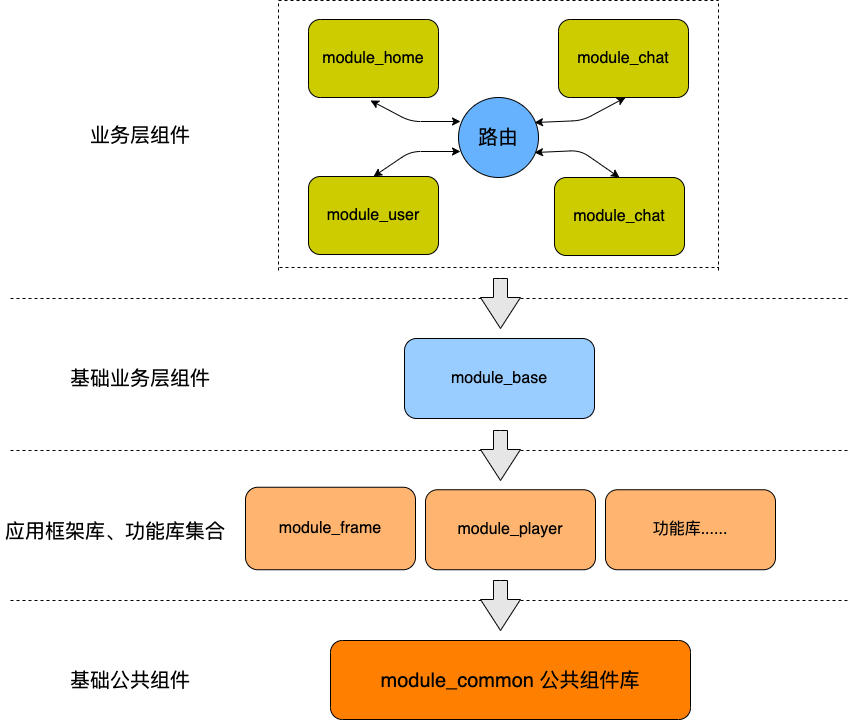
### 短视频初版架构:



#### 层级模块说明:

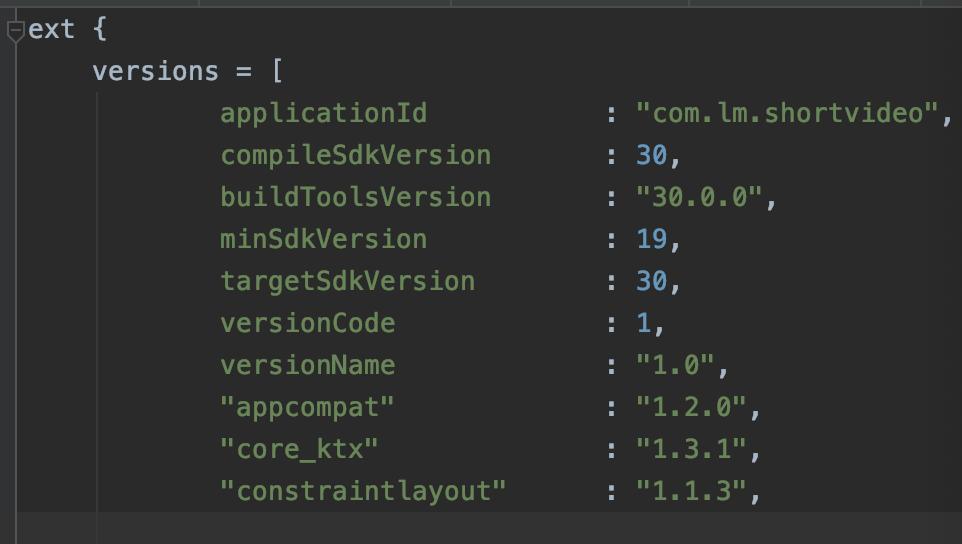
**module\_common**: 公用组件层，存放通用扩展函数、权限、动画、图片等工具和通用型独立功能.

