



EC SDK

6.1.T2

Demo 指南(iOS)

文档版本 01

发布日期 2018-06-06

版权所有 © 华为技术有限公司 2018。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<http://e.huawei.com>

目 录

1 简介.....	1
2 快速体验.....	2
2.1 业务帐号申请.....	2
2.2 开发环境准备.....	2
2.3 SDK 导入.....	2
2.4 调试运行 Demo.....	3
3 Demo 代码框架.....	4
3.1 文件夹结构.....	5
3.2 逻辑框架.....	6
4 Demo 功能列表.....	8
4.1 配套 CloudEC 功能.....	11
5 如何裁剪.....	16
5.1 裁剪说明.....	16
5.2 示例一：裁剪“即时消息”	16
6 FAQ.....	27
7 修订记录.....	28

1 简介

“CloudEC_Client_API_Demo_iOS” 配套CloudEC 6.1解决方案，基于Terminal SDK，为第三方应用提供基础业务实现样例，开发者可以根据样例进行灵活的界面定制，开发自己的特色业务。

2 快速体验

2.1 业务帐号申请

使用华为 eSDK 远程实验室环境

请访问[远程实验室操作指导](#)，查阅远程实验室预约申请及接入使用方法，获取业务服务器地址和业务账号信息。

使用合作厂商自有测试或生产环境

联系环境管理员，获取业务服务器地址和业务账号信息。

2.2 开发环境准备

开发工具

- Xcode 8.0及以上版本。

SDK 软件包

- 访问<http://developer.huawei.com>，获取eSDK API iOS SDK。

eSDK EC API V1.1.RC1.SPC001 iOS



下载

Demo 源码

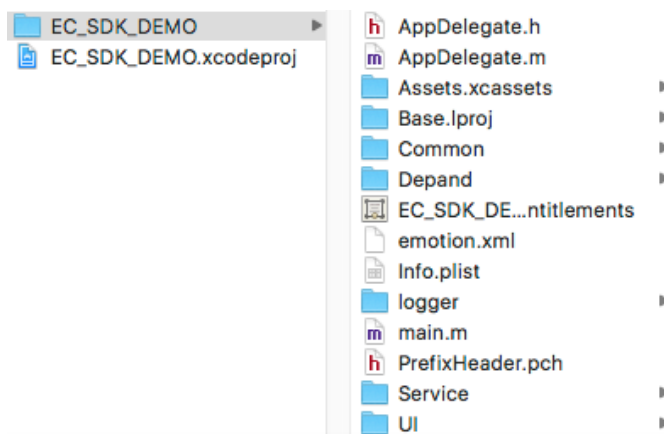
- 访问<https://github.com/Huawei>，获取“eSDK_CloudEC_Overall_Demo_iOS”。

