黑鲨 SDK 联运接入指南

1. 版本更新记录

SDK 版本	更新日期	更新说明
v5.9.3	2022.09.08	1、更新移动安全联盟 msa sdk 2、优化未成年防沉迷下线逻辑
v5.7.7	2022.07.20	1、正确索取应用列表、去掉 IMEI、新增儿童隐私政策、去除短信权限等合规调整 2、优化用户升级及等级信息 3、在线客服优化及客服邮件变更 4、支持实名信息修改 5、其他 bug 修复
v5.6.8	2022.05.30	1、兼容性提升 2、稳定性提升 3、修复已知问题
v5.6.6	2022.04.22	1、兼容性提升 2、优化网络稳定性 3、修复部分机型更新失败问题 4、修复已知问题
v5.5.9	2022.03.21	1、合规权限梳理优化 2、兼容适配优化 3、VIP 相关功能优化 4、新增历史帐号切换功能 5、游戏服务拉起策略优化 6、其他 bug 修复及优化
v5.4.10	2022.01.21	1、升级支付宝 SDK 2、登录界面优化，方便切换账号 3、响应政策，合规优化 4、优化了悬浮球并支持更多活动信息 5、提升对非黑鲨设备的兼容
V5.2.6	2021.12.15	1、去除 SDK 申请的所有权限 2、移除 forceDarkAllowed 标签
V5.2.5	2021.09.16	1、统一了 SDK 界面的显示颜色 2、增加了手势线的适配体验 3、提高了兼容和适配 4、修复已知问题 5、优化微信支付体验
V4.1.17	2021.08.26	1、解决收银台消失的问题 2、解决更新弹窗和悬浮球无法显示的问题

V4.1.12	2021.07.15	1、优化悬浮球菜单内容 2、修复福利活动无法跳转的问题 3、增加限时抢优惠券功能及余量显示 4、支付异常取消的反馈提示 5、增加新福利展示

2. 黑鲨游戏平台

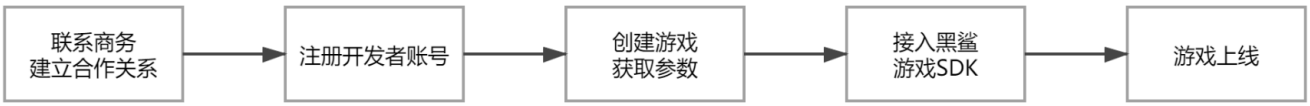
2.1. 游戏平台介绍

黑鲨游戏平台是黑鲨公司推出的面向所有黑鲨手机、JoyUI 用户的游戏平台，包含的分发渠道有游戏中心、应用商店、黑鲨浏览器、Joy 助手、黑鲨时刻、黑鲨社区等系统级资源；黑鲨游戏平台提供游戏下载、游戏搜索、游戏计费、游戏福利、游戏资讯等多种平台支撑能力，力求打造一个良好的游戏生态系统。

2.2. 黑鲨游戏 SDK 介绍

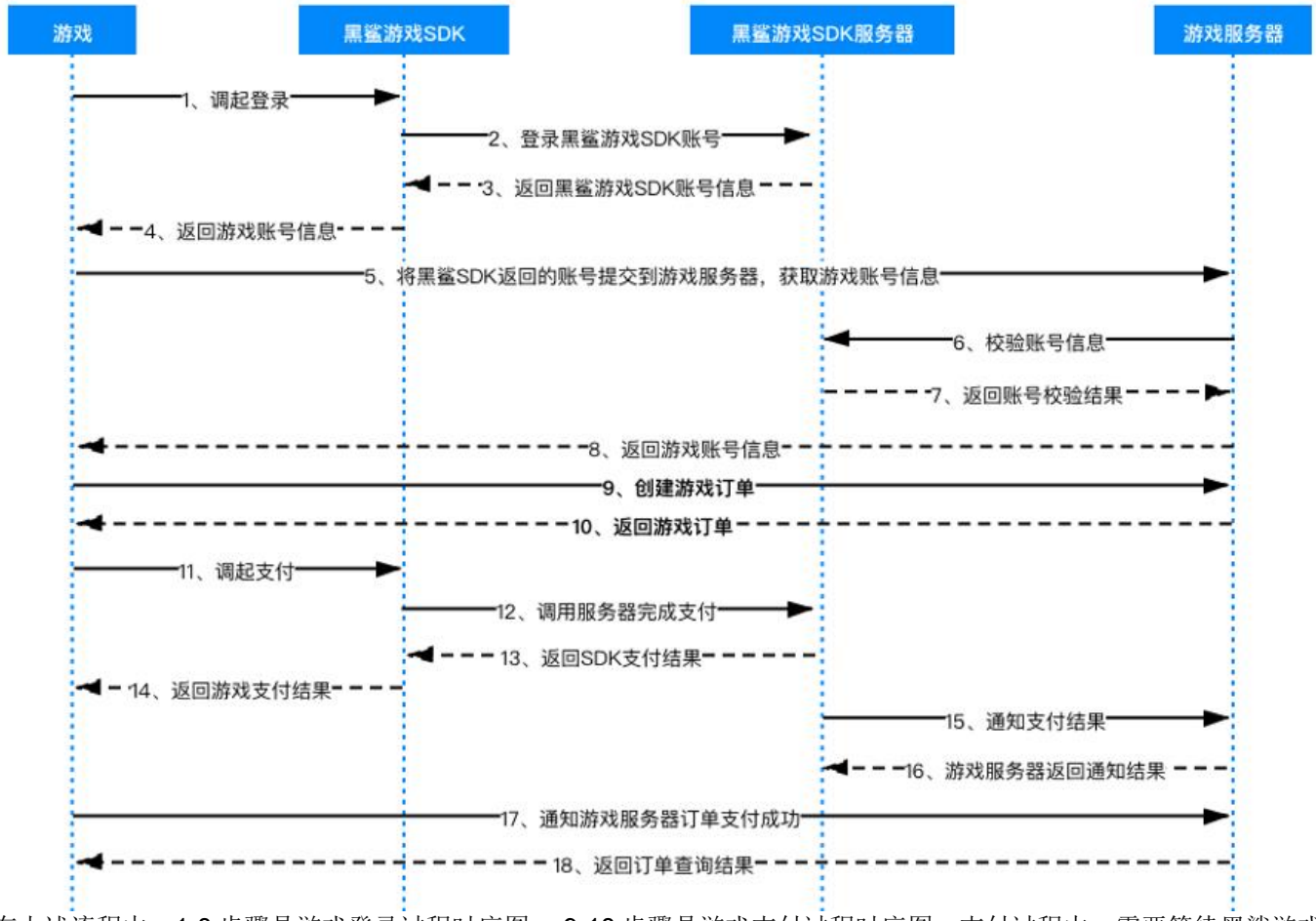
黑鲨游戏 SDK 是为联网手游打造的高质量运营开发包，为游戏提供完善的黑鲨帐号及黑鲨支付体系；同时，为联运游戏提供活动公告、福利中心等功能；后续将不断完善，为玩家及游戏厂商提供更多的贴心服务。

