目录

前言 1.1

RCS开发概述 1.2

- RCS演化历史 1.2.1

RCS开发基础知识 1.3

- RCS技术架构 1.3.1
- RCS相关组织和协议规范 1.3.2
  - RCS自身 1.3.2.1
  - RCS和其他 1.3.2.2
  - 规范下载 1.3.2.3
- RCS版本 1.3.3
  - RCS UP 1.3.3.1
- RCS具体实现 1.3.4

RCS相关知识 1.4

RCS开发 1.5

- 找RCS服务商 1.5.1
  - 菊风 1.5.1.1
  - Google 1.5.1.2
- 自己从头开发 1.5.2

rcsjta 1.6

- rcsjta资料 1.6.1
- rcsjta开发 1.6.2
  - 下载代码 1.6.2.1
  - 准备环境 1.6.2.2
  - 编译apk 1.6.2.3
  - 运行apk 1.6.2.4
  - 演示RCS 1.6.2.5
    - 核心参数 1.6.2.5.1
- rcsjta源码分析 1.6.3

附录 1.7

- 名词术语 1.7.1
  - MNO 1.7.1.1
  - MSIN 1.7.1.2
- 有价值的资料 1.7.2
- 参考资料 1.7.3

RCS技术开发总结

- 最新版本： `v1.0`
- 更新时间： `20200918`

简介

总结RCS技术开发相关知识，包括开发概述和RCS演化历史，和技术相关总结，比如RCS技术架构，RCS版本历史和UP版本详情，以及RCS具体实现；以及如何进行RCS开发，包括找RCS服务商，比如国外的Google，国内的菊风，或自己从头开发，比如参考开源项目rcsjta，以及详细开发步骤比如下载代码、准备环境、编译apk、运行apk、演示RCS功能及其中的核心参数设置、rcsjta的源码分析。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- crifan/rcs_tech_dev_summary: RCS技术开发总结

如何使用此Gitbook源码去生成发布为电子书

详见：crifan/gitbook_template: demo how to use crifan gitbook template and demo

在线浏览

- RCS技术开发总结 book.crifan.com
- RCS技术开发总结 crifan.github.io

离线下载阅读

- RCS技术开发总结 PDF
- RCS技术开发总结 ePub
- RCS技术开发总结 Mobi

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com` ，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 `crifan` 还写了其他 `100+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

RCS开发概述

关于RCS技术，之前已有基本知识概述：

5G消息RCS技术总结

现在继续深入介绍，如何进行RCS开发。

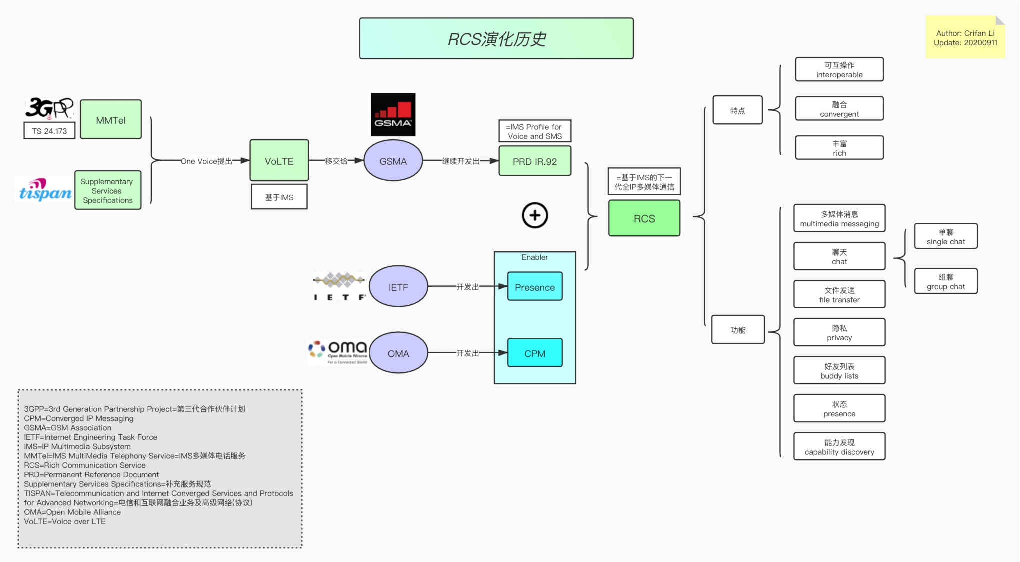
大致思路是：

- 需要先对RCS的架构和概念有深入了解
    - 然后才好继续深入了解技术细节
        - 以及最后利用开源项目rcsjta作为入手
            - 真正开始开发RCS

RCS演化历史

此处从技术角度来解释，`RCS` 的演化历史，如何诞生的，以及核心特点：

- RCS演化历史
    - 图
        - 本地

            3GPP=3rd Generation Partnership Project=第三代合作伙伴计划
            CPM=Converged IP Messaging
            GSMA=GSM Association
            IETF=Internet Engineering Task Force
            IMS=IP Multimedia Subsystem
            MMTel=IMS MultiMedia Telephony Service=IMS多媒体电话服务
            RCS=Rich Communication Service
            PRD=Permanent Reference Document
            Supplementary Services Specifications=补充服务规范
            TISPAN=Telecommunication and Internet Converged Services and Protocols for Advanced Networking=电信和互联网融合业务及高级网络(协议)
            OMA=Open Mobile Alliance
            VoLTE=Voice over LTE

        - 在线浏览
            - RCS演化历史 在线浏览
                - https://www.processon.com/view/link/5f5ad62b07912902cf81b60a
    - 文字版
        - 简述
            - 3GPP的MMTel
            - ->（One Voice提出）VoLTE
            - -> 移交给GSMA，独立出RCS项目
            - `+` 辅助协议：IETF和OMA的Presence和CPM
            - =完整的RCS
                - 概述：基于IMS的全IP多媒体通信
                - 特点：
                    - 可互操作
                    - 融合
                    - 丰富
                - 功能
                    - 多媒体消息multimedia messaging
                    - 聊天chat
                        - single chat 单聊
                        - group chat 群聊=组聊
                    - 文件发送file transfer
                    - 隐私privacy
                    - 好友列表buddy lists
                    - 状态 presence
                    - 能力发现capability discovery
        - 详解

- 最早在部署LTE期间产生了需求：基于IMS的电话服务，即VoLTE
- 通过考虑到3GPP第7版中定义的TS 24.173中规定的 MMTel=IMS MultiMedia Telephony Service=IMS多媒体电话服务 标准，One Voice行业计划构想了基于LTE的基于IMS的电话服务的简单配置文件
- One Voice还考虑了由ETSI TISPAN开发的补充服务规范Supplementary Services Specifications，该规范已转移到3GPP
- One Voice规范已提交给GSM协会（GSMA），作为基于IMS的VoLTE解决方案进行进一步增强。 基于一种语音规范，GSMA开发了永久参考文档（PRD）IR.92。 IR.92也称为语音和SMS的IMS配置文件，它指定了3GPP第8版中定义的最低强制性功能集，无线设备和网络将VoLTE实施为可互操作的，高质量的，基于IMS的电话通过LTE服务。 虽然VoLTE适合替代现有的CS语音，但是GSMA PRD IR.92还介绍了支持CS-IMS语音转换的功能。
- 在此期间，IETF=Internet Engineering Task Force=互联网工程任务组 和 OMA=Open Mobile Alliance=开放移动联盟 等标准机构正在开发使能技术，例如Presence=状态，CPM=Converged IP Messaging=融合IP消息系统 和其他支持使能器，用作下一代消息框架。 很明显，有必要发展现有的基于xMS的消息传递服务，以提供更丰富的用户体验，并允许基于它的新服务的开发。
- RCS最初是由多家主要运营商和OEM，于2007年发起的一项行业倡议。他们希望探索下一代全IP多媒体通信，从而能够快速采用移动应用程序和服务，并基于IMS架构，提供可互操作，融合和丰富的通信体验。 当由于业界的极大兴趣而使这一计划变得太大时，把计划转向了GSMA，以利用其组织支持和项目管理技能使之成为真正的全球项目。 在GSMA中，Network 2020计划=Network 2020 Program 下的 全球规范小组=Global Specification Group 负责开发和维护相关的RCS规范

RCS开发基础知识

进行RCS开发之前，需要搞清楚很多基础知识，包括但不限于：

- RCS的基本功能
- 相关各种组织、协议规范
    - 有哪些协议规范
        - 分别是哪个组织制定的
        - 大概作用是什么
        - 各个组织之间的关系如何
- RCS的各个版本
    - 有哪些版本
    - 分别对应着哪些缩写、简写
- RCS的核心架构
    - 以及所依赖的 `IMS` 的架构

RCS的功能组成

从功能角度来说：

- 基于IMS
    - 利用了IMS的控制功能，实现了丰富的多媒体服务产品组合
        - 多媒体消息multimedia messaging
        - 聊天chat
            - 单聊
            - 群聊=组聊
        - 文件发送file transfer
        - 隐私privacy
        - 好友列表buddy lists
        - 状态 presence
        - 能力发现capability discovery
            - 使得用户得以知道其好友支持哪些功能
                - 利于选择合适的通信方式与其沟通
                - 避免了各种错误和尝试，提升了用户体验
            - 实现了
                - 互操作性interoperability
                - 可触达性reachability
                - 不同用户之间的无缝通信

可以总结为：

- `RCS` = `IMS` + `其他`
    - 其他：
        - 认证 authentication
        - 授权 authorization
        - 注册 registration
        - 收费 charging
        - 路由 routing

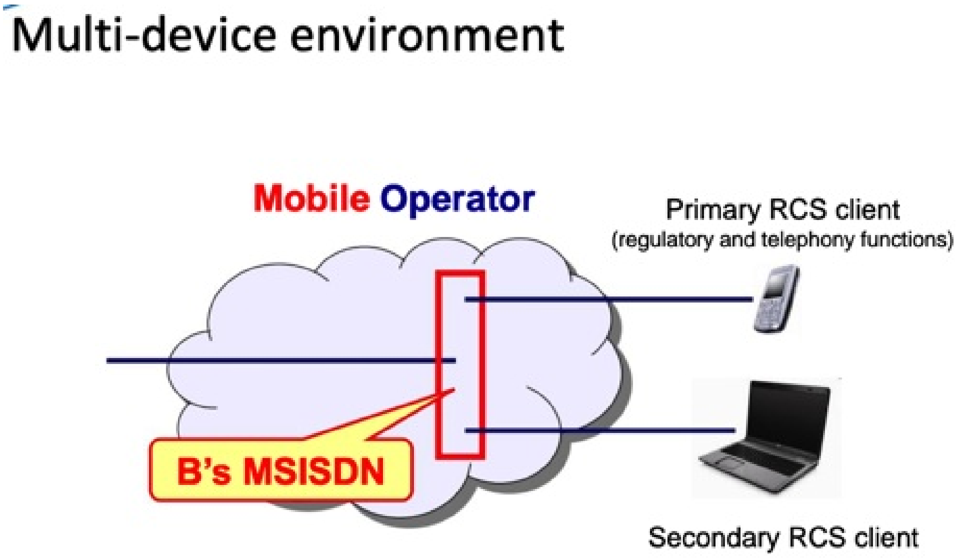
RCS涉及多方

- 为了实现RCS，涉及到
    - 多方
        - 运营商=网络运营商=移动网络运营商=MNO
        - 服务提供商=SP
    - 网络相关
        - 寻址addressing
        - 信号处理signalling
            - 包括
                - 运营商网络
                    - 内部=intra：intra-operator
                    - 之间=inter：inter-operator

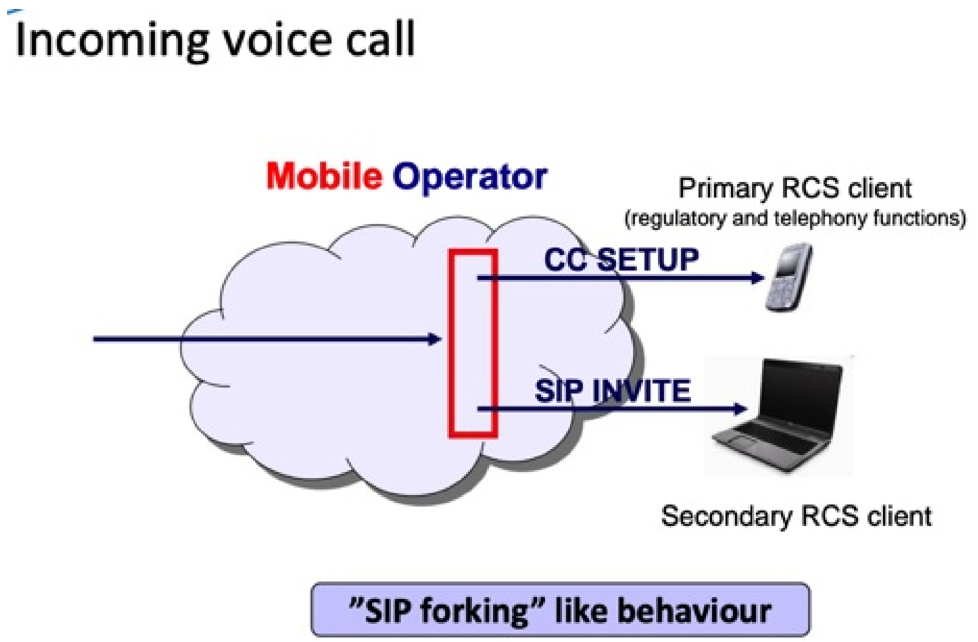
运营商部署RCS

- 部署
    - 不同运营商部署RCS的方案不同
        - 举例
            - 在VoLTE之前就部署RCS消息服务
                - 用于满足用户对于多媒体消息的需求
                    - 面对智能手机时期的OTT的各种应用的竞争

RCS多设备支持

- RCS 多设备环境
    - 多设备环境
        - 
          Multi-device environment

    - 接入语音电话

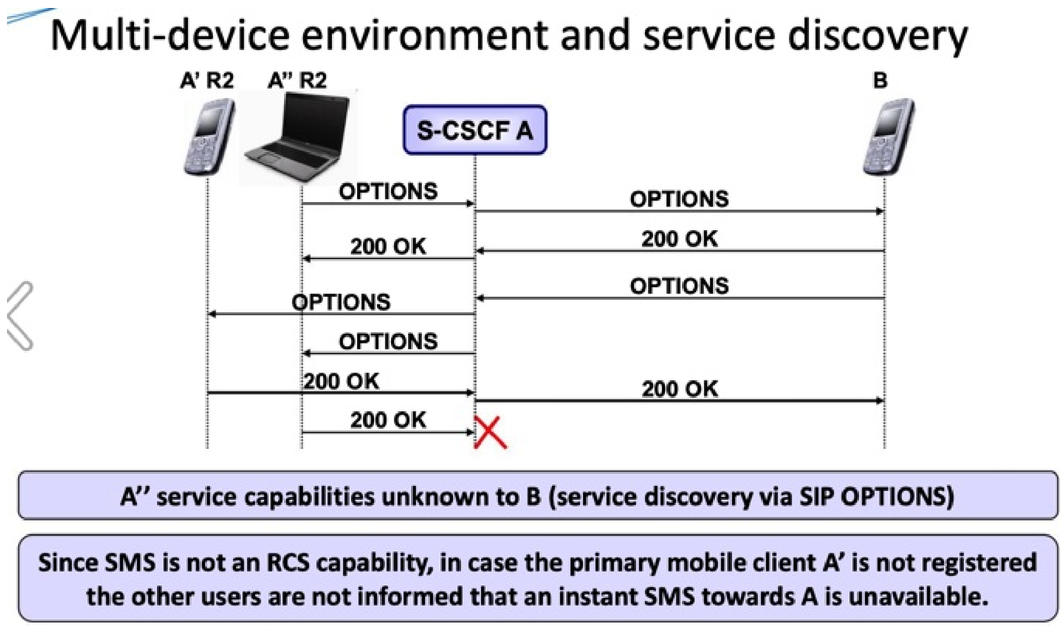


Incoming voice call

”SIP forking” like behaviour

- ■

  - 多设备环境和服务发现



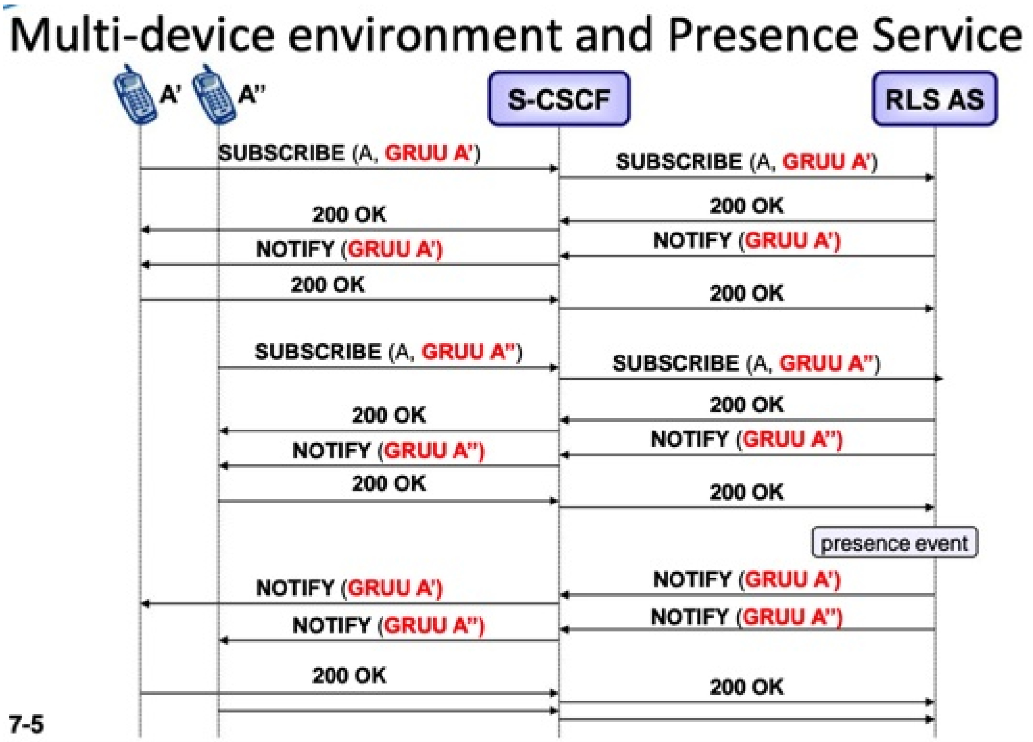
Multi-device environment and service discovery

A’’ service capabilities unknown to B (service discovery via SIP OPTIONS)

Since SMS is not an RCS capability, in case the primary mobile client A’ is not registered the other users are not informed that an instant SMS towards A is unavailable.

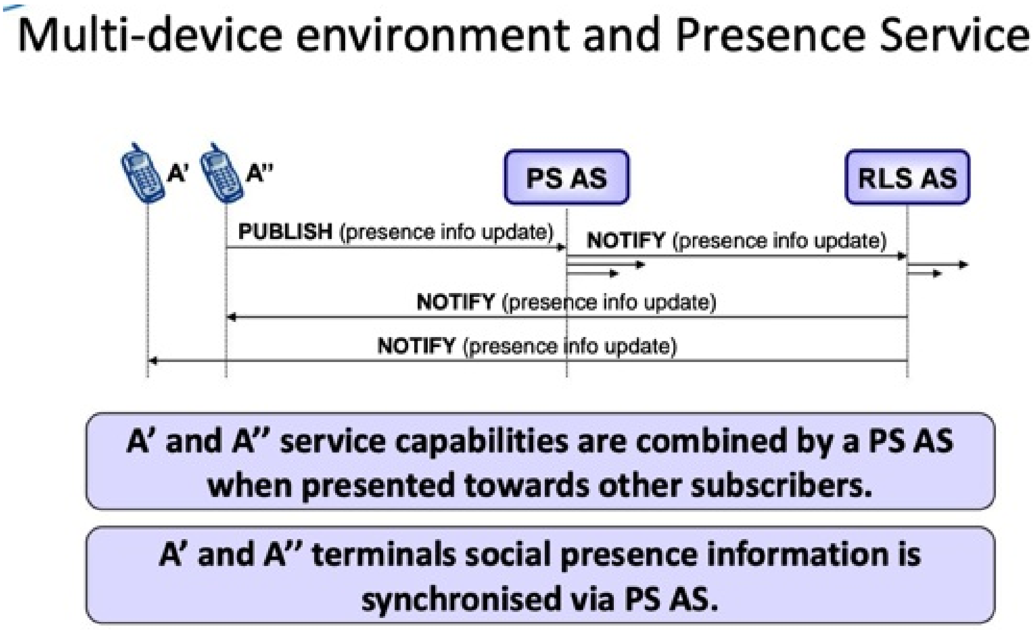
- ■

  - 多设备环境和存在服务



Multi-device environment and Presence Service

- ▪



Multi-device environment and Presence Service

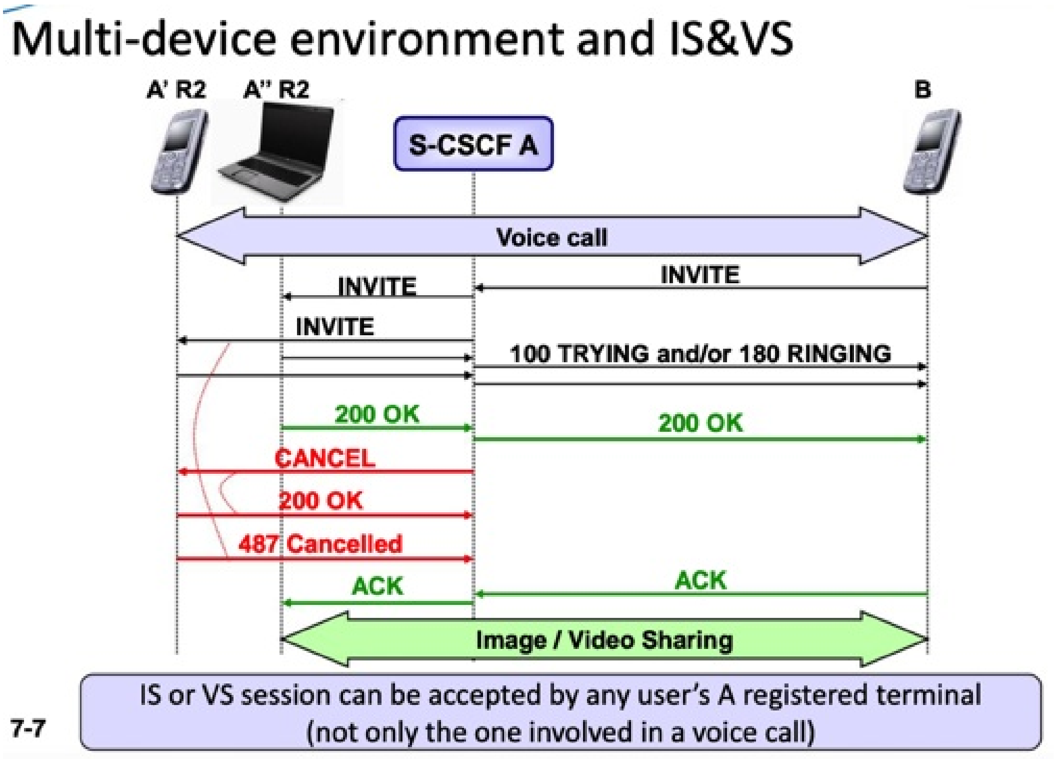
A'
A''
PS AS
RLS AS
PUBLISH (presence info update)
NOTIFY (presence info update)
NOTIFY (presence info update)
NOTIFY (presence info update)

A' and A'' service capabilities are combined by a PS AS when presented towards other subscribers.

A' and A'' terminals social presence information is synchronised via PS AS.

- ▪

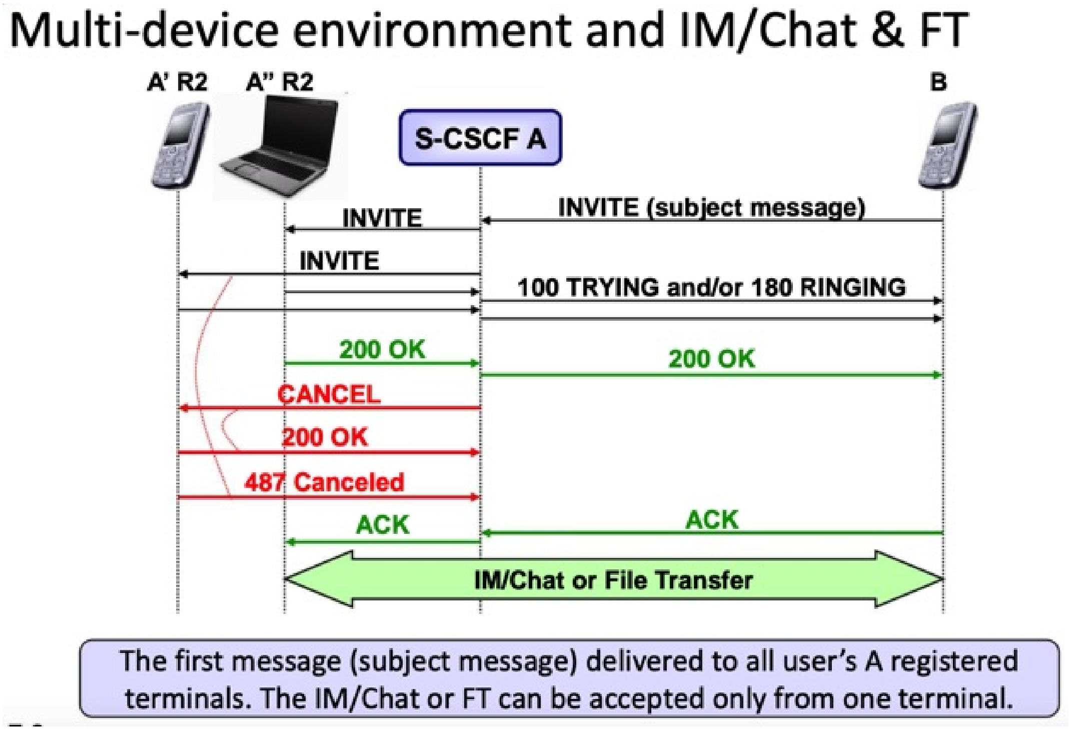
- 多设备环境和IS&VS

        - 

          Multi-device environment and IS&VS

          IS or VS session can be accepted by any user's A registered terminal (not only the one involved in a voice call)

          7-7

    - 多设备环境和 即时通讯/聊天/文件传输
        - 

          Multi-device environment and IM/Chat & FT

          The first message (subject message) delivered to all user's A registered terminals. The IM/Chat or FT can be accepted only from one terminal.

常见问题

手机没SIM卡能支持RCS吗?

- 答：可以
    - 详见：
        - https://www.gsma.com/futurenetworks/wp-content/uploads/2015/03/RCS5.3_UNI.zip

        - -》 RCC.07 v6.0.pdf
            - For the configuration of additional RCS capable devices (i.e. devices not carrying the SIM associated with a subscriber’s main identity)
                - 意思应该就是：没有SIM卡的手机，理论上也可以支持RCS的

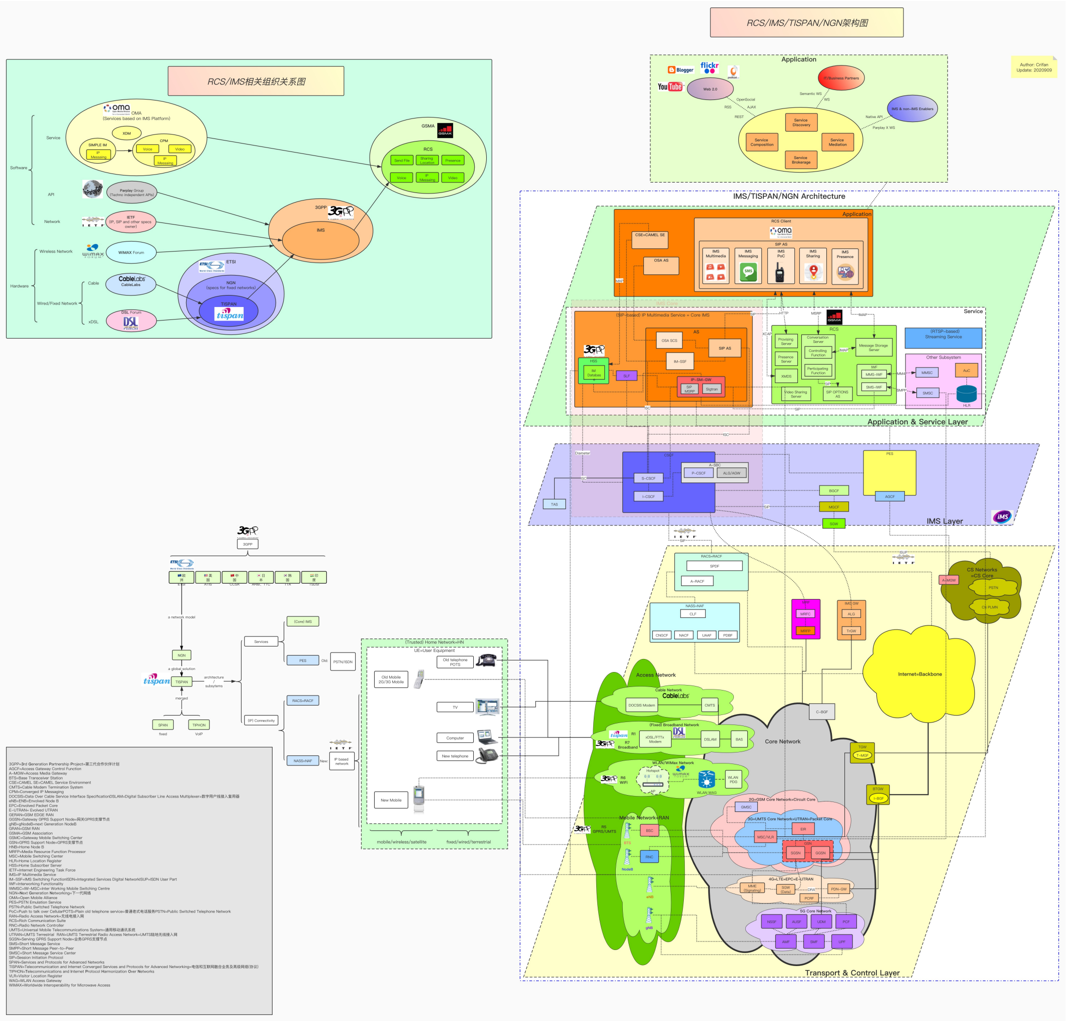
RCS技术架构

`GSMA` 制定的 `RCS` 技术(规范)，主要基于 `IMS` 技术(规范)

而 `3GPP` 制定的 `IMS` ，是从侧重固定网络的 `TISPAN` 演化而来。

而 `TISPAN` 则是 `ETSI` 的 `NGN` 架构的一种具体实现。

关于 `RCS` 和 `IMS` 相关的各种组织机构的关系，以及 `RCS` 和 `IMS` 及相关的 `NGN` 和 `TISPAN` 的架构，见下图：

- RCS/IMS/TISPAN/NGN架构和相关组织关系图
  - 图
    - 

  - 在线查看
    - RCS/IMS/TISPAN/NGN架构和相关组织关系图 | ProcessOn免费在线作图
      - https://www.processon.com/view/link/5f48c086f346fb2e2959bfbb

