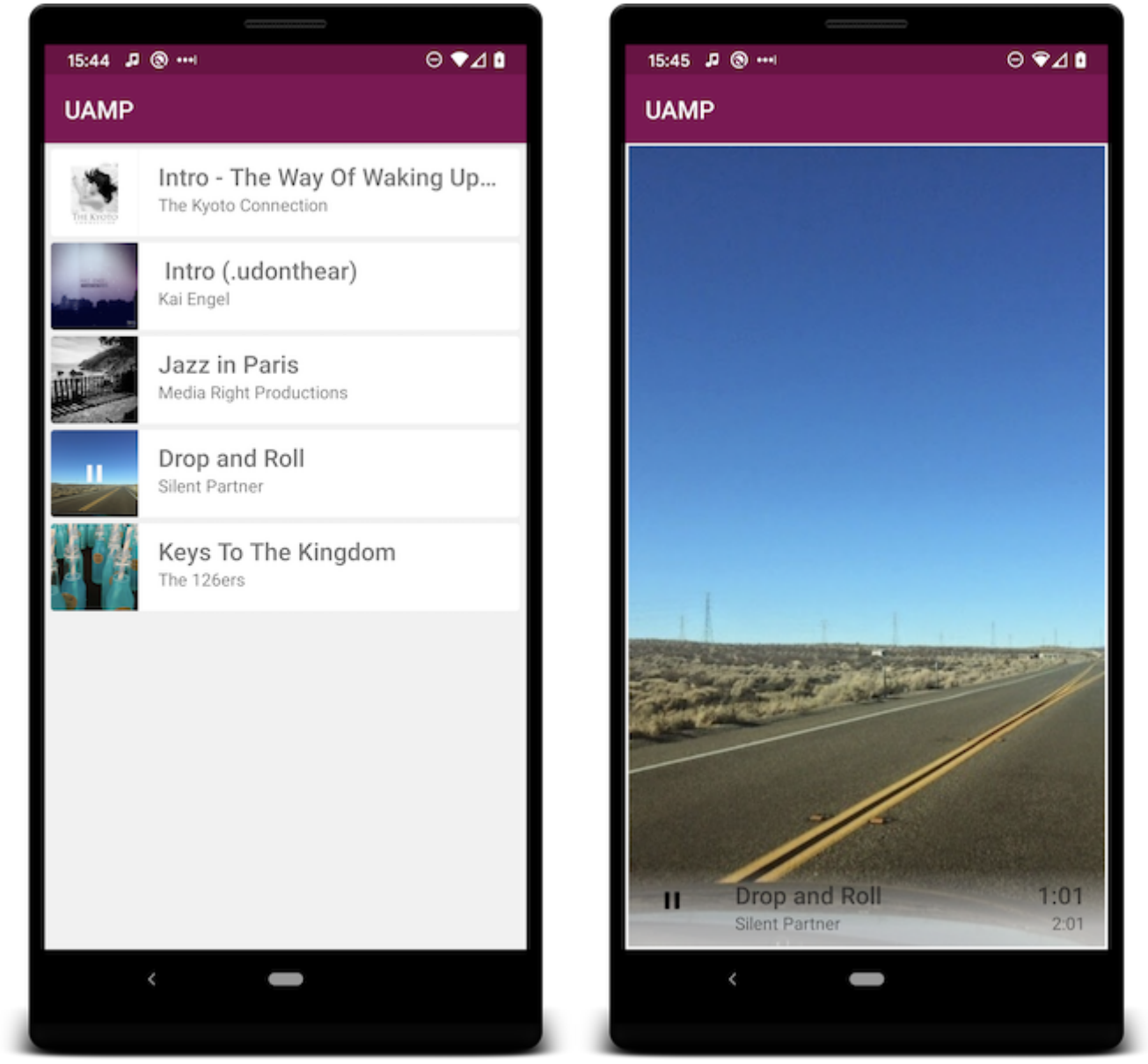
UAMP 完整指南

通用 Android 音乐播放器 (UAMP) 是一个用 Kotlin 编写的适用于 Android 的示例音乐播放器应用程序。它支持许多功能，包括后台播放、音频焦点处理、多个平台（如 Wear、TV 和 Auto）和助手集成。