2.3. 黑鲨联运游戏合作流程



2.4. 游戏服务器端总体接入流程

游戏接入黑鲨游戏 SDK 后，典型的工作流程如下：



在上述流程中：1-8 步骤是游戏登录过程时序图 。9-18 步骤是游戏支付过程时序图。支付过程中，需要等待黑鲨游戏 SDK 服务器订单支付结果通知。

3. 接入前准备

3.1. 快速开始

在继续看文档之前，建议您先把随本文档一起分发的 Demo 程序（“02-Demo”文件夹内的 SharkGameSdkSample-release.apk）安装到手机，这个程序完整演示了黑鲨游戏 SDK 的工作流程，有助于您快速理解我们 SDK支付的整个流程。请注意手机 android 版本不能低于 9.0。

请参考 SharkGameSdkSample 项目。

所有的 SDK 的资源、依赖均在 SharkGameSdkSample 的 sdk 子 module 中。

应用内支付 SDK 核心是 shark_sdk.jar（位于 libs目录下，根据不同版本，名称会有所变化，以实际名称为准），需要将此 jar 包放入游戏 App 的工程中，我们已将要使用的接口封装好。

应用内支付 SDK 还需要一系列第三方支持的 jar 包，这些 jar 包也在 libs目录下。

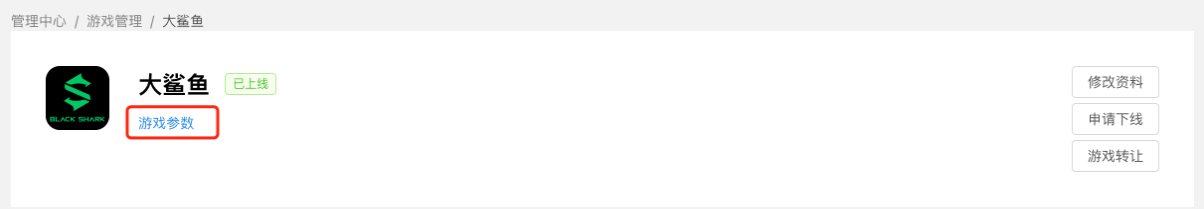
额外的还有 BSGameCenterService.apk（demo 包中未提供，可咨询黑鲨工作人员提供），目前在非JOYUI 下只提供最常用的登录（黑鲨、小米和微博账号）和支付（微信和支付宝）功能。而如果用戶安装了此 apk 并给予了相应的权限（特别是关联启动的权限）则能提供和 JOYUI 上一样的完整的登录和支付功能，可自行决定是否引导用戶安装并配置此 apk（一般是在非JOYUI用戶反馈功能问题时引导用戶进行安装和授予权限）。