API架构

- TAPI：General Terminal API Architecture Overview

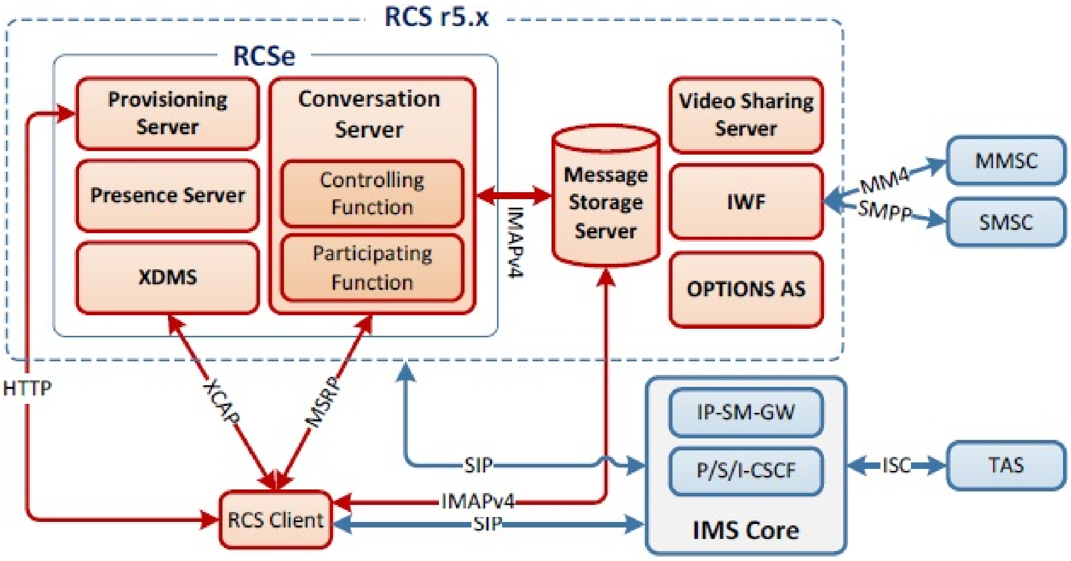
  -
- NAPI：General Network API Architecture Overview

    -

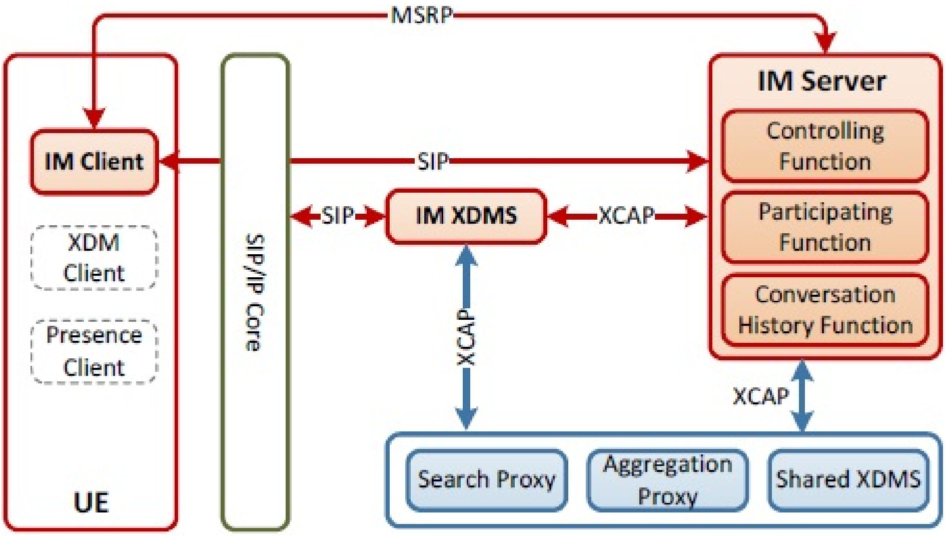
参考的RCS架构

整理出参考过的一些RCS架构相关资料：

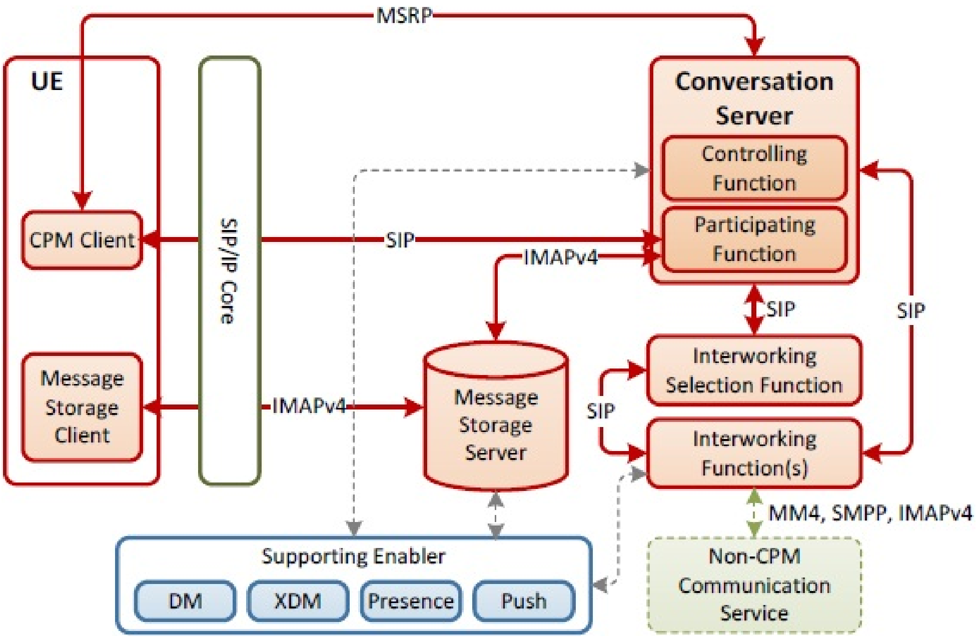
- RCS
    - RCS Architecture 5.x

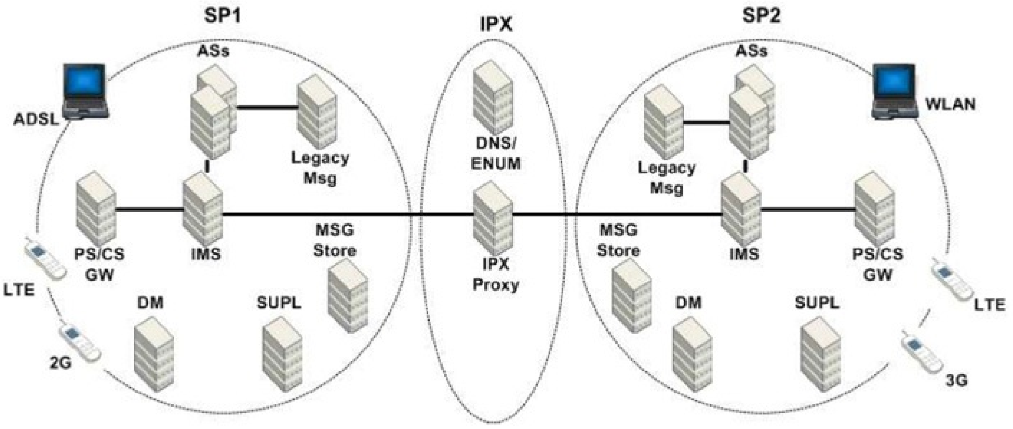


- 之前架构
  - SIMPLE IM Architecture



  - CPM Architecture



  - 采用了CPM后的RCS架构
    - Adoption of CPM architecture in RCS - Overall RCS architecture
    - 

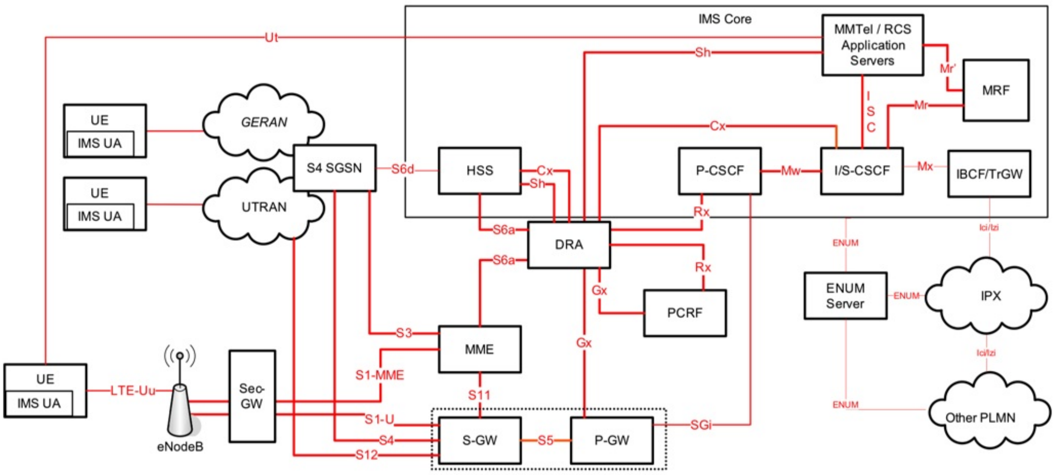
- VoLTE
  - Basic RCS VoLTE Architecture
    - 

    FIGURE 1 – BASIC RCS VOLTE ARCHITECTURE

RCS相关组织和协议规范

此处整理，RCS相关的组织和协议规范有哪些，分别作用是什么，具体关系如何。

RCS自身

RCC

`RCC` 是 `GSMA` 的旗下的一系列的规范specification=standard标准的统称

其中每个子项目编号对应着其中一个部分的内容

具体关系整理如下：

- RCC相关：
    - 通用逻辑
        - RCC NN = RCC.NN = PRD-RCC.NN=GSMA PRD-RCC.NN
            - N：数字编号
            - PRD=Permanent Reference Document
                - 注：参考自 RCC.14-v7.0-1.pdf
    - 举例
        - 文件名
            - RCC.15-v7.0.pdf
        - 文件首页内容标题
            - IMS Device Configuration and Supporting Services Version 7.0 16 October 2019
                - 核心内容：IMS Device Configuration and Supporting Services
                - 版本：7.0
                - 发布日期：16 October 2019
        - 引用此文件时的：简写=别称
            - RCC 15
            - RCC.15
            - PRD-RCC.15
- 具体包含
    - RCC.07 = RCS Advanced Communications Services and Client Specification
        - RCSe = RCS-e = RCS v4.0 ~ v5.0
            - RCSe v1.2.1
                - RCS-e - Advanced Communications: Services and Client Specification Version 1.2.2 04 July 2012
                    - https://www.gsma.com/futurenetworks/wp-content/uploads/2012/03/rcs-e_advanced_comms_specification_v1_2_2_approved.pdf
        - v2.0 = RCS 5.1
            - Rich Communication Suite 5.1 Advanced Communications Services and Client Specification Version 2.0 03 May 2013
                - https://www.gsma.com/futurenetworks/wp-content/uploads/2013/05/RCS-5.1-V2.0-UNI.pdf
        - v3.0 = RCS 5.1
            - GSMA PRD RCC.07 - RCS 5.1 - Advanced Communications: Services and Client Specification, Version 3.0, 25 September 2013
        - v5.0
            - RCS 5.0
                - RCS 5.0 Advanced Communications Services and Client Specification
            - RCS 5.2
                - Advanced Communications Services and Client Specification
        - v6.0 = RCS 5.3

    - GSMA PRD RCC.07 - RCS 5.3 - Advanced Communications: Services and Client Specification, Version 6.0, 28 February 2015
  - v8.0 = RCS 7.0 = PRD- RCC.07v8.0
    - GSMA PRD RCC.07 Rich Communication Suite 7.0 Advanced Communications Services and Client Specification, Version 8.0, 28 June 2017
  - v11.0 = RCS 11.0
    - RCC.07 - Rich Communication Suite - Advanced Communications Services and Client Specification
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.07-v11.0-1.pdf
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.07-v11.0-2.pdf
- RCC 08=Service level Interworking for Messaging Services
  - v4.0
    - GSMA PRD RCC.08 RCS 5.3 Endorsement of 3GPP TS 29.311 Service level Interworking for Messaging Services, Version 4.0 28 February 2015
      - RCS5-3GPP-SMSIW-ENDORS
  - v9.0
    - GSMA PRD RCC.08 RCS Endorsement of 3GPP TS 29.311 Service level Interworking for Messaging Services, Version 9.0, 16 October 2019
      - RCS-3GPP-SMSIW-ENDORS
        - https://www.gsma.com/newsroom/wp-content/uploads//RCC.08-v9.0.pdf
- RCC 09=Message Storage
  - v5.0
    - GSMA PRD RCC.09 RCS 5.3 Endorsement of OMA CPM 2.0 Message Storage, Version 5.0 28 February 2015
      - RCS5-CPM- MSGSTOR- ENDORS
- RCC 10=Interworking
  - v4.0
    - GSMA PRD RCC.10 RCS 5.3 Endorsement of OMA CPM 2.0 Interworking, Version 4.0 28 February 2015
      - RCS5-CPM-IW-ENDORS
  - v9.0
    - GSMA PRD RCC.10 RCS Endorsement of OMA CPM 2.2 Interworking, Version 9.0, 16 October 2019
      - RCS-CPM- IW-ENDORS
        - https://www.gsma.com/newsroom/wp-content/uploads//RCC.10-v9.0.pdf
- RCC 11= Conversation Functions
  - v4.0
    - GSMA PRD RCC.11 RCS 5.3 Endorsement of OMA CPM 2.0 Conversation Functions, Version 4.0 28 February 2015
      - RCS5-CPM-CONVFUNC-ENDORS
  - v9.0
    - GSMA PRD RCC.11 RCS Endorsement of OMA CPM 2.2 Conversation Functions, Version 9.0, 16 October 2019
      - RCS-CPM-CONVFUNC-ENDORS
        - https://www.gsma.com/newsroom/wp-content/uploads//RCC.11-v9.0.pdf
- RCC 12 = Endorsement of OMA SIP SIMPLE IM
  - v1.0, RCS 5.0 ?
    - Rich Communication Suite 5.0 Endorsement of OMA SIP/SIMPLE IM 1.0
      - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/rcs5.0_endorsement_of_oma_sip_simple_im_version1.0.pdf
  - v2.0, RCS 5.1
    - Rich Communication Suite 5.1 Endorsement of OMA SIP SIMPLE IM 2.0 Version 2.0 25 September

    2013
      - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/RCC12_RCS5_1_endorsement_of_oma_sip_simple_im_v2.0.pdf
  - v3.0, RCS 5.2 ?
    - RCS 5.2 Endorsement of OMA SIP Simple IM
      - RCSR5OMAIMEND
  - v4.0, RCS 5.3
    - GSMA PRD RCC.12 RCS 5.3 Endorsement of OMA SIP/SIMPLE IM 2.0, Version 4.0 28 February 2015
      - RCS5- SIMPLE IM- ENDORS
- RCC 13 = RCS API
  - API 2.1
    - RCS API Detailed Requirements Version 2.1 5 July 2012
      - https://www.gsma.com/esim/wp-content/uploads/2012/03/rcsapirequirementsv2_1.pdf
      - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/rcsapirequirementsv2_1.pdf
  - API 2.4
    - v1.0
      - RCC.13 - Rich Communication Suite RCS API Detailed Requirements 2.4 Version 1.0 05 June 2014
        - https://www.gsma.com/esim/wp-content/uploads/2013/02/RCC.13v1.0.pdf
  - v5.0
    - Rich Communication Suite RCS API Detailed Requirements Version 5.0 16 October 2019
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.13-v5.0.pdf
- RCC 14 = (HTTP-based) Service Provider Device Configuration
  - 版本
    - v2.0
      - GSMA PRD RCC.14 HTTP-based Service Provider Device Configuration, Version 2.0 28 February 2015
    - v5.0
      - RCC.14 - Service Provider Device Configuration Version 5.0 28 June 2017
        - https://www.gsma.com/newsroom/wp-content/uploads//RCC.14_v5.0.pdf
    - v7.0
      - GSMA PRD RCC.14 HTTP-based Service Provider Device Configuration, Version 7.0, 16 October 2019
        - https://www.gsma.com/newsroom/wp-content/uploads//RCC.14-v7.0-1.pdf
- RCC 15 = IMS Device Configuration and Supporting Services
  - v1.0
    - GSMA PRD RCC.15 IMS Device Configuration and Supporting Services, Version 1.0 02 February 2015
  - v5.0
    - IMS Device Configuration and Supporting Services Version 5.0 16 May 2018
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15_v5.0.pdf
  - v6.0
    - IMS Device Configuration and Supporting Services Version 6.0 06 December 2018
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15-v6.0.pdf
  - v7.0
    - GSMA PRD RCC.15 IMS Device Configuration and Supporting Services, Version 7.0, 16 October 2019
      - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15-v7.0.pdf
- RCC 20=Enriched Calling

        - v6.0
            - GSMA PRD RCC.20, Enriched Calling Technical Specification, Version 6.0, 16 October 2019
                - https://www.gsma.com/newsroom/wp-content/uploads//RCC.20-v6.0-2.pdf
                - https://www.gsma.com/newsroom/wp-content/uploads//RCC.20-v6.0-3.pdf
    - RCC 53=RCS Device API
        - v1.5
            - v2.0
                - RCS Device API 1.5 Specification Version 2.0
        - v1.5.1
            - v3.0
                - RCS Device API 1.5.1 Specification, Version 3.0, 23 June 2016
                    - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/RCC-53-v3-0.pdf
    - RCC 61 = RCS Common Core Service Description Document
        - GSMA PRD RCC.61 RCS Common Core Service Description Document, Version 1.1 28 February 2015
    - RCC 71=RCS UP = RCS Universal Profile
        - v1.0
            - RCS Universal Profile Service Definition Document Version 1.0 16 November 2016
                - https://www.gsma.com/futurenetworks/wp-content/uploads/2017/07/RCC.71_v1.0.pdf
        - v2.0
            - RCS Universal Profile Service Definition Document Version 2.0 28 June 2017
                - https://www.gsma.com/futurenetworks/wp-content/uploads/2017/07/RCC.71_v2.0.pdf
        - v2.4
            - GSMA PRD RCC.71 RCS Universal Profile Service Description Document, Version 2.4, 16 October 2019
                - https://www.gsma.com/newsroom/wp-content/uploads//RCC.71-v2.4.pdf

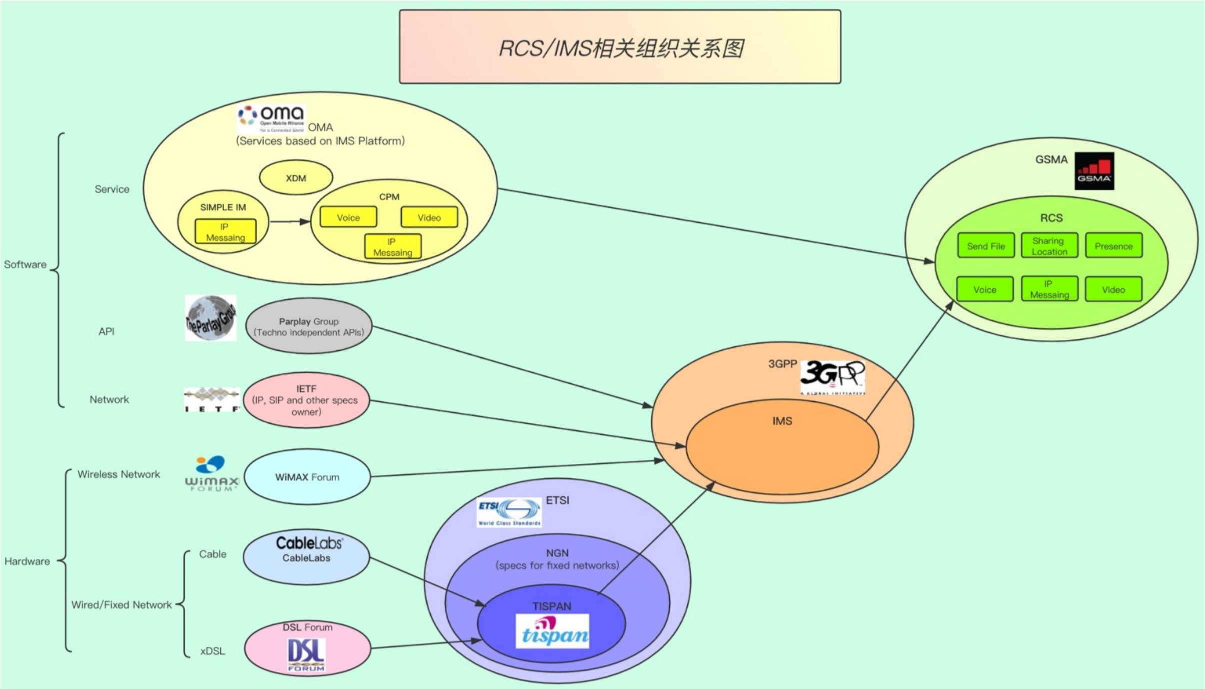
RCC和RCS核心协议关系

- 属于RCS的core或stack的
    - RCC.07 = RCS Advanced Communications Services and Client Specification
        - v3.0
        - v6.0
        - v8.0
        - 。。。
        - v11.0
- 以及后期出了UP标准：
    - RCC 71=RCS UP(Universal Profile) Service
        - v1.0
        - 。。。
        - v2.4
- 然后是RCS的API：
    - RCC 13 = RCS API
        - API v2.1
        - API v2.4
        - 等
- 以及RCS的设备的API
    - RCC 53=RCS Device API
        - v1.5
        - v1.5.1
        - v1.6.0

        - v1.6.1
        - 等
- 和如何配置IMS的设备：
    - RCC 15 = IMS Device Configuration and Supporting Services
        - v1.0
        - v7.0

RCS和其他

概览：



RCS和其他组织的协议关系

- 3GPP
  - TS 24.229
    - IMS procedure
  - `TS 22.340`
    - = `IMS spec`
    - `v15.0.0`
      - https://www.3gpp.org/ftp/Specs/archive/22_series/22.340/22340-g00.zip
  - TS 24.341
    - SMS over IMS and SMS over SGs for a cohesive SMSoIP
      - SMSoIP=SMS over IP
- GSMA
  - 概述
    - 基于IP的服务
      - `VoXXX`
        - `VoLTE` = `IR.92`
        - `ViLTE` = `IR.94`
        - `VoWiFi` = `IR.51`
      - `RCS` = `RCC.07`
        - `RCS UP` = `RCC.71`
  - 细节
    - IR
      - IR.92
        - IMS Profile for Voice and SMS

          - = `VoLTE`
            - VoLTE=Voice over LTE
      - IR.90
        - RCS Interworking
      - IR.94
        - IMS Profile for Conversational Video Service
          - = `ViLTE`
            - ViLTE=Video over LTE
      - IR.58
        - IMS Profile for VoHSPA
          - `VoHSPA` =Voice over HSPA
      - IR.51
        - IMS Profile for Voice, Video and SMS over untrusted Wi-Fi access
          - = `VoWiFi`
            - VoWiFi=Voice over Wi-Fi
    - RCC
      - RCC.07
        - = `RCS`
      - RCC.71
        - = `RCS UP` = `RCS Universal Profile`
          - `v2.3`
            - https://www.gsma.com/futurenetworks/wp-content/uploads/2018/12/RCC.71-v2.3.pdf
          - `v2.4`
            - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/10/RCC.71-v2.4.pdf
- IETF
  - `XCAP` = `XML Configuration Access Protocol`
- RCS 和 IR协议
  - 背景
    - VoLTE技术
      - Voice over LTE，基于LTE发送语音数据
      - 目的：增强传统的功能
      - 技术：基于IMS
      - 增强为：
        - 电话 -> 增强的（视频）电话
          - 通过ViLTE=Video over LTE技术
        - 短信 -> 增强的多媒体消息
          - 通过RCS技术
  - 技术关系总结
    - IMS
      - ->VoLTE
        - ->RCS
  - 而VoLTE相关协议 = IR协议
    - ->和RCS相关的IR协议
      - IR 92
        - IR.92 (VoLTE) – IMS Profile for Voice and SMS
          - 介绍
            - The IP Multimedia Subsystem (IMS) Profile for Voice and SMS, documented in this Permanent Reference Document (PRD), defines a profile that identifies a minimum mandatory set of features. These are defined in 3GPP specifications that a wireless device (the User Equipment [UE]) and network are required to implement in order to

guarantee an interoperable, high quality IMS-based telephony service and IMS-based and SGs-based Short Message Service (SMS) over Long Term Evolution (LTE) radio access.

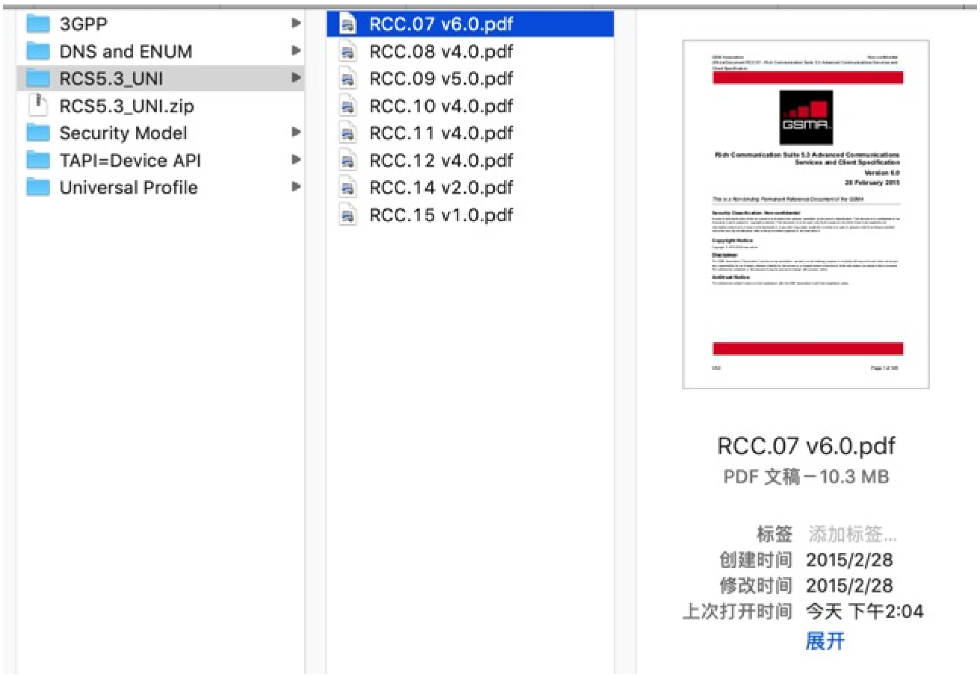
- IR 94
    - IR.94 (Video) – IMS Profile for Conversational Video Service
        - 介绍
            - The IP Multimedia Subsystem (IMS) Profile for Conversational Video Service, documented in this Permanent Reference Document (PRD), defines a minimum mandatory set of features which are defined in 3GPP specifications that a wireless device and a network are required to implement to guarantee an interoperable, high quality IMS-based conversational video service over Long Term Evolution (LTE) radio access and/or High-Speed Packet Access (HSPA) radio access. The UE and the network may support the video service on either LTE or HSPA access or on both.
- IR 34
    - IR.34 – Guidelines for IPX Provider networks (Previously Inter-Service Provider IP Backbone Guidelines)
        - 介绍
            - The Internet Protocol (IP) Packet eXchange (IPX) Network is an inter-service Provider IP backbone, which comprises the interconnected networks of IPX Providers and General Packet Radio Service (GPRS) Roaming eXchange (GRX) Providers. The purpose of this document is to provide guidelines and technical information on how these networks are set up and interconnect and how service providers will connect to the IPX Provider networks.
- IR 39
    - IR.39 – IMS Profile for High Definition Video Conference (HDVC) Service
        - 介绍
            - The IP Multimedia Subsystem (IMS) Profile for High Definition Video Conference (HDVC) service, documented in this Permanent Reference Document (PRD), defines a minimum mandatory set of features that a video communication client and the network are required to implement to guarantee an interoperable, high quality IMS-based video communication service over fixed and mobile access.
- IR 58
    - IR.58 – IMS Profile for Voice over HSPA
        - 介绍
            - The IP Multimedia Subsystem (IMS) Profile for Voice and SMS, documented in this Permanent Reference Document (PRD), defines a profile that identifies a minimum mandatory set of features which are defined in 3GPP specifications that a wireless device (the User Equipment [UE]) and network are required to implement in order to guarantee an interoperable, high quality IMS-based telephony service over High-Speed Packet Access (HSPA) radio access.
- IR 64
    - IR.64 – IMS Service Centralisation and Continuity Guidelines
        - 介绍
            - The 3rd Generation Partnership Project (3GPP) has specified the solution for centralisation of services in the IP Multimedia Subsystem (IMS) and of IMS-based service continuity in Release 8 onwards. The user shall receive services in a consistent manner when the user accesses IMS either via the Circuit Switched (CS) or the Packet Switched (PS) domain. Service continuity is supported between CS and PS domains.
- IR 65
    - IR.65 – IMS Roaming and Interworking Guidelines

      - 介绍
        - The 3rd Generation Partnership Project (3GPP) architecture has introduced a subsystem known as the IP Multimedia Subsystem (IMS) as an addition to the Packet-Switched (PS) domain. IMS supports new, IP-based multimedia services as well as interoperability with traditional telephony services. IMS is not a service per se, but a framework for enabling advanced IP services and apps on top of a packet bearer.
      - v30.0
        - IMS Roaming, Interconnection and Interworking Guidelines Version 30.0 08 April 2019
          - https://www.gsma.com/newsroom/wp-content/uploads/IR.65-v30.0.pdf
  - IR 67
    - IR.67 – DNS/ENUM Guidelines for Service Providers and GRX/IPX Providers
      - 介绍
        - Inter Service Provider IP communications are starting to evolve to support services other than GPRS Roaming. Many, if not all, of these services rely upon DNS. Therefore, it is of utmost importance for the interworking and stability of such services that service providers have all the necessary information to hand to ease configuration of their DNS servers upon which such services rely.
  - IR 88
    - IR.88 – LTE Roaming Guidelines
      - 介绍
        - This document aims to provide a standardised view on how Long-Term Evolution (LTE) and Evolved Packet Core (EPC) networks can interwork in order to provide “Next Generation Mobile Network” capabilities when users roam onto a network different from their HPMN. Expectations of the “Next Generation Mobile Network” capabilities are described in the GSMA Project Document: Next Generation Roaming and Interoperability (NGRAI) Project Scope White Paper.
  - NG 102
    - NG.102 – IMS Profile for Converged IP Communications
      - 介绍
        - This document defines a profile that identifies a minimum mandatory set of common IMS functionalities that are defined in 3GPP specifications and other GSMA PRDs that a wireless device (the User Equipment [UE]) and network are required to support in order to guarantee interoperable, high quality IMS-based and Mobile Operator provided Converged IP Communications Services (VoLTE, ViLTE, VoWi-Fi and RCS).
  - IR 95
    - IR.95 – SIP/SDP Profile for inter-IMS NNI
      - 介绍
        - This document describes a generic SIP/SDP profile for interconnection and roaming NNI between operators’ IMS networks for the purposes of exchanging traffic originating from and terminating to the respective operators’ customers. This document profiles SIP/SDP for the GSMA defined IMS based services (i.e. VoLTE, ViLTE, VoWi-Fi, SMSoIP and RCS services).
      - v7.0
        - SIP-SDP Inter-IMS NNI Profile Version 7.0 15 April 2020
          - https://www.gsma.com/newsroom/wp-content/uploads//IR.95-v7.0.pdf
          - https://www.gsma.com/newsroom/wp-content/uploads//IR.95-v7.0-2.pdf
  - IR 90
    - v16.0
      - RCS Interworking Guidelines Version 15.0 08 April 2019
        - https://www.gsma.com/newsroom/wp-content/uploads//IR.90-v16.0.pdf
- `rcsjta/docs/SUPPORTED-STANDARDS.txt` 中涉及到的

- IETF
    - 多个RFC相关协议
        - 很多都和SIP和SDP相关
            - SIP=Session Initiation Protocol
            - SDP=Session Description Protocol
        - -> chat聊天等内部的会话session 会用到这些协议
- 3GPP
    - TS 24.229 - IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), Stage 3
    - TS 24.279 - Combining Circuit Switched (CS) and IP Multimedia Subsystem (IMS) services, Stage 3
- OMA=Open Mobile Alliance
    - Instant Messaging using SIMPLE Architecture, v1.0
    - Instant Messaging Requirements, v1.0
    - Instant Messaging using SIMPLE, v1.0
    - Presence SIMPLE Architecture, v2.0
    - Resource List Server (RLS) XDM Specification, v2.0
- GSMA
    - RCS Release 1 - Functional Description
    - RCS Release 1 - Technical Realization
    - RCS Release 2 - Functional Description
    - RCS Release 2 - Technical Realization
    - RCS Release 2 - Endorsement of OMA SIP/SIMPLE IM 1.0
    - RCS Release 2 - Management Objects
    - RCS-e Version 1.1 - Services and Client Specification
    - RCS-e Version 1.2.1 - Services and Client Specification
    - RCS-e GSMA RCS IOT RCS-e Implementation Guidelines 2.1
    - RCS-e GSMA RCS IOT RCS-e Implementation Guidelines 3.1
- IR的
    - IR74 - Video Share Interoper ability Specification
    - IR79 - Image Share Interoperability Specification

