Proposal for AGL Sound management

AGL F2F meeting @ San Jose

July 2017

Naohiro Nishiguchi
nnishiguchi@jp.adit-jv.com
Advanced Driver Information Technology

Who am I ?

- 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.
- Currently, I started more on platform, especially audio routing management framework to be more easily applicable to actual customer projects.

Contents

- Use cases in automotive
- System architecture
- Requirement for automotive sound management
- Comparison between Advanced ALSA audio agent and Genivi Audio manager
- Proposal
- Sound manager PoC

Example Use Cases

- **Active Source Change**
  - Driver is listening to Mediaplayer in the car.
  - Incoming phone call and answer the phone
  - IVI system **automatically** pause Mediaplayer, and then play Phone sound.
  - After Phone call is completed, IVI system **automatically** resume play Mediaplyer

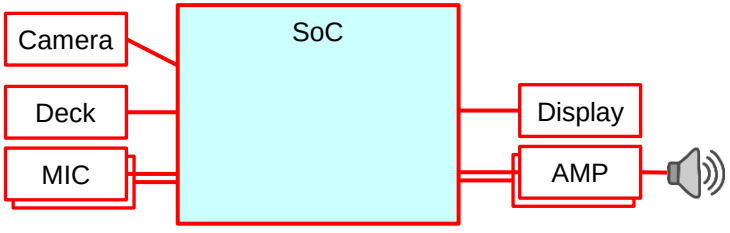
- **Last Audio (Persistence)**
  - Driver is listening to Radio in the car.
  - Driver turns off/on the engine.
  - IVI system **automatically** start playing Radio.

- **Mixing & Volume Attenuation**
  - Driver is listening to Radio in the car.
  - Car detect moving objects when parking.
  - IVI system **automatically** mute(or reduce)the volume of Radio, and then start playing Alarm using another speaker.
  - After Alarm is completed, IVI system **automatically** recover the volume level of Radio.

