

# deepin/UOS 操作系统架构之音频栈

徐小东

统信软件

2023-03-22

# 目录

- ① 音频栈概述
- ② 内核层
- ③ 用户层
- ④ 参考资料

## Section 1

# 音频栈概述

# 音频栈整体架构

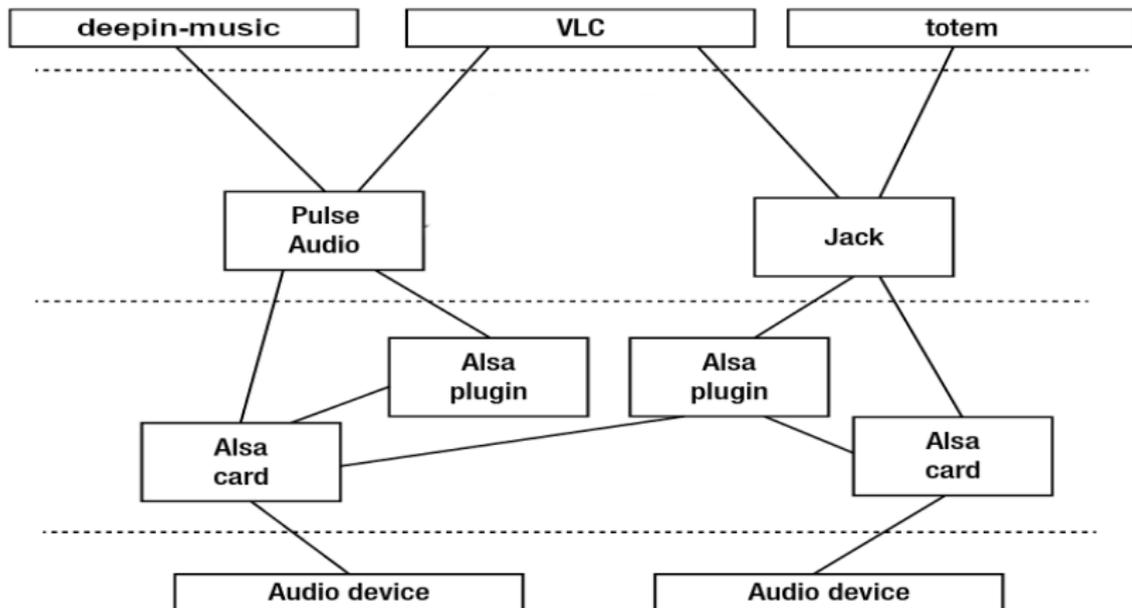


图 1: 音频栈整体架构

## 设备驱动

- ▶ 最底层是硬件：音频设备（声卡）
- ▶ OSS, Open Sound System
  - Unix 类操作系统的声音接口
  - 1993 ~ 1997 年间 Linux 上声音系统的唯一选择
  - OSS 后来走向闭源
  - 被 ALSA 取代
- ▶ ALSA, Advanced Linux Sound Architecture
  - 由 Linux 内核支持
  - 为声卡设备驱动提供 API
  - 声音服务器及高层 API 工作于 ALSA 及声卡设备驱动之上

# 声音服务器

- ▶ 提问：  
不同的应用程序如何同时使用声卡？

## ▶ 声音服务器

- 运行在设备驱动之上并管理复杂的任务
- 应用程序跟声音服务器通信，将合成的数字信号传递给设备驱动

## ▶ 不同的使用场景：

- 消费类音频系统：更友好的控制，例子：PulseAudio
- 专业音频系统：低延迟，例子：JACK
- 两者兼顾：例子：PipeWire

# 底层库

- ▶ libao  
跨平台的音频库
- ▶ GStreamer  
开源的多媒体处理框架
- ▶ FFmpeg  
全面、跨平台的录制、转换及流媒体解决方案

## 上层应用

- ▶ 深度音乐 (deepin-music)  
专注于本地音乐播放
- ▶ 深度影院 (deepin-movie)  
本地视频及流媒体播放
- ▶ VLC  
开源、跨平台的媒体播放器
- ▶ Audacity  
多轨音频编辑及录音，音频后期处理
- ▶ Ardour  
数字音频工作站：录音、混音、编辑

## Section 2

# 内核层

# ALSA

- ▶ Linux 内核的音频子系统
  - 提供声卡驱动
  - 支持的声卡<sup>1</sup>
- ▶ 目的
  - 支持声卡的自动配置
  - 处理系统中的多个声音设备
- ▶ 主要特性
  - 基于硬件的 MIDI 合成
  - 多通道硬件混音
  - 双向同时通信
  - 多处理器友好、线程安全的设备驱动程序
- ▶ 拓扑架构

