

Mestrado em Engenharia Informática  
Estágio  
Relatório Final

# Desenvolvimento de aplicações de Rich Communications para Android

Manuel Nunes  
macnunes@student.dei.uc.pt

Orientador do DEI:  
Pedro Furtado

Orientador da WIT-Software:  
Tiago Leitão

Data: 4 de Setembro de 2013



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



## **Resumo**

Atualmente milhões de pessoas em todo o mundo tiram partido de dispositivos de comunicação móvel. A emergência dos *smartphones* no mercado das telecomunicações confere aos seus utilizadores não só um vasta gama de funcionalidades como uma ligação constante à Internet e às potencialidades que esta proporciona.

O consumidor torna-se cada vez mais exigente, procurando soluções de comunicação que o permita ir além dos tradicionais serviços de chamadas e SMS. Com base nesta premissa, verifica-se uma proliferação de aplicações que oferecem serviços mais ricos de comunicação, sem a inconveniência de custos adicionais.

As operadoras móveis encontram-se assim em risco de perderem a supremacia a nível do fornecimento de serviços. Como resposta a esta ameaça resulta o *Rich Communication Suite*. Uma solução de comunicações enriquecidas, desenvolvida com base em critérios de interoperabilidade e qualidade de serviço.

O desenvolvimento de aplicações de *Rich Communications* para Android apresenta-se assim como uma das prioridades da indústria de telecomunicações atual.

Este Estágio de Mestrado Integrado em Engenharia Informática tem como objetivo desenvolver e lançar para o mercado uma aplicação RCS. Está inserido num projeto para um cliente real, tornando-o relevante para o atual mercado das comunicações móveis.

## **Palavras-Chave**

“Android”, “Enhanced Address Book”, “Enhanced Messaging”, “Enhanced Voice”, “joyn”, “Rich Communication Suite”, “Rich Communication Suite Enhanced”, “Smartphone”



# Índice

1. Introdução .....	1
1.1 Contextualização .....	1
1.1.1 O Problema.....	1
1.1.2 A Solução .....	1
1.2 Objectivos .....	2
1.3 Estrutura do Documento .....	3
2. Estado da Arte.....	4
2.1 Funcionalidades .....	4
2.2 Plataformas .....	9
2.3 Encargos de utilização .....	10
2.4 Popularidade.....	10
3. Requisitos .....	13
3.1 Especificações RCS .....	13
3.1.1 RCS-e .....	13
3.1.2 Hot Fixes.....	14
3.1.3 RCS Release 5.....	15
3.1.4 RCS Release 5.1.....	15
3.1.5 joyn Blackbird.....	15
3.2 Análise de Requisitos .....	15
3.3 RCS-e Joyn Hot Fixes.....	16
3.3.1 Blocking RCS-e Services.....	16
3.3.2 RCS-e Service Touch Points.....	16
3.3.3 Chat .....	17
3.3.4 File Transfer.....	17
3.3.5 In Call Share.....	17
3.3.6 Settings.....	17
3.3.7 Roaming.....	18
3.3.8 Multi-device/Multi-client Scenarios .....	18
3.4 joyn Blackbird.....	18
3.4.1 Service Activation .....	18
3.4.2 Video Share orientation .....	18
3.4.3 Location Share.....	19

3.5 Requisitos Diferenciadores .....	19
3.5.1 SMS Inbox .....	19
3.5.2 Content Backup .....	19
3.5.3 Messaging .....	20
4. Arquitetura.....	21
4.1 Tecnologias.....	21
4.1.1 Android.....	21
4.1.2 Protocolos .....	22
4.2 Visão geral.....	22
4.2.1 IP Multimedia Core Network Subsystem.....	23
4.2.2 Application Servers .....	24
4.2.3 Cliente RCS.....	24
4.3 Arquitetura da aplicação .....	24
4.3.1 RCS-e stack .....	25
4.3.2 Camada Gráfica.....	29
4.3.3 Bibliotecas externas .....	30
5. Controlo de Qualidade .....	31
5.1 Testes de Acreditação .....	31
5.2 Testes de Desenvolvimento.....	31
6. Trabalho Desenvolvido .....	32
6.1 Implementação.....	32
6.2 Contactos .....	33
6.2.1 O mesmo número de telefone em diferentes contactos .....	33
6.2.2 Contactos com múltiplos números de telefone.....	34
6.3 Content Sharing .....	34
6.3.1 Call on Hold .....	34
6.3.2 Video Share em Landscape .....	35
6.4 File Transfer .....	35
6.4.1 File Transfer Auto-Accept .....	35
6.4.2 File Transfer thumbnails.....	36
6.5 Messaging.....	36
6.5.1 Separação entre 1-to-1 e Group Chat .....	36
6.5.2 Sair explicitamente de um Group Chat.....	37
6.5.3 Definição de Alias.....	37

6.5.4 Serviços com contato bloqueado .....	37
6.5.5 Group Chat com contactos bloqueados .....	37
6.5.6 Redesign do ComService.....	38
6.5.7 Funcionalidade de Desenho.....	40
6.6 Preferences.....	40
6.6.1 Preferences Design.....	41
6.7 Interface Geral da Aplicação .....	41
6.7.1 Integração da ActionBar.....	41
6.8 Controlo de qualidade.....	41
7. Gestão do Projeto.....	42
7.1 A Equipa Scrum.....	42
7.2 Os Eventos Scrum.....	43
7.3 Artefactos .....	43
7.4 Ferramentas de Gestão .....	43
8. Planeamento .....	45
8.1 Plano do Projeto .....	45
8.2 Plano para o Primeiro Semestre .....	45
8.3 Plano para o Segundo Semestre .....	46
9. Conclusões.....	47
10. Referências.....	48

