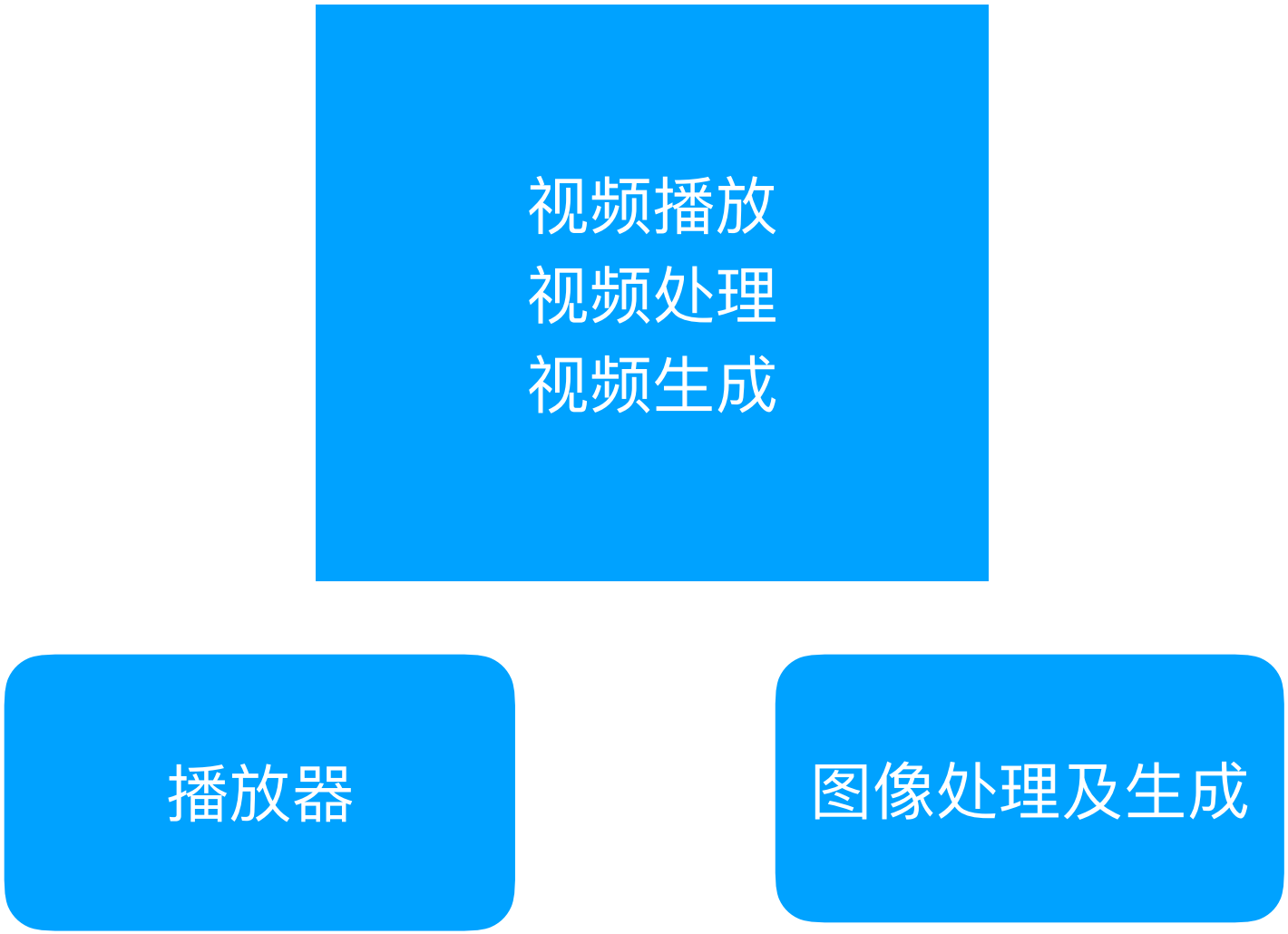
UGC实现

隔离型

保证稳定性，一是性能调用，二是代理系统或底层行为，隔离上层繁多功能。

sdk



隔离层

XSurfaceCanvasManager
XPlayerManager
XPublishManager
封装播放和视频处理功能，信息输出

XSurfaceCanvasManager
封装视频处理信息及信息输出

XPlayerManager
封装播放器功能及信息输出

XPublishManager
封装视频生成及信息输出

Android端业务实现

容器结构
Activity+多Fragment

编码结构
Fragment处理视图编码
MemoryManager提供model模版
Presenter隔离业务逻辑代码

交互处理
符号收集
模型处理
结果输出

并发处理
包含多个不同类型线程池的 FrameThreadMgr

Rx组件，处理有序任务

容错处理

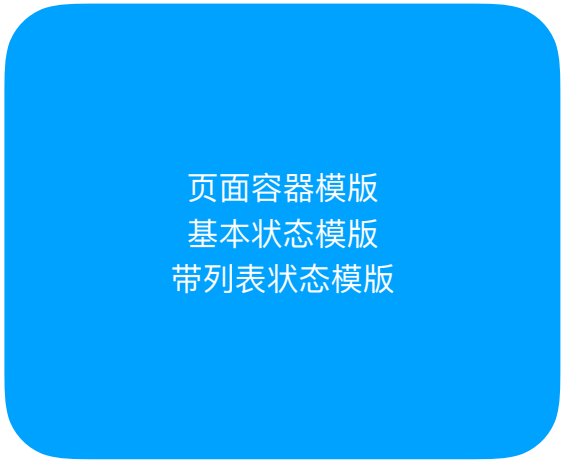
系统无序行为
频繁点击
绘制顺序
响应优先级
绘制适配
变量周期
任务优先级

主app资源引用
运行时注入

Activity+多Fragment

复用性

整个工程都在复用，view模版，页面状态模版，mvp结构模版，特殊控件组合模版，业务模块模版

