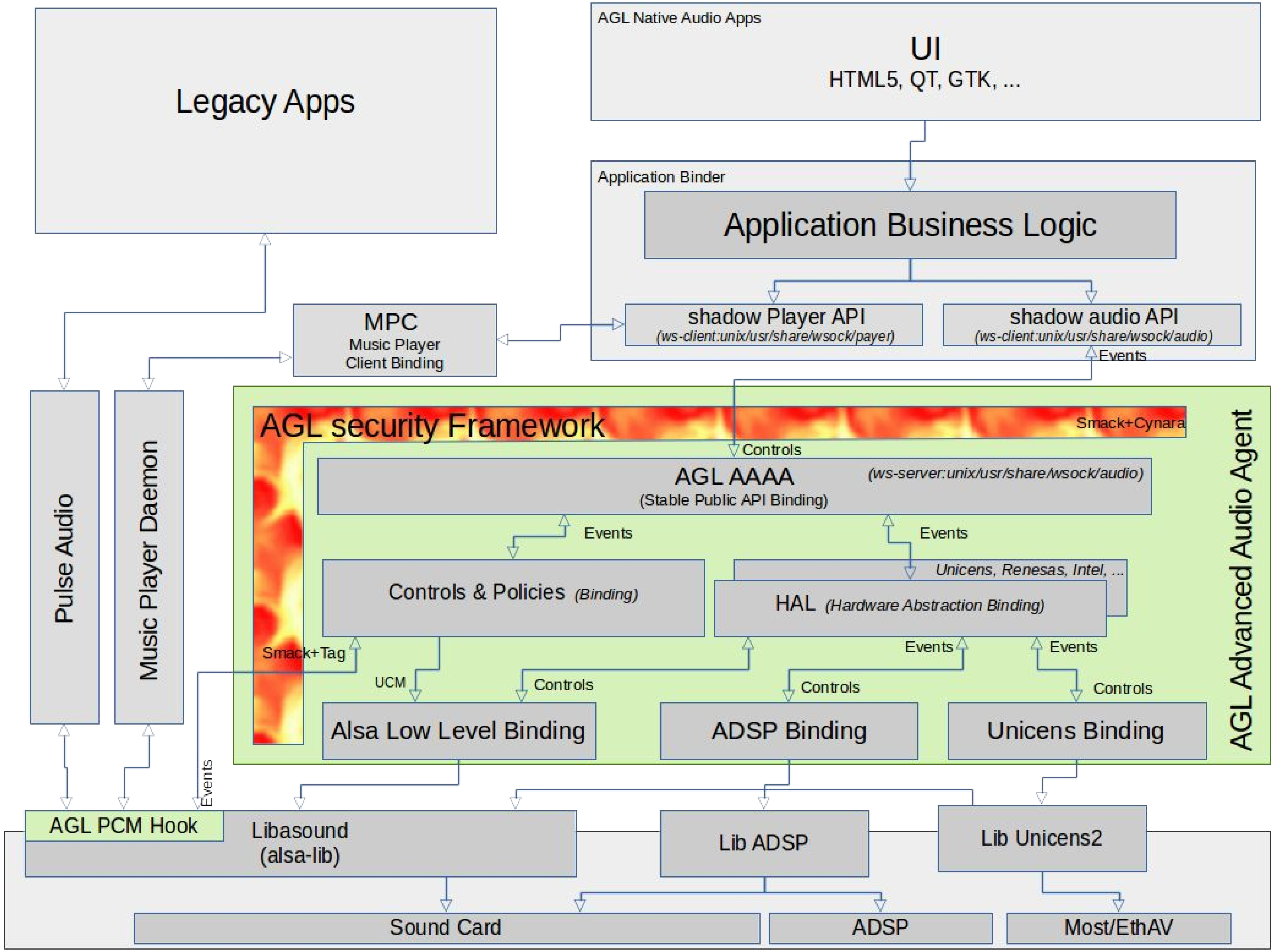
AGL High Level Audio API Design

August 30th, 2017

François Thibault, Audiokinetic

AGL Audio Architecture



AGL High Level Audio API Objectives

- Provide stable and **standardized access** to audio features for applications
  - Allow different underlying implementations of common API
- Provide **easily extensible** API
  - Expose single access point to current and future AGL audio capabilities
- Provide **application isolation**
  - Applications should not be able to control audio behaviors of others
    - This responsibility should entirely reside on policy enforcement

Request feedback on API proposal

High-level Audio Binding API Concepts

- Audio **roles** (e.g. entertainment, warning, communications, etc.)
- Audio **endpoints** (source and sink devices)
- Application audio **streams** and device **routings**
- Endpoints volume, **properties** (balance, eq,...) and **state** (mute,suspend,...)
- Sound **events** (e.g. HMI events, startup/ending sound, etc.)
- Audio **zones** (implicit in current version through endpoint selection)

Bring common high-level audio concepts to API

AGL High Level Audio Binding Features

- Audio device enumeration and monitoring (role support)
- Stream and device routings (automatic or explicit)
  - Provided with audio role and endpoint type
  - Customized according to concurrency behaviors and priorities config
  - Return appropriate PCM name to application
  - Return target device for volume/state/property changes
- Isolated endpoints volumes/properties/state changes
- Permissions scheme
  - none (monitoring) / stream / routing / sound event
  - Association of stream/routing resources with application ID

Define common features expected from applications

API Overview

- Endpoint enumeration, stream and routing management
  - GetSources / GetSinks → for explicit routing
  - StreamOpen / StreamClose → application streaming (e.g. media player)
  - AddRouting / RemoveRouting → device connection (e.g. handsfree)
- Endpoints (source or sinks)
  - Set/Get EndpointVolume → Absolute or step with optional ramp time
  - Set/Get EndpointProperties → Absolute or step with optional ramp time
  - Set/Get EndpointState → E.g. mute. May trigger audio policy actions
- Sound events
  - PostSoundEvent → Sound generation services
- Events
  - Endpoints volume/status/property changes (e.g. from policy application)
  - Endpoint availability changes
  - Audio streaming changes (start/stop/pause/resume, PCM name, etc.)
  - Stream/routing activity changes
  - ...

Timeline

- Audio Workshop (Sept 13-14)
  - https://wiki.automotivelinux.org/agl-distro/sep2017-audio-f2f
  - Discuss integration of high level API with other layers
  - POC definition and demo use case definition
  - …
- Dresden AMM (Oct 18-20)
  - Present API v1.0 to community
  - Demo against use case requirements
  - Show working reference implementation of high level binding
- CES 2018
  - TBD

Join us in Montreal and provide feedback!