---

<sup>1</sup><https://www.alsa-project.org/main/index.php/Matrix:Main>

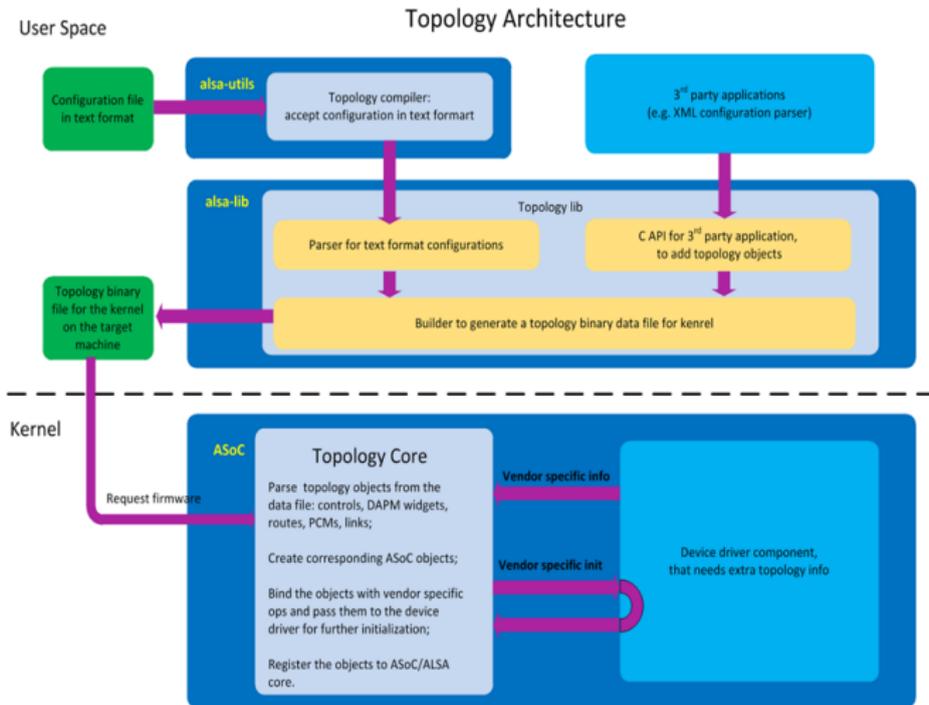


图 2: ALSA 拓扑架构

- ▶ 设备接口： /dev/snd
  - pcmC?D?p: pcm 播放设备
  - pcmC?D?c: pcm 捕获设备
  - controlC?: 控制设备
  - hwC?D?: hwdep 设备
  - seq: 音序器设备
  - timer: 定时器设备
- ▶ 驱动状态及配置接口  
/proc/asound
- ▶ 配置文件
  - /usr/share/alsa
  - /etc/alsa

## Section 3

# 用户层

# ALSA

- ▶ 两部分：
  - 与声卡通信的 API: `libasound*`
  - 用户层应用程序集: `alsa-utils`
- ▶ `alsamixer`  
选择声卡并控制其接口
- ▶ `alsactl`  
用于 ALSA 配置的控制程序
- ▶ `speaker-test`  
测试声卡输出
- ▶ `aplay/arecord`  
播放或录制
- ▶ `alsa-info`  
收集 ALSA 信息

# PulseAudio 概述

- ▶ 跨平台且具备网络能力的声音服务器
- ▶ 在内核音频组件和应用程序之间充当代理的角色
- ▶ 接受多个音源输入，混音并播放
- ▶ 针对消费类音频而设计
- ▶ 在桌面、笔记本、移动设备上更易使用声音
- ▶ 低延迟不是其设计目标，不适合专业级音频