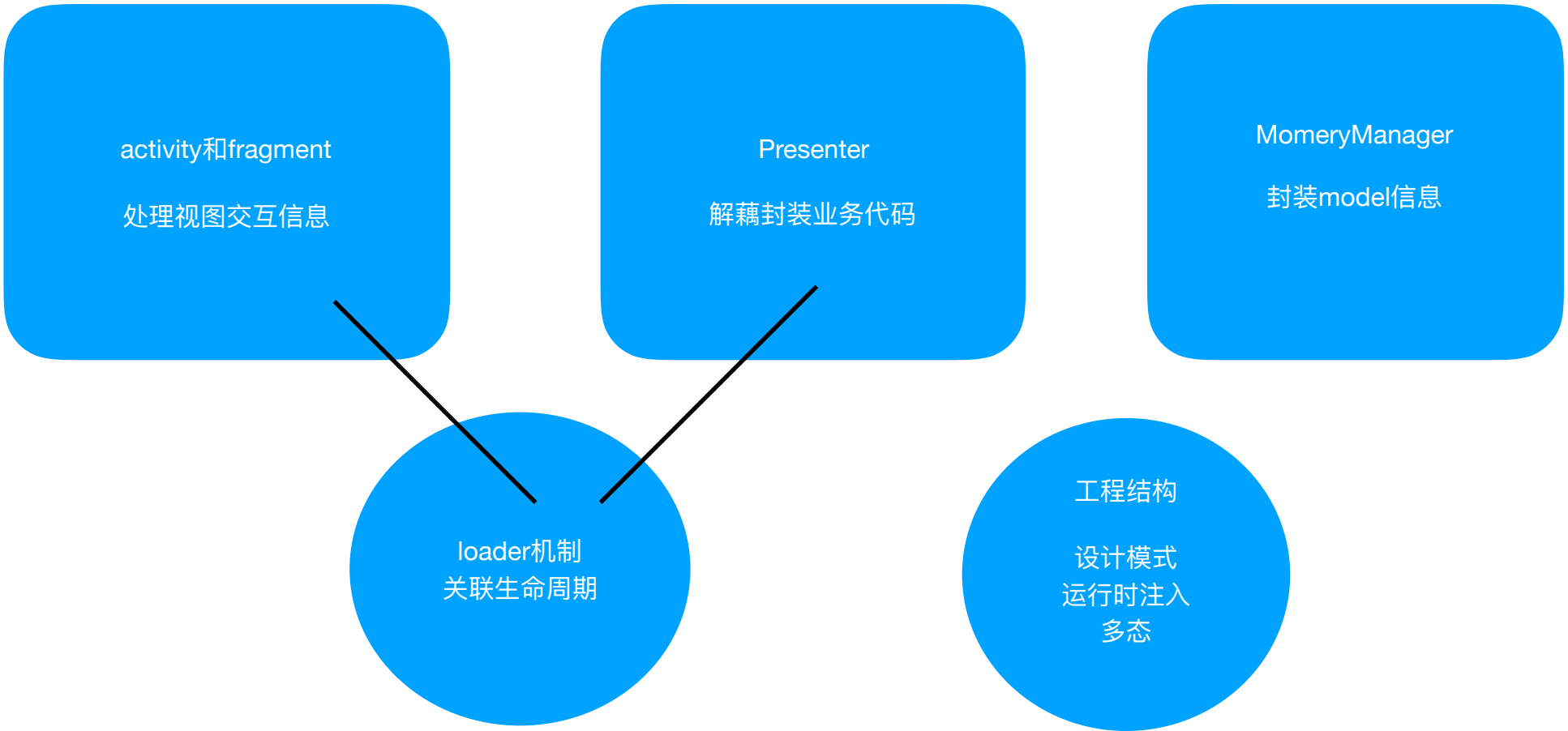


编码结构

1 复用性
2 边界稳定性

自动化，封装型

具体实现在mvp结构里，利用语言特性，运行时注入等技术，下沉自动化行为，保持简洁性，保证最小编程聚焦。



边界处理

把边界处理下沉为自动化行为，如生命周期，基本的调用流程，异步，修改和处理都在封装层。

交互模型

画布交互
符号收集
触摸拦截
模型处理
ViewDragHelper，触摸拦截
结果输出
MemoryManager收集信息
画布重绘，或矩形变换显示呈现

frame选择器交互
符号收集
触摸拦截
模型处理
触摸拦截，联动关联
结果输出
MemoryManager收集信息
重绘，scroll相关控件，内部重绘

交互联动处理

doubleviewobserver　处理 上层画布与下层选择器的交互及功能关联

各层控件内部联动处理，例如底层包含滚动，帧轴，seekbar选择器

单向或双向联动处理

MemoryManager模块

编辑态时，对应特效效果的数据结构管理类，统一管理

PublishManager模块

封装发布流程相关信息

XSurfaceManager&XplayerManager模块

特效编辑模块，播放模块

对下，包装底层sdk提供特效功能，api，特效执行信息监听
对上，提供特效处理，播放功能接口

并发处理

包含多个不同类型线程池的 FrameThreadMgr

Rx组件，处理有序任务

