ICS 号：ICS 33.160.25
中国标准文献分类号：M74

UHD World Association
世界超高清视频产业联盟

世 界 超 高 清 视 频 产 业 联 盟 标 准

T/UWA xxx.x-202x

三维声技术规范 第 2-3 部分：应用指南
智能终端系统软件框架设计

3D Audio Technology Specification Part 2-3:

Application Guide - Systems Software Framework Design for Smart Terminals

(点击此处添加与国际标准一致性程度的标识)

（本草案完成时间：2026-4-9）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

××××-××-××发布　　××××-××-××实施

世界超高清视频产业联盟

目　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定编制。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由世界超高清视频产业联盟提出并归口。

本文件主要起草单位：OPPO广东移动通信有限公司、华为技术有限公司、马栏山音视频实验室、北京声响节拍科技有限公司、深圳市腾讯计算机系统有限公司、荣耀终端股份有限公司、海信视像科技股份有限公司。

本文件主要起草人：林松、来航曼、高原、朱梦尧、马学睿、周凯旋、李大龙、杨枭、张宏伟、王文江、姚永明、高烨、田立、徐良、周梦蝶。

引言

本文件的发布机构提请注意，声明符合本文件时，可能使用以下涉及的相关专利：

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本文件的发布机构保证，其愿意同任何申请人在合理且无歧视的条款和条件下，就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案，相关信息可以通过以下联系方式获得：

联 系 人：

通讯地址：

邮政编码：

电子邮件：

电　　话：

传　　真：

网　　址：

请注意除上述专利外，本文件的某些内容仍可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

三维声技术规范 第2-3部分：应用指南

智能终端系统软件框架设计

1 范围

本文件描述了智能终端支持三维声的系统能力要求、系统模块要求和系统软件框架设计思想，提供了基于智能终端操作系统的三维声软件框架设计指南。

本文件适用于智能终端（手机/平板/智能电视/XR 设备/车载等消费级产品）操作系统侧三维声采编播能力的设计和实现，同时作为开放能力提供给开发者，用于开发智能终端三维声相关应用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 33475.3 信息技术 高效多媒体编码 第3部分：音频

T/UWA 009.1-2023 三维声技术要求 第1部分：编码、分发与呈现

T/UWA 009.2-2-2025 三维声技术规范 第2部分：应用指南 媒体格式

T/UWA 009.3-2-2023 三维声技术规范 第3-2部分：技术要求和测试方法 便携式数字设备

3 术语和缩略语

3.1 术语

下列术语和定义适用于本文件。

3.1.1

位流 bitstream

用作数据编码表示的有一定次序的一组比特。

3.1.2

编码 coding

读入音频采样值，并产生一个符合本文件的有效位流。

3.1.3

编码器 coder

编码处理的实体。

3.1.4

编码位流 coded bitstream

音频信号的编码表示。

3.1.5

声对象 object

被感知为一个整体的声音或由一个声源发出的独立于环境的声音。

3.1.6

解码 decoding

读入编码位流，并输出音频采样值。

3.1.7

解码器 decoder

解码处理的实体。

3.1.8

声道 channel

声音在录制或播放时在不同空间位置采集或重放的相互独立的音频信号。

3.1.9

双声道立体声 stereo audio

一种音频格式，使用两个声道承载有一定相位关系或者幅度关系或者相位和幅度混合关系的音频信号，通常通过位于听音者前方的两个对称的扬声器重放，带给听音者更宽的声场感觉。

3.1.10

环绕声 surround audio

一种音频格式，使用多个声道承载构成完整音频内容的多路音频信号，通过位于听音者耳部高度层的环绕听音者的多个扬声器重放，给听音者带来被环绕的声场感觉。

3.1.11

三维声 3D Audio

一种音频格式，多个声道承载构成完整音频内容的多路音频信号，通过环绕听音者的位于不同高度层的多个扬声器直接重放，或经过渲染或映射后重放，提供更高的声像空间解析度，并给听音者带来沉浸式的声场感觉。