2.3 SDK 导入

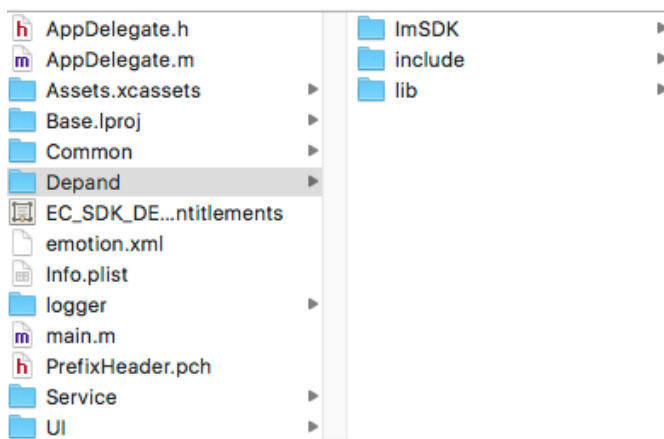
步骤1 将下载的eSDK API iOS SDK软件包解压缩到磁盘：



步骤2 将下载的Demo源代码解压后拷贝至工作目录：



步骤3 将“步骤1”解压后的软件包拷贝至“Depand”文件夹：



----结束

2.4 调试运行 Demo

步骤1 打开Xcode，导入2.3章节中处理好的Demo工程。

步骤2 Build。



一般情况下，构建过程无错误，若存在报错，请根据Xcode提示进行修正。

步骤3 连接iOS调试真机，进行调试。

步骤4 完成服务器地址、端口设置，输入登录帐号和密码，完成登录。

----结束

3 Demo 代码框架

3.1 文件夹结构

图 3-1 文件夹结构

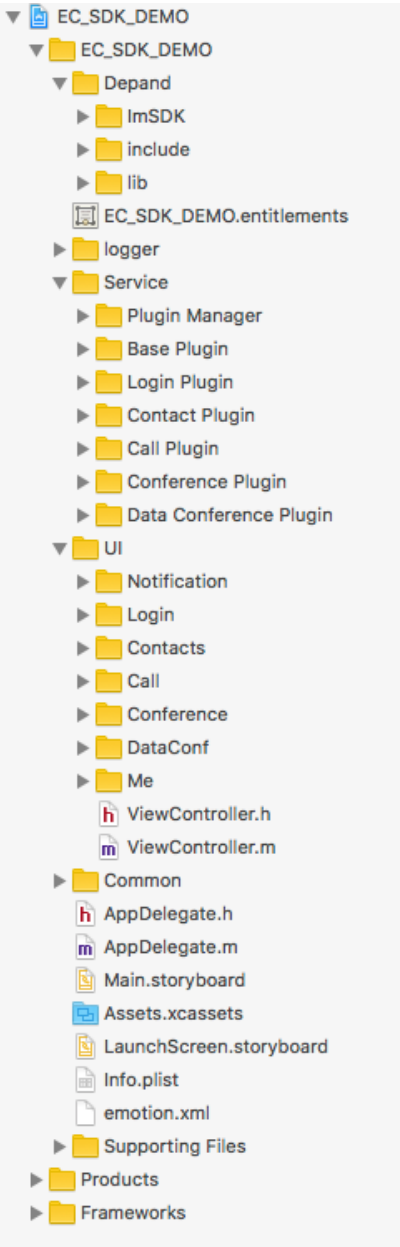
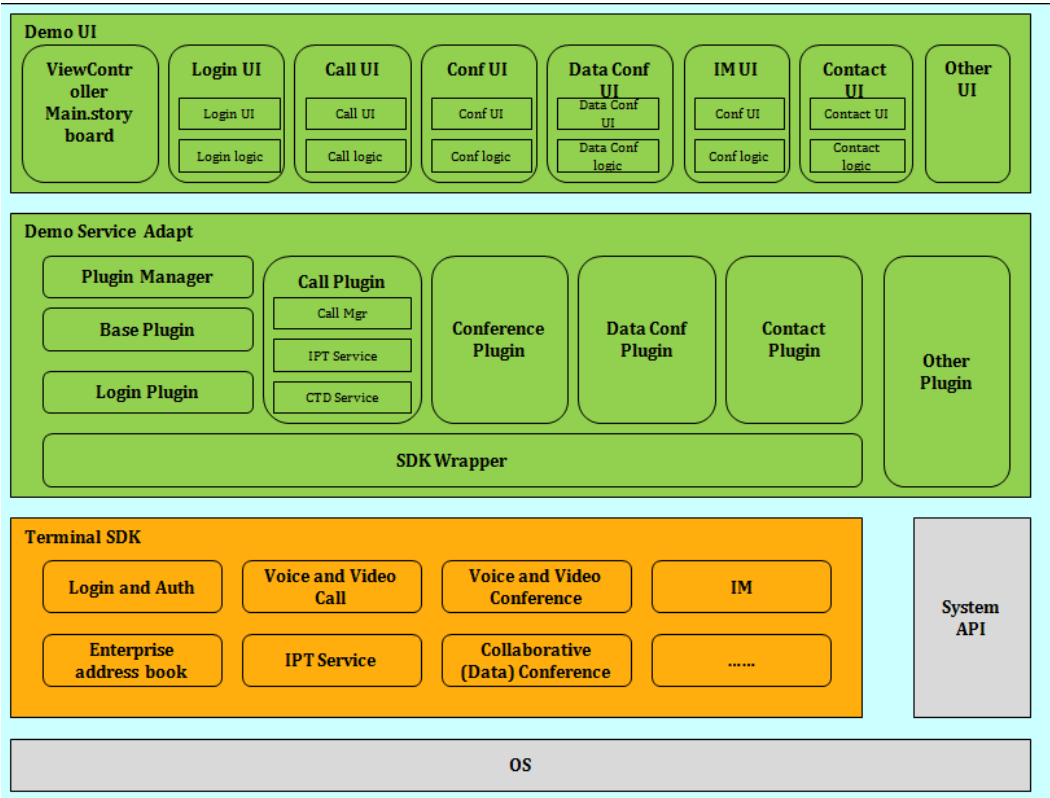


表 3-1 文件夹说明

文件夹	说明
EC_SDK_DEMO\Depand	ImSDK,Tup库，Tup头文件存放路径
EC_SDK_DEMO\logger	日志打印文件存放路径

文件夹	说明
EC_SDK_DEMO\Service\Plugin Manager	“服务管理”工程库文件夹，包含Demo对具体业务的配置(是否支持某业务)、启动和停止等服务管理的实现代码。
EC_SDK_DEMO\Service\Base Plugin	“基础服务”工程文件夹，包含Demo的各个模块一些基础消息分发进行管理的实现代码。
EC_SDK_DEMO\Service>Login Plugin	“登录服务”工程库文件夹，包含基于SDK实现Demo所有业务逻辑总体登录入口的实现代码。
EC_SDK_DEMO\Service>Contact Plugin	“联系人服务”工程库文件夹，包含基于SDK实现企业联系人(地址本)的业务代码
EC_SDK_DEMO\Service\Call Plugin	“音视频呼叫服务”工程库文件夹，包含基于SDK实现音视频呼叫、IPT业务和CTD的业务代码。
EC_SDK_DEMO\Service\Conference Plugin	“视频会议服务”工程库文件夹，包含基于SDK实现视频会议的业务代码。
EC_SDK_DEMO\Service\Data Conference Plugin	“数据会议服务”工程库文件夹，包含基于SDK实现数据会议的业务代码。
EC_SDK_DEMO\UI	各个服务模块界面实现代码，包含IM、登陆、联系人、呼叫、会议、数据会议、设置等界面代码。
EC_SDK_DEMO\Common	“公共服务”工程库文件夹，包含其他业务依赖的公共功能实现代码。
EC_SDK_DEMO\Main.storyboard	主界面的xib实现。

3.2 逻辑框架



4 Demo 功能列表

功能项	功能子项	是否开放	Demo是否支持
统一鉴权与登录	普通登录	Y	Y
	第三方鉴权登录	Y	N
	注销	Y	Y
音频呼叫	发起音频呼叫	Y	Y
	挂断/取消/拒绝呼叫	Y	Y
	接听音频呼叫	Y	Y
	设置(或取消)麦克风闭音	Y	Y
	发送DTMF信号	Y	Y
视频呼叫	发起视频呼叫	Y	Y
	挂断/取消/拒绝呼叫	Y	Y
	接听视频呼叫	Y	Y
	发起音频转视频	Y	Y
	发起视频转音频	Y	Y
	同意或拒绝音频转视频	Y	Y
	打开本地视频预览	Y	N
	关闭本地视频预览	Y	N
	开启或关闭本地摄像头	Y	Y
设备管理	切换麦克风	Y	N
	切换扬声器	Y	Y
	切换摄像头	Y	Y
	设置通话音量	Y	N

保持&恢复	保持和恢复音频通话	Y	Y
	保持和恢复视频通话	Y	Y
偏转	呼叫偏转	Y	Y
转移	呼叫盲转	Y	Y
登记IPT业务	登记或取消登记呼叫等待	Y	Y
	登记或取消登记DND	Y	Y
	登记或取消登记前转	Y	Y
CTD	发起CTD通话	Y	Y
会议管理	预约会议	Y	Y
	查询会议列表	Y	Y
	查询会议详情	Y	Y
基础会控	创建即时会议	Y	Y
	获取会议信息	Y	Y
	邀请与会者	Y	Y
	删除与会者	Y	Y
	结束会议	Y	Y
	离开会议	Y	Y
	闭音会场	Y	Y
	闭音与会者	Y	Y
	锁定会议	N	NA
	释放主席	N	NA
	申请主席	N	NA
	延长会议	N	NA
	设置或取消举手	N	NA
	显示发言人	Y	N
	会议列表一键入会	Y	Y
	会议接入码入会	Y	N
	IVR入会	Y	Y
渐进式会议	两转三会议	Y	N
	升级会议	Y	Y
基本控制	加入数据会议	Y	Y

