

Audio management Architecture to be extended for Cockpit system

Naohiro Nishiguchi

Advanced Driver Information Technology Corporation

ADIT is joint venture company of DENSO Corporation and Robert Bosch GmbH/Robert Bosch Car Multimedia

- I have more than 10 years experiences in IVI product development for Audio domain
 - ◆ Application layer
 - ◆ Middleware (e.g. Media playback engine, Beep playback engine)
 - ◆ Audio routing management to adapt/configure real customer projects
 - ◆ ALSA
- I am audio experts from low-level to high-level application layer covered, and had been working more on customer project.
- Now, I started more on platform, especially audio routing management framework to be more easily applicable to actual customer projects.

■ Background/Introduction

- ◆ Motivation and necessity of audio management in cockpit system

■ Use case and system architecture in Cockpit system

- ◆ Typical audio management use case example
- ◆ Example system architecture
- ◆ Case study

■ Current AGL audio architecture

■ Open points

■ Next step

- ◆ Proposal approach status of proposal approach

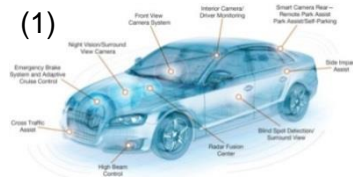
■ Trends in Vehicle

- ◆ **Increasing amount of information** handled by vehicle
 - Connected to infrastructure, telematics, vehicle to vehicle, vehicle internal information in addition to Multimedia.
- ◆ **More intelligent HMI**
 - To present those information with appropriate timing, appropriate contents



■ Cockpit system

- ◆ Comprehensive system-wide information management with centralized HMI
- ◆ Arbitration, prioritization including Multimedia/Vehicle content can be handled.
- ◆ Especially sound related information is more important in cockpit system
 - In general drivers don't see the screen during driving, so audible information is more important than desktop-PC or smartphone
- ◆ Audio management for Multimedia and for other vehicle information **to be integrated for arbitration.**



(1) <http://johndayautomotivelectronics.com/freescale-neusoft-and-green-hills-are-collaborating-to-simplify-next-generation-adas-development/>

(2) <http://venturebeat.com/2017/01/28/how-automakers-are-cashing-in-on-the-connected-car-craze/>

(3) http://news.mynavi.jp/articles/2016/01/08/CES2016_panasonic/002.html

(4) <http://gendai.ismedia.jp/articles/-/39143>

■ AGL for Cockpit system

- ◆ Usually cockpit system consists of Multi-ECUs
- ◆ AGL based system should provide centralizes information management
 - Integration of audio management for Multimedia and other vehicle information is required.
- ◆ AGL should be flexible
 - Center of Cockpit system or controlled by other ECU
- ◆ Current AGL distro has
 - No mechanism to communicate/control other ECU
 - No audio management considering Multi-ECU system architecture

Typical Use case example in IVI/Cockpit system

■ Active Source change

1. Driver is listening to Radio in the car.
2. Driver push the button to change audio source to USB Audio.
3. System automatically start playing USB Audio

■ Mixing & Volume control

1. Driver is listening to Radio in the car.
2. Car detect moving objects when parking.
3. System mute(or reduce)the volume of Radio, and then start playing Alarm by another speaker.
4. After Alarm is completed, system recover the volume level of Radio and play again.

■ Welcome music

1. Driver is listening to Radio in the car.
2. Driver turn off, then turn on.
3. System play welcome music as soon as possible.