3.1.12

元数据 metadata

描述音频数据的数据。

3.1.13

渲染 rendering

将给定的音频传输格式转换为适用于终端扬声器耳机配置的、可直接重放的音频格式的过程。

3.1.14

扬声器渲染 speaker rendering

将音频信号转换为特定配置的扬声器重放信号的过程。

3.1.15

双耳渲染 binaural rendering

将音频信号转换为双耳重放信号的过程。

3.2 缩略语

AOSP：安卓开放源代码项目（Android Open Source Project）

BRIR：双耳房间脉冲响应（Binaural Room Impulse Response）

DRM：数字版权管理（Digital Rights Management）

FOA：一阶球谐分解（First Order Ambisonics）

HAL：硬件抽象层（Hardware Abstraction Layer）

HOA：高阶球谐分解（Higher Order Ambisonics）

HRIR：头相关脉冲响应（Head Related Impulse Response）

HRTF：头相关传递函数（Head Related Transfer Function）

PCM：脉冲编码调制（Pulse Code Modulation）

PTS：显示时间戳（Presentation Time Stamp）

SDK：软件开发工具包（Software Development Kit）

VBAP：矢量基振幅平移（Vector Base Amplitude Panning）

XR：扩展现实（Extended Reality）

4 系统能力要求

从三维声技术整体视角来看，终端系统至少应具备三维声的关键特性支持能力：

- 音频信号的支持能力，包括单声道、立体声、环绕声、三维声
- 音频信号和元数据的编解码能力
- 音频信号的渲染能力，包括扬声器渲染和双耳渲染

由此拆解，系统如需具备基本的三维声播放能力，则需引入：

- 特性集 1：

——音频播放相关元数据处理能力及应用接口
——位流容器读取和位流解码能力
——实时双耳渲染能力（HRIR+BRIR、头部跟踪、AI 渲染...）
——播放能力拓展（扬声器输出、个性化 HRTF....）

进一步扩展到消费级采编能力，则需额外引入：

- 特性集 2：

——位流编码和位流容器写入能力
——实时三维声采集能力
——环境效果渲染能力
——元数据效果处理能力
——服务离线调用能力

针对场景拓展需求，系统可以提供额外的能力支持：

- 特性集 3

——车载互联场景，支持车机数据流转
——互动游戏场景，支持游戏引擎对接
——XR 场景：支持三维声实时生成
系统软件框架参考设计示例图如图 1 所示。

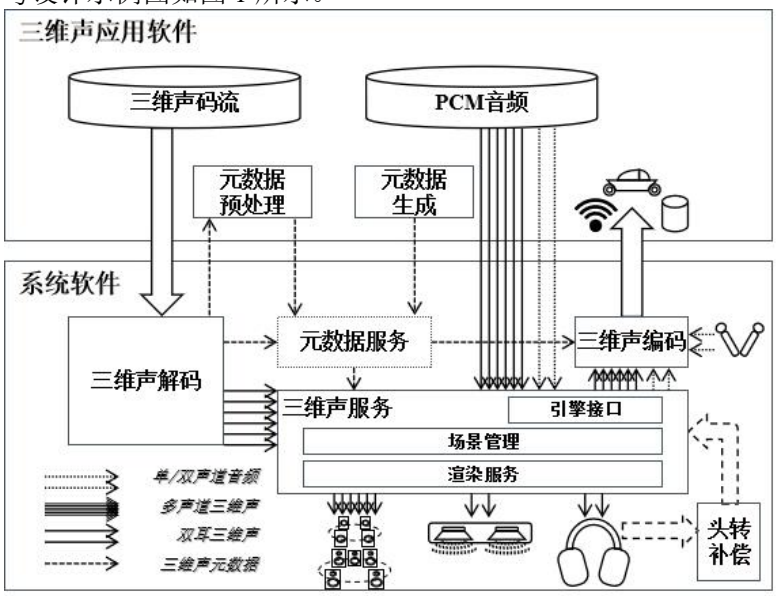


图 1　智能终端三维声系统软件框架参考设计示例图

5　系统能力接口（以安卓系统为例）

5.1　系统对外接口能力

能力检测：
　　OSpatialAudio.isAvailable()　判断当前系统是否支持三维声。
开关控制：
　　OSpatialAudio.setEnabled(boolean)　开启/关闭对当前应用的三维声支持。
版本查询：
　　OSpatialAudio.getVersion()　获取当前系统三维声能力和软件版本。
播放侧扩展：
　　OAudioTrack.setSpatialTrackType()　设置三维声类型；
　　OAudioTrack.writeExt()　下发 PCM 音频数据和元数据。
系统对外接口以 SDK 形式提供。