# PulseAudio 架构

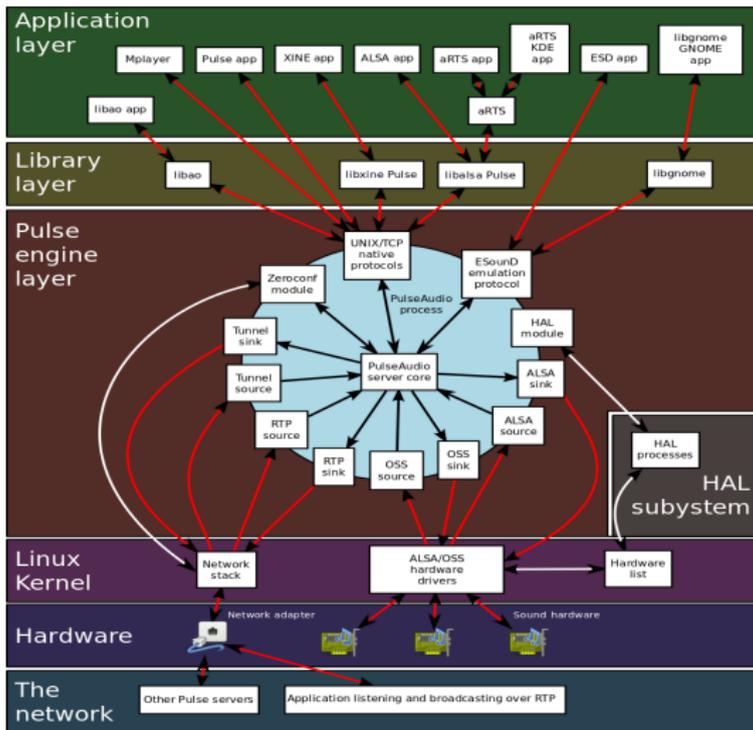


图 3: PulseAudio 架构

# PulseAudio 组成

- ▶ daemon  
后台服务
- ▶ 服务端及客户端库
  - libpulsecore
  - libpulsecommon
  - libpulse
  - libpulse-simple
- ▶ 服务端扩展  
modules
- ▶ 工具

# PulseAudio 工具

- ▶ pavucontrol  
控制音频设备的输入和输出音量
- ▶ parec/paplay/pacat  
录制及播放声音文件
- ▶ pactl/pacmd  
控制运行的 PulseAudio 服务器
- ▶ 配置文件
  - /etc/pulse
  - \$HOME/.config/pulse

# JACK

- ▶ JACK Audio Connection Kit
- ▶ 专业级的声音服务器
- ▶ 在音频和 MIDI 数据间提供实时、低延迟的连接
- ▶ Linux 上专业音频软件的关键基础架构和默认标准
- ▶ 实现
  - jack1 (C): 维护模式
  - jack2 (C++)
- ▶ 组成
  - daemon: 后台服务
  - API: 构建信号处理及音乐制作应用

# PipeWire 概述

- ▶ 用于处理音频及视频流的服务器
- ▶ 提供统一的框架访问多媒体设备
- ▶ 上层应用通过其 API 来使用硬件
- ▶ 主要特点
  - 用最低延迟捕获及播放音频和视频
  - 针对音频和视频的实时多媒体处理
  - 多进程架构允许应用程序共享多媒体内容
  - 无缝支持 ALSA、PulseAudio、JACK、GStreamer 应用程序
  - 对沙盒应用（如 Flatpak）提供支持

