role of Policy Manager

ID	Role	Description
1	External condition collection	(1) Receives the external conditions.
2	Judgment of priority of GUI resource	(1) Receives the input/output/control request of GUI resources. (2) Judgment the GUI resource owner according to external conditions.
3	GUI resource control	(1) Issue the GUI resource control according to judgment. (2) Notify the driving mode and day night mode that is calculated by external conditions.


Application
Homescreen
UI Component
Car model, Destination
x ID of role in the above table.
Directly related data flow
Indirectly related data flow
Policy Manager
2 Resource request
3 Driving mode Day/Night mode
GUI Resource
2 Judgment
1 Rules
Policy DB
Application manifest
Application Manager
1 Device status Vehicle status User status Infrastructure status
3 Resource operation, Routing rule setting
HMI input
Device information User information
Vehicle Info Control

* Homescreen also contains common features just like other applications.

Figure 7-3: Definition of Role

GUI resource classifies screen resource, sound resource and input resource. Details of each resource type are as follows:

a. Screen Resource

a-1. External Condition Collection

Policy Manager collects the below definition that is related with screen resource.

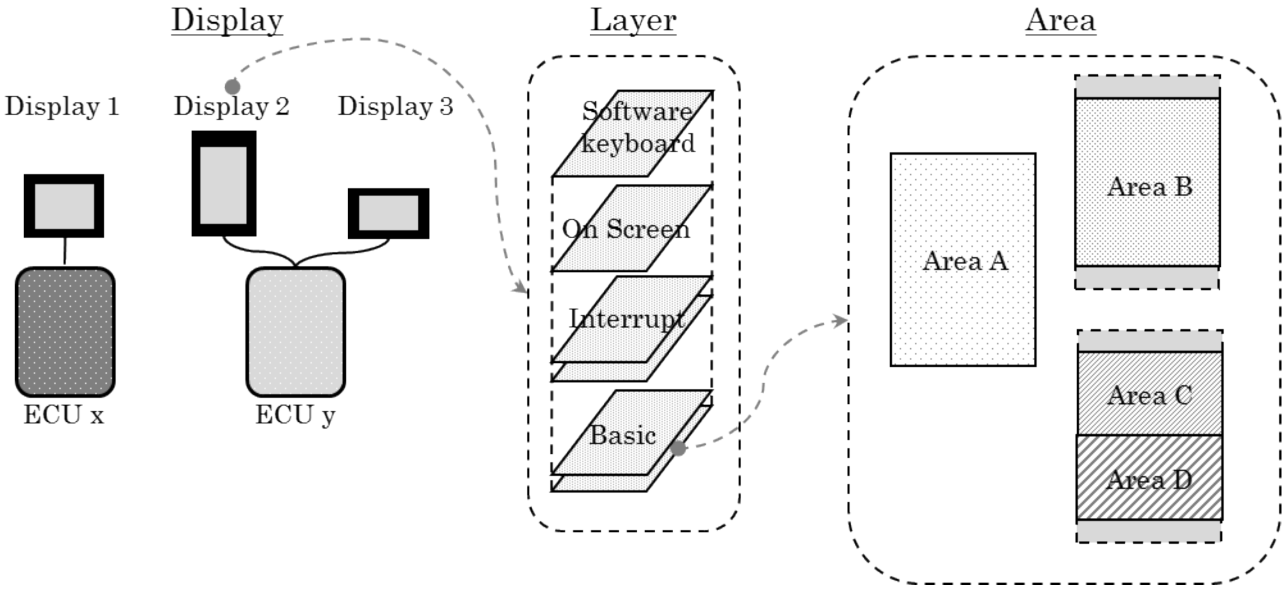


Figure 7-4: Definition of screen resource

- Concept of Display, Layer, Layout and Area

AGL supports not only one physical display but also two or more displays. Each display has one or more layer. And each layer must be connected to one layout defined by Homescreen. Layout consists of one or more areas. "Area" is graphics composed area to display specific graphics window.

The z-order of layers is flexible. Policy Manager decides the z-order of each layer depending on objectives of them. For example, layer-1 was used as "phone call notification", and layer-2 was used as displaying "map", then Policy Manager will decide that layer-1 should be upper than layer-2.

Layer is created by application including Homescreen. When application creates layer, application specifies layer type. Layer type is roughly categorized as "Basic" and "Interrupt". "Basic" layers are used to display application itself such as media playback, map drawing and setting menu. "Interrupt" layers are used to display overlay windows such as information alert and enlarged view.

When application creates layer with "Basic" type, application must specify layout type for it. On the other hand, the case layer with "Interrupt", application must specify corresponding "Basic" layer. The layout of "Interrupt" layer is followed by "Basic" layer's layout.

From the capability of Policy Manager point of view, the main purpose of layer is to decide z-order. In other words, if there is a scenario to change z-order of two or more windows triggered by system status change and/or user operation, then such kind of window must assign to individual layer.

- Concept of Layer Owner, Role and Surface

"Layer owner" is application which created that layer. "Layer owner" can request each area of that layer. When "Layer owner" requests specific area, "Layer owner" also specify "Role" of area. "Role" represents how to be used that area, and used to define z-order of layers by Policy Manager.

"Layer owner" also can request to change "Role" for specific area, however, whether "Role" change is acceptable or not is decided by Policy Manager by using policy rule.

One area should connect to one graphics window. AGL defines the term "Surface" as graphics window to display into one area.

Surface is a canvas to draw graphical image by application. To show via physical display, surface drawn by application must be assigned to specific area. Figure 7-16 describes simplest example to assign one surface to full screen with one layer. If layer has two or more areas, then corresponding surfaces are mapped to each area. According to example of Figure 7-16, surface is fit to area size as "squeeze", however AGL also provide a way to fit as "letterbox" and "pan & scan".

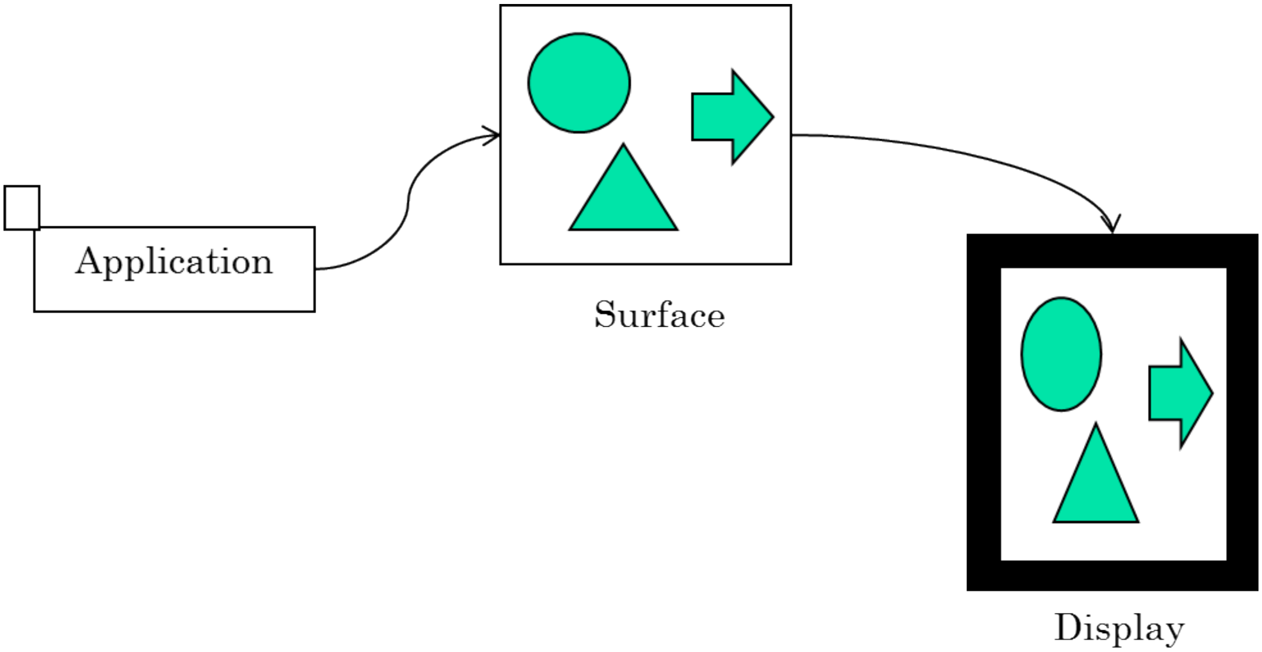


Figure 7-5: Definition of Surface

- Subdivision of "Interrupt" Layer

Basically, "Basic" layer corresponding to "Interrupt" layer is used to display application's main surface. However there are some exceptions. For example virtual keyboard is not needed main surface. However, to follow this layer type rule, virtual keyboard must have corresponding "Basic" layer. But this "Basic" layer never used to display. Also on-screen, such as alert message is not needed main surface too. But it must have corresponding "Basic" layer from same reason.

According to above concept and some exceptions, AGL defines four layer types described as Table 7-3.

Table 7-3: Definition of Layer Type

No	Type	Summary	Example
1	Basic	This is application's basic screen. Typically, application requests this layer at first time.	Map of navigation
2	Interrupt	This is application's popup screen.	Enlarged view of navigation
3	On-screen	This is system popup screen. Typically, On-screen service (e.g. Homescreen) requests this layer.	Warning message popup
4	Software keyboard	This is the software keyboard screen. Typically, software keyboard service requests this layer.	Software keyboard

a-2. Judgment of Priority of GUI Resource

Policy Manager receives the request with "Role" that is related with each screen resource. Role is the category name of screen resource priority. Role is used to judgment of priority by Policy Manager. Table 7-4 and Figure 7-6 describes the definition of role and sub role.

Table 7-4: Definition of Role

No	Contents	Summary	Example
1	Role	This is screen owner (such as application or service) role.	Navigation
2	Sub role	This is specific screen role.	Enlarged view

Role consists of role and sub role. Role is screen owner role such as "Navigation" and "Software keyboard”. Sub role defines when layer type of the screen resource is not “Basic”. Sub role is popup screen role such as “Enlarged view” (of Navigation).

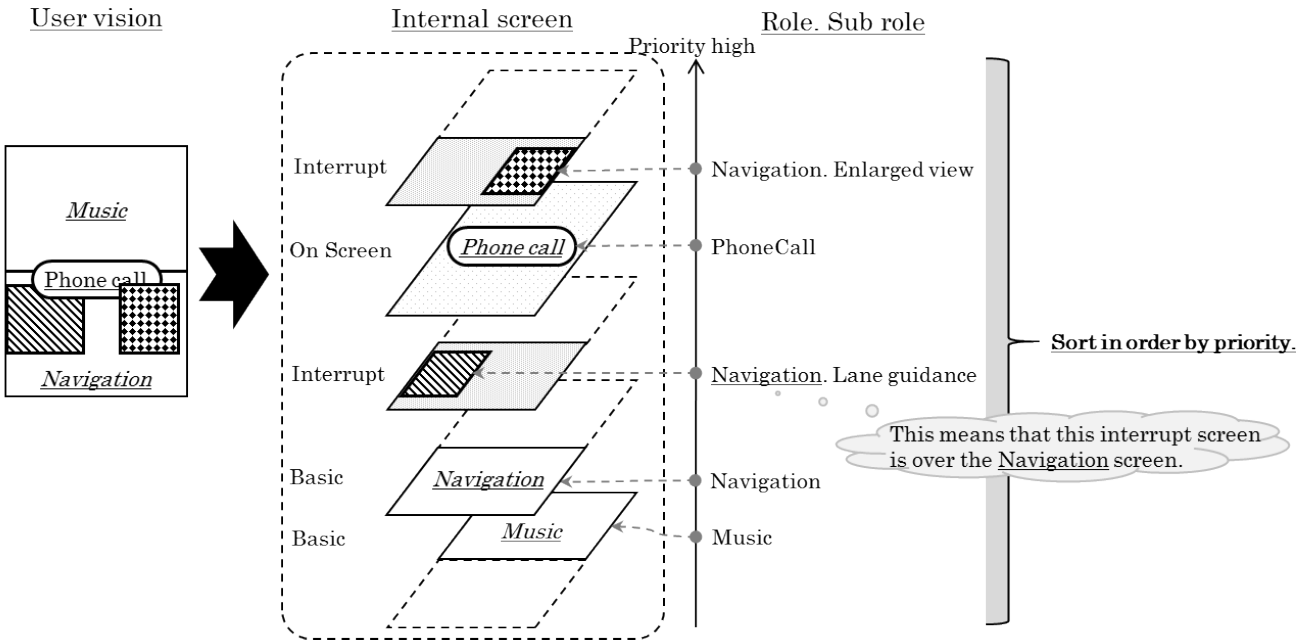


Figure 7-6: Definition of Role and Sub role

The screen resources are sorted of priority that is related to role by Policy Manager. If display has two or more layers, then all layers will be superimposed by z-order.

In addition, Policy Manager decides the area of "Interrupt" layer using role. Area of "Interrupt" layer must be same area of the related "Basic" layer. "related" means that "Role" (is not "Sub role") of "Basic" and "Interrupt" is same. For examples, if "Interrupt" layer is set “Navigation” role and “Lane guidance” sub role, this is set in same area of "Navigation" role.

a-3. GUI resource control

Policy Manager controls the screen resources using Vehicle Info Control. Policy Manager only issues to control the screen resources but it is actually controlled by Vehicle Info Control directly.

There are three types of screen resource control:

One is allocation of each surface such as position, size and size-fitting method.

Second one is visibility control. Basically, visibility should be "ON" during area owner was assigned. However, visibility may set to "OFF" during driving mode due to driving restriction.

Last one is order control of each layer. Policy Manager decides the order of each layer, and issue z-order information for each layer.

b. Sound Resource

b-1. External Condition Collection

Policy Manager receives the below definition that is related with sound resource.

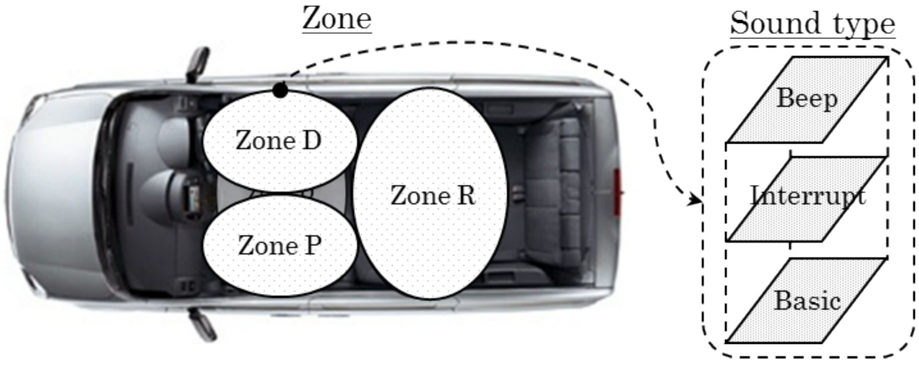


Figure 7-7: Definition of Sound Resource

- Zone

Zone is a place in the car, such as driver zone, passenger zone, rear seat zone. Each zone can play at the same time.

- Sound type

Sound type is the category of sound resource. Sound type must be set by each sound resource owner such as application. If application wants to play sound, it must be assigned to proper sound type of proper zone. Only one sound stream can occupy specific sound type of specific zone. In other words, if two or more sound streams should be mixed in same zone, then each sound stream must assign to individual sound type.

AGL supports the following sound type, however it's just sample and should be configurable.

Table 7-5: Definition of sound type

No	Type	Summary	Example
1	Basic	This is application's basic sound.	Music of media player
2	Interrupt	This is application's interrupt sound.	Guidance of Navigation
3	Beep	This is beep. Typically, Homescreen requests this type.	Display touch sound

- Stream

Stream is connection of sound resource that is made in applications. Sound is transferred in stream.

b-2. Judgment of Priority of GUI resource

Policy Manager receives the request with "Role" that is related with each sound resource. Role is the category name of sound resource. Role is used to judgment of priority by Policy Manager. Figure 7-8 describes the definition of role.

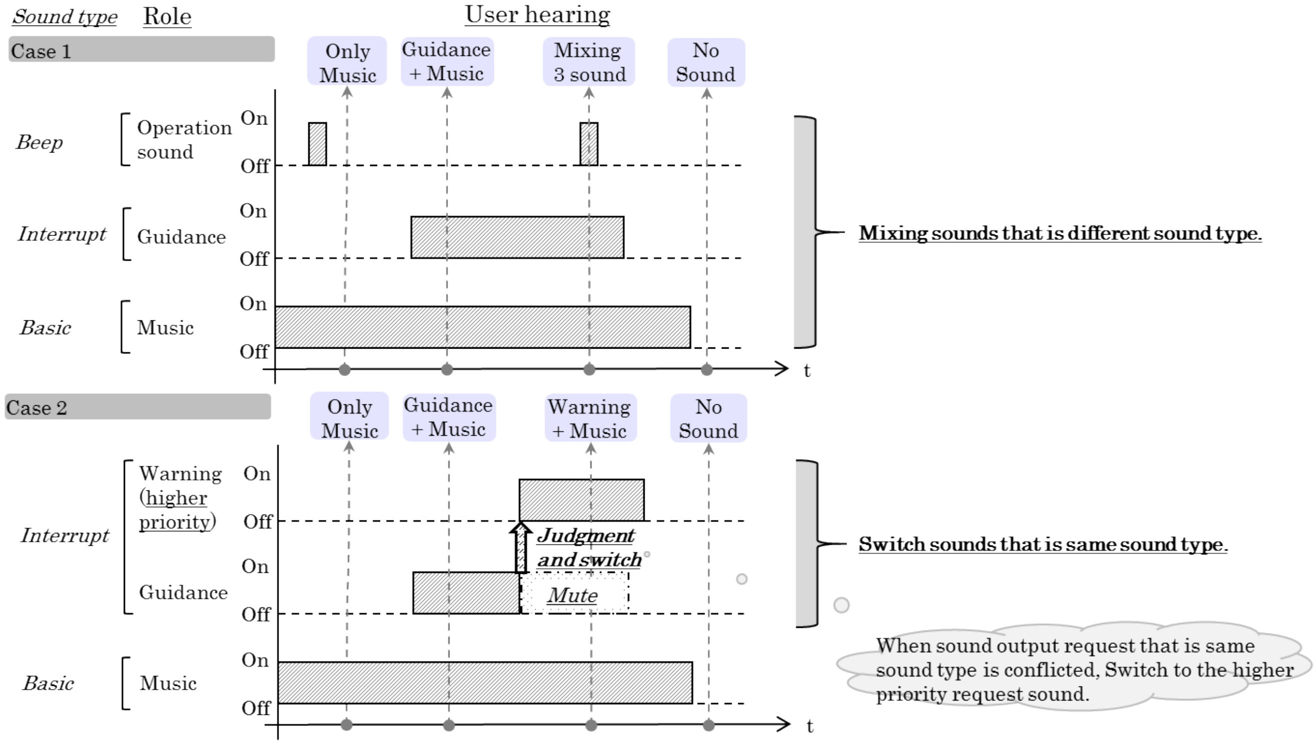


Figure 7-8: Sample Role

The sound resources in the same zone and same sound type are switched along the priority that is related to role by Policy Manager. In other words, the sound resources of different zones or different sound type are not switched. They are mixed.

b-3. GUI Resource Control

Policy Manager controls the sound resources using Vehicle Info Control. Policy Manager only issues to control the sound resources but it is actually controlled by Vehicle Info Control directly.

There are two types of sound resource control:

One is playback control such as play, pause and stop. Policy Manger issues to play sound for sound area owner, and if area owner was changed, then issue to stop previous playing sound stream and to start play latest area owner.

Other one is volume control. Two or more sound streams of same zone may playback simultaneously if each sound streams are assigned to different sound type. In this case, Policy Manager specifies volume parameter for each sound stream. For example, if route guidance and music playback are mixed, assign higher volume to route guidance and volume down for music playback.

c. Input Resource

c-1. External Condition Collection

Policy Manager receives the below definition that is related with input resource.

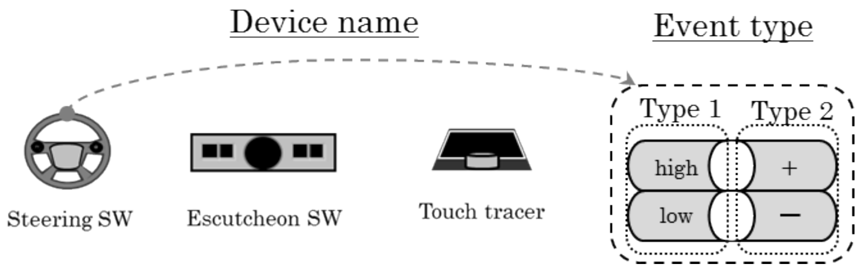


Figure 7-9: Definition of Input Resource

- Device Name

Device name is identity of input device such as steering SW and microphone.

- Event Type

Event type is logical group of input event from each input device such as volumes and temperatures.

c-2. Judgment of Priority of GUI resource

If application wants to be notified input event, it must request input event notice with device name and event type. The request is judged whether to notify by Policy Manager using policy DB. And Vehicle Info Control notifies input event to applications along the result of the judgment as below.

