AGL Audio Agent

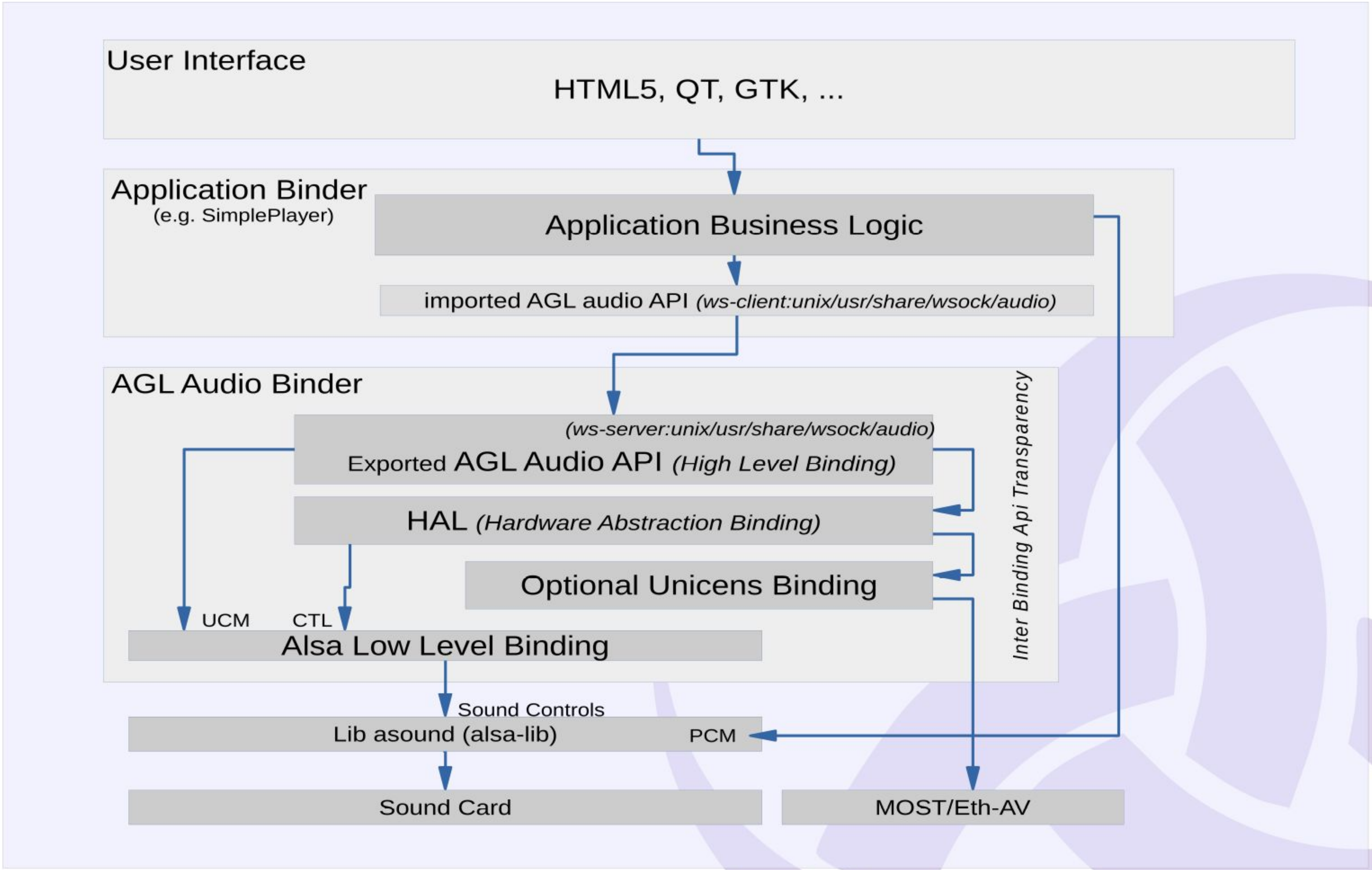
AGL Technical F2F@Microchip
March/2017

Fulup Ar Foll Lead Architect
fulup@iot.bzh

Goals

- Friendly API to developers
  - Normalized high level API (Application portability)
  - Independence to sound card hardware
  - AGL framework native service (API transparency + Security)
- Support of advanced sound card controls
  - Volume Rump Up/Down
  - Mixing / DSP
  - Etc.
- Capability to integrate non-alsa features
  - Microchip Unicens
  - Power On/Off
  - External players
  - Etc.
- Functional Channels Mixing
  - Multimedia
  - Navigation
  - Telephony
  - Notification
  - Etc.