它从远程服务器加载音乐目录，并允许用户浏览专辑和歌曲。点击歌曲将通过连接的扬声器或耳机播放。它使用 ExoPlayer。

如果您的应用程序的主要目标是播放音频，那么 UAMP 是一个不错的起点。

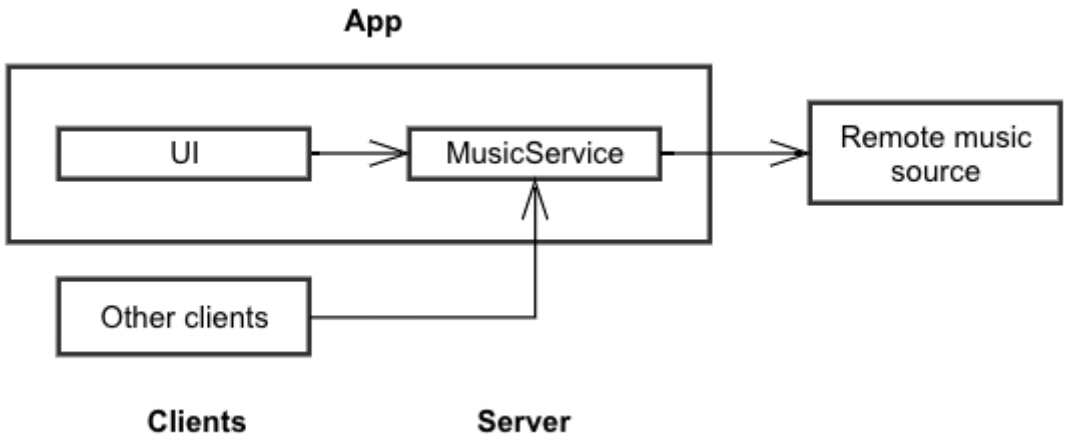


屏幕截图：使用 UAMP 浏览专辑并播放歌曲

架构概述

UAMP 遵循"如何构建音频应用程序"官方文档中描述的客户端/服务器架构。

这是一个架构概述：



图：UAMP的整体架构

服务器架构

音乐服务

服务器端最重要的类是MusicService，它是 MediaBrowserService 的子类。 MediaBrowserService 允许来自 UAMP 和其他应用程序的 MediaBrowser 客户端发现服务、连接到媒体会话并控制播放。