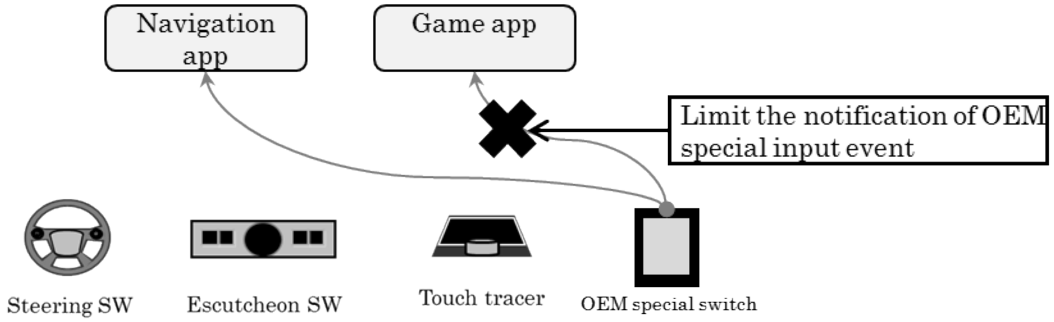


Figure 7-10: Definition of routing rule

OEM special switch means product variant configuration in Figure 7-10.

c-3. GUI Resource Control

Policy Manager controls the input resources using Vehicle Info Control. Policy Manager only issues to control the input resources but it is actually controlled by Vehicle Info Control directly.

Input resource control is to specify event target to Vehicle Info Control.

4.1.3.1.3 System Resources

(1) Definition

Policy Manager controls System resources according to external conditions. For example, Policy Manager limits memory usage of background applications when memory shortage was occurred.

Policy Manager controls System resources by using "Resource Control" of kernel layer. So, target resources are CPU, memory, storage bandwidth and network bandwidth.

a. Role

ID	Role	Description
1	External condition collection	(1) Receives the external conditions.
3	System resource control	1. Issue the System resource control according to external condition change. 2. Kill process(s) forcibly according to external condition change.

4.1.3.2 Requirements

4.1.3.2.1 Screen Resource

(1) External Condition Collection

System must provide a mechanism to receive the definition that is used judgment of resource owner.

System must provide a mechanism to receive the physical display information. Because system uses physical display information with to control surface to other system. The receive information must include as follows.

a. ID

b. Display resolution (Vertical and horizontal number of pixels)

c. DPI

d. Connected ECU

System must provide a mechanism to receive the layout definition. Layout definition must be able to identify the all areas of display. As a result, system recognizes the available area list according to current layout of each display.

The receive definition must include the follows.

a. ID

b. Area list

System must provide a mechanism to receive the area definition. Area is set application surface by system if the request is accepted by system. As a result, application surface displays on the device.

The receive request must include the follows.

a. Layout ID

b. ID

c. Area position (Coordinate of the upper-left)

d. Area size (Length * Width)

System must provide a mechanism to receive the layout type of each display. System can specify the available areas if layout type is defined. The receive information must include the follows.

a. Display ID

b. Layout ID

System must provide a mechanism to receive the priority rule. Because system must judge the providing resource using it when the request is collision.

The receive information must include the follows.

a. Role

b. Priority

System must provide a mechanism to receive the vehicle status. Because system must judge driving mode.

The receive information must include the follows.

a. Velocity

b. Brake status

System should provide a mechanism to receive the vehicle status. Because system should judge day night mode.

The receive information should include the follows.

a. The brightness of the interior

System should provide a mechanism to receive the user status. Because system should judge the providing resource using it.

System should provide a mechanism to receive the infrastructure status. Because system should judge the providing resource using it.

(2) Judgment of Priority of GUI Resource

System must provide a mechanism to assign resource owner to the requested resource according to external condition. This means that system judges the providing resource.

System must provide a mechanism to receive the layer request. System allocates the physical resource. Application must request the area on this layer if application needs to display the resource.

The receive request must include as follows.

a. Role

b. Layer type

The receive request should include as follows.

c. Display ID

System must provide a mechanism to receive the area request. System sorts layers in order by priority that is related with the specified role. Then system displays the application surface on the specified area on the specified layer.

The receive request must include as follows.

a. Role

b. Layer ID

The receive request must include as follows when layer type of the specified layer is “Basic”. Because there is a specification that the area on layer except basic type must be located on the related basic type area.

c. Area ID

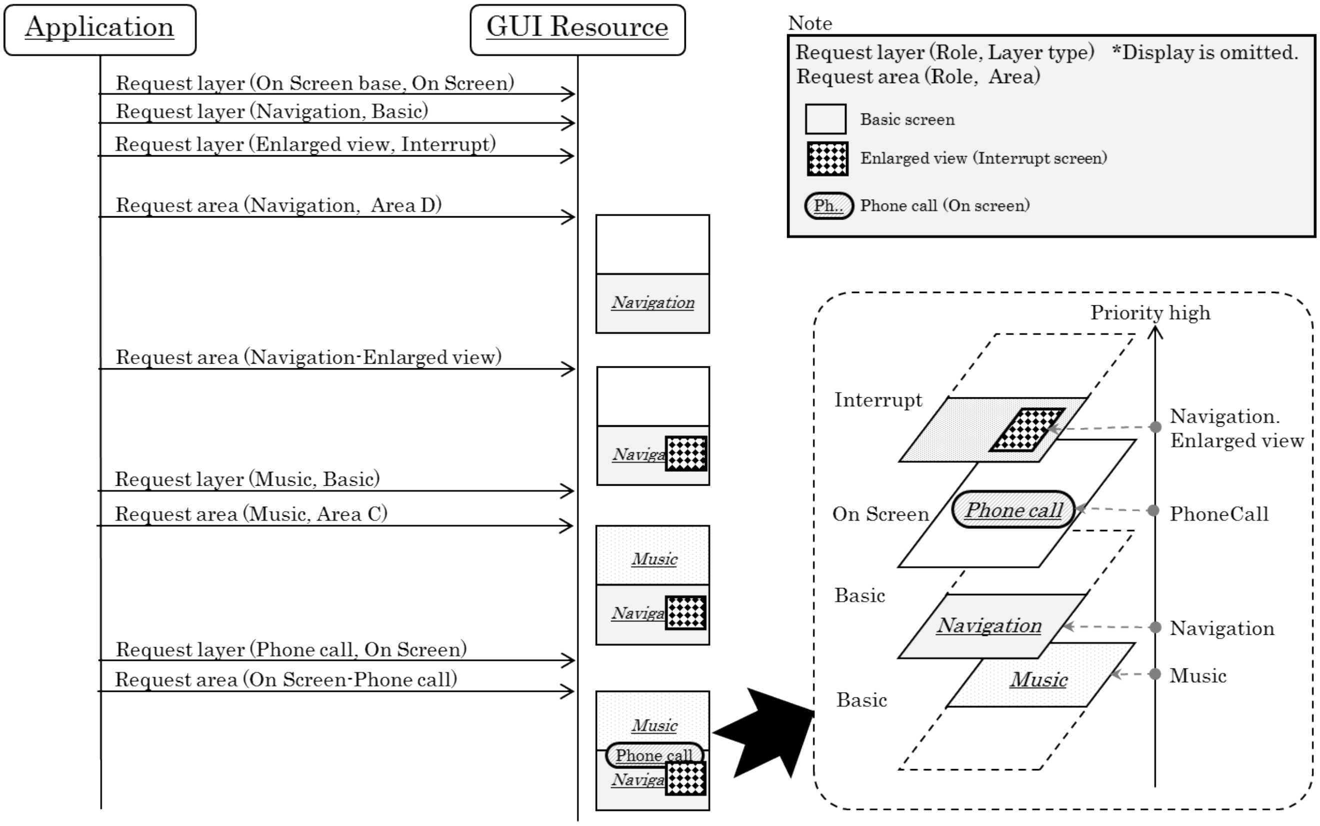

Application
GUI Resource
Request layer (On Screen base, On Screen)
Request layer (Navigation, Basic)
Request layer (Enlarged view, Interrupt)
Request area (Navigation, Area D)
Navigation
Request area (Navigation-Enlarged view)
Naviga
Request layer (Music, Basic)
Request area (Music, Area C)
Music
Naviga
Request layer (Phone call, On Screen)
Request area (On Screen-Phone call)
Music
Phone call
Naviga
Note
Request layer (Role, Layer type) *Display is omitted.
Request area (Role, Area)
Basic screen
Enlarged view (Interrupt screen)
Ph..
Phone call (On screen)
Priority high
Interrupt
Navigation.
Enlarged view
On Screen
Phone call
PhoneCall
Basic
Navigation
Navigation
Music
Music
Basic

Figure 7-11: Sequence to display

System should provide an interface to request both screen and sound resource simultaneously. In this request, requester should choose below options.

a. Requester needs both screen and sound. For example, if screen resource was available, but sound resource was occupied by other owner of higher priority, then, request should be refused.

b. Requester wants screen and sound resource as much as possible. For example, if screen resource was available, but sound resource was occupied by other owner of higher priority, then, only screen resource should be assigned to requester.

System should provide a mechanism to receive the request of forcibly acquire and forcibly release. System should be able to forcibly acquire and forcibly release request during system running. System should raise the requested surface to the top of the display.

The receive request should include the follows in addition to the information of the normal request.

a. Effective period (Can set unlimited)

System should not raise the other surface above its during effective period.

System should provide a mechanism to receive the request that is specified the following effect.

a. The effect at the transition

b. The effect of display surface

System must provide a mechanism to judge priority of resources. The screen resources are sorted of priority that is related to role by system. If display has two or more layers, then all layers will be superimposed by z-order.

System must provide a mechanism to judge visible surfaces according to vehicle running state. System must hide the surface that has too much information.

(3) GUI Resource Control

System must provide a mechanism to issue the resource control according to judgment.

System must provide a mechanism to issue the following resource control.

a. Visible / Invisible

b. Change position

c. Raise

The receive request must include as follows.

i. Surface ID *Only case of visible.

ii. Display ID *Only case of visible.

iii. Layer ID *Only case of visible.

iv. Position (Coordinate of the upper-left) *Only case of visible and change position.

v. Size (Length * Width) *Only case of visible.

System should provide a mechanism to set the following effect of the surface to other system.

a. The effect at the transition

b. The effect of display surface

4.1.3.2.2 Sound Resource

(1) External Condition Collection

System must provide a mechanism to receive the definition that is used judgment of resource owner.

System must provide a mechanism to receive the zone definition. Because system uses zone information with to control stream to other system. The receive information must include as follows.

a. ID

b. Sound device ID

System must provide a mechanism to receive the sound type definition. Because system uses sound type information with to control stream to other system. The receive information must include as follows.

a. ID

(2) Judgment of Priority of GUI resource

System must provide a mechanism to assign resource owner to the requested resource according to external condition. This means that system judges the providing resource.

System must provide a mechanism to receive the owner request. System must be able to receive request during system running.

The receive request must include as follows.

a. Role

b. Zone ID

c. Sound type ID

System should provide a mechanism to receive the request of forcibly acquire and forcibly release. System should be able to forcibly acquire and forcibly release receive request during system running.

The receive request should include as follows in addition to the information of the normal request.

a. Effective period (Can set unlimited)

System must assign resource owner as requested. And system must not assign resource owner by other request on same area during effective period.

System should provide a mechanism to receive the request that is specified the following effect.

a. The effect at the transition

b. The effect of output sound

System must provide a mechanism to judge priority of resources when there are two or more resources on same sound type on same zone. System judges the providing resource by priority of resources that is related to role.

* Boundary of the role between Policy Manager and application.

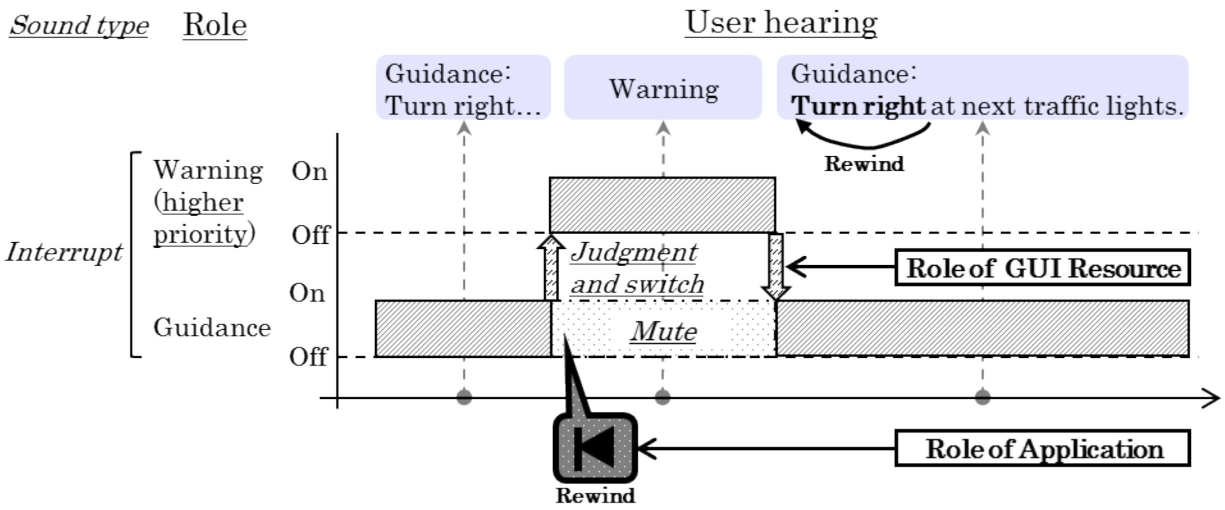


Figure 7-12: Boundary of role (Case of reverse)

System should provide a mechanism to manage order of the owner request. Because system should provide a mechanism to hold the request until the request is approved.

For example, if current playing interrupt sound completed, select the next play interrupt sound from request history based on the priority.

(3) GUI Resource Control

System must provide a mechanism to issue the resource control according to judgment.

System must provide a mechanism to issue the following resource control.

a. Mute / Unmute

b. Change zone

The receive request must include as follows.

i. Stream ID

ii. Device

In the case of multi-channel speaker, the receive request should include as follows.

iii. Channel ID

System should provide a mechanism to set the below effect of the sound to other system.

a. The effect at the transition

b. The effect of output sound

4.1.3.2.3 Input Resource

(1) External Condition Collection

System must provide a mechanism to receive the definition that is used judgment of resource owner.

System must provide a mechanism to receive the input device information. Because system uses input device information with to control input event to other system. The receive information must include as follows.

a. ID

System must provide a mechanism to receive the event type definition. Because system uses input device definition with to control input event to other system. The receive definition must include as follows.

a. ID

b. Related event IDs

(2) Judgment of Priority of GUI resource

System must provide a mechanism to assign resource owner to the requested resource according to external condition. This means that system judges the providing resource.

System must provide a mechanism to receive the owner request. System must be able to receive request during system running.

The receive request must include as follows.

a. Input device ID

b. Event type ID

System should provide a mechanism to judge whether to accept request according to the limitation routing rule of policy DB.

(3) GUI Resource Control

System must provide a mechanism to issue the resource control according to judgment.

System must provide a mechanism to issue the following resource control.

a. Set the routing rule

The receive request must include as follows.

i. Input device ID

ii. Event type ID

The receive request must include either as follows.

iii. The allowed application

iv. The denied application

System should provide a mechanism to set the following information.

a. Application that has active surface

System should notify the touch event from touch panel to user operating application. This feature is needed because there may be case that privilege application such as Homescreen changes the active surface.

4.1.3.2.4 System Resources

(1) External Condition Collection

System must provide a mechanism to collect external conditions to be used by Policy Manager to decide proper system resource.

Policy Manager must detect creation and deletion of process.

To detect creation of process, Policy Manager can assign proper system resource to created process.

Also, to detect deletion of process, Policy Manager can assign resources of deleted process to other active processes.

To assign proper system resource to specific process, system must provide a mechanism to identify process's role. In other words, Policy Manager must recognize the purpose of each active process.

Policy Manager must detect current memory consumption periodically.

To detect current memory consumption, Policy Manager can control maximum memory to each process to prevent memory shortage. Also, Policy Manager may kill processes which were thought as not so important process.

Policy Manager must detect current CPU consumption periodically.

To detect current CPU consumption, Policy Manager can control priority to each process to keep system performance. Also, Policy Manager may kill processes which seem to be in unexpected busy state.

System must provide a mechanism to notify application status change to Policy Manager. Application status includes as below.

- GUI resource status, such as foreground or background.
- Resuming last status or not. When system starts up or log-in user changes, system must resume last status. In this case, Policy Manager should assign much resource to last application to resume quickly as much as possible.

(2) System Resource Control

System must provide a mechanism to change assigned system resource per process or process group according to external conditions.

According to policy based decision, Policy Manager must assign proper system resource to target process or process group by using "Resource Control" of kernel layer. (typically cgroups will be used)

System must provide a mechanism to kill process or process group forcibly.

4.1.3.2.5 Resource Management

Resource Management shall consist of three functional components - Resource Manager, Policy Manager, Connection Manager.

Resource Management shall provide CORBA interfaces to rest of the components in the system.

Each resource request shall be in form a:

AppID,

SourceID,

RequestorZoneID,

NeedAll Flag (to specify if all the resources need to be allocated),

Required Resource List.

Resource Management shall be able to handle resource requests for Audio Sinks (eg: Cabin Speakers, HeadPhones)

Resource Management shall be able to handle resource requests for Video Sinks (eg: Display)

Resource Management shall be able to handle Source arbitration (Mic, WavPlayer instances, Tuners etc.)

Resource Management shall be able to validate all the input parameters for a resource request from resource requestors.

Resource Management shall be able to keep track of all the available resources.

Use CCF data to identify all the resources that are possible in the system. (static identification)

Use dynamic registration by the resource owners to identify what resources out of the above list are available at a point of time in the system. (dynamic identification)

Resource Management shall inform about resource availability and unavailability in the system through status update.

Resource Management shall support stacking/queuing of resource requests.

> Receive the requests from the resource requestors.

> Handle each request in chronological order and check for policy validation through Policy Manager.

> Add the validated requests into a priority queue.

> Process each request from the top of the queue for establishing the connection.

> If a request is still in the pending queue and the requestor requests to withdraw the request, it shall be removed from the queue.

Each request for resource shall be handled as an independent request irrespective of any earlier request by the same requestor. In case of multiple resources requested in a single request, it shall be treated as a single request and will be processed based on the request parameters.

If the NeedAll flag is set by the requestor, it shall either grant all the requested resources to the requestor or none of them shall be granted. There shall be no partial allocation of resources.

If the NeedAll flag is not set, it shall be able to do partial allocation of resources i.e. grant some/all of the resources requested by the requestor.

Resource Management shall provide an interface to a request owner to remove/withdraw an existing resource request.

Resource Management shall check for every requested resource against a pre-defined set of policies if the request can be served at this point of time or not. Below is a list of possible inputs for the policy decision:

> Currently Free or InUse Sink status

> Who is the resource owner of the currently used sink resource (if it is in use)

> Priority of the new requestor compared to the currently using requestor.

Resource Management shall use the system state as an additional input to make a decision if a request can currently be serviced or not. Below system states can be taken as input to the policy decision:

> Based on the speed restriction setting for a specific region, a request can be granted/kept pending.

> Low Power Mode, Eco Mode, System errors shall also be used to make policy decisions.

At any point of time it shall maintain the following information for each ZONE for use by resource requestor:

> Zone ID

> Allocated Source Instance

> Allocated Sink Instance

> Mute status

Resource Management shall not consider requirements to achieve a specific feature functionality (e.g. : Lowering audio volume of rest of the sinks when a phone call is in progress) as an input to the resource management policy.

Resource Management shall not provide support for requirements to achieve a specific feature functionality (e.g.: Pausing a pausable source when phone call is in progress).

Resource Management shall maintain priorities for all non-entertainment sources (eg: AMFM_TA, PHONE_NORMAL, NAV_VG, etc. shall all have priorities). In case two sources have same priority, the first requestor shall be granted a resource. In case of difference in priorities, the highest priority resource request shall be the one that is granted the resource.

Resource Management shall maintain same priority for all entertainment sources (eg: MP, DVD, AMFM_NORMAL, etc. shall all have the same priority). The last received Entertainment resource request will be the one that is granted the resource.

A valid (parameter and policy validated) resource request shall never be denied to the requestor. It shall either be granted or kept as a pending request in the priority queue.

Resource Management shall be responsible for reporting a broken resource status.

It shall be the responsibility of the resource requestor to remove the request from Resource Manager if the resource is no longer needed.

Resource Management shall assign a sink instance (the specific instance allocated out of all available instances of the requested sink type for a particular zone) to a resource request, once the request is granted against the set policy.

Resource Management shall maintain connection state of an already granted connection. Possible connection states are Active or Passive.

> When a source has the primary (master) control over a sink, the connection state will be active.

Ex: In normal mode, a driver requesting for AMFM source to Driver HeadPhone Sink connection.

> When a source has the secondary (slave) control over a sink, the connection state will be passive.

Ex: Driver using the AMFM source, at the same time the rear passenger requesting for same AMFM source on Rear headphone sink.

Resource Management shall be responsible for connecting/building a new source-sink connection using the underlying platform support.

Resource Management shall be responsible for removing/releasing an existing source-sink connection using the underlying platform support.

Resource Management shall request to mute the audio sink before an existing connection is removed/released.

Resource Management shall provide an interface to unmute the audio sink when a connection is re-established and the active source is ready to use the sink for audio routing.

Resource Management shall provide an interface to unmute an audio sink.

Resource Management shall inform the resource requestor when the sink is connected and ready to be used for audio routing.

Resource requestor needs to inform the Resource Manager when they are ready to start audio routing. This information shall be used to unmute the allocated sink.

Resource Management shall maintain the system connection table at any point of time. Connection table contains information regarding which sink is currently allocated to which source instance.

Resource Management shall support handling of change in behaviour based on Limo setting:

> Share the source between the Rear Seat headphone (Limo mode owner) and Cabin Speakers.