<https://www.extremeaudio.org/in-car-navigation/>

Example system architecture in Cockpit system

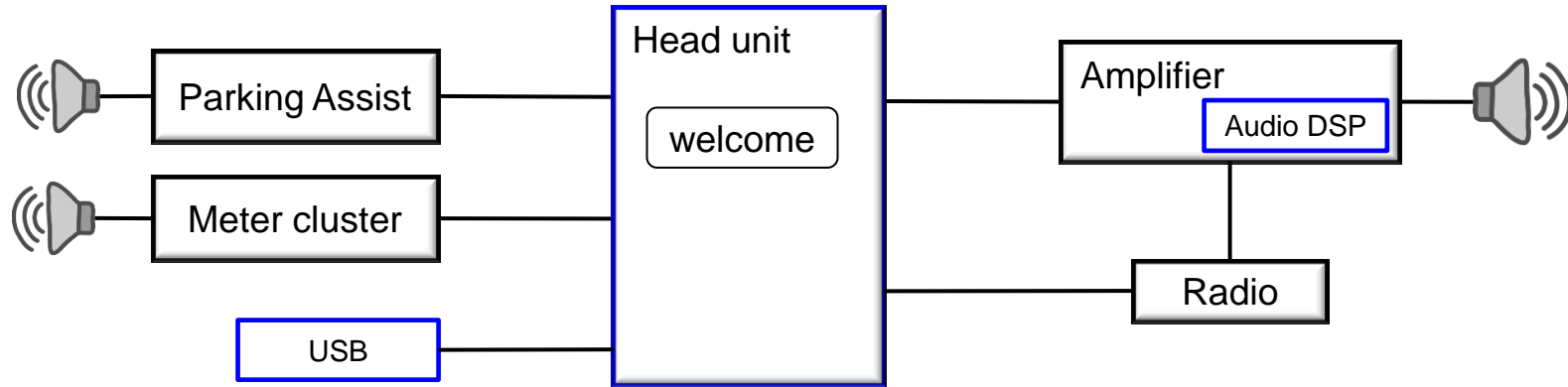
- System architecture is important for implementing audio feature on Multi-ECU.
 - ◆ As case-study on typical audio use cases is tightly coupled with the Multi-ECU system architecture
- IVI system especially in audio features/domain has been developed with multiple audio suppliers for multiple ECUs system.
That will be more cases for Cockpit system, as more suppliers will be involved.
- In general, automotive products are Multi-ECU.
Difference between single ECU system and multiple ECU system here.

Single ECU	Multi-ECU
In most cases sound devices are connected to one ECU directly.	In most cases sound devices are NOT connected to one ECU directly.
Audio streaming data can process directly. Because audio application can gather the audio streaming data from source device, and deliver to sink device.	Audio streaming data might NOT be processed directly from only one ECU.

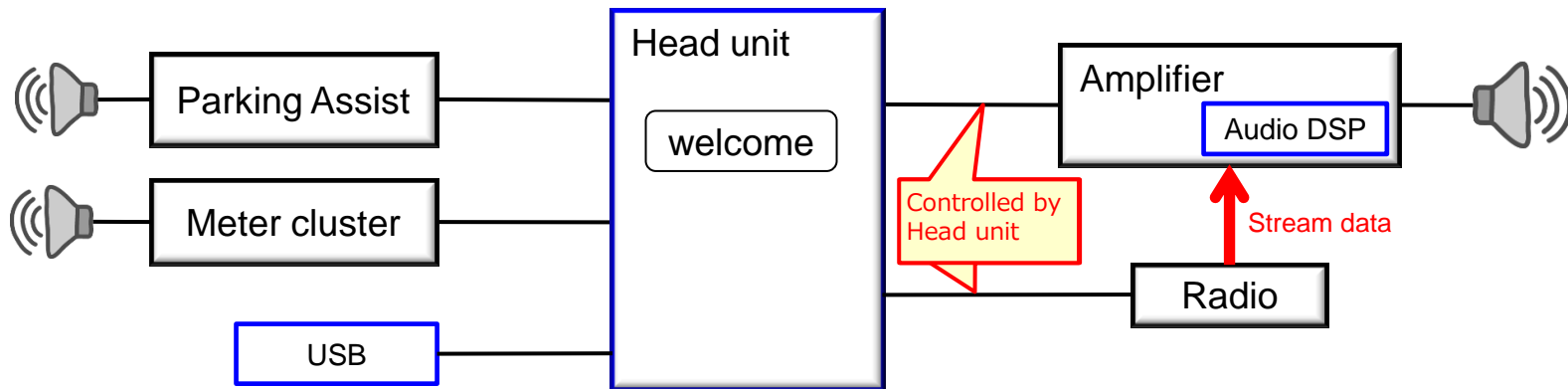
Above difference is key point to develop audio system on Multi-ECU.