5.2　接口调用建议

- 类型设置建议：三维声类型应在开始播放前设置，避免播放过程中频繁切换。
- 调用顺序建议：初始化能力 → 设置三维声类型 → 开始播放 → 写入数据。
- 兼容性建议：当 isAvailable() 返回 false 或 SDK 调用失败时，应禁用三维声数据下发并回退至普通音频播放流程。
- 安全校验：建议应用侧做好调用时参数校验与异常处理。

5.3　三维声控制接口

类路径：android.media.OSpatialAudio

5.3.1　构造函数

接口	参数说明	返回值	功能说明
public OSpatialAudio (@NonNull AudioManager am)	am：安卓音频管理器实例	实例化对象	初始化三维声控制接口对象

5.3.2 isAvailable

接口	参数说明	返回值	功能说明
public boolean isAvailable()	无	true 支持 false 不支持	判断当前系统是否支持三维声

5.3.3 setEnabled

接口	参数说明	返回值	功能说明
public void setEnabled(boolean enabled)	true 开启 false 关闭	无	开启/关闭对当前应用的三维声支持

5.3.4 getVersion

接口	参数说明	返回值	功能说明
public String getVersion()	无	版本号	获取当前系统三维声能力和软件版本

5.4 音频数据下发接口

类路径：android.media.OAudioTrack

5.4.1 setSpatialTrackType

接口	参数说明	返回值	功能说明
public void setSpatialTrackType(int type)	传入类型参数与 AudioVivid typeLabel 表述一致	无	设置当前音轨的三维声类型

5.4.2 writeExt

接口	参数说明	返回值	功能说明
public int writeExt(@NonNull byte[] audioData, int sizeInBytes, int metadataOffset)	audioData (byte[])：包含 PCM 音频数据+元数据的连续字节流 sizeInBytes (int)：PCM 音频数据+元数据总长度 metadataOffset (int)：元数据起始偏移	实际写入字节数	下发 PCM 音频数据和元数据

6 系统模块要求

6.1 元数据服务模块

在影音播放这类场景中，应用下发包含三维声元数据的原始位流，经过系统媒体解封装服务得到元数据并回传给应用；应用对元数据进行预处理，并通过开放接口传递给系统侧。此处的应用预处理，主要涵盖一些与渲染无关信息的预处理，如语种等等；或是应用希望对系统隐藏部分元数据信息，可以将其抹去或置标志位确保系统不会处理该类数据。

对于智能终端上的互娱/编创场景，应用实时产生部分处理参数或完整格式的元数据并传递给系统，针对必需参数缺失的场景，系统会使用默认值进行补齐并统一到元数据标准格式后进行下一步处理。

在具体实现方案中，元数据服务一般会作为三维声服务的一部分，对外提供统一的接口。服务的核心功能是针对不同标准的元数据应用适当的处理/转换逻辑，以将合适的参数用于声音场景的渲染。可通

过调整元数据本身来实现最终渲染效果的变化，如实现轨迹生成、均衡优化、场景化渲染等功能。

6.2 三维声编/解码器模块

三维声编解码能力在安卓平台可以集成在 MediaCodec 框架中并对应用开放。解码服务提供 AudioVivid 位流/元数据解封装和 AudioVivid 解码能力，解码得到的元数据通过开放接口回传给应用，或在应用要求下可以直接传递给三维声渲染器。编码服务提供 AudioVivid 位流/元数据封装和 AudioVivid 编码能力，可以对渲染前后的音频数据进行编码并与元数据一并保存。