## Lista de Figuras

Figura 1 - Cotas de mercado dos sistemas operativos móveis (Fonte:[28]) .....	10
Figura 2 - Evolução das especificações RCS .....	13
Figura 3 - Diagrama de comunicação de um sistema RCS .....	23
Figura 4 - Arquitetura de alto nível .....	24
Figura 5 - Diagrama de comunicação entre a Camada Gráfica e a <i>stack</i> .....	27
Figura 6. Diagrama de sequência da RCS API .....	28
Figura 7 - Diagrama de difusão de <i>Intents</i> .....	29
Figura 8 – Comunicação entre uma <i>Activity</i> e os restantes componentes .....	29
Figura 9 - Diagrama de classes do serviço de gestão de chat e file transfer.....	39
Figura 10 - Diagrama de sequência da comunicação com o ComService .....	40
Figura 11 - Estrutura da equipa <i>Scrum</i> .....	42
Figura 12. Diagrama do processo Scrum .....	43



## Lista de Tabelas

Tabela 1 - Comparação das funcionalidades das OTTs.....	7
Tabela 2 - Comparação das plataformas suportadas pelas OTTs .....	10
Tabela 3 - Comparação de <i>reviews</i> e número de instalações das OTTs .....	11
Tabela 4. Requisitos de contactos bloqueados .....	16
Tabela 5. Requisitos de pontos de acesso à aplicação .....	16
Tabela 6. Requisitos do serviço de <i>chat</i> .....	17
Tabela 7. Requisitos de serviço de <i>file transfer</i> .....	17
Tabela 8. Requisitos do serviço de <i>content share</i> .....	17
Tabela 9. Requisitos de preferências .....	17
Tabela 10. Requisitos de roaming.....	18
Tabela 11. Requisitos de cenários <i>multi-device</i> e <i>multi-client</i> .....	18
Tabela 12. Requisitos de ativação do serviço.....	18
Tabela 13. Requisitos do serviço de <i>video share</i> .....	19
Tabela 14. Requisitos do serviço de <i>location share</i> .....	19
Tabela 15. Requisitos da integração de SMS.....	19
Tabela 16. Requisitos de backup de conteúdos.....	20
Tabela 17. Requisitos da funcionalidade de <i>messaging</i> .....	20
Tabela 18 - Comparação de clientes RCS .....	25
Tabela 19 - Correspondência entre trabalho desenvolvido e requisitos .....	33
Tabela 20. Planeamento do primeiro semestre.....	45
Tabela 21. Planeamento do segundo semestre.....	46
Tabela 22 - Trabalho desenvolvido no segundo semestre.....	46

## **Anexos**

Anexo A – Especificação dos Requisitos

## Acrónimos

Termo	Descrição
AIDL	<i>Android Interface Definition Language</i>
API	<i>Application Programming Interface</i>
GSMA	<i>Global System for Mobile communications Association</i>
HSS	<i>Home Subscriber Server</i>
HTTP	<i>HyperText Transfer Protocol</i>
I-CSCF	<i>Interrogating Call/ Session Control Function</i>
IM	<i>Instant Messaging</i>
IMS	<i>Internet Protocol Multimedia Subsystem</i>
IP	<i>Internet Protocol</i>
MMS	<i>Multimedia Message Service</i>
OEM	<i>Original Equipment Manufacturer</i>
OMA	<i>Open Mobile Alliance</i>
OTT	<i>Over-the-Top</i>
P-CSCF	<i>Proxy Call/ Session Control Function</i>
RCS	<i>Rich Communication Suite</i>
RCS-e	<i>Rich Communication Suite - enhanced</i>
RIL	<i>Radio Interface Layer</i>
RTP	<i>Real-time Transport Protocol</i>
S-CSCF	<i>Serving Call Session Control Function</i>
SDP	<i>Session Description Protocol</i>
SIP	<i>Session Initiation Protocol</i>
SMS	<i>Short Message Service</i>
SO	Sistema Operativo
VoIP	<i>Voice over Internet Protocol</i>
UI	<i>User interface</i>
XML	<i>eXtensible Markup Language</i>