3.2. 创建游戏

访问 黑鲨开放平台，注册开发者帐号等待审核通过后进入“管理中心-游戏管理-发布游戏”
注意：名（PackageName）必须以“.shark”结尾。

3.3. 获取参数

进入“管理中心”，点击游戏“管理”，进入游戏详情页点击“游戏参数”，查看 AppId、AppKey 和 AppSecret 信息：



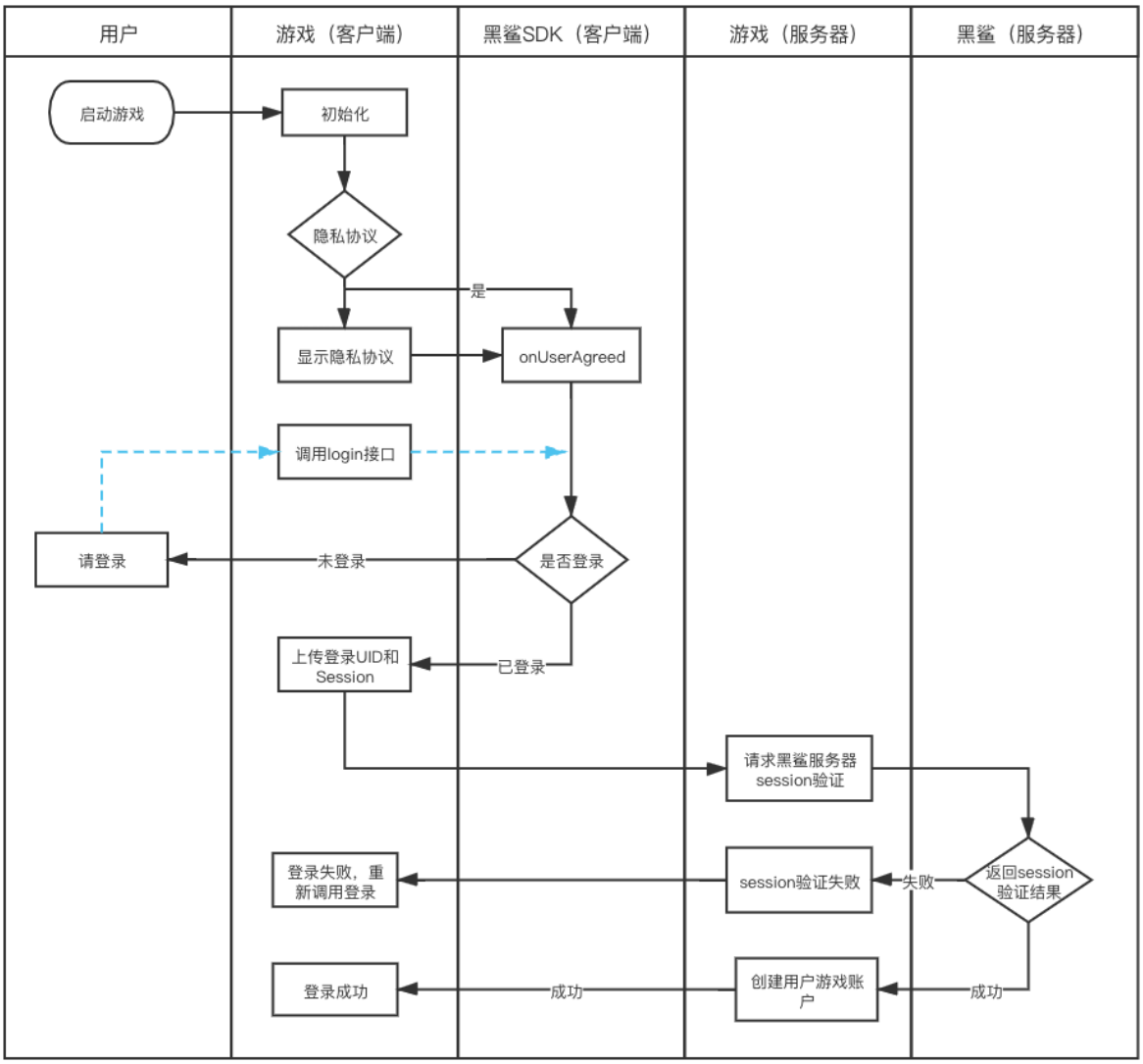
注意：AppSecret 用于服务器端签名，不要在客户端里使用

4. 客户端接入流程

4.1. 流程概述

黑鲨游戏 SDK 主要提供两方面核心功能：登录和支付。游戏每次启动，都需要先调用 login()接口来进行游戏登陆，通过黑鲨游戏平台的 uid 来创建或者查询游戏的 id 等信息。登录成功后，根据用户需要，可以调用 uniPay()接口来进行物品购买。

