

sdk_base.js
<div>入口: sdk.clientVersion = “1.0 build xxx” sdk.gameId = 0; sdk.token = null; sdk.frames = {}; sdk.gameInfo = null; sdk.shareQRInfo = null; sdk.headImg = null;</div> <div> sdk.init(); //sdk.token从当前url中获取</div>
<div> sdk.init //sdk.token = 从当前url中获取</div> <div>config 配置SDK： 配置gameID, 添加onShare onPay回调, 并 postTopMessage (“config”)</div> <div>logout 登出 showFocus 显示关注二维码遮罩层 showShare 显示分享遮罩层 shareTitle shareIcon shareDesc shareParams 设置分享标题、图标、分享语、 参数: 参数必须以cp_开头 getShareBackup 获取备份的分享数据</div> <div>setChatPrevent 设置聊天限制 getShareBackup 获取备份的分享数据</div> <div>showFriend hideFriend 显示、隐藏好友 copy 复制 showTopImg 显示顶层图片 joinChatRoom 游戏聊天 bindCash 绑定提现 gotoBBS 进入社区 downJetApp 下载微端 refresh 刷新主页面 stat 统计 getGameInfo 获取游戏信息 getShareQRInfo 获取分享二维码信息 pay 支付 getHeadImg 获取头像信息</div> <div>isQQ isWeixin isAndroid isiOS isPCWeixin isMobile isJetAPP isJetboxAPP isSafari isDesktopApp 判断是否是主屏幕微端 (ios)</div> <div>getURLQuery 获取URL中所有参数对象 getURLVar setURLVar cleanURLVar 操作URL中的参数 buildURL getServerURL saveGameDesktop 保存游戏至桌面 isAllowPay</div> <div>setItem getItem removeItem //本地存储 setSession getSession removeSession //本地对话</div> <div>stopRecord stopRecord uploadVoice playVoice stopVoice 录音相关</div> <div>randomString 随机一个字符串 randomNumber 随机一个数字</div> <div>createFrame 创建一个FRAME loadSingleScript 加载单个JS文件 appendSingleScript 向DOM追加单个JS文件引用 createQRCode 生成二维码</div> <div>addShortcut 创建桌面快捷方式</div> <div>__onAddShortcut 发送到桌面回调 httpGet HTTP GET请求 postTopMessage 向上发送信息</div>

sdk_host.js
<div>入口: sdk.hostVersion = “1.0 build xxx” ; sdk.wxappId = “xx”; sdk.wxopenId = “xx”; sdk.qqappId = “xx”; sdk.gameId = 0; sdk.shareCallback = null; sdk.readyCallback = null; sdk.readyMsgList = []; sdk.code = null; sdk.token = null; sdk.uid = null; sdk.shareDatas = null; sdk.isFocus = false; sdk.isHideFocus = false; sdk.isBridgeShareReady = false; sdk.channelInfo = null; sdk.getShareDomain = null; sdk.getPreventInfo = null; sdk.isTrial = 1; //试玩账号 (1: 未绑定, 0: 已绑定) sdk.apiHost = “https://aip.xx.com”; sdk.userHost = “https://user.xx.com”; sdk.loginHost = “https://login.xx.com”; sdk.webHost = “https://web.xx.com”; sdk.cdnHost = “https://cdn.xx.com”; sdk.onlineHost = “https://online.xx.com”;</div> <div>\$("#share_logo").remove(); \$("body").prepend(share_logo);</div>
<div>initHost sdk.auth wxQrcodeLogin qqLogin sinaLogin wxAppLogin redirectAuth(pf, gameid, nocache) // pf 1 微信 2 微信二维码 3 手机QQ</div> <div>createCode // token交换code checkCode // code交换token与uid checkToken // token交换uid getChannelInfo //获取CPS配置 checkDevice //获取设备类型 wxReady // 初始化完成 wxInit // 初始化环境 wxApplnit // 初始化APP分享环境 userPlay // 记录最后一次玩的游戏</div> <div>confirmShare // 分享统计 showFocus // 执行显示关注功能 showShare hideShare // 执行显示、隐藏分享层功能 logout // 执行登出功能 showFriend hideFriend // 执行显示、隐藏好友功能 shareDesc shareTitle shareIcon shareParams // 执行设置分享语、标题、图标、参数 功能 stat // 执行统计功能 showTopImg // 执行显示顶层图片 copy // bindCash // 执行绑定提现 showLoading hideLoading // 显示、隐藏 LOADING遮罩层</div> <div>closePayBox // 关闭支付弹出层 closePayRule //关闭条例 loadPayBox // 唤醒支付弹出层 pay // 执行支付功能 wxPay // 微信支付 checkTransStatus // 检查订单支付状态: true 完成 false 未支付 wxAppPay // 微信APP支付 qqPay // QQ钱包支付 wxh5Pay // 微信H5支付 wxqrcodePay // 微信二维码支付 alih5Pay // 支付宝H5支付 aliPay // 支付宝支付 aliAppPay // 支付宝APP支付 checkNotify // 检查是否有通知 (从服务器) initWebViewJavascriptBridge // 初始化苹果APP initAndroidAPPShare // 初始化安卓APP分享 yybPay // 应用宝: 支付 yybLogout // 应用宝: 登出 yybAddShortCut // 应用宝: 添加桌面</div> <div>getShareQRInfo // 获取分享二维码信息 shareDomain // 备份的分享信息 execMessage // 开始侦听消息 startRecord stopRecord uploadVoice downloadVoice playVoice stopVoice // 录音相关</div> <div>downJetApp // 微端下载 saveGameDesktop // 保存到桌面 statOnline // 统计在线 hideTrialBox // 关闭试玩界面 loadTrialInfo // 加载试玩个人信息界面 loadTrialLogin // 加载试玩登录界面 enableExitBanner // 启用返回Banner enableFloatBox // 启用小浮窗按钮 jetLoginError // 多纷账号登录错误码 mobileCheckError // 手机登录*异常处理</div> <div>//自定义账号登录 jetLoginHasCode = null; //是否需要验证码 jetLogin // 登录 mobileObjectBase // 手机登录注册绑定接口 getVerifyImg // 获取验证码 getSmsCode // 获取短信验证码 smsCountDown // 统一短信验证码倒计时 mobileParamsCheck // 绑定登录参数检查 loadLoginBox // 唤醒登录窗口 closeLoginBox // 关闭登录窗口 hideConfirmDialog // 关闭选择弹窗 loadHeadImg // 微信头像加载 8*</div>

index

入口:

var chid = sdk.getURLVar("chid");
var query = sdk.getURLQuery(location.href);
var trial = sdk.getURLVar("trial");
var isShowFloatBox = true;
var isShowExitBanner = false;
sdk.gameInfo = {
 "url": "http://union.xxx.com/login.html?gameid=157",
 "name": "GameNameXXX",
 "game_url": "",
 "type": 2,
 "dc_appid": "1",
 "td_appid": "XXXXXXX", //统计ID
 "ext": {
 "desktopIcon": "http://game.xxx.com/static/xx/xx.png"
 },
 "token_type": 1,
 "orientation": 0,
 "wx_option": 1,
 "entry_url": "http://play.XXX.com/game/?gameid=xxx",
 "content_url": "http://jump.h5.XXX.com/webserver/jet",
 "use_vucoupon": 0
}
sdk.desktopIcon = "http://game.xxx.com/xxx/xxx.png";

desktopCheck();
initSplash();
start();

prepareEnv // 准备环境: CPS等
postGameMessage
onShare
onPay
initStat // 初始化统计
sendStat // 执行统计
desktopCheck // 设置桌面按钮、分享图片
initSplash 初始化宣传图
start()