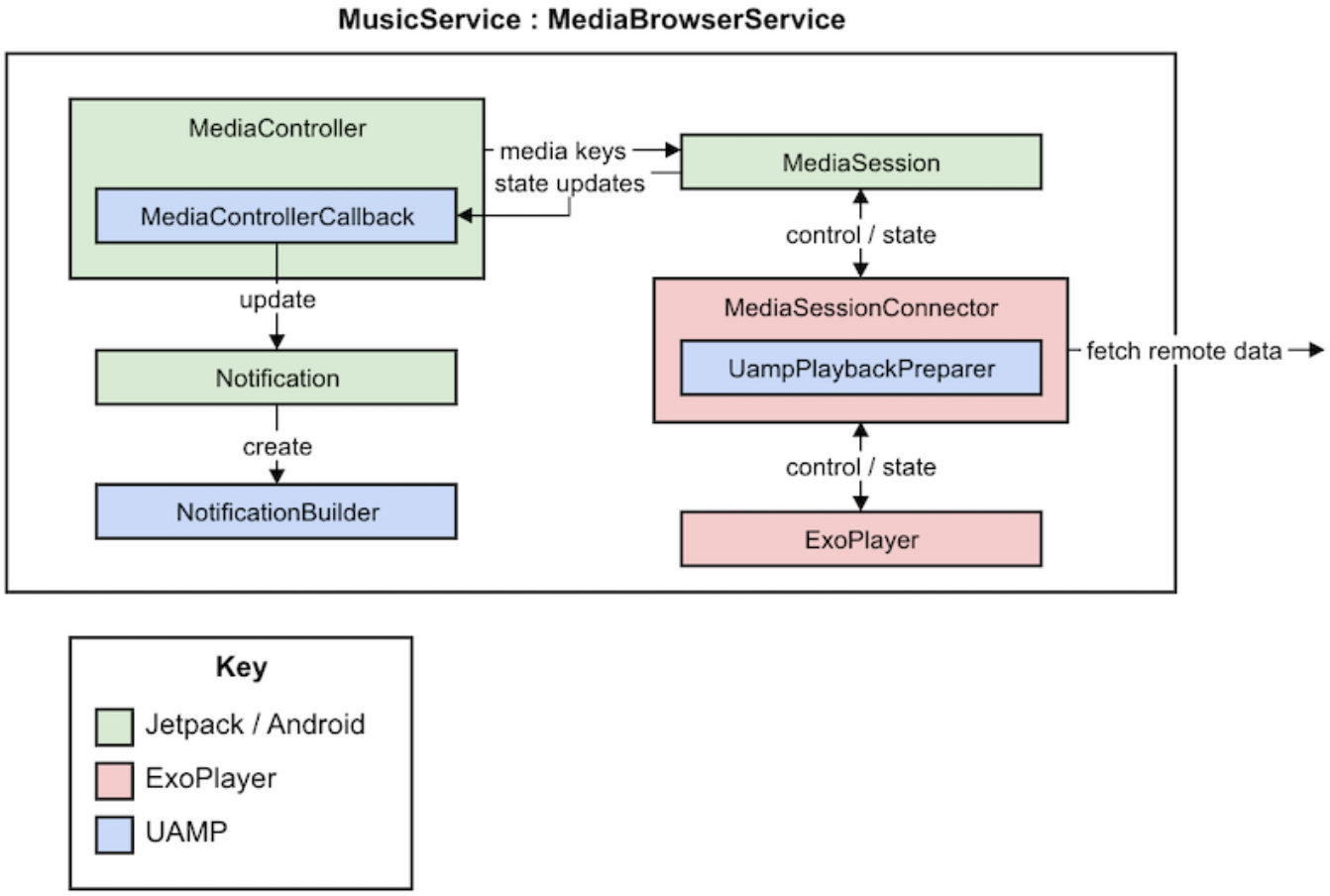
UAMP 实际上使用 MediaBrowserServiceCompat - 由 androidx.media 库 提供的向后兼容的 MediaBrowserService 版本（更多信息）。

MusicService 负责：

- 音频播放器（由 ExoPlayer 提供）
- 媒体会话 和与之通信的对象
- 维护一个 通知 显示当前媒体的信息和控件
- 从远程 URI 加载 媒体目录（一个 JSON 文件）并将其提供给 MediaBrowser 客户端