图 2 安卓平台三维声编/解码器参考集成方案

6.3 三维声服务模块

完成音频流的渲染，并提供算法封装接口以适配不同的渲染器和音频算法子模块。

- 场景管理

维护不同音轨/声道所关联的配置信息和渲染参数，并提供传感器数据的实时融合能力，如耳机头部跟踪，或者是未来各种可能的环境实时感知能力等等。

- 渲染服务

通过算法封装接口配置并调用通用渲染器和/或音频算法子模块完成声音渲染。子模块包括 VBAP、FOA/HOA、串扰消除等等。

7 系统软件框架设计指南

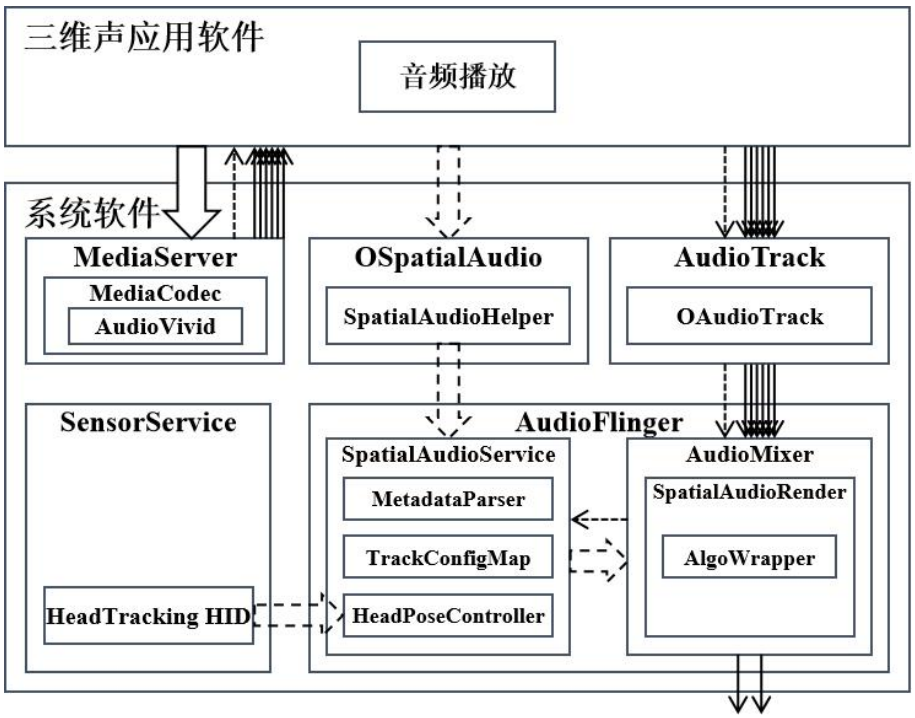
7.1 服务架构

7.1.1 设计原则

本文件在框架设计上遵循以下原则：优先最大化复用 AOSP 操作系统原生音频组件与接口，降低定制维护成本；满足三维声采编播及开放能力的使用要求；保持模块边界清晰并预留扩展点，便于后续能力演进；在保证一致体验的前提下兼顾与既有音频通路及相关三维声技术路线的兼容性。

7.1.2 概述

本文件以 AudioVivid 标准为基础，以常见的手机操作系统为例，同时考虑了和其它三维声技术的兼容性以及其它操作系统的可移植性。相关软件框架基于安卓 10.0 AOSP 版本基础上实现。支持章节 4 所定义特性集的 1a、1b、1c 项，同时计划在未来版本更新时支持更多的特性项。



MediaServer：AOSP 系统播放模块接口，包括解码和播放
OSpatialAudio：三维声系统控制接口模块
AudioTrack：AOSP 系统音频播放接口，仅支持播放 PCM 数据，不包含解码
OAudioTrack：三维声扩展 AudioTrack 模块，支持 PCM 数据和元数据下发
SensorService：AOSP 系统 Sensor 模块，支持 HeadTracking 设备
AudioFlinger：AOSP 系统音频播放引擎，负责执行音频播放的服务
AudioMixer：AOSP 系统音频混音模块，负责 Track 的重采样、混音等
SpatialAudioRender：三维声渲染框架模块，负责三维声 Track 渲染逻辑实现
AlgoWrapper：三维声渲染算法，负责三维声 Track 渲染效果
SpatialAudioService：三维声服务框架，负责接收元数据、TrackConfig、头转数据等
MetadataParser：三维声元数据解析模块
TrackConfigMap：三维声 Track 和元数据映射关系
HeadPoseController：头部跟踪数据控制模块

：未解码的音频数据流
：音频接口控制流
：解码后的 PCM 音频数据流
：三维声元数据数据流

图 3　服务架构参考方案

7.1.3　音频框架

1、 扩展 AudioTrack 接口（OAudioTrack）
该接口原本仅用于应用写入音频数据，扩展后可以支持同步写入元数据。
2、 改造 AudioMixer 模块
保证在系统音频框架做混音前，完成三维声的正确渲染。
3、 新增 SpatialAudioService 服务
实现场景管理、元数据处理等三维声/元数据服务核心特性。