Index loadGame流程

loadGame(url,token,orien)
逻辑

token是否
存在

否
sdk.auth()

是

设置 args:{
 "focus": "sdk.isFocus",
 "trial": "全局 (从URL中获取) ",
 "cp_*": "cp_* (从URL中获取) "
}

准备环境: CPS等
prepareEnv(callback{
 gameURL=sdk.buildURL(gameURL, args + exArgs);
 然后用iframe加载游戏
 //然后根据条件显示浮动按钮及ExitBanner
 sdk.enableExitBanner();
 sdk.enableFloatBox();
})

sdk.checkDevice();

Index -> start 主逻辑流程

Start()

token是否
存在

是
createCode,后台返回
code
然后重新加载本页。

否 : sdk.initHost

sdk.checkNotify()
isJetAPP && IOS
 sdk.initWebView.JavascriptBridge(function () {
 if (window.WebView.JavascriptBridge) {
 window.WebView.JavascriptBridge
 .callHandler("setTitle", document.title);
 }
 });
isJetAPP && isAndroid
 window.android.setTitle(document.title);

// 初始化
sdk.wxInit(onShare, function () {
 if (sdk.isWeixin()) && !sdk.gameInfo["wx_option"] {
 // 隐藏菜单
 wx.hideOptionsMenu();
 }
});
var gameURL = decodeURIComponent(sdk.gameInfo["content_url"]);
// 根据所需token类型传递
if (sdk.gameInfo["token_type"]) {
 // 需要的是userToken
 sdk.getUserToken(sdk.token, function (data) {
 loadGame(gameURL, data["userToken"], orientation);
 });
} else {
 // 需要的是token
 loadGame(gameURL, sdk.token, orientation);
}
sdk.userPlay(); // 记录最后一次玩的游戏
// 启用统计
if (sdk.gameInfo["td_appid"]) {
 initStat(sdk.gameInfo["td_appid"], sdk.uid);
}

window.addEventListener("message",...){
 config:
 stat: sendStat(event.data.args["item"])
 pay: sdk.pay(event.data.args, onPay)
 showShare:
 QQ/WX/App/Mob: onShare("onMenuShareAppMessage")
 否则: sdk.showShare(event.data.args);
 addShortcut:
 getShareQRInfo:
 sdk.getShareQRInfo(function (data) {
 postGameMessage({cmd: "getShareQRInfo", args: data})
 }, event.data.args)
 getShareDomain:
 sdk.shareDomain(function (data) {
 postGameMessage({cmd: "getShareDomain", args: data})
 });
 startRecord:
 sdk.startRecord(function (data) {
 postGameMessage({cmd: "startRecord", args: data});
 });
 //其它语音部分
 default:
 sdk.execMessage(event.data); //该函数在分享那部分
}

sdk.initHost 逻辑流程

开始
sdk.code = sdk.getURLVar("code"); ?
sdk.gameId = sdk.getURLVar("gameid"); ?
sdk.token = sdk.getItem("token"); 从本地取

tokenkey &&
主屏幕微端 (ios)

是
if(sdk.token)

否
if(sdk.token &&
与本地缓存code不同)

是
__checkTokenKey

否
__checkToken

是
sdk.auth
参见鉴权相关

否
__checkCode

访问Server
成功

成功&&
data["token"]

是
sdk.setItem("token", data.token);
sdk.token = sdk.getItem("token");

否
tokenkey &&
主屏幕微端 (ios)

是
sdk.removeItem("token");

否
访问Server
成功

是 &&
data["uid"]

是 &&
data["token"]

是 &&
data["uid"];
sdk.isFocus = data["focus"] > 0;
sdk.isTrial = data["trial"];

是 &&
data["token"];
sdk.uid = data["uid"];
sdk.isFocus = data["focus"] > 0;
sdk.isTrial = data["trial"];

__checkSuccess

isJetAPP

否
sdk.statOnline();
callback();