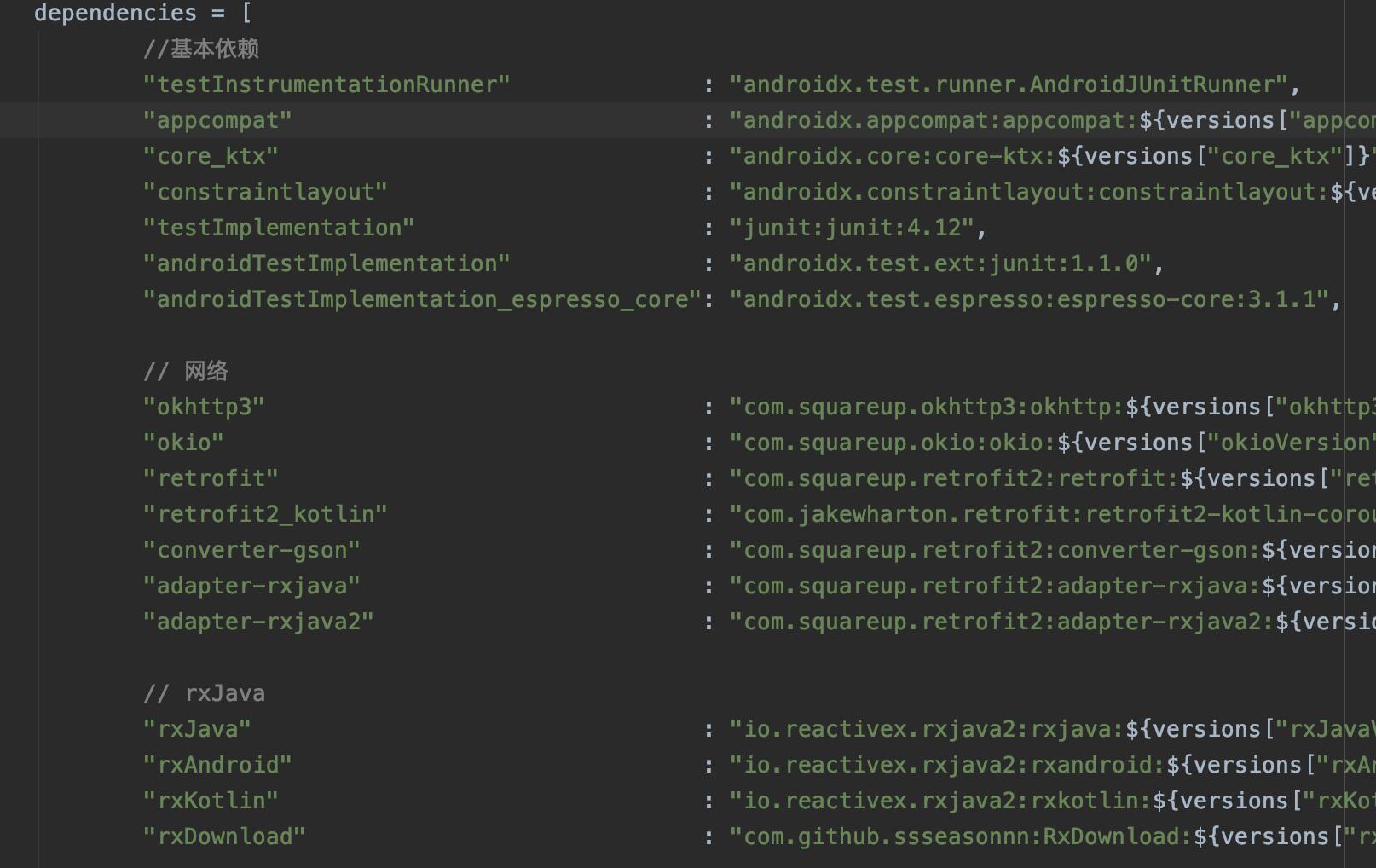
**module\_frame**: 应用框架层，封装网络请求、上层开发框架.