7.1.4　媒体服务

1、 改造 MediaCodec 模块
改造后可支持 AudioVivid 的正确解码。

2、 新增 OSpatialAudio 服务
三维声业务对外统一的 SDK，可用来查询三维声相关服务信息。

7.2 处理流程

7.2.1 正常播放流程

- 初始化阶段
应用通过 OSpatialAudio 获取三维声的相关能力；OSpatialAudio 通过 binder 和 SpatialAudioService 服务通信，获取相应的能力和规格。
- 播放阶段
应用选择三维声框架播放时，对于每帧 AudioVivid 数据，需要先调用 MediaCodec 的 AudioVivid 解码器进行解码后获得 PCM 音频数据和元数据，再通过 OAudioTrack 提供的扩展接口写入上述数据。AudioFlinger 中接收到上述数据后，解析元数据并融合头转角度数据后渲染声对象和声床；渲染完成后写入底层 HAL 和驱动。如此遍历所有音频帧后完成整个播放任务。

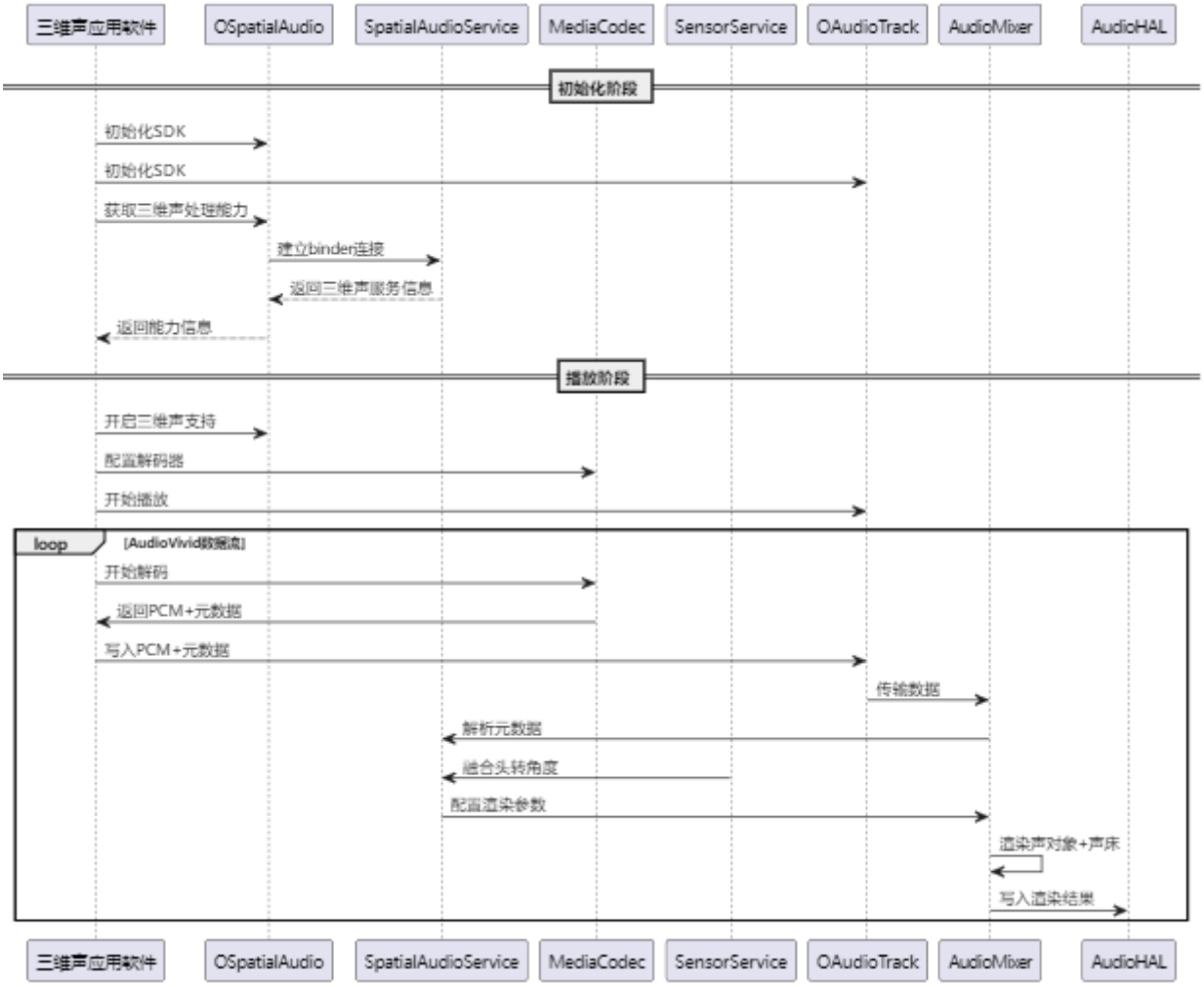


图 4 正常播放流程示例

7.2.2 音画同步流程

- 获取当前音频播放时间