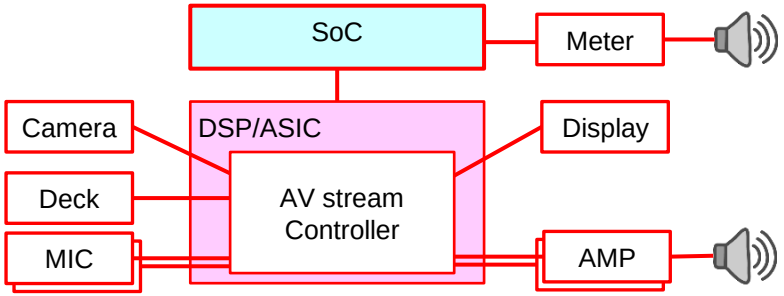
Implicit policy management is required

System architecture

- **There are several Hardware architectures as presented by MAZDA in ALS 2017**



Compact spec



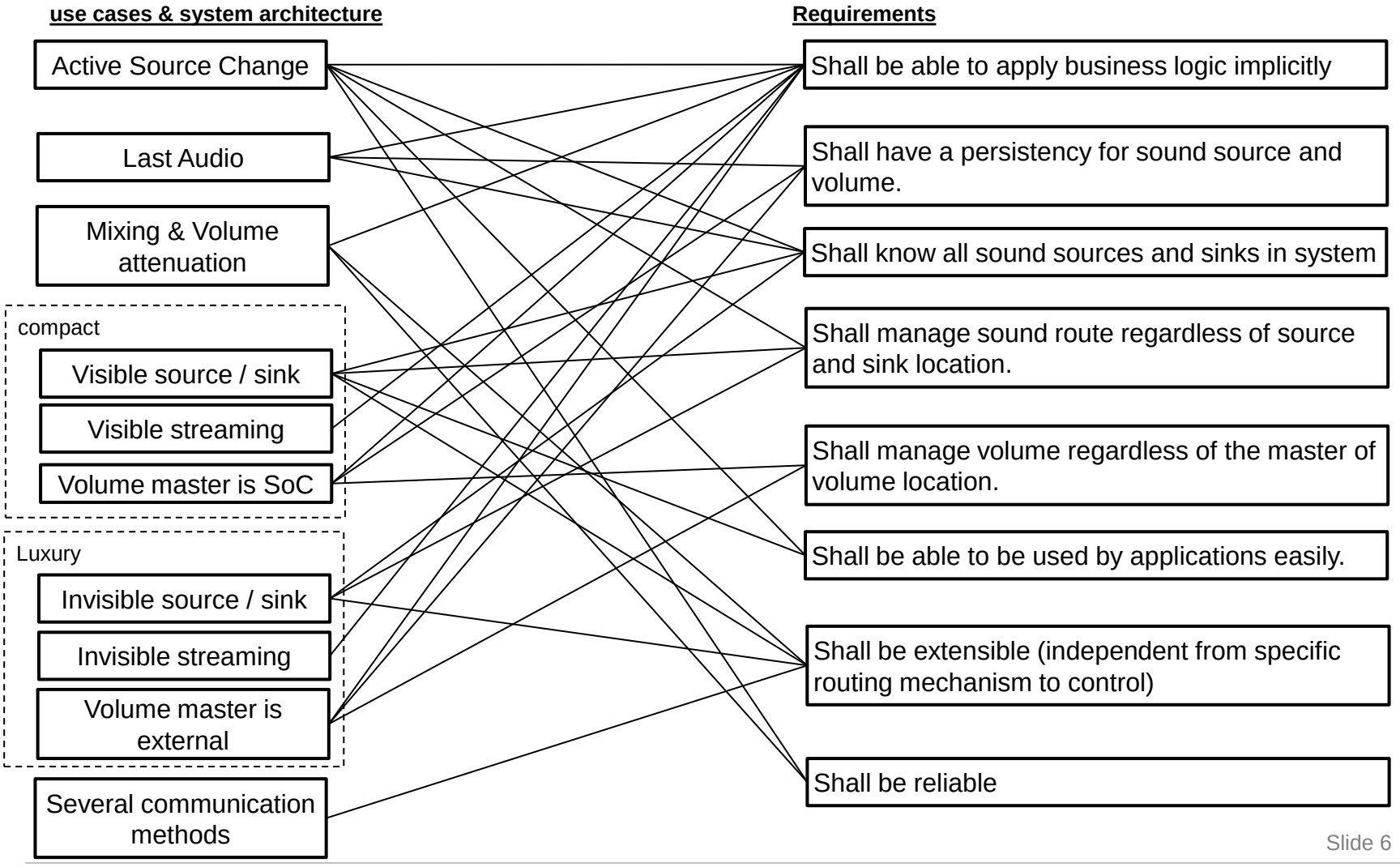
Luxury spec

- **Compact**
  - There are several communication protcol between SoC and others.
  - Sound devices are connected to SoC **directly**.
  - All audio streaming are **visible** and appliication **can control** audio streaming and volume directly.
    - SoC is master of volume
- **Luxury**
  - There are several communication protcol between SoC and others.
  - Some sound devices are **NOT** connected to SoC.
  - Some audio streaming are **INVISIBLE** and application **can NOT** control audio streaming and volume. e.g. Meter, Camera to AMP
    - External ECU is master of volume