RCS相关规范下载

- RCS 5.3
    - GSMA PRD RCC.07 - RCS 5.3 - Advanced Communications Services and Client Specification, Version 6.0, 28 Feb 2015
        - https://www.gsma.com/futurenetworks/wp-content/uploads/2015/03/RCS5.3_UNI.zip
            - 说明：成套的RCS的相关规范文件



3GPP
DNS and ENUM
RCS5.3_UNI
RCS5.3_UNI.zip
Security Model
TAPI=Device API
Universal Profile

RCC.07 v6.0.pdf
RCC.08 v4.0.pdf
RCC.09 v5.0.pdf
RCC.10 v4.0.pdf
RCC.11 v4.0.pdf
RCC.12 v4.0.pdf
RCC.14 v2.0.pdf
RCC.15 v1.0.pdf

RCC.07 v6.0.pdf
PDF 文稿 – 10.3 MB

标签 添加标签...
创建时间 2015/2/28
修改时间 2015/2/28
上次打开时间 今天 下午2:04
展开

- GSMA规范下载主入口
    - GSMA | Resources - Newsroom
        - https://www.gsma.com/newsroom/resources/
- 其他
    - http://www.gsma.com/network2020/wp-content/uploads/2015/06/RCC-55-v2-0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.14_v5.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.07-v11.0-2.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.13-v5.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15-v7.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.08-v9.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.14-v7.0-1.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.20-v6.0-3.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.20-v6.0-2.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.10-v9.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.11-v9.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.71-v2.4.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.07-v11.0-1.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.07-v11.0-2.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15_v5.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.14_v5.0.pdf
    - https://www.gsma.com/newsroom/wp-content/uploads//RCC.15-v6.0.pdf
    - https://www.gsma.com/esim/wp-content/uploads/2013/02/RCC.13v1.0.pdf
    - http://www.gsma.com/esim/wp-content/uploads/2012/03/rcsapirequirementsv2_1.pdf

  - https://www.gsma.com/futurenetworks/wp-content/uploads/2012/03/rcs-e_advanced_comms_specification_v1_2_2_approved.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2013/05/RCS-5.1-V2.0-UNI.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2015/03/RCS5.3_UNI.zip
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2017/07/RCC.71_v1.0.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2017/07/RCC.71_v2.0.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/rcsapirequirementsv2_1.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/rcs5.0_endorsement_of_oma_sip_simple_im_version1.0.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2019/09/RCC12_RCS5_1_endorsement_of_oma_sip_simple_im_v2.0.pdf
  - https://www.fftelecoms.org/app/uploads/2017/08/interface_specification_for_rcs_interconnection_dec_2016-2.pdf
  - https://www.gsma.com/futurenetworks/wp-content/uploads/2015/05/RCC-59-v1-0.pdf
  - https://www.gsma.com/newsroom/wp-content/uploads/IR.65-v30.0.pdf
  - https://www.etsi.org/deliver/etsi_es/282000_282099/282001/02.00.00_50/es_282001v020000m.pdf
  -

RCS版本

关于RCS的总体版本演化，摘录之前在

RCS版本历史 · 5G消息RCS技术总结

已整理的

RCS 版本			UP(Universal Profile)版本	
主版本	主要功能	发布时间	版本	主要功能
RCS 1.0	**增强型通信簿**：多种发起通信方式，如语音呼叫，文件传输，即时消息；显示联系人的能力，显示公开的社交呈现信息，如肖像，昵称等；黑名单；网络通信簿 **增强消息**：支持群组聊天、表情、多媒体内容、“正在输入文字”的状态、文件传输等 **增强型通话体验**：通话过程中进行图片共享、视频共享等。 支持宽带接入的客户端使用 RCS 功能	2008.12		
RCS 2.0	支持多设备，支持无卡客户端作为次设备登陆使用 支持网络通讯录在多个设备同步 支持地理位置信息呈现	2009.06		
RCS 3.0	支持非通话过程中的视频共享 允许宽带接入的终端作为唯一终端使用 允许宽带接入终端收发 SMS/MMS，显示送达状态通知 允许宽带接入终端收发 SMS/MMS，显示送达状态通知 全面支持 VoLTE 支持大文本消息，消息记录在多种设备上显示，与短信后向兼容	2009.11		
RCS 4.0	支持流媒体共享时的暂停和恢复功能 支持双方对共享图片的涂鸦功能 全面整合 RCS1-4 以及 RCS-e 功能 支持窝蜂网络与 Wi-Fi 之间无缝切换	2010.05		
RCS-e v1.0		2011.05		
RCS-e v1.2		2011.11		
RCS-e V1.2.2		2012.4		
RCS 5.0	支持 OMA CPM 和 OMA SIMPLE IM,离线消息存储转发，断点续传等 支持 IP 视频电话，地理位置信息共享 支持群消息存储转发 支持离线文件的存储转发，群组中进行文件传输	2012.04		
RCS 5.1	支持地理位置信息在地图上显示 支持网络黑名单	2012.08		
RCS 6.0	支持可视语音邮件等	2016.03		
		2016.11	UP 1.0	包括诸如功能发现之类的核心功能，这些功能将在区域之间互操作，聊天，群组聊天，文件传输，音频消息传递，视频共享，多设备，丰富的呼叫，位置共享和实时素描
RCS 7.0	支持聊天机器人，SMS 后备功能等	2017.07	UP 2.0	引入了消息即平台（MaaP），并改进了消息传递性，API，插件集成以及身份验证和安全性 它还引入了会话商务(Conversational Commerce)，使用户能够与平台内的品牌进行交互，通过消息完成搜索、购物、订票、支付等一站式业务体验 提供了个人到应用程序(A2P,Person-to-Application)消息传递和消息传递插件的初始规范，包括发现，丰富的消息气泡和标签，以及对现有功能（如群聊图标，群聊主题更改）的改进和文件传输增强功
		2017.12	UP 2.1	
RCS 8.0	支持其他 Chatbots 功能和 vCard 4.0	2018.05	UP 2.2	新增 Chatbots 的验证，用于增加用户与 Chatbot 之间的信任 优化了在 P2A(Person to Application) 场景中，和聊天机器人对话时的用户的匿名的实现 极大地增强了 Chatbot 平台与网络运营商之间的接口定义，以提供有关可用聊天机器人的信息，扩展了搜索中的目录查询功能
RCS 9.0	支持 EAP AKA 作为对配置请求进行身份验证的一种方式消息传递 支持大型寻呼机模式独立消息 支持撤销独立消息 提供一种在寻呼机模式独立消息中指示功能的方法 启用对聊天机器人使用独立消息传递使用 P2P 消息传递时回退到 Chatbot 通信的过程与聊天机器人联系 富媒体通话 提供一个在端到端 VoLTE 呼叫中使用 SIP 标头的过程，以提供呼叫前服务	2018.12	UP 2.3	
RCS 10.0	在 RCS 中添加对 5G NR 的支持 在 SIP 请求和响应中对内容压缩的可选支持 消息书签 聊天机器人具有永久菜单 Rich Card 可以具有特定的 CSS 建议操作来打开 Web 视图	2019.10	UP 2.4	删除插件集成，并包括集成的无缝 Webview

RCS协议版本演化

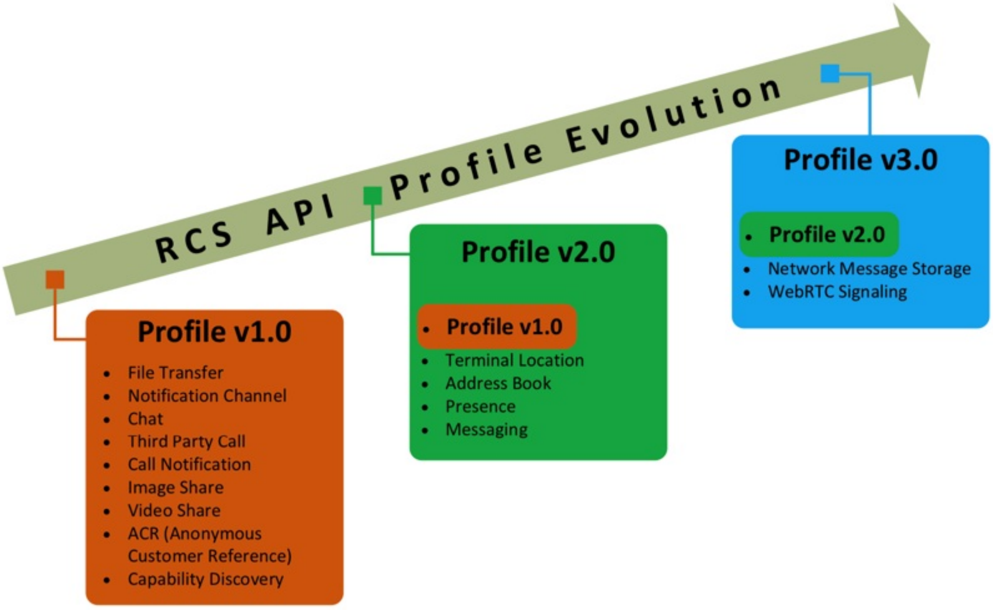
- Evolution of GSMA RCS standards
  -

RCS的API的版本

`RCS` 的 `API` 是 `OMA` 组织定义的

- OMA定义的RCS的API的版本历史
  - 图
    - RCS API Profiles Evolution in Open Mobile Alliance

Figure 22. RCS API Profiles Evolution in Open Mobile Alliance



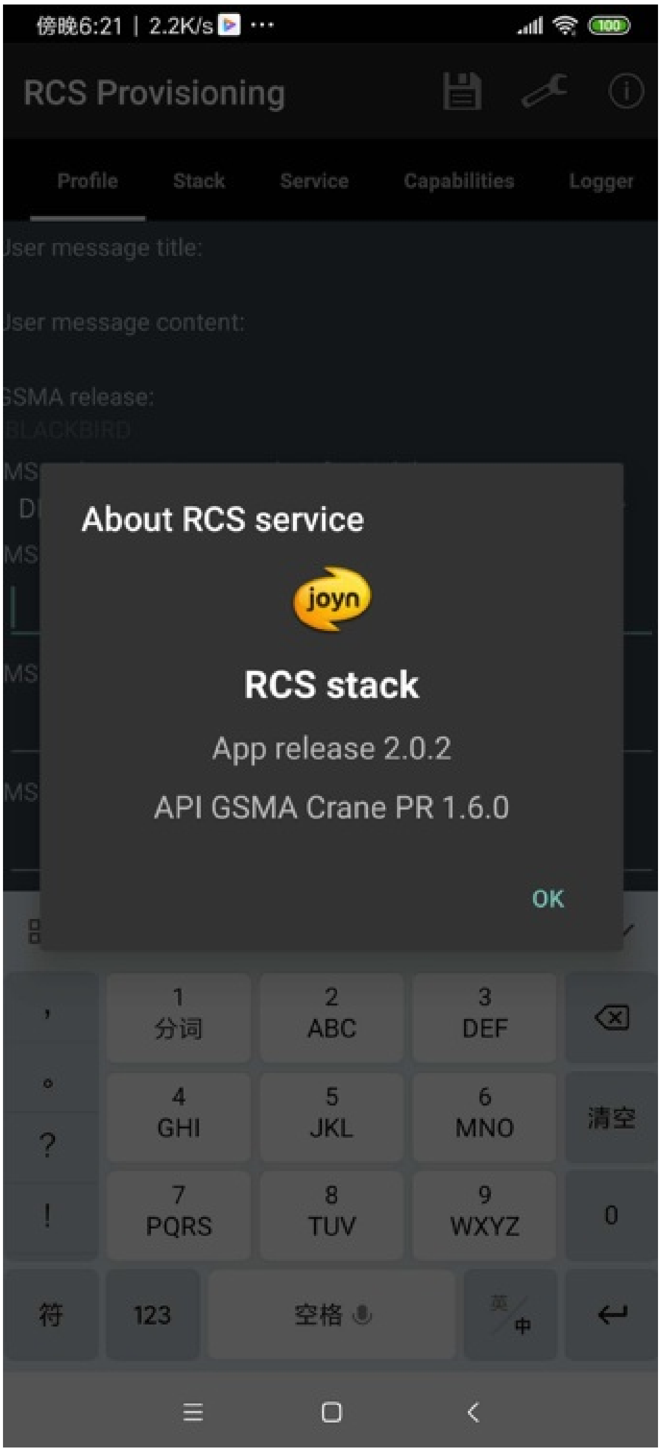
  - 文字版
    - Profile v3.0

      - Network Message Storage
      - WebRTC Signaling
      - Profile v2.0
        - Terminal Location
        - Address Book
        - Presence
        - Messaging
        - Profile v1.0
          - File Transfer
          - Notification Channel
          - Chat
          - Third Party Call
          - Call Notification
          - Image Share
          - Video Share
          - ACR (Anonymous Customer Reference)
          - Capability Discovery

RCS的名词关系：Albatros、Blackbird、Crane、joyn

目前理解：

- RCS版本=API 版本
  - Albatros
    - =RCS 5.1
  - Blackbird
    - = RCS 5.2
  - Crane
    - = API 1.6.0 / 1.6.1
      - RCS的core的apk -> About中看到 `API GSMA Crane PR 1.6.0`

- 

Jser message title:

Jser message content:

GSMA release:
BLACKBIRD

About RCS service

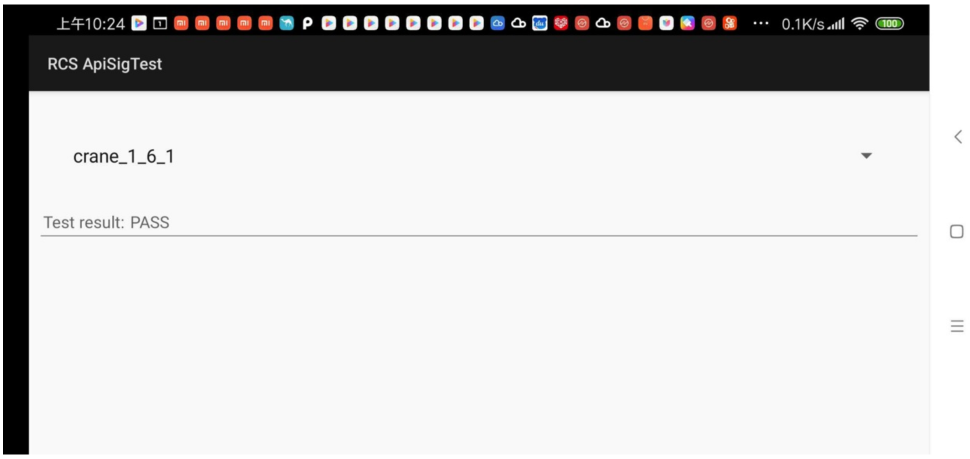
RCS stack

App release 2.0.2

API GSMA Crane PR 1.6.0

OK

- ApiSigTest = API签名测试 中是： `crane_1_6_1`

        - 

上午10:24 0.1K/s

RCS ApiSigTest

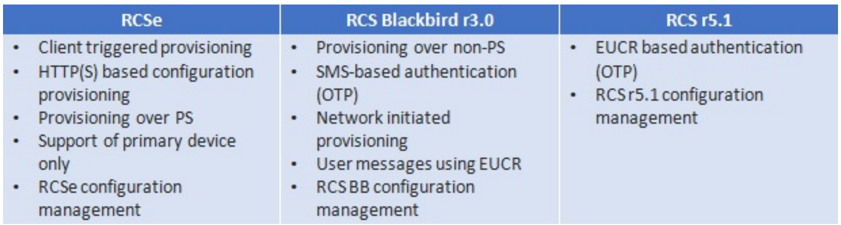
crane_1_6_1

Test result: PASS

根据 rcsjta/Branches.md at wiki · android-rcs/rcsjta 最新理解：

- RCS分2部分
  - RCS 的core 或stack：RCS stack的版本
  - TAPI=Terminal API：终端设备API接口 的版本
- 目前已发布的版本
  - 主分支
    - master
      - RCS Crane = RCS Crane PR (Priority Release)：RCS 5.3
      - TAPI 1.6.1
    - tapi_1.6.1
      - RCS Blackbird：RCS 5.30
      - TAPI 1.6.1
    - tapi_1.5.1
      - RCS Blackbird：RCS 5.0 和 5.1 + RCS-e 1.2.2
      - TAPI 1.5.1
    - Albatros
      - RCS Albatros 2.0
      - Albatros TAPI
  - 临时分支
    - securitymodel3
      - RCS Blackbird：RCS 5.0 和 5.1 + RCS-e 1.2.2
      - TAPI 1.5
      - 安全模型
    - cpm
    - tapi_0.9.0

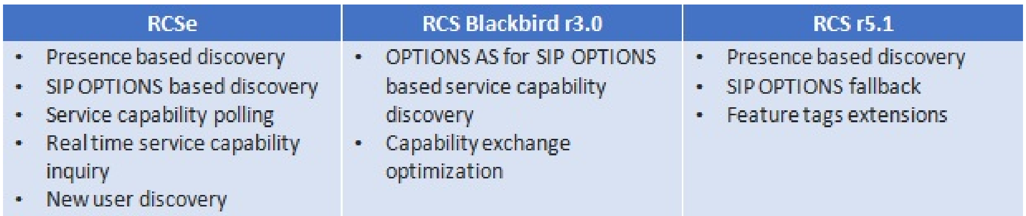
- RCS相关概念
  - 协议版本
    - Albatros=RCS 5.1
    - Blackbird= RCS 5.2
      - RCS Blackbird
        - 目的：在 RCSe - RCS r5.1 之间的过渡
          - 功能对比
            - 图

          - 

RCSe	RCS Blackbird r3.0	RCS r5.1
• Client triggered provisioning • HTTP(S) based configuration provisioning • Provisioning over PS • Support of primary device only • RCSe configuration management	• Provisioning over non-PS • SMS-based authentication (OTP) • Network initiated provisioning • User messages using EUCR • RCS BB configuration management	• EUCR based authentication (OTP) • RCS r5.1 configuration management

        - 文字
          - RCSe
            - 关注重点：简单的IP聊天
          - RCS r5.1
            - 关注重点：聚合IP通讯服务
              - converged IP communication service
    - 发展经历了
      - RCS 3.0
      - RCS 4.0
    - 包含功能
      - 基于IP的聊天功能
      - 基于HTTP的文件传输
      - 兼容旧消息
      - 支持多设备
      - 基于SIP OPTIONS的服务发现
      - 增强的Provisioning功能
      - 内容分享
      - 基于WiFi的语音电话
    - joyn Blackbird支持情况
      - 已认证
        - Samsung
        - LG
        - Nokia
        - Microsoft
        - Sony
        - HTC
        - Jibe Mobile
        - Huawei
        - TCL
      - 支持的设备
        - 运营商?
          - Germany的Deutsche Telekom
          - Vodafone Germany的多个设备
            - HTC One M7
            - M8 Mini
            - LG70
            - Samsung S4
            - Samsung S4 Mini
            - Samsung S5
            - Sony Xperia Z1
- Crane
  - 基于
    - RCS 5.3
    - RCC.61 Common Core feature set

      - 是Blackbird的升级版
      - API
        - TAPI 1.6.0
        - TAPI 1.6.1
      - 主要更新
        - Common Message Store- for the backup and restoration of messages and an enabler for a multi-device experience.
        - Voice and Video calling over Wi-Fi
        - Green Button Promise for voice and video
        - Audio Messaging
        - Security against malware
  - RCS extensions
    - Applications adding functionality to native devices utilising RCS APIs
  - RCS领域内的Provisioning
    - Provisioning
      - 直译：
        - n. 准备金提取
        - v. 供应补给品（provision的ing形式）
          - provision
            - 直译：
              - n. 规定；条款；准备；[经] 供应品
              - vt. 供给…食物及必需品
    - 可以理解为：服务service（的供给）
      - -》提供（RCS）服务
      - 最新：正确翻译应该是：
        - `Provisioning` = `开通` = `开通服务`
          - 开通RCS服务，需要各种条件
            - 需要进行各种参数配置，才能开通RCS服务
    - Configuration Provisioning
      - 是一个过程
        - 获取配置
          - 用于让支持了RCS的设备，发现和知道相关服务提供方的对应的配置信息
      - 目的
        - RCS设备，获取了配置参数（configuration parameters）后，才方便访问IMS core和RCS的服务
      - 何时、次数
        - 在用户注册到IMS core之前，有且要做1次Configuration Provisioning
      - 配置参数configuration parameters
        - 包含
          - 各种MO=Management Objects
      - 设备接入（用于访问configuration server）对比
        - RCSe：只支持PS（数据包交换 ）网络 方式
        - RCS Blackbird：支持多种设备
          - 接入设备类型
            - 带SIM卡的
            - 不带SIM卡的
          - 举例
            - 用户的主设备，通过WiFi接入
            - 用户的不带（或忽略是否有）SIM卡的从设备
      - 设备接入方式=Provisioning方式
        - PS网络

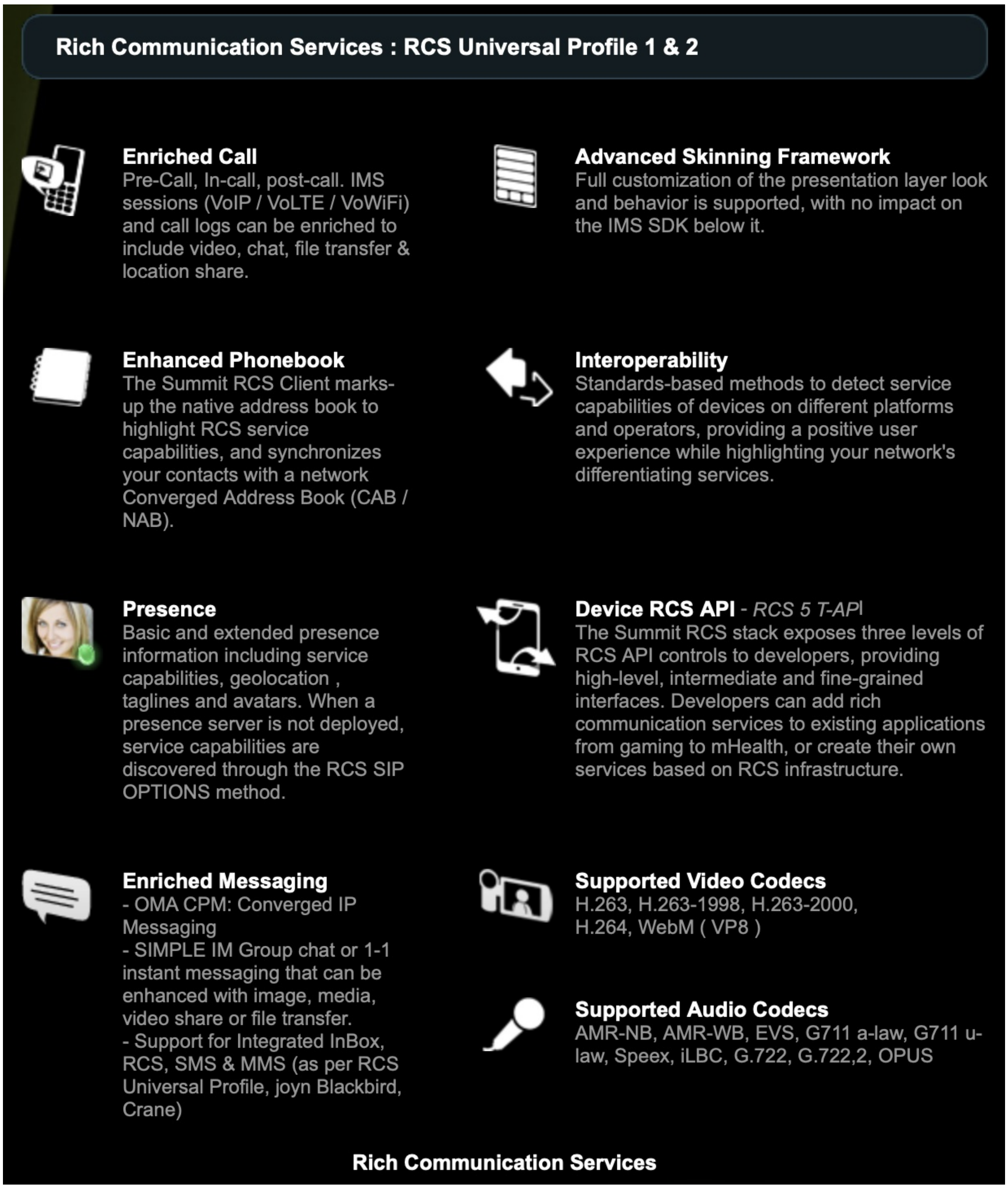
                - 配置服务器 认证设备（是否有效）有多种方式
                    - 基于IMS的认证
                    - 基于用户IP地址的认证
            - WiFi网络
                - 特殊性：无法通过上述的IMS或IP地址认证
                    - 因为设备往往是私有IP地址
                        - 进一步：从设备无法获取（主设备的）IMSI
                - 认证方式
                    - OTP认证
                        - 逻辑：配置服务器生成OTP，通过SMS，发送给主设备
                        - 注：
                            - RCS 5.1中，也可通过EUCR去申请获取OTP
                                - 通过IMS网络返回的XML的SIP消息的body中包含OTP
        - 更新Provisioning配置
            - 背景：configuration server也可以触发让RCS客户端进行配置初始化
                - 实现方式
                    - SMS
                    - EUCR
            - 常见场景
                - 服务配置更改后
                - RCS客户端版本升级后
                - 用户设备中（重新）激活被禁用的RCS客户端
- Service Capability Discovery=服务能力发现
    - 是什么：一个过程
        - 寻找发现一个地址本里的联系人，是否具有对应能力
            - 以此去确定其是否是一个RCS的用户
    - 实现方式
        - 根据技术不同和使用场景可分2种
            - 基于Presence
            - 基于SIP OPTIONS
    - 不同协议对比
        - 

RCSe	RCS Blackbird r3.0	RCS r5.1
• Presence based discovery • SIP OPTIONS based discovery • Service capability polling • Real time service capability inquiry • New user discovery	• OPTIONS AS for SIP OPTIONS based service capability discovery • Capability exchange optimization	• Presence based discovery • SIP OPTIONS fallback • Feature tags extensions

- RCS UP
    - 支持功能
        - capability discovery能力发现
            - 实现不同区域间互操作
        - chat聊天
        - group chat群聊
        - file transfer文件传输
        - audio messaging语音消息
        - video share视频分享
        - multi-device多设备支持
        - enriched calling增强电话
        - location share位置分享
        - live sketching=Live Sketch Sharing
            - Sketch=素描？ = 涂鸦？

    - 核心
      - MaaP=Messaging as a Platform
        - 支持
          - RCS business messaging=RCS商业消息
          - Rich Cards=富媒体卡片
          - privacy control=隐私控制
          - spam protection=垃圾过滤防护
  - MaaP和Chatbot
    - RCS UP 2.0协议支持Chatbot平台
      - 不同厂家会实现自己的API
        - 举例
          - 三星的MaaP API
            - GSMA MaaP API (1.8.*)
    - 相关协议是
      - RCS UP 2.0
        - Section 15 and Annex A
          - 针对：Chatbot product manager and designer
      - RCS 7.0
        - Section 3.6.10
          - 针对Chatbot developer
  - 相关理解
    - RCS重点在于：Advanced Communications （更）高级的通讯
      - 和谁对比：
        - 传统的 普通的通讯
          - 早期：基于文本的SMS消息
          - 后来的：多媒体消息MMS
      - 高级在于支持更多功能
        - 如前所述的 群聊，视频分享，位置分享等等
          - 早期通讯方式所不支持（也无法支持）的功能

RCS UP

- `RCS UP`
  - = `RCS Universal Profile`
  - 核心功能
    - 概览
      - 

        Rich Communication Services : RCS Universal Profile 1 & 2

        Enriched Call
        Pre-Call, In-call, post-call. IMS sessions (VoIP / VoLTE / VoWiFi) and call logs can be enriched to include video, chat, file transfer & location share.

        Advanced Skinning Framework
        Full customization of the presentation layer look and behavior is supported, with no impact on the IMS SDK below it.

        Enhanced Phonebook
        The Summit RCS Client marks-up the native address book to highlight RCS service capabilities, and synchronizes your contacts with a network Converged Address Book (CAB / NAB).

        Interoperability
        Standards-based methods to detect service capabilities of devices on different platforms and operators, providing a positive user experience while highlighting your network's differentiating services.

        Presence
        Basic and extended presence information including service capabilities, geolocation , taglines and avatars. When a presence server is not deployed, service capabilities are discovered through the RCS SIP OPTIONS method.

        Device RCS API - *RCS 5 T-API*
        The Summit RCS stack exposes three levels of RCS API controls to developers, providing high-level, intermediate and fine-grained interfaces. Developers can add rich communication services to existing applications from gaming to mHealth, or create their own services based on RCS infrastructure.