应用可以获取当前播放的音频时间，具体实现可能是音频通路延迟、播放头位置换算、已写样本数+采样率等，不同应用之间有所差异。
- 同步基准
用上述“当前音频播放时间”作为基准，将视频 PTS 与之比较。视频分支落后过多则丢帧；否

则按 PTS 或目标时间渲染到 Surface（如 releaseOutputBufferAtTime 或延迟后渲染）。

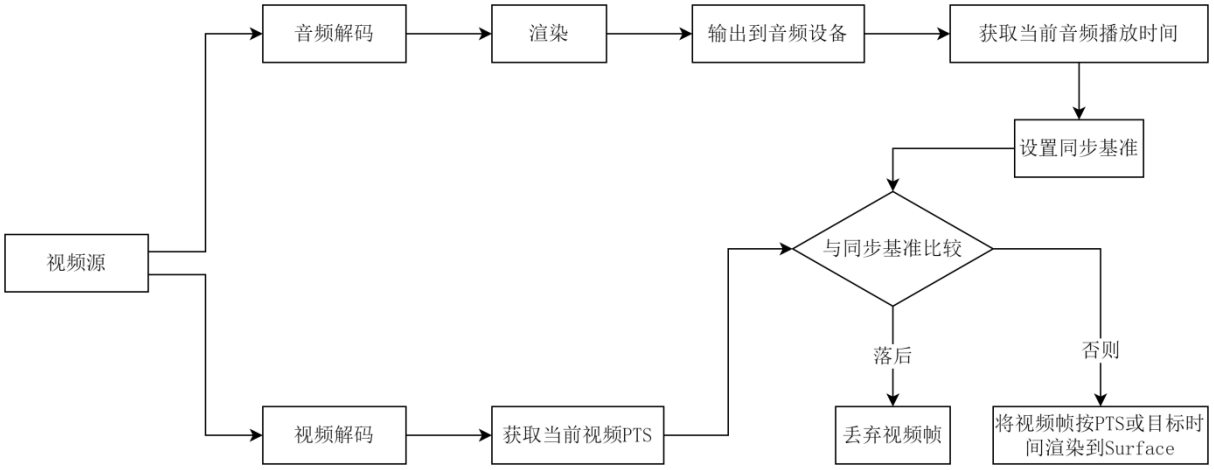


图 5　音画同步流程示例

7.2.3　DRM 流程

见附录 A。

附 录 A

（资料性）

DRM 流程示例

A.1 DRM 流程示例（实验性方案）

1、 MediaPlayer 直接播放方案

- 使用 MediaPlayer/NuPlayer 等系统播放器，并由播放器在系统进程内完成容器读取、解密、解码、写入 AudioSink 等操作。
- 或使用 offload/direct/tunneled 等模式（依赖 HAL 与 profile），以减少上层处理。
- 明文 PCM 不回到应用，安全性取决于厂商实现与授权策略。
- **风险**：需要应用使用 MediaPlayer 播放，潜在适配成本过高；并且需要系统集成 AudioVivid 解码器，谷歌认证兼容性风险较高。