通过将负责音频播放的对象保留在服务中，它允许在后台播放音频，从而将播放与应用程序的 UI 分离。

这是 MusicService 的更详细视图。

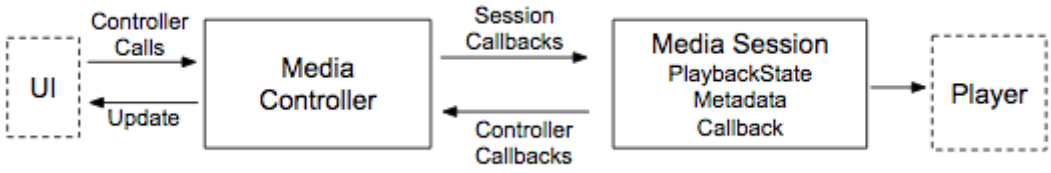


图：MusicService 的详细视图

媒体会话和控制器

MediaSession 表示正在进行的媒体播放会话。它提供了各种机制来控制播放、接收状态更新和检索有关当前媒体的元数据。

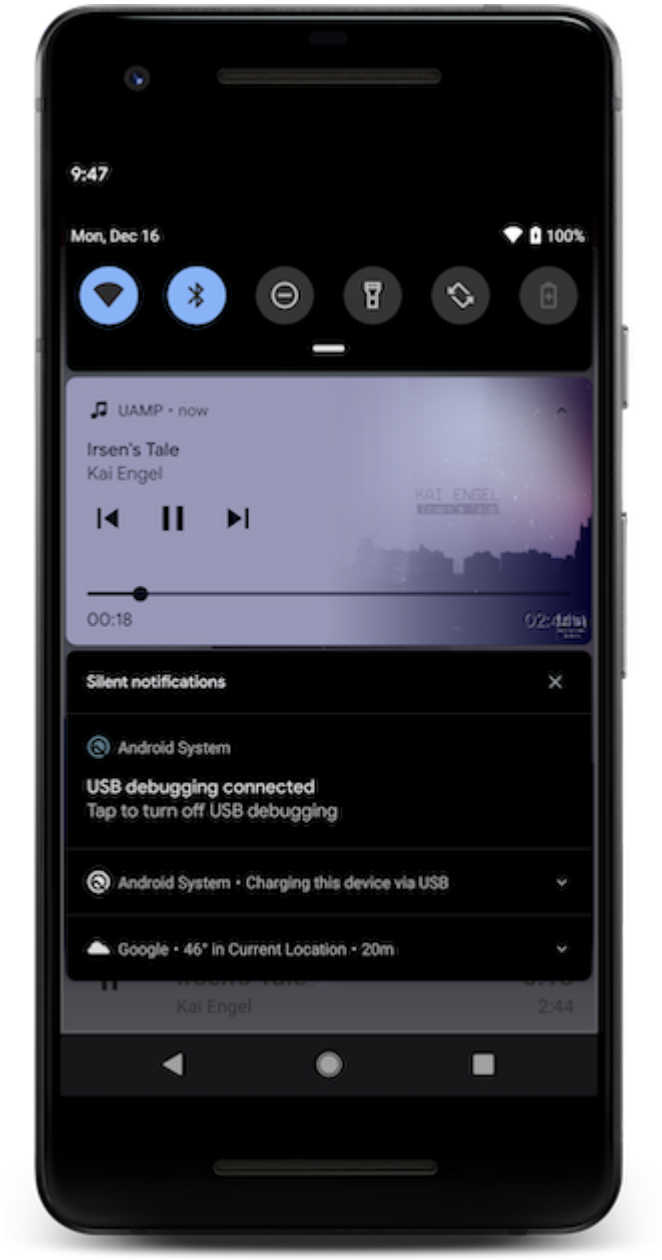
MediaController 用于与媒体会话进行通信。它接收媒体按钮事件并将它们转发到媒体会话。来自媒体会话的状态和元数据更新通过 MediaController.Callback 执行。



图表（来自 官方 Android 文档）：显示了 MediaController 和 MediaSession 如何通信。

UAMP 使用两个 MediaController。一个在客户端与 UI 通信（稍后解释），另一个在 MusicService 内部监听媒体会话状态和元数据变化。然后将其用于更新通知。

通知用户



截图：通知显示有关当前正在播放的歌曲和播放控件的信息

通知 允许用户查看正在播放的歌曲并控制播放。这也是 前台服务 的强制性要求，并阻止 MusicService 被杀死。

UAMP 将其通知的显示和更新委托给 ExoPlayer 提供的 `PlayerNotificationManager`。

播放音频

音频播放由 ExoPlayer 提供。它负责通过UampPlaybackPreparer加载媒体源，通过可用的音频硬件（耳机或扬声器）播放音频并响应媒体命令（播放、暂停、跳过等）。