        Enriched Messaging
        - OMA CPM: Converged IP Messaging
        - SIMPLE IM Group chat or 1-1 instant messaging that can be enhanced with image, media, video share or file transfer.
        - Support for Integrated InBox, RCS, SMS & MMS (as per RCS Universal Profile, joyn Blackbird, Crane)

        Supported Video Codecs
        H.263, H.263-1998, H.263-2000, H.264, WebM (VP8)

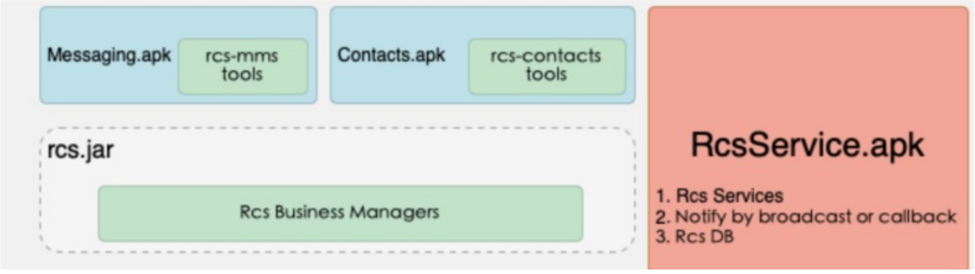
        Supported Audio Codecs
        AMR-NB, AMR-WB, EVS, G711 a-law, G711 u-law, Speex, iLBC, G.722, G.722,2, OPUS

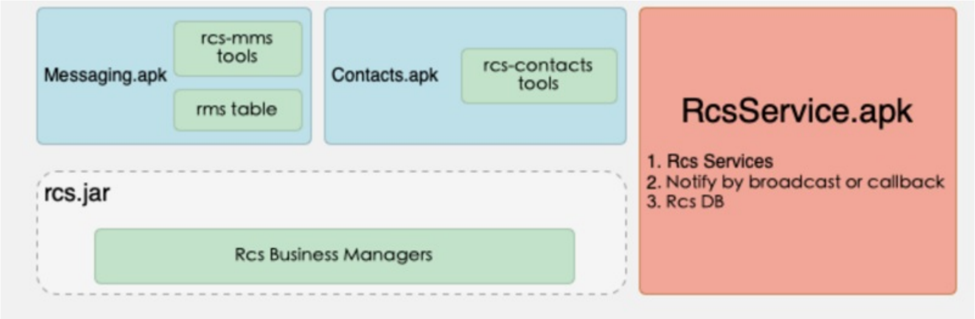
        Rich Communication Services
    - 详解
      - Enriched Call
        - Pre-Call, In-call, post-call. IMS sessions (VoIP / VoLTE / VoWiFi) and call logs can be enriched to include video, chat, file transfer & location share.
      - Enhanced Phonebook
        - The Summit RCS Client marks-up the native address book to highlight RCS service capabilities, and synchronizes your contacts with a network Converged Address Book (CAB / NAB).
      - Enriched Messaging

    - OMA CPM: Converged IP Messaging
    - SIMPLE IM Group chat or 1-1 instant messaging that can be enhanced with image, media, video share or file transfer.
    - Support for Integrated InBox, RCS, SMS & MMS (as per RCS Universal Profile, joyn Blackbird, Crane)
- Presence
    - Basic and extended presence information including service capabilities, geolocation , taglines and avatars. When a presence server is not deployed, service capabilities are discovered through the RCS SIP OPTIONS method.
- Advanced Skinning Framework
    - Full customization of the presentation layer look and behavior is supported, with no impact on the IMS SDK below it.
- Interoperability
    - Standards-based methods to detect service capabilities of devices on different platforms and operators, providing a positive user experience while highlighting your network's differentiating services.
- Device RCS API
    - = RCS 5 T-API
    - The Summit RCS stack exposes three levels of RCS API controls to developers, providing high-level, intermediate and fine-grained interfaces. Developers can add rich communication services to existing applications from gaming to mHealth, or create their own services based on RCS infrastructure.
- Supported Video Codecs
    - H.263, H.263-1998, H.263-2000, H.264, WebM (VP8)
- Supported Audio Codecs
    - AMR-NB, AMR-WB, EVS, G711 a-law, G711 u-law, Speex, iLBC, G.722, G.722,2, OPUS

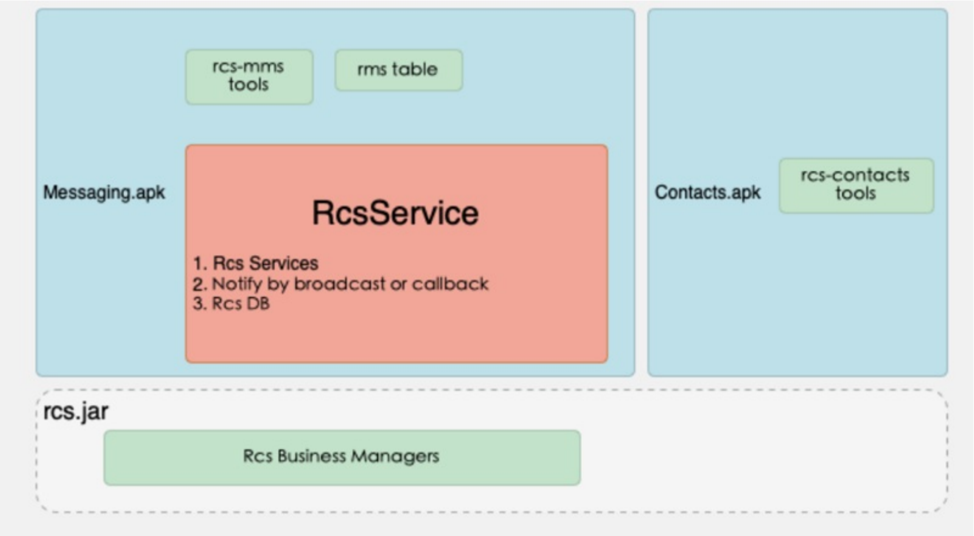
RCS具体实现

关于RCS具体的实现，实施，有多种方式。总结如下：

- RCS Native
  - 菊风 三大方案
    - 方案一：主要包含 `RCSService` （ `RCS` 服务）； `TelephonyProvider` （系统短彩数据库）； `Messaging/Mms Contacts`
      - 代表性案例： `海信` 、 `小米` 、 `酷比` 、 `天珑`
      - 架构：
        - 

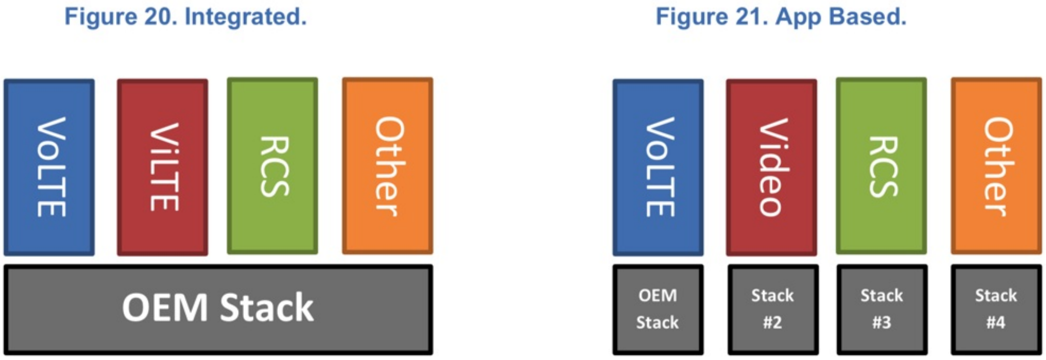
      - 细节
        - 针对这些手机厂商的方案，牵扯到了系统应用，菊风在其系统上增加 `RCSService.apk`
        - 改造原短彩信的数据库，升级 `MaaP` 需整个系统升级即可实现
        - 但由于 `service.apk` 是系统级应用保证常驻，能够长时间保留信息
        - 改动短信应用会在系统上增加 `RCSService.apk` ，修改系统短信的数据库
    - 方案二：针对不想修改 `TelephonyProvider` 源码的用户，在该场景下 `RCSService` 不处理任何数据库操作，上层可以在收到广播后进行数据的存储
      - 代表性案例： `联想` 、 `360`
      - 架构
        - 

      - 细节
        - 菊风为联想、360提供的 `RCS Native` 解决方案并不改造系统数据库，只需添加 `service.apk` ，再改造短信应用即可完成
    - 方案三：一般是在方案2的基础上将 `RCSService` 代码打成 `aar` 包供上层直接集成，确保集成该 `aar` 应用存放在 `system/priv-app/` 下
      - 代表性案例： `OPPO` 、 `魅族`
      - 架构



- 细节
    - 菊风为 OPPO 和魅族提供sdk助其短信集成，不会修改系统中的其他任何应用，包括数据库、服务增加等；用户只需升级短信应用即可提供 MaaP 的相应能力，魅族本身短信应用快速升级十分方便，升级代价最小；只改动短信应用

- RCS消息
    - 美国



crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-11-04 22:20:11

RCS相关知识

关于和RCS相关基础和背景知识，比如

- IMS
- VoLTE
- 移动网络基础知识

等等，详见：

移动网络演化史

RCS开发

对于真正需要从头开发RCS的话，可以有2种思路：

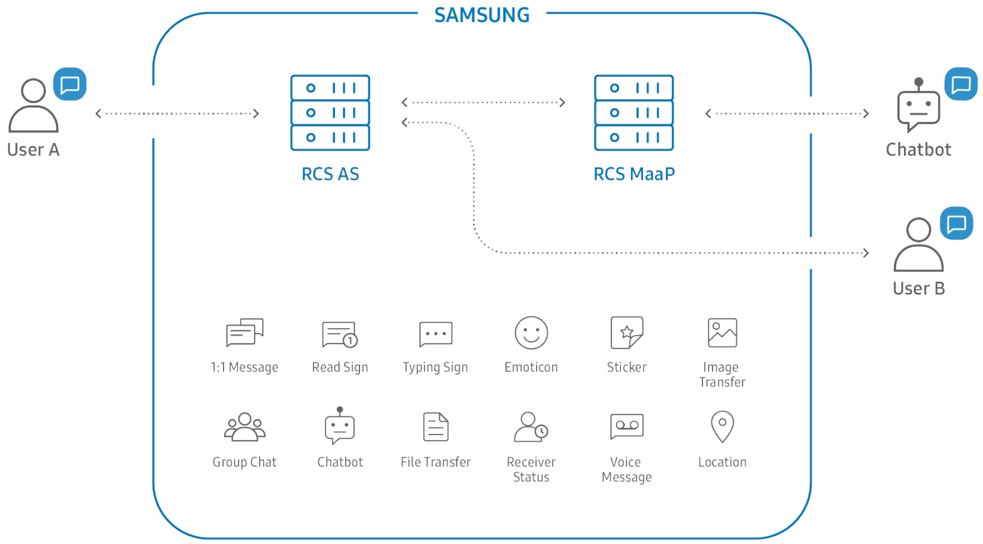
- 找RCS服务商=开发商
- 自己从头开发

下面详细介绍如何开发。

找RCS服务商

想要开发RCS应用，可以借助已有的，可以提供RCS方案的服务商=供应商，去整合出自己的应用。

此处列出一些可供参考的服务商，或者是RCS生态相关的各方资源。

- Samsung
  - Samsung RCS Developers
    - https://rcs.developer.samsung.com
  - Samsung RCS chatbot
    - https://rcs.developer.samsung.com/develop/Introduction/overview.html
  - Samsung RCS ecosystem
    - https://rcs.developer.samsung.com/develop/Introduction/Samsung%20RCS%20ecosystem/RCS-chatbot-ecosystem.html
      - 架构
        - 

  - Samsung Bot SDK API
    - https://rcs.developer.samsung.com/develop/Development/Javascript%20chatbot%20development/API%20reference/javascript-chatbot-api-reference.html
- Oracle
  - Developing Rich Communication Services Applications
    - https://docs.oracle.com/cd/E55119_01/doc.71/e55126/wd_rcsapps.htm#WSEWD363
- VoipSwitch
  - RCS SDK
    - 主页
      - Rich Communication Suite software development kit
        - http://www.voipswitch.com/products/voip-softphones/rcs-sdk/
    - 简介
      - The Softphone Software Development Kit (SDK) provides a comprehensive framework by which a solution developer can quickly build their own RCS applications and benefit from all the functionalities of our whitelabel RCS products.
    - 平台
      - iOS
      - Android
    - 主要功能特性

        - Built in media stack
        - SIP signaling stack
        - Presence
        - Chat and group chat
        - High Definition audio and video codecs including OPUS, VP8 and others
        - ICE framework with TURN support for the most effective NAT traversal and peer to peer media connectivity
        - Secure VoIP – TLS encryption and SRTP for media
        - Ready to use examples for Android Studio and Xcode
- Sinch
    - RCS HTTP REST API - Sinch Documentation For Developers
        - https://developers.sinch.com/docs/rcs-http-rest

菊风

国内的RCS服务商，做的比较专业的有：菊风。

- 菊风
    - 官网
        - https://rcs.juphoon.com/cn/
    - RCS SDK
        - https://rcs.juphoon.com/cn/products/rcs-sdk.html
            - 英文
                - https://rcs.juphoon.com/en/products/rcs-sdk.html
    - Chatbot 开放平台
        - https://rcs.juphoon.com/cn/products/chatbot.html

如何用菊风开发RCS?

- 问：想要用菊风去开发RCS，如何操作?
- 答：需要单独联系合作 暂无SDK下载
    - QQ： `2703650800`
    - 咨询电话： `13056832331`
- 研究过程：

目前已有的官网下载

菊风能力开放平台（beta）

其实只是VoIP的SDK，文档也是VoIP，SIP相关的文档：

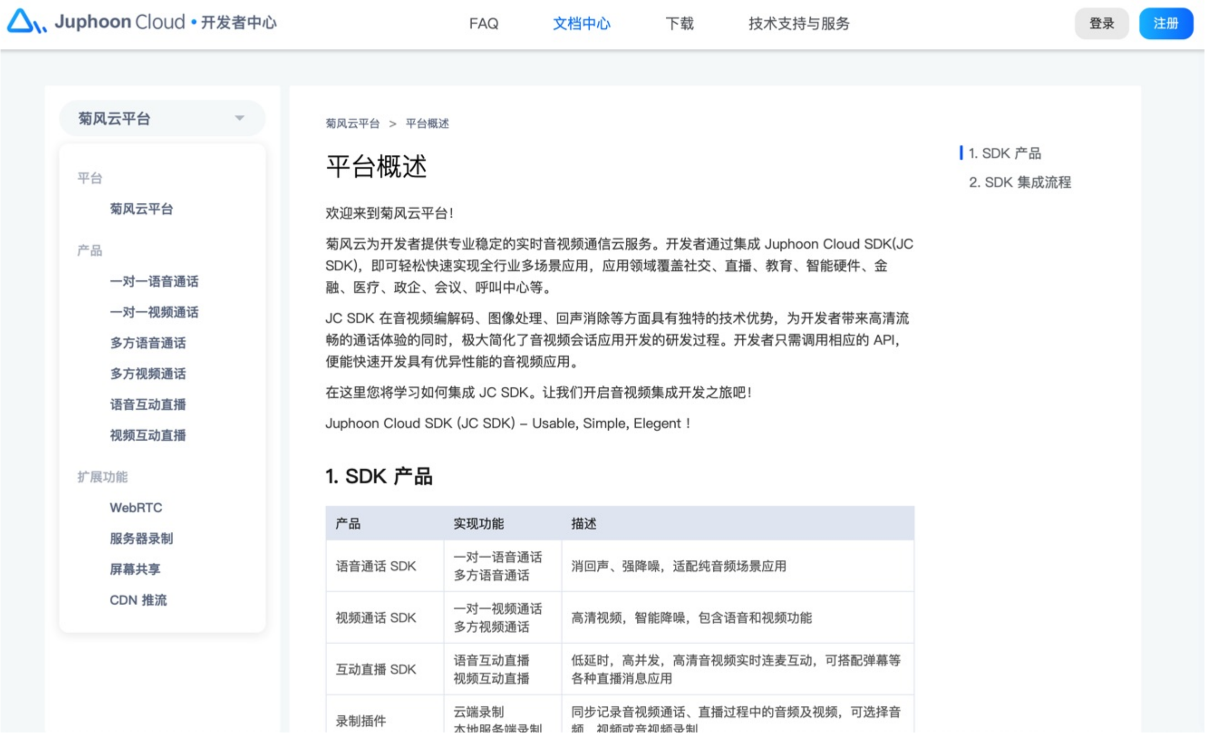
- 下载 Juphoon VoIP 控件化SDK
    - VOIP_SDK 控件化集成指南 Android 版
        - http://download.juphoon.com:8083/SDKCapOpen/doc/voip_sdk_doc_android/juphoon-voip-sdk-user-manual-android.pdf
- 下载 Juphoon VoIP客户端App
    - JusPhone for iOS 用户手册
        - http://download.juphoon.com:8083/SDKCapOpen/doc/jusphone_for_ios/jusphone_for_ios.pdf

而不是RCS的文档

以及目前平台：菊风云平台

主要也都是 音视频相关的

菊风云开发者网站



菊风云平台 > 平台概述

平台概述

欢迎来到菊风云平台！

菊风云为开发者提供专业稳定的实时音视频通信云服务。开发者通过集成 Juphoon Cloud SDK(JC SDK)，即可轻松快速实现全行业多场景应用，应用领域覆盖社交、直播、教育、智能硬件、金融、医疗、政企、会议、呼叫中心等。

JC SDK 在音视频编解码、图像处理、回声消除等方面具有独特的技术优势，为开发者带来高清流畅的通话体验的同时，极大简化了音视频会话应用开发的研发过程。开发者只需调用相应的 API，便能快速开发具有优异性能的音视频应用。

在这里您将学习如何集成 JC SDK。让我们开启音视频集成开发之旅吧！

Juphoon Cloud SDK (JC SDK) – Usable, Simple, Elegent！

1. SDK 产品

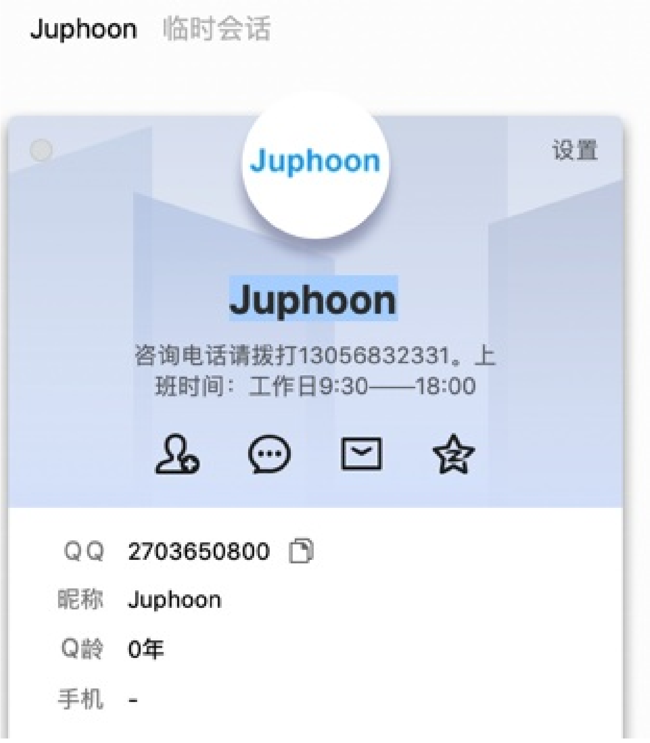
产品	实现功能	描述
语音通话 SDK	一对一语音通话 多方语音通话	消回声、强降噪，适配纯音频场景应用
视频通话 SDK	一对一视频通话 多方视频通话	高清视频，智能降噪，包含语音和视频功能
互动直播 SDK	语音互动直播 视频互动直播	低延时，高并发，高清音视频实时连麦互动，可搭配弹幕等各种直播消息应用
录制插件	云端录制 本地服务端录制	同步记录音视频通话、直播过程中的音频及视频，可选择音频、视频或音视频录制

暂没找到RCS相关内容

通过

菊风融合通信-5G消息|RCS商业富媒体消息标准技术解决方案提供商 -> `Chatbot 开放平台` -> 立即咨询

找到了联系方式：



Juphoon

咨询电话请拨打13056832331。上班时间：工作日9:30——18:00

QQ 2703650800

昵称 Juphoon

Q龄 0年

手机 -

- QQ：`2703650800`
- 昵称：Juphoon
- 咨询电话请拨打：`13056832331`

- 上班时间：工作日9:30——18:00

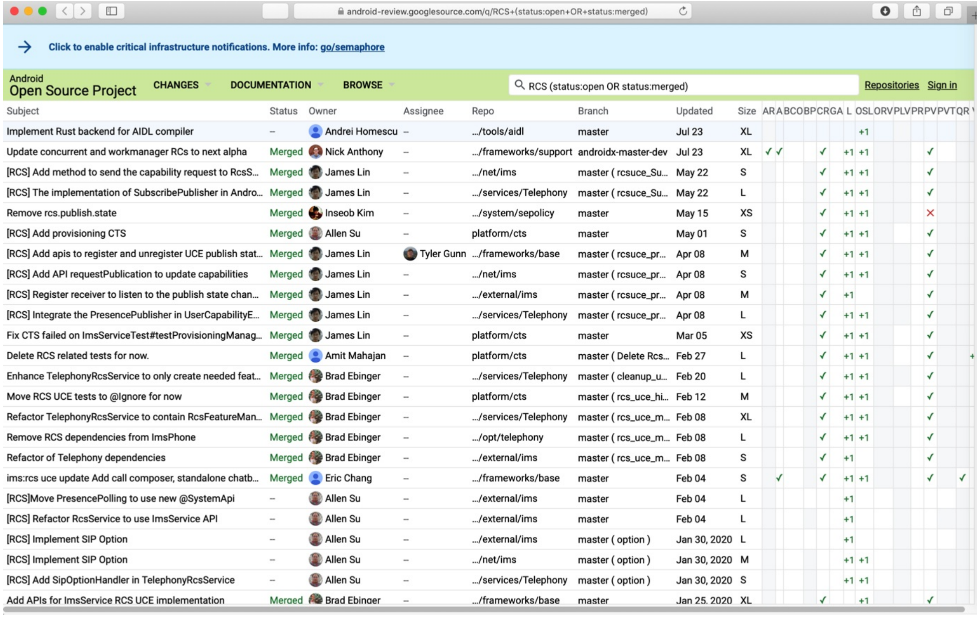
需要的可以去联系。

Google

Google也对RCS支持的很好。

如果需要开发RCS，可以考虑直接用Google相关平台和服务。

- Google
    - 概览
        - `Messages`
            - 支持RCS消息：Chat features=聊天功能
        - `Jibe` = `Jibe Platform`
            - 主页：https://jibe.google.com/jibe-platform/
            - 包含
                - `Jibe Cloud`
                    - 概述
                        - Easily launch and manage RCS services with Google-hosted infrastructure
                        - The Jibe Cloud can help carriers quickly scale RCS services, iterate in short cycles, and benefit from improvements immediately. A GSMA-certified, hosted service, the Jibe Cloud supports clients that implement the Universal Profile across leading smartphone platforms.
                - `Jibe Hub`
                    - 概述
                        - Access the global RCS network with one connection
                        - The Jibe Hub provides mobile operators with a simple connection to the global RCS network. Easily interoperable with third-party RCS networks, one connection delivers worldwide interconnection
    - RCS开发
        - RCS Business Messaging | Google Developers
            - https://developers.google.com/business-communications/rcs-business-messaging
        - Business Messaging (B2C)
            - https://jibe.google.com/business-messaging/
        - IMS
            - 实现 IMS | Android 开源项目 | Android Open Source Project
                - https://source.android.com/devices/tech/connect/ims
                    - Android 9 引入了一个名为 ImsService 的新 SystemApi 接口，可帮助您实现 IP 多媒体子系统 (IMS)。ImsService API 是在 Android 平台与供应商或运营商提供的 IMS 实现之间明确定义的接口。
        - google的源码中已包含部分RCS功能了？
            - Android 11 RCS API? : UniversalProfile
            - ->
            - RCS (status:open OR status:merged) · Gerrit Code Review)



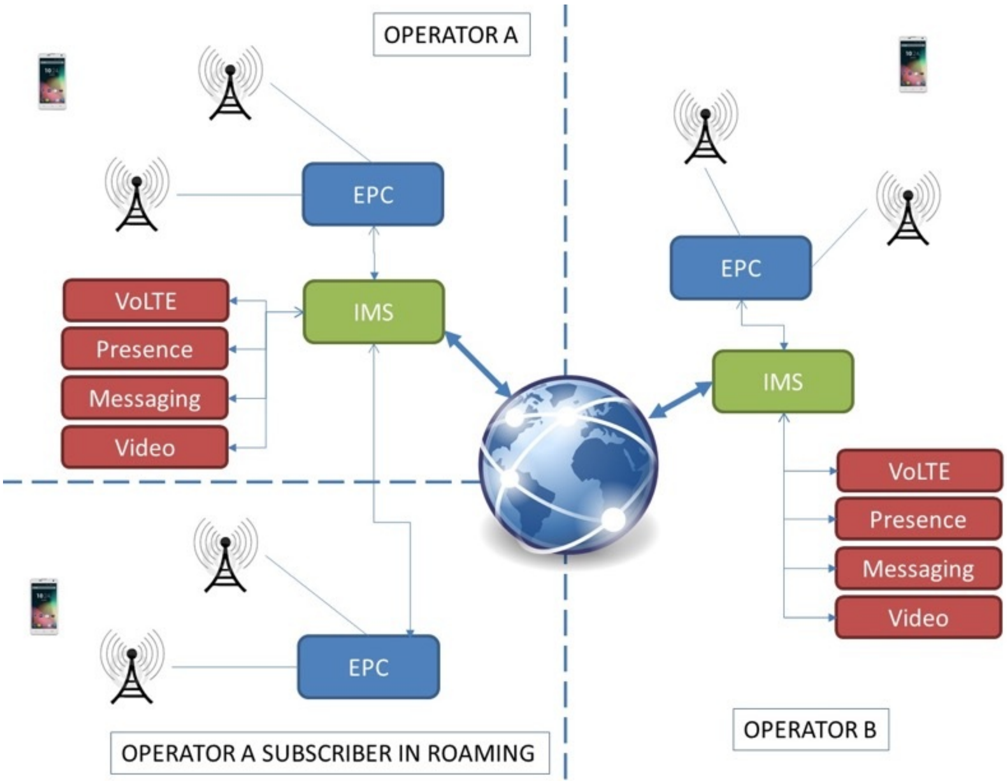
android-review.googlesource.com/q/RCS+(status:open+OR+status:merged)

Click to enable critical infrastructure notifications. More info: go/semaphore

Android Open Source Project CHANGES DOCUMENTATION BROWSE RCS (status:open OR status:merged) Repositories Sign in

Subject	Status	Owner	Assignee	Repo	Branch	Updated	Size	AR	A	BCO	BP	CR	GA	L	OSL	ORV	PLV	PR	PV	PVT	QR	
Implement Rust backend for AIDL compiler	–	Andrei Homescu	–	.../tools/aidl	master	Jul 23	XL								+1							
Update concurrent and workmanager RCs to next alpha	Merged	Nick Anthony	–	.../frameworks/support	androidx-master-dev	Jul 23	XL	✓	✓			✓		+1	+1				✓			
[RCS] Add method to send the capability request to RcsS...	Merged	James Lin	–	.../net/ims	master (rcsuce_Su...	May 22	S					✓		+1	+1				✓			
[RCS] The implementation of SubscribePublisher in Andro...	Merged	James Lin	–	.../services/Telephony	master (rcsuce_Su...	May 22	L					✓		+1	+1				✓			
Remove rcs.publish.state	Merged	Inseob Kim	–	.../system/sepolicy	master	May 15	XS					✓		+1	+1				✗			
[RCS] Add provisioning CTS	Merged	Allen Su	–	platform/cts	master	May 01	S					✓		+1	+1				✓			
[RCS] Add apis to register and unregister UCE publish stat...	Merged	James Lin	Tyler Gunn	.../frameworks/base	master (rcsuce_pr...	Apr 08	M					✓		+1	+1				✓			
[RCS] Add API requestPublication to update capabilities	Merged	James Lin	–	.../net/ims	master (rcsuce_pr...	Apr 08	S					✓		+1	+1				✓			
[RCS] Register receiver to listen to the publish state chan...	Merged	James Lin	–	.../external/ims	master (rcsuce_pr...	Apr 08	M					✓		+1					✓			
[RCS] Integrate the PresencePublisher in UserCapabilityE...	Merged	James Lin	–	.../services/Telephony	master (rcsuce_pr...	Apr 08	L					✓		+1	+1				✓			
Fix CTS failed on ImsServiceTest#testProvisioningManag...	Merged	James Lin	–	platform/cts	master	Mar 05	XS					✓		+1	+1				✓			
Delete RCS related tests for now.	Merged	Amit Mahajan	–	platform/cts	master (Delete Rcs...	Feb 27	L					✓		+1	+1				✓			+
Enhance TelephonyRcsService to only create needed feat...	Merged	Brad Ebinger	–	.../services/Telephony	master (cleanup_u...	Feb 20	L					✓		+1	+1				✓			
Move RCS UCE tests to @Ignore for now	Merged	Brad Ebinger	–	platform/cts	master (rcs_uce_hi...	Feb 12	M					✓		+1	+1				✓			
Refactor TelephonyRcsService to contain RcsFeatureMan...	Merged	Brad Ebinger	–	.../services/Telephony	master (rcs_uce_m...	Feb 08	XL					✓		+1	+1				✓			
Remove RCS dependencies from ImsPhone	Merged	Brad Ebinger	–	.../opt/telephony	master (rcs_uce_m...	Feb 08	L					✓		+1	+1				✓			
Refactor of Telephony dependencies	Merged	Brad Ebinger	–	.../external/ims	master (rcs_uce_m...	Feb 08	S					✓		+1					✓			
ims:rcs uce update Add call composer, standalone chatb...	Merged	Eric Chang	–	.../frameworks/base	master	Feb 04	S		✓			✓		+1	+1				✓		✓	
[RCS]Move PresencePolling to use new @SystemApi	–	Allen Su	–	.../external/ims	master	Feb 04	L							+1								
[RCS] Refactor RcsService to use ImsService API	–	Allen Su	–	.../external/ims	master	Feb 04	L							+1								
[RCS] Implement SIP Option	–	Allen Su	–	.../external/ims	master (option)	Jan 30, 2020	L							+1								
[RCS] Implement SIP Option	–	Allen Su	–	.../net/ims	master (option)	Jan 30, 2020	M							+1	+1							
[RCS] Add SipOptionHandler in TelephonyRcsService	–	Allen Su	–	.../services/Telephony	master (option)	Jan 30, 2020	S							+1	+1							
Add APIs for ImsService RCS UCE implementation	Merged	Brad Ebinger	–	.../frameworks/base	master	Jan 25, 2020	XL					✓			+1				✓			

- google用RCS实现WhatsApp的功能 架构图



自己从头开发

想要实现RCS的应用或服务，也可以从头自己开发。

目前可以借鉴使用的的资源是：网上有个开源项目，可供入手：

- `rcsjta`
  - android-rcs/rcsjta: RCS-e stack for Android with GSMA API
    - https://github.com/android-rcs/rcsjta

rcsjta

- rcsjta
  - 是什么：一个`RCS`的开源项目
  - 最大特点：虽然代码很老，但是至少有源码
  - 作用：可以基于此，借鉴其代码，实现自己的RCS应用或服务
  - GitHub
    - android-rcs/rcsjta: RCS-e stack for Android with GSMA API
      - https://github.com/android-rcs/rcsjta

rcsjta资料

rcsjta有一些资料和文档

- 其中适合入手学习的是
    - Home · android-rcs/rcsjta Wiki
        - https://github.com/android-rcs/rcsjta/wiki
- 其中适合开发入手的是
    - rcsjta/DevEnv.md at wiki · android-rcs/rcsjta
        - https://github.com/android-rcs/rcsjta/blob/wiki/DevEnv.md

RCS协议栈开通服务Provisioning

在能使用RCS服务之前，需要设置基本的参数，才能 `开通服务` = `Provisioning`

可以参考：

rcsjta/Provisioning.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/Provisioning.md

去搞清楚有哪些基本参数，以及分别设置好，才可以进行后续RCS功能演示。

开通服务的模板

Provisioning templates · android-rcs/rcsjta Wiki

https://github.com/android-rcs/rcsjta/wiki/Provisioning-templates

相关协议标准：

- Albatros RCS 5.1 standard
- Blackbird RCS 5.2 standard

此处是手动的manual provisioning：

- Albatros template
    - template-ota_config-Albatros.xml
        - https://github.com/android-rcs/rcsjta/blob/tapi_0.9.0/data/provisioning_templates/albatros/template-ota_config-Albatros.xml
    - 参数
        - ConRef, Private_User_Identity, Public_user_identity_List, Home_network_domain_name, LBO_P-CSCF_Address, Public_user_identity, Realm, UserName, UserPwd, conf-fcty-uri, exploder-uri
- Blackbird template
    - template_config-Blackbird.xml
        - https://github.com/android-rcs/rcsjta/blob/tapi_0.9.0/data/provisioning_templates/blackbird/template_config-Blackbird.xml
    - 参数
        - ConRef, Private_User_Identity, Public_user_identity_List, Public_user_identity, Home_network_domain_name, LBO_P-CSCF_Address, Realm, UserName, UserPwd, ftHTTPCSURI, ftHTTPCSUser, ftHTTPCSPwd, conf-fcty-uri

测试RCS协议栈

如果想要测试RCS协议栈本身，可以参考

rcsjta/TestStack.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/TestStack.md

需要：

- IMS平台
    - 选择：作者有 Telco平台，如果你没有，可以安装第三方IMS平台
        - 用kamailio作为P/I/S-CSCF
            - http://www.kamailio.org/
        - 用bind作为DNS服务器
        - 用FHoSS作为HSS
            - http://www.openimscore.org/docs/FHoSS/index.html
    - 如何设置搭建
        - Setup Kamailio IMS Servers (P-CSCF / I-CSCF / S-CSCF) | LM Tools
            - https://lmtools.com/node/76
        - tutorials:ims:installation-howto [Kamailio SIP Server Wiki]
            - http://www.kamailio.org/wiki/tutorials/ims/installation-howto
- AS=Application Server=应用程序服务器
    - 用于：
        - IM=Instant Message=即时消息 服务
            - 1对1消息
            - 组聊
            - 文件传输
    - 说明
        - 其他服务 不需要AS
            - Capabilities=能力?
            - 视频/图片/地理位置共享
            - MM 会话
    - 如何选择AS?
        - 目前没有开源的AS可用
            - 不过也可以测试1对1聊天和文件传输
                - 原设备设为 active mode=主动模式
                - 目的设备设为 passive mode=被动模式

RCS一些实现细节

部分核心功能实现代码

想要了解部分RCS核心功能的具体实现代码，可以参考

rcsjta/CodeSamples.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/CodeSamples.md