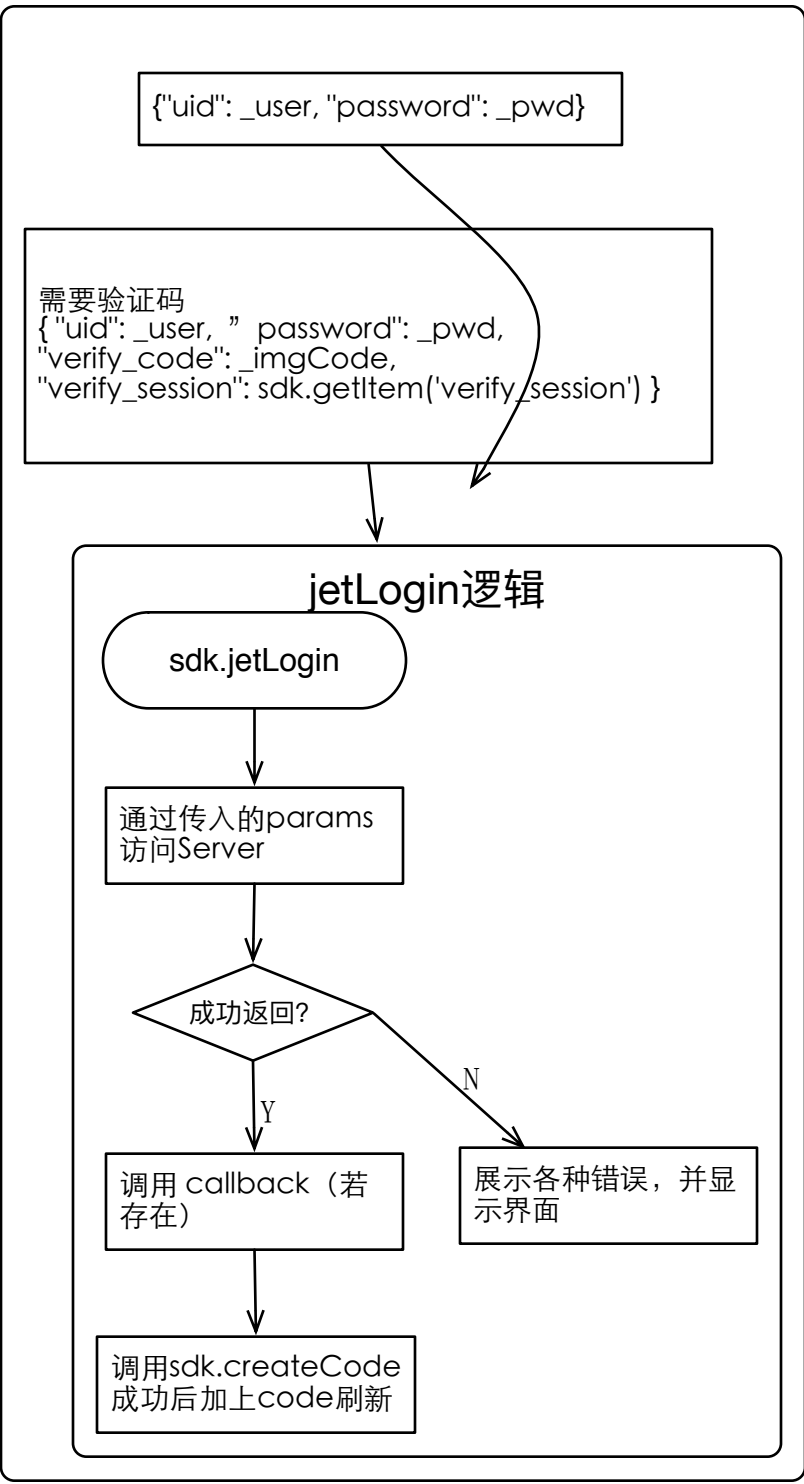
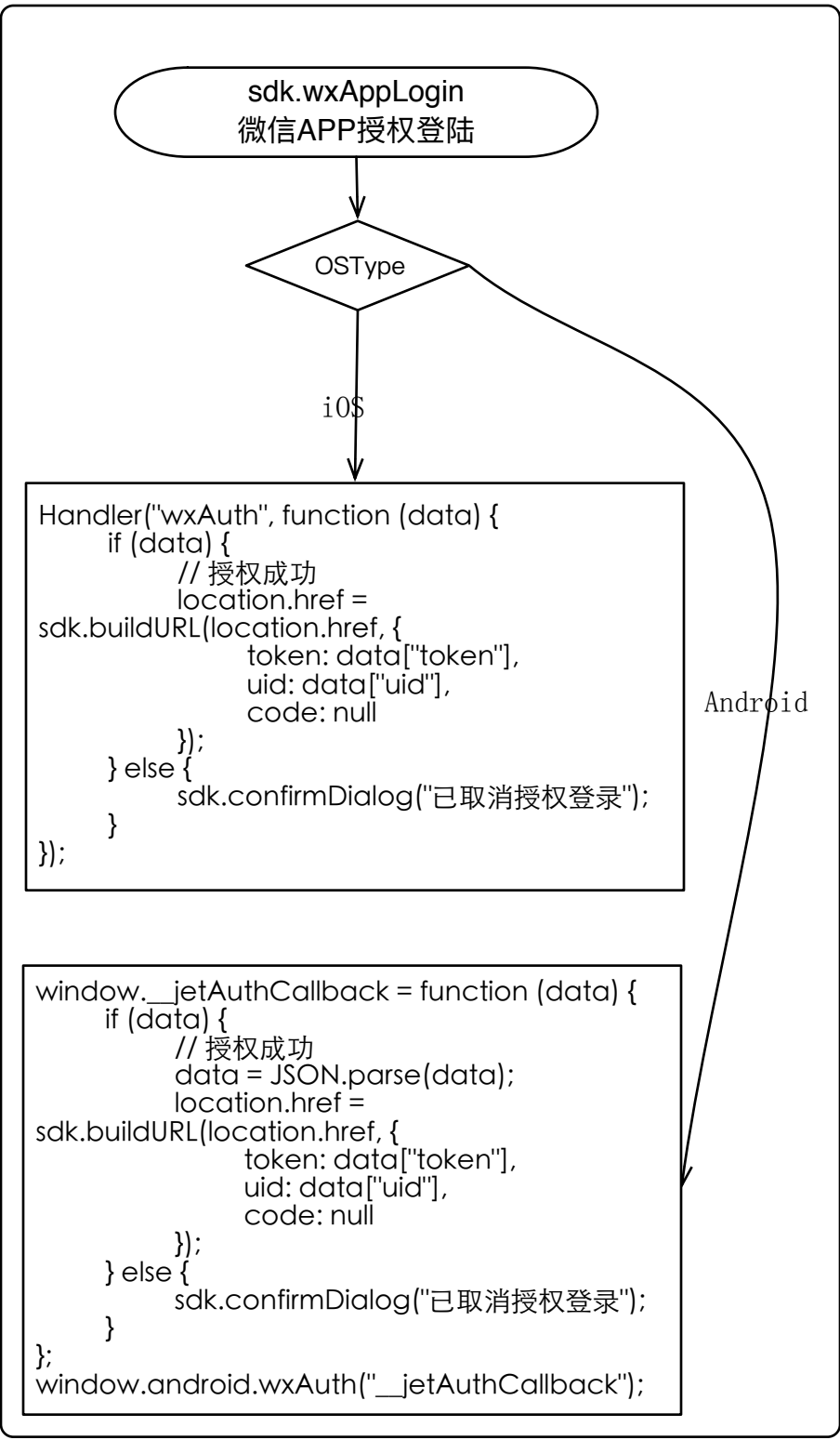
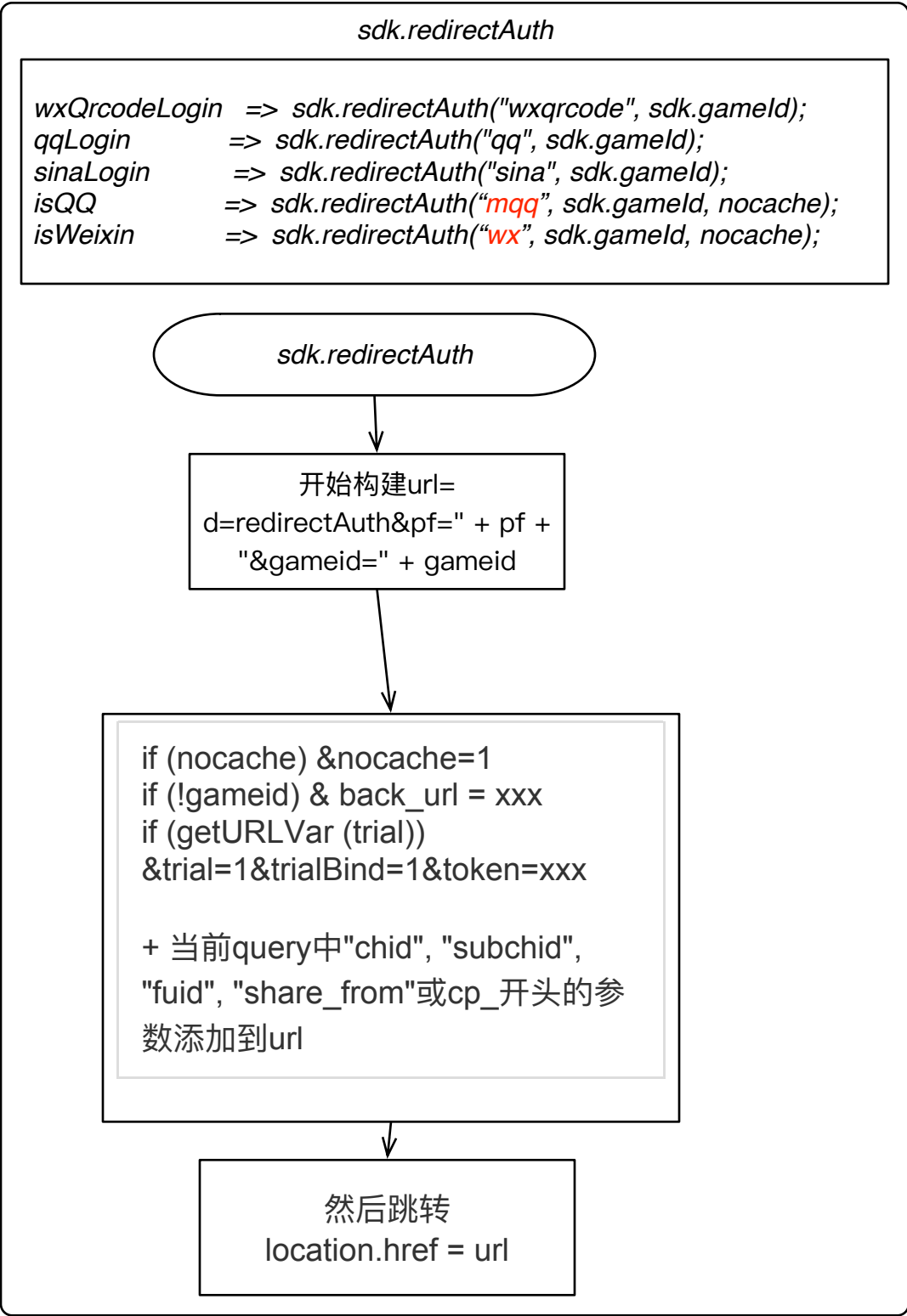
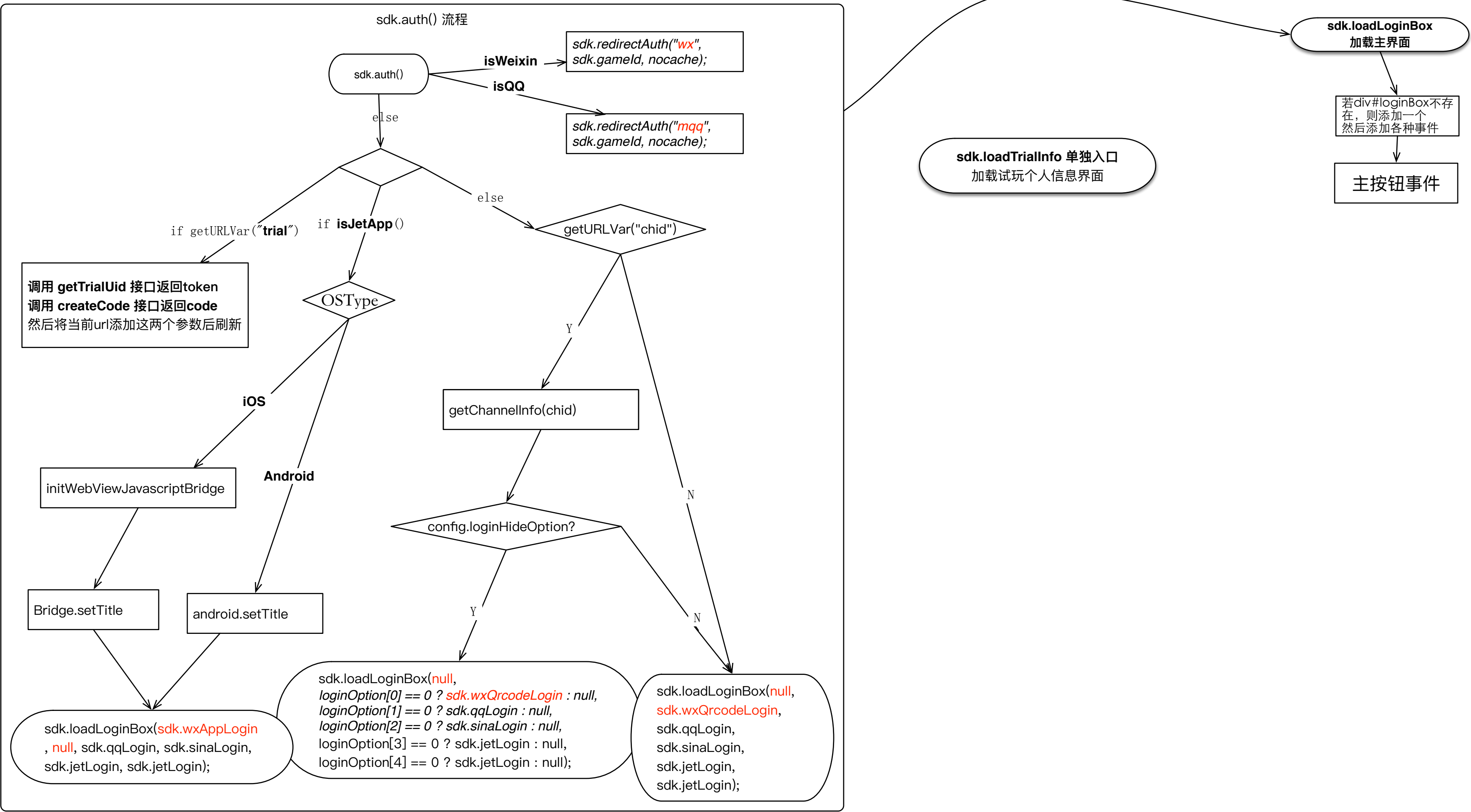
是 (设置Native的uid/token/title)