- One typical example system architecture on multiple ECU in next page

Example system architecture in Cockpit system



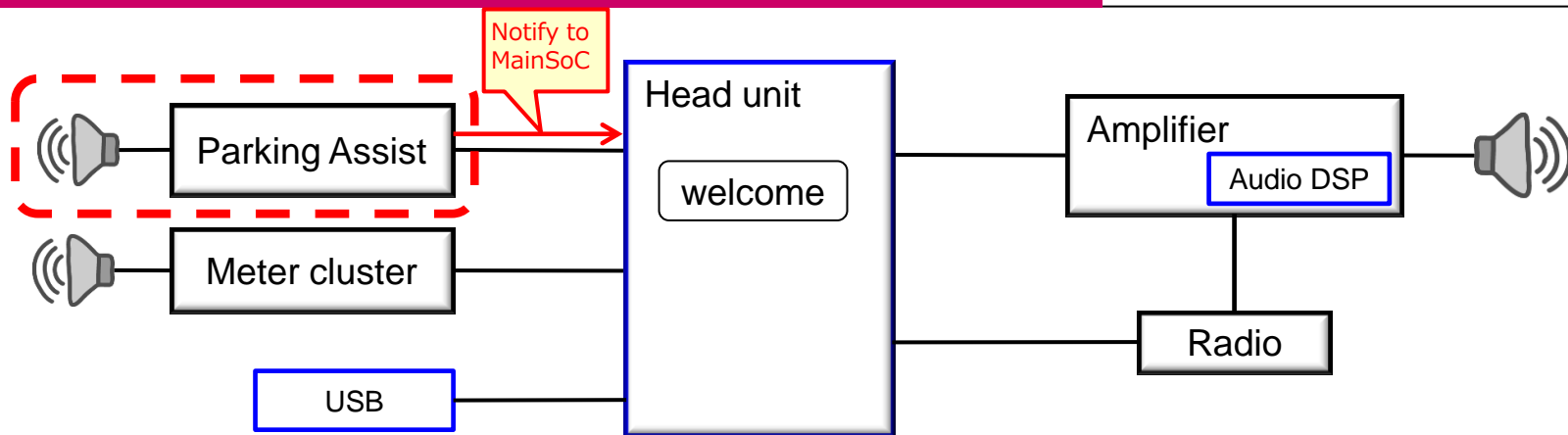
- There are separated H/W.
- Some H/W have a own speaker (Parking Assist, Meter cluster, Amplifier) **NOT one speaker**
- Head unit control Amplifier using communication like MOST.
- Blue frames are controlled by Head unit
- Amplifier (Audio DSP) has responsibility about below.
 - ◆ mix some audio input, and controlled by Head unit
 - ◆ control Volume



■ Active Source change

- ◆ Radio sound data does **NOT go through** Head unit
- ◆ Radio is controlled by Head unit.
- ◆ Head unit **should control Amplifier** to change the sound settings when changing active source.

At this time, it is necessary to follow specific communication spec. to each car maker.
(Such as MOST, CAN and others)