	离开数据会议	Y	Y
	结束数据会议	Y	Y
视频	基础视频控制	Y	Y
	选看视频	Y	Y
	设置视频参数	Y	Y
屏幕共享	观看桌面共享	Y	Y
程序共享	观看应用程序共享	Y	Y
文档	观看文档共享	Y	Y
白板	观看白板共享	Y	Y
标注	观看标注	Y	Y
聊天	发送公共即时消息	Y	N
个人信息获取和设置	获取和设置用户信息	Y	Y
	获取和设置个人头像	Y	Y
个人通讯录	获取好友和分组列表列表	Y	Y
	获取头像	Y	Y
	添加好友	Y	Y
	删除联系人或好友	Y	Y
企业通讯录	查询企业通讯录	Y	Y
	查询用户信息	Y	Y
状态与提醒	设置个人状态	Y	Y
	订阅用户状态	Y	Y
固定群	解散固定群	Y	Y
	修改固定群	Y	Y
	审批固定群邀请	Y	Y
	退出固定群	Y	Y
	获取固定群列表	Y	Y
	获取固定群成员列表	Y	Y
	获取固定群详情	Y	Y
讨论组	新建讨论组	Y	Y
	固化讨论组	Y	Y
	修改讨论组	Y	Y

	邀请讨论组成员	Y	Y
	删除讨论组成员	Y	Y
	退出讨论组	Y	Y
	获取讨论组列表	Y	Y
	获取讨论组成员列表	Y	Y
	获取讨论组详情	Y	Y
文本消息	发送即时消息	Y	Y
	获取漫游消息	Y	Y
	设置消息已读	Y	Y
	消息通知	Y	Y
Push	开启Push	Y	N
	关闭Push	Y	N
	Push消息处理	Y	N
富媒体	发送和接收图片	Y	Y
	发送和接收语音	Y	Y
	发送和接收视频	Y	Y
	发送和接收普通离线文件	Y	Y
网络地址本	查询企业联系人	Y	Y
	查询部门	Y	N
	获取和设置个人头像	Y	Y

4.1 配套 CloudEC 功能

特性	功能项	功能子项	是否开放 (Hosted)	Demo是否支持	Demo说明
Login	统一鉴权与登录	普通登录	Y	Y	暂不支持 STG
		第三方鉴权登录	Y	N	无配套服务器环境，优先级暂放低
		注销	Y	Y	暂不支持 STG

VoIP	音频呼叫	发起音频呼叫	Y	Y	
		挂断/取消/拒绝呼叫	Y	Y	
		接听音频呼叫	Y	Y	
		设置(或取消)麦克风闭音	Y	Y	
		发送DTMF信号	Y	Y	
	视频呼叫	发起视频呼叫	Y	Y	
		挂断/取消/拒绝呼叫	Y	Y	
		接听视频呼叫	Y	Y	
		发起音频转视频	Y	Y	
		发起视频转音频	Y	Y	
		同意或拒绝音频转视频	Y	Y	
		打开本地视频预览	Y	Y	视频预览显示方向无法旋转
		关闭本地视频预览	Y	Y	不支持关闭打开，支持切换前后摄像头
		开启或关闭本地摄像头	Y	Y	
	设备管理	切换麦克风	Y	N	使用系统设置
		切换扬声器	Y	Y	
		切换摄像头	Y	Y	
		设置通话音量	Y	N	使用系统设置
IPT	保持&恢复	保持和恢复音频通话	Y	Y	
		保持和恢复视频通话	Y	Y	
	偏转	呼叫偏转	Y	Y	
	转移	呼叫盲转	Y	Y	

	登记IPT业务	登记或取消登记呼叫等待	Y	Y	
		登记或取消登记DND	Y	Y	
		登记或取消登记前转	Y	Y	
	CTD	发起CTD通话	Y	Y	ctd不支持视频通话
ConfCtrl	会议管理	预约会议	Y	Y	
		查询会议列表	Y	Y	
		查询会议详情	Y	Y	
	基础会控	创建即时会议	Y	Y	
		获取会议信息	Y	Y	
		邀请与会者	Y	Y	
		删除与会者	Y	Y	
		挂断与会者	Y	Y	
		重拨与会者	Y	Y	
		结束会议	Y	Y	
		离开会议	Y	Y	
		闭音会场	Y	Y	
		闭音与会者	Y	Y	
		锁定会议	Y	Y	
		释放主席	Y	Y	
		申请主席	Y	Y	
		显示发言人	Y	Y	
		延长会议	Y	N	会议自动延长
		设置或取消举手	Y	Y	
		会议列表一键入会	Y	Y	
		会议接入码入会	Y	Y	
		IVR入会	Y	Y	
	视频会控	设置会议模式	Y	Y	
		广播会场	Y	Y	

		选看会场	Y	Y	
	渐进式会议	两转三会议	Y	Y	
		升级会议	Y	Y	
		升级会议	Y	Y	
	辅流共享	主动共享辅流	Y	Y	
		观看辅流	Y	Y	
DataConf	基本控制	加入数据会议	Y	Y	
		离开数据会议	Y	Y	
		结束数据会议	Y	Y	
	屏幕共享	观看桌面共享	Y	Y	
	程序共享	观看应用程序共享	Y	N	
	文档	观看文档共享	Y	N	
	白板	观看白板共享	Y	N	
	标注	观看标注	Y	N	
	聊天	发送公共即时消息	Y	Y	需要同时在聊天界面才能收到消息
IM	个人信息获取和设置	获取和设置用户信息	Y	Y	
		获取和设置个人头像	Y	Y	
	个人通讯录	获取好友和分组列表列表	Y	Y	
		获取头像	Y	Y	
		添加好友	Y	Y	
		删除联系人或好友	Y	Y	
	企业通讯录	查询企业通讯录	Y	Y	
		查询用户信息	Y	Y	
	状态与提醒	设置个人状态	Y	Y	
		订阅用户状态	Y	Y	
	固定群	解散固定群	Y	Y	
		修改固定群	Y	Y	
		审批固定群邀请	Y	Y	