# PipeWire 架构

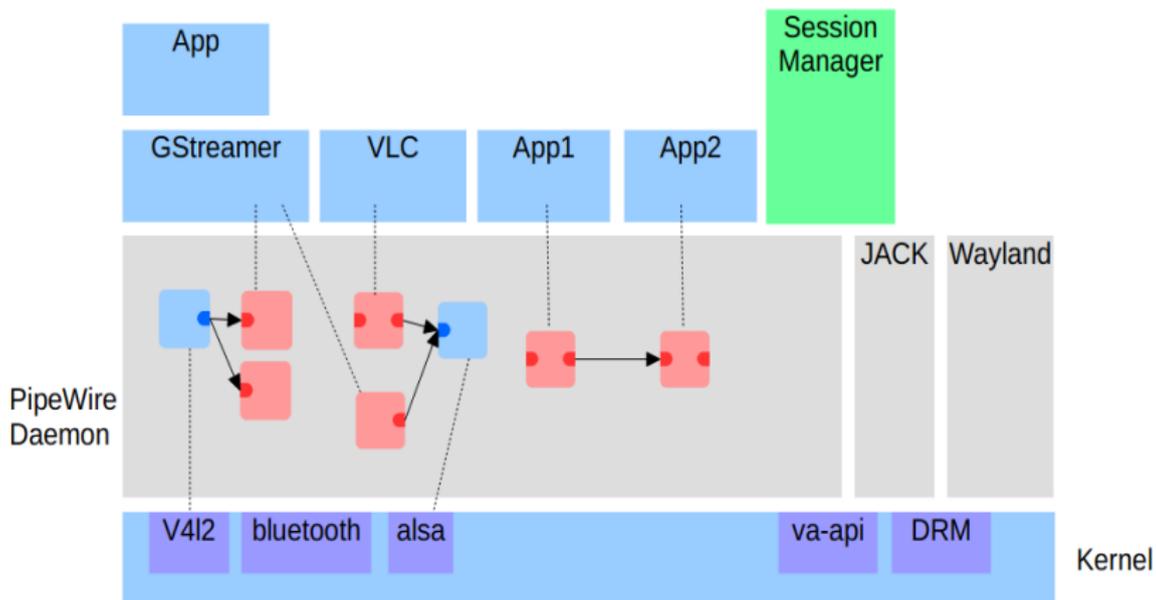


图 4: PipeWire 架构

# PipeWire 组成

- ▶ daemon: 后台服务
- ▶ Session Manager: 会话管理器, 管理 daemon 中的对象
- ▶ libpipewire: 开发库
- ▶ 兼容层
  - pipewire-pulse: 兼容 PulseAudio
  - libjack: 兼容 JACK
  - gstreamer-pipewire: 兼容 GStreamer
- ▶ 工具
  - pw-cli: 命令行界面
  - pw-dump: 转存 PipeWire 对象
  - pw-mon: 监视器
  - pw-cat: 播放录制媒体
  - pw-play: 仅用于播放