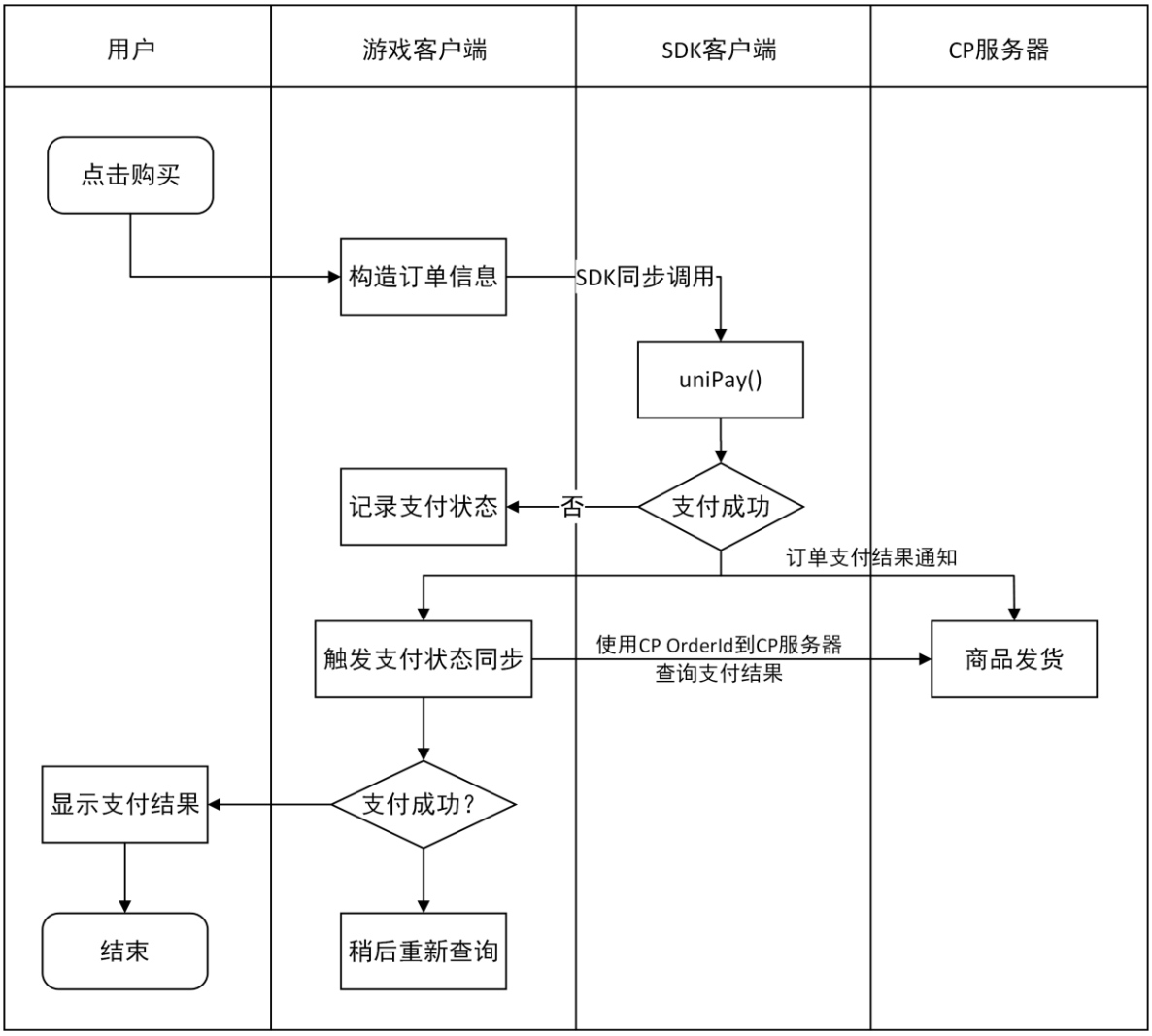
4.1.1. 登录流程概述



注意：

1. UID 不是黑鲨帐号 ID，但与黑鲨帐号 ID 有对照关系
2. 开发者必须使用这个 UID 作为用户标识，不要使用本机的 IMEI、IMSI 或 Mac 地址。上线审核阶段会检查测试

4.1.2. 支付流程概述



4.2. SDK 接入方法

4.2.1. 开发环境配置

SDK 所需要的 AndroidManifest 清单配置

```
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />

    <supports-screens
        android:anyDensity="true"
        android:largeScreens="true"
        android:normalScreens="true"
        android:resizeable="true"
        android:smallScreens="true" />

    <application>
        <activity
            android:name="com.blackshark.gamecenter.sdk.ui.SharkVerifyActivity"
            android:configChanges="orientation|screenSize"
            android:screenOrientation="behind"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />

        <activity
            android:name="com.blackshark.gamecenter.sdk.anti.ui.SharkAntiAlertActivity"
            android:configChanges="orientation|screenSize"
            android:screenOrientation="behind"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />
        <activity
            android:name="com.blackshark.gamecenter.sdk.ui.PayListActivity"
            android:configChanges="orientation|screenSize"
            android:exported="false"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />

        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />

        <activity
            android:name="com.alipay.sdk.app.H5PayActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="false"
            android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
        <activity
            android:name="com.alipay.sdk.app.H5AuthActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="false"
            android:theme="@android:style/Theme.NoTitleBar" >
        </activity>
        <activity
            android:name="com.alipay.sdk.app.PayResultActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="true"
            android:launchMode="singleInstance"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" >
        </activity>
        <activity
            android:name="com.alipay.sdk.app.AlipayResultActivity"
```

```
android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="true"
            android:launchMode="singleTask"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" >
        </activity>
        <activity
            android:name="com.alipay.sdk.app.H5OpenAuthActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="false"
            android:screenOrientation="behind"
            android:windowSoftInputMode="adjustResize|stateHidden" >
        </activity>
        <activity
            android:name="com.alipay.sdk.app.APayEntranceActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize|locale|keyboard|screenLayout|d
ensity|fontScale|layoutDirection|smallestScreenSize"
            android:exported="false"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" >
        </activity>

        <activity
            android:name="com.blackshark.gamecenter.sdk.ui.PaymentActivity"
            android:configChanges="orientation|screenSize"
            android:exported="true"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />

        <activity
            android:name="com.blackshark.gamecenter.sdk.ui.InstallAppTransitionActivity"
            android:configChanges="orientation|screenSize"
            android:screenOrientation="behind"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" >

            <intent-filter>
                <action android:name="${applicationId}.shark.InstallAppTransitionActivity" />

                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
            </intent-filter>
        </activity>

        <provider
            android:name="com.blackshark.gamecenter.sdk.provider.BsFileProvider"
            android:authorities="${applicationId}.shark_provider"
            android:exported="false"
            android:grantUriPermissions="true">

            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths" />
        </provider>
        <activity
            android:name="com.blackshark.gamecenter.sdk.ui.SharkActivity"
            android:configChanges="orientation|screenSize"
            android:screenOrientation="behind"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />
    </application>
```

游戏包主题设置

建议将 AndroidManifest 中 application 主题设置为
android:theme="@android:style/Theme.Light.NoTitleBar.Fullscreen",
设置为其它主题可能导致某些弹框样式不正确，xml 如下所示：

```
<application
    ...
    android:theme="@android:style/Theme.Light.NoTitleBar.Fullscreen">
```