Several types of hardware have to be supported

Requirement for automotive sound management

Requirement mapping



Comparison between Advanced ALSA audio agent and Genivi Audio manager

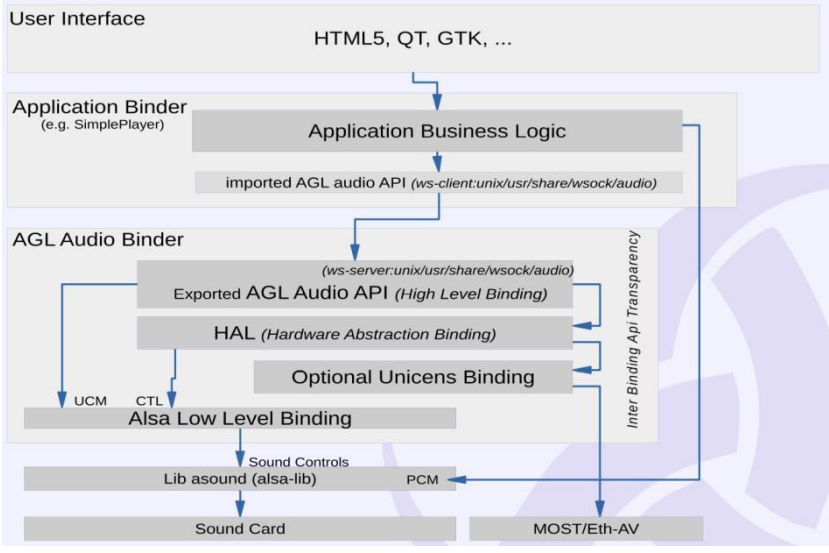
Requirement coverage

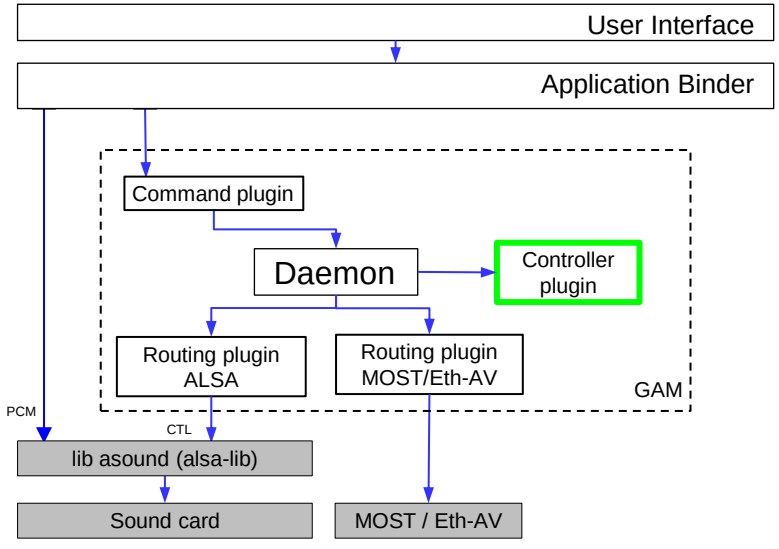
Requirements	AAAA	GAM
Shall be able to apply business logic implicitly	??? The feasibility of “Pause” is unclear.	FEASILBE
Shall have a persistency for sound source and volume.	FEASILBE Cooperation with other components is required	FEASILBE Already exist
Shall know all sound sources and sinks in system	FEASILBE By dedicated binder	FEASILBE By registration mechanism
Shall manage sound route regardless of source and sink location.	FEASILBE Dedicated binding development is required for external	FEASILBE Dedicated plugin development is required for external
Shall manage volume regardless of the master of volume location	??? How to abstract external source?	FEASILBE Dedicated plugin development is required for external
Shall be able to be used by applications easily.	It is general in Linux	It is OSS.
Shall be extensible (independent from specific routing mechanism to control)	Dedicated binding is required	Dedicated plug-in is required
Shall be reliable	Developing	Already in the market