		退出固定群	Y	Y	
		获取固定群列表	Y	Y	
		获取固定群成员列表	Y	Y	
		获取固定群详情	Y	Y	
	讨论组	新建讨论组	Y	Y	
		固化讨论组	Y	Y	
		修改讨论组	Y	Y	
		邀请讨论组成员	Y	Y	
		删除讨论组成员	Y	Y	
		退出讨论组	Y	Y	
		获取讨论组列表	Y	Y	
		获取讨论组成员列表	Y	Y	
		获取讨论组详情	Y	Y	
	文本消息	发送即时消息	Y	Y	
		获取漫游消息	Y	Y	
		设置消息已读	Y	Y	
		消息通知	Y	Y	
	Push	开启Push	Y	N	
		关闭Push	Y	N	
		Push消息处理	Y	N	
	富媒体	发送和接收图片	Y	Y	
		发送和接收语音	Y	Y	
		发送和接收视频	Y	Y	
		发送和接收普通离线文件	Y	Y	
eAddr	网络地址本	查询企业联系人	Y	Y	
		查询部门	Y	Y	
		获取和设置个人头像	Y	Y	

5 如何裁剪

5.1 裁剪说明

EC SDK Demo(iOS)基于功能特性进行模块划分，可以根据开发者的实际需要，在修改少量代码的情况少，对不需要的功能进行裁剪。

5.2 示例一：裁剪“即时消息”



说明

无特殊说明，文中红框内容均表示需删除内容。

步骤1 删除无需使用的工程、包、文件和文件目录

图 5-1

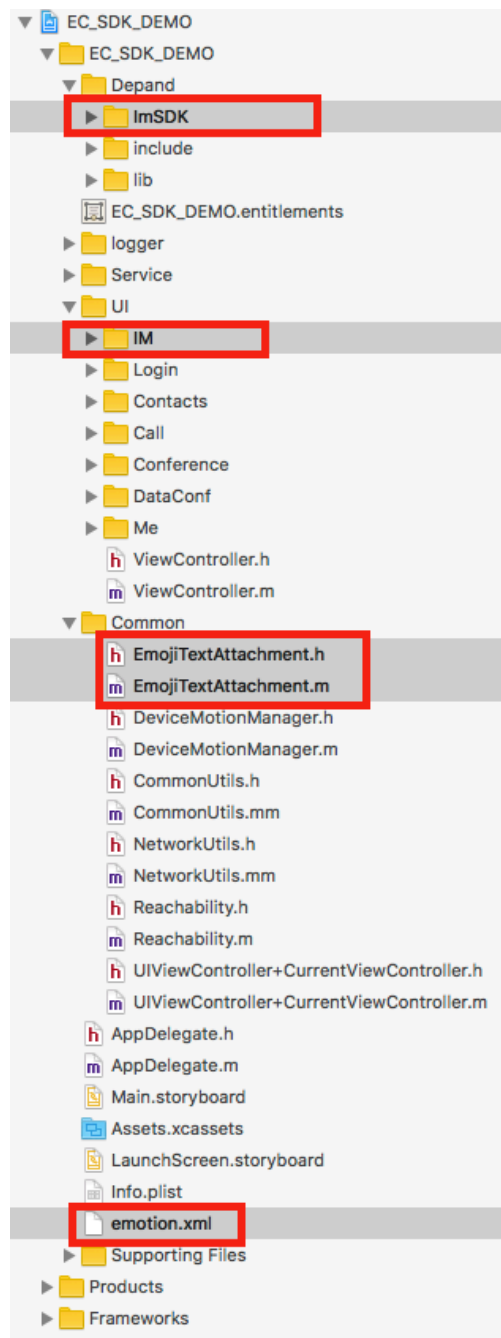
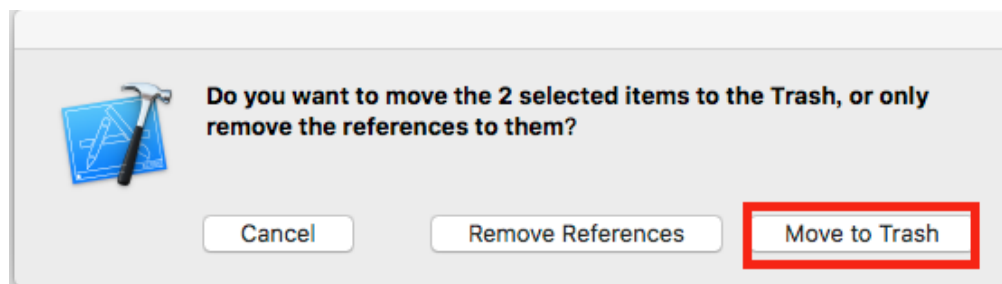
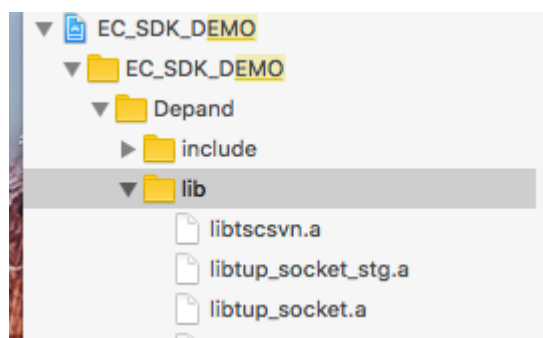