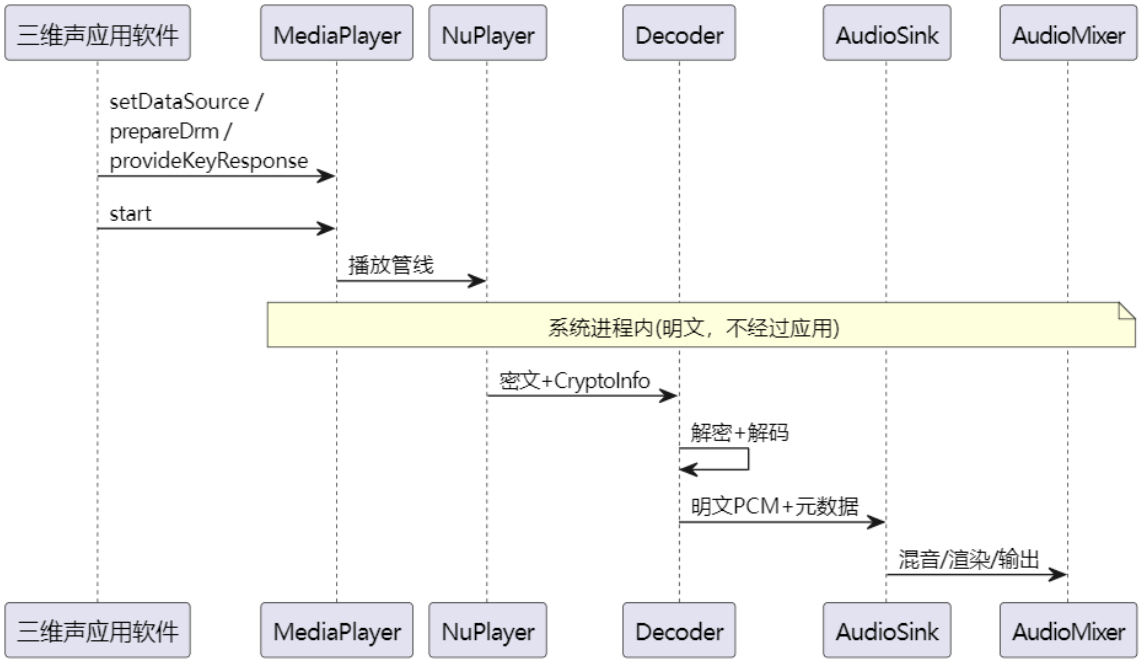


图 A.1 DRM 流程示例（MediaPlayer 直接播放）

2、 MediaCodec 二次加密方案

- **总体**：MediaCodec 解码得到明文 PCM，再经过二次加密后回传给应用；应用写密文 PCM 到 OAudioTrack 再解密后播放。
- 解码：加密数据由应用调用 MediaCodec/Crypto 解密，解码得到明文 PCM。
- 加密后回传：在 MediaCodec 输出链路上对明文 PCM 使用 AES 或其它等同强度算法进行加密（需平台/扩展支持）。dequeueOutputBuffer 回传应用的是密文 PCM，应用从未拿到明文 PCM。
- 播放：应用仅持有密文 PCM，并通过 OAudioTrack.write 接口写入。OAudioTrack 识别到加密轨道后进行解密，解密后的 PCM 进行混音/渲染/输出播放。
- **风险**：需在 MediaCodec 输出路径增加加密功能、在 OAudioTrack 增加解密与密钥管理，改造面较大；倍速等效果需要重新适配，倍速播放/跳转等能力复杂化；安全收益上，系统侧仍会出现明文；整体兼容性与维护成本高。