## Glossário

Termo	Descrição
Android	Sistema operativo <i>open-source</i> para dispositivos móveis
Blacklist	Lista de contactos excluídos de serviços de comunicações
Capability Discovery	Processo que possibilita obter informação relativamente aos serviços <i>joyn</i> disponibilizados por um determinado contacto em certo instante temporal.
Chat	Serviço de mensagens instantâneas. É uma forma de comunicação em tempo real através da troca de mensagens de texto entre dois ou mais contactos.
Content Share	A partilha de uma imagem ou vídeo durante uma chamada de voz.
File Transfer	Transferência de um ficheiro de um remetente para um destinatário e o armazenamento do mesmo no dispositivo do destinatário.
Group Chat	Uma sessão de <i>chat</i> com múltiplos participantes
Instant Messaging	Serviço de <i>Chat</i>
Internet Service Provider	Negócio ou organização que oferece acesso à Internet aos utilizadores
In Call Screen	Ecrã mostrado durante uma chamada de voz.
joyn	Imagem de marca para serviços RCS que é propriedade da GSMA
Native Address Book	Lista de contactos nativa de um dispositivo móvel.
Notification	Entrega de uma mensagem a um destinatário com a intenção de captar a sua atenção.
RCS-e Capable Contact List	Uma lista de contactos onde é possível identificar contactos RCS-e
RCS-e Filtered Contact List	Uma lista de contactos filtrada que apenas contém contactos RCS-e
Scrum	<i>Framework</i> para gestão de projetos ágeis
Touch Point	Ponto de contacto com utilizadores antes, durante e depois de alguma interação com serviços RCS-e
Video Call	Comunicação de vídeo bidirecional



# 1. Introdução

## 1.1 Contextualização

### 1.1.1 O Problema

Qualquer serviço que seja oferecido ao utilizador que não seja fornecido diretamente pelo seu *Internet Service Provider* é considerado como sendo “*over-the-top*”.

As aplicações que oferecem serviços de *instant messaging*, chamadas de voz e vídeo, partilha de conteúdos multimédia, entre outros, são consideradas aplicações *over-the-top*. Como exemplo de aplicações deste tipo temos o Facebook Messenger, WhatsApp Messenger, Skype, Viber, KakaoTalk e o LINE.

Atualmente existe um grande número de aplicações deste tipo no mercado e a sua grande maioria proporciona um leque de funcionalidades e serviços variado, mas aqueles que despertam o interesse dos utilizadores são comuns a todas. São estes:

- o envio de mensagens de texto
- o envio de mensagens multimédia
- chamadas de voz

Todos eles sem implicarem quaisquer encargos financeiros extra para o utilizador, apenas necessitam de um plano de dados.

Este tipo de aplicações constitui uma grande ameaça para as operadoras de telecomunicações uma vez que estas deixam de ser fornecedoras de serviços para passarem a ser apenas fornecedoras de ligações de dados. É estimado que em 2011 tenham apresentada uma perda de lucros na ordem dos 10.3 milhares de milhão de euros devido a estas aplicações [1] e já há previsões para perdas na ordem dos 40 milhares de milhão de euros para o ano 2016 [2].

### 1.1.2 A Solução

Face a esta ameaça, as operadoras uniram-se na tentativa de solucionar o problema. Desta colaboração resultou em 2008 a primeira especificação do *Rich Communication Suite* [3] e desde essa data, algumas operadoras já lançaram o serviço para o mercado, como são os casos de Espanha [4], EUA [5] e Coreia do Sul [6].

O *Rich Communication Suite* é uma iniciativa da GSMA. Trata-se de um esforço colaborativo dos principais agentes da indústria das telecomunicações, unindo fornecedores de soluções de redes, operadoras, fabricantes de telemóveis e empresas de desenvolvimento de clientes, na criação de uma solução interoperável e convergente de uma experiência de comunicação enriquecida. É a resposta à ameaça das aplicações OTT, com o intuito de alterar a posição das operadoras de básicos fornecedores de comunicações de dados para fornecedores de serviços. Serviços esses que englobam chamadas de voz e vídeo sobre IP, mensagens instantâneas entre dois ou mais utilizadores (*Chat 1-2-1* e *Group Chat*), transferência de ficheiros (*File Transfer*), partilha de localizações geográficas (*Location Push*) e a partilha de vídeo, imagens e localizações geográficas durante uma chamada (*Video Share*, *Image Share* e *LocationShare*).

De modo a facilitar o reconhecimento destes mesmos serviços por parte do público, o RCS é representado pela imagem de marca *joyn*. Marca essa que só pode ser usada por aplicações aprovadas no processo de acreditação da GSMA.

Relativamente à sua arquitetura, o RCS não tem como propósito a definição de novos standards, pelo contrário, o objetivo da solução passa por proporcionar serviços baseados em standards globais, como é o caso do *IP Multimedia Subsystem* (IMS).

Uma das prioridades no seu desenvolvimento é a interoperabilidade global: a garantia de que os utilizadores podem comunicar entre si, independentemente do dispositivo que possuam, da rede operadora com a qual estão subscritos ou da aplicação que usem.

O RCS posiciona-se assim como alternativa sólida às OTTs, sendo as principais vantagens relativamente às referidas:

- O utilizador não necessita de qualquer tipo de conta para aceder aos serviços, o seu número de telemóvel é a sua identidade e o provisionamento é automático;
- Os serviços RCS são integrados nativamente nos dispositivos móveis e de forma intuitiva, evitando assim que o utilizador tenha de escolher entre diferentes aplicações (OTTs, SMS, chamadas de voz, entre outras) para iniciar uma comunicação;
- Sendo as operadoras as detentoras das redes de comunicações móveis, a qualidade e privacidade dos serviços são asseguradas;
- O tráfego gerado na utilização dos serviços RCS não é contabilizado no plano de dados a que o utilizador esteja subscrito.