图1中圈出部分，右键Delete，选择彻底删除（弹出框后点击图2所圈出部分）

图 5-2



步骤2 添加相关库文件



右键lib，点击Add File to “xxxx”，选择如下库文件后，点击Add。

libtscsvn.a ,libtup_socket_stg.a ,libtup_socket.a ,libtupService.a ,libuspsdk.a

步骤3 从Main.storyboard删除无需使用的界面及相关入口

图 5-3

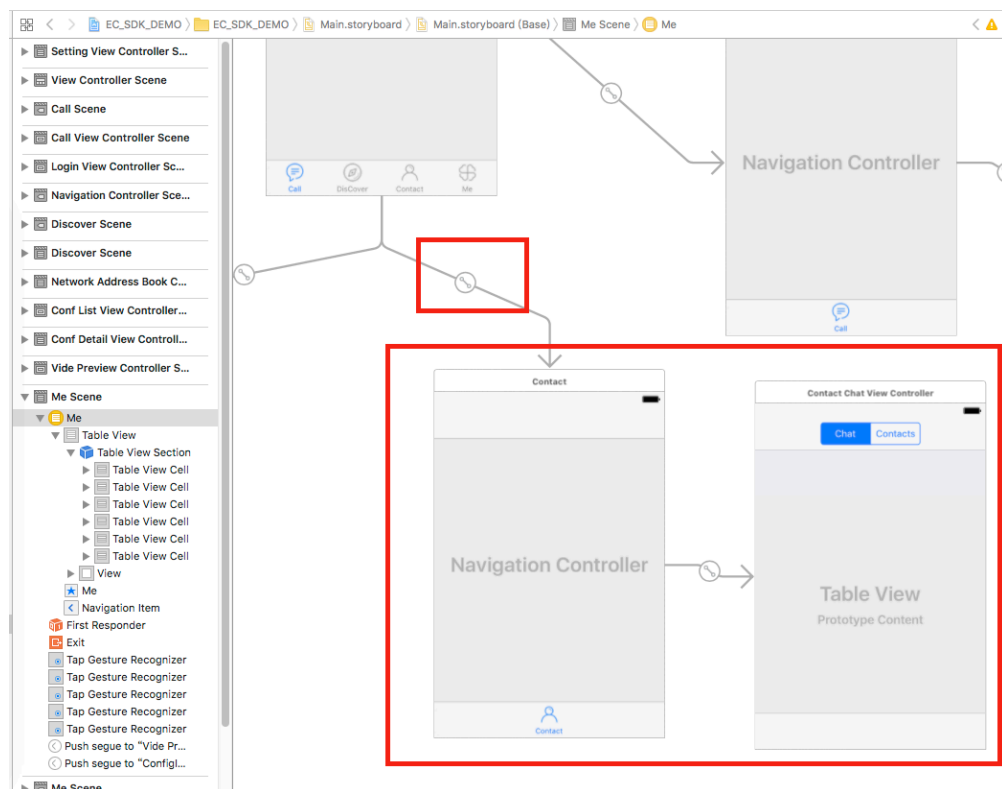


图3中圈出部分的连接和界面删除。

图 5-4

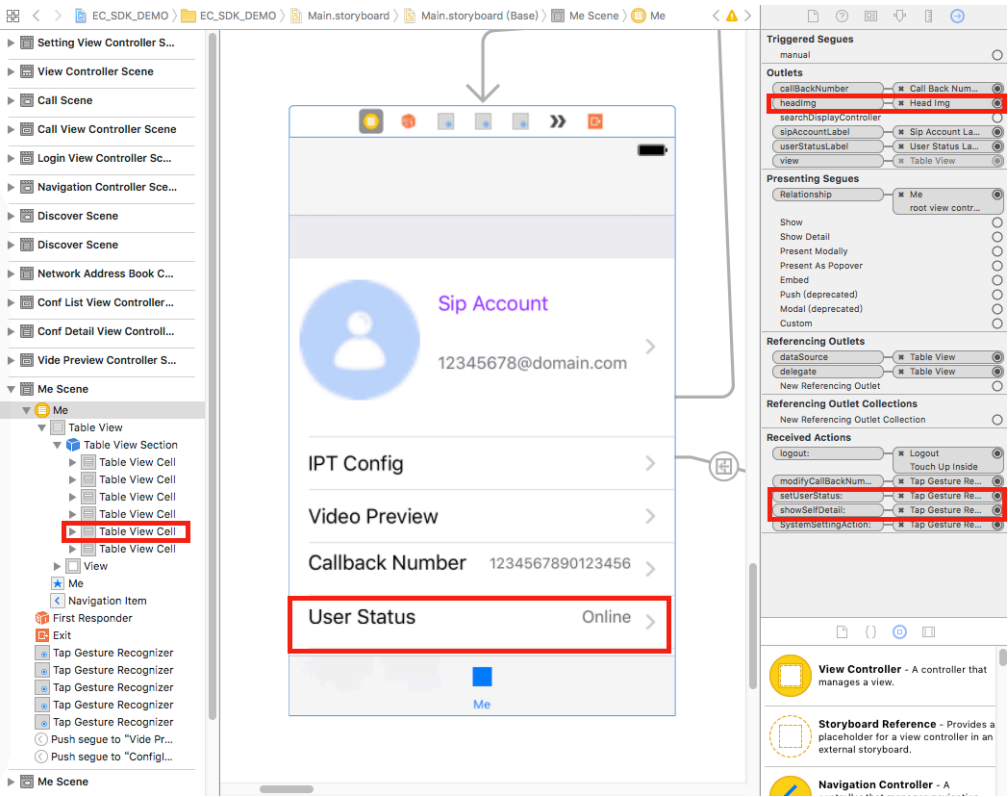


图4中先选中ME控制器，圈出部分点击中间x删除,左侧圈出的table view cell（看下是否和中间圈出部分对应）。

图 5-5

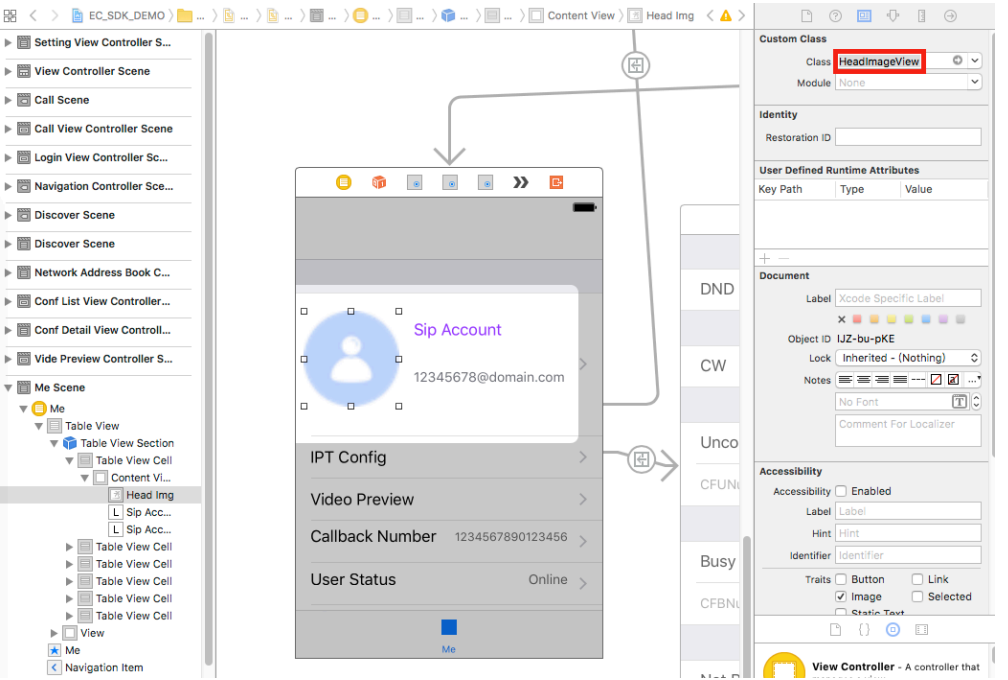


图5中先选择对的头像视图，删除圈出部分，然后回车。

步骤4 删除AppDelegate.m中无需使用代码。

图 5-6

```
#import "AppDelegate.h"
#import "NetworkUtils.h"
#import <TUPIOSSDK/TUPIOSSDK.h>
#import "ServiceManager.h"
#import "CallWindowController.h"
#import "ConfRunningViewController.h"
#import "ECSLogger.h"
#import "LocalNotificationCenter.h"
#import "ConfStatus.h"

@interface AppDelegate ()

@end
```

图 5-7

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)
launchOptions {

    [[UINavigationController appearance] setBarTintColor:[UIColor colorWithRed:44.0/255 green:110.0/255 blue:
    232.0/255 alpha:1]];
    [[UINavigationController appearance] setTintColor:[UIColor whiteColor]];
    [[UINavigationController appearance] setTitleTextAttributes:@{NSForegroundColorAttributeName : [UIColor
    whiteColor]}];
    // Override point for customization after application launch.

    // DDLog init (write log to file)
    [[ECSLogger sharedInstance] addFileLogger];
    [NetworkUtils sharedInstance];

    // Config TUPIOSSDK log path
    NSString *logPath = [[NSHomeDirectory() stringByAppendingPathComponent:@"Documents"]
    stringByAppendingString:@"TUPOG60log"];
    [ECSUserDefaultsHelper sharedInstance].logFileSuperPath = logPath;

    // init TUPIOSSDK (module: Contact/IM/Group/MAALogin)
    [ECSAppConfig sharedInstance].appLogLevel = kECSLogDebug;
    [ECSAppConfig sharedInstance].isLogEnabled = YES;
    [ECSAppConfig sharedInstance].version = @"V2.0.4.5";
    if ([ECSAppConfig sharedInstance].isFirstUsed)
    {
        [[ECSAppConfig sharedInstance] initializeSecurityRandomKey];
    }
    [TUPIOSSDKService start];

    // start up tup module.(module: Call/Conference/SipLogin)
    [ServiceManager startup];

    // other UI Things
    [CallWindowController sharedInstance];
    [[LocalNotificationCenter sharedInstance] start];

    return YES;
}
```

图 5-8

```