SDK 所需的 android support 依赖

android studio

```
implementation 'com.android.support:support-v4:27.1.1'
implementation 'com.google.code.gson:gson:2.8.6'
implementation(name: 'analytics-release-2.1.12', ext: 'aar')
```

eclipse

见 02-Demo/eclipse_support/

游戏如果设置 targetSdkVersion 为 30 或更高，为了适配 Android 11，需要在 AndroidManifest 增加以下配置

```
<queries>
    <package android:name="com.blackshark.gamesdkapp" />
    <package android:name="com.blackshark.bsamagent" />
    <package android:name="com.blackshark.bsaccount" />
    <package android:name="com.tencent.mobileqq" />
    <package android:name="com.tencent.mm" />
    <package android:name="com.sina.weibo" />
    <package android:name="com.xiaomi.account" />
    <package android:name="com.eg.android.AlipayGphone" />
    <package android:name="com.eg.android.AlipayGphone" />
    <package android:name="com.eg.android.AlipayGphoneRC" />
    <package android:name="hk.alipay.wallet" />
    <package android:name="com.tencent.cmocmna" />
</queries>
```

SDK 依赖 analytics_release-2.1.12 库，需设置项目的 minSdkVersion 为 19 或以上，游戏的 minSdkVersion 低于 19 可在 AndroidManifest 文件中添加以下配置避免打包失败：

```
<uses-sdk tools:overrideLibrary="com.blackshark.analyticssdk" />
```

Unity 4 兼容性 如果您使用 Unity 4，请在继承了 UnityPlayerNativeActivity 的标签中添加如下配置：

```
<activity android:name="com.unity3d.player.UnityPlayerNativeActivity"
    android:label="@string/app_name">
    ...其它配置
    <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
    <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="true" />
</activity>
```

Unity 5.6.x 兼容性 Unity5.6.x 上无法显示悬浮窗等解决方案：自定义 UnityPlayer，重写 addView 方法，将 SurfaceView 的 zOrderOnTop 设为 false。示例代码如下

```
 /****************************自定义 UnityPlayerActivity****************************/
package com.xxx.yyy;

import com.unity3d.player.*;
import android.os.Bundle;

public class CUnityPlayerActivity extends UnityPlayerActivity {
    @Override
    public void onCreate(Bundle bundle) {
        requestWindowFeature(1);
        super.onCreate(bundle);
        getWindow().setFormat(2);
        //mUnityPlayer = new UnityPlayer(this);
        mUnityPlayer = new CUnityPlayer(this);
        setContentView(mUnityPlayer);
        mUnityPlayer.requestFocus();
    }

/****************************自定义 UnityPlayer****************************/
package com.xxx.yyy;

import com.unity3d.player.*;
import android.content.ContextWrapper;
import android.view.SurfaceView;
import android.view.View;

public class CUnityPlayer extends UnityPlayer {
    public CUnityPlayer(ContextWrapper contextwrapper) {
        super(contextwrapper);
    }

    public void addView(View child) {
        if (child instanceof SurfaceView) {
            ((SurfaceView)child).setZOrderOnTop(false);
        }
        super.addView(child);
    }
}
```

4.2.2. SDK 初始化

1、在黑鲨开发者后台创建应用并获取 AppId 、AppKey 和 AppSecret，创建应用时如果是游戏 PackageName 必须以 “.shark” 为后缀。

注意：AppSecret 是用于服务器端签名，不要在客戶端里使用

2、将 SDK 包中 res目录中的资源，完整添加到游戏 Android工程的 res目录。如果样式有冲突，可酌情修改。
3、将 SDK 包中 libs目录下的文件放在你的项目的 libs目录下（核心jar 包 shark_sdk.jar 会根据不同版本，名称会有所变化，以实际名称为准，部分第三方包也可能由于版本不同而有变化，以实际为准），在 buildpath 中引用，然后对 SDK 进行初始化。
****注意：需要检查下面的一致性，如果不一致会导致调用登录和其它 SDK 接口失败 ****

1. 在黑鲨开发者站后台配置好 AppId/AppKey 和包名；
2. 检查 AndroidManifest.xml里面所设置的 package 必须与开发者站配置的一致；
3. 上传对应签名的 apk 到黑鲨开发者站后台（可不申请审核）；
4. 在 Application.onCreate 中调用初始化方法；
5. 在游戏主 activity 中，显示隐私协议，并且在用戶选择之后，调用onUserAgreed方法。

4、在 Application.onCreate 中调用初始化方法： **重要!!!** 防沉迷功能已在 SDK 内集成，会要求初始化必须放在 AndroidManifest 中声明的 Application 类中，请务必注意！

```
 SharkGameInfo gameInfo = new SharkGameInfo();
gameInfo.setAppId("请申请获得");
gameInfo.setAppKey("请申请获得");

 SharkOpenPlatform.init(this, gameInfo, new OnInitProcessListener() {
     @Override
    public void finishInitProcess(List<String> loginMethod, int gameConfig) {
        Log.i("Demo", "init success");
    }

    @Override
    public void onSharkSplashEnd() {
      //黑鲨闪屏页结束回调，黑鲨闪屏可配，无闪屏也会返回此回调，游戏的闪屏应当在收到此回调之后开始。
        //游戏自己的闪屏处理，可参考 SplashActivity 的实现
    }
});
```

5、隐私协议：用户在第一次打开游戏时，需要显示相关隐私协议，用户选择之后，调用 SharkOpenPlatform.getInstance().onUserAgreed(Activity activity)；告知 SDK 用户是否同意隐私协议。

```
 请确保该方法在游戏场景所在的 Activity 中调用，避免在类似开屏等界面中调用

@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate( savedInstanceState );
   setContentView( R.layout.activity_main );
   //……隐私协议流程...
   if (agreed){ // 用户同意隐私协议
      SharkOpenPlatform.getInstance().onUserAgreed(this);

   } else {
     // 退出游戏或者其它处理
   }
}
```