There is no big deference between AAAA and GAM in requirement point of view

Comparison between Advanced ALSA audio agent and Genivi Audio manager

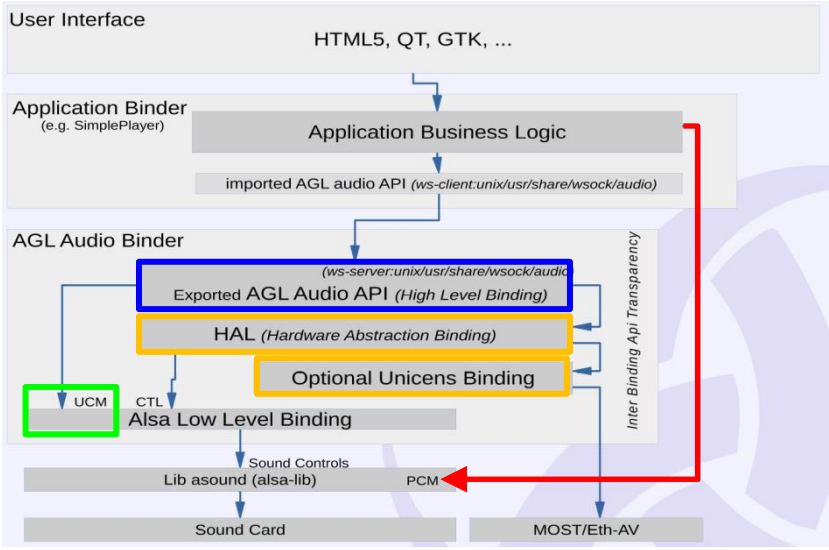
Architecture point of view

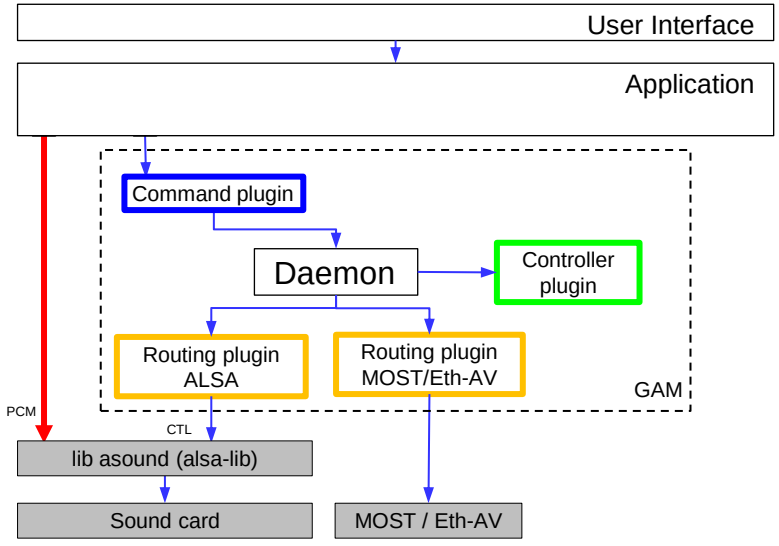




Comparison between Advanced ALSA audio agent and Genivi Audio manager

Architecture point of view





	AAAA	GAM
Interface	High level Binding	Command plugin
Policy	ALSA UCM + ???	Controller plugin