- (void)applicationWillResignActive:(UIApplication *)application {
    // Sent when the application is about to move from active to inactive state. This can occur for
    // certain types of temporary interruptions (such as an incoming phone call or SMS message) or
    // when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering
    // callbacks. Games should use this method to pause the game.

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}

- (void)applicationDidEnterBackground:(UIApplication *)application {
    // Use this method to release shared resources, save user data, invalidate timers, and store
    // enough application state information to restore your application to its current state in case
    // it is terminated later.
    // If your application supports background execution, this method is called instead of
    // applicationWillTerminate: when the user quits.

    __block UIBackgroundTaskIdentifier taskID = [[UIApplication sharedApplication]
        beginBackgroundTaskWithExpirationHandler:^(
            [[UIApplication sharedApplication] endBackgroundTask:taskID];
            taskID = UIBackgroundTaskInvalid;
        )];

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}
```

图 5-9

```
- (void)applicationWillTerminate:(UIApplication *)application {
    // Called when the application is about to terminate. Save data if appropriate. See also
    // applicationDidEnterBackground:.

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}

- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:
    (NSData *)deviceToken {
    // Config APNS type and device token to MAA for APNS Notification.
    const unsigned *tokenBytes = (const unsigned *) [deviceToken bytes];
    NSString *hexToken = [NSString stringWithFormat:@"%08x%08x%08x%08x%08x%08x%08x%08x",
        ntohs(tokenBytes[0]), ntohs(tokenBytes[1]), ntohs(tokenBytes[2]),
        ntohs(tokenBytes[3]), ntohs(tokenBytes[4]), ntohs(tokenBytes[5]),
        ntohs(tokenBytes[6]), ntohs(tokenBytes[7])];

    [ECSAppConfig sharedInstance].deviceToken = hexToken;
    [ECSAppConfig sharedInstance].apnsType = APNS_DEV;
}
```

删除图6、7、8、9中圈出部分代码。

步骤5 删除CommonUtils.mm中无需使用代码。

图 5-10

```
// CommonUtils.m
// EC_SDK_DEMO
//
// Created by EC Open support team
// Copyright(C), 2017, Huawei Tech
//