## 1.2 Objectivos

O principal objectivo deste trabalho é o desenvolvimento de uma aplicação totalmente compatível com a especificação *RCS-e* com *Hot Fixes* e com algumas das funcionalidade e serviços do *joyn Balckbird*:

- *Enhanced Address Book* – uma lista de contactos com informação relativa aos serviços RCS disponibilizados por cada um, implementada a nível da aplicação e a nível nativo;
- *Enhanced Voice* – serviços de partilha de vídeo em tempo real e imagens durante uma chamada de voz;
- *Enhanced Messaging* – serviços de *instant messaging*, partilha de ficheiros e conversas individuais ou de grupo;
- *Geolocation Share* – serviço de partilha de localizações geográficas;
- *VoIP Calls* – chamadas de voz e vídeo baseados em IP.

Outro ponto importante é a interação com o utilizador. Tratando-se de uma aplicação para o comum utilizador de dispositivos de comunicações móveis, um dos objectivos é também proporcionar uma experiência rica de utilização. Isto com o intuito de fomentar o uso da aplicação e a sua substituição pelas tradicionais funcionalidades de comunicação oferecidas nativamente pelos dispositivos. Este objectivo tem como foco:

- Integração nativa com os serviços – acesso às funcionalidades da aplicação a partir da lista de contatos nativa e dos acessos nativos de partilhas;
- Interface funcionalmente rica e com uma curva de aprendizagem pouco acentuada;

- Máxima performance – interação sem qualquer bloqueio para o utilizador e uma gestão de recursos e bateria do dispositivo de forma otimizada.

### **1.3 Estrutura do Documento**

O documento é composto por 8 capítulos:

1. Introdução
2. Estado da Arte
3. Requisitos
4. Arquitetura
5. Controlo de Qualidade
6. Trabalho Desenvolvido
7. Gestão do Projeto
8. Planeamento
9. Conclusão
10. Referências

No capítulo 2 – Estado da Arte é feita uma análise das aplicações OTT. No capítulo 3 são expostos os requisitos da aplicação. O capítulo 4 descreve a arquitetura de alto nível, apresentando uma visão geral da aplicação e os seus componentes. O capítulo 5 aborda o controlo de qualidade durante o processo de desenvolvimento. No capítulo 6 são abordados os desenvolvimentos efetuados durante o período de estágio. No capítulo 7 é explicado o processo de gestão do projeto, apresentando a metodologia utilizada. O capítulo 8 descreve o planeamento geral do projeto em que o estagiário está inserido e os planos de trabalho para ambos os semestres de estágio. Por fim, os capítulos 8 e 9, são referentes à conclusão e às referências, respetivamente.



## 2. Estado da Arte

Neste capítulo são analisadas as OTTs relativamente às funcionalidades oferecidas, às plataformas suportadas, aos custos envolvidos e à popularidade. À exceção das plataformas, todos os vectores foram analisados com base na versão Android disponível na Play Store da Google. Para cada OTT em análise, o processo passou pela obtenção de informação relativamente aos vectores em análise e pela instalação e teste da aplicação. O principal ponto tido em conta na seleção das aplicações foi a sua popularidade, considerando não só as aplicações com uma posição estabelecida no mercado, mas também aplicações que entraram na disputa recentemente.

Nesta análise são analisadas as seguintes aplicações:

- Facebook [7]
- Skype [8]
- WhatsApp [9]
- KakaoTalk [10]
- LINE [11]
- Tango [12]
- Viber [13]
- ChatON [14]
- Kik [15]
- ooVoo [16]
- WeChat [17]
- Voxer [18]
- Fring [19]
- textPlus [20]
- Cubie [21]
- FreePP [22]
- Hike [23]
- Yuilop [24]
- Text Me! [25]
- Aire [26]
- Vippie [27]

### 2.1 Funcionalidades

Tendo por base um estudo das funcionalidades mais comuns presentes nas OTTs e os serviços do RCS, a seguinte análise irá focar-se na avaliação de um conjunto definido de métricas. Estas podem ser divididas em três áreas mais específicas: comunicação, partilha de conteúdos e gestão de contactos.

As métricas definidas para a Comunicação são avaliadas em dois diferentes contextos: entre dois utilizadores e em grupo.

#### Comunicação

- *Voice Call* – Possibilidade de realizar chamadas VoIP.
- *Video Call* – Possibilidade de realizar chamadas de vídeo.
- *Chat* – Disponibiliza um serviço de mensagens instantâneas.

### **Partilha de Conteúdos**

- *File Share* – Disponibiliza um serviço de envio de ficheiros.
- *Location Share* – Suporta o envio de localizações geográficas.
- *Drawing* – Dispõe de uma funcionalidade de desenho.

### **Gestão de Contactos**

- *Native Address Book Sync* – Capacidade de sincronizar com a lista de contactos do dispositivo, providenciando acesso a serviços a partir de um contacto.
- *Network Address Book* – Capacidade de sincronizar e guardar uma lista de contactos num servidor externo.
- *Black List* – Possibilidade de bloquear comunicações provenientes de contactos seleccionados.

Na tabela 1 é apresentada a comparação destas funcionalidades enunciadas.