System shall support 4 ForegroundBeep sinks and 2 ForegroundSpeech sinks. 2 additional sinks are reserved for Engine noise synthesis which is outside the scope of this document. Additionally 1 FG speech sink and 1 FG beep sink is reserved for future use by ISC.

The number of sinks supported by the system shall be configurable through LCF parameter.

Headphones shall not be required to support any foreground sinks.

In case of Foreground sources and Tuner interrupt sources, any sink that is taken away from a source because of a high-priority interruption, need to be returned back to the previous source (if the request from the previous source is still valid and it's the next highest priority request).

As part of requirement to improve connection handling efficiency, it shall have exceptions to not disconnect the active connection while switching between any Tuner Source-Sink Background connection to another Tuner Interrupt Source with same sink connection.

It shall inform Resource Manager about a errors/failure in any of the existing sinks.

It shall inform Resource Manager about a errors/failure in any of the existing sources.

It shall provide the error state information about all resources to the Platform Error State Manager.

It shall inform the resource requestors in case the request is for an erroneous or faulty sink.

It shall wait for the application manager to notify it to prepare for shutdown.

It shall interact with the data storage manager to access (read and write) persistence data.

It shall interact with the data storage manager to access CCF data.

It shall support rules/exceptions (Blacklist) that define resource allocation strategy based on current system scenario.

E.g.: If there is a blacklist rule that says a Speech session shall not be allowed while phone call is in progress, then even if a FG sink is available, Speech shall be denied resources and kept as a pending request.

It shall provide an interface to receive Limo mode setting status.

It shall provide an interface to receive status when a rear-user selects to take Cabin control.

It shall use interfaces of early app to receive information if it's already using Audio/Video resources and update its internal status accordingly.

On any change in input to the Policy Manager (system state) it shall reevaluate all active connections and reconnect or disconnect if required.

E.g. An Amp gets disconnected, then all active connects have to be disconnected.

Once the Amp gets reconnected, the connection info shall be reevaluated and final set of connections shall be rebuilt with Amp.

It shall provide CORBA interfaces to the Resource Manager.

It shall be responsible for connecting/building a new source-sink connection using the underlying platform support.

It shall be responsible for removing/releasing an existing source-sink connection using the underlying platform support.

It shall request to mute the audio sink before an existing connection is removed/released.

It shall provide an interface to unmute an audio sink.

System shall support 4 ForegroundBeep sinks and 2 ForegroundSpeech sinks. 2 additional sinks are reserved for Engine noise synthesis which is outside the scope of this document. Additionally 1 FG speech sink and 1 FG beep sink is reserved for future use by ISC.

The no. of sinks supported by the system shall be configurable through LCF parameter.

It shall inform Resource Manager about a errors/failure in any of the existing sinks.

Headphones shall not be required to support any foreground sinks.

It shall wait for the application manager to notify it to prepare for shutdown.

It shall interact with the data storage manager to access (read and write) persistence data.

It shall interact with the data storage manager to access CCF data.

4.1.4 Sound Manager

A sound manager is a mechanism in which a sound output demand in two or more zones from two or more applications is arbitrated, an audio server manages control of a sound output and a policy manager manages a mediation rule.

A zone is a place in the car divided by the purpose of output power of sound like a driver zone, a passenger zone, and a rear seat zone. Each zone can play at the same time. Refer to "Sound resource" of "7.1.1.2 (2) Role" of "7.1 Policy Manager" for the details of a zone.

Applications that play and capture audio via the audio server, applications that control things like volume and routing via the audio server, and a policy manager that works with the audio server to implement automatic audio policies.

4.1.4.1 Use Case

Please refer “sound resource control” of Policy Manger section.

Table 7-14 describes the role of sound manager to be satisfied above purpose and use cases.

Table 7-15 : Role of Resource Control

No.	Role	Description
1	Routing sound streams	To route each sound stream to proper zone(s).
2	Mixing level control	Mixing two or more sound streams after volume control of each sound streams.
3	Sound effect	Provide a capability of sound effect as follows, · When changing sound stream. E.g. fade-in, fade-out and cross-fade.
4	Reduced dependency of hardware	Provide well-defined interface to reduce dependency of hardware. Well-defined interface also makes it possible to increase the effect of portability and development cost.

4.1.4.2 Requirements

4.1.4.2.1 Routing Sound Streams

System must provide a mechanism to manage sound ”zone”.

Refer to "(2) Sound resource" of "7.3.1.2.2 Role" of "7.3 Policy Manager" for the details of a zone and how to manage zone.

System must provide a mechanism to manage one or more connected sound devices, and each channels of each sound device.

One or more sound devices are usually connected to a system, and each sound device consists of one or more channels. And each channel outputs the sound of a monophonic recording.

For example, as for a stereo sound, a speaker is connected to each of two channels, and it is arranged at the driver side of a car, and the passenger seat side. If a telephone call is got when outputting stereo music from both of speakers, only the channel of a driver side needs to lower musical volume, and needs to mix and output the sound of a telephone (to louder sound than music). For this reason, the system needs to recognize and control each channel of each sound device.

The system must determine the route which outputs two or more sound streams to two or more zones.

Although the output place zone of a sound stream may change dynamically according to the present state of vehicles and a policy manager makes the decision, sound manager requires the mechanism in which a route is smoothly changed based on the determination of policy manager.

System must provide a mechanism to manage two or more sound zone as grouped zone.

System must provide a mechanism to do volume control for specific zone.

All the sound outputted to a certain zone is adjusted by the volume of the zone.

System must provide a mechanism to control sound stream.

Control of a sound stream is as follows.

- Mute/unmute: System must provide a mechanism to do mute or unmute to any sound stream.
- Suspend/resume: System must provide a mechanism to suspend or resume to any sound

stream.

Volume control: System must provide a mechanism to change volume to any sound stream.

4.1.4.2.2 Mixing Level Control

The system must offer the mechanism for arbitrating two or more sound streams outputted to the same zone according to a policy manager's arbitration.

System must provide a mechanism to do mixing after volume control of each sound streams.

System must provide a mechanism to attenuate sound volume when other sound stream requested to play into same sound zone.

In this case, system must also provide a mechanism to return to the volume before attenuating the volume of a sound stream when interrupted sound stream was ended.

System must provide a mechanism to mute sound volume when other sound stream requested to play into same sound zone.

In this case, system must also provide a mechanism to unmute sound volume when interrupted sound stream was ended.

System must provide a mechanism to suspend sound stream playback when other sound stream requested to play into same sound zone.

In this case, system must also provide a mechanism to resume playback when interrupted sound stream was ended.

4.1.4.2.3 Sound Effect

When sound stream was changed, system must provide a mechanism to do sound effect.

System must provide typical sound effect such as fade in and fade out.

System must provide a mechanism to add, replace and delete sound effect easily by using plugin architecture.

4.1.4.2.4 Reduced Dependency of Hardware

Sound Manager must be able to retrieve system structure regarding sound device and channels of each device. And the system must enable addition/deletion of a sound device by the means which does not need rebuild of systems, such as a configuration.

4.1.5 Input Manager

The Input Manager provides a capability to deliver input events to the proper application depending on request from Policy Manager. Policy Manager will decide event target per each input area. Also, the IVI system may use various car-oriented input devices such as steering switch. Input manager provides a capability to abstract such kind of input event.

4.1.5.1 Use Case

Please refer “input resource control” of Policy Manger section.

By the way, associated input devices are listed below.

No.	Input type	Associated device	Description
1	Key	Steering switch	Simple key event. Deliver to application.
2	Keyboard	Virtual keyboard	Keyboard event. Deliver to application, then use input method backend if needed.
3	Touch	Touch panel	Touch event, such as start, stop and move. Also supports double click and multi-touch capability. Deliver to application.
4	Sound	Microphone	Sound input.

			Deliver to application or voice recognition engine.

Table 7-14 describes the role of input manager to be satisfied above purpose and use cases.

Table 7-14 : Role of Resource Control

No.	Role	Description
1	Abstract device event	Provide capability to abstract from device event to application readable event name, such as "volume up" and "right arrow".
2	Event delivery	Provide capability to deliver input event to specified application.

4.1.5.2 Requirements

4.1.5.3 Abstract Device Event

System must provide a mechanism to re-configuration regarding input devices without re-build.

Because, connected input devices may different by car grade, car type, destination and optional equipment.

4.1.5.4 Event Delivery

System must provide a mechanism to deliver any input event to any application.

System must provide an interface to apply event delivery rule by using attribute pair "device id" and "destination application id".

Device id specifies a logical device name. Logical device name will link to physical device by UIM.2.1.2.

Also, system must provide a mechanism to change event delivery rule dynamically.

System must provide a mechanism to link between logical device name and physical device.

System must provide a mechanism to deliver any input event to any application depending on delivery rule defined in UIM.2.1.1.

System must provide a mechanism to inhibit any event delivery.

This function makes it possible to restrict input event during driving mode.

4.1.6 User Manager

4.1.6.1 Use Case

4.1.6.2 Personal Identification

User manager provides multi-user environment. A car may be used by two or more people, and a person may use two or more cars, by using rent-a-car, for example.

4.1.6.3 User Preference

Multi-user environment provides same user experience for each user.

Also, multi-user aims seamless personal data sharing not only between cars but also including other devices such as smartphones and smart TVs. Furthermore, it will include seamless data sharing from your home and your office.

Identify the person, and log-in to the IVI system as a specified user. Personal identify may be provided by traditional user name and password pair, smart key or biometrics.

Once a user has logged-in to IVI system, IVI system should provide personalized user experience. For example, Bob uses English, but Alice uses French. Also, Bob likes rock-music, but Alice likes classic-music. In this case, English and rock-music should be selected when Bob is logged-in, and Japanese and classic-music should be selected when Alice is logged-in.

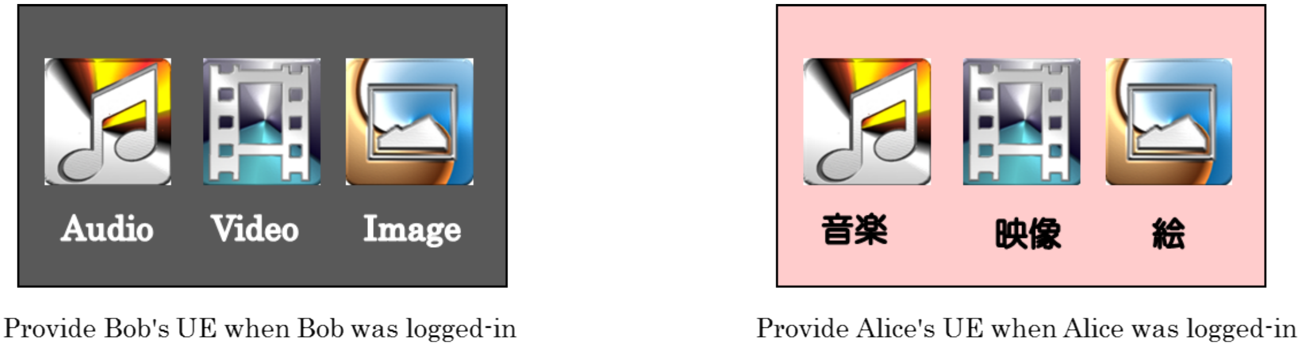

Audio
Video
Image
音楽
映像
絵
Provide Bob's UE when Bob was logged-in
Provide Alice's UE when Alice was logged-in

Figure 7-24 : Provide Logged-in User’s UE (User Experience)

4.1.6.4 Rent-a-car and/or Replacing a Car

When Bob uses a rent-a-car, same preference should be adapted as if he rode his own car. If Bob’s preference was stored in a cloud, then this can be supported. However, security is important in this scenario. For example, Bob must not be able to access to other user’s preference.

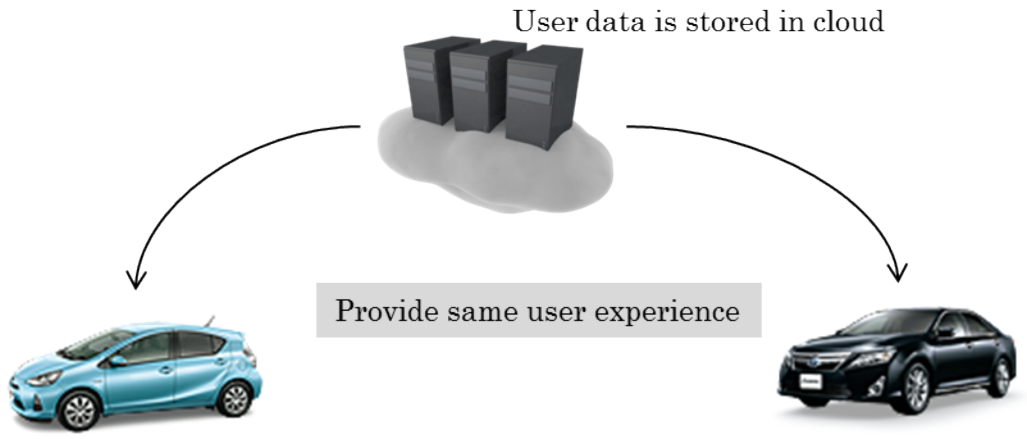


Figure 7-25 : User data sharing between cars

4.1.6.5 Seamless Data Sharing

Cloud-based user data syncing will enable seamless data sharing between IVI systems and smart-phones, home networks and accessing from your offices.

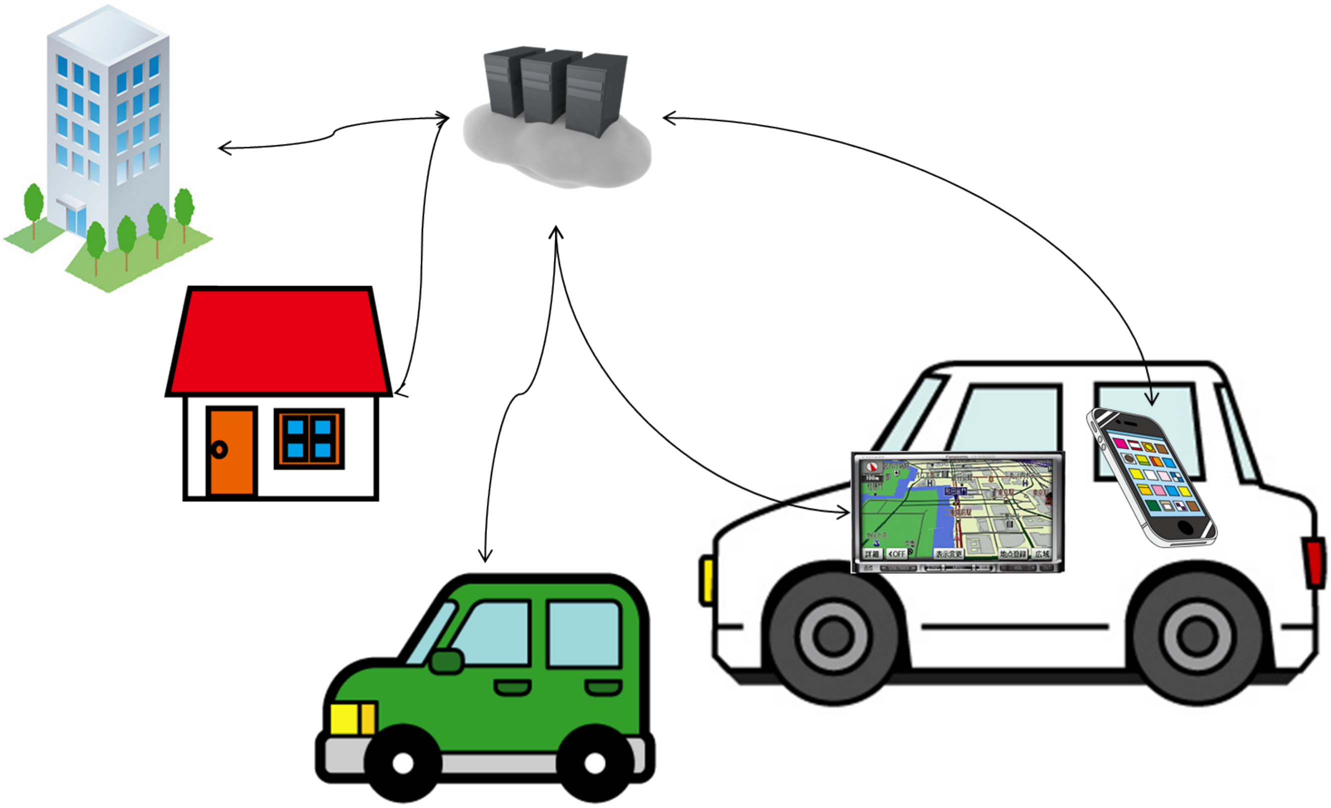


Figure 7-26 : User data sharing over the cars

4.1.6.6 Role

Error! Reference source not found. describes the role of the User Manager to satisfy the above purpose and use cases.

Table 7-17 : Role of User Manager

No.	Role	Description

1	User identification	Provide a mechanism to identify user, such as user name and password pair, smart key and biometrics. Provide a mechanism to log-in to the IVI system as a specified user. When a different user logs in, proper user preference for the user must be applied, and resume last state of corresponding user. Also, each application can store application's data per user. In such cases, proper user data must be applied when a different user logs in.
2	User preference	Provide a mechanism to apply user preference of logged-in user. User preference includes the following data. · User interface, such as locale and wall-paper. · Resume last application's status of specified user. · Application specific data.
3	User data management	Provide a mechanism to manage cloud based user data. The following capabilities are required. · Download user data of the logged-in user from the cloud. · Update cloud data if the user data was updated by user operation or otherwise. · Periodically sync-up w/ cloud because user data may be updated by other devices. In addition to the above basic capabilities, user data cache is essential for a car, since a car may not always have a reliable network connection.
4	Security	Because cloud based sharing user data may be accessed from any place, user data must be protected from unexpected data access.

		So, IVI system must provide security mechanism regarding accessing to cloud based user data.

4.1.6.7 Requirements

4.1.6.7.1 User Identification

System must provide a mechanism to identify logged-in user.

System must provide a mechanism to enter user name and password, and verify password to identify logged-in user.

System should provide a mechanism to read smart key attribute to identify logged-in user. For example, using NFC.

System should provide a mechanism to identify logged-in user by using biometrics.

4.1.6.7.2 User Preference

When a logged-in user is identified, system must apply user preference depending on the currently logged-in user.

System must provide a mechanism to apply personalized user experience as follows.

- Locale settings
- UX theme

Wall paper

System must provide an easy mechanism to add plugin function and/or attribute of personalized user experience.

System must provide a mechanism to switch application data per user, and apply logged-in user's application data automatically.

When user is identified and logged-in, the system must apply last status of logged-in user. Last status refers to the status of the system as the current logged-in user has last logged-out of the system. Specifically, last status includes the following.

- Foreground applications. That means displayed applications.

Background applications.

When user logs in for the first time, the system must apply user preference for new log-in user.

System must provide a mechanism to apply default preference attributes for new log-in user.

System must provide default preference attributes and HMI to apply for first time log-in user.

4.1.6.7.3 User Data Management

System must provide a mechanism to manage user data.

AGL defines "user data" as a general term which includes all the data necessary to realize user preference.

User data shall be stored in the cloud. The cloud provides user data not only to IVI systems but also other systems and/or devices such as smartphones, Home-PCs, business-PCs, HEMS and home electronics.

System must provide a mechanism to apply user preference and to supply user data to application by using cloud based user data.

System must provide a mechanism to download cloud based user data and apply it as user data of the IVI system.

When user data is updated in the IVI system, then the system must upload updated user data to the cloud.

Also, since other device or system may update shared user data elsewhere, system must provide a mechanism to sync with the cloud periodically to keep user data in the IVI system up-to-date.

Because the IVI system is not necessarily connected to a network, the system must provide a mechanism to cache downloaded user data.

If the IVI system re-connected to a network, system must sync with the cloud as soon as possible.

4.1.6.7.4 Security

Because user data may include personal information, system must provide a mechanism to protect user data from risks including but not limited to leakage, tampering and theft.

System must provide a mechanism to protect user data when accessing to the cloud.

- System must authenticate communication entity. In other words, IVI system must authenticate cloud server, and cloud server must authenticate client such as IVI system, smartphone or PC.
- System must provide a mechanism to encrypt transported data via a network.
- System must provide a mechanism to transport data via a network with protection against falsification of data from unauthorized access or illegal access.
- Cloud server must provide a mechanism to authenticate individual user, and provide user data only to the authorized user.

Because, two or more user's user data may be stored in IVI system as a cache, system must provide a mechanism to protect cache data from other users. The protection of cached data to include not only the current multi-user environment risk, but also the risk of attacks against cached data. In other words, only logged-in user's cache data can be accessed.

4.2 Web HMI

Web based HMI. Contains applications, web runtime environment, and web-based home screen.

4.2.1 Web API

It is discussed that HMI parts of IVI system will be developed using HTML5. APIs to use service function in IVI system from web applications is needed. Audio Visual API provides APIs for audio visual equipment control to web applications. (e.g. Media files on storage, CD, DVD, BT-Audio, Photo, etc.)

Web applications use Audio Visual API to play audio visual contents on IVI system. Use case of Audio Visual API is shown in Figure 6-1.