Architecture


User Interface
HTML5, QT, GTK, ...
Application Binder
(e.g. SimplePlayer)
Application Business Logic
imported AGL audio API (ws-client:unix/usr/share/wsock/audio)
AGL Audio Binder
(ws-server:unix/usr/share/wsock/audio)
Exported AGL Audio API (High Level Binding)
HAL (Hardware Abstraction Binding)
Optional Unicens Binding
Inter Binding Api Transparency
UCM
CTL
Alsa Low Level Binding
Sound Controls
Lib asound (alsa-lib)
PCM
Sound Card
MOST/Eth-AV

Low Level Binding

- Expose ALSA native API (controls+UCM)
  - .name="getinfo", .info= "List All/One Sound Cards Info" },
  - { .name= "getctls", .info= "Get Controls" },
  - { .name= "setctls", .info= "Set Controls" },
  - { .name= "subscribe", .info= "Subscribe to events" },
  - { .name= "getcardid", .info= "Get CardId from its short/long name" },
  - { .name= "registerHal", .info= "Register Hal CardName/ApiPrefix" },
  - { .name= "ucmquery", .info= "Use Case Query" },
  - { .name= "ucmset", .info= "Use Case Set" },
  - { .name= "ucmget", .info= "Use Case Get" },
  - { .name= "ucmreset", .info= "Use Case Reset to Default" },
  - { .name= "ucmclose", .info= "Use Case Close Manager" },
- callbinder('alsacore','getctls', {devid:devid, numids:17})
- callbinder('alsacore','setctls', {devid:devid, numids:[{id:1,val:[50,50]}, {id:17,val:50,50}]})

Generic Exposure of Lib Alsa API through AGL Framework.

Hardware Abstraction Binding

```
STATIC alsaHalMapT  alsaHalMap= {
  { .alsa={.control=Master_Playback_Volume
       ,.numid=16,.group=OUTVOL,.values=1,.minval=0,.maxval= 87 ,.step=0,.acl=RW}
       , .info= "Master Playback Volume" },
  { .alsa={.control=PCM_Playback_Volume
       ,.numid=27,.group=PCMVOL,.values=2,.minval=0,.maxval= 255,.step=0,.acl=RW}
       , .info= "PCM Playback Volume" },
  { .alsa={.control=PCM_Playback_Switch
       ,.numid=17,.group=SWITCH,.values=1,.minval=0,.maxval= 1  ,.step=0,.acl=RW}
       , .info= "Master Playback Switch" },
  { .alsa={.control=Non_Alsa_Call
       ,.numid=12,.group=INVOL ,.values=2,.minval=0,.maxval= 31 ,.step=0,.acl=RW}
       ,.cb={callback=MyCustomFunction, handle=MyHandle}
       , .info= "Capture Volume" },
```

Normalize an AGL virtual Sound Card

High Level Binding

- Normalized API for application portability
- Handle client context

```
{ .name= "open",          .info= "Open a Dedicated SoundCard" },
{ .name= "close",         .info= "Close previously open SoundCard" },
{ .name= "setvolume",     .info= "Set Volume" },
{ .name= "getvolume",     .info= "Get Volume" },
{ .name= "subscribe",     .info= "Subscribe AudioBinding Events" },
{ .name= "ucmset",        .info= "Set Usecase and retrieve PCM value" },
```

Alsa UCM *(Use Case Manager)*

Mixing Sound Channels

```
aplay  -D plug:music  trio-divi.wav
speaker-test  -D plug:navi  -c 2 -twav
```

Selecting Use Cases

```
alsaucm -c "MySoundCard" list _verbs
alsaucm -c "MySoundCard" _verb HiFi
alsaucm -c "MySoundCard" _verb Navi
```

```
SectionVerb {
	EnableSequence [
		cset "name='MasterMusic' 80%"
	]
	DisableSequence [
		cset "name='MasterMusic' 60%"
	]
	Value {
		TQ "Music"
		OutputDspName "Multi Media"
		PlaybackPCM "plug:music"
	}
}
SectionDevice."Speaker" {
	Comment "Speaker"
}
```

Alsa Virtual Channels

```
pcm.SoftMixer {
    type dmix
    ipc_key 1024
    ipc_key_add_uid false
    ipc_perm 0600
    slave {
        pcm "hw:v1340"  #Jabra Solmate
        rate 44100
    }
}
```

```
pcm.music {
    type        softvol
    slave.pcm   "SoftMixer"
    control {
        name    "MasterMusic"
        card    0
    }
}
```

```
aplay -D plug:music ./htdocs/sounds/trio-divi-alkazabach.wav
speaker-test -D plug:navi -c 2 -twav
amixer -D hw:xxx cset name="MasterMusic" 80%
amixer -D hw:xxx cset name="MasterNavi" 100%
```

Almost Done Work

- Low Level Bindings (90%)
  - Introspection
  - Controls get/get
  - Events (e.g. volume was changed)
  - Use Case Manager
- Hardware Abstraction Binding (50%)
  - Get normalized controls
  - Callback hooking for non Alsa controls
- Hight Level Bindings (10%)
  - Template with basic set controls

Work To Be Done

- Low Level
  - Polish
  - Documentation (especially on UCM)
  - Security interface with Smack & Cynara
- HAL
  - Set capability
  - Event normalization
  - Add HAL for reference hardware (Renesas, Microchip)
- High Level
  - Agree on AGL AudioAPI
- Applications Sample
  - Finalize SimplePlayer (equivalement to aplay)
  - Implement a SimpleMixer (equivalent to amixer)
- Documentation
  - Configuration guide
  - Developer guide

Source Code

- http://github.com/iotbzh/audio-bindings
- http://github.com/iotbzh/audio-utils