	Voice Call		Video Call		Chat		File Share	Location Share	Drawing	Native Address Book	Network Address Book	Black List
	1-1	1-N	1-1	1-N	1-1	1-N						
Facebook	✓				✓	✓	✓	✓	✓	✓	✓	✓
Skype	✓		✓		✓	✓	✓				✓	✓
WhatsApp					✓	✓	✓	✓		✓		✓
KakaoTalk	✓	✓			✓	✓	✓				✓	
LINE	✓				✓	✓	✓	✓	✓		✓	✓
Tango	✓		✓		✓		✓			✓		✓
Viber	✓				✓	✓	✓	✓		✓		
ChatON					✓	✓	✓	✓			✓	
Kik					✓	✓	✓		✓		✓	
ooVoo	✓	✓	✓	✓	✓	✓			✓	✓		✓
WeChat	✓		✓		✓	✓	✓	✓		✓		✓
Voxer					✓	✓	✓	✓		✓		✓
Fring	✓	✓	✓	✓	✓					✓		
textPlus	✓				✓	✓	✓	✓		✓		✓

Cubie					✓	✓	✓	✓	✓	✓		✓
FreePP	✓				✓	✓	✓		✓	✓		✓
Hike					✓	✓	✓	✓			✓	✓
Yuilop	✓				✓	✓	✓	✓		✓		✓
Text Me!	✓		✓		✓	✓	✓	✓		✓		✓
Aire	✓		✓		✓	✓	✓	✓		✓		✓
Vippie	✓	✓	✓	✓	✓	✓	✓			✓		

Tabela 1 - Comparação das funcionalidades das OTTs

Relativamente às comunicações, podemos constatar que a grande maioria das aplicações disponibiliza funcionalidades de *Instant Messaging*, tanto num contexto um-para-um como num contexto de grupo. A possibilidade de chamadas VoIP num contexto um-para-um é também bastante comum, sendo disponibilizada por mais de metade das aplicações em análise. Por outro lado, Chamadas de conferência e chamadas de vídeo são funcionalidades pouco comuns.

Quanto à partilha de conteúdos, a grande maioria possibilita a partilha de ficheiros e a partilha de localizações geográficas é também bastante comum. No entanto, a funcionalidade de desenho apenas é disponibilizada por cinco das aplicações em análise.

No que à gestão de contactos diz respeito, a maioria das aplicações sincroniza apenas com a lista de contactos do dispositivo em vez de sincronizar com um servidor externo. O custo de implementação de um servidor de listas de contactos poderá ser a razão de a grande maioria das aplicações optarem apenas pela integração com a lista de contactos do dispositivo.

## 2.2 Plataformas

A disponibilidade da aplicação nas várias plataformas móveis é um factor importante para a sua popularidade, uma vez que incide diretamente no público alvo e consequentemente nos possíveis utilizadores. Na tabela 2 é apresentada uma comparação entre as aplicações relativamente às plataformas suportadas.

Nome	Android	iOS	Windows Phone	BlackBerry	Bada	Symbian
Facebook	✓	✓		✓		
Skype	✓	✓	✓			
WhatsApp	✓	✓	✓	✓	✓	
KakaoTalk	✓	✓	✓		✓	
LINE	✓	✓	✓	✓		
Tango	✓	✓	✓			
Viber	✓	✓	✓	✓	✓	✓
ChatON	✓	✓	✓	✓	✓	
Kik	✓	✓	✓	✓		✓
ooVoo	✓	✓				
WeChat	✓	✓	✓	✓		✓
Voxer	✓	✓				
Fring	✓	✓				✓
textPlus	✓	✓	✓			✓
Cubie	✓	✓				
FreePP	✓	✓	✓			✓
Hike	✓	✓	✓	✓		✓
Yuilop	✓	✓				
Text Me!	✓	✓	✓			

Aire	✓	✓				
Vippie	✓	✓	✓	✓		✓

Tabela 2 - Comparação das plataformas suportadas pelas OTTs

Como é ilustrado na tabela, todas as aplicações oferecem suporte para Android e iOS. Isto pode dever-se ao facto de as duas plataformas referidas serem as com maior cota no mercado de dispositivos móveis. Windows Phone e BlackBerry são respectivamente a terceira e quarta plataformas mais comuns, posições essas que podem corresponder à terceira e quarta maior cota de mercado, como é representado na figura 1.

