Application：使用App.java

BaseSplashActivity PrivacyNoticeActivity StopServiceActivity

依次是启动、隐私变更、停止在线

其他：用户打点（上传服务器，收集点击事件）

BaseSplashActivity：

父类1：BasePromptActivity(提示Activity)

父类2：BaseSplashActivity(主页面上级页，弹框基础页Activity)

主页面SplashScreenActivity

Oncreate方法：

1. init()：初始化广告模板，修改是否跟新标志
2. isFromByIntent()：查找是否是其他应用跳转，如果是设置isFromOpenAcility为true，否则设置为false
   1. 如果true，直接走showPromptDialog方法判断是否需要显示协议策略
   2. 如果是false，那么属于用户手动启动该页面，需要走下面流程
3. 设置状态栏，设置SP的值KEY\_JUST\_TRY\_GRS为false
4. 判断横屏状态，延迟加载或者调用showPromptDialog方法
5. View挖孔

ShowPromptDialog方法：

SignServiceHelper#getUserSignRecord()：获取用户签署记录

->调用CheckSignStatusTask异步加载

->start()方法：判断当前版本号是否小于服务器版本，是：进行无记录判断，否则使用HVILogicSDK获取当前版本号回调判断，如果HVILogicSDK返回未签署，那么走无记录判断（这里没有即使判断拒绝和跳过的状态）

->dealWithNoSignRecord处理没有签名记录的情况：

1. 获取缓存判断是否已经同意过签署CheckUserIsAgreeMgr.getInstance().userIsAgree()

2. 如果之前版本同意过，而且是4版本号，那么请求上传

3. 回调callback的签署状态Agree/NeedSign

\*\*\*\*\*最后结果走SingnServiceHelper的getUserSignRecord匿名内部类回调，使用EventBus发送最后通知，接收方在：BaseSplashActivity下面的signRecordCheckReceiver

发送：

EventMessage eventMessage = new EventMessage(ACT\_SIGN\_RECORD\_RESULT);

eventMessage.putExtra(SIGN\_RECORD\_RESULT\_KEY, signStatus.ordinal());

GlobalEventBus.getInstance().getPublisher().post(eventMessage);

接收：

private Subscriber subscriber = GlobalEventBus.getInstance().getSubscriber(signRecordCheckReceiver);

subscriber.addAction(SignServiceHelper.ACT\_SIGN\_RECORD\_RESULT);

subscriber.addAction(SignServiceHelper.ACT\_ADD\_SIGN\_RECORD\_RESULT);

subscriber.register();

处理方法：dealWithSignRecordCheckResult

dontNeedShowDialog：不需要Dialog

needShowDialog：需要Dialog提示授权

PrivacyNoticeManager.getInstance().createPromptDialog创建国内/海外提示弹框

根据BuildConfig.ENABLE\_SIGN\_RECORD来判断需要显示什么弹框

国内版本提示框：

同意，设置是否接受推送，（设置push开关状态），回调一次

拒绝，回调一次

海外版本提示框：

1. 海外隐私通知弹框（2者都没有复选框或者推送通知框）

如果拒绝：直接回调

如果接受：创建海外用户协议弹窗（如果此刻接受，回调接受，否则创建上一个Dialog）

返回的接受和拒绝提示弹框需要如下操作：

BaseSplashActivity#needShowDialog#createPromptDialog🡪new PromptDialogCallBack()

拒绝：

关闭页面

同意：

1. 使用saveLocalAfterAgree

ServiceTermModel.getInstance().agreeTerms提交同意请求，回调EventBus发送ACT\_ADD\_SIGN\_RECORD\_RESULT消息

1. ServiceTermUtils.updateNewTerms();跟新默认使用协议4
2. EventBus接收端在BaseSplashActivity里面调用dealWithAddSignRecordResult(boolean isComplete)
3. 请求权限requestPermissionIfNeed