Adaptation	HAL, Optional Unicens Binding Dedicated plug-in development is required for external	Routing plugin ALSA and MOST/Eth-AV Dedicated plug-in development is required for external
Play streaming	**App -> ALSA**	**App -> ALSA**

There is no big deference in architecture point of view.
Complexity of both is not different

Proposal

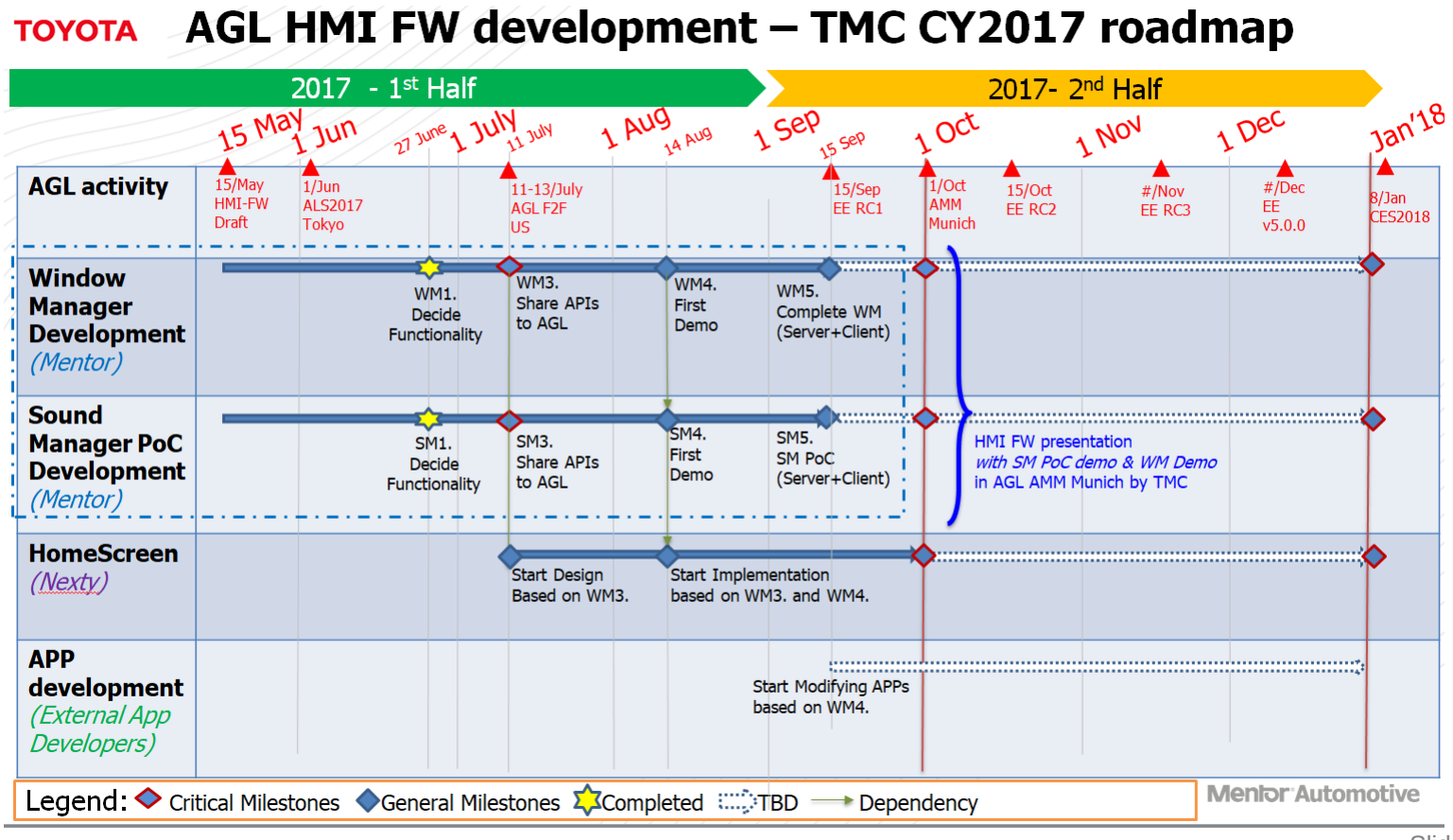
- **We can realize requirements whichever we choose AAAA or GAM**
  - There is no big difference between AAAA and GAM in requirement and architecture point of view.
  - Realization with AAAA has already been realized by GAM.