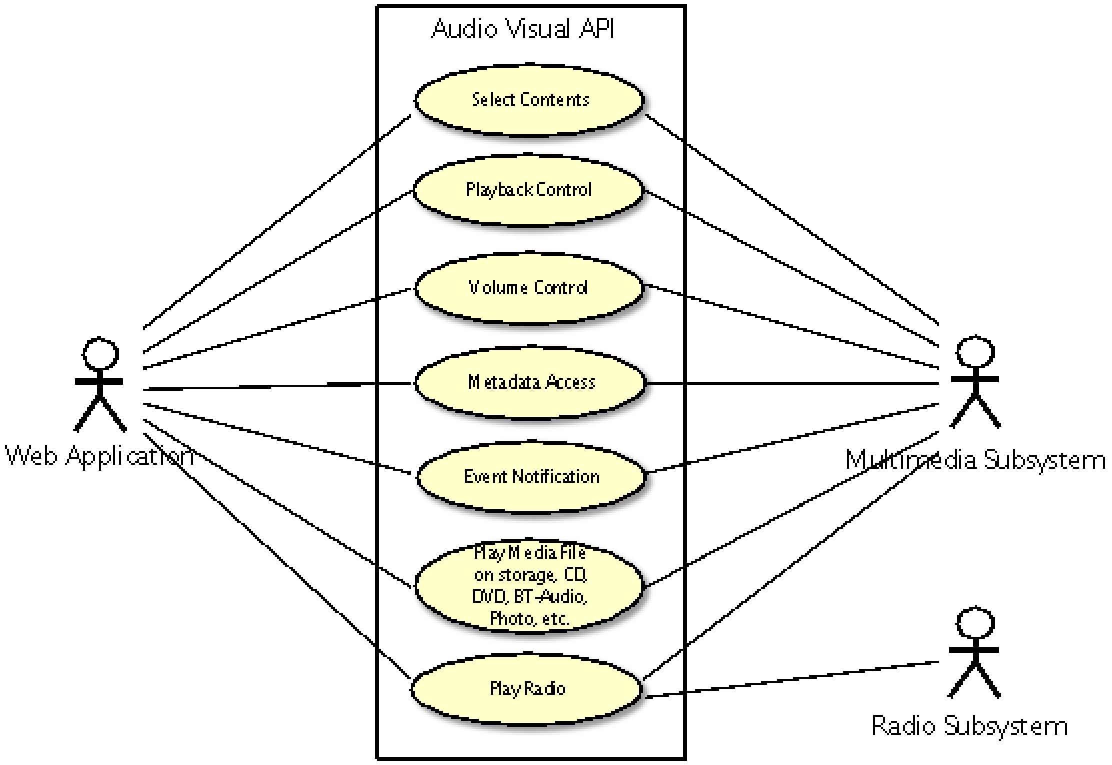


Figure 6-1: Use case of Audio Visual API

4.2.1.1 Requirements

Audio Visual API must provide API to select Audio Visual contents.

- Select content using URL
- Select content using contents list provided by multimedia subsystem

Audio Visual API must provide API to playback Audio Visual contents. (Media file on storage, CD, DVD, BT-Audio, Photo, etc.)

- Play
- Pause
- Fast-forward
- Rewind
- Track up
- Track down
- Select playmode (Repeat/Random)

Audio Visual API must provide API to control a volume.

- Volume up
- Volume down
- Mute

Audio Visual API must provide API for metadata access about Audio Visual contents.

Audio Visual API must provide API for notifications.

- The case that playback state is changed
- The case that Audio Visual contents is add / removed

Audio Visual API must provide API to play AM/FM radio.

- Change the frequency.
- Change the broadcasting stations.
- Receive the list of broadcasting stations.
- Select the preset channel.
- Get the information of the broadcasting station.

Audio Visual API must provide API to play digital radio.

- Store the broadcast program information.

- Get the broadcast program information.
- Get the play time.
- Play the radio broadcast cached.

AGL System must support a web API to access Vehicle information.

AGL System must support web API to control STT/TTS daemon.

AGL System must support web API to control navi engine.

AGL System needs to provide a Web API to allow peer to peer communication between two web apps.

AGL System needs to provide an API to allow peer to peer communication between a web app and a native app.

AGL System must support access control over app to app communications. Service provider should be able to restrict subscriber.

AGL System must support W3C/HTML5 DOM, Forms and Styles.

AGL System must support W3C/HTML5 Device APIs: Touch Events, Device Orientation, Network Information

AGL System must support W3C/HTML5 Graphics APIs: canvas, canvas 2D context, and SVG

AGL System must support W3C/HTML5 Media: audio and video tags, user media and web audio

AGL System must support W3C/HTML5 Communication APIs: websocket, web messaging, server sent events, session history of browsing context

AGL System must support W3C/HTML5 Storage APIs: Web storage, File, Database, Web SQL

AGL System must support W3C/HTML5 Security APIs: Cross-Origin Resource Sharing, HTML5 The iframe element, Content Security Policy 1.0.

AGL System must support W3C/HTML5 UI APIs: Clipboard, DnD, Web Notifications

AGL System must support W3C/HTML5 Performance APIs: Web workers, Page Visibility, Timing control, Navigation timing

AGL System must support W3C/HTML5 Location API: Geolocation

AGL System must support W3C/HTML5 Widget: Widget Packaging and XML Configuration, Widget Interface, XML Digital Signatures for Widgets, Widget Access Request Policy

AGL System must support Khronos WebGL API.

4.2.2 Web Runtime

The Web Runtime module contains the bindings for the Web Application Framework to access the AGL Application Framework and Services.

4.2.2.1 Requirements

AGL system Web Runtime shall provide full web application lifecycle management (e.g., installation/removal).

AGL System Web Runtime shall provide full execution environment for web apps (i.e., launch, view generation, rendering, etc.)

AGL system Web Runtime shall provide a mechanism to implement plugins/extensions to add better device/platform integration.

AGL system Web Runtime shall provide a mechanism to manage apps' access control and also to categorize apps with different privileges.

System must provide high level GUI components for Web application.

At least, below components are required.

- Text labels
- Button
- Radio button
- Check box
- Tab panel
- Animation (e.g. MNG, GIF animation)
- Slider
- Accordion list
- Anchor
- Text input form
- Dropdown list box
- Date picker

4.3 Native HMI

The Native HMI provides an application framework for those applications that are not written using Javascript or other web technologies.

4.3.1 Native App Runtime

The Native Runtime module contains the bindings for the Native Application Framework to access the AGL Application Framework and Services.

4.3.1.1 Requirements

System must provide high level GUI components for native application.

At least, below components are required.

- Text labels
- Button
- Radio button
- Check box
- Tab panel
- Animation (e.g. MNG, GIF animation)
- Slider
- Accordion list
- Anchor
- Text input form
- Dropdown list box
- Date picker

4.3.2 Native Application Framework

The platform can support multiple application frameworks any of which may be built into an SDK or product build. The application framework contains any code specifically written for that framework as well the bindings to the Services and Operating Systems layers that the application framework provides for its applications.

5 Services Layer

The Services Layer contains user space services that all applications can access. Generally the services provide either an IPC type interface or a subroutine/ function API. These interfaces remain the same for a given implementation and it is up to the Application Framework Runtime modules to provide access to these interfaces to the applications. Since we are trying to avoid unnecessary interface shims, it is not necessary for AGL to define standard service layer interfaces for a given module. Unless otherwise specified the API depends upon the interfaces provided by the open source packages chosen for a module. Different implementations may choose different packages for a given function and it is left to the Application Framework runtime to adjust to any new interfaces,

5.1 Platform Services

Platform Services Layer. Conventional Linux platform services

5.1.1 Bluetooth

This document describes requirements regarding registration, (dis)connection and device information management between Bluetooth device and infotainment system. Necessary Bluetooth profiles in automotive use case are defined here.

5.1.1.1 Requirements

The Telephony system shall be designed to

support a minimum of BT3.0+EDR, but shall be possible to upgrade to Bluetooth 4.0+EDR without hardware upgrade.

A Bluetooth hands-free system shall provide the following BT profiles:

- Core 2.0 + EDR inc. GAP (Generic Access Profile)
- HFP (Hands Free Profile)
- OBEX (Object Exchange)
- OPP (Object Push Profile)
- PBAP (Phonebook Access Profile)
- SPP (Serial Port Profile)
- SDAP (Service Discovery Access Profile)

If the BT system is designed to operate with BT Media Players (E.g. control and stream music from), the system shall also support the following incremental BT profiles:

- A2DP (Advanced Audio Distribution Profile)
- AVRCP (Audio Visual Remote Control Profile)

The link key shall be minimum 128 bits. The encryption key is negotiated and shall be set at the highest supported value by the remote device. The Telephony system shall be capable of generating up to 128-bit encryption key. The Telephony system will not be the limiting device in encryption key length negotiation.

When implemented by the remote device Simple Secure Pairing 'Numeric comparison' method as default pairing mechanism. However when remote device is limited a configurable priority scheme will be adopted where the order of mechanisms will be determined at configuration time.

The Telephony system shall provide Bluetooth Power Class 2. The operating range of Class 2 is 10 meters and maximum power is 2.5 mW (4 dBm).

The Telephony system shall have provision for 1, 3 and 5-slot packet transmission. It shall allow using five-slot packet transmission for faster data rate.

The Telephony system shall use IrMC standards as directed by the BT specification. It is a standard from IrDA, including IrOBEX for object exchange including vCards, vCalendars, etc.

vCard is the electronic business card. It is used for Personal Data Interchange (PDI). vCards are often attached to e-mail messages, and can be exchanged on Instant Messaging. vCard contain name and address information, phone numbers, and e-mail addresses.

vCard version 2.1 is widely adopted by e-mail clients. It contains FN, N, PHOTO, BDAY, ADR, LABEL, TEL, EMAIL, MAILER, TZ, GEO, TITLE, ROLE, Logo, Agent, ORG, NOTE, REV, SOUND, URL, UID, Version, and KEY properties.

vCard version 3.0 is IETF standards format. It is defined in following two parts:

MIME Content-Type for Directory Information

vCard MIME Directory Profile

It contains NICKNAME, CATEGORIES, PRODID, SORTSTRING and CLASS properties along with the vCard version 2.1 properties.

The touch-screen or head unit HMI must have the ability to delete a Bluetooth device and any associated data (E.g. phonebook, voicemail number) when required, even if the BT device list is not full.

The Telephony system shall use SCO link for voice data if eSCO link is not supported else eSCO shall be used.

5.1.1.1.1 Hands Free Profile

The Telephony system shall implement Hands-Free Profile (HFP) as per the hands-free Profile specification version 1.6 or later.

The Telephony system shall enable a headset, or an embedded Hands-Free unit to connect, wirelessly, to a cellular phone for the purposes of acting as the cellular phone's audio input and output mechanism and allowing typical Telephony functions to be performed without access to the actual phone.

It shall provide following roles:

Hands-Free unit (HF)

Table 19 : List of HFP supporting functions

No.	Feature	Support in HF	AGL
1	Connection management	Mandatory	x
2	Phone status information	Mandatory	x
3	Audio Connection handling	Mandatory	x
4	Accept an incoming voice call	Mandatory	x
5	Reject an incoming voice call	Mandatory	x
6	Terminate a call	Mandatory	x
7	Audio Connection transfer during an ongoing call	Mandatory	x
8	Place a call with a phone number supplied by the HF	Option	x
9	Place a call using memory dialing	Option	-
10	Place a call to the last number dialed	Option	-

11	Call waiting notification	Option	x
12	Three way calling	Option	x(*1)
13	Calling Line Identification (CLI)	Option	x
14	Echo canceling (EC) and noise reduction (NR)	Option	x
15	Voice recognition activation	Option	x
16	Attach a Phone number to a voice tag	Option	-
17	Ability to transmit DTMF codes	Option	x
18	Remote audio volume control	Option	-
19	Respond and Hold	Option	x
20	Subscriber Number Information	Option	x
21a	Enhanced Call Status	Option	x
21b	Enhanced Call Controls	Option	-
22	Individual Indicator Activation	Option	-
23	Wide Band Speech	Option	x
24	Codec Negotiation	Option	x

*1: Does not support Multi-party (conference) call

The Telephony system shall be able to use the AT+CGMM query/response to determine the model of the phone over the HFP profile connection. Whatever is returned shall be stored as a string in a phone model CGMM variable.

- Phone Model CGMM:
- Type: string
- Max length: 200 chars

- Persistence: No

A property shall exist for each device which is connected to the system.

The request shall be made each time a HFP Service Level Connection is established with the device.

The Telephony system shall be able to use the AT+CGMI query/response to determine the Manufacturer of the phone over the HFP profile connection. Whatever is returned shall be stored as a string in a phone model CGMI variable.

- Phone Model CGMI:
- Type: string
- Max length: 200 chars
- Persistence: No

A property shall exist for each device which is connected to the system.

The request shall be made each time a HFP Service Level Connection is established with the device.

The Telephony system shall be able to use the AT+CGMR query/response to determine the revision of the phone over the HFP profile connection. Whatever is returned shall be stored as a string in a phone model CGMR property.

- Phone Model CGMR:
- Type: string
- Max length: 200 chars
- Persistence: No

A property shall exist for each device which is connected to the system.

The request shall be made each time a HFP Service Level Connection is established with the device.

5.1.1.1.2 Advanced Audio Distribution Profile (A2DP)

The Telephony system shall implement Advanced Audio Distribution Profile as per the A2DP specification version 1.2 or later.

The Telephony system shall use this profile for audio streaming. This profile shall be use to realize distribution of audio content of high-quality in mono or stereo on ACL channels.

It shall provide following roles:

Sink (SNK) - A device is the SNK when it acts as a sink of a digital audio stream delivered from the SRC on the same piconet.

Items marked with "x" in AGL column in Table 20 should be supported.

Decode functions of codec marked with "x" in AGL column in Table 21 should be supported.

Copyright protection technology SCMS-T should be supported.

Table 20 : List of A2DP Supporting Functions

No.	Feature	Support in SNK	AGL
1	Audio Streaming	Mandatory	x

Table 21 : Supporting Codec

No.	Codec	Support	AGL
1	SBC	Mandatory	x
2	MPEG-1,2 Audio	Option	-
3	MPEG-2,4 AAC	Option	-
4	ATRAC family	Option	-

5.1.1.1.3 Phone Book Access Profile

The Telephony system shall implement Phonebook Access Profile as per the PBAP specification version 1.1 or later.

The Telephony system shall use this profile to allow exchange of Phonebook Objects between devices.

Phonebook is automatically downloaded into the system from mobile device for browsing. The Telephony system shall store user's Phonebook and the Phonebook details of the connected device shall be available to the user. The Telephony system shall manage the contacts by, listing and copying contact information.

It shall provide following roles:

- Phonebook Client Equipment (PCE)

It shall provide following types of Phonebook objects:

- The main Phonebook object
- The Incoming Call History object
- The Outgoing Call History object
- The Missed Call History object
- The Combined Call History object

A Bluetooth hands-free system must download the phonebook from the connected BT device automatically if the BT device has provision for the transfer of phonebook data. The Phonebook download shall be performed by any one of the following methods listed in priority of usage:

- Using PBAP profile

All the BT device's phonebook entries must be transferred - those on any external memory (E.g. SIM) and also any stored in the BT device's memory.

The number type data (if stored with the contact) shall also be transferred and stored in the vehicle phonebook. The Phonebook shall be associated to only the BT device it was downloaded from.

5.1.1.1.4 Dial Up Networking (DUN) Profile

Dial-Up Networking Profile (DUN) has to be supported as well as Profiles/Protocols for necessary lower layers.

It has to comply with the specification for “Data Terminal (DT)”

Items marked with "x" in AGL column in Table 23 should be supported.

Table 23 : List of DUN Supporting Functions

No.	Service	Support in DT	AGL
1	Data call without audio feedback	Mandatory	x
2	Data call with audio feedback	Option	-
3	Fax services without audio feedback	N/A	-
4	Fax services with audio feedback	N/A	-
5	Voice call	N/A	-
6	Incoming calls	Option	x
7	Outgoing calls	Mandatory	x

5.1.1.1.5 Object Push Profile (OPP)

Object Push Profile (OPP) has to be supported as well as Profiles/Protocols for necessary lower layers.

It has to comply with the specification for “Push Server”.

Items marked with "x" in AGL column in Table 24 should be supported.

Table 24 : List of OPP Push Server Supporting Functions

No.	Feature	Support in Push Server	AGL
1	Object Push	Mandatory	x
2	Business Card Pull	Option	-
3	Business Card Exchange	Option	-

5.1.1.1.6 Audio/Video Remote Control Profile (AVRCP)

The System shall implement Audio / Video Remote Control Profile version 1.6.

The system shall use this profile for audio streaming control for each connected media device plus one remote control..

The system must comply with the specification for Controller (CT) items marked with "x" in AGL column in Table 25 should be supported.

C2: Mandatory if device supports Metadata Attributes for Current Media Item or optional otherwise

C3: Mandatory to support at least one Category

C4: Mandatory if Category 2 supported, excluded otherwise

C6: Mandatory if Browsing (item 18) is supported, optional otherwise

EX: Excluded

Table 25 : List of AVRCP Supporting Functions

No.	Feature	Support in CT	AGL
1	Connection establishment for control	Mandatory	x
2	Release connection for control	Mandatory	x
3	Connection establishment for browsing	C6	x

4	Release connection for browsing	C6	x
5	AV/C Info commands	Option	x
6	Category 1: Player/Recorder	C3	x
7	Category 2: Monitor/Amplifier	C3	-
8	Category 3: Tuner	C3	-
9	Category 4: Menu	C3	-
10	Capabilities	Option	x
11	Player Application Settings	Option	x
12	Metadata Attributes for Current Media Item	Option	x
13	Notifications	C2	x
14	Continuation	C2	x
15	Basic Group Navigation	Option	x
16	Absolute Volume	C4	-
17	Media Player Selection	Option	x
17.1	- Supports Multiple Players	Option	x
18	Browsing	Option	x
18.1	- Database Aware Players	Option	x
19	Search	Option	-
20	Now Playing	C6	x
20.1	- Playable Folders	Option	x
21	Error Response	EX	-
22	PASSTHROUGH operation supporting press and	Option	x

	hold		

The AVRCP profile realisation shall implement an Inform Battery Status of CT parameter and pass this information up to so it can be passed to the User.

5.1.1.1.7 Message Access Profile

Message Access Profile (MAP) has to be supported as well as Profiles/Protocols for necessary lower layers.

It has to comply with the specification for “Message Client Equipment (MCE)”.

Items marked with "x" in AGL column in Table 26 should be supported.

C1: The MCE to support at least one of the C1-labelled features

C2: The MCE shall support Message Notification Registration if it supports Message Notification. Not applicable otherwise.

Table 26 : List of MAP Supporting Functions

No.	Feature	Support by the MCE	AGL
1	Message Notification	C1	x
2	Message Browsing	C1	x
3	Message Uploading	Option	x
4	Message Delete	Option	-
5	Notification Registration	C2	x

5.1.1.1.8 Serial Port Profile (SPP)

The Telephony system shall implement Serial Port Profile as per the SPP specification version 1.1 or later.

It shall provide following roles:

Initiator - This is the device that takes initiative to form a connection to another device.

Acceptor - This is the device that waits for another device to take initiative to connect.

Following features shall be provided by the Supplier:

Establish link and setup virtual serial connection

Accept link and establish virtual serial connection

Register Service record for application in local SDP database

5.1.1.1.9 Personal Area Network (PAN) Profile

Personal Area Network Profile (PAN) has to be supported as well as Profiles/Protocols for necessary lower layers.

It has to comply with the specification for “PAN User (PANU)”.

Items marked with "x" in AGL column in Table 27 should be supported.

Table 27 : List of PAN Supporting Functions

No.	Feature	Support in PANU	AGL
1	Initialization of NAP/GN service	-	-
2	Shutdown of NAP/GN service	-	-
3	Establish NAP/GN service Connection	Mandatory	x

4	Lost NAP/GN Service Connection	Mandatory	x
5	Disconnect NAP/GN Service Connection	Mandatory	x
6	Management Information Base (MIB)	-	-

5.1.1.1.10 Service Discovery Profile (SDP)

The Telephony system shall implement Service Discovery Application Profile as per the SDAP specification version 1.1.

The Telephony system shall use this profile to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device.

It shall provide following roles:

Local Device - A device that initiates the service discovery procedure.

Remote Devices(S) - A device that participates in the service discovery process by responding to the service inquiries generated by Local Device.

The following features shall be provided by the Supplier:

Search for services by service class

Search for services by service attributes

Service browsing

5.1.1.1.11 Device Information Profile

Device Identification Profile (DIP) has to be supported as well as Profiles/Protocols for necessary lower layers.

Items marked with "x" in AGL column in Table 28 should be supported.

Table 28 : List of DIP Supporting Functions

No.	Feature	Support	AGL
1	SpecificationID	Mandatory	x
2	VendorID	Mandatory	x
3	ProductID	Mandatory	x
4	Version	Mandatory	x
5	PrimaryRecord	Mandatory	x
6	VendorIDSource	Mandatory	x
7	ClientExecutableURL	-	-
8	ServiceDescription	-	-
9	DocumentationURL	-	-

5.1.1.1.12 Bluetooth Smart Ready

Bluetooth Smart Ready shall be supported.

It shall comply with Bluetooth Low Energy standard.

5.1.1.1.13 Generic Object Exchange Profile (GOEP)

The Telephony system shall implement Generic Object Exchange Profile as per the GOEX specification version 2.0 or later.

The Telephony system shall use this profile to facilitate the exchange of binary objects between devices. The usage model shall be Synchronization, File Transfer or Object Push model.

It shall provide following roles:

Server - This is the device that provides an object exchange server to and from which data objects shall be pushed and pulled, respectively.

Client - This is the device that can push or/and pull data object(s) to and from the Server.

The following features shall be provided by the Supplier:

Establishing an object connection

Pushing a data object

Pulling a data object

Performing an action on data objects

Creating and managing a Reliable Object Exchange Connection

5.1.1.1.14 Generic Audio/Video Distribution Profile

The Telephony system shall implement Generic Audio/Video Distribution Profile as per the GAVDP specification version 1.2 or later.