其中给出了演示相关api如何使用的部分核心代码实现。

相关的几种典型的功能：

- 聊天=IM=Chat
- 文件传输=FT=File Transfer
- Video=视频
- Image=图片

- Geoloc=地理位置
- Extensions=扩展（功能）

兼容性测试

此处RCS功能实现方式和演示方式是基于Android的app的

而在Android中实现了RCS功能后，需要进行：兼容性测试

可以参考：

rcsjta/TestHarness.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/TestHarness.md

去操作。

此处的自动化测试套件允许：

- 验证API签名
    - 检测一个设备，是否兼容对应的API（签名）
        - 目前有几种
            - Albatros
            - Blackbird
            - Crane
    - 从贴图的细节看，的确好像是：测试API是否兼容 如果不兼容，会失败，无法通过测试
        - API参考定义 都放在xml中：res/xml/
            - 从别处拷贝的：rcsjta/doclava/reference/current.xml
- 验证内容提供者content providers
    - 验证可操作性（读/写）、多列？等

> 注：
>
> 兼容性测试套件 | Android 开源项目 | Android Open Source Project
>
> 兼容性测试套件 (CTS) 是一个免费的商业级测试套件，可在此处下载。CTS 代表兼容性的“机制”

扩展

官网文档

rcsjta/Extensions.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/Extensions.md

介绍了：

- extension=扩展
    - 起源
        - 此开源项目最开始就引入了extension扩展机制，支持部署新的RCS/IMS服务
    - 现状
        - 目前已被GSMA采用放入RCS 5.2中
            - 详见
                - specification中的RCS Extensibility部分
    - 此处
        - Capability API：声明有哪些扩展
        - Multimedia Session API：实现扩展的功能

安全模型

rcsjta/SecurityModel.md at wiki · android-rcs/rcsjta

https://github.com/android-rcs/rcsjta/blob/wiki/SecurityModel.md

除了本身安卓的sign外，还会有tag的sign等过程，包括安装期间和使用期间，都会有权限校验的。

用于校验的工具：

- PorteCle
    - http://portecle.sourceforge.net
        - Portecle is a user friendly GUI application for creating, managing and examining keystores, keys, certificates, certificate requests, certificate revocation lists and more.
- keytool
    - android JDK中的工具
- iaritool
    - RCS的工具
    - 源码位置：rcsjta/tools/security/tag-auth-util
        - 最新源码：
            - 没有
                -
            - 只有代码中有些iari相关代码
                -
            - 算了，等用到再说。
                - 或许最新android studio中已无需此工具了

证书

Certificates · android-rcs/rcsjta Wiki

https://github.com/android-rcs/rcsjta/wiki/Certificates

- provisioning DM server
- File Transfer content servers of the OrangeLabs integration platforms

自己实现了https的校验证书后，需要安装到安卓设备中

-
    1. Provisioning server
    2. DM server certificate
        - https://github.com/android-rcs/rcsjta/blob/tapi_0.9.0/data/certificates/provisioning/dmserver_ofr.cer
-
    1. File Transfer content servers
    2. NSN 7.2 + RCS8
        - https://github.com/android-rcs/rcsjta/blob/tapi_0.9.0/data/certificates/filetransfer/Cert_RCS8_NSN_7_2.crt
    3. NSN 9.2 + RCS14
        - https://github.com/android-rcs/rcsjta/blob/tapi_0.9.0/data/certificates/filetransfer/CertRCS14NSN92.crt

> 注意：
>
> 虽然这几个文件都是存在的。不过注意到了是旧的tag：0.9.0
>
> 最新版中，没有data目录了，这几个文件都不存在了

rcsjta开发

下面真正开始基于rcsjta开发RCS。

下载代码

```
git clone https://github.com/android-rcs/rcsjta.git
```

crifan版rcsjta

自己基于rcsjta折腾RCS开发期间，基于源码做了很多改动和优化，目的是为了能编译出可演示的apk。

此处贴出来改动后的最新代码供参考：

crifan/rcsjta: RCS-e stack for Android with GSMA API

https://github.com/crifan/rcsjta

准备环境

确保当前系统环境已准备好：

Java

此处安装的版本是： `1.8.0_212`

相关信息：

```
~  java -version
java version "1.8.0_212"
Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
```

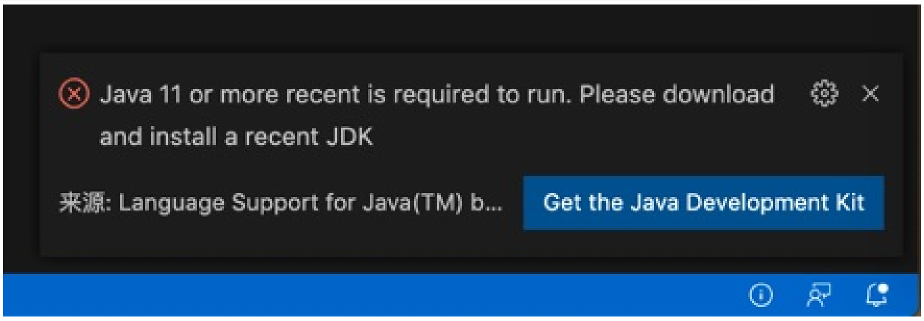
和：

```
/usr/libexec/java_home -V
Matching Java Virtual Machines (1):
    1.8.0_212, x86_64:	"Java SE 8"	/Library/Java/JavaVirtualMachines/jdk1.8.0_212.jdk/Contents/Home

/Library/Java/JavaVirtualMachines/jdk1.8.0_212.jdk/Contents/Home
```

其他说明：

- 记得安装后，把相关环境变量

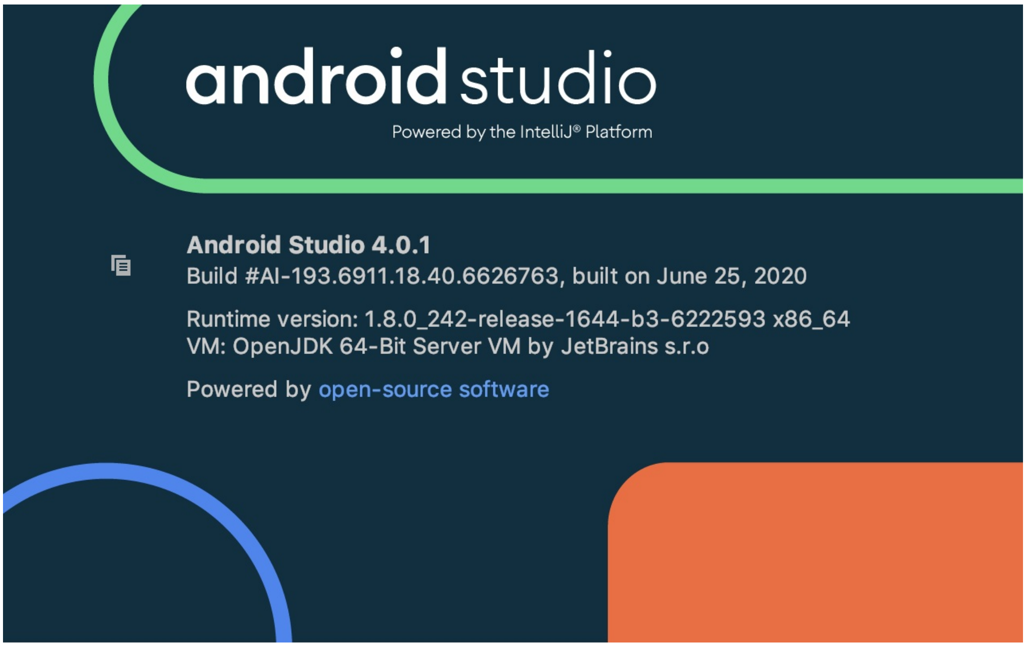
  ```
  export JAVA_HOME=$(/usr/libexec/java_home)
  export PATH=.:$PATH:$JAVA_HOME/bin
  export CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar:.
  ```

  - 加到启动脚本（此处是 `.zhsrc` ）中
- 如果用VSCode打开 `rcsjta` 后会提示
  - `Java 11 or more recent is required to run. Please download and install a recent JDK`
    - 
      Java 11 or more recent is required to run. Please download and install a recent JDK
      来源: Language Support for Java(TM) b...

  - 可以忽略不理会

Android Studio

- 下载
  - 从官网
    - [SDK 平台工具版本说明 | Android 开发者 | Android Developers]
    - 得到安装包
      - 此处下载的是 `Mac` 版 `android-studio-ide-193.6626763-mac.dmg`

        - 正常安装即可
        - 最新更新后版本
            - 

android studio

Powered by the IntelliJ® Platform

Android Studio 4.0.1
Build #AI-193.6911.18.40.6626763, built on June 25, 2020

Runtime version: 1.8.0_242-release-1644-b3-6222593 x86_64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Powered by open-source software

```
Android Studio 4.0.1
Build #AI-193.6911.18.40.6626763, built on June 25, 2020
Runtime version: 1.8.0_242-release-1644-b3-6222593 x86_64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
macOS 10.14.6
GC: ParNew, ConcurrentMarkSweep
Memory: 1981M
Cores: 8
Registry: ide.new.welcome.screen.force true
Non-Bundled Plugins:
```

- 设置环境变量
    - 然后把对应变量

```
export ANDROID_SDK_ROOT=/Users/limao/Library/Android/sdk
PATH=$PATH:$ANDROID_SDK_ROOT/build-tools/29.0.2
PATH=$PATH:$ANDROID_SDK_ROOT/tools:$ANDROID_SDK_ROOT/platform-tools
ANDROID_HOME=$ANDROID_SDK_ROOT
```

    - 加到启动脚本（此处是 `~/.zshrc` ）中
- 下载 `SDK`
    - 然后在 `Android Studio` 中下载对应所需版本的 `SDK`

相关说明：

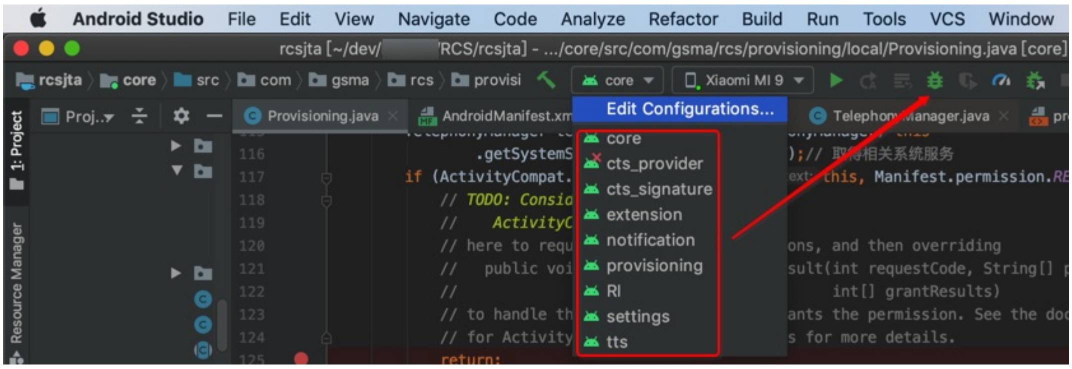
- 还是要加 `ANDROID_HOME`
    - 虽然官网 环境变量 | Android 开发者 | Android Developers已解释的很清楚
    - 对于 设置 SDK 安装目录的路径 的环境变量：
        - 旧的：ANDROID_HOME
            - 已废弃
                - 为了兼容，目前也还支持ANDROID_HOME而已
        - 新的：ANDROID_SDK_ROOT
            - 建议用新的
    - 但是上述还是要设置 `ANDROID_HOME`
    - 否则就会报错 `A problem occurred configuring project SDK location not found`

编译apk

然后去编译rcsjta的项目，生成apk，调试安装到安卓手机中。

编译apk

有2种方式：

- `gradlew` 命令行
- Android Studio中图形界面中点击调试
    - 即可编译和生成apk，并安装到已连接的安卓手机中
        - 

gradlew命令行

查看有哪些任务

```
gradlew tasks
```

输出：

```
------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------


Android tasks
-------------
androidDependencies - Displays the Android dependencies of the project.
signingReport - Displays the signing info for each variant.
sourceSets - Prints out all the source sets defined in this project.


Build tasks
-----------
assemble - Assembles all variants of all applications and secondary packages.
assembleAndroidTest - Assembles all the Test applications.
assembleDebug - Assembles all Debug builds.
assembleRelease - Assembles all Release builds.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
clean - Deletes the build directory.
compileDebugAndroidTestSources
compileDebugSources
compileDebugUnitTestSources
compileReleaseSources
```

```
compileReleaseUnitTestSources
extractDebugAnnotations - Extracts Android annotations for the debug variant into the archive file
extractReleaseAnnotations - Extracts Android annotations for the release variant into the archive file
mockableAndroidJar - Creates a version of android.jar that's suitable for unit tests.


Build Setup tasks
-----------------
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]


Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project 'rcsjta'.
components - Displays the components produced by root project 'rcsjta'. [incubating]
dependencies - Displays all dependencies declared in root project 'rcsjta'.
dependencyInsight - Displays the insight into a specific dependency in root project 'rcsjta'.
help - Displays a help message.
model - Displays the configuration model of root project 'rcsjta'. [incubating]
projects - Displays the sub-projects of root project 'rcsjta'.
properties - Displays the properties of root project 'rcsjta'.
tasks - Displays the tasks runnable from root project 'rcsjta' (some of the displayed tasks may belong to subprojects).


Install tasks
-------------
installDebug - Installs the Debug build.
installDebugAndroidTest - Installs the android (on device) tests for the Debug build.
uninstallAll - Uninstall all applications.
uninstallDebug - Uninstalls the Debug build.
uninstallDebugAndroidTest - Uninstalls the android (on device) tests for the Debug build.
uninstallRelease - Uninstalls the Release build.


Verification tasks
------------------
check - Runs all checks.
connectedAndroidTest - Installs and runs instrumentation tests for all flavors on connected devices.
connectedCheck - Runs all device checks on currently connected devices.
connectedDebugAndroidTest - Installs and runs the tests for debug on connected devices.
deviceAndroidTest - Installs and runs instrumentation tests using all Device Providers.
deviceCheck - Runs all device checks using Device Providers and Test Servers.
lint - Runs lint on all variants.
lintDebug - Runs lint on the Debug build.
lintRelease - Runs lint on the Release build.
test - Run unit tests for all variants.
testDebugUnitTest - Run unit tests for the debug build.
testReleaseUnitTest - Run unit tests for the release build.


Other tasks
-----------
assembleDefault
extractProguardFiles
jarDebugClasses
jarReleaseClasses
javadoc
transformResourcesWithMergeJavaResForDebugUnitTest
transformResourcesWithMergeJavaResForReleaseUnitTest
```

查看子项目

```
gradlew projects
```

输出：

- Root project rcsjta
  - RI

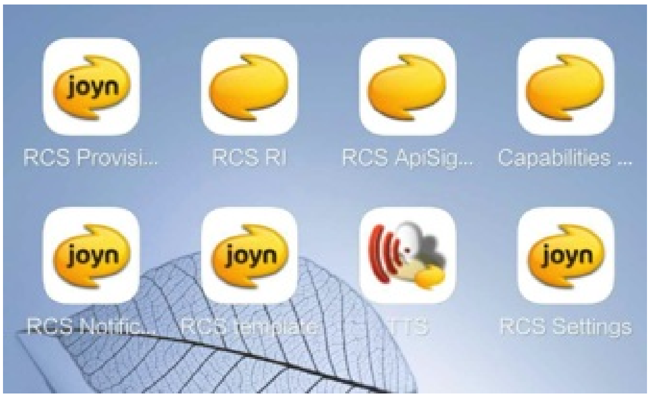
    - api
    - api_cnx
    - bouncycastle
    - core
    - cts_provider
    - cts_signature
    - extension
    - mediaplayer
    - nist_sip
    - notification
    - provisioning
    - settings
    - tts

用命令行编译RCS的apk

- core
    - 编译： `gradlew :core:build`
    - 清理： `gradlew :core:clean`
    - 输出
        - rcsjta/core/build/outputs/apk
            - `RCS_Core-debug.apk`
                - 用于调试期间，安装到安卓手机中
            - `RCS_Core-release-unsigned.apk`
    - 说明
        - 编译core期间还会编译所依赖的其他子项目
            - api
            - bouncycastle
            - nist_sip
- RI
    - = `Reference Implement` =参考实现
    - 编译： `gradlew :RI:build`
    - 清理： `gradlew :RI:clean`
    - 输出
        - rcsjta/RI/build/outputs/apk
            - `RI-debug.apk`
            - `RI-release-unsigned.apk`
- settings
    - 编译： `gradlew :settings:build`
    - 清理： `gradlew :settings:clean`
    - 输出
        - rcsjta/settings/build/outputs/apk
            - `settings-debug.apk`
            - `settings-release-unsigned.apk`

运行apk

然后去把rcsjta编译出的几个apk安装到安卓手机中：

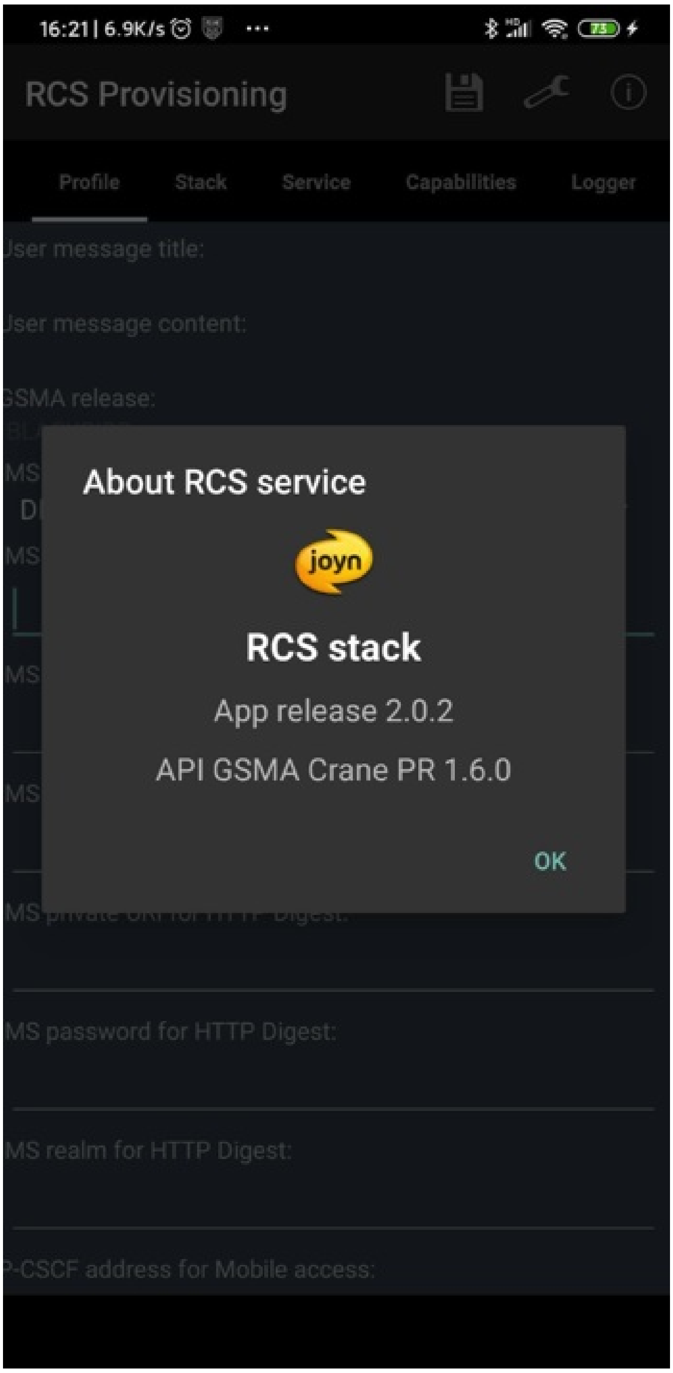


启动看看效果。

core

core=RCS Stack=带Provisioning开通服务相关配置

core编译出apk安装后：



16:21 | 6.9K/s

RCS Provisioning

Jser message title:

Jser message content:

GSMA release:

MS private URI for HTTP Digest:

MS password for HTTP Digest:

MS realm for HTTP Digest:

P-CSCF address for Mobile access:

About RCS service

RCS stack

App release 2.0.2

API GSMA Crane PR 1.6.0

子页面：

- Profile

- Stack

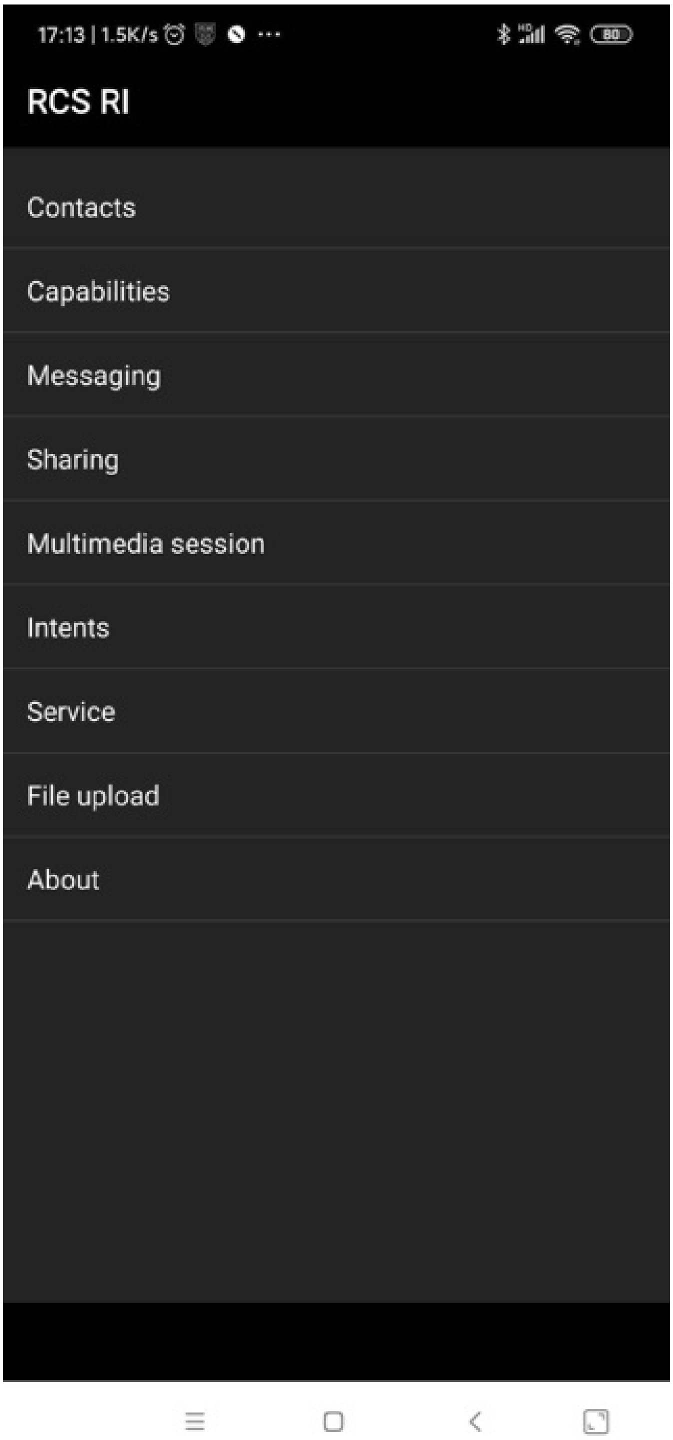
    -
- Service

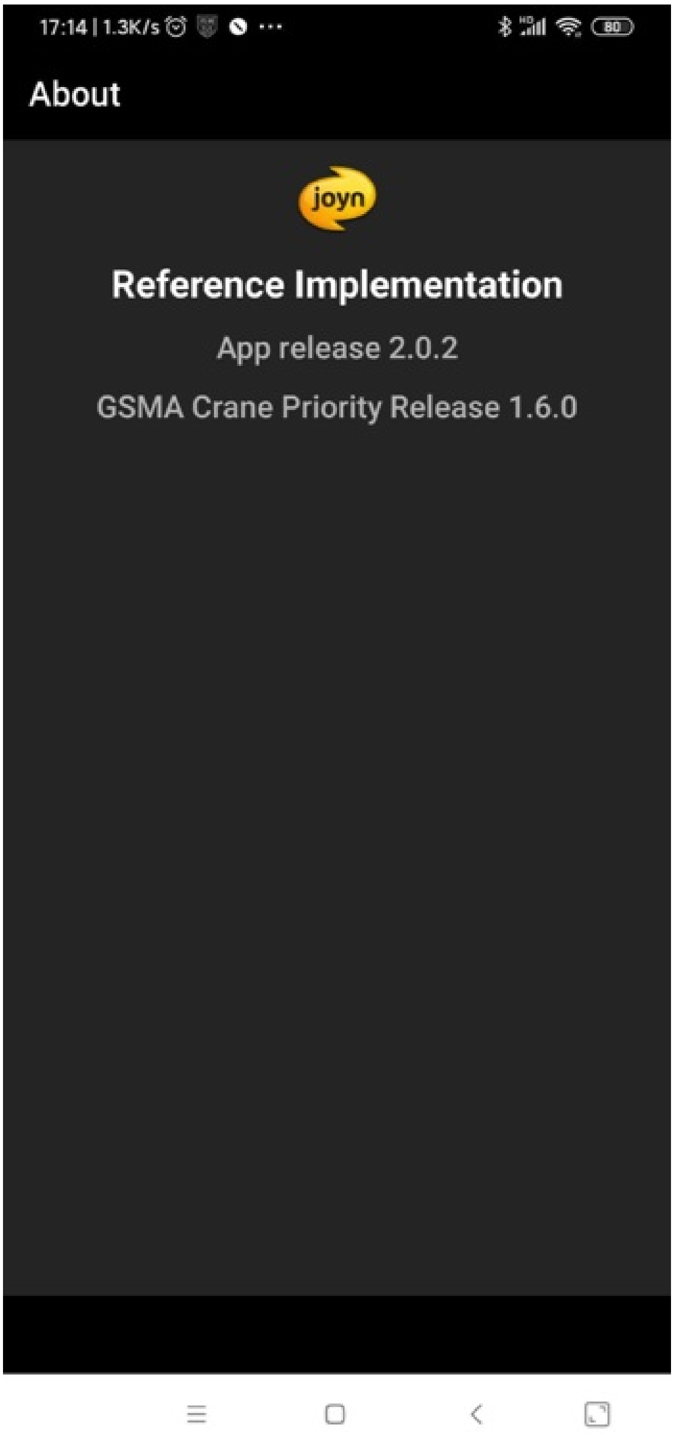
    -
- Capabilities

  -
- Logger

RI

RI编译出apk安装后：


17:13 | 1.5K/s
RCS RI
Contacts
Capabilities
Messaging
Sharing
Multimedia session
Intents
Service
File upload
About



17:14 | 1.3K/s

About

Reference Implementation

App release 2.0.2

GSMA Crane Priority Release 1.6.0

子页面：

- Contacts

- Capabilities

  -

- Messaging

    -
- Sharing

  -
- Multimedia session

    -
- Intents

  -
- Service

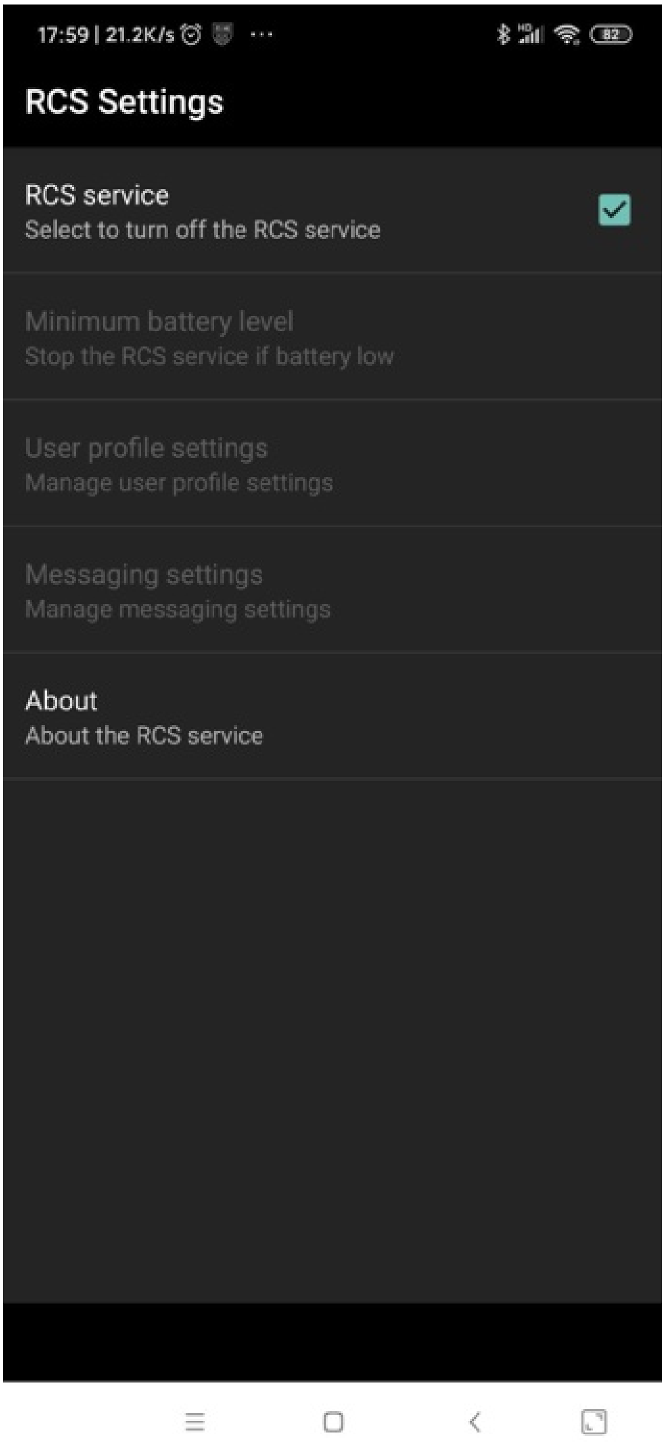
    -

- File upload

◦

settings

settings编译出apk安装后：



RCS Settings

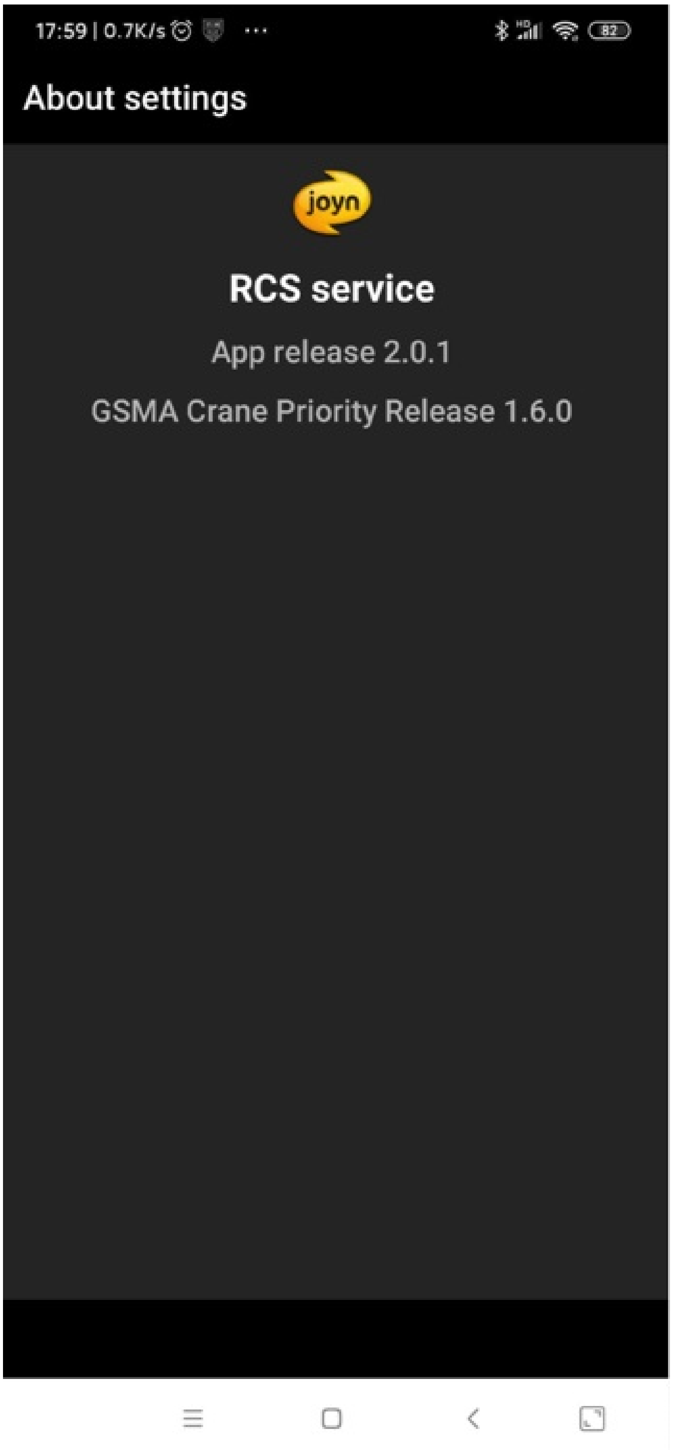
RCS service
Select to turn off the RCS service ☑

Minimum battery level
Stop the RCS service if battery low

User profile settings
Manage user profile settings

Messaging settings
Manage messaging settings

About
About the RCS service



17:59 | 0.7K/s

About settings

RCS service

App release 2.0.1

GSMA Crane Priority Release 1.6.0

版本是：

```
RCS service
App release 2.0.1
GSMA Crane Priority Release 1.6.0
```

演示RCS

已编译安装好RCS的apk后，接着去演示RCS。

在演示之前，需要去 Provisioning开通服务 这个RCS的apk中，设置对应的必要参数，然后才能用于演示RCS服务。

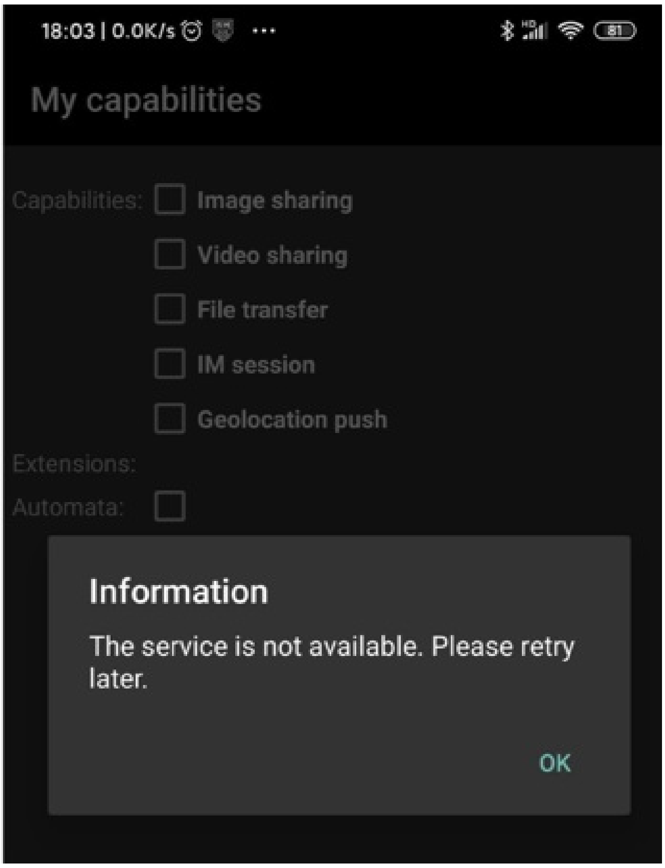
最近进展：没有能够成功的演示RCS的功能，但是搞懂了一些参数配置。

无法演示RCS的功能

暂时还没彻底搞清楚，如何才能让RCS服务正常的运行起来。

导致：上述的几个（core，RI，settings等）apk，尝试点击服务，都会出现：

```
Information
The service is not available Please retry later
```



My capabilities

Capabilities: □ Image sharing
□ Video sharing
□ File transfer
□ IM session
□ Geolocation push

Extensions:

Automata: □

Information

The service is not available. Please retry later.