- **I suggest using GAM for sound policy management because:**
  - In order to develop sound management of AGL earlier, it is better to use existing software.
    - GAM is already integrated in AGL
    - Interface and sequence of GAM are already specified and published.
    - GAM has been already evaluated in the market.
  - We **should focus on** integrating new technologies (e.g. UNICENS, CAN communication or something) to sound management of AGL, rather than **implementing new back end of sound management**.

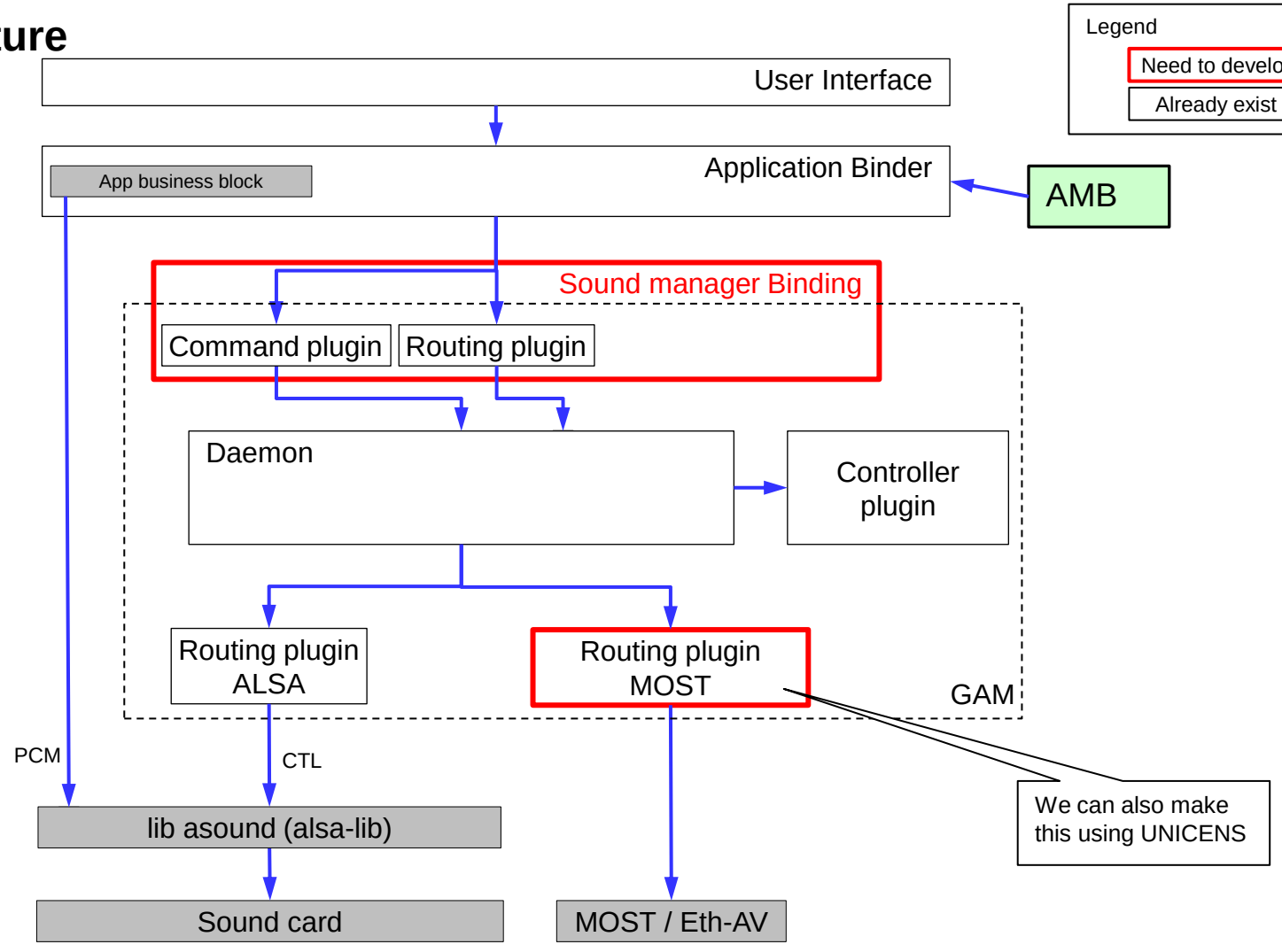
- **We already started PoC development to apply GAM to Binder**