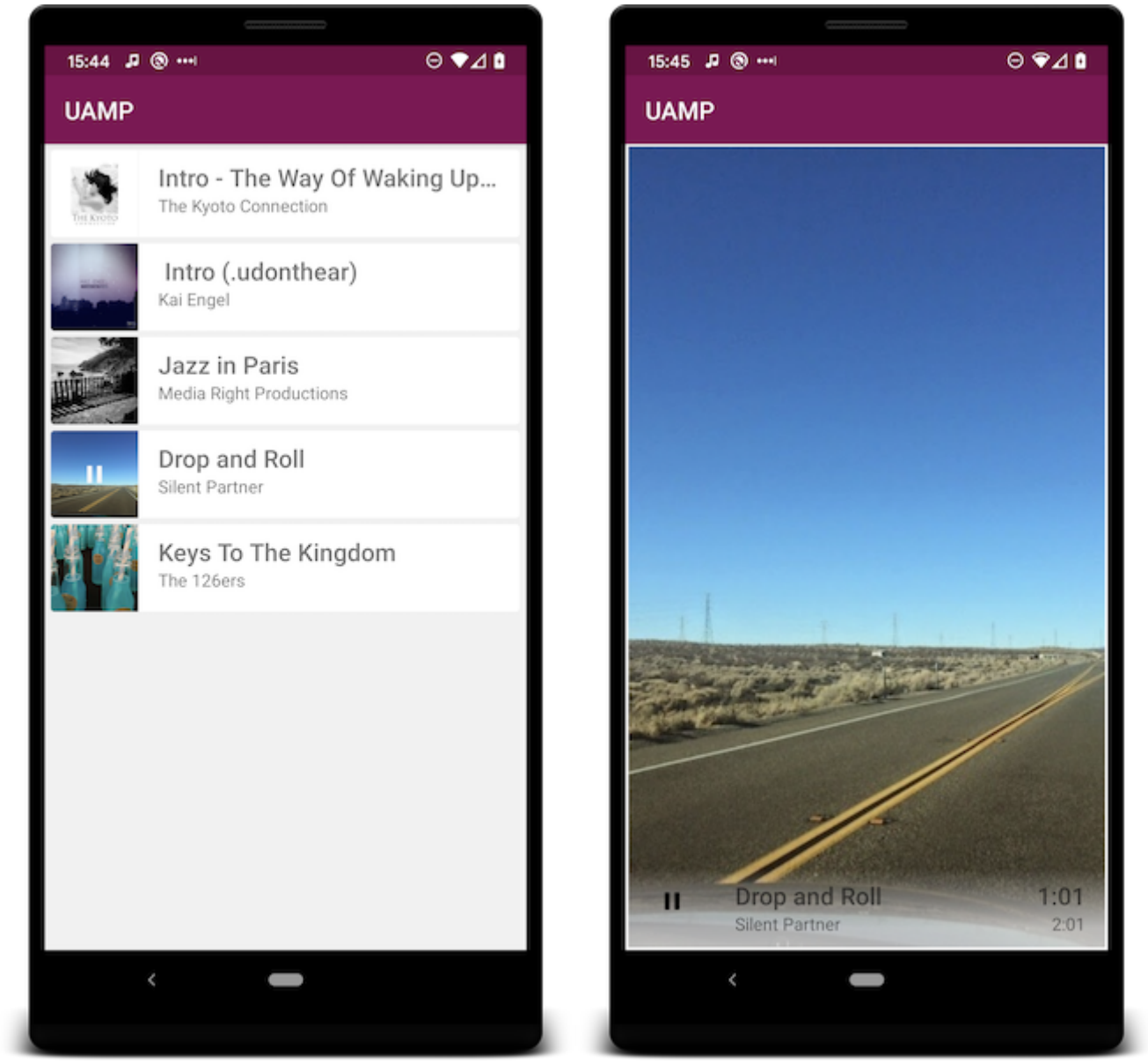
媒体会话连接器

MediaSessionConnector 是 ExoPlayer 的媒体会话扩展。它提供了 ExoPlayer 和 MediaSession 之间的粘合剂（此处的完整详细信息）。

其主要职责是：

- 使用 UampPlaybackPreparer 从 URI 准备媒体源
- 将播放状态更新从 ExoPlayer 发送到媒体会话
- 将媒体会话中的操作（例如播放、暂停和跳过）转发到 ExoPlayer