该接口必须在用户选择同意隐私协议(游戏自己的隐私协议)之后调用，否则无法正常登录；若用户拒绝隐私协议，不可调用该接口，且无法登录。

注：

1. SDK 本身不会弹出任何隐私协议，隐私协议游戏内自行设计实现。
2. **同意隐私协议接口需要在主线程(UI 线程)调用!**

4.2.3. 登录调用代码

```
SharkOpenPlatform.getInstance().login(activity, new OnLoginProcessListener() {
        @Override
        public void finishLoginProcess(int code, SharkAccountInfo accountInfo) {
            switch (code) {
                case SharkErrorCode.SHARK_LOGIN_SUCCESS: // 登陆成功
                    // 获取用戶登陆后的 UID（即用戶唯一标识）
                    String uid = accountInfo.getUid();
                    // 以下为获取 session 并校验流程，如果是网络游戏必须校验，如果是单机游戏或应用可选
                    // 获取用戶的登陆的 Session（用于后续流程校验 Session 有效性）
                    String session = accountInfo.getSession();
                    // 请开发者完成将 uid 和 session 提交给开发者自己服务器进行session 验证
                    break;
                case SharkErrorCode.SHARK_LOGIN_ERROR_CANCEL:
                    // 取消登录
                    break;
                case SharkErrorCode.SHARK_LOGIN_ERROR_REPEAT_OPERATION:
                    // 登录操作正在进行中
                    break;
                case SharkErrorCode.SHARK_LOGIN_FAILED:
                default:
                    // 登录失败
                    break;
            }
        }
    });
```

可以通过实现 OnLoginProcessListener 接口来捕获登录结果。
注：登录接口需要在主线程(UI 线程)中调用!

4.2.4. SDK 支付接口参数说明

需要 CP 服务器端配合生成 sign，返回如下表格数据给 CP 客户端调用 SDK 支付接口用。
客户端需要请求服务器生成 sign，返回给客户端，客户端调用支付接口需要用到 sign。

字段名称	对应客户端 SharkBuyInfo 属性	字段类型	必填	是否参与签名	字段说明
app_id	appId	String	Y	Y	游戏 ID(不能为空)
product_code	productId	String(32)	Y	Y	CP 游戏道具 ID ，可为空
product_body	productName	String	Y	Y	游戏道具名称(不能为空)
product_detail	productDesc	String(255)	Y	Y	游戏道具说明 ，可为空
total_amount	price	int	Y	Y	总金额，单位分(不能为空)

out_trade_no	orderId	String(32)	Y	Y	CP 订单 ID(不能为空)
notify_url	payNotifyUrl	String	Y	Y	支付回调 URL(不能为空)
uid	uid	String	Y	Y	黑鲨游戏账户用户 ID(不能为空)
time_start	timeStart	String	Y	Y	订单创建时间，格式为 yyyyMMddHHmmss，可为空
time_expire	timeExpire	String	Y	Y	订单超时时间，格式为 yyyyMMddHHmmss，可为空
sign	sign	String	Y	Y	参数签名(不能为空)
buyNum	buyNum	int	Y	N	道具购买的数量，不参与签名

签名说明：
拼接签名串：key=value&key=value，key 按照字母升序，空值""参与签名；
sign=MD5(拼接签名串&app_key=申请的 appSecret)（注意这里键名是 app_key，键值是申请的 appSecret），签名后 sign 值全部大写，其中 appSecret 是从开发者后台获取到的参数。

备注：sign 生成最好在服务器端做，同时还可以在服务器端保存对应 cp 自己生成的订单信息，方便后续支付成功回调根据 cp 订单号查询对应订单，并校验回调金额和订单金额是否一致。

如对生成 MD5 还有疑问，参考 Demo 里获取 MD5 签名的示例。

4.2.5. 支付调用代码

```
        SharkBuyInfo payParams = new SharkBuyInfo();
        payParams.setAppId(DemoApp.gameInfo.getAppId()); // 游戏 ID(不能为空)，必填
        payParams.setProductId(productId); // CP 游戏道具 ID，可为空，必填
        payParams.setProductName(productName); // 游戏道具名称(不能为空)，必填
        payParams.setProductDesc(productDesc); // 游戏道具说明，可为空，必填
        payParams.setOrderId(orderId); // CP 订单 ID(不能为空)，必填
        payParams.setPrice(price); // 总金额，单位 分(不能为空)，必填
        payParams.setBuyNum(buyNum); // 道具购买的数量，不参与签名，必填
        payParams.setPayNotifyUrl(notifyUrl); // 支付回调 Url(不能为空)，必填

        long current = System.currentTimeMillis();
        payParams.setTimeStart(DateUtils.formatYMDHMS(current)); // 订单创建时间，格式为
yyyyMMddHHmmss，可为空，必填
        payParams.setTimeExpire(DateUtils.formatYMDHMS(current + 30 * 1000 * 60L)); // 订单超时时间，
格式为 yyyyMMddHHmmss，可为空，必填
        payParams.setUid(uid); // 黑鲨游戏账户用户 ID(不能为空)，必填

        Map<String, String> orderParam = OrderUtils.buildOrderParam(payParams);
        String sign = OrderUtils.getSign(orderParam, "申请的 appSecret"); // 这里使用申请的 AppSecret 进行
签名
        payParams.setSign(sign); // 参数签名(不能为空)，必填

        SharkOpenPlatform.getInstance().uniPay(activity, payParams, new OnPayProcessListener() {
            @Override
            public void finishPayProcess(int code) {
                switch (code) {
                    case SharkErrorCode.SHARK_PAYMENT_SUCCESS:
                        // 购买成功
                        break;
                    case SharkErrorCode.SHARK_PAYMENT_ERROR_PAY_CANCEL:
                       // 取消购买
                        break;
                    case SharkErrorCode.SHARK_PAYMENT_ERROR_LOGIN_FAIL:
                       // 未登录
                        break;
                case SharkErrorCode.SHARK_PAYMENT_ERROR_ACTION_EXECUTED:
                        // 操作正在进行中
                        break;
                    case SharkErrorCode.SHARK_PAYMENT_ERROR_PAY_FAILURE:
                    default:
                        // 购买失败
                        break;
                }
            }
        });
    }
```