折腾过很久，解决了各种编译错误和研究了各种代码逻辑，依旧没解决。

需要先配置好核心参数

不过后续搞懂了一些入手的基本逻辑：

要去

- 给 `core` = `RCS Provisioning` 的apk中的各种参数配置好，才（可）能实现RCS服务正常运行

    - 以及可能还需要：额外的`AS` = `Application Server`，才能确保服务可用。

运行RCS是否需要AS?

目前理解是：

此处需要额外的服务器，但（可以不用，不是）AS服务器，而是：`IMS platform` = `IMS平台`

详细解释是：

- 想要测试RCS协议栈
    - 至少需要
        - IMS platform
            - 可以测试功能：
                - Capabilities
                - Video/Image/Geoloc sharing
                - MM session
        - 在没有AS=AS IM的情况下
            - 也可以测试功能
                - 1-1 chat
                - File transfer
    - 只有需要
        - AS=IM的AS
            - 才能（完整）测试
                - 1-1 chat
                - Group chat
                - File transfer
- 如何选择IMS platform
    - 开源产品（组合出IMS platform）
        - kamailio
            - 作为：P/I/S-CSCF
        - bind
            - 作为：DNS server
        - FHoSS=Open IMS Core=OpenIMS=OpenIMSCore=Open Source IMS Core
            - 作为：HSS

Kamailio

有机会再去折腾：

Kamailio SIP Server

tutorials:ims:installation-howto Kamailio SIP Server Wiki

core中的Profile的部分配置

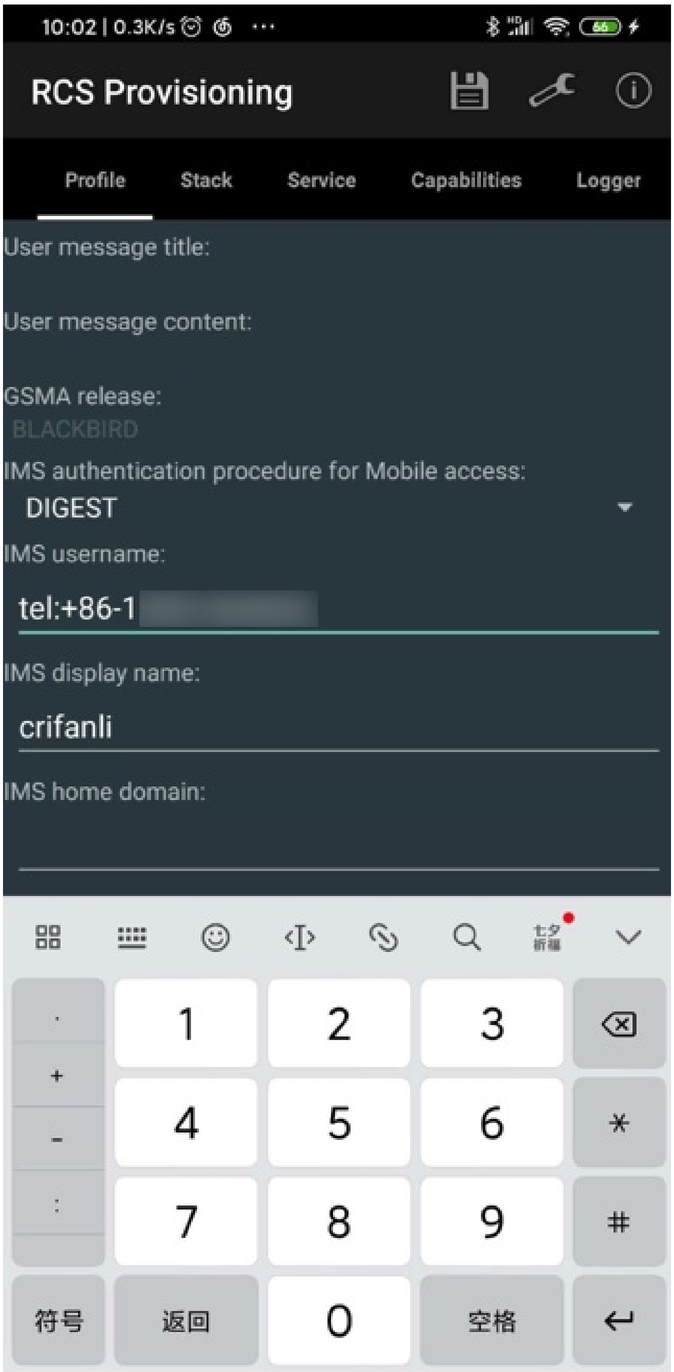
此处贴出`core`，目前已搞懂的参数：

`RCS Provisioning`中的`Profile`中的一些核心参数：

- IMS authentication procedure for Mobile access
    - `DIGEST`
- IMS username
    - `tel:+86-13812345678`
- IMS display name

  - `crifanli`
- IMS home domain
  - `ims.mnc002.mcc460.3gppnetwork.org`

如图：



User message title:

User message content:

GSMA release:
BLACKBIRD

IMS authentication procedure for Mobile access:
DIGEST

IMS username:
tel:+86-1[redacted]

IMS display name:
crifanli

IMS home domain:

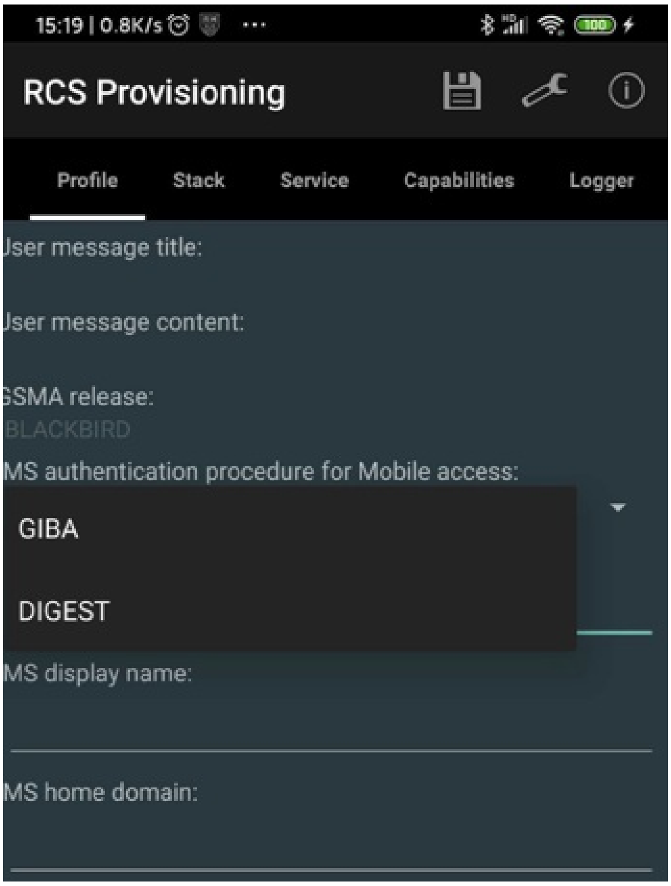
核心参数

此处对于`core`即`RCS Provisioning`中的核心参数，主要是`Profile`中的部分参数，进行详细的解释其含义和计算逻辑。

参数：IMS authentication procedure for Mobile access

此处的`IMS authentication procedure for Mobile access`是2种可能：

- `GIBA`
- `DIGEST`



Jser message title:

Jser message content:

GSMA release:
BLACKBIRD

MS authentication procedure for Mobile access:
GIBA
DIGEST

MS display name:

MS home domain:

下面详细解释背景和如何选择：

IMS认证，属于安全security方面的内容

- Security涉及到
  - IMS authentication
  - SIP confidentiality and integrity
  - `SIP Digest`
  - `SIP Digest with TLS`
  - `GIBA` = `GPRS-IMS-Bundled Authentication`
  - `GAA` = `Generic Authentication Architecture`

其中此处的：

- `DIGEST` = `SIP DIGEST` = `SIP Digest with HTTPS`

    - 不是 `SIP Digest with TLS`
- `GIBA` = `GPRS-IMS-Bundled Authentication`

先直接贴结论：

- RCS的认证，是用的RCS所依赖的IMS的认证
    - RCS认证中是2步认证
        - PS领域(PS domain)的认证 = Network level 认证
        - IMS领域(IMS domain)的认证 = Service Level 认证
    - IMS的认证
        - 有两种
            - `GIBA`
                - 旧手机，比如2G手机，不支持USIM/ISIM的，采用GIBA
                    - 简介实现最基本的账号的安全认证
            - （基于HTTP的，SIP协议的） `DIGEST`
                - 对比：另外一种是（基于HTTPS的，即TLS的，SIP协议的DIGEST）
                - 概述：是最新的认证方式
                    - 最新手机和终端和账号，一般都用这个

而关于IMS中账号认证的详细逻辑和过程，详见：

用户和认证 · 移动网络演化史

参数：IMS username

此处的 `IMS username` = `用户标识` ，是 `IMS` 中用于授权和认证的用户名，相当于账号，

概述：

- IMS系统
    - 用户标识
        - 支持多种方式
            - 其中有IMPI和IMPU，都是一种URI
        - 具体来说
            - 以是数字：Tel URI
                - 举例
                    - `tel: +1-555-123-4567`
            - 也可以是字符标识符SIP URI
                - 举例：
                    - `sip:john.doe@example.com`
            - SIP address ?

详细解释参见：用户和认证 · 移动网络演化史

参数：IMS home domain

此处先是通过安卓9的安卓手机中获取到IMSI：

```
~  adb shell getprop | grep gsm
[gsm.current.phone-type]: [1,1]
[gsm.defaultpdpcontext.active]: [false]
[gsm.network.type]: [Unknown,LTE]
[gsm.operator.alpha]: [中国联通,中国移动]
[gsm.operator.iso-country]: [cn,cn]
[gsm.operator.isroaming]: [false,false]
```

```
[gsm.operator.numeric]: [46001,46000]
[gsm.operator.orig.alpha]: [蜗牛移动,中国移动]
[gsm.sim.operator.alpha]: [中国联通,中国移动]
[gsm.sim.operator.iso-country]: [cn,cn]
[gsm.sim.operator.numeric]: [46009,46002]
[gsm.sim.operator.orig.alpha]: [蜗牛移动,CMCC]
[gsm.sim.state]: [LOADED,LOADED]
[gsm.version.baseband]: [MOLY.LR13.R1.TC8.SP.V1.P70,MOLY.LR13.R1.TC8.SP.V1.P70]
[gsm.version.ril-impl]: [android reference-ril 1.0]
[vendor.gsm.external.sim.timeout]: [13,13]
[vendor.gsm.gcf.testmode]: [0]
[vendor.gsm.modem.vsim.capability]: [2,2]
[vendor.gsm.project.baseband]: [XIAOMI_P0MP9_K85V1_64(LWCTG_CUSTOM)]
[vendor.gsm.ril.ct3g]: [0]
[vendor.gsm.ril.ct3g.2]: [0]
[vendor.gsm.ril.eboot]: [0]
[vendor.gsm.ril.fulluicctype]: [USIM]
[vendor.gsm.ril.fulluicctype.2]: [USIM]
[vendor.gsm.ril.uicc.mccmnc]: [46009]
[vendor.gsm.ril.uicc.mccmnc.1]: [46002]
[vendor.gsm.ril.uicctype]: [USIM]
[vendor.gsm.ril.uicctype.2]: [USIM]
[vendor.gsm.sim.retry.pin1]: [3]
[vendor.gsm.sim.retry.pin1.2]: [3]
[vendor.gsm.sim.retry.pin2]: [3]
[vendor.gsm.sim.retry.pin2.2]: [3]
[vendor.gsm.sim.retry.puk1]: [10]
[vendor.gsm.sim.retry.puk1.2]: [10]
[vendor.gsm.sim.retry.puk2]: [10]
[vendor.gsm.sim.retry.puk2.2]: [10]
[vendor.gsm.sim.ril.phbready]: [true,true]
[vendor.gsm.sim.ril.testsim]: [0]
[vendor.gsm.sim.ril.testsim.2]: [0]
[vendor.gsm.sim.slot.lock.card.valid]: [0]
[vendor.gsm.sim.slot.lock.card.valid.2]: [0]
[vendor.gsm.sim.slot.lock.device.lock.remain.count]: [5]
[vendor.gsm.sim.slot.lock.policy]: [3]
[vendor.gsm.sim.slot.lock.service.capability]: [0]
[vendor.gsm.sim.slot.lock.service.capability.2]: [0]
[vendor.gsm.sim.slot.lock.state]: [0]
[vendor.ril.data.gsm_imsi0]: [460099388822228]
[vendor.ril.data.gsm_imsi1]: [460023514643095]
[vendor.ril.data.gsm_mcc_mnc0]: [46009]
[vendor.ril.data.gsm_mcc_mnc1]: [46002]
[vendor.ril.data.gsm_spn0]: [00808717725B79FB52A8FFFFFFFFFFFFFF]
[vendor.ril.data.gsm_spn1]: [00434D4343FFFFFFFFFFFFFFFFFFFFFFFF]
```

对应详细信息是：

- 手机中有2个插槽，对应2个IMEI?
    - [ro.ril.miui.imei0]: [861119047863804]
    - [ro.ril.miui.imei1]: [861119047863812]

其中：

- 联通的卡，（虚拟）运营商是 蜗牛移动
    - 对应手机卡已经有了
        - [gsm.sim.state]: [LOADED]
    - 是：
        - [gsm.sim.operator.iso-country]: [cn]
            - 中国
    - 的
        - [gsm.sim.operator.alpha]: [中国联通]
            - 对应编号是
                - [gsm.sim.operator.numeric]: [46009]
                    - 另外还有个类似的：

            - [gsm.operator.numeric]: [46001]
      - 等价于
        - [vendor.gsm.ril.uicc.mccmnc]: [46009]
      - 包括两部分
        - MCC：460
          - 指的是：中国
        - MNC：09
          - 指的是：联通
  - 的
    - [gsm.sim.operator.orig.alpha]: [蜗牛移动]
  - 对应的IMSI是
    - [vendor.ril.data.gsm_imsi0]: [460099388822228]
- 自己的中国移动的手机号，运营商是 中国移动
  - [vendor.ril.data.gsm_mcc_mnc1]: [46002]
    - MCC：460
      - 中国
    - MNC：002
      - 中国移动
  - [vendor.ril.data.gsm_imsi1]: [460023514643095]
    - IMSI：460023514643095
    - MCC：460
      - 中国
    - MNC：002
      - 中国移动 CMCC
    - MSIN：3514643095
  - [gsm.network.type]: [Unknown,LTE]
    - gsm.network.type=LTE
      - 网络是LTE网络
  - [gsm.operator.alpha]: [中国联通,中国移动]
    - gsm.operator.alpha=中国移动

此处使用其中一个中国移动的手机号的IMSI：

- IMSI：460023514643095
  - MCC：460
    - 中国
  - MNC：002
    - 中国移动 CMCC
  - MSIN：3514643095

根据用户和认证 · 移动网络演化史 的：

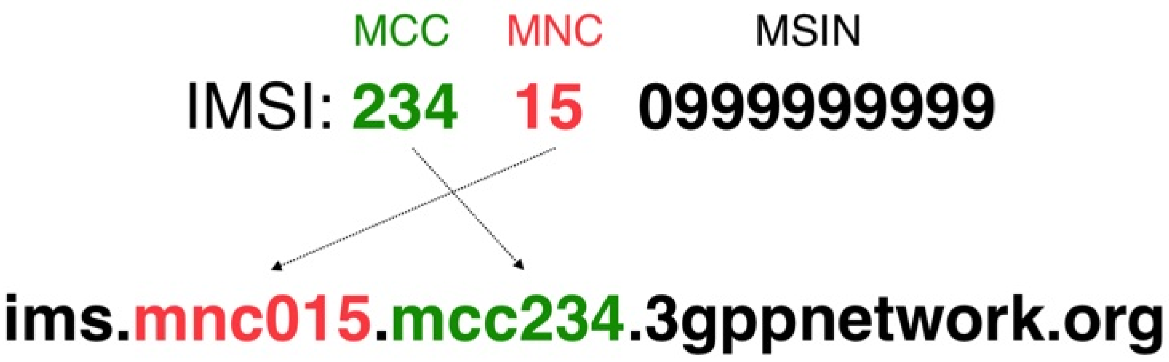


Figure 4 Home network domain name derivation from IMSI

从 `IMSI` 去生成 `IMS home domain` = `IMS home network domain` ：

```
ims.mnc002.mcc460.3gppnetwork.org
```

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-11-04 22:20:11

rcsjta源码分析

折腾rcsjta开发期间，尤其是尝试分析RCS服务为何没有启动 `The service is not available` 期间，研究了相关代码的逻辑，整理如下供参考。

The service is not available 相关代码逻辑

搜：

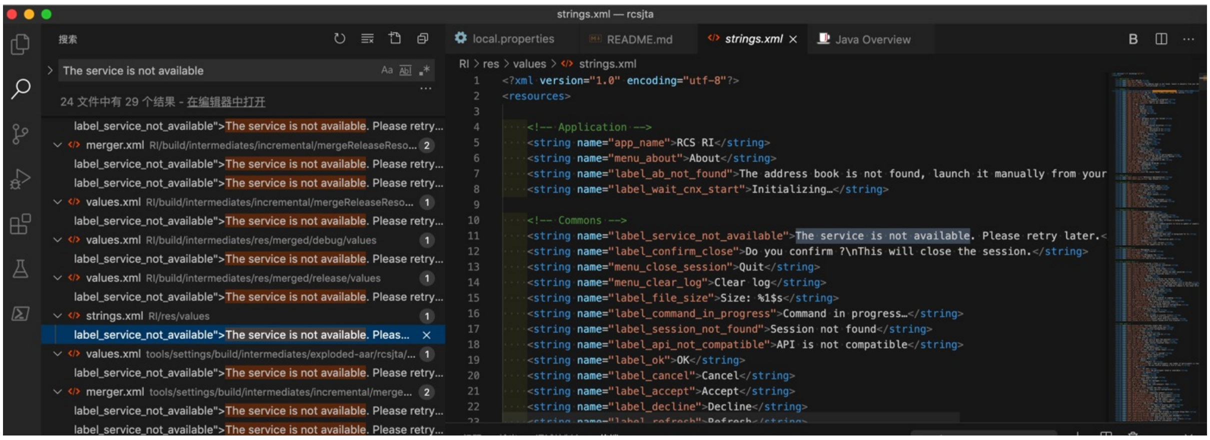
The service is not available

找到很多个。

其中RI中的是：

`rcsjta/RI/res/values/strings.xml`

```
<string name="label_service_not_available">The service is not available. Please retry later.</string>
```



strings.xml — rcsjta

```
The service is not available
24 文件中有 29 个结果 - 在编辑器中打开
    label_service_not_available">The service is not available. Please retry...
  merger.xml RI/build/intermediates/incremental/mergeReleaseReso... 2
    label_service_not_available">The service is not available. Please retry...
    label_service_not_available">The service is not available. Please retry...
  values.xml RI/build/intermediates/incremental/mergeReleaseReso... 1
    label_service_not_available">The service is not available. Please retry...
  values.xml RI/build/intermediates/res/merged/debug/values 1
    label_service_not_available">The service is not available. Please retry...
  values.xml RI/build/intermediates/res/merged/release/values 1
    label_service_not_available">The service is not available. Please retry...
  strings.xml RI/res/values 1
    label_service_not_available">The service is not available. Pleas...
  values.xml tools/settings/build/intermediates/exploded-aar/rcsjta/... 1
    label_service_not_available">The service is not available. Please retry...
  merger.xml tools/settings/build/intermediates/incremental/merge... 2
    label_service_not_available">The service is not available. Please retry...
    label_service_not_available">The service is not available. Please retry...
```

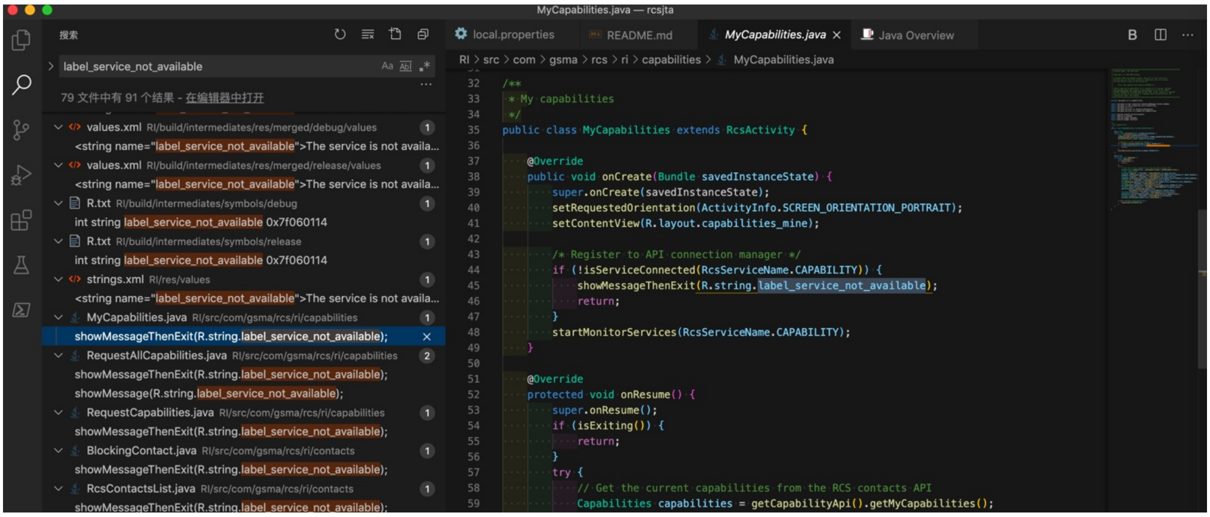
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- Application -->
    <string name="app_name">RCS RI</string>
    <string name="menu_about">About</string>
    <string name="label_ab_not_found">The address book is not found, launch it manually from your
    <string name="label_wait_cnx_start">Initializing…</string>

    <!-- Commons -->
    <string name="label_service_not_available">The service is not available. Please retry later.<
    <string name="label_confirm_close">Do you confirm ?\nThis will close the session.</string>
    <string name="menu_close_session">Quit</string>
    <string name="menu_clear_log">Clear log</string>
    <string name="label_file_size">Size: %1$s</string>
    <string name="label_command_in_progress">Command in progress…</string>
    <string name="label_session_not_found">Session not found</string>
    <string name="label_api_not_compatible">API is not compatible</string>
    <string name="label_ok">OK</string>
    <string name="label_cancel">Cancel</string>
    <string name="label_accept">Accept</string>
    <string name="label_decline">Decline</string>
    <string name="label_refresh">Refresh</string>
```

去找找代码中哪里用到了： `label_service_not_available`

看到了 Capability的对应代码

`rcsjta/RI/src/com/gsma/rcs/ri/capabilities/MyCapabilities.java`

```java
public class MyCapabilities extends RcsActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.capabilities_mine);

        /* Register to API connection manager */
        if (!isServiceConnected(RcsServiceName.CAPABILITY)) {
            showMessageThenExit(R.string.label_service_not_available);
            return;
        }
        startMonitorServices(RcsServiceName.CAPABILITY);
    }
```



MyCapabilities.java — rcsjta
搜索
local.properties
README.md
MyCapabilities.java
Java Overview
RI > src > com > gsma > rcs > ri > capabilities > MyCapabilities.java

```
label_service_not_available
79 文件中有 91 个结果 - 在编辑器中打开
values.xml RI/build/intermediates/res/merged/debug/values 1
    <string name="label_service_not_available">The service is not availa...
values.xml RI/build/intermediates/res/merged/release/values 1
    <string name="label_service_not_available">The service is not availa...
R.txt RI/build/intermediates/symbols/debug 1
    int string label_service_not_available 0x7f060114
R.txt RI/build/intermediates/symbols/release 1
    int string label_service_not_available 0x7f060114
strings.xml RI/res/values 1
    <string name="label_service_not_available">The service is not availa...
MyCapabilities.java RI/src/com/gsma/rcs/ri/capabilities 1
    showMessageThenExit(R.string.label_service_not_available);
RequestAllCapabilities.java RI/src/com/gsma/rcs/ri/capabilities 2
    showMessageThenExit(R.string.label_service_not_available);
    showMessage(R.string.label_service_not_available);
RequestCapabilities.java RI/src/com/gsma/rcs/ri/capabilities 1
    showMessageThenExit(R.string.label_service_not_available);
BlockingContact.java RI/src/com/gsma/rcs/ri/contacts 1
    showMessageThenExit(R.string.label_service_not_available);
RcsContactsList.java RI/src/com/gsma/rcs/ri/contacts 1
    showMessageThenExit(R.string.label_service_not_available);
```

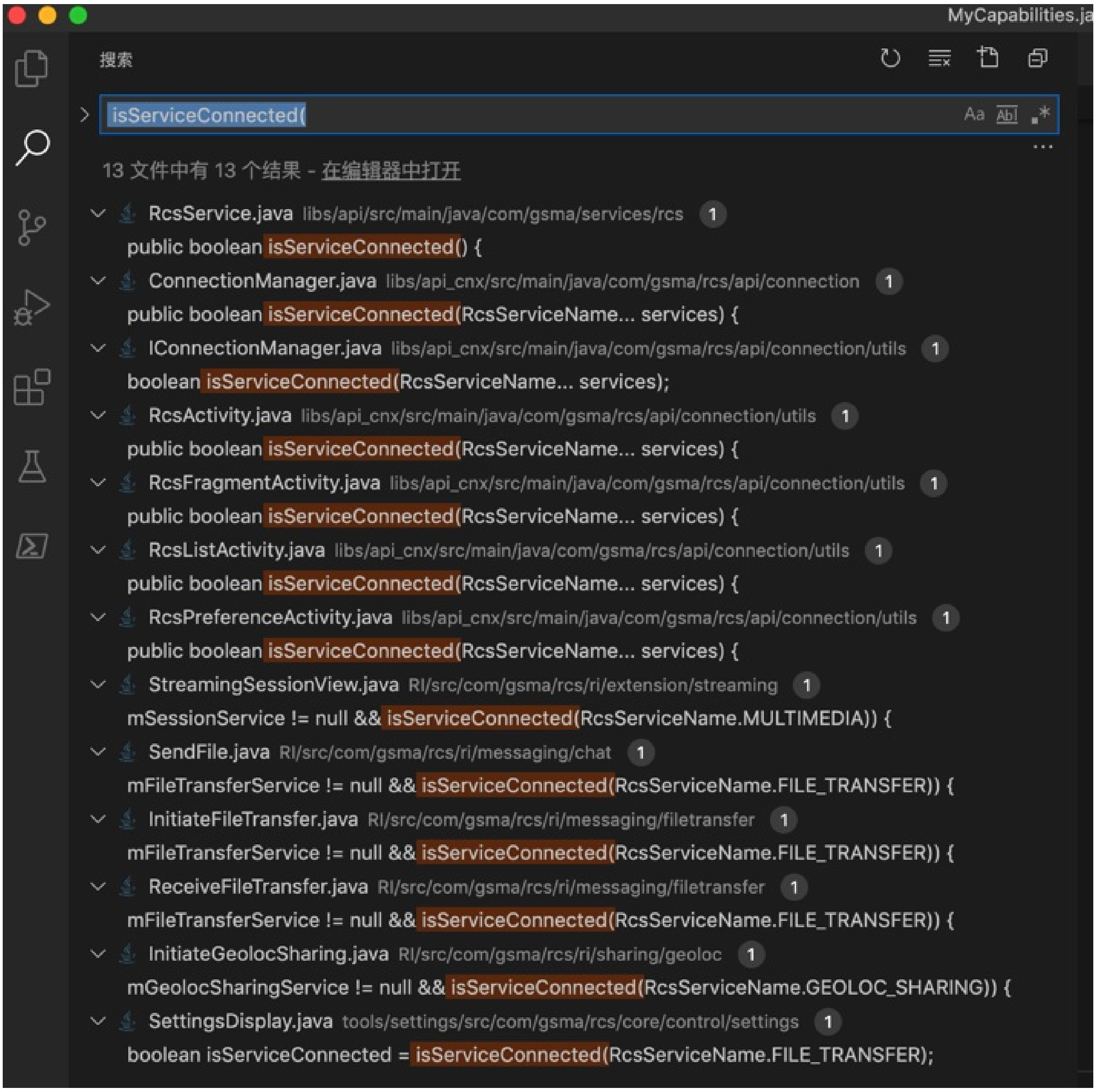
```java
/**
 * My capabilities
 */
public class MyCapabilities extends RcsActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.capabilities_mine);