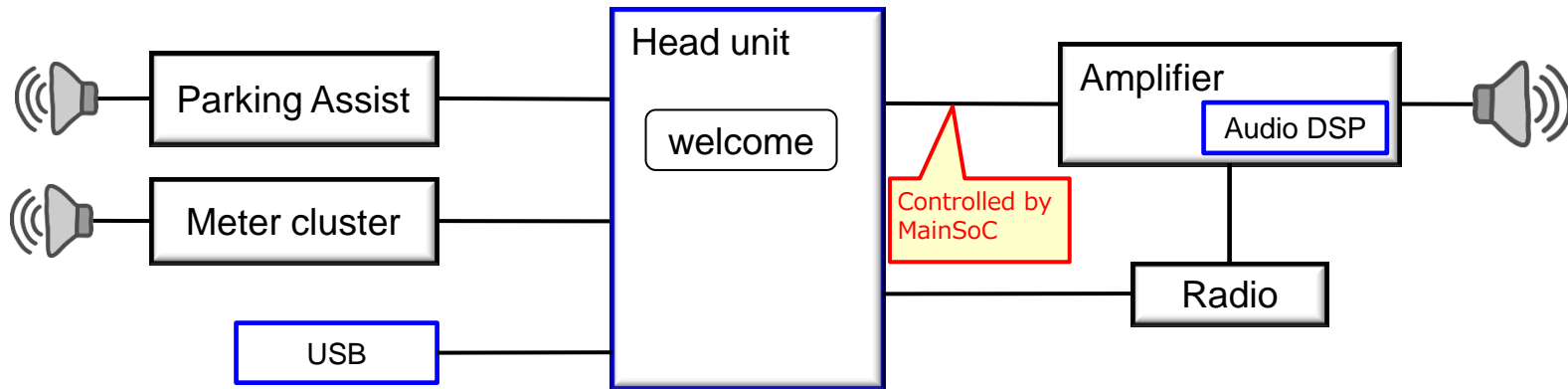
■ Mixing & Volume control

◆ Control volume according to external notification.

- Playback alarm of Parking Assist is completed in own H/W.
- When playback alarm or after playback Parking Assist notify playback status to Head unit
Then Head unit have to mute/demute Radio

◆ Using various Communication between Parking Assist and Head unit

- CAN is standard as physical layer but **logical commands are different** for each products.



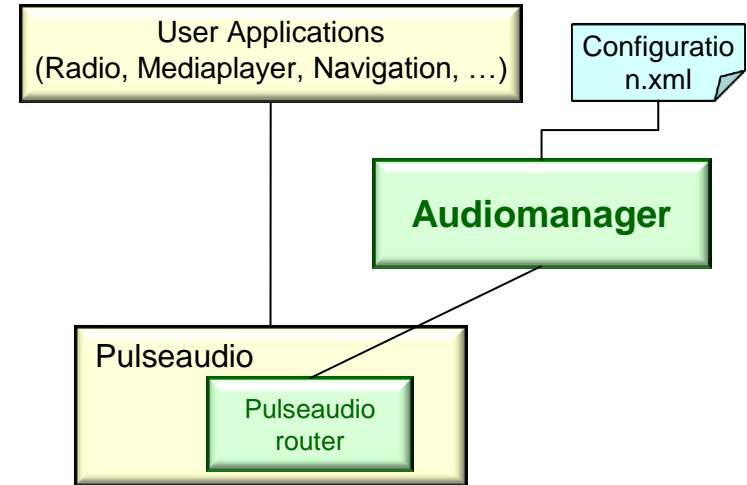
■ Welcome music

- ◆ In most cases Welcome music component is in Head unit
 - It is necessary to reduce number of ECUs related to features, in order to playback as early as possible.
- ◆ **Control the Amplifier independently from Audio management.**
 - Because the start up of the Head unit might be slow.
- ◆ **Migration** from early component to normal audio management component.
 - After playback, hand over Audio management in order to manage comprehensively

- Audio management in automotive cockpit system requires
 - ◆ Intelligent policy decision according to user operation.
 - Case study #1, #2
 - ◆ Manage entire audio routing over Multiple ECUs
 - Case study #1, #2
 - ◆ Migration mechanism from early audio
 - Case study #3

■ CES demo software Architecture

- ◆ User applications uses the Pulseaudio client library
 - User applications are **independent** from Audiomanager. Can develop it like desktop application.
- ◆ “Pulseaudio router” is pulseaudio server module(plug-in)
 - Hook routing request from applications and communicate it to Audiomanager.
 - Makes use of Audiomanager for policy decision.
 - Runs in context of Pulseaudio.
- ◆ Audiomanager
 - Decides whether to change the source or to control the volume of current source
 - Connect/Disconnect the whole audio route
 - Reduce/increase the volume using Pulseaudio