收银台保活（打开收银台时，跳出游戏或回到桌面等操作后，再次回到游戏，需要保留收银台状态），如收银台消失了，可尝试此方法：
把游戏启动 activity 的 launchMode 设成 standard。 若游戏启动 activity 同时是主 activity，不能修改 launchMode，建议增加一个 launchMode=standard 的闪屏 activity 作为启动入口。

注: 支付接口需要在主线程(UI 线程)中调用!

4.2.6. 游戏退出接口

SDK 自带了退出弹窗，游戏不需要自己实现，推荐重写 activity 的 onKeyDown 方法，当用户未登录时，游戏自己处理退出逻辑；用户已登录则调用 SDK 接口处理退出逻辑。帐号退出或注销时，在适当的地方调用退出接口。可参考 demo 处理逻辑。

```
SharkOpenPlatform.getInstance().appExit(activiy, new OnExitListener() {
    @Override
    public void onExit(int code) {
        if (code == AppConstant.PROMOTE_ON_WINDOW_EXIT) {
            Process.killProcess(Process.myPid());
        }
    }
});
```

注:退出游戏接口需要在主线程(UI 线程)中调用!

4.2.7. 提交游戏角色信息接口（必须接入，用于支持更多平台活动）

调用时机：
1. 创建角色/已有角色进入游戏后提交游戏角色数据；
2. 游戏内升级时调用提交游戏角色数据。

```
GameRoleInfo info = new GameRoleInfo();
info.setRoleId("404");
info.setRoleName("一鸣");
info.setRoleLevel(99);
info.setRolePower(999);
info.setRoleServerId("1000");
info.setRoleServerName("国服 001");
info.setRoleZoneId("b1");
info.setRoleZoneName("桃花岛");

SharkOpenPlatform.getInstance().submitRoleInfo(activity, info);
```

4.2.8. 混淆文件说明

需要注意，SDK 主要以 jar 包提供给开发者，SDK 不用混淆，您在混淆自己游戏的 apk 包时，需要在 proguard-rules.pro 里加入，以避免二次混淆。

```
#----------------library-----------#

#libraryjars alipaySdk**.jar
-dontwarn com.alipay.**
-keep   class com.alipay.** { *; }

#zxing
-dontwarn com.google.zxing.**
-keep class com.google.zxing.** {*;}

# gson
-keep class * implements com.google.gson.TypeAdapterFactory
-keep class * implements com.google.gson.JsonSerializer
-keep class * implements com.google.gson.JsonDeserializer
-keepclassmembers,allowobfuscation class * {
  @com.google.gson.annotations.SerializedName <fields>;
}
-dontwarn sun.misc.**
-keep class com.google.gson.stream.** { *; }

#shark_sdk.jar
-dontwarn com.blackshark.**
-keep class com.blackshark.** {*;}
-dontwarn com.leon.channel.**
-keep class com.leon.channel.** {*;}
#Glide
-dontwarn com.bumptech.glide.**
-keep class com.bumptech.glide.**{*;}
-keep public class * implements com.bumptech.glide.module.GlideModule
-keep class * extends com.bumptech.glide.module.AppGlideModule {
 <init>(...);
}
-keep public enum com.bumptech.glide.load.ImageHeaderParser$** {
  **[] $VALUES;
  public *;
}
-keep class com.bumptech.glide.load.data.ParcelFileDescriptorRewinder$InternalRewinder {
  *** rewind();
}
```

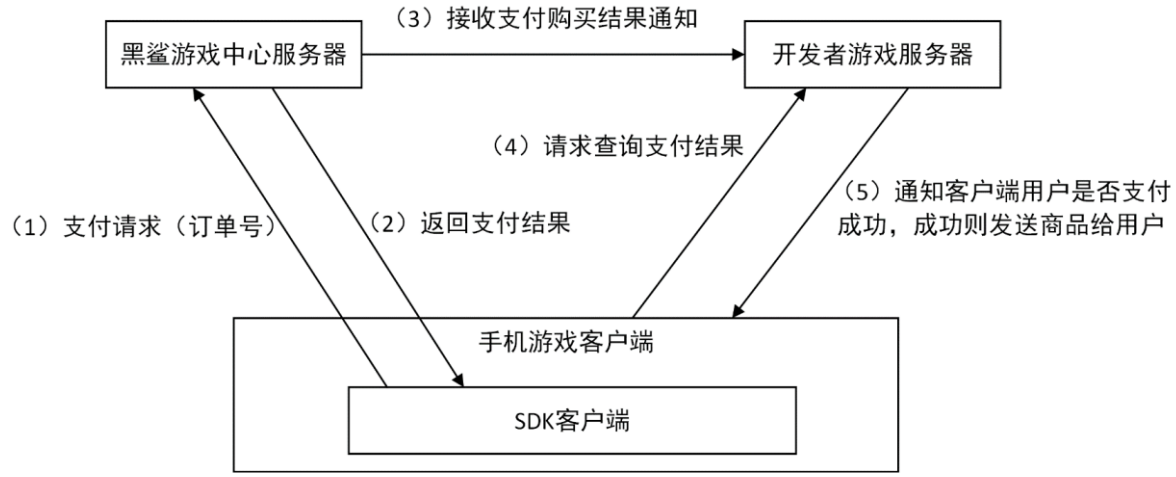
5. 服务端接入说明

本部分主要提供黑鲨游戏平台“SDK 服务器”和黑鲨游戏合作商“CP 游戏服务器”的交互的接口规范。

5.1. 订单支付结果通知

5.1.1. 流程说明

此接口由开发者负责开发。在订单支付成功后，黑鲨游戏支付服务器会将支付结果通知给开发者预先提供的服务器上。若开发者所提供的服务器地址不可用，在一定时间段内游戏平台服务器会按照周期进行轮询（前 10 次，每分钟通知 1 次；10 次后每小时通知 1 次）
具体流程如下：