isiOS

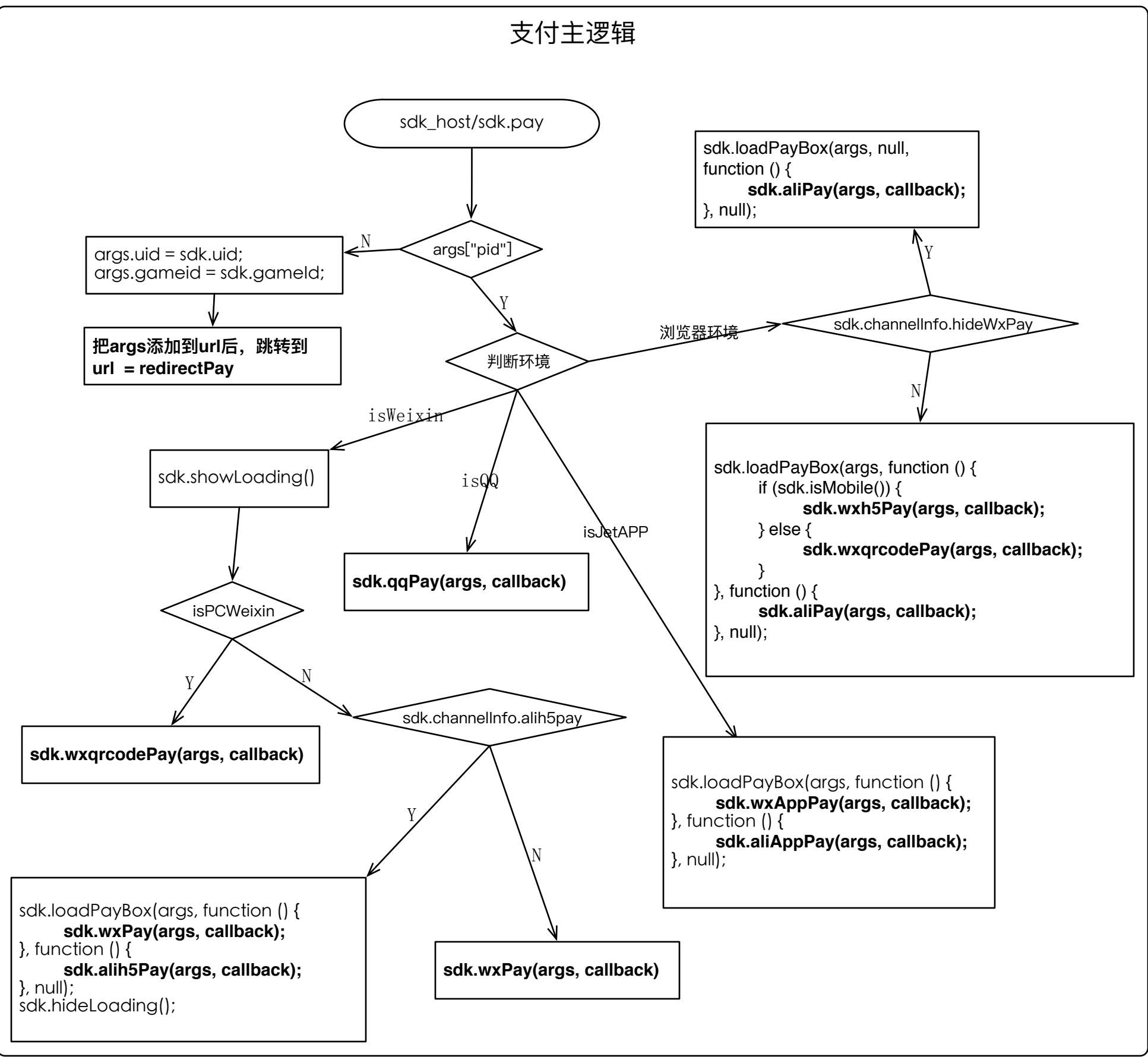
是
Bridge.callHandler("getUserAccount", function (data) {
 data = data || {};
 data.uid = sdk.uid;
 data.token = sdk.token;