The Telephony system shall use this profile to specify signalling transaction procedures between two devices to set up, terminate, and reconfigure streaming channels.

It shall provide following roles:

Initiator (INT)

Acceptor (ACP)

Following are the feature requirements for this profile:

Connection

Transfer Control

Signalling Control

Security Control

Note: This profile is currently being enhanced to version 1.3. Release date of this version is not yet finalized. The Telephony system shall be able to upgrade to the newer version in the future.

5.1.1.1.15 Bluetooth Diagnostics

5.1.2 Error Management

The Error Management module provides platform error handling mechanisms. This includes detecting system errors that occur after start up to provide a recovery function by localized restart. In addition,

in case of a broad ranged malfunction, Error Management provide quick detection and recovery to issue in a short amount of time.

5.1.2.1 Use Cases

5.1.2.1.1 System Surveillance and Recovery

While using in-car information device, if the whole system or part of the function stops, an immediate error detection and automatic recovery will be needed. For example, when updating the screen while route guidance is on or voice recognition cannot be used, restart the function to try and recover. When an error occurs in the core of a system such as an output communicating middle ware, reboot the whole system to try and recover.

There are several supposed cases for system surveillance such as a case where the system that adopted AGL and monitors by itself or monitored by the system that has not adopted AGL. The AGL Error Management scope includes parts of the system that adopted AGL.

The way of recovery has to be assessed by the status of the system behavior. For example, even if the way to recover the car navigation error might be reboot, the system reboot should not be done when the car navigation is displaying back camera image. Because of these use cases, Error Management should focus on the degree of importance for surveillance list process and the degree should be adjusted by its behavior status.

5.1.2.1.2 Collecting Information

For when the system failure occurred after the launch, the most urgent item is a prompt recovery but what is also a point that is worth noting is to collect the information to specify the cause for its failure. Therefore, gathering information with the minimum recovery time is needed.

With Linux system, memory image dump (core dump) of generally abended process is used. On the other hand, a scale of middleware which is an in- car application is increasing and has come to the point where the time to dump the entire memory image is impermissible. To avoid this, the Error Management function will provide the system to leave the light log.

5.1.2.2 Requirements

Prevent the system failure shutoff and also in case of failure provided the function that judge its status automatically and recover

The Error Management module should support both surveillance of the whole system and each process.

The Error Management module should monitor the memory usage of whole system cyclically. When memory usage exceeds set threshold value, a set action should be done. Cycle, threshold value, action is changeable by AGL user.

Kernel function that requires Error Management surveillance, driver has to send a notification to Error Management when an error occurs. The subjects that sends error notifications are output communication or disk I/O.

Error Management should be able to execute the action after obtaining the error notification by kernel function and the driver. Action should be changeable by AGL user. For example, an error for CAN communication is critical so system restart could be done but USB communication error can be ignored since it may be caused by a compatibility issue between devices.

Error Management should monitor processes for existence or non-existence, when abended it should execute a set action. The set action should be changeable by the AGL user. Termination of resident process is a defect but termination of a temporal behaving process is correct so those two should be able to set separately.

Error Management should monitor the process with a set cycle and when it goes over threshold value, should be able to execute the set action. Cycle, threshold value, action should be changeable by AGL user. The subjects of surveillance are CPU usage and memory usage.

Should be able to vanish process forcibly including subsidiary process

Make the software that works by system have the concept of level importance.

Appropriate recovery depending on the level of importance. The level of importance should be adjustable depending on the status of operation by coordinating with Policy.

The process that detecting an external communication error within the Error Management module and recovering has to be set to complete before external monitoring detects.

The application that is monitored by the Error Management modulehas to be independent as more than one process.

The application that is monitored by the Error Management moduleshould not combine multiple applications to one process. Application's runtime part does not have the structure where multiple applications can be moved by the same process.

Service providing side has to be nondense to the application. For example, the Service providing process such as a software keyboard should not go wrong with the state of App. Such as process crash, exit, etc..

An application has to be nondense to an application. When linking two application one ends suddenly the other will not become abnormal state.

The process that communicates with the external system has to be independent from the other process while recovering that does not include system restart so that it can notify alive towards external side.

When the software that is under the surveillance of RAS can not recover with one restart additional process can be done such as deleting the subject files that were registered beforehand.

The system has to have a structure where overwrite the files that are stored in a pinned file system without destroying them.

When system down occurs (kernel panic), should be able to collect the information need for analyzing.

When making the system down happen intentionally(BUG_ON etc.),make sure to leave a message that can specify the cause.

Both the log which is for debug and can take time to output and the log that leaves minimum log in a short period of time have been equipped and able to select.

In any abnormal cases log output does not lock the system (stand by for spin lock etc.) or system down does not occur (self-destruction on log output process).

Should be able to leave the aberrance occurred in kernel area on the log.

Should be able to select the level of log output.

Should be able to record the aberrance log with the time of occurrence.

Should be able to obtain the information linked to the system resources.

Should be able to leave the information corresponding to core dump in a short period of time.

Both the log which is for debug and can take time to output and the log that leaves minimum log in a short period of time have been equipped and able to select.

As the smallest amount of information, the following information should be left.

- Register information
- Process logical memory map
- Stack dump or back trace from the exceptional place of occurrence
- Time of occurrence
- Information that can specify the occurred process thread (name of an executing file□name of the thread etc.)

- The signal that occurred

Lightweight core dump is a core dump that can set the restrictions below.

- Select the memory mapping category of process executing memory image that targeted for an output.
- Specify the order of an output and output high-priority memory mapping first to prevent dropping the information needed.
- Output only the memory mapping that is linked to the abnormal process (text area). [O]
- Compress the data for each memory mapping category and output up to the fixed maximum size.
- NOTE information of ELF header and program header will not change.

Selectable memory mappings are the following.

- anonymous private mappings
- anonymous shared mappings
- file-backed private mappings
- file-backed shared mappings
- private huge page
- shared huge page

Setting parameters of the output context are the following.

- Memory mapping category which is for an output object can be set.
- The order of outputting memory mapping can be set.

Should be able to leave the log in increments of process. Possible to filter and have a look in increments of process.

Should be able to leave a trace log in increments of process during process crash. Should be able to leave a trace log in increments of process during system running, if necessary.

Should be able to obtain the information related to system resource of process.

There should be a structure to be able to error trace among the whole process in a user space.

5.1.3 Graphics

Graphics subsystem; HMI input, wayland, windowing, etc.

5.1.4 Location Services

Location services includes support for GPS, location, and positioning services including dead reckoning. Time of day support is also included in Location Services since time is a primary output of the GPS receiver.

5.1.4.1 Position

5.1.4.2 Time of Day

With Linux, time adjusting is generally done by using date command or NTP but since in-car device can obtain the accurate time from GPS, GPS time is often used as Abs Time. Because of its advantage where this GPS demand can be done anywhere in the world, it would continue in future. Therefore, we are going to need a structure for adjusting the Linux system time.

Monotonic and Absolute Time Support

As a weak point of GPS, when cold start, it takes a long time to obtain the accurate time.

Because of this, it will not set the right time for booting the system and will adjust it while it's moving. As for in-car device, the demand to make the system boot faster is rather strong and Abs Time can vary while it's working for one of the middle ware applications.

On the other hand, although POSIX API which is used as a standard for Linux, provides the time that has not been effected by the adjusting in case of a simple latency, but for resource latency, some of them can only set with Abs Time. Therefore, in-car Linux needs an API that supports Monotonic Time.

Kernel Time Precision

In-car device needs to support all kinds of communicating system such as CAN. Those communicating system includes the device that needs ms order procedure.

In Linux Kernel space, jiffies are used as mere time. However 1jiffies time differs depending on the CPU architecture and the architecture differs depending on SOC. Because of this, the lowest value for unit of time that AGL environment has to support needs to be decided.

5.1.4.3 Requirements

Should be able to adjust the system time from GPS middle ware.

Adjust the system time after the time is determinate.

GPS middle ware has to have the system where it can implement GPS driver control parts using the plugin (source plugin). Must tolerate proprietary GPS component.

GPS middle source plugin must tolerate proprietary. Source plugin has to be a license that is not imposed a duty to open source. For example, header library’s license that is needed to make Source plugin can not be GPL or LGPL.

When waiting, can use both absolute time and monotonic time

Resource obtaining time out such as mutex, semaphore can use both absolute time and monotonic time.

Resource obtaining time out such as mutex, semaphore can use both absolute time and monotonic time.

System time must be able to use consecutively at least until 2099.

Absolute time has to support leap year and leap seconds.

1 jiffies have to be smaller than 1ms.

Time waiting that involve context switch, must be done with the accuracy over 1ms.

From timer / ISR, can boot tasklet with the accuracy 1ms.

A system has to be able to handle time with at least the accuracy 1ms.

5.1.5 Health Monitoring

Platform monitoring services such as watchdog or active monitoring

5.1.6 IPC

Standard platform interprocess and interprocessor communication mechanism.

5.1.7 Lifecycle Management

Startup, shutdown, state change, etc.

5.1.8 Network Services

Includes standard networking protocols such as TCP/IP via any networking physical layer including Wifi, Bluetooth, or ethernet.

5.1.9 Persistent Storage

Power safe persistent storage

5.1.10 Power Management

Amount of ECUs in the car and their complexity has grown dramatically over last decade. Needs in processing power are constantly growing to catch up with demands of automotive industry. This, in turn has impact on power budget and temperature/heat dissipation characteristic of modern ECUs

In parallel, success of green, electric cars is pushing power budget limits down as never before, in distant future we may see “battle for watts” in automotive electronics. Finding optimal balance between performance and ECU operating modes, frequencies, voltages is also important for overall durability characteristic.

Suspend/resume techniques and retention of the ECU in lower power states now becoming more welcomed over traditional cold boot approaches.

Linux community has been working on power management architecture for many years, it has become a state of art framework and set of components that addresses needs not only consumer electronics industry, but also industrial automation, security, etc.)

5.1.10.1 Requirements

AGL kernel shall allow switching between active and suspend states. Exact definition of suspend states is platform/architecture-specific (e.g. ”suspend to memory”, ”suspend to disk” /”hibernate” correspond to S3 and S4 in ACPI terminology)

Kernel and peripheral device drivers shall not be affected by suspend/resume transitions.

AGL kernel shall provide sufficient APIs for application to control active/suspend state transitions and receive appropriate events/notifications. Kernel should not initiate power state transitions if no requests provided from applications.

Detailed definition of steps/actions required for suspend/resume sequence is out of the scope of this specification (it is also platform-dependent).

AGL kernel for SMP configurations shall allow enabling/disabling of individual cores (or group of cores) (NOTE: on some platforms/architectures enabling/disabling may be achieved by putting core in one of its low power states)

AGL kernel shall only provide mechanism for applications to request enabling/disabling particular cores from SMP group.

AGL kernel shall support CPU frequency and voltage scaling. Exact definition of operating points (table of frequencies/voltages allowed by hardware) is platform/architecture-specific (moreover, some of operating points may be omitted/ignored in AGL kernel as their impact on power budget insignificant)

Kernel and peripheral device drivers shall not be affected by CPU frequency and voltage scaling

Only application-defined policies shall be allowed for CPU frequency and voltage scaling. Default in-kernel governors/policies (e.g. on-demand or performance) shall not be used and they may have negative impact on overall system performance/predictability

AGL kernel shall allow switching between active and idle states. Exact definition of idle states is platform/architecture-specific (e.g. C0..C4 in ACPI terminology or WFI+… for ARM)

Kernel and peripheral device drivers shall not be affected entering/leaving one of idle states

Only application-defined policies shall be allowed for CPU Idle

AGL kernel shall support run-time power management of I/O (peripheral) devices

AGL kernel shall support I/O (peripheral) device voltage and frequency scaling

5.1.11 Resource Management

Resource and device management.

Resource Management shall provide an interface to be used for informing status of a resource request by the Resource Manager.

5.1.12 Telephony Services

5.1.12.1 Requirements

5.1.12.1.1 Telephony variants

Purpose: To define the variants of Telephony

Requirement:

There will be 2 variants of phone system.

Variant 1: Front User only Telephony.

Variant 2: Rear and Front Telephony.

All variants will have Bluetooth capability. The feature will be configurable so that the feature can be disabled via car configuration.

5.1.13 Wi-Fi

This Wi-Fi subsystem controls registration, connection management, and device information management between a wireless LAN device and infotainment system.
Necessary Wi-Fi specification in automotive use case is defined here.

5.1.13.1 Use Cases

5.1.13.1.1 Construct WiFi Network

In-Vehicle Infotainment systems constructs 3 types of Wi-Fi networks.

a. STA

In-Vehicle Infotainment system acts as a STA (Station) and connects to an external network via an Access Point.

It also connects to Access Points which support Wi-Fi Hotspot.

b. AP

In-Vehicle Infotainment system acts as an AP (Access Point) and connects multiple Wi-Fi devices with an external network.

It also connects Wi-Fi devices which support Wi-Fi Hotspot.

c. P2P

In-Vehicle Infotainment system and Wi-Fi device makes P2P (Peer to Peer) connection using Wi-Fi Direct.

5.1.13.1.2 Miracast

In-Vehicle Infotainment system and Wi-Fi device shares a display using Miracast.-(a)

They are also remotely operated to a Wi-Fi device from the infotainment system, or vice versa, by using UIBC (User Interface Back Channel).-(b)

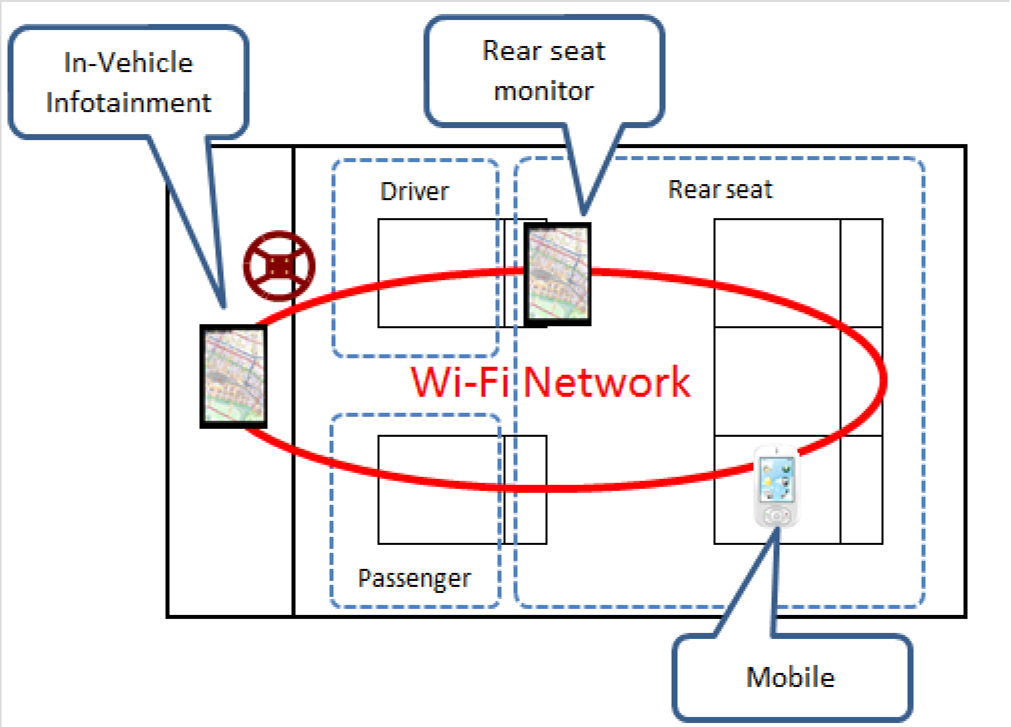


Figure 8-29 : Overview of Miracast

a. Shared Displayed Content

Use case examples of shared displayed content are:

- A passenger on the passenger seat views the multimedia content played on Wi-Fi Device (e.g. Mobile) on In-Vehicle Infotainment system.
- A rear seat passenger views the multimedia content played on In-Vehicle Infotainment system on Wi-Fi Device(e.g. Rear seat monitor).

b. Remote Operation

Use case examples of remote operation are:

- A passenger on the passenger seat plays the multimedia content stored in Wi-Fi Device (e.g. Mobile) by operating In-Vehicle Infotainment system.
- A passenger on the rear seat controls air conditioner functionality in In-Vehicle Infotainment system by operating a Wi-Fi Device (e.g. Mobile).
- While the vehicle is in motion, a passenger on the rear seat controls the navigation functionality in a passenger on the rear seat controls by operating a Wi-Fi Device(e.g. Mobile).

5.1.13.1.3 DLNA

In-Vehicle Infotainment system connects with a DLNA device via Wi-Fi.

5.1.13.2 Requirements

5.1.13.2.1 Security

The WiFi module shall support security standard WEP.

It shall support 40 bit WEP encryption method.

It shall support 104 bit WEP encryption method.

It shall support security standard WPA Personal.

It shall support TKIP encryption method.

It shall support CCMP encryption method.

It shall support security standard WPA2 Personal.

It shall support TKIP encryption method.

It shall support CCMP encryption method.

It shall support security standard WPA Enterprise.

It shall support TKIP encryption method.

It shall support CCMP encryption method.

It shall support security standard WPA2 Enterprise.

It shall support TKIP encryption method.

It shall support CCMP encryption method.

5.1.13.2.2 Simple Configuration

It shall comply with WPS (Wi-Fi Protected Setup) standard.

It shall be able to perform connection with PIN (Personal Identification Number) method.

It shall support Configuration Method for Display.

It shall support Configuration Method for Keypad.

It shall be able to perform connection with PBC (Push button configuration) method.

It shall support Configuration Method for PushButton.

It shall be able to perform connection with NFC (Near Field Communication) method.

5.1.13.2.3 QoS

It shall comply with WMM (Wi-Fi Multimedia) standard.

It shall comply with WMM-PS (Wireless Multimedia Power Save) standard.

5.1.13.2.4 STA

The In-Vehicle system shall be able to function as a STA (Non-AP Station).

5.1.13.2.5 AP

The In-Vehicle system shall be able to function as an AP (Access Point).

5.1.13.2.6 WiFi Direct

It shall comply with Wi-Fi Direct standard.

It shall support the WiFi Direct functions as listed in Table 29.

Table 29 : List of Wi-Fi Direct Supporting Functions

No.	Feature		(Reference) Support in Wi-Fi Direct
1	P2P Provision Discovery	□	Mandatory
2	P2P Device Discovery	Scan Phase	Mandatory
3	□	Find Phase	Mandatory
4	P2P GO Negotiation	□	Mandatory
5	P2P Service Discovery	□	Option
6	P2P Invitation	Temporary P2P Group	Option
7	□	Persistent P2P Group	Option
8	Persistent P2P Group / Persistent Reconnect		Option
9	Intra-BSS Distribution	□	Option
10	Concurrent Operation	□	Option
11	P2P Service Discovery	UPnP	Option
12	□	Bonjour	Option
13	□	Wi-Fi Display	Option
14	□	WS-Discovery	Option
15	□	Vendor specific	Option

5.1.13.2.7 Miracast

It shall comply with Miracast standard.

It shall support the Miracast functions identified in Table 30.

Table 30 : List of Miracast Supporting Functions

□No.	Feature	□	(Reference) Support in Miracast
1	WFD Device type	WFD Source	Mandatory
2	□	Primary Sink	Mandatory
3	□	Dual-role possible	Option
4	WFD Service Discovery	□	Option
5	WFD connection establishment with Wi-Fi P2P		Mandatory
6	WFD connection establishment with Wi-Fi TDLS		Option
7	Persistent WFD Group	via Wi-Fi P2P	Option
8	□	via TDLS	Option
9	WFD Capability Negotiation (RTSP)		Mandatory
10	WFD Session Establishment (RTSP)		Mandatory

11	AV Streaming and Control (MPEG-TS/RTP/RTSP)		Mandatory
12	WFD Standby (RTP/RTSP)		Option
13	Video CODEC formats		Option
14	Audio CODEC formats		Option
15	UIBC	Generic	Option
16		HIDC	Option

5.1.13.2.8 WiFi Hotspot

It shall comply with Wi-Fi Hotspot standard.

In-Vehicle system which acts as an a STA(Non-AP Station)shall be able to connect with Hotspot service.

In-Vehicle system which acts as an AP (Access Point) shall be able to provide Hotspot service.

5.1.13.2.9 DLNA via WiFi

The In-Vehicle system shall be able to connect with DLNA devices via Wi-Fi.

5.1.14 Window System

A window system is a software component that facilitates an implementation of graphical user interface. A window system is responsible for managing display devices, Graphics Processing Units (GPUs), input devices, and graphics memory. A window system works with the software component named window manager that is responsible for a layout management of windows, and a routing of user interactions.

5.2 Automotive Services

Automotive Services Layer contains services that are not found in a typical Linux distribution but contains services specialized for automotive applications.

5.2.1 Audio Services

BTBF, equilization, mult-zone audio control, etc.

5.2.2 Camera Services

Standard interface to vehicle mounted cameras; backup camera, side and front cameras, etc.

5.2.3 Configuration Services

Service for storing configuration parameters.

5.2.4 Diagnostic Services

Diagnostic services.

(This is automotive diagnostics such as storing and retrieving DTC.)

5.2.5 Multimedia Services

CD, DVD, Blu-Ray, MP3, etc.

(Factor out metadata into separate component.)