        /* Register to API connection manager */
        if (!isServiceConnected(RcsServiceName.CAPABILITY)) {
            showMessageThenExit(R.string.label_service_not_available);
            return;
        }
        startMonitorServices(RcsServiceName.CAPABILITY);
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (isExiting()) {
            return;
        }
        try {
            // Get the current capabilities from the RCS contacts API
            Capabilities capabilities = getCapabilityApi().getMyCapabilities();
```

继续去研究代码

搜

`isServiceConnected(`

找到多处定义



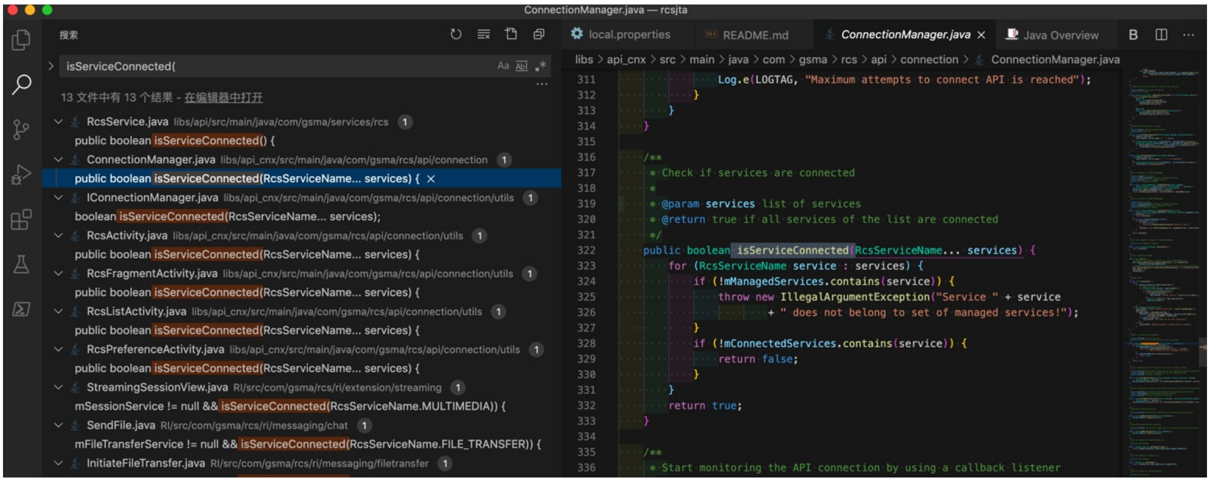
isServiceConnected(

```
RcsService.java libs/api/src/main/java/com/gsma/services/rcs 1
    public boolean isServiceConnected() {
ConnectionManager.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection 1
    public boolean isServiceConnected(RcsServiceName... services) {
IConnectionManager.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    boolean isServiceConnected(RcsServiceName... services);
RcsActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsFragmentActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsListActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsPreferenceActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
StreamingSessionView.java RI/src/com/gsma/rcs/ri/extension/streaming 1
    mSessionService != null && isServiceConnected(RcsServiceName.MULTIMEDIA)) {
SendFile.java RI/src/com/gsma/rcs/ri/messaging/chat 1
    mFileTransferService != null && isServiceConnected(RcsServiceName.FILE_TRANSFER)) {
InitiateFileTransfer.java RI/src/com/gsma/rcs/ri/messaging/filetransfer 1
    mFileTransferService != null && isServiceConnected(RcsServiceName.FILE_TRANSFER)) {
ReceiveFileTransfer.java RI/src/com/gsma/rcs/ri/messaging/filetransfer 1
    mFileTransferService != null && isServiceConnected(RcsServiceName.FILE_TRANSFER)) {
InitiateGeolocSharing.java RI/src/com/gsma/rcs/ri/sharing/geoloc 1
    mGeolocSharingService != null && isServiceConnected(RcsServiceName.GEOLOC_SHARING)) {
SettingsDisplay.java tools/settings/src/com/gsma/rcs/core/control/settings 1
    boolean isServiceConnected = isServiceConnected(RcsServiceName.FILE_TRANSFER);
```

找到了

`rcsjta/libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/ConnectionManager.java`

```
    /**
     * Check if services are connected
     *
     * @param services list of services
     * @return true if all services of the list are connected
     */
    public boolean isServiceConnected(RcsServiceName... services) {
        for (RcsServiceName service : services) {
            if (!mManagedServices.contains(service)) {
                throw new IllegalArgumentException("Service " + service
                        + " does not belong to set of managed services!");
            }
            if (!mConnectedServices.contains(service)) {
                return false;
            }
        }
        return true;
    }
```



ConnectionManager.java — rcsjta
搜索
local.properties
README.md
ConnectionManager.java
Java Overview
libs > api_cnx > src > main > java > com > gsma > rcs > api > connection > ConnectionManager.java

isServiceConnected(

13 文件中有 13 个结果 - 在编辑器中打开

```
RcsService.java libs/api/src/main/java/com/gsma/services/rcs 1
    public boolean isServiceConnected() {
ConnectionManager.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection 1
    public boolean isServiceConnected(RcsServiceName... services) {
IConnectionManager.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    boolean isServiceConnected(RcsServiceName... services);
RcsActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsFragmentActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsListActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
RcsPreferenceActivity.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/utils 1
    public boolean isServiceConnected(RcsServiceName... services) {
StreamingSessionView.java RI/src/com/gsma/rcs/ri/extension/streaming 1
    mSessionService != null && isServiceConnected(RcsServiceName.MULTIMEDIA)) {
SendFile.java RI/src/com/gsma/rcs/ri/messaging/chat 1
    mFileTransferService != null && isServiceConnected(RcsServiceName.FILE_TRANSFER)) {
InitiateFileTransfer.java RI/src/com/gsma/rcs/ri/messaging/filetransfer 1
```

```java
                Log.e(LOGTAG, "Maximum attempts to connect API is reached");
            }
        }
    }

    /**
     * Check if services are connected
     *
     * @param services list of services
     * @return true if all services of the list are connected
     */
    public boolean isServiceConnected(RcsServiceName... services) {
        for (RcsServiceName service : services) {
            if (!mManagedServices.contains(service)) {
                throw new IllegalArgumentException("Service " + service
                        + " does not belong to set of managed services!");
            }
            if (!mConnectedServices.contains(service)) {
                return false;
            }
        }
        return true;
    }

    /**
     * Start monitoring the API connection by using a callback listener
```

其他地方都是调用这个 `ConnectionManager.java` 中的 `isServiceConnected`

且其中的： `mManagedServices` 是开始初始化包含的一个列表：

```java
    /**
     * The set of managed services
     */
    private final Set<RcsServiceName> mManagedServices;

    /**
     * Constructor
     *
     * @param context The context
     * @param managedServices Set of managed services
     * @param rcsServiceControl instance of RcsServiceControl
     */
    private ConnectionManager(Context context, Set<RcsServiceName> managedServices,
            RcsServiceControl rcsServiceControl) {
        mCtx = context;
        mCnxIntent = PendingIntent.getBroadcast(context, 0, new Intent(ACTION_CONNECT), 0);
        mAlarmManager = (AlarmManager) mCtx.getSystemService(Context.ALARM_SERVICE);

        mManagedServices = managedServices;
        mRcsServiceControl = rcsServiceControl;
        /* Construct list of connected services */
        mConnectedServices = new HashSet<>();
        /* Construct list of clients to notify */
        mClientsToNotify = new HashMap<>();
        /* Construct list of APIs */
        mApis = new HashMap<>();

        if (managedServices == null || managedServices.isEmpty()) {
            throw new RuntimeException("Incorrect parameter managedService!");
        }
        /* Instantiate APIs */
        for (RcsServiceName service : mManagedServices) {
            switch (service) {
                case CAPABILITY:
                    mApis.put(RcsServiceName.CAPABILITY, new CapabilityService(context,
                            newRcsServiceListener(RcsServiceName.CAPABILITY)));
                    break;
                case CHAT:
                    mApis.put(RcsServiceName.CHAT, new ChatService(context,
                            newRcsServiceListener(RcsServiceName.CHAT)));
                    break;
                case CONTACT:
                    mApis.put(RcsServiceName.CONTACT, new ContactService(context,
                            newRcsServiceListener(RcsServiceName.CONTACT)));
                    break;
                case FILE_TRANSFER:
                    mApis.put(RcsServiceName.FILE_TRANSFER, new FileTransferService(context,
                            newRcsServiceListener(RcsServiceName.FILE_TRANSFER)));
                    break;
            case FILE_UPLOAD:
                mApis.put(RcsServiceName.FILE_UPLOAD, new FileUploadService(context,
                        newRcsServiceListener(RcsServiceName.FILE_UPLOAD)));
                break;
            case GEOLOC_SHARING:
                mApis.put(RcsServiceName.GEOLOC_SHARING, new GeolocSharingService(context,
                        newRcsServiceListener(RcsServiceName.GEOLOC_SHARING)));
                break;
            case HISTORY:
                mApis.put(RcsServiceName.HISTORY, new HistoryService(context,
                        newRcsServiceListener(RcsServiceName.HISTORY)));
                break;
            case IMAGE_SHARING:
                mApis.put(RcsServiceName.IMAGE_SHARING, new ImageSharingService(context,
                        newRcsServiceListener(RcsServiceName.IMAGE_SHARING)));
                break;
            case MULTIMEDIA:
                mApis.put(RcsServiceName.MULTIMEDIA, new MultimediaSessionService(context,
                        newRcsServiceListener(RcsServiceName.MULTIMEDIA)));
                break;
            case VIDEO_SHARING:
                mApis.put(RcsServiceName.VIDEO_SHARING, new VideoSharingService(context,
                        newRcsServiceListener(RcsServiceName.VIDEO_SHARING)));
                break;
        }
    }
}
```

其中容易看出是各个功能和服务项目：

- CAPABILITY
- CHAT
- CONTACT
- FILE_TRANSFER
- FILE_UPLOAD
- GEOLOC_SHARING
- HISTORY
- IMAGE_SHARING
- MULTIMEDIA
- VIDEO_SHARING

再去看看 `mConnectedServices`

```
/**
 * Set of connected services
 */
private final Set<RcsServiceName> mConnectedServices;

    mConnectedServices = new HashSet<>();

/**
 * Create a RCS service listener to monitor API connection
 *
 * @param service the service to monitor
 * @return the listener
 */
private RcsServiceListener newRcsServiceListener(final RcsServiceName service) {
    return new RcsServiceListener() {
        @Override
        public void onServiceConnected() {
            mConnectedServices.add(service);
            notifyConnection(service);
        }

        @Override
        public void onServiceDisconnected(ReasonCode error) {
            mConnectedServices.remove(service);
            notifyDisconnection(service, error);
            }
        };

    }

        public void notifyConnection() {
            if (mListener == null) {
                return;
            }
            if (mConnectedServices.containsAll(mMonitoredServices)) {
                /* All monitored services are connected -> notify connection */
                mListener.onServiceConnected();
                mRetryCount = 0;
            }
        }
```

其中的： `newRcsServiceListener` ，就是前面调用的。

不过注意到开始的 `ConnectionManager()` 初始化期间，是根据传入的 `mManagedServices` 去初始化对应服务的。所以要再去找找有哪些地方用到了：

`ConnectionManager(`

发现只有自己

`libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/ConnectionManager.java`

```
    /**
     * Get an instance of ConnectionManager.
     *
     * @param ctx the context
     * @param rcsServiceControl instance of RcsServiceControl
     * @param managedServices Set of managed services
     * @return the singleton instance.
     */
    public static ConnectionManager createInstance(Context ctx,
            RcsServiceControl rcsServiceControl, Set<RcsServiceName> managedServices) {
        if (sInstance != null) {
            return sInstance;
        }
        synchronized (ConnectionManager.class) {
            if (sInstance == null) {
                if (ctx == null) {
                    throw new IllegalArgumentException("Context is null");
                }
                sInstance = new ConnectionManager(ctx, managedServices, rcsServiceControl);
            }
        }
        return sInstance;
    }
```

继续研究原因。

继续找：

`createInstance(`

找到：

`libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/ConnectionManager.java`

```
    /**
     * Get an instance of ConnectionManager.
     *
     * @param context the context
     * @param rcsServiceControl instance of RcsServiceControl
     * @param services list of managed services
     * @return the singleton instance.
     */
    public static ConnectionManager createInstance(Context context,
            RcsServiceControl rcsServiceControl, RcsServiceName... services) {
        Set<RcsServiceName> managedServices = new HashSet<>();
        Collections.addAll(managedServices, services);
        return createInstance(context, rcsServiceControl, managedServices);
    }
```

以及真正的调用去初始化：

`RI/src/com/gsma/rcs/ri/RiApplication.java`

```
    @Override
    public void onCreate() {
        super.onCreate();
...

        mRcsServiceControl = RcsServiceControl.getInstance(mContext);

        /* Starts the RCS service notification manager */
        startService(new Intent(this, RcsServiceNotifManager.class));

        /* Do not execute the ConnectionManager on the main thread */
        Handler mainThreadHandler = new Handler(Looper.getMainLooper());
        final ConnectionManager cnxManager = ConnectionManager.createInstance(mContext,
                mRcsServiceControl, EnumSet.allOf(RcsServiceName.class));
        mainThreadHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                try {
                    cnxManager.start();
                    sCnxManagerStarted = true;

                } catch (RuntimeException e) {
                    Log.e(LOGTAG, "Failed to start connection manager!", e);
                }
            }
        }, DELAY_FOR_STARTING_CNX_MANAGER);
    }
```

和：

`tools/settings/src/com/gsma/rcs/core/control/CoreControlApplication.java`

```
public class CoreControlApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        Context context = getApplicationContext();
        mRcsServiceControl = RcsServiceControl.getInstance(context);

        final ConnectionManager cnxManager = ConnectionManager.createInstance(context,
                mRcsServiceControl, RcsServiceName.FILE_TRANSFER, RcsServiceName.CHAT,
                RcsServiceName.CONTACT);

        /* Do not execute the ConnectionManager on the main thread */
        Handler mainThreadHandler = new Handler(Looper.getMainLooper());

        mainThreadHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                cnxManager.start();
                sCnxManagerStarted = true;
            }
        }, DELAY_FOR_STARTING_CNX_MANAGER);
    }
```

此处很明显 `Core` 的 `control` 的 `application` 中，最开始启动时，只初始化了：

- FILE_TRANSFER

- CHAT
- CONTACT

并没有初始化其他的服务，比如Capability之类的，不知道为何？

去搜索其他的 `RcsServiceName.CAPABILITY` ，能找到其他的，但是没有找到启用的相关逻辑

所以难怪app中没看到服务启动

那如何启动其他如Capability的服务？

直接修改代码去加上？

感觉好像不太对啊，应该是哪里有配置，最开始启动哪些服务才对

去 `Core` 的 `app` 中看看

不过才注意到：

`RI/src/com/gsma/rcs/ri/RiApplication.java`

中的：

```
final ConnectionManager cnxManager = ConnectionManager.createInstance(mContext,
                mRcsServiceControl, EnumSet.allOf(RcsServiceName.class));
```

就是启动了RcsServiceName的所有的类名

就是启动了所有的服务才对

即RI的参考实现中，最开始是去启动了所有的服务的

对于 `RcsServiceName` 的定义，找了半天终于通过搜：

`package com.gsma.rcs.api`

间接的，最后找到了位置，是在api_cnx中的：

`libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/ConnectionManager.java`

而以为能找到原始定义：

```
        /* Instantiate APIs */
        for (RcsServiceName service : mManagedServices) {
            switch (service) {
                case CAPABILITY:
                    mApis.put(RcsServiceName.CAPABILITY, new CapabilityService(context,
                            newRcsServiceListener(RcsServiceName.CAPABILITY)));
                    break;
                case CHAT:
                    mApis.put(RcsServiceName.CHAT, new ChatService(context,
                            newRcsServiceListener(RcsServiceName.CHAT)));
                    break;
                case CONTACT:
                    mApis.put(RcsServiceName.CONTACT, new ContactService(context,
                            newRcsServiceListener(RcsServiceName.CONTACT)));
                    break;
                case FILE_TRANSFER:
                    mApis.put(RcsServiceName.FILE_TRANSFER, new FileTransferService(context,
                            newRcsServiceListener(RcsServiceName.FILE_TRANSFER)));
                    break;
                case FILE_UPLOAD:
                    mApis.put(RcsServiceName.FILE_UPLOAD, new FileUploadService(context,
                            newRcsServiceListener(RcsServiceName.FILE_UPLOAD)));
                    break;
                case GEOLOC_SHARING:
                    mApis.put(RcsServiceName.GEOLOC_SHARING, new GeolocSharingService(context,
                            newRcsServiceListener(RcsServiceName.GEOLOC_SHARING)));
                    break;
        case HISTORY:
          mApis.put(RcsServiceName.HISTORY, new HistoryService(context,
              newRcsServiceListener(RcsServiceName.HISTORY)));
          break;
        case IMAGE_SHARING:
          mApis.put(RcsServiceName.IMAGE_SHARING, new ImageSharingService(context,
              newRcsServiceListener(RcsServiceName.IMAGE_SHARING)));
          break;
        case MULTIMEDIA:
          mApis.put(RcsServiceName.MULTIMEDIA, new MultimediaSessionService(context,
              newRcsServiceListener(RcsServiceName.MULTIMEDIA)));
          break;
        case VIDEO_SHARING:
          mApis.put(RcsServiceName.VIDEO_SHARING, new VideoSharingService(context,
              newRcsServiceListener(RcsServiceName.VIDEO_SHARING)));
          break;
      }
```

还是没有直接定义。

然后去找最原始的定义。通过 `VIDEO_SHARING` 找到：

`core/src/com/gsma/rcs/provider/sharing/VideoSharingProvider.java`

```
  private static final class UriType {

    private static final class VideoSharing {
      private static final int VIDEO_SHARING = 1;

      private static final int VIDEO_SHARING_WITH_ID = 2;
    }

    private static final class InternalVideoSharing {
      private static final int VIDEO_SHARING = 3;

      private static final int VIDEO_SHARING_WITH_ID = 4;
    }
  }
```

看来是不同的类和实现中有最原始的定义，且是 `int` 值

不过后来终于找到了定义：

`libs/api_cnx/src/main/java/com/gsma/rcs/api/connection/ConnectionManager.java`

```
  /**
   * Enumerated type for RCS service name
   */
  @SuppressWarnings("javadoc")
  public enum RcsServiceName {
    CAPABILITY, CONTACT, CHAT, FILE_TRANSFER, IMAGE_SHARING, VIDEO_SHARING, GEOLOC_SHARING, FILE_UPLOAD, MULTIMEDIA, H
ISTORY
  }
```

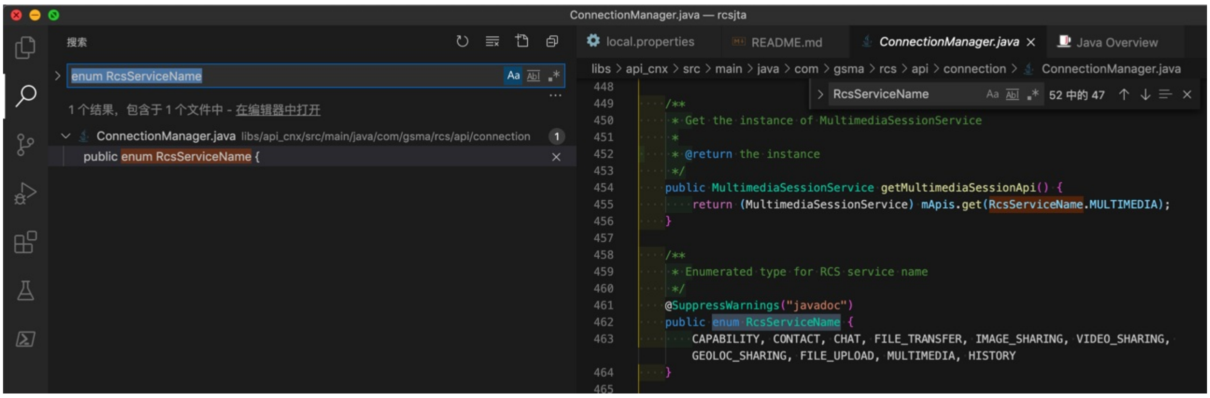
即：该文件中是有定义的。

不是在每个具体实现中定义的。前面理解有误。

反推再去搜：

`enum RcsServiceName`

只能找到此处这一处：


ConnectionManager.java — rcsjta
搜索
enum RcsServiceName
Aa
1 个结果，包含于 1 个文件中 - 在编辑器中打开
ConnectionManager.java libs/api_cnx/src/main/java/com/gsma/rcs/api/connection
public enum RcsServiceName {
local.properties
README.md
ConnectionManager.java
Java Overview
libs > api_cnx > src > main > java > com > gsma > rcs > api > connection > ConnectionManager.java
RcsServiceName
52 中的 47

```
    /**
     * Get the instance of MultimediaSessionService
     *
     * @return the instance
     */
    public MultimediaSessionService getMultimediaSessionApi() {
        return (MultimediaSessionService) mApis.get(RcsServiceName.MULTIMEDIA);
    }

    /**
     * Enumerated type for RCS service name
     */
    @SuppressWarnings("javadoc")
    public enum RcsServiceName {
        CAPABILITY, CONTACT, CHAT, FILE_TRANSFER, IMAGE_SHARING, VIDEO_SHARING,
        GEOLOC_SHARING, FILE_UPLOAD, MULTIMEDIA, HISTORY
    }
```

也是对的。

至此基本上确定，代码逻辑上来说，应该就是：

`RI/src/com/gsma/rcs/ri/RiApplication.java`

```
    /* Do not execute the ConnectionManager on the main thread */
    Handler mainThreadHandler = new Handler(Looper.getMainLooper());
    final ConnectionManager cnxManager = ConnectionManager.createInstance(mContext,
            mRcsServiceControl, EnumSet.allOf(RcsServiceName.class));
    mainThreadHandler.postDelayed(new Runnable() {
        @Override
        public void run() {
            try {
                cnxManager.start();
                sCnxManagerStarted = true;

            } catch (RuntimeException e) {
                Log.e(LOGTAG, "Failed to start connection manager!", e);
            }
        }
    }, DELAY_FOR_STARTING_CNX_MANAGER);
  }
```

这句：