用户界面

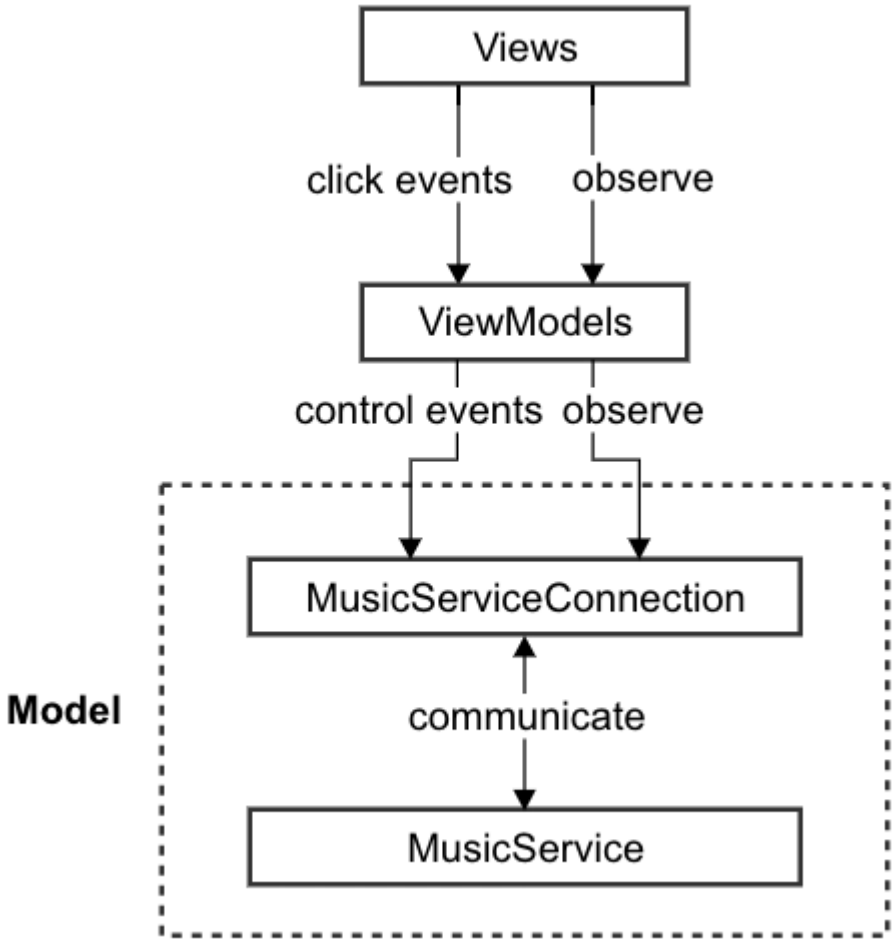


屏幕截图：UAMP 用户界面屏幕

UAMP 用户界面允许用户：

- 浏览和播放歌曲
- 播放和暂停歌曲
- 查看底层播放器的变化，例如播放时长
- 查看当前播放歌曲的元数据，包括专辑封面、标题和艺术家

UAMP 通过使用 Model-View-ViewModel 架构来实现这一点。这允许在每一层之间分离职责。

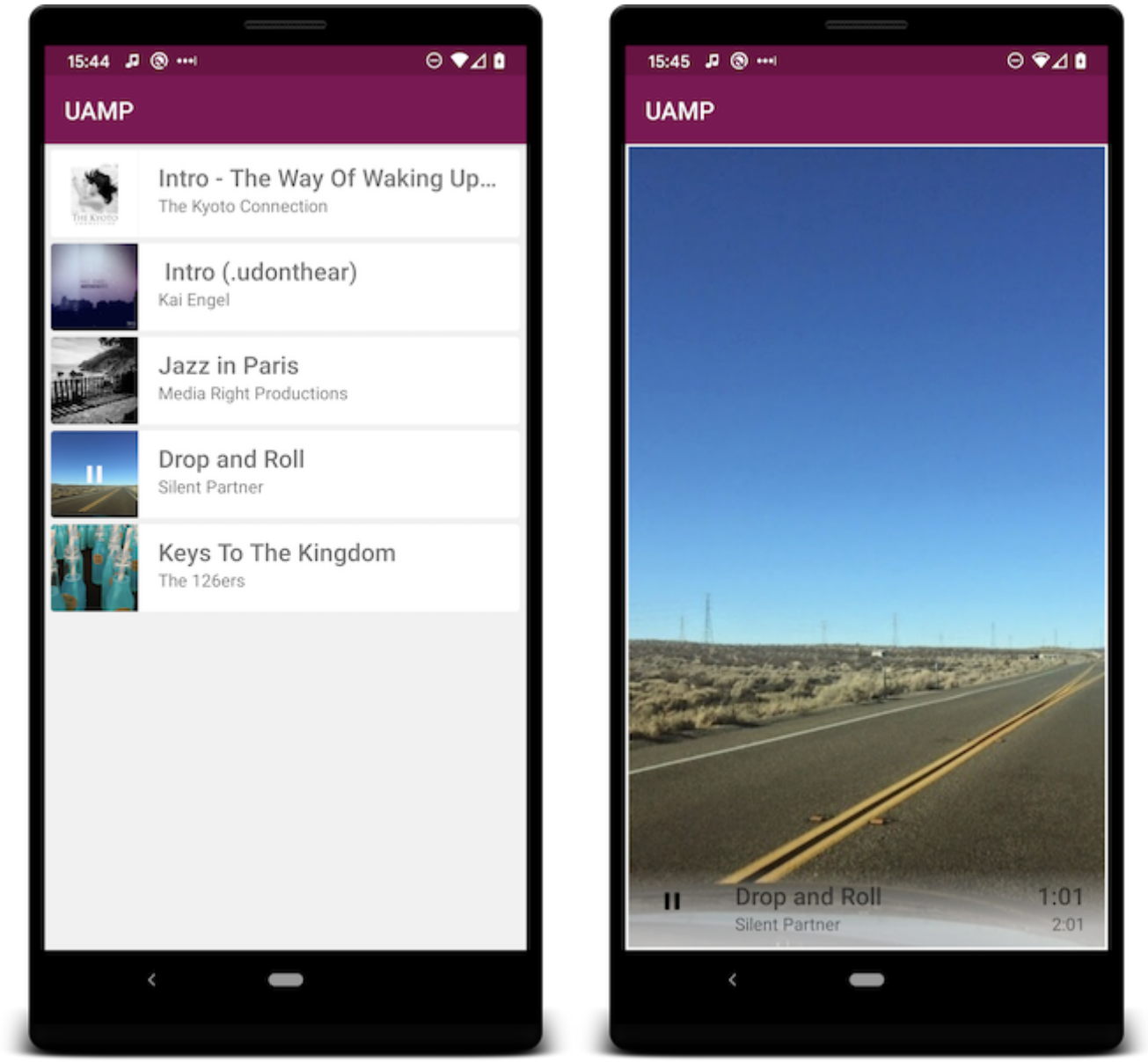


图表：显示 UAMP 的 Model-View-ViewModel 架构的类图

观看次数

UAMP 具有三个主要视图类 - 一个 Activity 和两个 Fragments:

- **MainActivity** 负责两个片段之间的交换。
- **MediaItemFragment** 用于浏览音乐目录。它显示可以是专辑或歌曲的媒体项目列表。点击专辑将显示一个新的 MediaItemFragment，其中包含该专辑中的歌曲。点击一首歌曲将开始播放该歌曲并显示 NowPlayingFragment。
- **NowPlayingFragment** 显示当前正在播放的歌曲。



截图：MediaItemFragment 和 NowPlayingFragment 由 MainActivity 显示

视图模型

Activity 和 Fragments 由它们自己的 视图模型 支持。视图模型表示相应视图的底层状态。当模型更改时，例如播放新歌曲时，视图会更新以反映更改。这是使用 LiveData 类型实现的。