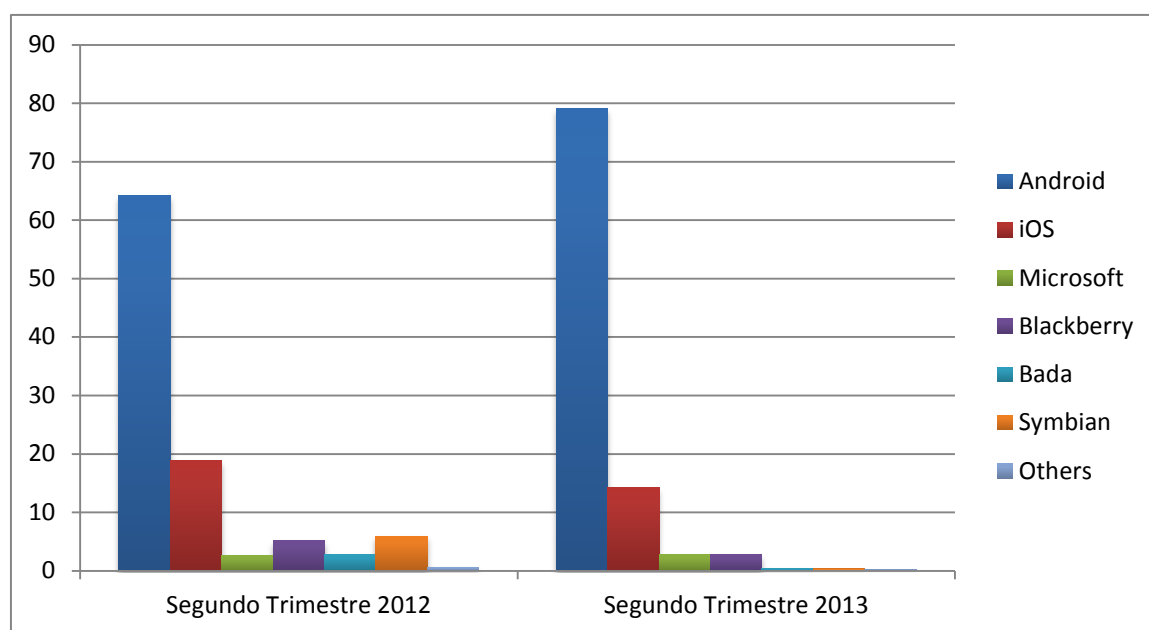


Figura 1 - Cotas de mercado dos sistemas operativos móveis (Fonte:[28])

### 2.3 Encargos de utilização

Um importante factor para a popularidade das OTTs são os encargos financeiros relativos à sua utilização. À exceção do WhatsApp, o qual cobra uma anuidade de 0.99USD após o primeiro ano de utilização gratuito, nenhuma das aplicações em análise apresenta encargos no download e utilização dos seus serviços entre os seus clientes. O utilizador apenas tem que estar conectado a uma rede *WiFi* ou à sua ligação de dados, sendo que esses custos são externos às aplicações em questão.

### 2.4 Popularidade

Os aspectos analisados anteriormente nos pontos 2.1.1, 2.1.2 e 2.1.3 são provavelmente os mais decisivos na popularidade de uma aplicação. Uma vez que informações relativas ao número de utilizadores ativos não são livremente divulgadas, a seguinte análise tem por base os dados disponíveis na *Play Store*[29] da Google, nomeadamente o número de instalações da aplicação e a pontuação atribuída pelos utilizadores. Na tabela 3 é apresentada a comparação

desses aspetos, estando as aplicações ordenadas pelo número de instalações e tendo a pontuação dos utilizadores como segundo factor de ordenação.

Nome	Reviews (0 - 5)	Instalações
<b>WhatsApp</b>	4.5	100,000,000 - 500,000,000
<b>Facebook</b>	4.4	100,000,000 - 500,000,000
<b>Viber</b>	4.4	100,000,000 - 500,000,000
<b>LINE</b>	4.2	100,000,000 - 500,000,000
<b>Skype</b>	4.0	100,000,000 - 500,000,000
<b>ChatON</b>	4.0	100,000,000 - 500,000,000
<b>KakaoTalk</b>	4.4	50,000,000 - 100,000,000
<b>Tango</b>	4.4	50,000,000 - 100,000,000
<b>WeChat</b>	4.4	50,000,000 - 100,000,000
<b>Voxer</b>	4.4	10,000,000 - 50,000,000
<b>Kik</b>	4.4	10,000,000 - 50,000,000
<b>ooVoo</b>	4.2	10,000,000 - 50,000,000
<b>textPlus</b>	4.2	10,000,000 - 50,000,000
<b>Fring</b>	3.9	10,000,000 - 50,000,000
<b>Cubie</b>	4.4	1,000,000 - 5,000,000
<b>FreePP</b>	4.3	1,000,000 - 5,000,000
<b>Hike</b>	4.3	1,000,000 - 5,000,000
<b>Text Me!</b>	4.3	1,000,000 - 5,000,000
<b>Yuilop</b>	4.2	1,000,000 - 5,000,000
<b>Aire</b>	4.1	100,000 - 500,000
<b>Vippie</b>	4.1	100,000 - 500,000

Tabela 3 - Comparação de *reviews* e número de instalações das OTTs

Como a tabela 3 ilustra, cerca de um terço das aplicações em análise apresenta um número de instalações entre os 100 e os 500 milhões. Apesar destes valores não refletirem os número

de utilizadores ativos, podem ser indicativos relativamente à curiosidade e procura de alternativas grátis para comunicações entre dispositivos móveis.

Outro ponto a destacar é a elevada pontuação atribuída pelos utilizadores. À exceção do Fring, todas as aplicações apresentam pontuações iguais ou superiores a 4. Sendo este valor o principal indicador da satisfação dos utilizadores, é possível concluir que as aplicações oferecem uma qualidade bastante elevada.



### 3. Requisitos

Neste capítulo são abordados os requisitos da aplicação. Sendo uma aplicação de RCS, os requisitos são definidos nos documentos de especificação, pelo que se começa por introduzir as versões do RCS mais relevantes para o desenvolvimento e posteriormente a análise dos requisitos em si.

#### 3.1 Especificações RCS

A primeira especificação do RCS data de 2008, com o lançamento da *Release 1* [30]. Desde essa data novas versões foram produzidas como é ilustrado na figura 2.