主要使用方法：

6.checkSignStatus

2. agreeTerms

PrivacyNoticeActivity隐私变更

1. MainActivity在OnCreate方法里面使用TermsManager.getInstance()使用checkTermsUpdate(new CheckTermsUpdateLister(MainActivity.this))方法
2. CheckTermsUpdateTask#start检查
3. 执行ServiceTermModel.getInstance().checkUpdateUserSignRecord(null, new CheckTermsUpdateListener());

3.检查通知，返回并创建三种通知类型，第一种是通知类，第二种是确认类型A，第三种是确认类型B

1. 启动PrivacyNoticeActivity页面，传入参数，但是此时的callback对象存在于静态单列PrivacyNoticeManager里面，来源
   1. 如果是通知类，那么使用CheckTermsUpdateTask#匿名内部类，不存在引用
   2. 如果是确认类型弹框A，那么使用匿名内部类，点击同意则使用agreeTerms，不同意则使用创建二次确认弹框B，该匿名内部类最终引用Mainactivity
   3. 如果同意使用agreeTerms，不同意则使用cancelTerms取消该协议
2. PrivacyNoticeActivity页面
   1. 根据传入参数显示不同信息，该页面只用于显示信息，具体执行确定或者取消主要用于之前PrivacyNoticeManager静态对activity以及其匿名内部包装类Callback引用进行静态方法执行回调（可能存在内存泄漏）

StopServiceActivity停止在线服务页面

引用见ref

StopServiceActivity页面

事件处理：

1.点击关闭，进行校验是否登陆过CBG

2.如果登陆过CBG，执行checkIsAllowed，否则弹框提示，确认则进行验证

校验是否允许关闭服务CheckStopServiceReq#queryStopServiceReqAsyn(checkStopServiceEvent)

2.1.不允许提示弹框

2.2.允许执行游客或者登录用户校验状态（密码或者指纹认证，完成之后使用EventBus接收）

3.承接未登录CBG，执行游客或者用户提示

如果是游客（未登录用户），那么执行ServiceTermModel.getInstance().cancelTerms

如果是登录用户，执行：

IAccountLogic accountLogic = HVILogicSDK.getLogic(IAccountLogic.class);

accountLogic.checkHwIDPassword(this, BuildConfig.LOGIN\_CHANNEL);

3.1 检查用户登录状态，使用上面方法会接收EventBus消息，该消息返回AccountEventMessageActions.ACT\_CHECK\_HWID\_PASSWORD的Action

3.2 获取bundle消息内容，如果bundle.getBoolean(AccountEventMessageActions.RESULT\_OF\_BUNDLE) == true，那么执行signOnServer();上报签署记录ServiceTermModel.getInstance().cancelTerms

隐私变更

1. MianActivity#Oncreate检查隐私变更

2. TermsManager#checkTermsUpdate

国内版本：

游客：

从LoginConfig中获取保存在本地的签署记录

HVI（LocalTermsManager）获取本地DB签署记录，互相比较每种协议类型，取ConfigLogin中最新的签署记录加入NoticeList列表

返回是否需要显示通知以及最新NoticeList列表（只有通知和A类弹框）

用户：

获取DB的签署记录（海外版需要用户ID查询，国内版本查询所有数据库），从TMS服务器中获取最新签署记录，判断每种协议本地与TMS的最新，如果本地最新需要放入ServerUpdateList，服务器最新需要放入LocalUpdateList，将有变化的协议都放入NotciceList列表，然后分别上传服务器，保存本地，返回是否需要弹框（B类弹框优先，其次A类，其次通知）