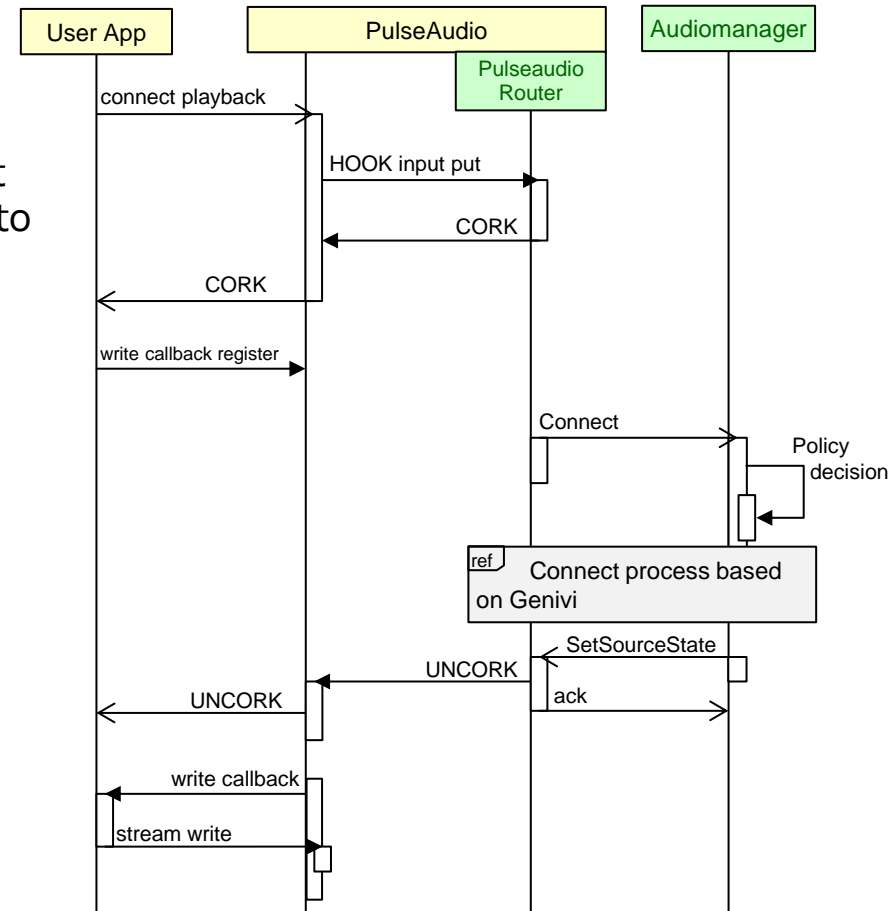


Explain details in next slide

Current AGL audio Architecture

■ Playback sequence

- ◆ User application sends playback request to Pulseaudio (`pa_stream_connect_playback`)
- ◆ This event is hooked by the "Pulseaudio router" it corks the stream and sends the connect request to the Audiomanager.
- ◆ "Pulseaudio router" send event to Pulseaudio. (`PA_STREAM_EVENT_REQUEST_UNCORK`)
- ◆ Audiomanager decide connect or not according to policy (described in `configuration.xml`)
- ◆ Audiomanager establish whole route
 - Audiomanager send request to "Pulseaudio router"
 - "Pulseaudio router" receive this event then send event to Pulseaudio (`PA_STREAM_EVENT_REQUEST_UNCORK`)
- ◆ Pulse audio send event to User applications
- ◆ User applications start playback