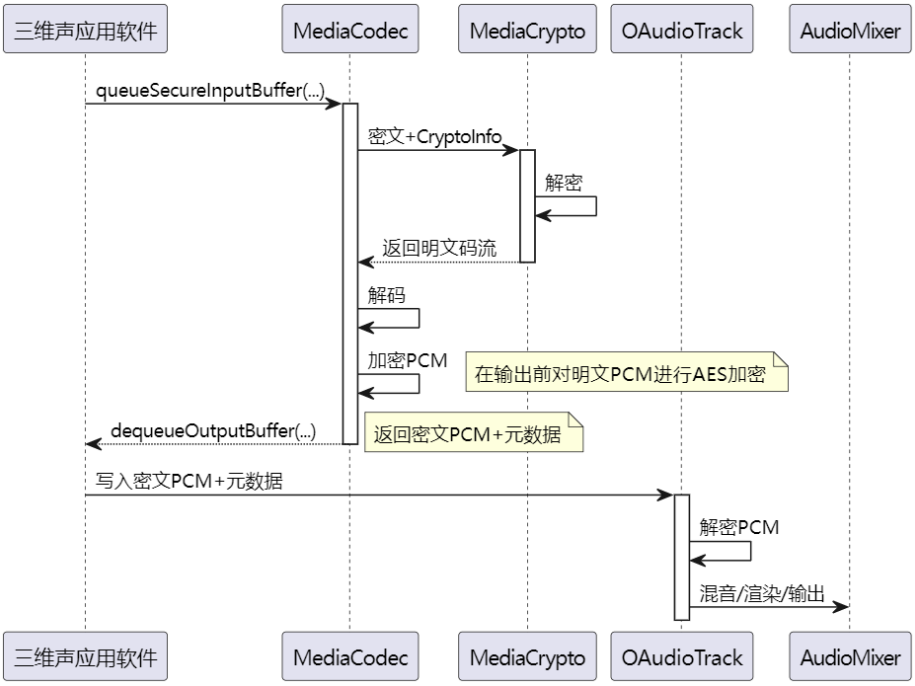


图 A.2　DRM 流程示例（MediaCodec 二次加密）

3、 期望方案（音频安全渲染通路，待 AOSP 未来版本支持）

- **总体**：在不使用 MediaPlayer、不依赖 offload/direct 等模式的前提下，实现类似视频 Surface 的音频安全渲染：即 MediaCodec 解码后的明文 PCM 不回传应用，而是由系统直接输出到音频服务。
- 仍由应用持有 MediaDrm 和 MediaCodec，负责授权管理与 queueSecureInputBuffer 调用。
- 与方案 2 的主要差异：MediaCodec 配置一个音频接收器（AudioSink，类比视频 Surface）。解码后的明文 PCM 被直接写入该接收器，并由系统不经过应用处理直接发送到音频服务，该明文数据也就不会出现在应用侧的内存中。
- 对于部分硬件平台，所有涉及明文 PCM 的模块都可以运行在具备硬件加密能力的子系统中，以进一步提升数据安全性。
- **风险**：目前 AOSP 还没有相关的架构，需要谷歌在未来支持，周期较长。

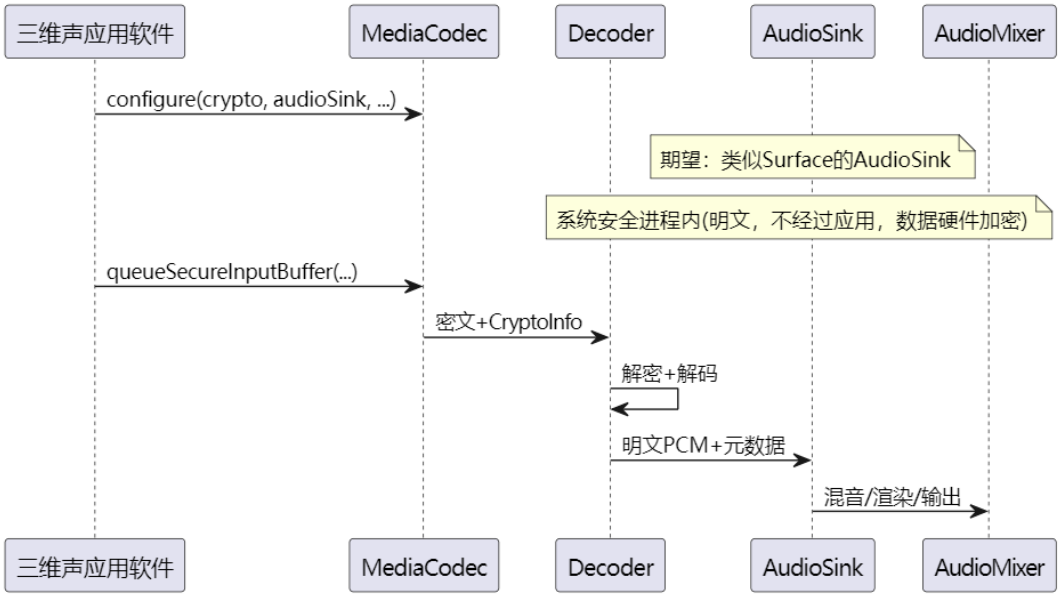


图 A.3　DRM 流程示例（期望方案）

参考文献

[1]ISO/IEC 13818-1 信息技术 运动图像及其伴音信息的通用编码 第1部分:系统 (Information technology -- Generic coding of moving pictures and associated audio information -- Part 1: Systems)
[2]ISO/IEC 14496-12 信息技术 音视频对象的编码 第12部分：ISO基本媒体文件格式 (Information technology – Coding of audio-visual objects – Part 12: ISO base media file format)
[3] ITU-R BS.2076-2 音频定义模型（Audio Definition Model）
[4] ITU-R BS.2094-1 音频定义模型通用定义（Common definitions for the Audio Definition Model）