5.2.5.1 Media Player

In-vehicle multimedia system shall provide rich and robust user-experience that includes not just support of multiple audio-video formats, but also variety of input and output audio/video devices, both static and dynamically pluggable. In contrast to mobile or desktop applications, there is normally more than one consumer of multimedia content in a car, with front- and rear-seat passengers as well as driver all having independent requirements.

The following requirements are considered essential for in-vehicle multimedia system:

- Supported multimedia formats shall correspond to major end-user expectations, i.e. the ones encountered in mobile and desktop world.
- Multiple audio / video sources and sinks, both static (i.e. always existing in the system) and dynamic (i.e. appearing and disappearing when user connects a Bluetooth headset or establishes a network connection.)
- Multiple independent consumers of multimedia data and globally configurable routing of audio / video processing chains.

Latency requirements of audio/video processing may also vary depending on a type of the data processed; e.g. data from rear-view camera shall be decoded and visualized "instantly" in comparison to a movie clip displayed on rear-passenger monitor, voice notification from navigation software shall not be delayed significantly, speech data passed to and from Bluetooth headset during phone conversation shall have reasonably bounded latencies and so on.

It is considered that multimedia system may consist of multiple processing units, and therefore processing load balancing mechanism shall be present. Mechanisms of audio/video processing offloading to dedicated processing units (hardware acceleration) shall be provisioned, with particular implementation freedom left for a silicon vendor.

The following requirements formalize these considerations.

5.2.5.2 Requirements

5.2.5.2.1 Media Containers

AGL shall provide an API that allows handling of various media data within the system. This includes audio/video playback and recording as well as media streaming over the network. It shall be possible to run multiple media streams in parallel for all IVI users, with configurable input/output devices routing. Multimedia framework does not necessarily need to be isolated from application (that is, it may run in the same address space as application), however it shall be guaranteed that independent applications using the framework are isolated from each other.

AGL shall provide support for extraction from media containers streams other than audio-visual, for example subtitles. Application shall be able to retrieve timing information as well as stream identification data from media container.

AGL shall provide support for major network streaming protocols such as:

- HTTP
- RTPS
- Digital Radio (DAB)
- DigitalTV (DVB-T) etc.

It shall be possible to extend the set of supported streaming protocols in accordance with system requirements.

AGL shall provide a mechanism to utilize available hardware accelerators to offload computationally extensive processing to specialized units in vendor-specific way. Such extension, if available, shall be transparent to the applications.

Lip Synch must be implemented as plug-in software for Multimedia Framework.

AGL shall provide a mechanism to automatically detect type of media data contained in the source file, and to instantiate all required components to organize data processing without intervention of the application. It shall be, however, possible for application to control this process if it is essential for its functionality. Example of such intervention would be selection of particular audio track (in user-chosen language) or selection of particular video stream from multiple choices.

AGL shall provide an API to control execution of audio/video processing chain, specifically shall support the following functionality:

- Selection of data source and destination (files, devices, network resources)
- Pausing/resuming of multimedia streams
- Rewinding in forward and reverse directions (for playback)
- Audio volume control on per-stream basis
- Retrieval of current stream position (timestamp)

- Notifications on error conditions preventing multimedia stream processing

AGL shall provide a mechanism to specify routing of input and output devices that are involved into multimedia data processing. In particular, for playback scenario it shall be possible to specify where audio and video data is rendered, and for recording scenario it shall be possible to specify capturing source. It shall be possible to organize broadcasting of decoded raw audio/video streams to multiple renderers as well.

AGL shall include a dedicated sound server that simplifies routing, mixing, post-processing and synchronization of raw PCM audio streams. Specifically, the following functionality is expected:

- Support for multiple audio sources and audio sinks with arbitrary (configurable) routing.
- Per-stream volume and audio effects control.
- Resampling and format conversion (e.g. channels downmixing, sample width conversion).
- Sample-accurate streams synchronization (e.g. for echo-cancellation purpose).
- Mixing and broadcasting of the audio streams.

AGL shall provide a mechanism to control sound server configuration in run-time, that is, to specify the rules and policies defining system response to external events like adding or removing of new audio device (e.g. Bluetooth headset connection), receiving of the phone call, emergency system alarm output and so on.

AGL shall provide support for major multimedia containers, such as:

- MPEG2-TS/PS (ISO/IEC 13818-1)
- MP4 (MPEG-4 Part 14, ISO/IEC 14496-14:2003)

It shall be possible to extend the set of supported multimedia formats in accordance with system requirements.

It must be possible to extend AGL to support additional optional multimedia containers such as:

- OGG (RFC 3533)
- 3GPP (ISO/IEC 14496-12)
- etc

5.2.5.2.2 Media Audio Codecs

AGL shall provide support for major audio codecs, such as:

- MP3 (MPEG-1/2 Audio Layer-3, ISO/IEC 11172-3, ISO/IEC 13818-3)
- AAC (ISO/IEC 13818-7, ISO/IEC 14496-3)

It shall be possible to extend the set of supported audio codecs in accordance with system requirements.

It must be possible to extend AGL to support additional audio codecs, such as:

- VORBIS (http://xiph.org/vorbis/)
- Windows Media Audio
- etc.

5.2.5.2.3 Media Video Codecs

AGL shall provide support for major video codecs, such as:

- MPEG-2 (ISO/IEC 13818-2)
- MPEG-4 Part 2 (ISO/IEC 14496-2)
- H.264 (MPEG-4 Part10, ISO/IEC 14496-10, ITU-T H.264)

It shall be possible to extend the set of supported video codecs in accordance with system requirements.

It must be possible to extend AGL to support additional video codecs, such as:

- Theora (www.theora.org)
- Windows Media Video
- etc

5.2.5.2.4 Image File Formats

The system shall be able to perform all required operations on viewing of Image in BMP, up to 32 bit true colour.

Compression formats

- RLE 8 bits/pixel
- RLE 4 bits/pixel
- Bit field or Huffman 1D compression
- JPEG or RLE-24
- PNG

The system shall be able to perform all required operations on Viewing of Image in JPEG/JPEG 2000

The system shall be able to perform all required operations on viewing of Image in JPEG XR/HD, including Exchangeable Image File Format (EXIF) format.

The system shall implement the ability to perform all required operations on Viewing of Image in PNG, including transparency

The system shall be able to perform all required operations on viewing of Image in GIF 87a and enhanced version 89a and also animation in GIFF images.

The system shall be able to perform all required operations on viewing images in TIFF format.

The system shall implement the ability to perform all required operations on viewing of Image in WBMP format.

The system shall implement the ability to perform all required operations on viewing of Image in WBMP format.

5.2.6 Navigation Services

Navigation engine

5.2.7 PIM

Personal Information Manager; calendar, appointments, reminders, etc.

5.2.8 Smartphone Link

This section describes regarding Smartphone link. Smartphone Link is the technology which realizes that video and audio streaming play which data from smartphone. And touch operation is possible to share between IVI and smartphone. MirrorLink, Miracast, SmartDeviceLink and AirPlay are technologies that realize Smartphone Link. By this technology, it is possible to use smartphone content (map, music, browser...) by IVI.

Figure 8-30 shows the system structure of the Smartphone Link.

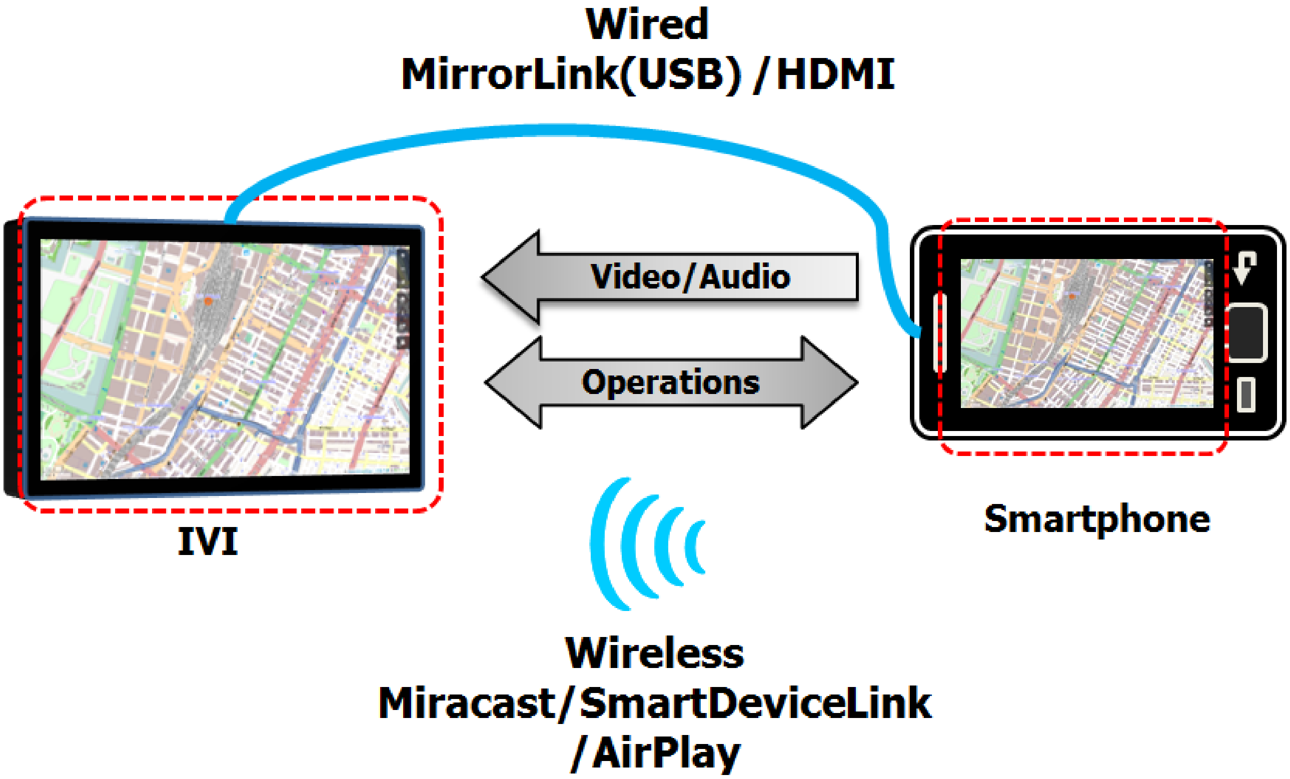


Figure: 8-30

AGL defines following requirements of Smartphone link.

1. The screen of smartphone shall be mirrored to IVI.

2. The sound of smartphone shall be linked to IVI.

3. The sound shall be synchronized with the screen.

4. IVI should operate smartphone.

5. The response time of operations from IVI should be less than 200ms.

6. If connection between smart phone and ivi was disconnected by external factor, then should inform the "disconnection" to a user and return to the normal state.

This document describes “Miracast” and “SmartDeviceLink” from the reference of Smartphone link.

5.2.8.1 Miracast

This section describes requirements regarding Smartphone link (Miracast).

Miracast is the display transfer technology using wireless connection which was defined by Wi-Fi Alliance. Send screen data from source device to sink device and it realize display sharing between source device and sink device.

Following figure (Figure: 8□31) shows the system structure of Miracast.

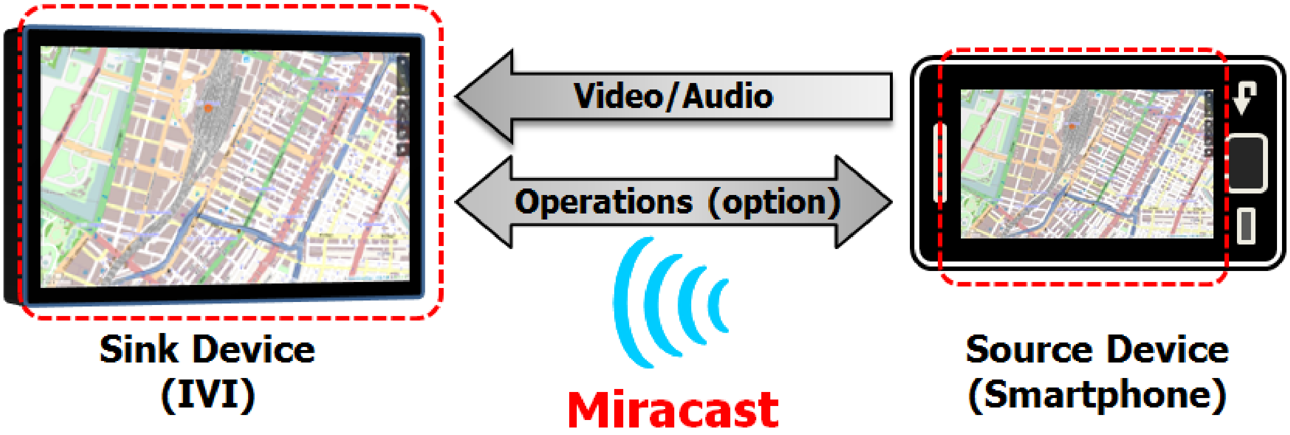


Figure: 8-31

Follow reference documents to support Miracast if there was no description of this section.

References

[1] Wi-Fi Display Technical Specification Version 1.0.0

[2] W-Fi Peer-to-Peer (P2P) Technical Specification Version 1.2

[3] High-bandwidth Digital Content Protection System Interface Independent Adaption Revision 2.2

[4] DCP (Digital Content Protection) http://www.digital-cp.com/

AGL provide display sharing technology between Smartphone and IVI system using Miracast.

AGL defines following Smartphone link (Miracast) specification as Table 8□14.

No	Requires	Description
SPL.1.1	WFD Topology	Define role of Miracast
SPL.1.2	Connection Topology	Define connection condition between a smartphone and an IVI
SPL.1.2.1	P2P Topology	Define connection method of P2P (Wi-Fi Direct).
SPL.1.2.2	Wi-Fi Frequency	Define Wi-Fi frequency
SPL.1.3	Video Format	Define Video format
SPL.1.4	Audio Format	Define Audio format
SPL.1.5	Session Control	Define Miracast session state
SPL.1.6	Link Content Protection	Define content protection function required for implementing Miracast
SPL.1.7	Resource Management	Define resource management

SPL.1.8	Fail-safe Control	Define Fail-safe control

Table 8-14: Smartphone Link (Miracast) Requirements

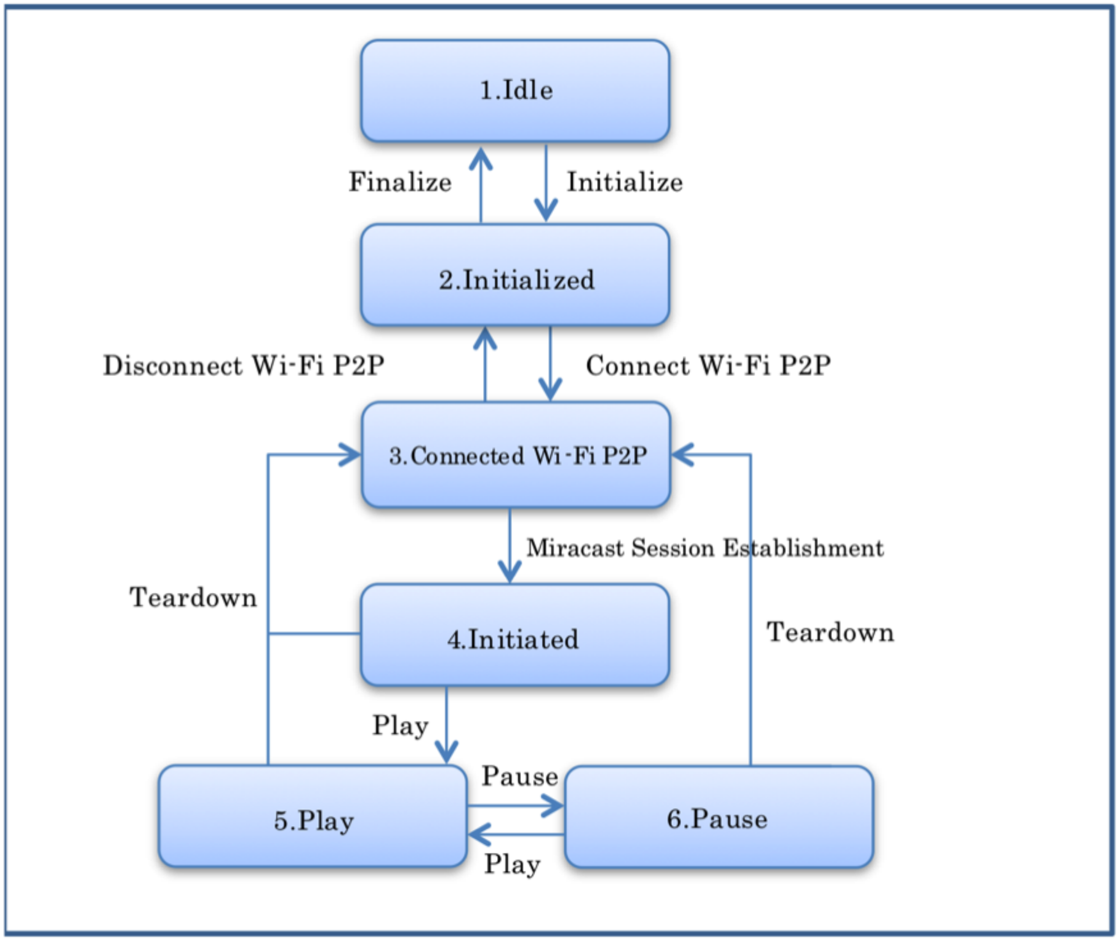


Figure: 8-32 State Change Diagram

The states of Smartphone link (Miracast) is defined in Table 8-32.

Table 8-32: State Definition

No.	State	Description
1	Idle	Smartphone link (Miracast) function is not initialized.
2	Initialized	Smartphone link (Miracast) function is initialized and waiting for Wi-Fi P2P connection from source device.
3	Connected Wi-Fi P2P	Established Wi-Fi P2P connection with source device.
4	Initiated	Smartphone link (Miracast) session is established.
5	Play	Streaming the audio and video content from source device to sink device.
6	Pause	Paused the streaming of audio and video content from source divide to sink device.

5.2.8.2 Smart Device Link

"Smart Device Link", aka "SDL", is template based approach of smartphone link capability. Application itself is in a mobile phone, however, HMI is provided by IVI system. This approach makes it possible to apply IVI adapted user experience, such as larger button to prevent driver's distraction and voice recognition.

That means, application requests to IVI system, then IVI system will respond by using remote procedure calls. Application's HMI will be rendered by IVI system by using IVI's HMI framework and/or policy, though all the application's logic is contained in mobile phone.

SDL provides more suitable HMI for IVI rather than mirroring type approach, however, mobile phone's application must support SDL capability. In other words, only SDL supported applications can be launched.

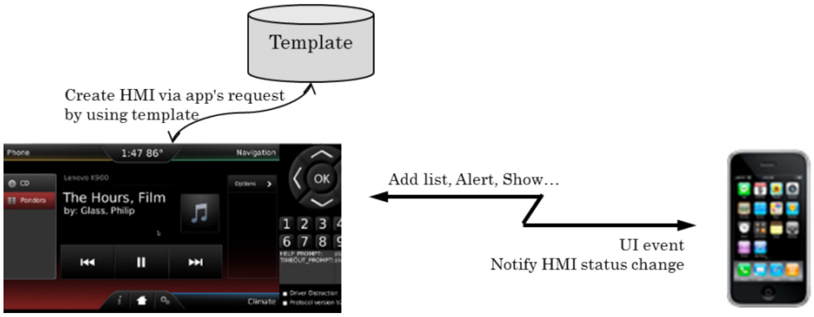


Figure 8-33 : SDL overview

5.2.8.3 Requirements

5.2.8.3.1 Miracast

System must provide a capability of Miracast as smartphone link function.

- Support WFD Primary Sink and support MPEG2-TS(Video, Audio) streaming play which from Source Device□Smartphone□.
- Supporting WFD Source is an option.
- Support customize function using “Miracast setting file” which used for negotiation (*1) source device and sink device (*1. Video format, audio format and other parameters).
- Screen data which from Smartphone may not support Drivers Destruction, therefore take measures to Drivers Destruction. (e.g. Disable Miracast during vehicle speed over 5Km/H)

- Support Wi-Fi P2P connection.
- Follow reference [1] and reference [2] to support Wi-Fi P2P function, parameters in Miracast connection and so on if there was no description of this section.
- Wi-Fi TDLS connection is an option.
- AGL do not define confliction specification regarding Wi-Fi connection. (e.g. User select Wi-Fi P2P connect ion during accessing Wi-Fi connection.)
- AGL do not define confliction specification regarding Sink device operation when receive

connection request from Source device. (e.g. Connect automatically, ask user for confirmation)

- Support P2P Group Owner and P2P client as the topology of Wi-Fi P2P connection.
- Support DHCP server and DHCP client for TCP/IP seamless connection after P2P connection established.

- Support 2.4GHz band for the frequency of Wi-Fi P2P connection.
- Supporting 5GHz band is an option, but support DFS (Dynamic Frequency Selection) function if support 5GHz band.

- Follow reference [1] for Video Codec.
- Support follow format for Video Resolution and Frame rate.
  - 640*480□VGA□□Progressive 60 fps
  - 1280*720□HD□Progressive 30 fps

Regarding Video resolution and Frame rate, other formats are an option.

- Support follow format for Audio.
  - LPCM 48ksps 16bit 2ch
  - AAC 48ksps 16bit 2ch

Regarding Audio Format, other formats are an option.

When the state changes "Pause", take measures to give notice of pause for user. (e.g. pop-up notification)

Screen data which from Smartphone may be protected by content protection, therefore support content protection function.

- AGL recommend HDCP function which described reference [2], [3]. But AGL do not define HDCP function. Each vendor should support content protection function as for vendor’s own reason.