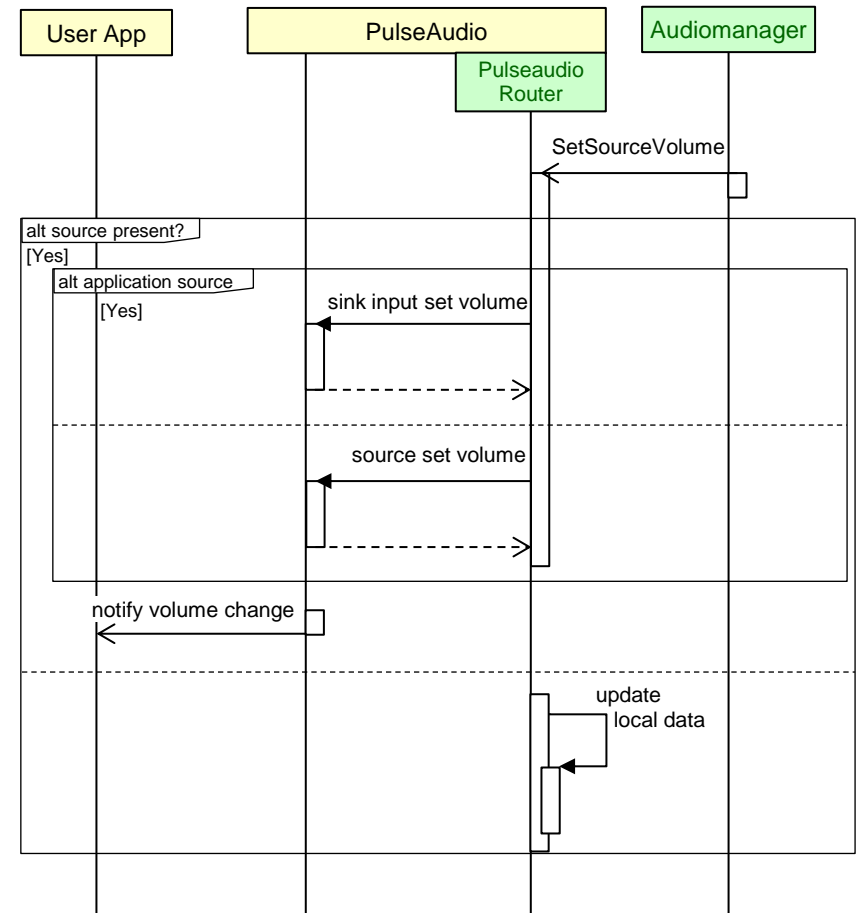
Current AGL audio Architecture

Volume control sequence

- ◆ Audiomanager request setVolume to Pulseaudio router under the following condition.
 - Trigger: receive Connect/Disconnect of new source.
 - Policy : Mix and reduce/increase volume of current source and new source.
- ◆ Pulseaudio router reduce/increase volume of current source

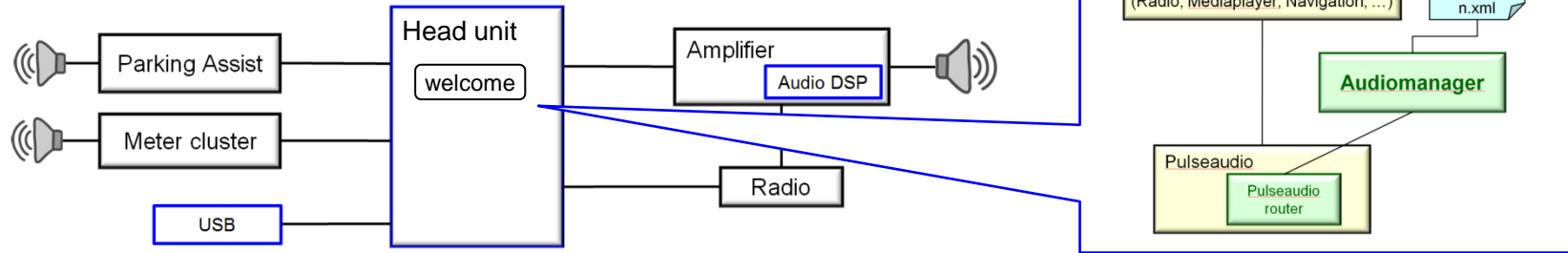
This audio management architecture satisfy the requirements of CES demo on single ECU.

But there remains some open points for Multi-ECUs Cockpit system.



■ The actual example system and current AGL audio architecture

- ◆ Map current architecture to system architecture



■ Open points for Multi-ECUs

- ◆ Can NOT control external ECU (Amplifier(include AudioDSP), Radio) from Main SoC
 - Can NOT achieve “Active source change” use case
- ◆ Can NOT receive notification from other ECU
 - Can NOT achieve “Mixing & Volume control” use case
- ◆ Welcome music feature is missing.