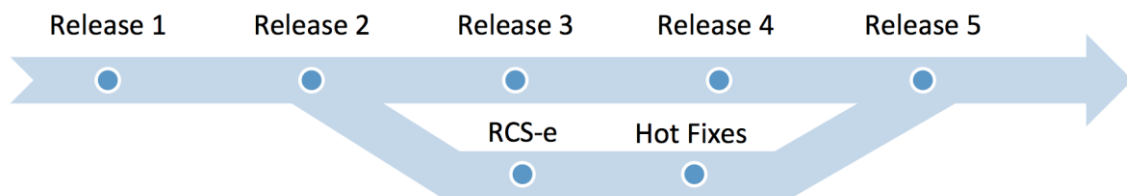


Figura 2 - Evolução das especificações RCS

Para lhe ser atribuído um determinado nível de acreditação pela GSMA, a aplicação tem que reunir os requisitos descritos na especificação em causa. Uma vez que a aplicação tem por base a versão RCS-e, apenas esta e as seguintes especificações serão abordadas nas próximas secções.

##### 3.1.1 RCS-e

A especificação do *Rich Communication Suite enhanced* [31] é uma solução pragmática, com o foco em acelerar o processo de desenvolvimento, de modo a responder mais rapidamente às pressões competitivas do mercado. Na sua origem estão 5 das operadoras líderes do mercado europeu: a Deutsche Telekom, a Orange, a Telecom Italia, a Telefónica e a Vodafone.

Esta especificação tem por base a o RCS *Release 2* [32], sendo definidas várias optimizações com o objectivo de reduzir o tempo de lançamento para mercado. Para além das operadoras que estão na sua origem, também vários fabricantes de dispositivos móveis garantiram o lançamento de dispositivos compatíveis com RCS-e.

A visão do RCS-e é permitir aos utilizadores criar o seu próprio círculo de contactos no qual as interações têm por base um identificador global, o número de telemóvel. As especificações RCS estavam a caminhar para uma fusão entre os serviços de chamadas de voz, SMS e MMS tradicionalmente oferecidos, com os do RCS. Com o RCS-e, o foco é a extensão desses serviços já existentes.

Os utilizadores passam a poder usufruir dos serviços de *chat*, transferência de ficheiros e partilha de conteúdos durante uma chamada em adição aos com que já está familiarizado. Isto com a garantia principal do RCS, a interoperabilidade entre diferentes clientes.

As principais funcionalidades do RCS-e são:

- *Service Capability Discovery*
- *Enhanced Voice*
- *Enhanced Messaging*

O *Service Capability Discovery* consiste em oferecer uma visão clara dos serviços de comunicação para os quais determinado contacto está disponível e permitir ao utilizador escolher qual desses mais lhe convém. Uma das grandes influências nesta disponibilidade é a cobertura de rede e a sua largura de banda disponível. Serviços como a partilha de vídeo necessitam de uma largura de banda substancial, razão pela qual o utilizador pode constatar que certo contacto não disponibiliza este serviço se a cobertura de rede em que se encontra não o permitir.

O *Enhanced Voice* confere aos utilizadores a possibilidade de partilhar vídeos e imagens no decorrer de uma chamada de voz, proporcionando assim uma extensão ao tradicional serviço de chamadas.

O *Enhanced Messaging* proporciona uma experiência mais rica que o tradicional serviço de mensagens SMS, suportando funcionalidades como conversas de grupo e transferência de ficheiros.

Devido ao seu foco na aceleração do lançamento para o mercado, os serviços de *Social Presence* e *Network Address Book* contemplados na *Release 2* não fazem parte da especificação do RCS-e. Estes serviços são opcionais. As operadoras podem disponibilizá-los em adição aos definidos pelo RCS-e.

A aplicação tem por base esta especificação, e quando o estagiário integrou a equipa, a aplicação já cumpria os requisitos necessários para a acreditação e esse processo já estava a decorrer.

### 3.1.2 Hot Fixes

Os RCS-e *joyn Hot Fixes* é um conjunto de melhorias relativas às melhores práticas em termo de apresentação das funcionalidades e serviços do RCS-e. Introduz melhorias tanto a nível da implementação dos serviços, definidas no documento *Hot Fixes Implementation Guidelines 3.0* [33], como da interface, definidas no documento *Hot Fixes UX Guidelines 2.2* [34].

Tem como objectivo assegurar que o design da aplicação proporciona uma melhor experiência de utilização. São estabelecidos vários requisitos obrigatórios que, apesar de manterem a liberdade no design do cliente, asseguram uma experiência consistente para o utilizador e um maior reconhecimento da marca *joyn*.

### 3.1.3 RCS Release 5

Com a criação do RCS-e, teve origem uma divergência nos desenvolvimentos da especificação do *Rich Communication Suite*. Passou assim a existir duas abordagens diferentes para a mesma solução. Por um lado o RCS continua a ser desenvolvido, tendo sido lançadas as especificações das *Releases* 3 [35] e 4 [36], por outro o RCS-e é também melhorado. O *RCS Release 5* [37] baseia-se em ambas as especificações referidas, criando um consenso a nível de funcionalidades e serviços.

A *Release 5* contempla não só todos serviços do RCS-e mas também grande parte da funcionalidade do *RCS Release 4*. São ainda introduzidos três novos serviços: *IP Voice Call*, *IP Video Call* e *Location Share*. Alguns dos serviços já existentes são também sujeitos a melhorias: *File Transfer*, *One-to-One Chat* e *Group Chat*.