注：由于是异步通知模型，（3）和（4）不一定是按序号产生。因此（4）和（5）需要进行轮询处理。

5.1.2. 接口及参数说明

接口地址：各开发者服务器的通知地址
请求方法：POST，数据格式为 JSON 格式

请求参数说明

名称	类型	重要性	游戏 ID
appId	string	必须	商户订单号
cpOrderNo	string	必须	商户订单号
bsOrderNo	string	必须	黑鲨订单号
orderStatus	int	必须	订单状态 1 代表成功
timeStart	string	可选	交易开始时间，订单生成时间，格式为 yyyyMMddHHmmss，如 2009 年 12 月 25 日 9 点 10 分 10 秒表示为 20091225091010
timeExpire	string	可选	订单失效时间，格式为 yyyyMMddHHmmss，如 2009 年 12 月 27 日 9 点 10 分 10 秒表示为 20091227091010。
totalAmount	int	必须	订单总金额，单位为分
productCode	string	可选	商品码
productBody	string	必须	商品描述
productDetail	string	可选	商品详情
channelId	int	必须	1:支付宝 2:微信

uid	string	必须	用户 ID
sign	string	必须	签名算法（后面说明）

返回参数说明

打印输出“success”（不包含引号）。

注意：对于同一个订单号的多次通知，开发商要自己保证只处理一次发货。

服务器 IP 地址

134.175.254.175

134.175.34.111

106.55.241.94

请开发商服务器开发人员加到 ip 白名单内，以免因为 ip 限制造成回调不成功。

验签

第一步： 在通知返回参数列表中，除去 sign 参数外，凡是通知返回回来的参数皆是待验签的参数；

第二步： 将剩下参数进行字典排序，使用 URL 键值对的格式（即 key1=value1&key2=value2…）拼接成字符串 stringA；

第三步： 在 stringA 最后拼接上 app_key=申请的 appSecret（这里键名用 app_key，键值用申请的 appSecret）得到 stringSignTemp 字符串，并对 stringSignTemp 进行 MD5 运算，再将得到的字符串所有字符转换为大写，得到 sign 值 signValue。

5.2. 用户 session 验证接口（必须）

此接口用于验证登录账户的有效性。

注意：用户的唯一标识是通过 SDK 获得的 uid，而不是 Session，Session 用于校验登录验证的有效性，必须经过 SDK、游戏中心服务器、开发者服务器进行三方验证，如果 Session 失效，需要重新调用登录。

5.2.1. 接口及参数说明

接口地址：https://gamesdk.blackshark.com/loginvalidate

请求方法：POST，数据格式为 JSON 格式

请求参数说明

名称	类型	重要性	说明
appId	string	必须	游戏 ID
session	string	必须	session
uid	string	必须	用户 ID
sign	string	必须	签名算法（后面说明）

返回参数说明

名称	类型	重要性	说明

errcode	string	必须	状态码 200：正确

102005：appId 错误；
102006：uid 错误；
109011：sign 错误；
102009： uid, session 不匹配(常为 session 过期)；
2：签名错误
例如：{ errcode: 200 }

签名

第一步： 在通知返回参数列表中，除去 sign 参数外，皆是待验签的参数；
第二步： 将剩下参数进行字典排序，使用 URL 键值对的格式（即 key1=value1&key2=value2…）拼接成字符串 stringA；
第三步： 在 stringA 最后拼接上 app_key（&app_key=123456）得到 stringSignTemp 字符串，并对 stringSignTemp 进行 MD5 运算，再将得到的字符串所有字符转换为大写，得到 sign 值 signValue。

示例：

如 appId 为 2500000000，uid 为 1000000000，session 为 4d992b55662b5262817a8c49005dad1c，app_key 为 53ds9923ebbc84833b76bdd9ec0e9e6db，则请求 body 如下：

```
{
   "appId":"2500000000",
   "uid":"1000000000",
   "session":"4d992b55662b5262817a8c49005dad1c",
   "sign":"56B2A231896910072A5EE7DC3F135AB4"
}
```

sign 生成代码如下（Go 语言）：

```
func main() {
   params := make(map[string]string)
   params["appId"] = "2500000000"
   params["session"] = "4d992b55662b5262817a8c49005dad1c"
   params["uid"] = "1000000000"
   key := getSignString(params, "53ds9923ebbc84833b76bdd9ec0e9e6db")
   sign := encode(key)
   fmt.Printf("sign======%v\n", sign)
}

func encode(key string) string {
   h := md5.New()
   h.Write([]byte(key))
   return strings.ToUpper(hex.EncodeToString(h.Sum(nil)))
}

func getSignString(m map[string]string, devAppKey string) string {
   keys := make([]string, len(m))
   i := 0
   for k, v := range m {
      keys[i] = k + "=" + v
      i++
   }
   sort.Strings(keys)
```

```
    return strings.Join(keys, "&") + "&app_key=" + devAppKey
}
```