 window.WebView.JavascriptBridge.callHandler("setUserAccount", data);
});
window.WebView.JavascriptBridge.callHandler("setTitle", document.title);

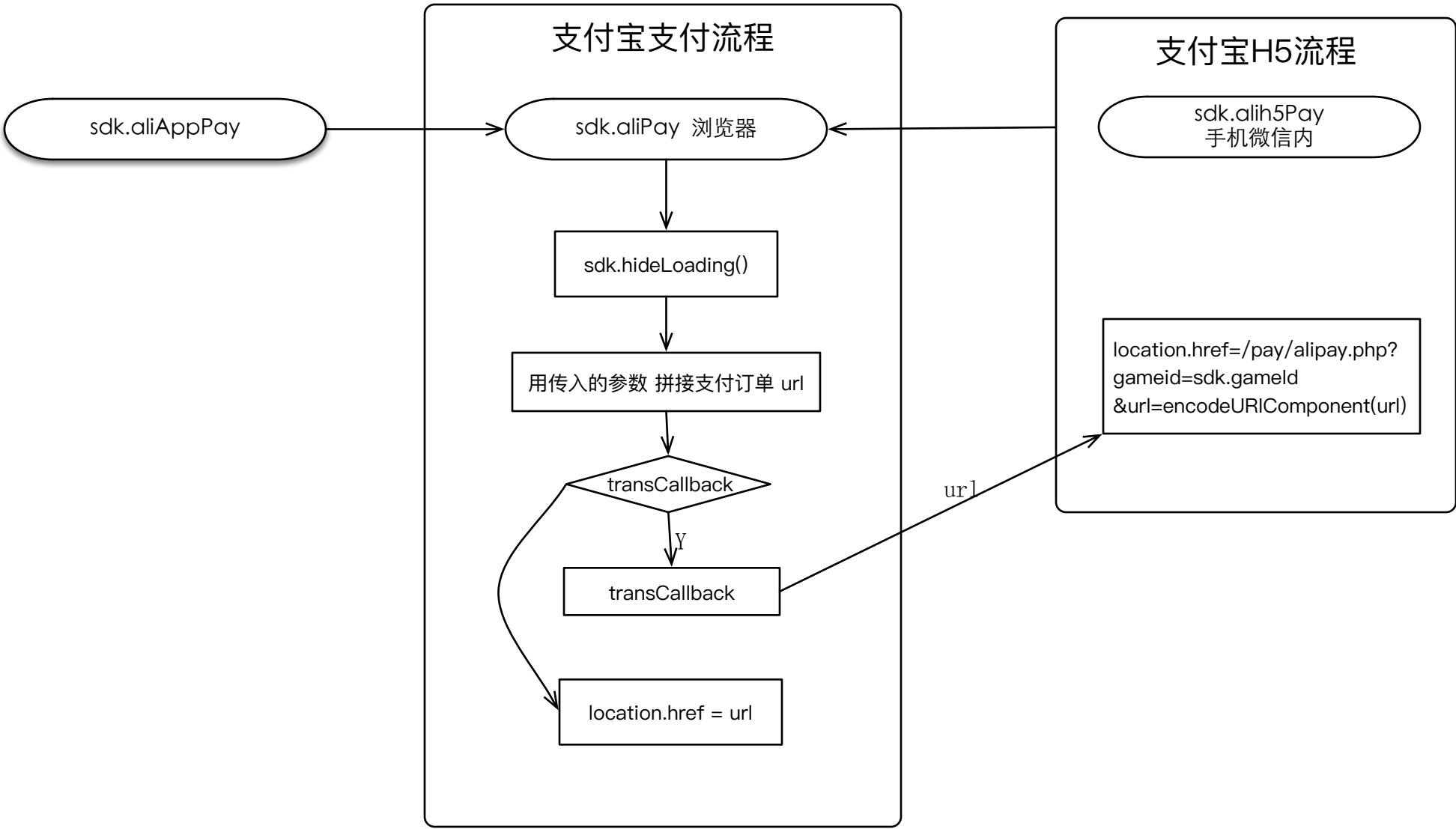
否
sdk.initAndroidAPPSHARE();
var data = window.android.getUserAccount();
data = data ? JSON.parse(data) : {};
data.uid = sdk.uid;
data.token = sdk.token;
window.android.setUserAccount(JSON.stringify(data));
window.android.setTitle(document.title);