■ Proposal

- ◆ Mechanism for controlling external ECU to be extended.(Red lines)
- ◆ Mechanism for welcome music feature to be extended.(Blue lines)
 - Both as plug-in of Audiomanager
 - ◆ Because Audiomanager is included in current AGL architecture.

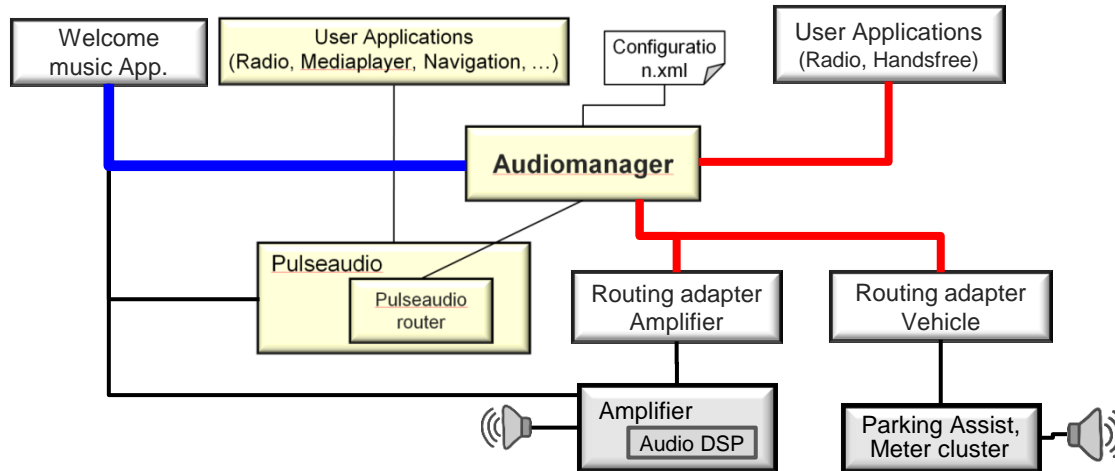


Fig : Diagram of software component

Yellow blocks are the AGL part now
White frame is the part that supplier develops according to each products.

■ Inter Node Communication is required on Multi-ECU system.

- ◆ To control external ECU from Head unit.
- ◆ To receive the request about audio management from external ECU.
(in case of sources that do not pass Head unit)
- ◆ Makes it easy to apply to various system architecture.