```
final ConnectionManager cnxManager = ConnectionManager.createInstance(mContext,
        mRcsServiceControl, EnumSet.allOf(RcsServiceName.class));
```

去启动了RCS所有的服务。

至于后续为何服务还是没启动，则需要后续深入调研其他方面，才可能找到原因。

附录

下面列出相关参考资料。

名词术语

- RCS相关
    - `RBM` = `RCS Business Messaging`
        - RCS在B2C方面的实现
    - `A2P` = `Application to Person`
        - 对应：消费领域名词： `B2C` = `Business to Customer`
    - `MaaP` = `Messaging as a Platform` = `消息即平台` = `MaaP平台`
        - 旨在构建开放标准的运营商消息平台生态系统，将现有行业短彩信升级到 `RCS` 消息，使用户可以在消息窗口内完成搜索、交互、支付等一站式的业务体验
        - 5G消息系统实现点与应用间消息的体系架构
        - 提供Chatbot的接入、控制，以及与5G消息系统的对接，主要提供Chatbot接入管理、消息审核、消息控制、能力发现、内容存储、垃圾消息投诉、垃圾消息屏蔽等功能
        - MaaP架构下应用的实现形态即 `Chatbot`
    - `Chatbot` = `聊天机器人`
        - 通过会话交互形式为个人用户提供服务的应用形态
    - `5G消息`
        - 短信业务的升级，面向用户提供增强的消息服务，能够为用户提供文本、图片、音频、视频、位置、联系人（vCard）等媒体内容的发送和接收
    - `5G消息终端`
        - 基于终端原生短信入口实现5G消息业务的终端，可以是4G或5G终端
    - `5G消息用户在线`
        - `5G消息用户在线` 是指终端具备数据连接且成功完成在5G消息系统的注册，其它状态统一称为“5G消息用户不在线”
    - `Chatbot服务方`
        - 以Chatbot形态向个人用户提供服务的客户，包括行业客户、梦网客户、公共应急通知服务等
- 通用
    - `XML` = `eXtensible Markup Language`
    - `B2B` = `Business-to-Business`
    - `API` = `Application Programming Interface`
    - `REST` = `Representational State Transfer`
    - `SME` = `Small and Medium Enterprise`
- 网络领域
    - `APN` = `Access Point Name`
    - `CBF` = `Content-Based Filtering`
    - `CF` = `Collaborative Filtering`
    - `DHCP` = `Dynamic Host Configuration Protocol`
    - `DNS` = `Domain Name System`
    - `HDVC` = `High Definition Video Conference`
    - `HTTP` = `Hyper-Text Transfer Protocol`
    - `IP` = `Internet Protocol`
    - `IR` = `Information Retrieval`
    - `LBS` = `Location Based Services`
    - 应用层协议
        - `MAP` = `Mobile Application Part`
            - 另： `MAP` = `Mobile Application Protocol`
        - `CAP` = `Camel Application Protocol`
        - `SGsAP` = `SGs Application Part`
    - `P2P` = `Peer-to-Peer`

  - `PSTN` = `Public Switched Telephone Network`
  - `RAVEL` = `Roaming Architecture for Voice over LTE with Local Breakout`
  - `SPI` = `Social Presence Information`
  - `SMTP` = `Simple Mail Transfer Protocol`
  - `URI` = `Uniform Resource Identifier`
  - `URL` = `Uniform Resource Locator`
  - `UDP` = `User Datagram Protocol`
  - `VoIP` = `Voice over IP`
  - `WebRTC` = `Web Real-Time Communication`
- 电信领域
  - 网络
    - `MSRP` = `Message Session Relay Protocol` = `消息会话中继协议`
    - `MO` = `Mobile Originating`
    - `MT` = `Mobile Terminating`
    - `RAN` = `Radio Access Network` = `无线电接入网`
      - `UTRAN=UMTS Terrestrial Radio Access Network`
      - `GERAN` = `GSM/EDGE Radio Access Network`
    - `2G`
      - `GSM` = `Global System for Mobile telecommunications`
    - `3G`
      - `HNB` = `Home Node B`
      - `VLR` = `Visitor Location Register`
      - `eHRPD` = `evolved High Rate Packet Data`
      - `E-UTRAN` = `Evolved UTRAN` = `Evolved Universal Terrestrial Radio Access Network`
      - `EDGE` = `Enhanced DGE` = `Enhanced Data rates for GSM Evolution`
    - `4G`
      - `LTE` = `Long Term Evolution` = `Long Term Evolution (of RAN)`
        - `LTE-SAE` = `LTE-System Architecture Evolution`
      - `eNB` = `eNodeB`
      - `EPC` = `Evolved Packet Core`
      - `UMTS` = `Universal Mobile Telecommunications System` = `通用移动通讯系统`
    - `5G`
      - `gNB` = `gNodeB` = `next Generation NodeB`
    - `SS7` = `Signaling System No.7`
    - `SMPP` = `Short Message Peer-to-Peer`
    - `ATGW` = `Access Transfer Gateway`
    - `CN` = `Core Network`
    - `CAPEX` = `Capital Expenses`
    - `DTM` = `Dual Transfer Mode`
    - `EPS` = `Evolved Packet System`
    - `ENUM` = `E.164 Number Mapping`
    - `EATF` = `Emergency Access Transfer Function`
    - `VoIMS` = `Voice & video over IMS`
      - (includes IR.92, IR.94 and IR.51)
    - `GSN` = `GPRS Support Node`
      - `GGSN` = `Gateway GPRS Support Node`
      - `SGSN` = `Serving GPRS Support Node`
    - `GRE` = `Generic Routing Encapsulation`
    - `GRX` = `GPRS Roaming eXchange` = `General Packet Radio Service Roaming eXchange`
    - `HDVC` = `High Definition Video Conference`
    - `iFC` = `Initial Filter Criteria`

  - `IM-SSF = IP Multimedia - Service Switching Functionality`
  - `IPX = IP eXchange = IP Packet eXchange = IP数据包交换`
  - `MMTel = Multimedia Telephony`
  - `GTT = Global Text Telephony`
  - `MS = Mobile Station`
  - `MSC = Mobile Switching Centre`
  - `MSISDN = MS-ISDN = Mobile Subscriber ISDN Number`
  - `NAPTR = Naming Authority Pointer DNS Resource Record`
  - `OAM = Operation, Administration and Maintenance`
  - `OMR = Optimal Media Routing`
  - `OPEX = Operational Expenses`
  - `OSA = Open Service Access`
  - `Control Function = 控制功能（组）`
    - `PCF = Policy Control Function`
    - `ATCF = Access Transfer Control Function`
    - `BGCF`
    - `MGCF`
    - `CSCF`
  - `PCC = Policy and Charging Control`
  - `PDP`
    - 2种缩写
      - `PDP = Packet Data Protocol`
      - `PDP = Policy Decision Point`
  - `PDU = Protocol Data Unit`
  - `PoC = Push-to-talk over Cellular = 随按即说`
    - 常称为：
      - `手机对讲服务`
    - 对应设备：
      - `对讲机`
  - `RAT = Radio Access Technology`
  - `RAU = Routing Area Update`
  - `Radio Access = 无线电访问`
    - `LTE`
    - `HSPA = High Speed Packet Access`
  - 音视频传输
    - `RTP = Real-time Transport Protocol = Real Time Protocol`
    - `RTCP = RTP Control Protocol`
    - `QoS = Quality of Service`
      - `QCI = QoS Class Indicator = Quality of Service Class Indicator`
    - `GBR = Guaranteed Bit Rate`
    - `SCTP = Stream Control Transport Protocol`
  - `SIGCOMP = SIGnalling COMPression`
  - `TAP3 = Transferred Account Procedure version 3`
  - `TRF = Transit and Roaming Function`
  - `T-ADS = Terminating Access Domain Selection`
  - `TAU = Tracking Area Update`
- 发送内容
  - 传统的： `SMS = Short Message Service = 普通短信(服务)`
    - `SMSC = Short Message Service Center`
  - 新的： `MMS = Multimedia Messaging Service = 多媒体消息 = 彩信` ？
  - 最新： `IMS`

      - 包含
        - 语音电话(voice call)
          - 涉及到 数据交换
            - `PS/CS GW` = `PS/CS GateWay` = `数据包交换 / 电路交换 网关`
              - 新技术： `PS` = `Packet Switched` = `数据包交换`
              - 旧技术： `CS` = `Circuit Switched` = `电路交换`
  - SIM卡
    - `SIM` = `Subscriber Identity Module` = `客户识别模块` = `（订阅）用户身份模块`
      - 两种SIM
        - `ISIM` = `IMS SIM` = `Internet Protocol Multimedia Services Subscriber Identity Module`
        - `USIM` = `Universal SIM` = `Universal Subscriber Identity Module`
    - 相关
      - `UE` = `User Equipment`
        - 指的是：a wireless device
      - `MSISDN` = `Mobile Subscriber Integrated Services Digital Network Number`
      - `MME` = `Mobility Management Entity`
  - `OTA` = `Over The Air` =通过空气
    - -> 通过无线方式去（升级版本，获取配置等相关技术）
      - 举例
        - requiring over the air provisioning
  - 不同利益方
    - `SP` = `Service Provider` = `服务提供商`
    - `ISV` = `Independent Software Vendor`
    - `MNO` = `Mobile Network Operator` = `移动网络运营商`
    - `MVNO` = `Mobile Virtual Network Operator` = `（移动）虚拟网络运营商=虚拟运营商`
    - `MVNA` = `Mobile Virtual Network Aggregator`
    - `MVNE` = `Mobile Virtual Network Enabler`
- 组织/机构
  - `OMA`
    - `OMA` = `Open Mobile Alliance` = `开放移动联盟`
    - `DM` = `Device Management` = `设备管理`
    - `CPM` = `Converged IP Messaging` = `Converged Internet Protocol Messaging` = `融合IP消息` = `聚合IP消息`
    - 协议规范简称
      - `CPMCONVFUNC` = `CPM Conversation Functions specification`
  - 3GPP
    - `3GPP` = `3rd Generation Partnership Project`
    - `CSFB` = `Circuit Switched Fall-Back`
    - 会话相关
      - `SIP` = `Session Initiation Protocol`
      - `SDP` = `Session Description Protocol`
    - Visited域和Home域
      - `PCRF` = `Policy and Charging Rules Function` ：计费
        - `V-PCRF` = `Visited Network - PCRF`
        - `H-PCRF` = `Home Network - PCRF`
      - `PMN` = `Public Mobile Network`
        - `V-PMN` = `VPMN` = `Visited PMN`
        - `H-PMN` = `HPMN` = `Home PMN` = `Home Public Mobile Network`
    - 网关Gateway
      - 包括
        - `AGW` = `Access Gateway`
          - `IMS-AGW` = `IMS Access Gateway`

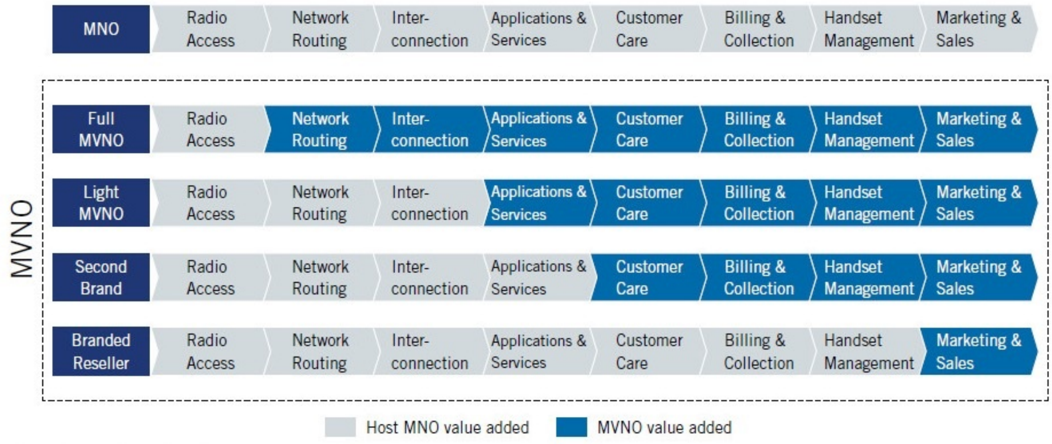
    - `BG` = `Border Gateway`
    - `TrGW` = `Transition Gateway`
      - `T-SGW` = `Transport Signalling Gateway`
    - `MGW` = `Media Gateway`
      - `IM-MGW` = `IP Multimedia - Media Gateway`
    - `SGW`
      - = `Signalling Gateway`
        - `R-SGW` = `Roaming Signalling Gateway`
      - = `Serving Gateway`
    - `P-GW` = `PGW` = `PDN-GW` = `PDN Gateway` = `Packet Data Network Gateway`
      - 简称： `Packet Gateway`
    - `THIG` = `Topology Hiding Inter-network Gateway`
    - `SGC` = `Session Gateway Controller`
    - `SBG` = `Session Border Gateway`
  - 网关控制
    - `BGCF` = `Breakout Gateway Control Function`
    - `MGCF` = `Media Gateway Control Function`
- IR协议?
  - `NVAS` = `Network Value Added Services`
  - `SIMPLE` = `Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions`
  - `IP-CAN` = `IP Connectivity Access Network`
  - `CDR` = `Charging Data Records`
  - `NAT-PT` = `Network Address Translation - Protocol Translation`
    - `NAT` = `Network Address Translation`
  - `XDM` = `XML Document Management`
  - interconnection相关
    - `II-NNI` = `Inter-IMS-NNI`
    - `IBCF` = `Interconnection Border Control Function`
  - 身份识别
    - `IMSI` = `International Mobile Subscriber Identity`
- IMS
  - IMS
    - `IMS` = `IP Multimedia Subsystem` = `Internet Protocol Multimedia Subsystem` = `IP多媒体子系统`
    - 所属组织：3GPP
      - -》3GPP定义的：IMS
  - IMS的
    - `ICS` = `IMS Centralized Service(s)`
    - `ICSI` = `IMS Communication Service Identifier`
    - `IARI` = `IMS Application Reference Identifier`
    - `IMRN` = `IMS Routing Number`
  - 基于IMS的服务=IMS based services
    - 也统称为：
      - `Converged IP Communications Services` = `融合IP通讯服务`
    - 包括
      - `VoLTE` = `Voice over LTE` = `Voice over Long Term Evolution`
        - 服务连续
          - `SRVCC` = `Single Radio Voice Call Continuity`
        - 编解码
          - `AMR` = `Adaptive Multi-Rate`
            - `AMR-NB` = `AMR Narrow Band`
            - `AMR-WB` = `AMR Wide Band`

            - `DTMF` = `Dual-Tone Multiple Frequency`
            - 漫游
              - `LBO` = `Local Brakeout`
              - `S8HR` = `S8 Home Routed`
          - `ViLTE` = `Video over LTE`
            - 是 `VoLTE` 的一种扩展
          - `VoWiFi` = `VoWi-Fi` = `Voice over WiFi` = `Voice over (EPC-integrated) Wi-Fi`
          - `RCS`
          - `SMSoIP`
      - 架构
        - `CSCF` = `Call / Session Control Function`
          - `I-CSCF` = `Interrogating CSCF`
          - `P-CSCF` = `Proxy CSCF`
          - `S-CSCF` = `Serving CSCF`
          - `E-CSCF` = `Emergency CSCF`
        - `SLF` = `Subscription Locator Function`
        - `HSS` = `Home Subscriber Server`
        - `AS` = `Application Server`
          - `TAS` = `Telephony Application Server`
          - `SCC AS` = `Service Centralization and Continuity Application Server`
      - 授权
        - `AAA` = `Authentication Authorization and Accounting`
          - `GBA` = `Generic Bootstrapping Architecture`
  - GSMA
    - `GSMA` = `GSM Association`
    - 协议规范
      - `PRD` = `Permanent Reference Document`
    - `RCS`
      - `RCS`
        - =旧： `Rich Communications Suite`
        - =新： `Rich Communications Service`
          - = `Rich Communications Services`
      - 协议
        - `RCS BB` = `RCS Blackbird`
        - `RCS UP` = `RCS Universal Profile`
      - `RCS API`
        - `TAPI` = `T-API` = `Terminal RCS API`
        - `NAPI` = `N-API` = `Network RCS API`
    - 接口
      - `UNI` = `User-Network Interface` = `User-to-Network Interface` = `用户对网络接口`
      - `NNI` = `Network-Network Interface` = `Network-to-Network Interface` = `网络对网络接口`
    - `MaaP` = `Messaging as a Platform`
    - 服务器
      - `AS` = `Application Server` = `应用程序服务器`
        - 另外
          - `AS` = `Autoconfiguration server` ？
        - 包含
          - `IWF` = `Interworking Function`
        - 用于
          - `IM` = `Instant Message` = `即时消息(服务)`
      - `Config Server` = `Auto Configuration Server`

        - 授权认证
          - `OTP` = `One Time Password` = `一次性密码`
    - RCS协议内部名词
      - 功能
        - `NAB` = `Network Address Books` = `网络地址簿`
        - `IS` = `Image Share` = `图片共享`
        - `FT` = `File Transfer` = `文件传输`
      - `EUCR` = `End User Configuration Requests`
      - `ACR` = `Anonymous Customer Reference`
      - `PNB` = `Personal Network Blacklist`
    - 项目
      - `NGRAI` = `Next Generation Roaming and Interoperability`
  - `IETF` = `Internet Engineering Task Force`
    - `XCAP` = `XML Configuration Access Protocol`

MNO

- `MNO`
  - = `Mobile Network Operator` = `移动网络运营商`
    - = `Mobile Operator` = `移动运营商`
  - 概述
    - Variants of the MVNO business model



MNO	Radio Access	Network Routing	Inter-connection	Applications & Services	Customer Care	Billing & Collection	Handset Management	Marketing & Sales
Full MVNO	Radio Access	Network Routing	Inter-connection	Applications & Services	Customer Care	Billing & Collection	Handset Management	Marketing & Sales
Light MVNO	Radio Access	Network Routing	Inter-connection	Applications & Services	Customer Care	Billing & Collection	Handset Management	Marketing & Sales
Second Brand	Radio Access	Network Routing	Inter-connection	Applications & Services	Customer Care	Billing & Collection	Handset Management	Marketing & Sales
Branded Reseller	Radio Access	Network Routing	Inter-connection	Applications & Services	Customer Care	Billing & Collection	Handset Management	Marketing & Sales

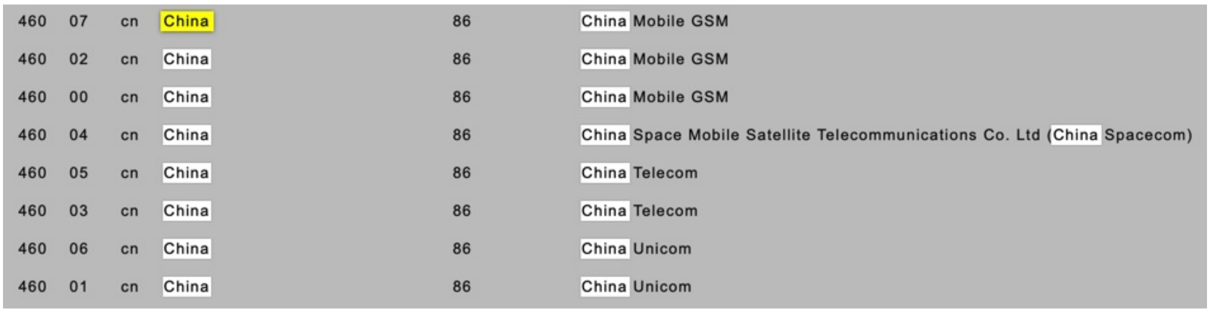
Variants of the MVNO business model

  - 常见
    - 国外
      - Vodafone
      - Verizon
      - T-Mobile
      - Orange
      - Telefonica
    - 中国国内
      - 中国移动
      - 中国联通
      - 中国电信
  - 职责
    - 需要维护
      - OSS=Operation Support System
      - BSS=Business Support System
- 相关
  - `MVNO`
    - = `Mobile Virtual Network Operator` = `移动虚拟网络运营商`
      - = `虚拟运营商`
    - 常见
      - 国外
        - Virgin Mobile
        - Lyca Mobile
        - Walmart Mobile
        - Lebara Mobile
      - 国内
        - 蜗牛移动
  - MVNA=Mobile Virtual Network Aggregator

    - MVNE=Mobile Virtual Network Enabler

MSIN

- MSIN
    - 全称
        - =Mobile Subscription Identification Number=国际移动用户识别码
        - 有时候也写成：Mobile Subscriber Identification Number
    - 别称
        - MIN=Mobile Identification Number
        - MSID=Mobile Station ID
        - IMSI_S=Short IMSI
    - 作用
        - MNO用来区分移动手机（用户）的
    - 特点
        - 唯一的
    - 数值
        - 最多包含10个数字
            - 15位的IMSI，取最后10位
                - the last 10 digits of the 15 digit long IMSI number are the MSIN



460	07	cn	China	86	China Mobile GSM
460	02	cn	China	86	China Mobile GSM
460	00	cn	China	86	China Mobile GSM
460	04	cn	China	86	China Space Mobile Satellite Telecommunications Co. Ltd (China Spacecom)
460	05	cn	China	86	China Telecom
460	03	cn	China	86	China Telecom
460	06	cn	China	86	China Unicom
460	01	cn	China	86	China Unicom

有价值的资料

- RCS演化
    - 美国的5G和LTE和RCS演化概览
        - https://www.5gamericas.org/wp-content/uploads/2019/07/VoLTE_RCS_TECHNOLOGY_ECO-SYSTEM_AND_EVOLUTION_Final_for_upload.pdf
- RCS概览
    - GSMA RCS Blackbird Overview | NETMANIAS
        - https://www.netmanias.com/en/post/techdocs/8102/lte-rcs/gsma-rcs-blackbird-overview
- IMS架构
    - Network Framework of IMS YUAN ZHANG China Telecom Beijing Research Institute
        - https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/SiteAssets/Pages/Events/2016/Oct-CandI2016/CAICT2016/Session%203-2%20Framework%20of%20IMS%E5%BC%A0%E5%9B%AD.pdf
    - IMS An Architecture for Convergent Next Generation Multimedia services
        - https://www.iaria.org/conferences2007/filesICSNC07/IMSConvergentMultimediaServices.ppt
- IMS初始化过程
    - IMS Identification and numbering
        - http://www.leliwa.com/downloads/ims_identification_and_numbering.pdf

3GPP协议规范下载

3GPP协议下载入口：

- Directory Listing /ftp/Specs/archive/22_series/
    - Directory Listing /ftp/Specs/archive/22_series/22.340/
        - TS 2.340 = IMS spec
            - https://www.3gpp.org/ftp/Specs/archive/22_series/22.340/22340-g00.zip

参考资料

DOING：

-

TODO:

- 【记录】研究Android项目rcsjta中RCS的core的service启动的代码调用逻辑关系和顺序
- 【记录】研究rcsjta中的RcsCoreService的代码调用逻辑和顺序
-
- 【未解决】IMS系统中用户账号注册和SIP会话控制相关流程和过程逻辑
- 【记录】继续研究rcsjta相关文档和RCS相关资料
- 【未解决】通过调试rcs的core加运行其他几个apk去研究rcsjta的服务没运行的原因
- 【整理】RCS相关协议学习：RCC.13v1.0.pdf Rich Communication Suite RCS API
- 【未解决】搞懂rcsjta的Provisioning和IMS核心参数：IMS private URI for HTTP Digest
- 【已解决】换安卓9的红米Note8Pro去尝试获取IMSI
-
- 【记录】学习RCS开源项目rcsjta的文档：RCSJTA_open_source.ppt
- 【记录】学习RCS开源项目rcsjta的文档：TAPI_architecture.ppt、TAPI_capabilities_contact.ppt、TAPI_chat.ppt
- 【未解决】确保已给rcsjta项目安装了匹配的版本的Java即JDK
- 【已解决】gradlew tasks报错：A problem occurred configuring project SDK location not found
- 【未解决】编译和测试RCS开源项目rcsjta
- 【已解决】rcsjta项目编译报错：Could not read platform-tools/api/annotations.zip
- 【未解决】搞懂rcsjta的项目的编译和测试的逻辑如何上手
- 【已解决】安装rcsjta的settings的build出的settings-debug的apk到安卓小米9中
- 【部分解决】安装rcsjta的RI的build出的RCS-debug的apk到安卓小米9中
- 【记录】根据Android Studio提示去升级Gradle和更新配置提高性能
- 【已解决】Android Studio中build编译rcstja项目报错：Could not find method leftShift for arguments on task notification installServiceFirst of type org.gradle.api.DefaultTask
- 【已解决】Android Studio中编译rcsjta报错：ERROR Failed to resolve play-services
- 【已解决】Android Studio编译rcsjta出现gradle警告：The specified Android SDK Build Tools version is ignored
- 【已解决】Android Studio编译rcsjta出现gradle警告：Configuration compile is obsolete and has been replaced with mplementation and api It will be removed soon
- 【已解决】Android Studio中编译rcsjta报错：ERROR The minSdk version should not be declared in the android manifestile
- 【已解决】Android Studio重新编译rctjsa报错：duplicate entry META-INF/MANIFEST.MF
- 【已解决】Android Studio中gradle提示：Android Studio is using this JDK location which is different to what Gradleses by default
- 【未解决】Android Studio中Project Structure提示错误：rootProject.compileSdkVersion Unresolved reference Resolvedndroid-23
- 【未解决】Mac中用Android Studio打开rctjsa项目并编译apk和实时调试安卓手机小米9
- 【未解决】寻找安卓中rcsjta的3个app的log日志文件
- 【无需解决】安卓项目rcsjta运行报错：Unknown bits set in runtime_flags 0x8000
- 【未解决】下载适合Mac的包含audit2allow的policycoreutils-python的rpm包
- 【未解决】Mac中找不到audit2allow：zsh command not found audit2allow
- 【未解决】Mac中如何安装rpm包policycoreutils-python
- 【未解决】安卓项目rcsjta的core运行报错：libc Access denied finding property ro.vendor.df.effect.conflict

- 【无法解决】安卓项目rcsjta的core运行报错：type 1400 audit avc denied read for name object_rendor_displayfeature_prop s0 dev tmpfs scontext tcontext tclass file permissive 0
- 【未解决】用Android Studio连接安卓真机小米9去安装调试apk
- 【未解决】安卓项目rcsjta的core运行报错：Perf Fail to get file list com.gsma.rcs getFolderSize Exception_1 java.lang.ullPointerException: Attempt to get length of null array
- 【未解决】Android Studio去红米Note8Pro中调试rcsjta的core的apk
- 【未解决】Android Studio真机小米9中调试rcsjta的core的apk
- 【未解决】Android Studio真机小米9中调试rcsjta的RI的apk
- 【未解决】Android Studio去红米Note8Pro中调试rcsjta的RI的apk
- 【已解决】rcsjta项目编译出错：MultimediaMessagingSessionListenerImpl.java使用或覆盖了已过时的API
- 【已解决】rcsjta项目编译出错：DisplayGeoloc.java使用了未经检查或不安全的操作
- 【整理】Android中的数据传递和序列化：Serializable和Parcelable
- 【已解决】Android项目中@TargetApi语法报错：Cannot resolve symbol TargetApi
- 【已解决】rcsjta项目编译出错：Task tts compileDebugJavaWithJavac 警告 deprecation Service中的onStart(Intent,int)已过时
- 【已解决】rcsjta项目编译出错：TextToSpeech中的speak(String,int,HashMap)已过时
- 【未解决】rcsjta项目编译出错：警告deprecation android.hardware中的Camera已过时
- 【未解决】Android Studio中gradle编译rcjsta项目多个使用或覆盖了已过时的API等报错
- 【已解决】rcsjta项目编译出错：警告 deprecation PreferenceActivity中的addPreferencesFromResource(int)已过时
- 【已解决】rcsjta中rcs/tts/Main.java和原先3个apk即core和RI和settings的关系
- 【未解决】rcsjta项目编译出错：警告deprecation PreferenceActivity中的getPreferenceScreen()已过时
- 【未解决】Android中导入androidX去实现PreferenceFragmentCompat的rcsjta的tts的设置
- 【未解决】Android Studio中用PreferenceFragmentCompat实现rcsjta的tts的参数设置页面
- 【未解决】Android项目rcsjta的子项目api_cnx编译报错：android.app中的ProgressDialog已过时
- 【未解决】rcsjta的RI启动报错：Cannot connect service API Failed to get result for com.gsma.services.rcs.action.ET_SERVICE_STARTING_STATE
- 【未解决】调试运行Android项目rcsjta的其他几个apk看看是否启动了rcs的core的service
- 【未解决】rcsjta中provisioning即RCS template运行报错：IOException occurred Unable to resolve host config.rcs.mnc009.cc460.pub.3gppnetwork.org
- 【记录】研究rcsjta的几个apk的运行现象尝试搞懂rcs的core的service服务相关逻辑
- 【未解决】通过调试core加运行RI去研究rcsjta的服务没运行的原因
- 【记录】研究Android项目rcsjta中RCS的core的service启动的代码调用逻辑关系和顺序
- 【无法解决】尝试通过卸载重装rcsjta的几个apk看看现象是否有变化以求找到rcs的core的service启动逻辑
- 【已解决】rcsjta中RI检测core的Service的status时报错：API is not compatible
- 【无法解决】调试rcsjta的settings以触发setActivationMode设置激活模式以去找RCS服务启动逻辑
- 【已解决】rcsjta的settings中关闭重新开启RCS Service会报错：Failed to get result for com.gsma.service.rcs.action.ET_ACTIVATION_MODE see Logcat
- 【未解决】Android项目rcsjta中如何才能运行到rcs的core的service
- 【未解决】rcsjta中调试core的apk即RCS Provisioning的app中功能和选项以寻找core的service启动逻辑
- 【已解决】安卓项目rcsjta中core的apk中入口activity或service是什么
- 【未解决】rcsjta中研究RCS的core的Service本身相关代码以希望找到为何没启动的原因
- 【未解决】rcsjta的core出错：FATAL EXCEPTION main android.os.FileUriExposedException file exposed beyond app through | ntent.getData()
- 【未解决】rcsjta的core出错：java.lang.NullPointerException Attempt to invoke virtual method java.lang.String com.sma.rcs.utils.ContactUtil PhoneNumber.getNumber on a null object reference
- 【记录】调试rcsjta的core的apk点击保存按钮触发Provisioning的onOptionsItemSelected相关代码调用逻辑
- 【整理】RCS相关协议学习：RCC.14_v5.0.pdf Service Provider Device Configuration
- 【整理】RCS相关协议学习：RCC.15 v1.0.pdf IMS Device Configuration and Supporting Services
- 【整理】RCS相关协议学习：RCC.07-v11.0-2.pdf Advanced Communications Services and Client Specification
- 【整理】RCS相关协议学习：RCC.13v1.0.pdf Rich Communication Suite RCS API

- 【整理】RCS相关协议学习：IR.90-v16.0.pdf RCS Interworking Guidelines

DONE:

- 移动网络演化史
- 【调研】5G消息RCS消息技术开发
- 【记录】iPhone中试用中国移动的RCS的app：5G消息
- 【整理】rcsjta项目中的RCS的版本和基础知识
- 【整理】RCC是什么及RCS有哪些相关协议和规范
- 【整理】RCS相关名词和含义解析和逻辑关系
- 【整理】VoLTE和RCS服务演化
- 【已解决】学习rcsjta的wiki文档搞懂开发逻辑
- 【未解决】确保已给rcsjta项目安装了匹配的版本的Java即JDK
- 【已解决】参考开发环境文档入手开发RCS的项目rcsjta
- 【已解决】Mac中安装android Studio和android SDK
- 【已解决】Android相关环境变量定义：ANDROID_SDK_ROOT和ANDROID_HOME
- 【未解决】gradlew tasks报错：A problem occurred configuring project SDK location not found
- 【已解决】rcsjta项目编译报错：Could not read platform-tools/api/annotations.zip
- 【已解决】编译rcsjta项目中的RCS的core
- 【未解决】给安卓小米9中安装rcsjta的core的build出的RCS_Core的apk
- 【已解决】编译rcsjta项目中的RCS的RI参考实现
- 【已解决】安装rcsjta的RI的build出的RCS-debug的apk到安卓小米9中
- 【已解决】编译rcsjta项目中的RCS的settings设置
- 【已解决】安装rcsjta的settings的build出的settings-debug的apk到安卓小米9中
- 【未解决】rcsjta中RCS的RI中各种功能都提示：The service is not available Please retry later
- 【未解决】通过调试rcsjta中core的入口Provisioning去寻找服务启动的逻辑
- 【已解决】搞懂rcsjta的core的Provisioning中IMS name合法用户账号的写法和逻辑
- 【已解决】什么是IMSI以及值是多少
- 【已解决】什么是MSIN以及中国移动的11位的手机号的MSIN是多少
- 【已解决】安卓10的小米9中中国移动手机号的SIM的MSIN是什么
- 【已解决】换安卓9的红米Note8Pro去尝试获取IMSI
- 【已解决】搞懂rcsjta的Provisioning中IMS domain domain含义和如何填写
- 【未解决】搞懂rcsjta的Provisioning和IMS核心参数：IMS private URI for HTTP Digest
- 【已解决】运行RCS服务是否需要IMS的AS服务器以及P-CSCF等相关名词含义
- 【基本解决】研究rcsjta中源码中和启动RCS的服务的相关代码和逻辑
- 【整理】搞清楚RCS和IMS的架构逻辑关系
- 【已解决】RCS和IMS相关机构组织及其协议之间的关系和逻辑
- 【整理】RCS历史背景相关知识总结
- 【整理】RCS相关概念和基础知识：Albatros、Blackbird、Crane、joyn
-
- 5G消息RCS技术总结
- RCS/IMS/TISPAN/NGN架构和相关组织关系图 | ProcessOn免费在线作图,在线流程图,在线思维导图
- 用户和认证 · 移动网络演化史
-
- VoLTE_RCS_TECHNOLOGY_ECO-SYSTEM_AND_EVOLUTION_Final_for_upload.pdf
- RCC-55-v2-0.pdf
- GSMA PRD RCC.07 - RCS 5.3 - Advanced Communications Services and Client Specification, Version 6.0, 28 Feb 2015
- GSMA | Accreditation of the RCS Blackbird release and growth of native RCS devices - Future Networks
- GSMA | Universal Profile - Future Networks

- GSMA | RCC.20 Enriched Calling Technical Specification v6.0 - Newsroom
- GSMA | RCC.10 Rich Communication Suite Endorsement of OMA CPM 2.2 Interworking v9.0 - Newsroom
- GSMA | RCC.11 Rich Communication Suite Endorsement of OMA CPM 2.2 Conversation Functions v9.0 - Newsroom
- GSMA | RCC.71 RCS Universal Profile Service Definition Document v2.4 - Newsroom
- GSMA | RCC.07 Rich Communication Suite - Advanced Communications Services and Client Specification v11.0 - Newsroom
- GSMA | RCC 14 v5.0 - Service Provider Device Configuration - Newsroom
- GSMA | RCC 15 v5.0 - IMS Device Configuration and Supporting Services - Newsroom
- GSMA RCS Blackbird Overview | NETMANIAS
- MNO, MVNO, MVNA, MVNE - The different types of operators | LinkedIn
- GSMA | ViLTE (Video over LTE) - Future Networks
- GSMA | VoWifi (Voice over Wi-Fi) - Future Networks
- GSMA | VoLTE (Voice over LTE) - Future Networks
- GSMA | RCS Demos & Use Cases - Future Networks
- IMS/SIP - RCS : Registration Home : www.sharetechnote.com
- IR.90-v16.0.pdf
- GSMA | VoLTE - Documents - Future Networks
- IMS Service Centralization and Continuity Guidelines Version 13.0 02 February 2016
- IMS Roaming, Interconnection and Interworking Guidelines Version 30.0 08 April 2019
- AAA architectures applied in multi-domain IMS (IP Multimedia Subsystem) - IEEE Conference Publication
- IP connectivity access network - Wikipedia
- 手机对讲服务 - 维基百科，自由的百科全书
- The evolution of 3GPP Communication Services VoLTE and RCS Technology Evolution and Ecosystem
- vol13_4_045en.pdf
- What is Voice over LTE (VoLTE)?
- S3P5_Doug_Makishima.pdf
- GSMA Compliance
- What is Voice over Wi-Fi (VoWiFi)?
- Cataleya - VoWiFi Solution
- GSMA | IR.51 IMS over Wi-Fi V8.0 - Newsroom
- IMS Profile for Voice, Video and SMS over untrusted Wi-Fi access Version 8.0 29 April 2020
- IMS Profile for Voice, Video and SMS over Wi-Fi Version 1.0
- Specification # 24.301 Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3
- IP Multimedia Subsystem (IMS) emergency sessions (Release 12)
- Multi Device Version 3.0 13 May 2020
- GSMA | Recent developments in Network 2020 Specifications - Future Networks
- (PDF) Solving cluster ensemble problems by bipartite graph partitioning | Carla Brodley - Academia.edu
- ClusterEnsembles.pdf
- ibcf.pdf
- A magic box called SBC | Real Time Communication
- Roaming in IMS | Real Time Communication
- Core Border Gateway Function - How is Core Border Gateway Function abbreviated?
- Border Gateway Function 6.5 to 11.1.1
- 1_10948-CRA1191170_1Uen.A.pdf
- Interconnect Border Gateway Function (I-BGF) on UMG8900-Core Network Info Community-Huawei Connect
- pdfnGz3qUZqNw.pdf
- ITU-T / ATIS Workshop “Next Generation Technology and Standardization” Las Vegas, 19-20 March 2006 Signaling Architecture
- Short Message Service technical realisation (GSM) - Wikipedia)
- Signalling System No. 7 - Wikipedia

- SIGTRAN - Wikipedia
- Mobile Application Part - Wikipedia
- 网络交换子系统 - 维基百科，自由的百科全书
- Short Message service center - Wikipedia
- Short Message Peer-to-Peer - Wikipedia
- YateUCN - Unified Core Network for EPC/IMS and GSM/GPRS
- Full MVNO Architecture - HSS/HLR, GMSC, GGSN, PGW and SMSC
- Network switching subsystem - Wikipedia
- What is an Authentication Center? - Definition from Techopedia
- GSM - Wikipedia
- Short Message Service (SMS) Test Solutions over LTE/IMS, UMTS and GSM Networks - Newsletter
- Short Message Service (SMS) Test Solutions
- Wireless Network Simulation (5G, 4G, 3G, 2G, IP, TDM)
- 2G-GSM-GPRS-Wireless-Lab-Simulation-Presentation.pdf
- Messaging SMSC, IP-SM-GW, and MMS Server
- Home NodeB - Wikipedia
- 5G Terminology: The gNB - 5G Networks
- Network Framework of IMS YUAN ZHANG China Telecom Beijing Research Institute
- IP多媒体子系统 - 维基百科，自由的百科全书
- Session Control - an overview | ScienceDirect Topics
- What is Call Session Control Function (CSCF)?
- 4.6 Roles of Session Control Functions
- CN101039268A - 限制用户设备接入的方法及mgc和cscf - Google Patents
- Global Ip Multimedia Subsystem (Ims) Market Trend Share Study Reports
- PPT - IP Multimedia Subsystem IMS PowerPoint Presentation, free download - ID:4364490
- PPT - Dr. Sorin Georgescu sorin.georgescu@ericsson PowerPoint Presentation - ID:5988890
- IMS An Architecture for Convergent Next Generation Multimedia services
- IMS functional architecture overview | Download Scientific Diagram
- Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Functional Architecture
- Telecoms & Internet converged Services & Protocols for Advanced Networks - Wikipedia
- FiberHome IMS Solution
- IMS Identification and numbering
- Domain management with IMS - IMS | VC4.com
- 兼容性测试套件 | Android 开源项目 | Android Open Source Project
- IMS/RCS Technology - Public Schedule - Leliwa
- RCS-5.1-V2.0-UNI.pdf
- rcs-e_advanced_comms_specification_v1_2_2_approved.pdf
- "IMS/RCS Technology" Chapter 05 Security (sample) by Leliwa - issuu
- IMS Identification and numbering
- US20120036270A1 - IP Multimedia Subsystem User Identity Handling - Google Patents
- IP Multimedia Services Identity Module - Wikipedia
- IP Multimedia Subsystem - Wikipedia
- Universal integrated circuit card - Wikipedia
- File:Sim card.png - Wikipedia
- Carrier Network - an overview | ScienceDirect Topics
- Kamailio SIP Server
- tutorials:ims:installation-howto Kamailio SIP Server Wiki
- Domain management with IMS - IMS | VC4.com
- IR.65-v30.0.pdf
- IP Multimedia Subsystem - Wikipedia

- International mobile subscriber identity - Wikipedia
- China Mobile - Wikipedia
- LTE frequency bands in China (CTC, CMCC, CUC)
- Most up to date list of MCC and MNC codes: mobile country codes – mobile network codes
- spn与plmn，imsi与imei简单记录 - 简书
- Final draft ETSI ES 282 001 V2.0.0 (2007-11) ETSI Standard Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Functional Architecture
- Telecoms & Internet converged Services & Protocols for Advanced Networks - Wikipedia
- Messaging in RCS | Real Time Communication
- GSMA Advanced Messaging – RCS Universal Profile | Real Time Communication
- GSMA RCS Architecture | NETMANIAS
- RCS and Communication Landscape | NETMANIAS
- Evolution of GSMA RCS standards | NETMANIAS
- RCS and NextGen Mobile | Pipeline Magazine | OSS and BSS News and Info
- Messaging in RCS | Real Time Communication
- Evolution of GSMA RCS standards | NETMANIAS
- RCS High-Level Functional and Subsystem Architecture. (a) Data Cube... | Download Scientific Diagram
- GSMA RCS Architecture | NETMANIAS
- Google rescues RCS to compete WhatsApp | Technology Issues Blog
- Mavenir: Monetizing RCS through Innovation on Cloud Native Network
- Radisys & Mavenir: Monetizing VoLTE and RCS
- "RCS-e/joyn Basics" Chapter 07 Multi-device (sample) by Leliwa - issuu
-