#import "CommonUtils.h"
#include <netdb.h>
#include <net/if.h>
#include <ifaddrs.h>
#include <arpa/inet.h>
#include <dlfcn.h>
#include <sys/sysctl.h>
#include <TUPIOSSDK/TUPIOSSDK.h>
```


图 5-11

```
/**
 *This method is used to get Image from disk
 *从图片文件夹获取图片
 *@param imageFile image file path
 *@return image
 */
+ (UIImage *) attachImageFile:(NSString*) imageFile {
    UIImage* image = [[ESpaceImageCache sharedInstance] imageWithKey:imageFile];
    if (image) {
        return image;
    } else {
        NSData* imageData = [[NSFileManager defaultManager] contentsAtPath:imageFile];
        UIImage* image = nil;
        if (imageData) {
            image = [UIImage imageWithData:imageData];
            return image;
        }
    }
    return nil;
}
```

删除图10、11中无需使用代码。

步骤6 删除LoginService.mm中无需使用代码。

图 5-12

```
1 #import "LoginService.h"
2 #import "LoginServerInfo+uportalInfo.h"
3 #include "string.h"
4 #include <netdb.h>
5 #include <net/if.h>
6 #include <ifaddrs.h>
7 #include <arpa/inet.h>
8 #include <dlfcn.h>
9 #include <sys/sysctl.h>
10 #import "login_interface.h"
11 #import "LoginInfo.h"
12 #import "Initializer.h"
13 #import "login_def.h"
14 #import "LoginCenter.h"
15 #import "CommonUtils.h"
16 #import "ManagerService.h"
17 #import <TUPIOSSDK/TUPMAALoginService.h>
18 #import <TUPIOSSDK/ECSAppConfig.h>
19 #import <TUPIOSSDK/eSpaceDBService.h>
20 #import <TUPNetworkSDK/ECSSocketController.h>
21 #import "call_interface.h"
22
23 #define NEEDMAALOGIN 1 // 是否需要MAA登陆
24 @interface LoginService()
```

删除图12中圈出部分代码，并修改宏定义NEEDMAALOGIN为0。

步骤7 删除MeViewController.m中无需使用代码。

图 5-13

```
#import "MeViewController.h"
#import "ManagerService.h"
#import "CommonUtils.h"
#import "HeadImageView.h"

#import <TUPIOSSDK/eSpaceDBService.h>
#import <TUPContactSDK/TupContactService.h>
#import <TUPIOSSDK/TUPUserService.h>

#import <TUPIOSSDK/eSpaceDBService.h>
#import "PersonDetailViewController.h"

#define NEEDREGISTERMAALOGOUT 1 //是否需要MAA注销
@interface MeViewController ()
@property(weak, nonatomic) IBOutlet UILabel *sipAccountLabel;
@property(weak, nonatomic) IBOutlet UILabel *callBackNumber;
@property(weak, nonatomic) IBOutlet UILabel *userStatusLabel;
@property(weak, nonatomic) IBOutlet HeadImageView *headImg;
@end

@implementation MeViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    sipAccountLabel.text = [ManagerService callService].sipAccount;
    NSString *account = [[eSpaceDBService sharedInstance].localDataManager
        userAccount];
    ESspaceUserOnlineStatus* status = [[TupContactService sharedInstance]
        onlineStatusForUser:account];
    [self reloadUserStatus:status.userStatus];
    [self updateCallBackNumber];
    [self.headImg setContactEntity:LOCAL_DATA_MANAGER.currentUser];
    // Do any additional setup after loading the view.
}

- (void)viewWillLayoutSubviews {
    [super viewWillLayoutSubviews];
    self.headImg.layer.cornerRadius = 50.0f;
}
```

删除图13中圈出部分代码,并修改宏定义NEEDREGISTERMAALOGOUT为0.

图 5-14

```
- (IBAction) showSelfDetail:(id) sender { ... }

- (IBAction) setUserStatus:(UITapGestureRecognizer *) sender { ... }

- (void) setSelfUserStatus:(ESpaceUserStatus) status
{ ... }

- (void) reloadUserStatus:(ESpaceUserStatus) status { ... }
```

删除图14中圈出部分代码。

步骤8 删除ViewController.m中无需使用代码。

图 5-15

```
#import "ViewController.h"
#import "AppDelegate.h"
#import "LoginViewController.h"
#import <TUPiOSSDK/TUPiOSSDK.h>
#import "ManagerService.h"
#import "UIViewController+CurrentViewController.h"
#import "DataConfBaseViewController.h"

@interface ViewController ()<LoginServiceDelegate>
@property (nonatomic,assign)BOOL isBeKickOut;

@end

@implementation ViewController

-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [ManagerService_loginService].delegate = self;
    [[TUPMAALoginService sharedInstance].authType = 4;

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    _isBeKickOut = NO;
    self.title = @"Main";
    [[TUPMAALoginService sharedInstance].loginService addObserver:self forKeyPath:@"serviceStatus"
    options:NSKeyValueObservingOptionNew context:NULL];
}
```

图 5-16

```
-(void)observeValueForKeyPath:(NSString *)keyPath
ofObject:(id)object
change:(NSDictionary<NSKeyValueChangeKey,id> *)change
context:(void *)context {}

-(void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

-(void)loginEventCallback:(TUP_LOGIN_EVENT_TYPE)loginEvent result:(NSDictionary *)resultDictionary
{
    switch (loginEvent)
    {
        case LOGINOUT_EVENT:
        {
            if (ECSserviceLogin == [TUPMAALoginService sharedInstance].loginService.serviceStatus) {
                [[TUPMAALoginService sharedInstance] logout:^(NSError *error) {
                    dispatch_async(dispatch_get_main_queue(), ^{
                        if (error) {
                            [self showMessage:@"Logout failed!"];
                        }else {
                            [[ECSAppConfig sharedInstance] save];
                            [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
                            [self goToLoginViewController];
                        }
                    });
                }
            }
            dispatch_async(dispatch_get_main_queue(), ^{
                [self goToLoginViewController];
            });
            break;
        }
        default:
            break;
    }
}
```

图 5-17