- Support both encryption cases if support HDCP function.
  - Case1 Videos data encryption
  - Case2 Both video and audio encryption

Take notice that it is necessary to satisfy security requirements specified according to DCP.(reference [4])

- Miracast must support interruption by other function. If some high priority event occurs, then Miracast release screen and audio resources for the event.
- It is selectable how to deal Miracast session. (Standby Miracast session or close Miracast session.)

- Support a notification to a user and returning to the normal state, if following events happen.
  - Failed to Wi-Fi connection
  - Failed to establish Miracast session
  - Wi-Fi link loss on Miracast
  - Break Miracast connection from smartphone

5.2.8.3.2 Smart Device Link

System must provide a capability of Smart Device Link as smartphone link function.

System must provide a mechanism to render HMI of SDL according to template.

System must provide a mechanism to enable user interface regarding SDL by using touch panel device of IVI device.

System must provide a mechanism to enable user interface regarding SDL by using voice recognition of IVI system.

System must provide a mechanism to link Android device regarding SDL capability. Connectivity method must be supported Bluetooth and/or Wi-Fi.

System must provide a mechanism to link iPhone device regarding SDL capability. Connectivity method must be supported Bluetooth and/or Wi-Fi.

5.2.9 Speech Services

The Speech Services module provides voice recognition and synthesis for AGL applications.

AGL system voice framework must be able to record and interpret voice commands

AGL system voice framework must be able to convert text to synthesized speech

5.2.10 Tuner Services

The Tuner Services module provides a mechanism that allows different tuner types to plug into the same API regardless of the receiver type. Support for AM/FM, HD Radio, SDARS, DAB, DRM, TV Tuners etc is provided. The Tuner Services module shall allow multiple tuners to be present in the same system and allow its clients to address each tuner in the system independently.

5.2.10.1 Receivers

The Receivers module of Automotive Grade Linux may control different receiver types including AM, FM, Hybrid Digital (HD) Radio, SDARS, and DAB tuners. The module may access any number of different tuners. For all tuner types the module supports accessing station data from the tuner, changing the receiver frequency or station and reading station metadata about current content.

The Receivers module shall provide a mechanism that allows different tuner types to plug into the same API regardless of the receiver type.

The Receivers module shall allow multiple receivers to be present in the same system and allow its clients to address each receiver in the system independently.

5.2.10.1.1 HD Radio

HD Radio is a proprietary In-Band on Channel (IBOC) system created and owned by Ibiquity. An HD radio receives analog AM/FM signals and can also use digital information in a subband to provide additional stations and/or enhance the audio quality of the main station. When the receiver is decoding digital data for AM/FM playback it is commonly thought of as HD Radio. The HD Radio system architecture shall conform to the broadcast system design proposed by the iBiquity Digital Corporation detailed in RX_SSFD_5029. Both the HD hardware and functional design shall meet all iBiquity Digital specifications, and satisfy the Type Approval specified by iBiquity Digital.

The IBOC hardware is assumed to have three modes which will be used to describe the requirements in this section.

1) AM - radio is decoding an over the air AM station.

2) FM - radio is decoding an over the air FM station.

3) HD - radio is decoding an AM or FM station using the subband for the over the air station. Each requirement may refer to AM and/or FM and/or HD to specify the modes the requirement is applicable to.

AM/FM/HD system shall be able to enable/disable the HD radio reception and present the status to the system.

AM/FM/HD tuner shall be able to tune to a specified frequency and report the result of the tuning process. The possible results are, Tuning successful and Tuning unsuccessful. If Tuning successful event is notified by the tuner, it shall play the audio through the selected audio output. If tuner notifies the Tuning unsuccessful event, the system shall inform that "No Reception" is available in that specific channel.

AM system shall extract following parameters from a successfully tuned channel and present to the system, which shall be added in the station database.

- Frequency
- Mono/Stereo

FM system shall extract following parameters from a successfully tuned channel and present to the system, which shall be added in the station database.

- Frequency
- PI Code (RDS only)
- PTY (RDS only)
- Radio Text (RDS only)
- PS Name (RDS only)
- Category (RDS only)
- Mono/Stereo

HD system shall extract following parameters from a successfully tuned channel and present to the system, which shall be added in the station database.

- Frequency
- PTY
- No of HD channels available
- Radio Text
- Channel Name
- Category
- Bit rate
- Station Logo
- Artist Experience

The System shall allow the tuned frequency to be incremented or decremented.

The System shall be able to tune to the next/previous valid station as determined by signal strength.

AM/FM/HD system shall be able to abort Seek Up/Down operations.

FM/HD system shall be able to set the stop sensitivity for seek over FM band and shall be possible to adjust by software.

- Range: 15 – 40 dBµV
- Step: 1 dBµV
- Default: 20 dBµV
- Other parameters like multipath shall be possible to use for determining Stop sensitivity level. TBD, Supplier to suggest solution.

AM/HD system shall be able to set the stop sensitivity for seek over AM band and shall be possible to adjust by software.

- Range: 20 – 40 dBµV
- Step: 1 dBµV
- Default: 34 dBµV
- It shall be possible to have different setting depending on atmospheric conditions (e.g. different for night and day).

The system shall be able to switch between AM and FM bands.

HD system shall be able to extract the Station Information Service (SIS) Short Name from the SIS Protocol Data Unit (PDU) on the Primary IBOC Data Service (PIDS) logical channel and present to the system. The implementation of SIS Short Name feature shall be in compliance with iBiquity Digital specification "HD Radio™ Air Interface Design Description Station Information Service Transport".

HD system shall be able to extract the Station Information Service (SIS) Long Name from the SIS Protocol Data Unit (PDU) on the Primary IBOC Data Service (PIDS) logical channel and present to the system. The implementation of SIS Long Name feature shall be in compliance with iBiquity Digital specification "HD Radio™ Air Interface Design Description Station Information Service Transport".

HD system shall indicate the HD channel number of current tuned channel. It shall be 1 to 8.

HD system shall extract the following PAD data from audio stream and present to the system.

- Song
- Artist
- Album
- Genre
- Comments
- Commercial
- Reference Identifier

The system implementation shall be in compliance with iBiquity Digital HD radio specification "HD Radio Air Interface Design Description - Program Service Data Rev. C"

FM/HD system shall be able to receive and extract the RDS/RBDS data and present to the system. The system implementation shall be in compliance with "BS EN 62106:2009, Specification of the radio data system (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 MHz to 108,0 MHz".

FM/HD system shall be able to enable/disable RDS/RBDS. When RDS/RBDS is enabled/disabled the system shall indicate this.

FM/HD system shall be able to enable/disable the radio text display.

FM/HD system shall present the Alternative Frequency (AF) setting status to the system.

FM/HD system shall be able to enable/disable alternative frequency switching.

FM/HD system shall be able to notify the system when an Emergency Alert Interrupt is received.

FM/HD system shall be able to skip the Emergency Alert when it is on-air.

FM/HD system shall be able to notify the system when Emergency Alert Interrupt is received through RDS.

FM/HD system shall be able to cancel the PTY31 interrupt notification.

FM/HD system shall be able to enable/disable the Traffic Announcement reception.

FM/HD system shall present the status of the FM traffic announcement to the system.

FM/HD system shall be able to skip the FM traffic announcement when it is on-air.

FM/HD system shall be able to enable/disable regionalisation.

FM/HD system shall be able to enable/disable the Traffic Message Channel (TMC) reception.

FM/HD system shall be able to enable/disable the Transport Protocol Expert Group (TPEG) reception.

FM/HD system shall be able to receive the traffic updates from the Japanese traffic channels.

FM/HD system shall be able to enable/disable the News announcement reception.

FM/HD system shall be able to skip the News when being broadcast.

HD system shall decode PNG images which shall be in compliance with HD Design specification.

HD system shall be able to decode the channel icon PNG images and present to the system.

AM/FM/HD system shall be able to mute the audio output.

AM/FM/HD system shall be able to un-mute the audio output.

HD system shall extract the album name, artist name, track number from the audio stream and present to the system.

The feature will store the data of a tagged song in non-volatile memory within the IMC. The feature will be able to store at least 50 tags.

5.2.10.1.1.1 Configuration

AM/FM/HD system shall be able to configure the frequency band through local configuration file.

AM/FM/HD system shall be able to configure the step frequency through local configuration file.

AM/FM/HD system shall be able to configure the seek stop level threshold through local configuration file.

5.2.10.1.2 Database Requirements

AM/FM/HD system shall require a database to store the channel list information which contains the following attributes:

- Frequency
- PTY (FM & HD only)
- Channel name (FM & HD only)
- Channel icon (HD Only)
- Category (FM & HD only)

AM/FM/HD system shall be able to update the channel list database based on the following conditions:

- New channel is found
- Existing channel disappears
- Channel list update shall not create any inconsistency on the current channel list database.

AM/FM/HD system shall sort the channel list database based on the channel name, and present to the system.

AM/FM/HD system shall sort the channel list database based on the ascending order of the frequency, and present to the system.

FM/HD system shall sort the channel list database based on the PTY (Program Type) category, and present to the system.

AM/FM/HD system shall create favourite station database which consists of the following information:

- Station name (FM and HD only)
- Frequency
- Status of HD (HD, HD1, HD2)
- HD SIS (HD only)

AM/FM/HD system shall be able to update the database based on following conditions:

- Favourite station changed
- Favourite station is removed
- New favourite is added

5.2.11 Vehicle Bus / Vehicle Info Control

Vehicle Info Control (VIC) provides a capability to access to various vehicle properties from applications and/or other middleware. Standardized interfaces are provided to vehicle CAN, and LIN bus. Figure 7-27 describes overall architecture of Vehicle Info Control. The main purpose of VIC is to provide API to application and/or middleware. Vehicle Info Control has four main functions.

- Vehicle Data Processing
- Communication between ECUs
- Vehicle Data Upload

- Simulator

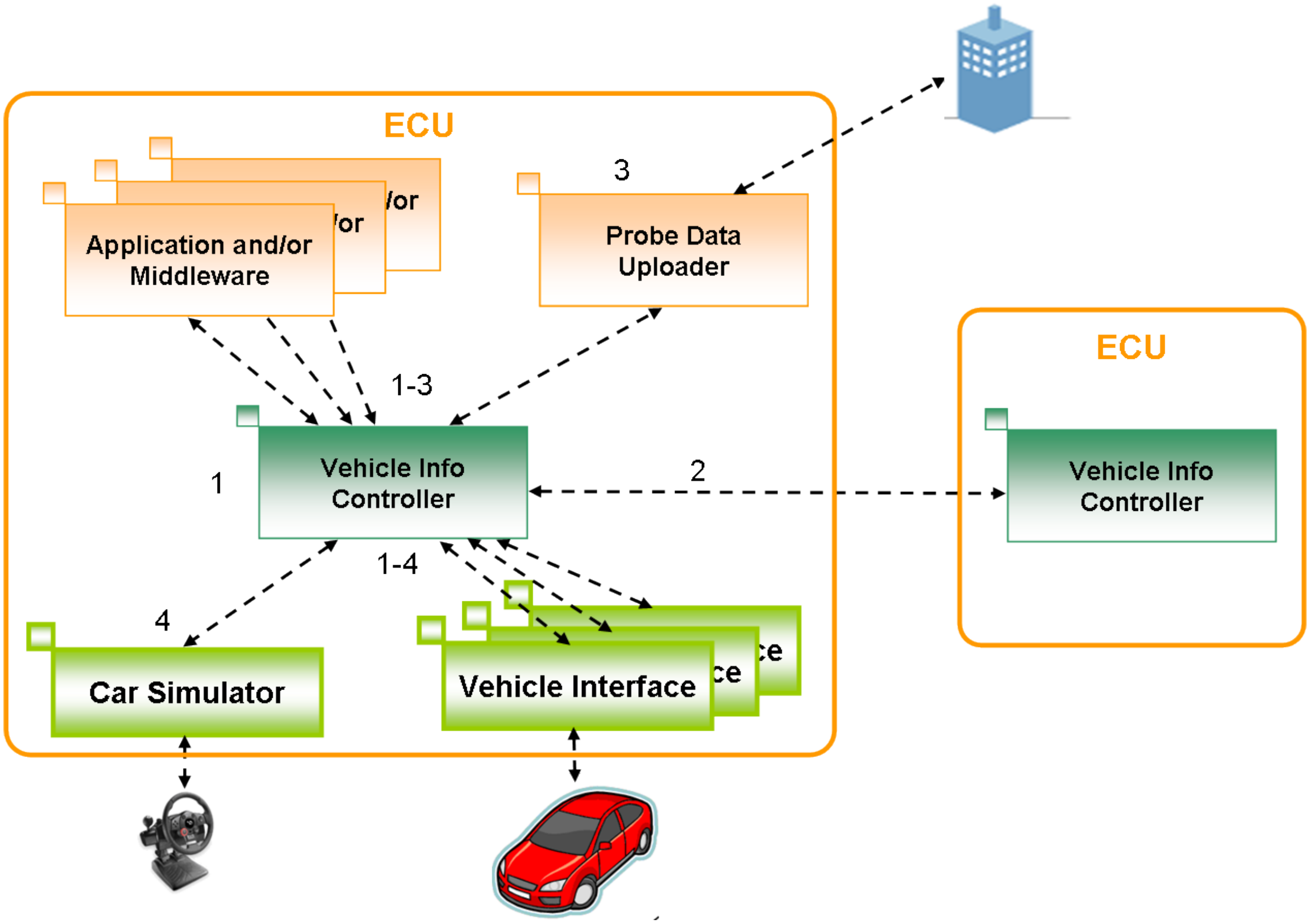


Figure 7-27 : Overview of Vehicle Info Control

5.2.11.1 Vehicle Data Processing

Vehicle data is the information about the vehicle itself, and the information in cars (for example, personal information on a driver, etc.). VIC deals with all the information which application and/or middleware need within vehicles. The following data is contained in these.

- Vehicle information about the vehicles itself, such as speed, a shift position,□temperature
- User Information, such as a name, taste, etc. of a driver
- The operation history of a driver

- The operation state of the vehicles which middleware determined based on vehicle conditions, such as speed and day and night

Vehicles data processing consists of the following functional elements further.

(1) Abstraction of Vehicles Data

In VIC, all vehicles data is treated as abstract data. it concerns and comes out of this to the kind of car, or the country of the destination. For example, though speed is detected at the revolving speed of the wheel, in VIC, vehicles data is abstracted and treated at speed and it provides for application and/or middleware. Thereby, application and/or middleware can treat the vehicles data of the same implications and the same unit.

(2) Maintenance of Vehicles Data

Each abstracted vehicles data is held. The vehicles data to hold is a current value and the past value (history).

(3) Application / Middleware Interface (API)

The accessing function of the vehicles data from application and/or middleware is offered as API. Acquisition of the current value of vehicles data or the past history, a setup of vehicles data, and the change notice function of vehicles data are included in this. However, each vehicles data restricts the application and/or middleware which can be accessed according to the importance (access control).

(4) Vehicles Interface

It is a function for managing the various data of vehicles of in-vehicle networks, such as CAN and FlexRay, etc. The component in which the exchange with actual vehicles performs the exchange with vehicles by a vehicle type since it is various is not included in requirements. However, the correspondence procedure of it and VIC is specified. It assumes that two or more Vehicle Interface is prepared depending on a communication method with vehicles, etc. In addition, the vehicles data which can be accessed for every Vehicles Interface is restricted.

5.2.11.2 Communications between ECUs

When a system consists of two or more ECUs, the vehicles data managed by ECU other than ECU in which application and/or middleware are working shall also be treated. For this reason, vehicle information processing communicates with it of other ECUs. Thereby, application and/or middleware can be treated, without caring about by which ECU required vehicles data is acquired. In addition, the communication function between ECUs also restricts the vehicle data which each ECU can access.

5.2.11.3 Vehicle Data Upload

When a system consists of two or more ECUs, the vehicles data managed by ECU other than ECU in which application and/or middleware are working shall also be treated. For this reason, vehicle information processing communicates with it of other ECUs. Thereby, application and/or middleware can be treated, without caring about by which ECU required vehicles data is acquired. In addition, the communication function between ECUs also restricts the vehicle data which each ECU can access.

5.2.11.4 Simulator

In the development environment of application and/or middleware, since actual vehicles data is unacquirable, it is preparing the simulator which imitated actual vehicles, and makes development environment construction easy. By a simulator, it assumes using the steering wheel controller for PC games. Since this function is an object for development environment, let it be an option.

5.2.11.5 Requirements

The system must hold vehicle information and must offer the mechanism in which application and/or middleware can access vehicle information.

The system must provide application and/or middleware with vehicle information as an abstract property. For example, the speed of vehicles must be not the number of rotations of a wheel but the speed of a car.

System must provide a mechanism to add or delete vehicle property easily.

System must support typical vehicle property as “standard property”.

As for a standard property, it is desirable for the same attribute name to be the same meaning.

System must provide a mechanism to add or delete custom vehicle property easily.

A custom property is a property which a system donor can add uniquely in addition to a standard property.

Let the unit of the value of Vehicle Info Data be an international unit(meter, gram, …etc)

The value of Vehicle Info Data should have sufficient accuracy which application and/or middleware need. For example, when a unit is made into Km/h, an integral value is not enough as the accuracy of Velocity. It is necessary to change Km/h into MPH in the country of a mile display. Moreover, it is because the error of the speed display is defined by law.

A vehicle information control facility requires the mechanism in which vehicle information is stored. A lot of events generate some information at high speed. About such information, the load to a system has few directions processed collectively. Moreover, when data is taken and spilt by an application, the structure which can carry out recovery is required.

It is not realistic to accumulate all the information that changes at high speed. For this reason, In corresponding to neither of the following, it shall not store the change data.

- The amount of change of a value. It is not accumulated when the difference from the accumulated newest value is less than a threshold value.
- Lapsed time from the last change It does not accumulate, if time has not passed since the newest accumulation.

About each vehicle information, the threshold value and cumulative dosage of accumulation need to be able to set up easily.

In addition, it also makes it possible not to accumulate specific vehicle information.

System must provide an interface to application and/or middleware regarding vehicle property access.

System must provide an interface to retrieve vehicle property from application and/or middleware.

Below attributes must include in this interface

- Zone(optional)
- Property name
- Value
- Timestamp - Timestamp specifies last updated time of corresponded vehicle property.

System must provide an interface to set abstracted value to vehicle property from application and/or middleware.

Below attributes must include in this interface.

- Zone(optional)
- Property name
- Value

System must provide an interface to subscribe status change of vehicle property from application and/or middleware.

When status changed, system must invoke callback function with below attributes.

- Zone(optional)
- Property name
- Value
- Timestamp
- Sequence number

Timestamp specifies last updated time of corresponded vehicle property.

Sequence number is useful to check event order.

The acceptable value of change can be specified for vehicle information about the notice of change of vehicle information.

In order to lower system-wide load, it will not notify, if it is change which is less than an acceptable value even if vehicle information changes.

For example, although engine number of rotations changes every moment, in the case of the application which displays it in 20 steps, it is not necessary to know less than several percent of change.

It shall not notify the change, in corresponding to neither of the following.

- The amount of change of a value - It does not notify, if the amount of change of the value from the last notice of change is less than specification.
- Lapsed time from the last change - From the last notice of change, if it is less than a definite period of time, it does not notify.

Depending on application, the notice with a fixed cycle is more convenient than the notice at the time of change.

What is notified only the specified cycle even if it changes two or more times into the specified notice interval is made possible.

The data stored is acquired collectively.

Below attributes must include in this interface.

- Zone(optional)
- Property name
- Values
- Timestamps

It is desirable that the time range to acquire can be specified. For example, data from 10 seconds before to the present, data from 13:20 to 14:00, etc.

There is an attribute for which change/reference is simultaneously needed in relation to mutual in vehicle information. For example, latitude, longitude, and an altitude are changed simultaneously. If these pieces of vehicle information is changed and referred to individually, the newest longitude may acquire the value in front of one, and a current position may be unable to recognize latitude correctly. For this reason, it is necessary to summarize the vehicle information relevant to mutual and to access it.

Access of ones of those vehicle information is deterred until renewal of all the vehicle information included in Property Set at the time of a setup of vehicle information is completed, and renewal of ones of those vehicle information is deterred until it completes acquisition of all those vehicle information at the time of reference.

The definition of the vehicle information included in Property Set is being able to change easily. Or the thing which can be changed from a program during operation.

System must provide a mechanism of access control per each property. For example, property "velocity" can be accessed from only application A and B, but property "turn signal" can be accessed from all applications.

System must also provide a mechanism of access control per each method even if same property. For example, about "seat setting", all applications can get this property, but only application C can set this property.

Permission for each property and method must be configurable easily. Because, access control policy may be different per car type, grade and destination.

System must provide a mechanism to enable routing any vehicle property both within same ECU and across two or more ECU's.

If a Property Change event is received from VIC, change can be notified to all the applications, middleware and other VICs which are subscribing change of the vehicle information. In addition, the notice of change must be able to be distributed also to the application and/or middleware which exist in a different ECU.

VIC can be requested to set the value specified as Property.

It can set, even if it exists on ECU from which an application and VIC differ.

The newest value can be returned immediately, without asking VIC to the acquisition demand from an application. For this reason, keep the newest value of each Property.

Even if it is in ECU from which VIC of the Property differs, the demand from an application responds.

It can exchange with two or more VICs. Addition and deletion of Data Provider can be performed easily.

The data exchange between ECUs should be permitted by VIC.

All data transmission and reception from other Software Component are refusing.

The system should have a mechanism which communicates the stored vehicles.

The vehicle information to upload is being able to choose.