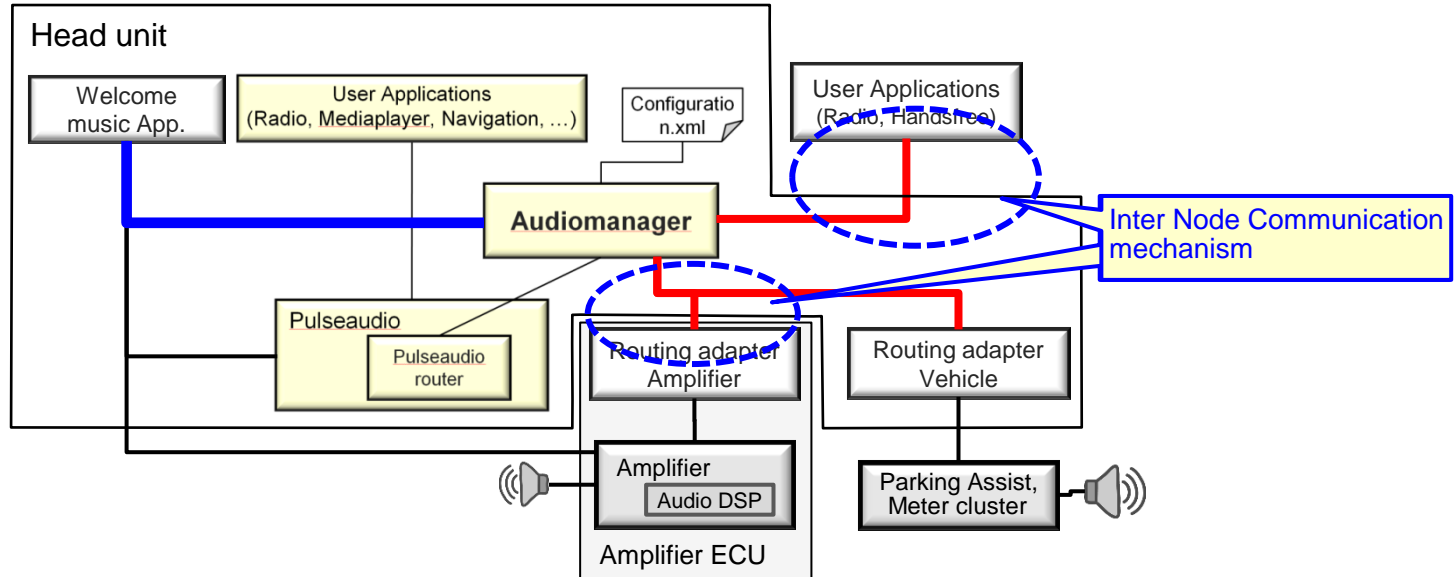


Fig : Diagram of software component mapped to physical view

Mechanism for welcome music feature

- ◆ **Migration mechanism is required.**
- ◆ As explained, Application that need to playback sound as early as possible (can not wait until audio management component starts up.)
- ◆ Application need to control external amplifier by application itself.
- ◆ As it is, it is impossible to do intelligent audio management.

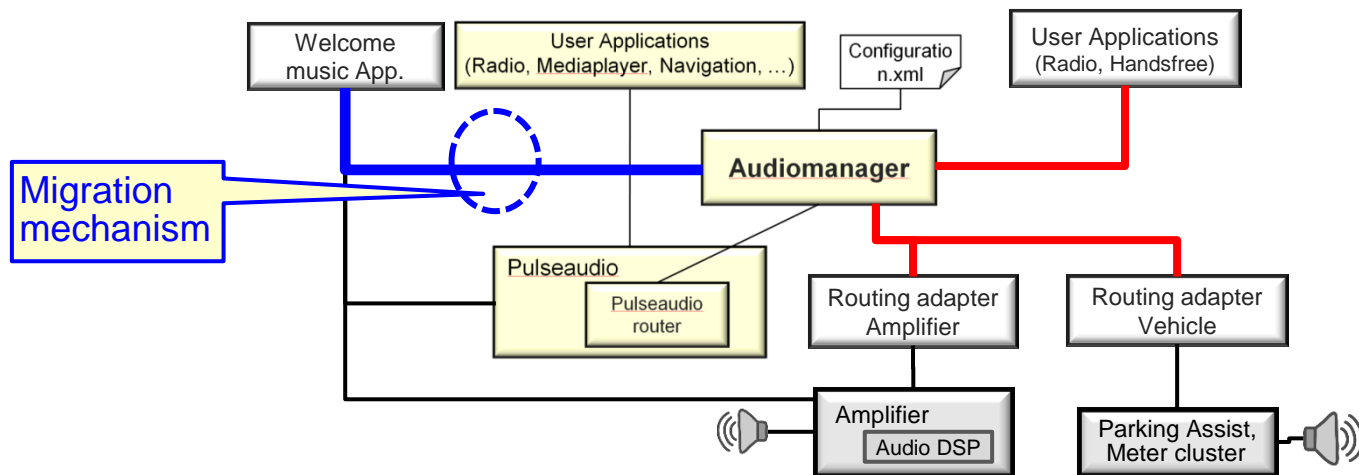


Fig : Diagram of software component mapped to physical view

◆ Inter Node Communication

- Use JSON-RPC based on socket as INC
 - ◆ Implementing it as Genivi Audiomanager Plug-in component

◆ Migration mechanism

- Migration mechanism of Audiomanager is ready
- Welcome application to be develop using migration mechanism of Audiomanager.

Questions?