**module\_base**: 基础业务层，定义页面路由信息、实现应用框架层扩展接口、公用网路请求、初始化全局配置，提供模块服务暴漏接口.

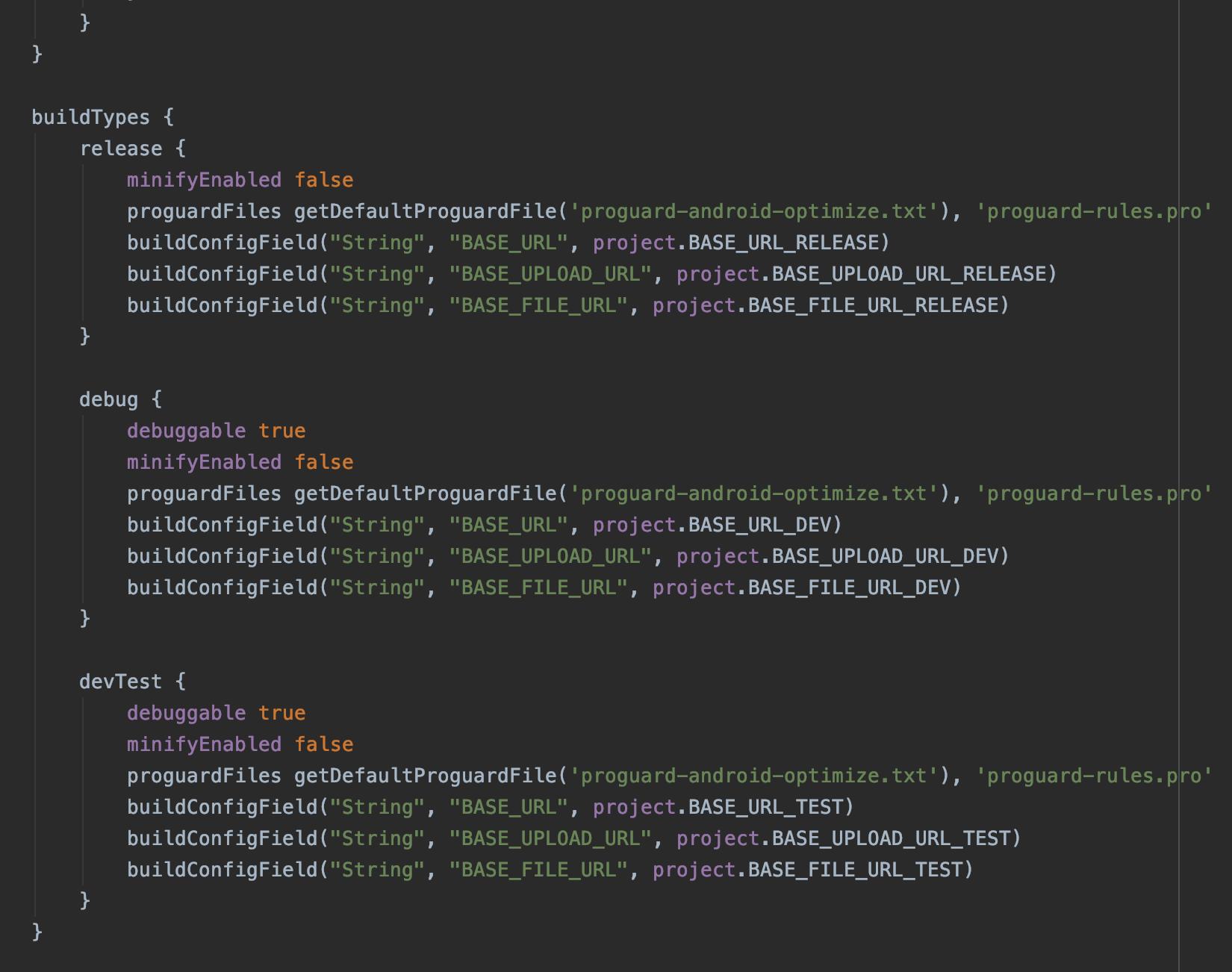
**业务层组件**: 各业务模块module，代码隔离，可单独运行.

#### 版本号和依赖库放入config.gradle统一管理:





#### 运行环境定义和配置放到module\_base的build.gradle中:



#### module\_base中定义页面路由、模块通信接口



###### 模块通信接口



#### 模块中实现功能：



#### 模块接口使用工具



框架选型：

**播放器ijkPlayer:**

API易于集成、编译配置可裁剪、支持硬解码，经对比其他播放器，播放兼容性更优.

**长连接 netty+protobuf:**

Nio高性能、多种通信协议编解码、易处理粘包拆包，protobuf数据体积小、传输快、可压缩.

**网络请求okhttp3 + retrofit2 + rxjava2:**

支持http/1.1、http/2协议，共享通信连接、减少对服务器请求连接次数，请求接口定义基于注解、使用灵活方便、支持同步、异步、请求的定义与实现完全解耦.

**应用层框架mvvm + livedata:**

View层通过泛型指定ViewModel类型，框架层根据泛型完成ViewMode初始化, View与ViewModel建立关联.

ViewModel层通过泛型指定Model类型，框架层根据泛型完成Model初始化，ViewModel与Model建立关联，可进行网络请求.

网络请求结果响应：在ViewModel中定义livedata，View中注册livedata的观察者，框架层通过livedata切换到主线程回调给观察者View.