A selection condition is that the following specification is possible at least.

- Date-and-time range
- Object vehicles data
- The change threshold value of vehicles data

Enable change of selection of vehicle information easily. As for this, it is desirable for it to be able to change dynamically from an external.

The simulator of vehicles data using the steering wheel controller for PC games, etc. as substitution of actual vehicles in development environment is prepared.

Car Simulator is being able to notify the following vehicles data to vehicles data processing activities through a vehicles interface function at least.

- Speed
- Shift position
- The direction of vehicles
- Latitude and longitude of a current position
- Turn signal

The steering wheel controller for PC games to be used is being able to obtain easily. Moreover, it is desirable that two or more steering wheel controllers can be used.

VIC should fill the following performance specifications and performance.

It is a value on condition of H/W assumed that the following values will be used for in-vehicle information machines and equipment in 2016.

- Maximum number of properties : 4,096
- Maximum number of property sets: 1,024
- Greatest data storage time : 12 hours

It is a value on condition of H/W assumed that the following values will be used for in-vehicle information machines and equipment in 2016.

- Get/Set method(one property) - 0.2ms
- Get/Set method(property set include 30 properties) -1.3ms
- Subscribe callback - 2.5ms (after change of a value)
- GetHistory method(for within 3 minutes after the present) - 0.2ms
- GetHistory method (older than 3 minutes from the present) - 50ms

VIC is being able to change without having composition which has pliability and extendibility about the vehicles data to manage, and reconstructing the whole VIC about the kind and attribute of vehicles data.

Vehicle Interface treats various kinds of in-vehicle LAN and sensors, and they are mounted by various H/W according to a maker or a vehicle type. For this reason, VIC needs to be able to add and change Vehicle Interface without reconstruction of VIC.

Abstraction of vehicles data is the duty of Vehicle Interface in principle. This is because it is necessary to change the concreteness data depending on H/W of in-vehicle LAN or sensors.

However, an abstract vehicles data value may be decided by combination of the concreteness vehicles data from two or more Vehicle Interface. In this case, VIC needs to change two or more concreteness vehicles data into one abstract vehicles data.

Since this conversion is dependent on H/W of in-vehicle LAN or sensors, so it cannot be mounted in the VIC itself.

In order to solve this, suppose that the mechanism in which such a conversion module can be added without reconstruction of VIC is prepared for VIC.

5.2.12 Telematics Services

V2V, V2I, RVI, Traffic information, etc.

5.2.13 Window System

A window system is a software component that facilitates an implementation of graphical user interface. A window system is responsible for managing display devices, Graphics Processing Units (GPUs), input devices, and graphics memory. A window system works with the software component named window manager that is responsible for a layout management of windows, and a routing of user interactions.

AGL specifies that automotive grade Linux shall support multiple windows on a display.

AGL specifies that automotive grade Linux shall support multiple windows owned by multiple processes to be rendered on a display.

AGL specifies that automotive grade Linux shall support rendering to off-screen buffer to achieve flicker less rendering.

AGL specifies that automotive grade Linux shall support composition of windows with off-screen buffers.

AGL specifies that automotive grade Linux shall support a translucent window, i.e. underlying objects underneath the translucent window is visible depending on the alpha values of pixels.

AGL specifies that automotive grade Linux shall make OpenGL/ES 2.0 API compliant to Khronos group available to clients for their rendering.

AGL specifies that automotive grade Linux shall have a window manager that uses only public APIs provided by Window System and OpenGL/ES 2.0 for rendering and user interaction.

AGL specifies that automotive grade Linux shall support window manager that is replaceable by configuration.

AGL specifies that automotive grade Linux shall provide a window system that abstracts the underlying display subsystem and GPU. AGL specifies that automotive grade Linux shall have a window manager that relies on a standard rendering API such as OpenGL/ES 2.0 only. The window manager shall not rely on any hardware specific API. A window system and OpenGL/ES 2.0 API are responsible for a hardware abstraction.

AGL specifies that automotive grade Linux shall support multi-headed display where available.

AGL specifies that automotive grade Linux shall support mirroring of windows to multiple displays.

AGL specifies that automotive grade Linux shall support hardware layers, such as DRM planes, where available.

AGL specifies that automotive grade Linux shall compose windows using available hardware acceleration capabilities.

AGL specifies that automotive grade Linux shall support management of windows and inputs from users depending on statuses of a vehicle. The statuses of vehicle include a speed of a vehicle, a motion of a vehicle, etc. For instance, the inputs may needs to be limited while the vehicle reaches to the certain speed.

AGL specifies that automotive grade Linux shall abstract physical input devices such as buttons, a touch panel, a control knob etc.

AGL specifies that automotive grade Linux shall support On-screen keyboard which takes input from available physical input devices.

6 Security Services

Security framework

6.1 Access Control

Access Control describes requirements for AGL Access Control.

Access control is a mechanism to grant / deny access to APIs/files in the system.

6.1.1 Requirements

AGL system must support a system-wide access control mechanism.

7 Operating System Layer

7.1 Kernel

7.1.1 Linux Kernel

Automotive Grade Linux uses the Linux Kernel. The kernel is constantly evolving with a new release about every sixty days. The automotive industry has design cycles of three to five years for IVI systems. Somehow a balance must be struck between updating operating system and kernel every few months and keeping up to date with modern features that the kernel and the rest of the open source community provides,

7.1.1.1 Requirements

AGL kernel shall be based on Long Term Support Initiative (LTSI) kernel.

At the moment LTSI kernel is the only open source/public kernel that gets closer to automotive industry needs – it has certain automotive industry demanded components integrated, it is fully aligned with Linux LTS trees so it leverages security fixes and/or generic bugfixes adapted by Linux community, LTSI kernel merge window is more flexible to industry demands and allow to accumulate wider set of features, components and bugfixes relevant for industry (comparing to regular Linux kernel merge/release cycle). LTSI kernel is thoroughly validated manually and with the help of automated tools to track and discover anomalies and regressions.

AGL development process should utilize bug tracker with ability to mark bugs as open/fixed on particular distribution branches. Open bugs should have direct impact on release decisions.

7.2 Boot Loader

7.3 Hypervisor

TBD. Need to add very basic "background" regarding virtualization, explain about OS-level virtualization/isolation, then about type1/type2 hypervisors (virtualization). In modern IVI systems OS-level virtualization is widely used (applications isolation, combination of Android and Linux apps together), future – maybe Linux/IVI + ADAS + Instrument Cluster = guests on top type1 hypervisor.

7.3.1 Requirements

AGL shall provide OS-level mechanisms for running multiple isolated instances (containers) that have its own directory structure, network devices, IP addresses and process table. The processes running in other containers shall not be visible from inside a container.

AGL Linux should be configurable to work as Type-1 "bare-metal" hypervisor "guest". Following functionality shall be supported by AGL Linux "guest":

- IPC (with hypervisor and other "guests")
- "paravirtualized" device drivers for peripherals shared with other "guests" (unless virtualization abstraction is supported by hardware)

7.4 Operating System

7.4.1 File Systems

File system (FS) requirements for AGL concentrate on Reliability, Accessibility, and Serviceability as their main characteristics.

- *Reliability*means data integrity protection, automatic error detection and correction, tolerance to power failures, robustness under stress I/O load in multi-process environment, extended lifetime via use of wear leveling and bad block management techniques.

- *Accessibility*means ability to use external storage devices, as well as accessing designated parts of internal file system over secure wired or wireless connections.
- *Serviceability*means ability to upgrade AGL as a whole or by updating individual packages, and availability of file system checking and optimization tools.

Below is short summary for better understanding of FS Requirements hierarchy.

FS Requirements	R-FS References
6. File Systems (P1) 6.1. Robust File System for managed internal storage (P1) 6.1.1. Power failure tolerance (P1) 6.1.2. Quick recovery after power loss (P1) 6.1.3. Multi-threaded I/O (P1) 6.1.4. On-demand integrity checker (P1) 6.1.5. Read-only mode (P1) 6.1.6. Non-blocking unmounting (P1) 6.1.7. Means for optimizing I/O performance if it may degrade under certain conditions. (P2) 6.1.8. File space pre-allocation (P2) 6.1.9. Meta-data error detection (P2) 6.1.10. File data error detection (P2) 6.1.11. Online integrity checking (P2) 6.1.12. Write timeout control (P2) 6.1.13. Compression support (P2) 6.1.14. Quota support (P2) 6.1.15. I/O process priority (P2) 6.1.16. File system event notifications	2. btrfs 2.1. btrfsck 3. ext2 3.1. e2defrag 4. ext3 5. ext4 5.1. e4defrag 5.2. e2fsck 6. vfat 7. UBIFS 8. Generic tools and APIs 8.1. fan

(P2) 6.1.17. Logical block size control (P2) 6.1.18. Snapshots (P2) 6.2. File System for non-managed internal storage (P1) 6.2.1. All P1 requirements from FS.1.1.x list (P1) 6.2.2. Wear leveling (P1) 6.2.3. Error detection/correction (P1) 6.2.4. Tolerance to flipping bits (P1) 6.2.5. Read/write disturb awareness (P1) 6.2.6. Bad block management (P1) 6.2.7. As many P2 requirements from FS.1.1.x list as possible (P2) 6.2.8. Wear leveling statistics (P2) 6.3. File Systems for removable storage (P1) 6.3.1. Restricted functionality from security point of view (P1) 6.3.2. Automount/autounmount (P1) 6.3.3. Automatic synchronous flushing of modified data to physical media (P2)	otify 8.2. fstrim

7.4.1.1 Requirements

AGL shall provide a set of file systems to support the following types of storage devices: internal managed (SSD, eMMC, etc.), internal non-managed (raw NOR and NAND FLASH memory), removable managed (USB stick, SD card).

AGL shall provide robust file system suitable for use on managed internal storage devices, AGL shall provide robust file system suitable for use on non-managed internal storage devices,

AGL shall provide a set of file systems popular on removable media devices.

A system must be able to withstand power failures under heavy load of meta-data-intensive, and data-intensive operations, including power-failures during OS startup, and shutdown.

A file system must be able to restore good data and meta-data state after unexpected power interruption without performing the full time-consuming integrity check. Such recovery should not add more than a second to the normal boot process after power failure on idle system. Normally this is achieved via journal- or log-based (also known as transactional or copy-on-write) operation.

A file system must be able to handle meta-data-intensive, and data-intensive I/O from multiple threads and/or processes simultaneously.

A file system must have integrity checking tool with ability to execute it on-demand.

A file system must be able to switch between read-only, (when no data is committed to physical storage device), and read/write modes in runtime. E.g. via "mount –o remount,ro <device>" command.

AGL must support "lazy" (delayed) unmounting.

AGL should provide means for optimizing potentially degraded I/O performance after prolonged file system and storage use. Often, this refers to offline or online file system defragmentation. Another example is periodic fstrim execution on SSD storage.

A file system should be able to pre-allocate space for created/extended files on request. This may be used to minimize fragmentation of frequently written files.

A file system should have an option of automatic error detection in its meta-data.

A file system should be able to associate error detection codes with separate blocks of stored data, and to verify the data against the codes in runtime upon each read from a physical device.

A file system should have a utility for meta-data integrity checking on mounted partition.

A file system should allow changing timeout after which it flushes modified data to physical media.

A file system should support automatic data compression.

It should be possible to enable file system quotas for particular users and/or groups.

AGL should allow to set I/O scheduling class and priority for particular processes.

AGL should allow user space applications to subscribe for file and directory change notifications.

Making logical block size equal to a power of physical block size may improve physical I/O performance, and decrease file fragmentation impact.

A file system should allow creation of snapshots.

A file system must perform wear leveling before writing data, so that the limited number of erase/program cycles is evenly distributed across all device blocks.

A file system must support the following error detection/correction algorithm(s): BCH4, BCH8.

A file system should not just be able to detect/correct a number of flipped bits but should also actively prevent the issue from happening in the first place, especially after unexpected power interruption. Known techniques include forced reprogramming of blocks that were in use at the time of power failure, and copying data to a fresh block after detected error correction.

A file system should not just be able to detect/correct errors caused by read/write disturb phenomenon but should also actively prevent the issue from happening in the first place. Known techniques include limiting the number of read cycles between erases, and copying data to a fresh block after detected error correction.

A file system must perform bad block detection and management transparently to file system users.

Current FLASH wear-related statistics should be accessible via user-space utility.

A file system must support noexec, and nodev mount options.

A file system must be able to automatically mount plugged-in removable media, and automatically unmount it when unplugged.

A file system must support sync mount option.

AGL shall provide a set of file systems to support the following types of storage devices: internal managed (SSD, eMMC, etc.), internal non-managed (raw NOR and NAND FLASH memory), removable managed (USB stick, SD card).

7.4.2 Resource Control

In IVI system, it depends time and occasion that which application and/or middleware should be higher priority. Resource control provides basic functionality regarding proper resource allocation for each process and/or process group.

(cgroups)

7.4.2.1 Use Case and Role

If end user specified a destination and started route guidance, map drawing following current position and voice and/or visual guidance should be treated as higher priority than others.

On the other hand, if end user is watching a movie, movie player and decoder should be assigned to higher priority than others.

Important point is that it may assign two or more high priority application and/or middleware at the same time. And, one function may be provided from two or more processes.

Table 9-33 describes the role of resource control to be satisfied above purpose and use cases.

Table 9-33 : Role of Resource Control

No.	Role	Description
1	Priority	Allocate resource via its own priority. High priority process and/or process group should be assigned more resource.
2	Time slot	To share resource per time slot.
3	Release	Forced release of partially or whole allocated resource.
4	Grouping	Grouping two or more processes, and allocate resource per defined process group.

AGL assumes four types of resources, CPU, memory, storage bandwidth and network bandwidth. Table 9-34 describes associated roles per each resource type.

Table 9-34 : Functions of System Resource Management

7.4.2.2 Requirements

7.4.2.2.1 Priority

System provides a mechanism to set resource priority per each process.

System provides an interface to set and refer resource priority of specific process.

This interface must be called from other process.

CPU resource must support "priority" based resource management.

Resource Manager should dynamically change the ratio of offering resources according to the status of resources using by system. And its configuration must be changed easily.

Resource Manager should log the status of resources using by system.

Resource Manager should offer resources separately to threads of user land and threads of kernel. And Resource Manager should treat the bottom half and software interrupts as high priority tasks.

7.4.2.2.2 Time Slot

When two or more process request to same resource at the same time, system must provide a mechanism to mediate to guarantee the time slot to obtain specific timeframe for each processes.

System must provide an interface to set specific timeframe to obtain time slot per each process.

System must provide a mechanism of resource sharing by time slot regarding CPU, storage bandwidth and network bandwidth.

Scheduler should detect the status of resources for each thread.

Scheduler must not run the specific thread for more than 10 micro second.

Scheduler should guarantee that threads can run periodically.

Scheduler should control the dispatches that occur extremely.

7.4.2.2.3 Release

System must provide an interface to release all or partial resource which had obtained by specific process.

System must provide a mechanism of resource releasing regarding memory resource.

7.4.2.2.4 Grouping

System must provide a mechanism to group two or more processes regarding resource management such as priority, time slot and releasing. System must able to assign same attributes to grouped processes altogether.

System must provide an interface to group two or more processes from other process.

System must provide a mechanism to group regarding CPU, memory, storage bandwidth and network bandwidth.

7.4.3 Startup/Shutdown Control

Boot/Shutdown Control is a mechanism to control boot and shutdown of a program running in a user space. The order of boot/shutdown in the target program can be easily swapped depending on the product configuration. Boot/Shutdown Control supports both “static order” which boots/shuts down the program according to the static dependency of each program, and ”dynamic order” which swaps the order dynamically in specific conditions.

7.4.3.1 Use Cases

(1) Static Modification of Boot/Shutdown Order

a. Setting up of Boot/Shutdown Order Based on Product Configuration

To support various product configurations, the integrator configures/modifies orders of boot/shutdown for all programs running on the target device.

b. Configuring the Order of Boot/Shutdown during a Program Development

In order to evaluate a developed program, the developer modifies only the order of the developed program in target programs.

c. Configuring the Order of Boot/Shutdown when Software Update

Maintainer modifies the order of boot/shut down for a program to be updated when software update.

(2) Dynamic Modification of Boot/Shutdown Order

a. Prioritized Boot of the Features which the User was Previously Using

It dynamically modifies the boot order of the target program in order for last used features (e.g. audio) to be operated by priority when ACC turns ON.

b. Prioritized Boot of Update Functionalities

Update related programs are booted by priority when connected with maintenance kit and ACC turned ON.

7.4.3.2 Requirements

Boot/Shutdown Control shall start components, which are configured to be started.

Boot/Shutdown Control shall ensure that dependent components are started in the order that has been configured.

Boot/Shutdown Control shall start independent components in parallel.

Boot/Shutdown Control shall stop components, which are requested to be stopped.

Boot/Shutdown Control shall ensure that dependent components are stopped in the order that has been configured.

Boot/Shutdown Control shall be configurable by run level to start corresponding modules.

7.4.4 Database

Due to the nature of AGL operating environment, it is very important for DB engine to guarantee database instance integrity after power failures. Other important feature for generic system database engine is rich set of bindings to various programming languages.

Below is short summary for better understanding of DBS Requirements and References hierarchy.

1. Power failure tolerance (P1)
2. Quick recovery after power loss (P1)
3. Multi-threaded I/O (P1)
4. API bindings for C programming language
5. On-demand integrity checker (P2)

DB instance integrity must be ensured after power failures under heavy load of read and write DB transactions.

DB engine must be able to quickly restore good data state after unexpected power interruption. Such recovery should not add more than a second to the normal boot process after power failure on idle system.

DB engine must allow read and write access to DB instance from multiple threads and/or processes simultaneously.

DB engine API must be available for C-based applications.

DB engine should have DB instance integrity checking tool with ability to execute it on-demand.

DB engine must be able to quickly restore to a previously defined state after unexpected power interruption during adding some data.

DB engine should have availability to merge some data from internal and external databases, such as vehicle information database and databases at data center.

And DB engine should have accessibility to allow read access to DB instance during merging. Also, DB engine should have durability not to break its data after unexpected power interruption during merging.

7.4.5 System Update

Maintenance of in-vehicle devices is also an important role for any automotive system. There are numerous use cases for updating the device software such as software failure,security patching, bug fixes, and new features. Because automotive devices are battery operated and subject to power cuts any System Updates must be robust enough to withstand sudden power loss.

System Update module should have a Robust version up function.

System Update moduleshould have a system difference version up function.

There should be a data update structure for each file or package (same as WindowsUpdate or apt of Linux distribution).

There should be a data update structure for each file or package (same as WindowsUpdate or apt of Linux distribution).

Difference update should be enabled for kernel, middle ware and application.

If power discontinuity (forced restart) occurs during update for differences, the system should be recovered after choosing the status (before or after update) for each update target.

If power discontinuity (forced restart) occurs during update for differences, the status (during update) should be detected and the system should restart.

Time required for applying patch should be 5 minutes maximum for single 10MByte data.

Memory usage for difference update should be maximum 1Mbyte.

Unit amount for difference data should be 10MByte maximum for difference update.

System Update moduleshould have full version up function for whole system.

Kernel, middle ware and application should be mass updated. System structure should allow mass update.

There should be mass update structure for kernel, middle ware and application.

If power discontinuity (forced restart) occurs while mass update of kernel, middle ware and application, the status (during update) should be detected and the system should restart.

If power discontinuity (forced restart) occurs while mass update of kernel, middle ware and application, the status (during update) should be detected and the system should restart.

7.5 Device Drivers

Device drivers may be in kernel space or user space or a combination of both.

7.5.1 Peripherals

Typical IO device drivers such as SPI, USB, memory, I2C that are typically present on a SOC.

The flash process must be robust with an endurance of more than 10k write/erase cycles and data retention over 15-years/10 ppm, assuming application specific worst-case conditions. For optimised timing for downloading and restoring data the programming access time shall be less than 50 s/byte average.

The EEPROM process must be robust with an endurance of more than 100k write/erase cycles and data retention over 15 years/10ppm. Higher programming voltage than 5 V for Flash or EEPROM is not allowed.

In applications that need to save data at power down, the programming access time must be fast. (target <1ms/byte)

N.B. EEPROM functionality can be emulated in flash memory passing the requirements above.

7.5.2 Graphics Drivers

Graphics drivers provide the interface to the graphical resources (e.g., GPU) within the system. This may include on-board graphical resources or a separate GPU from the main SOC.

7.5.3 Video Drivers

Video codecs allow the system to decode and/or encode video for playback or recording. Video codecs will nearly always be hardware based.

7.5.3.1 Requirements

The system shall provide device drivers to access any hardware implementation of video functionality.

7.5.4 Audio Codecs

7.5.4.1 Requirements

Automotive Grade Linux BSPs shall provide devices drivers to access audio codecs that are implemented in hardware.

Automotive Grade Linux BSPs should provide software implementations for those audio codecs that are required for AGL-based products and not supported in hardware.

7.5.5 Automotive Devices

Device drivers for automotive related devices. This may includes buses such as CAN, MOST, or LIN. Device drivers may be required for receivers (AM, FM, SDARS, etc). Drivers may also be required to directly interface to sensors that may not be on the bus such as gyros used for navigation or an air bag sensor for a telematics system.

8 Notices

Linux is a registered trademark of Linus Torvalds.

The Linux Foundation and Yocto Project are registered trademarks of The Linux Foundation.

Bluetooth is a registered trademark of the Bluetooth SIG Inc.

Miracast is a registered trademark of the Wi-Fi Alliance.

MirrorLink is a certification mark of the Car Connectivity Consortium.

AirPlay is a trademark of Apple, Inc.