视图模型能够使用 MusicServiceConnection 与 MusicService（在上面的"服务"部分中描述）进行通信。

音乐服务连接

MusicServiceConnection 是一个连接到 MusicService 的单例。它是 MediaBrowser 和 MediaController 的包装器类。

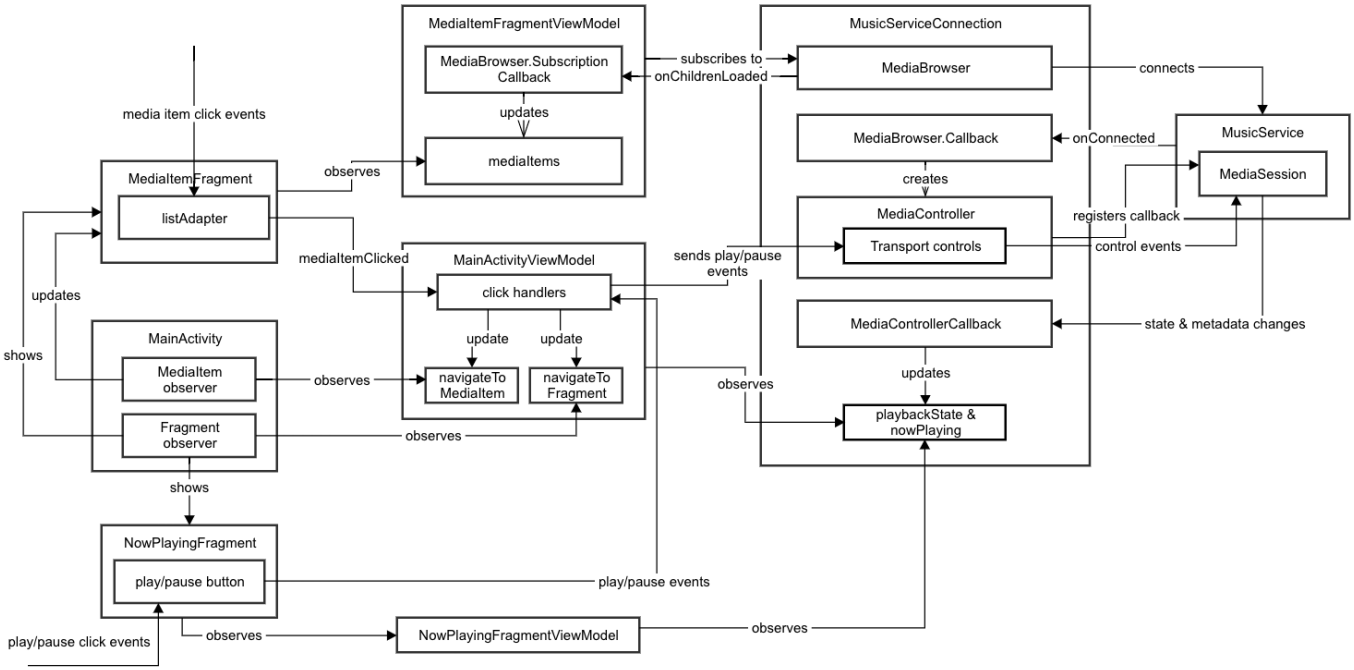
它负责：

- 使用 MediaBrowser 连接到 MusicService
- 接收 MusicService 的回调，表示 MediaBrowser 连接成功
- 允许视图模型通过 subscribe 订阅 MusicService 的媒体项目列表中的更改来检索专辑和歌曲列表
- 为当前的媒体会话创建一个 MediaController，视图模型可以将其用于：

- 通过传输控制控制会话
- 检索播放状态和元数据更改

类图

下图显示了 UI 类之间最重要的交互。



图：UI 类之间的重要交互

概括

开始使用 UAMP 的最简单方法是克隆它、运行它并使用 Android Studio 逐步执行源代码。如果您有任何问题或发现任何令人困惑的部分，请提交问题。

更多资源

如何构建音频应用

媒体播放最佳实践 - Google I/O 2016

ExoPlayer