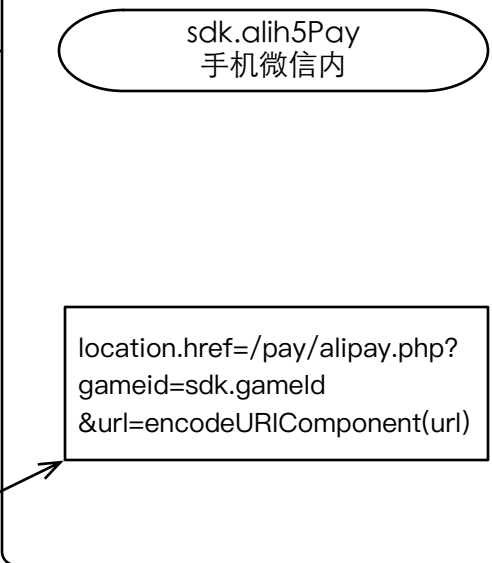
支付主逻辑



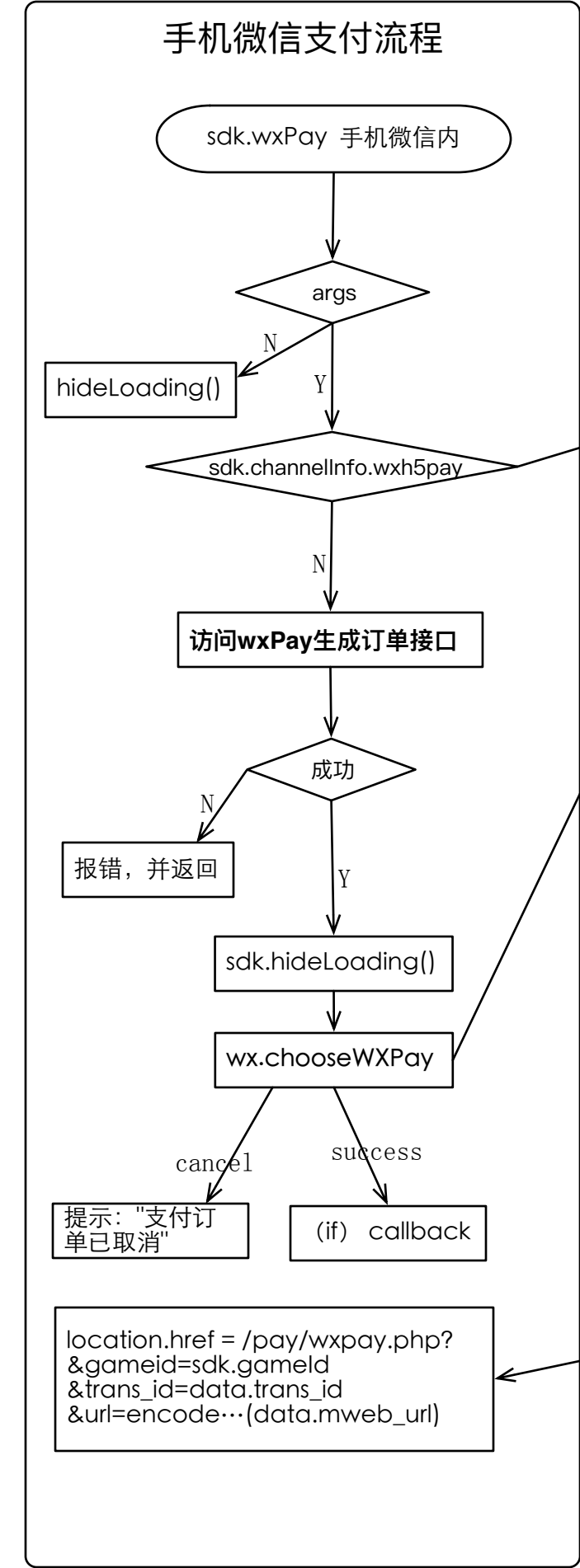
支付宝支付流程



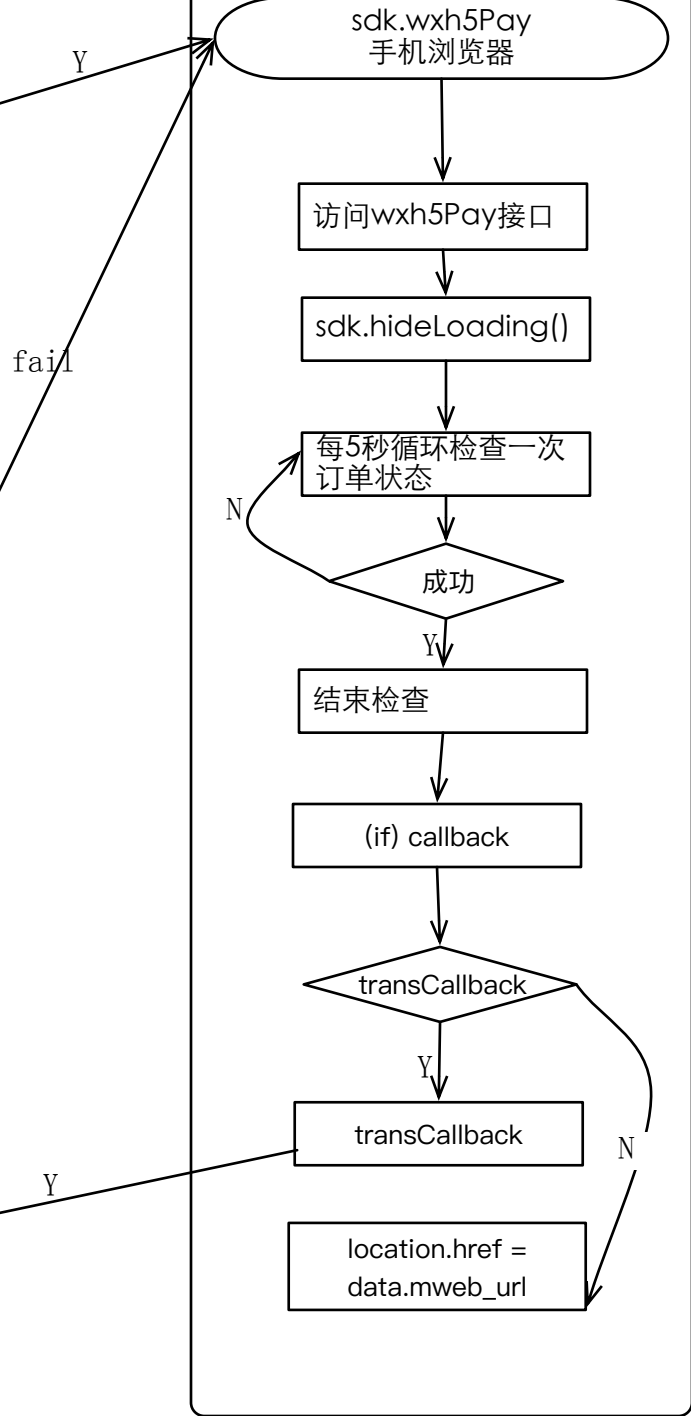
支付宝H5流程



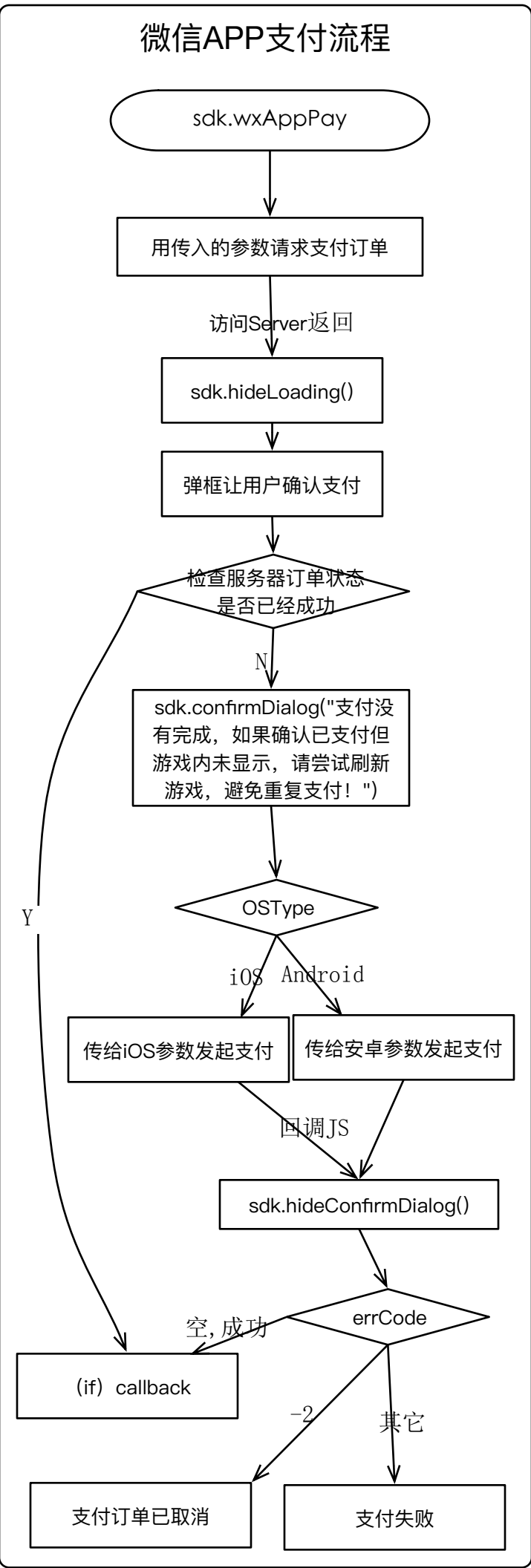
手机微信支付流程



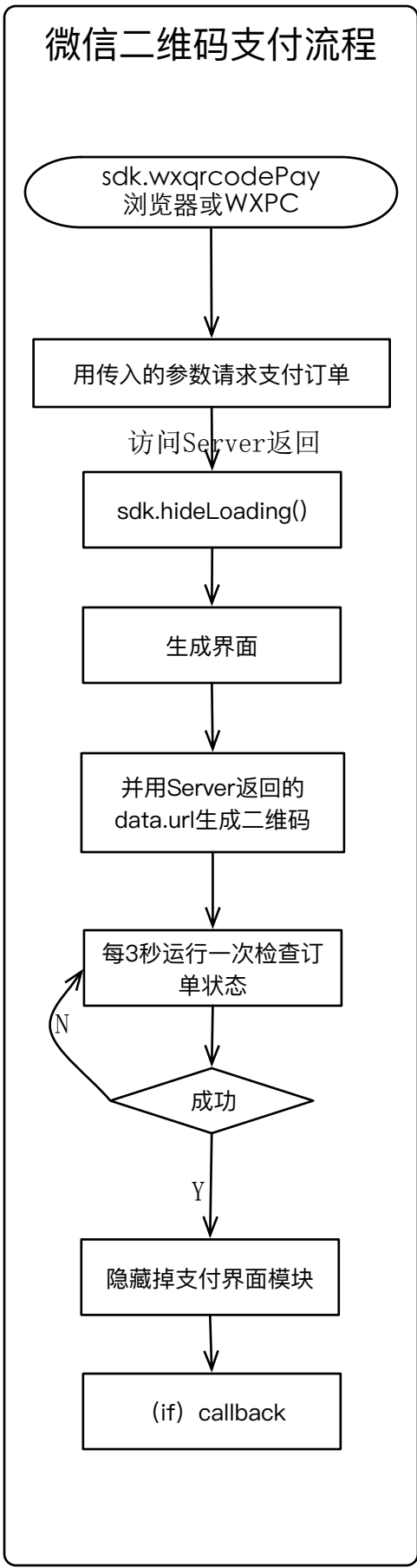
微信H5支付流程
手机外浏览器跳微信app
支付



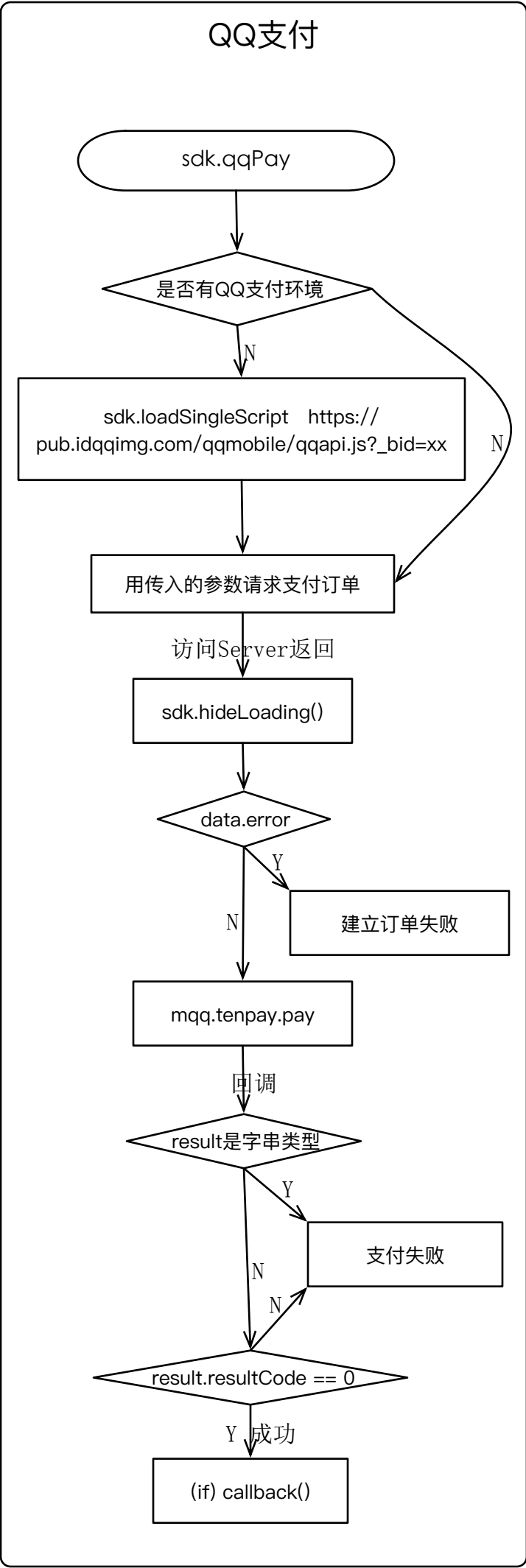
微信APP支付流程

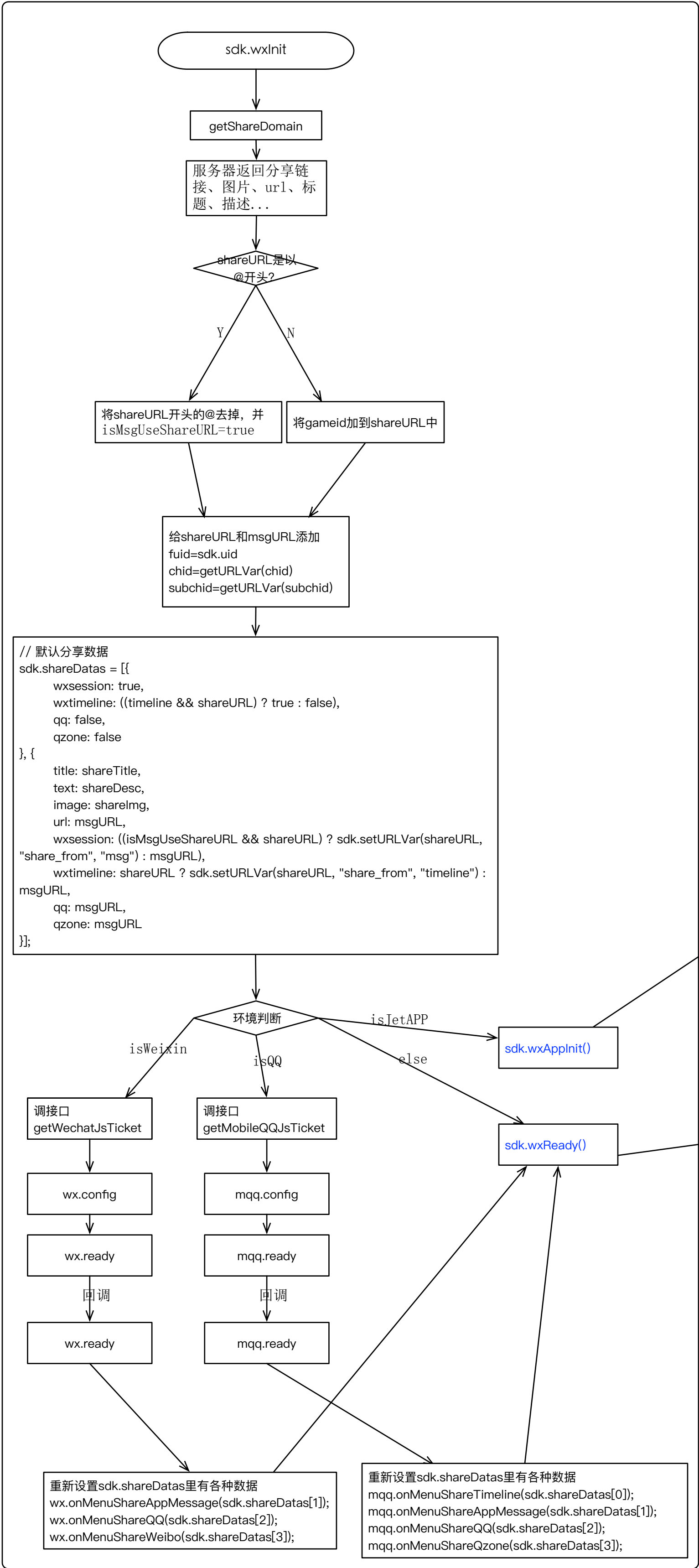


微信二维码支付流程



QQ支付



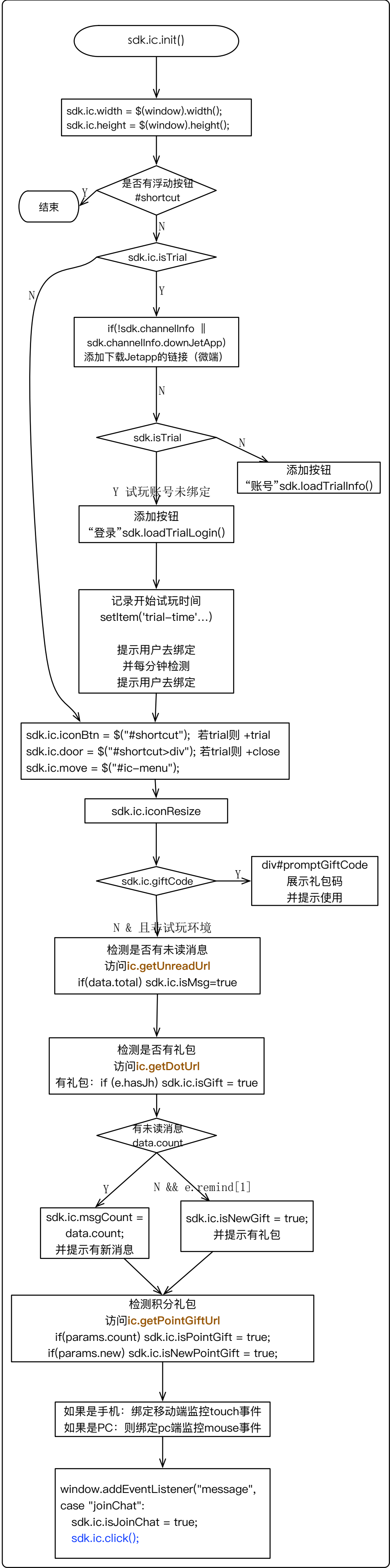


```
// 初始化APP分享环境
sdk.wxApplnit = function () {
  if (sdk.isiOS()) {
    ...
  } else if (sdk.isAndroid()) {
    window.android.setShare(JSON.stringify(sdk.shareDatas[0]));
    window.android.setShareContent(JSON.stringify(sdk.shareDatas[1]));
    sdk.wxReady();
  }
};
```

```
// 初始化完成
sdk.wxReady = function () {
  if (sdk.readyCallback) {
    sdk.readyCallback();
  }
  sdk.readyCallback = null;
  for (var i in sdk.readyMsgList) {
    sdk.execMessage(sdk.readyMsgList[i]);