启动页流程

通用流程：

1. Intent来源检测（开发能力，Video，桌面）

2.showPromptDialogIfNeed检查是否需要dialog

国内版：BaseSplashActivity

TermsManager#checkSignStatus

国内CheckSignStatueTask

使用LocalTermsManager查询本地数据库，拿出最新记录判断是否同意，返回无记录，需要签署，无需签署

成功回调，发送EVENT\_BUS广播，使用dealWithCheckSignStatus返回是否显示弹框

点击拒绝退出，否则如果需要签署，执行TermsManager#agreeTerms同意签署

HVI的AgreeTermsTask执行流程：

1.获取保存在本地的签署记录（LoginConfig）

2.保存本地签署记录

3.判断是否是用户，如果是用户，（ServerTermsManager）上传服务器

执行申请权限，reqestPermissionIfNeed

执行开屏广告，用户登录

海外版本：BaseSplashActivity

TermsManager#checkSignStatus

1. 添加签署状态监听

2. 启动自动登录，GRSLoader任务开启（完成之后发送EVENTBUS消息用于广告业跳转主页面），GetBeInfo任务开启，完成之后onComplete写入LoginConfig信息，并执行TermsManager#CheckSignStatus

海外CheckSignStatueTask

1. 使用LocalTermsManager查询本地数据库(海外有USERID，国内没有)，拿出本地数据库最新记录与LoginConfig判断是否同意签署，如果返回无记录，需要使用TMS查询服务器，获取结果保存在本地，判断服务器记录是否已经签署

2. 返回签署结果回调之前启动页添加的Listener，如果不在GRS检测（VideoFolderActivity设置的标记位）状态下，那么判断是否需要签署，如果需要显示弹框，否则判断国家渠道是否发生变化，如果变化重启启动页，如果需要签署，那么执行签署流程，不管签署成功与否都要上传TMS

3. 如果是从开发能力进来，那么请求权限

关闭在线服务流程

1. 点击StopServiceActivity的关闭按钮

2. 区分是游客还是用户，如果是游客直接执行cancelTerms，HVI底层将对游客清楚本地记录和缓存，如果是用户，那么检查用户登录状态，验证登录，收到HVI层回调EVENTBUS消息之后执行cancelTerms，HVI层将获取保存在LoginConfig中的签署记录使用ServiceTermsManager上传TMS，成功将执行清楚本地记录，（国内清楚所有，国外只清楚UserID对应数据库和用户缓存）否则回调重试

HVI层业务实现：

核心实现类：TermsManager单例

注册接口：LogicHolderImpl.registerLogic方法

1. checkTermsStatus：检查协议接口

使用场景：

MainActivity启动自检，WebActivity加载完成自检

1. agreeTerms：用户同意协议

使用场景：

2.1 CheckTermsUpdateTask检查协议变更任务MainActivity#onCreate->checkTermsUpdate

2.2 SignServiceHelper#agreeAndUpdateSignRecord用户更新用户签署记录，使用场景BaseSplashActivity#saveLocalAfterAgree

3. cancelTerms：用户取消协议

使用场景：

StopServiceActivity

4. clearLocalSignRecord：清除本地签署记录

使用场景：

（本地视频一级列表）VideoFolderActivity#clearSignRecord（）检查是否有本地签署记录，如有清除本地签署并提示用户

5. 添加检查是否同意协议监听

addCheckSignStatusListener

移除检查是否同意协议监听

removeCheckSignStatusListener

6. checkSignStatus：检查是否签署协议，需要使用5添加协议监听

7. setDefaultTermsConfig设置默认协议配置

setDefaultLatestTermsConfig设置预置的最新协议配置

使用场景：ServiceTermModel#init(App初始化调用)

8. checkSignStatusFromCache从缓存里校验是否同意协议

使用场景：无

9. notifySignResult：通知HVI用户是否同意签署

使用场景：

9.1 BasePromptActivity

lauchAgreePermissStartLauch保存本地签署记录，进入开放能力

ServiceTermModel#callbackUserChoose

9.2 OverseasOnlineController海外版打开在线控制VideoFolderActivity使用

10. setDatabaseName：设置HVI数据库名

使用场景：

ServiceTermModel#init使用，App初始化使用