# GStreamer 概述

- ▶ 基于 pipeline 的多媒体框架
- ▶ 提供的媒体处理组件
  - 音频、视频播放
  - 录制
  - 流媒体
  - 编辑

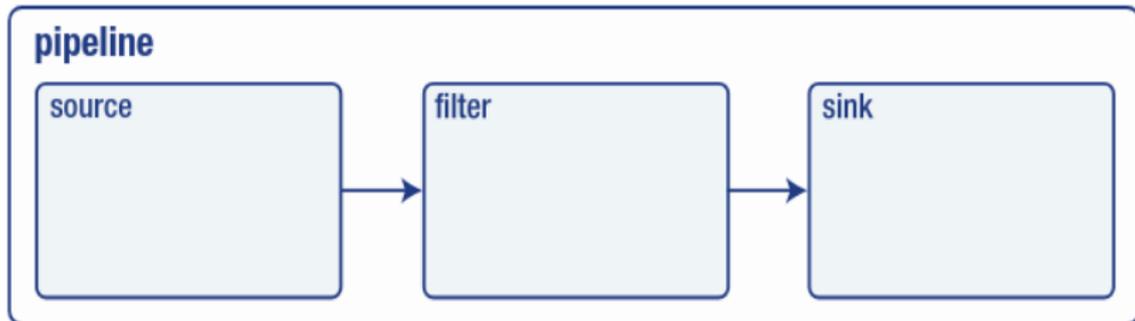


图 5: GStreamer pipeline 模型

# GStreamer 架构

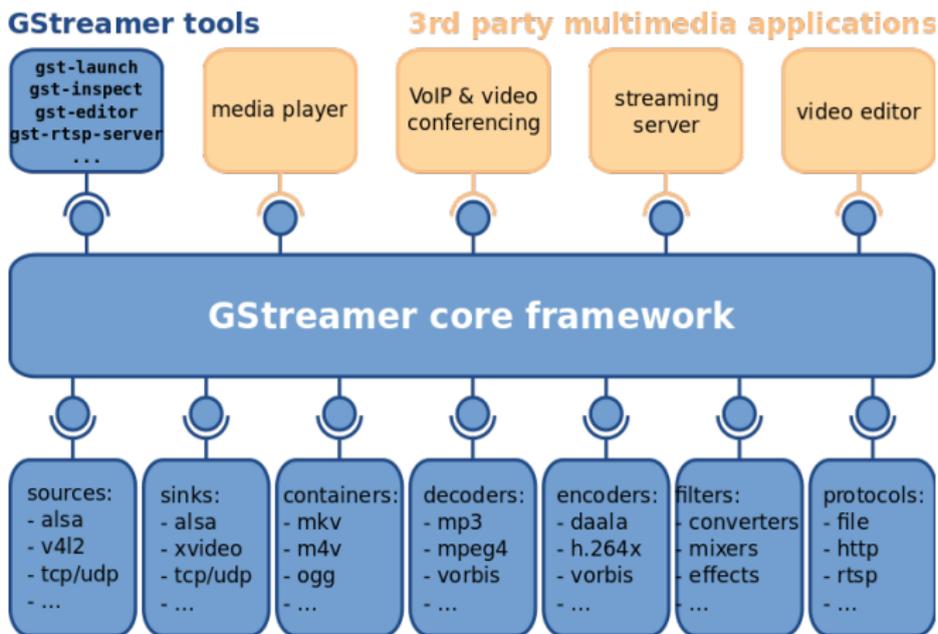


图 6: GStreamer 架构

- ▶ 插件化架构，以共享库实现
- ▶ 插件分组
  - Good：高质量插件，LGPL 许可
  - Bad：可能接近好质量，但缺少某些东西
  - Ugly：可能存在分发问题的插件
- ▶ 工具
  - gst-inspect：获取 GStreamer 插件信息
  - gst-discoverer：获取多媒体文件的信息
  - gst-device-monitor：获取设备信息
  - gst-play：播放多媒体文件
  - gst-launch：构建处理媒体数据的命令管道

# FFmpeg 概述

- ▶ 处理视频、音频及其它多媒体文件和流媒体的框架
  - 解码
  - 编码
  - 转码
  - mux/demux
  - 网络协议：RTMP、RTSP、HLS
- ▶ 支持 x86、ARM、MIPS、PPC 等 CPU
- ▶ 支持广泛地多媒体格式<sup>2</sup>
- ▶ 用于 YouTube 和 Bilibili 的核心处理过程

---

<sup>2</sup>[https://en.wikipedia.org/wiki/FFmpeg#Supported\\_codecs\\_and\\_formats](https://en.wikipedia.org/wiki/FFmpeg#Supported_codecs_and_formats)

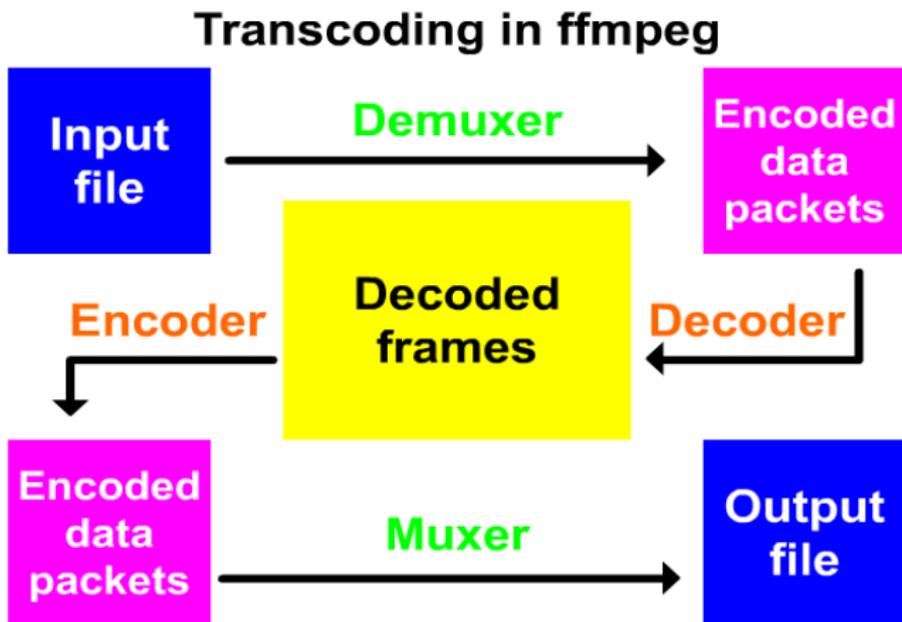


图 7: FFmpeg 转码

# FFmpeg 组件

## ▶ 命令行工具

- ffmpeg: 转换音频或视频格式  
`ffmpeg -i file.ogg file.mp3`
- ffplay: 简单的媒体播放器
- ffprobe: 显示媒体信息

## ▶ 主要开发库

- libavcodec: 音频、视频编解码库
- libavformat: 音频、视频容器 muxer 和 demuxer 库
- libavfilter: 增强、编辑滤镜的库
- libswscale: 视频图像缩放、colorspace/pixelformat 转换
- libavdevice: 支持的输入和输出设备
- libswresample: 音频重采样、采样格式转换
- libavutil: 简化编程的函数, 如: 核心多媒体工具

# libVLC

- ▶ 核心引擎及多媒体框架接口
- ▶ 插件化模块，可在运行时载入
- ▶ 支持特性
  - 播放多种媒体文件格式
  - 硬解码
  - 网络浏览（SMB、FTP、SFTP、NFS）