  }
  sdk.readyMsgList = null;
};
```

```
// 开始侦听消息
sdk.execMessage = function (msg) {
  switch (msg.cmd) {
    case "showFocus": {
      sdk.showFocus(msg.args);
      break;
    }
    case "logout": {
      sdk.logout();
      break;
    }
    case "showFriend": {
      sdk.showFriend();
      break;
    }
    case "hideFriend": {
      sdk.hideFriend();
      break;
    }
    case "shareTitle": {
      if (sdk.readyCallback) {
        sdk.readyMsgList.push(msg);
      } else {
        sdk.shareTitle(msg.args);
      }
      break;
    }
    case "shareIcon": {
      if (sdk.readyCallback) {
        sdk.readyMsgList.push(msg);
      } else {
        sdk.shareIcon(msg.args);
      }
      break;
    }
    case "shareDesc": {
      if (sdk.readyCallback) {
        sdk.readyMsgList.push(msg);
      } else {
        sdk.shareDesc(msg.args);
      }
      break;
    }
    case "shareParams": {
      if (sdk.readyCallback) {
        sdk.readyMsgList.push(msg);
      } else {
        sdk.shareParams(msg.args);
      }
      break;
    }
    case "stat": {
      //sdk.execStat(msg.args["item"], msg.args["sub_item"], msg.args["sub_sec_item"]);
      break;
    }
    case "showTopImg": {
      sdk.showTopImg(msg.args);
      break;
    }
    case "copy": {
      sdk.copy(msg.args);
      break;
    }
    case "bindCash": {
      sdk.bindCash();
      break;
    }
    case "downJetApp": {
      sdk.downJetApp();
      break;
    }
    case "refresh": {
      window.location.reload();
      break;
    }
    case "chatPrevent": {
      sdk.getPreventInfo = msg.args;
      break;
    }
    case "saveGame": {
      sdk.saveGameDesktop();
      break;
    }
  }
}
};
```

sdk.ic.isTrial表示是否为试玩环境。从URL里获取
 sdk.isTrial表示是否已经绑定。是在鉴权接口里返回的。0是已绑定



在悬浮按钮touchEnd事件、joinChat事件
 中有调用