Sound manager PoC - schedule -

- **PoC**
  - For CES2018, we are developing sound manager PoC with TOYOTA.
  - Schedule



Architecture



Sound manager PoC - interface -

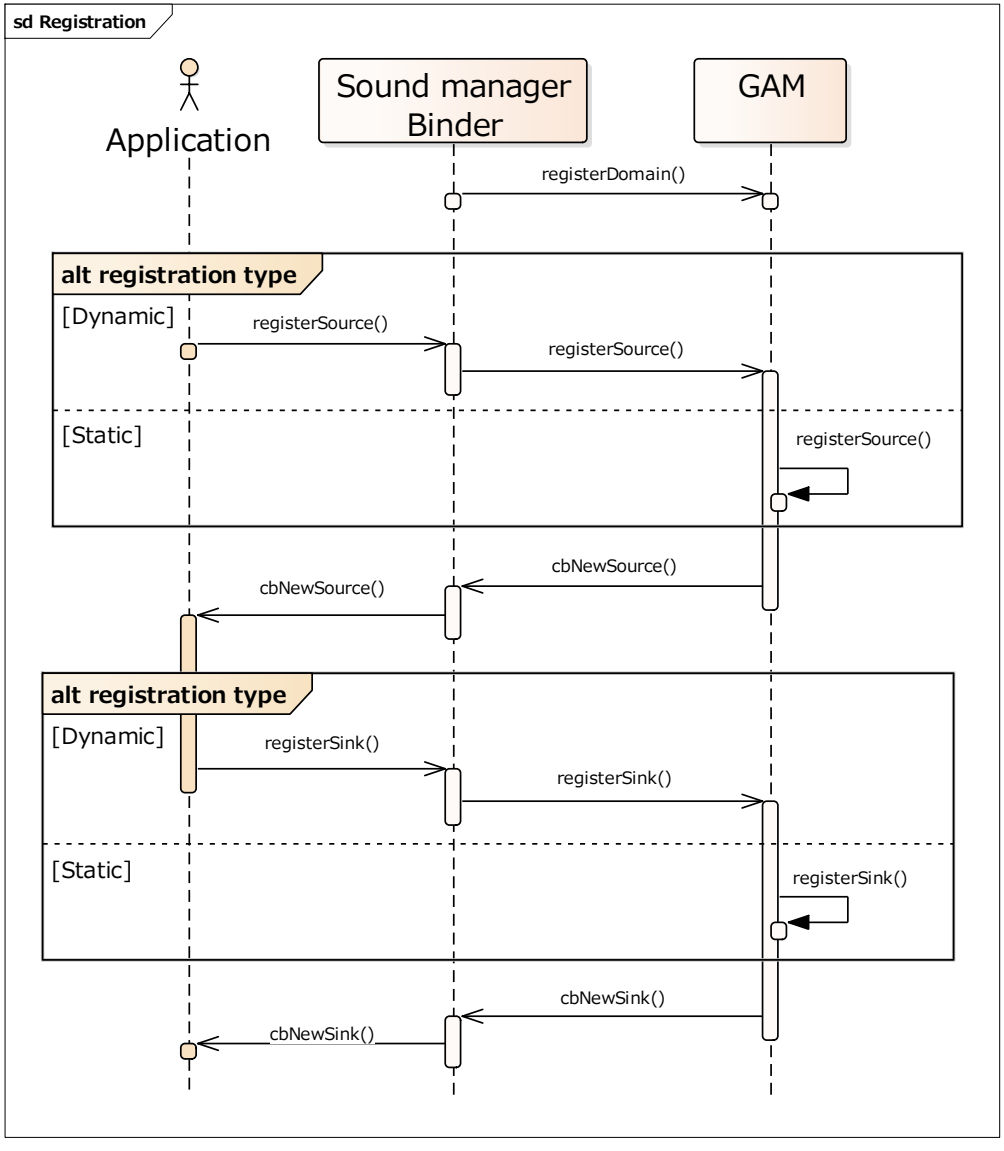
Advanced Driver Information Technology

Interface List

#	Interface	
1	connect (sourceID, sinkID, &mainConnectionID)	For command plugin
2	disconnect (mainConnectionID)	
3	setVolume (sinkID, volume)	
4	volumeStep (sinkID, volumeStep)	
5	setSinkMuteState (sinkID, muteState)	
6	getVolume (sinkID, &mainVolume)	
7	getListMainConnections (&listConnections)	
8	cbNewMainConnection (mainConnection)	
9	cbRemovedMainConnection (mainConnectionID)	
10	cbMainConnectionStateChanged (mainConnectionID, connectionState)	
11	cbVolumeChanged (sinkID, volume)	
12	cbSinkMuteStateChanged (sinkID, muteState)	
13	cbNewSource(source)	
14	cbNewSink(sink)	
1	asyncAbort (&handle)	For routing plugin
2	asyncConnect (&handle, &connectionID, sourceID, sinkID, connectionFormat)	
3	asyncDisconnect (&handle, connectionID)	
4	asyncSetSinkVolume (&handle, sinkID, volume, ramp, time)	
5	asyncSetSourceState (&handle, sourceID, state)	
6	ackConnect (handle, connectionID, error)	
7	ackDisconnect (handle, connectionID, error)	
8	registerSink (&sinkData, &sinkID)	
10	registerSource (&sourceData, &sourceID)	
12	hookInterruptStatusChange (sourceID, interruptState)	
13	hookSourceAvailablityStatusChange (sourceID, &availability)	
14	ackSetVolumes (handle, &listvolumes, error)	