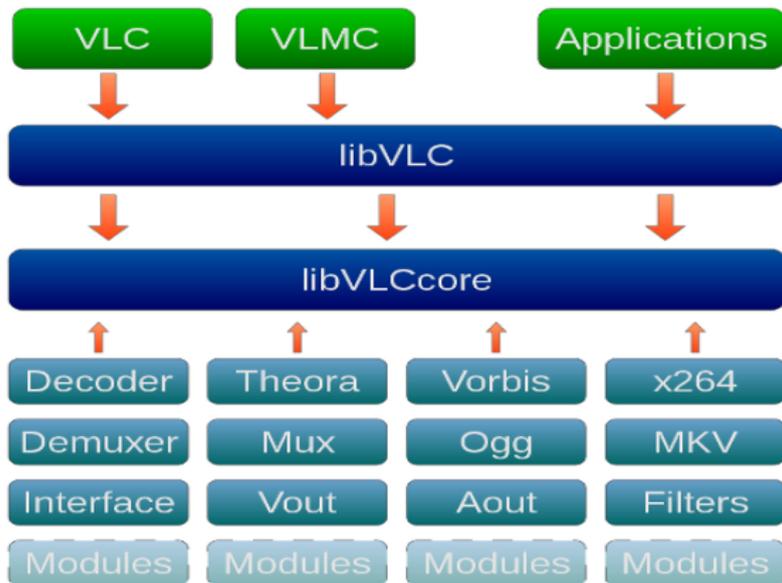


图 8: libVLC 架构

# deepin-music

- ▶ FFmpeg: 音频数据检测、分析
  - libavcodec
  - libavformat
  - libavutil
- ▶ VLC: 音频播放、均衡器
  - libvlccore
  - libvlc

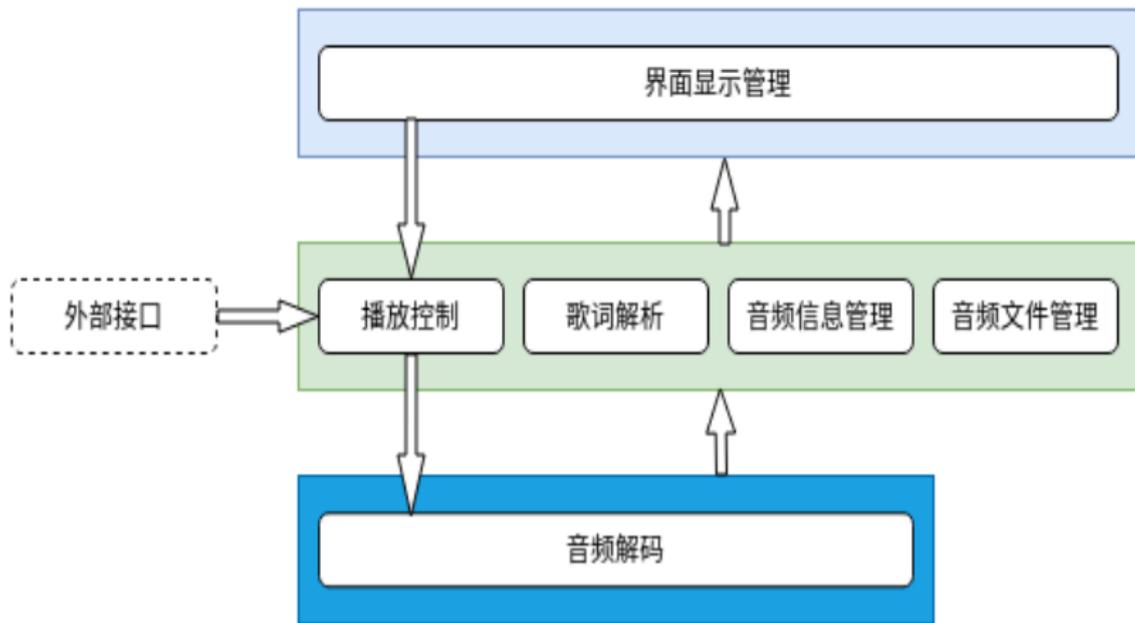


图 9: deepin-music 架构

## Section 4

参考资料

## 参考资料

- ▶ ALSA
- ▶ PulseAudio
- ▶ JACK
- ▶ PipeWire
- ▶ libao
- ▶ Wikipedia: GStreamer
- ▶ FFmpeg
- ▶ FFmpeg Basics
- ▶ libVLC
- ▶ deepin-music 概要设计说明书