RxJava 是一个思想优秀的框架，而且是那种在工程领域少见的带有学院派气息和理想主义色彩的框架，他是一种新型的事件驱动型编程范式。 RxJava 最重要的贡献，就是提升了我们原先对于事件驱动型编程的思考的维度，允许我们可以从时间和空间两个维度去重新组织事件。

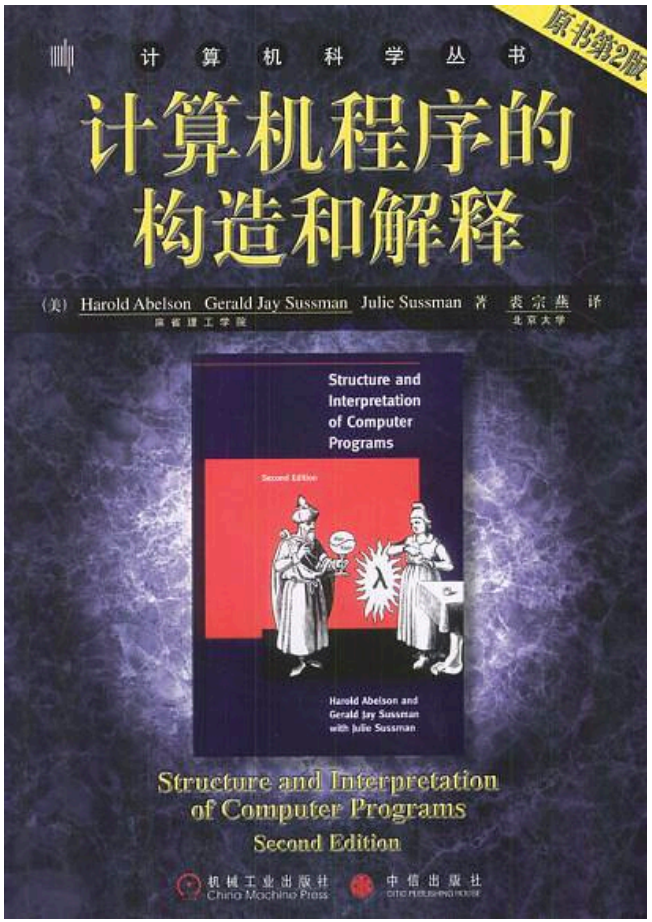
view复杂联动
抽帧业务
发布流程

函数式语言

数据流处理

Haskell，Lisp等

SICP，介绍了几种类型的编程语言，然后从实现编程语言的角度去坦述编程语言要解决的问题，即真实世界的各种抽象模型，了解编程思维很不错的书。



总结

1 端实现的隔离型

保证稳定性，一是性能调用，二是代理系统或底层行为，隔离上层繁多功能。

2 复用性

整个工程都在复用，view模版，页面状态模版，mvp结构模版，特殊控件组合模版，业务模块模版

3 自动化，封装型

具体实现在mvp结构里，利用语言特性，运行时注入等技术，下沉自动化行为，保持简洁性，保证最小编程聚焦。

4 边界处理

把边界处理下沉为自动化行为，如生命周期，基本的调用流程，异步，修改和处理都在封装层。

5 工程结构，计算结构

1 设计模式，反向代理，一种与结构有关的工程模式

2 更关注功能性输出，结构简单

6 编程模型

符号 - 模型 - 处理 - 输出

7 编程思维

画

分享

一 ugc 端上实现

二 总结