Sound manager PoC - sequence -

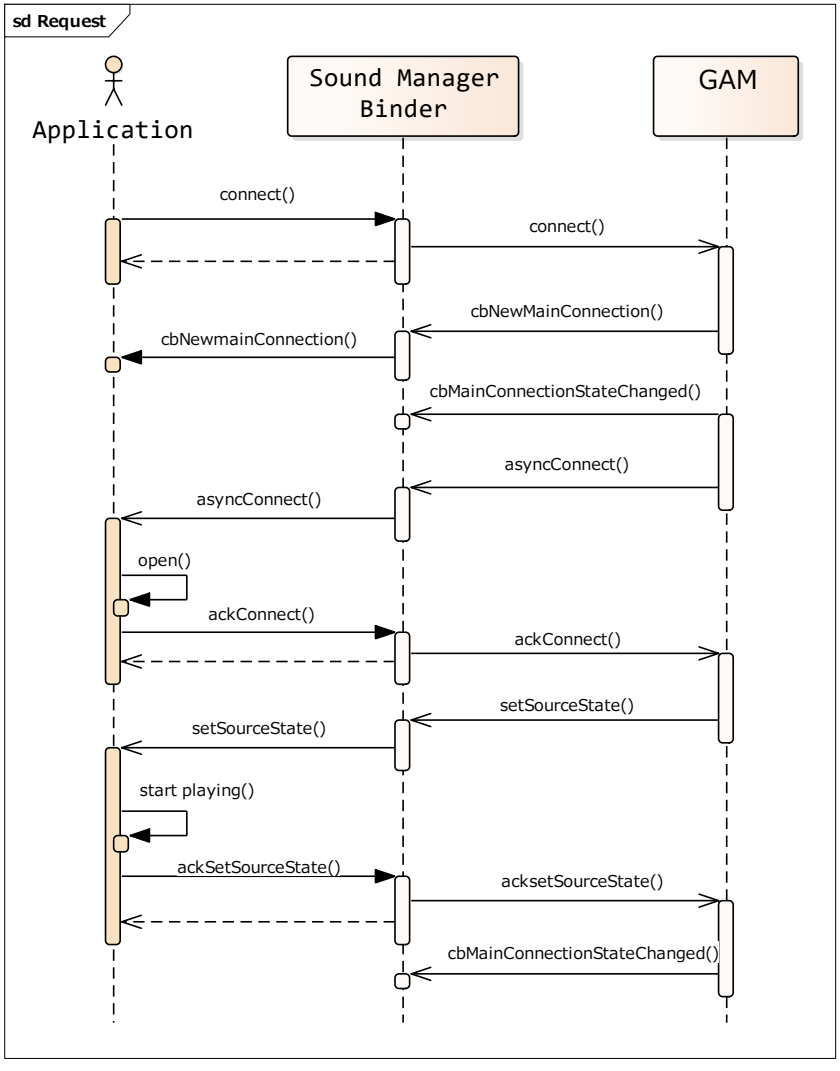
- **Sequence**

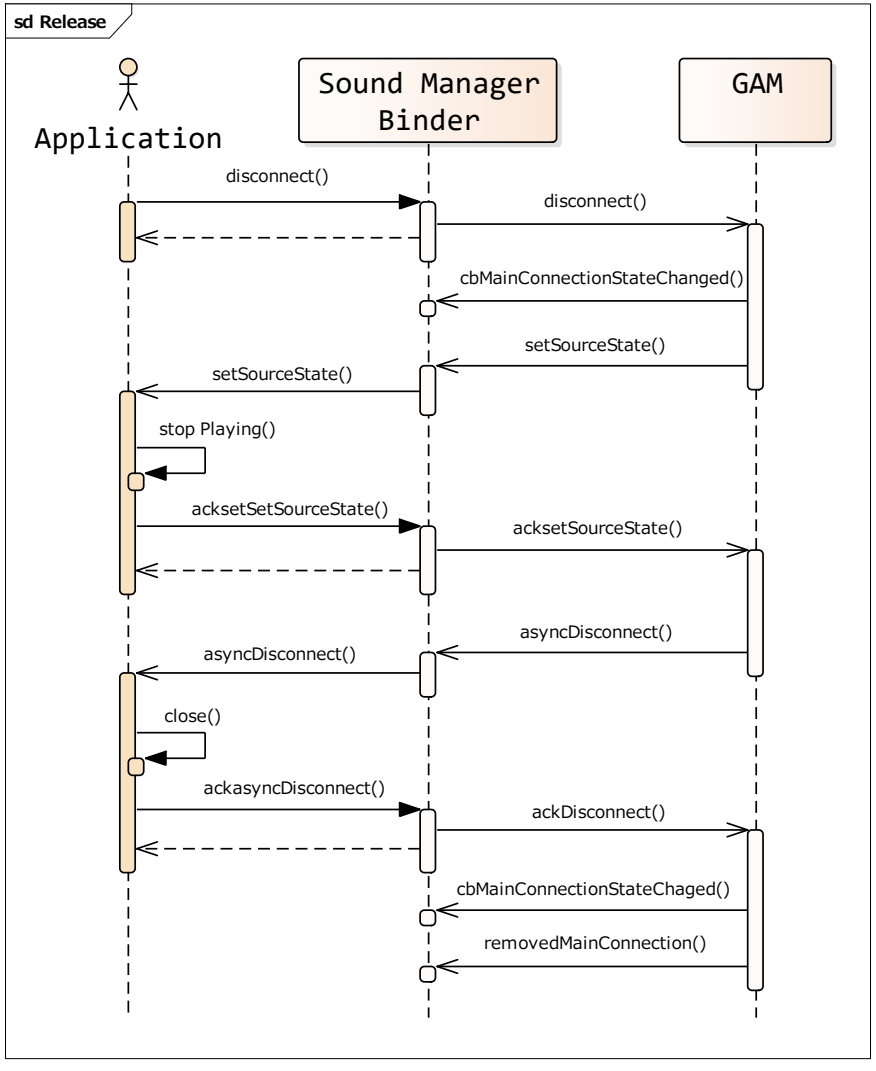


Sound manager PoC - sequence -

Advanced Driver Information Technology

Sequence





EOF

- **There are major 4 patterns of arbitration(policy) in automotive**
  1. The latter source win
     Discards the former source and output the latter source.

  2. The latter source win and the former source pause
     Pauses the former source and output the latter.

  3. The latter loose
     Continues former source and discards the latter source.

  4. The latter source is put on hold
     Continues former source and puts latter source on hold

Queuing management is required