```
-(void)dealloc
{
    [ManagerService_loginService].delegate = nil;
    [[TUPMAALoginService sharedInstance].loginService removeObserver:self forKeyPath:@"serviceStatus"
    context:NULL];
}
```

删除图15、16、17中圈出部分代码。

步骤9 删除ContactService.m中无需使用代码。

图 5-18

```
#import "Initializer.h"
#import "ContactService.h"
#import "LoginServerInfo.h"
#import "SearchParam.h"
#import "SearchResultInfo.h"
#import "DeptInfo.h"
#import "ContactInfo.h"

#import <UIKit/UIKit.h>
#import "tup_eaddr_def.h"
#import "tup_eaddr_interface.h"
#import <TUPIOSSDK/EmployeeEntity.h>
#import <TUPIOSSDK/eSpaceDBService.h>
#import <TUPIOSSDK/ESpaceImageCache.h>
```

图 5-19

```
/**
 * This method is used to set system head image (0~9) (if completionBlock result is YES,
 * set self head ID with sysIconID)
 * 设置系统头像
 * @param sysIconID          Indicates system head image ID
 *                          系统头像id
 * @param completionBlock    Indicates callback(result: set head image result. YES
 *                          or NO)
 *                          回调，返回设置头像成功与否
 */
- (void)setSystemHead:(int)sysIconID withCompletion:(void(^)(BOOL result))completionBlock {
    TUP_RESULT set_sys_result = tup_eaddr_set_sysicon((TUP_UINT32)sysIconID);
    BOOL result = set_sys_result == TUP_SUCCESS;
    if (completionBlock) {
        completionBlock(result);
    }
    if (result) {
        [self setHeadID:[NSString stringWithFormat:@"%d", sysIconID]];
    }
}

/**
 * headId's set method
 * 头像Id 的set方法
 * @param headId NSString
 */
- (void)setHeadID:(NSString *)headId {
    EmployeeEntity *selfEntity = LOCAL_DATA_MANAGER.currentUser;
    selfEntity.headId = headId;
}
```

图 5-20

```
* This method is used to set custom head image (if completionBlock result is YES, set
  self hedID with headID)
* 设置自定义头像
@param image                Indicates custom image
*                            自定义头像
@param completionBlock      Indicates callback(result: set head image result. YES
  or NO)                    回调, 返回设置头像成功与否
*/
- (void)setHeadImage:(UIImage *)image completion:(void(^)(BOOL result, NSString *headID))
  completionBlock {
    //自定义头像接口需要上传三种尺寸的图片: 52x52 120x120 320x320
    NSData *minImg = [self imgWithSize:SIZE52 image:image];
    NSData *midImg = [self imgWithSize:SIZE120 image:image];
    NSData *maxImg = [self imgWithSize:SIZE320 image:image];

    TUP_EADDR_S_ICON* icon_info = (TUP_EADDR_S_ICON*)malloc(sizeof(TUP_EADDR_S_ICON));
    memset(icon_info, 0, sizeof(TUP_EADDR_S_ICON));
    icon_info->pcLargeIcon_data = (TUP_CHAR *)maxImg.bytes;
    icon_info->pcMediumIcon_data = (TUP_CHAR *)midImg.bytes;
    icon_info->pcSmallIcon_data = (TUP_CHAR *)minImg.bytes;
    icon_info->ulLargeIcon_len = (TUP_UINT32)maxImg.length;
    icon_info->ulMediumIcon_len = (TUP_UINT32)midImg.length;
    icon_info->ulSmallIcon_len = (TUP_UINT32)minImg.length;

    TUP_CHAR *modifyTime = (TUP_CHAR *)malloc(16);
    memset_s(modifyTime, 16, 0, 16);
    TUP_UINT32 length = 16;
    TUP_RESULT ret_set_eficon = tup_eaddr_set_deficon(icon_info, modifyTime, length);
    free(icon_info);
    // 出参modifyTime时间戳, 作为联系人headId
    NSString *mTime = [NSString stringWithUTF8String:modifyTime];
    DDLogInfo(@"set image ret: %d modify time: %@", ret_set_eficon, mTime);
    BOOL result = ret_set_eficon == TUP_SUCCESS;
    if (completionBlock) {
        completionBlock(result, mTime);
    }
    if (result) {
        [self setHeadID:mTime];
    }
}
```

图 5-21

```
65 - (void)contactModule:(TUP_MODULE)module notification:(Notification *)notification {
66     if (CONTACT_MODULE == module) {
67         switch (notification.msgId) {
68             // 联系人搜索结果
69             case TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT: {
70                 DDLogInfo(@"TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT");
71                 BOOL result = notification.param1 == TSDK_SUCCESS;
72                 if (!result) {
73                     DDLogError(@"TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT,error:%@", [NSString stringWithUTF8String:
74                         (TSDK_CHAR *)notification.data]);
75                 }
76                 TSDK_S_SEARCH_CONTACTS_RESULT *searchContactorResult = (TSDK_S_SEARCH_CONTACTS_RESULT
77                     *)notification.data;
78                 if (searchContactorResult == NULL) {
79                     DDLogWarn(@"handleSearchContact result is empty.");
80                     return;
81                 }
82                 int pageIndex = searchContactorResult->page;
83                 int totalNum = searchContactorResult->total_num;
84                 TSDK_S_CONTACTS_INFO *pstContactorInfo = searchContactorResult->contact_info;
85
86                 //查询配置自己软终端号
87                 if (notification.param2 == 100) {
88                     if (pstContactorInfo != NULL) {
89                         ContactInfo *contactInfo = [ContactInfo contactInfoTransformFrom:pstContactorInfo[0]];
90                         [[ManagerService callService] configBusinessAccount:contactInfo.terminal token:nil];
91
92                         [eSpaceDBService sharedInstance].localDataManager = [[ESpaceLocalDataManager alloc]
93                             initWithUserAccount:contactInfo.staffAccount];
94                     }
95                     return;
96                 }
97             }
98         }
99     }
100 }
```

----结束

6 FAQ

暂无。

7 修订记录

发布日期	文档版本	修订说明
2018-06-06	1.0	配套6.1.0版本(暂仅支持融合会议组网), 文档首次发布。