O ponto mais fulcral desta especificação é o de conferir ao *Service Provider* a liberdade de escolher que serviços implementar. Existem no entanto dois fatores de critério obrigatório: manter a interoperabilidade dos serviços e implementar o serviço de *Capability Discovery*. Relativamente ao primeiro, quando um serviço é implementado, este tem que manter a interoperabilidade com outros *Service Providers* que implementem o mesmo serviço. O segundo refere-se ao mecanismo que permite ao utilizador saber que contactos disponibilizam serviços RCS e quais são disponibilizados.

### 3.1.4 RCS Release 5.1

A *Release 5.1* [38] é uma iteração da *Release 5*, sendo a especificação mais recente. Esta define novas funcionalidades e procedimentos, bem como a modificação e remoção de alguns existentes.

### 3.1.5 joyn Blackbird

A especificação do *joyn Blackbird* tem por base as especificações RCS 5 e 5.1, acrescentando também os *Hot Fixes* do RCS-e. Esta ainda não é oficial e é a especificação da próxima geração do *joyn*.

## 3.2 Análise de Requisitos

Os desenvolvimentos a efetuar durante o a primeira fase do período de estágio incidiram sobre os RCS *joyn Hot Fixes*, pelo que foi realizada uma análise aos documentos *Hot Fixes Implementation Guidelines 3.0* e *Hot Fixes UX Guidelines 2.2* de modo a obter a lista de requisitos relevantes para os desenvolvimentos realizados.

O mesmo processo foi empregue para o levantamento do requisitos da aplicação para a segunda fase do período de estágio, sendo que o documento analisado foi a especificação do *joyn Blackbird* Versão 0.1.

Além dos requisitos definidos nestes documentos, existem também requisitos que não estão relacionados com qualquer especificação do RCS, mas que são um acréscimo às funcionalidades oferecidas pela aplicação. Estes contribuem para a diferenciação relativamente às restantes aplicações *joyn* do mercado. Este levantamento teve por base as

indicações do *Product Owner* relativamente ao plano de desenvolvimento previsto e a consequente análise do *Product Backlog* do projeto.

Devido ao facto de a metodologia usada na gestão do projeto ser *Scrum*, os requisitos diferenciadores estão sujeitos a alterações, tendo os das especificações RCS uma prioridade mais alta no desenvolvimento.

Mais detalhes relativamente aos requisitos da aplicação podem ser encontrados no Anexo A – Especificação de Requisitos.

### 3.3 RCS-e Joyn Hot Fixes

Os requisitos RCS *joyn Hot Fixes* consistem num conjunto de indicações relativamente às melhores práticas para a apresentação das funcionalidades do RCS-e Versão 1.2. Encontram-se divididos por subsecções, sendo cada relativa a um diferente aspeto funcional da aplicação.

#### 3.3.1 Blocking RCS-e Services

A aplicação tem que permitir ao utilizador bloquear contactos presentes na sua lista nativa. O início de comunicações com contactos bloqueados deve ser notificada ao utilizador e os respetivos mecanismos de desbloqueio devem ocorrer consoante a sua escolha.

ID	Name
FR-RCSE-BC-01	RCS-e services for contacts with blocked RCS-e services
FR-RCSE-BC-02	Group Chat for contacts with blocked RCS-e services

Tabela 4. Requisitos de contactos bloqueados

#### 3.3.2 RCS-e Service Touch Points

A aplicação tem que providenciar um conjunto de ações que assegurem um nível de exposição da marca *joyn* de uma forma não intrusiva. Isto implica a integração de pontos de acesso aos vários serviços *joyn* com as funcionalidades nativas do dispositivo.

ID	Name
FR-RCSE-TP-01	Single notification per Chat conversation
FR-RCSE-TP-02	RCS-e capable contact list
FR-RCSE-TP-03	RCS-e dedicated contact list
FR-RCSE-TP-04	RCS-e alert notifications
FR-RCSE-TP-05	RCS-e call screen

Tabela 5. Requisitos de pontos de acesso à aplicação

### 3.3.3 Chat

É necessário criar uma experiência para o utilizador que seja constante para o serviço de *Chat* entre diferentes aplicações *joyn*. A criação, gestão e finalização de conversas tem que obedecer a critérios específicos e é diferente para o *Chat 1-2-1* e o *Group Chat*.

ID	Name
FR-RCSE-CH-01	Multiple Chat Sessions
FR-RCSE-CH-02	RCS-e Chat message status
FR-RCSE-CH-03	Group Chat
FR-RCSE-CH-04	In/out of coverage Chat experience
FR-RCSE-CH-05	Deleting messages and Chat sessions

Tabela 6. Requisitos do serviço de *chat*

### 3.3.4 File Transfer

É necessário garantir uma coerência funcional para o serviço de *File Transfer* entre as diferentes aplicações *joyn*. As notificações dos eventos do serviço, os comportamentos em relação à ligação de dados devem seguir condutas específicas.

ID	Name
FR-RCSE-FT-01	File Transfer user interaction and presentation
FR-RCSE-FT-02	File Transfer notifications
FR-RCSE-FT-03	RCS-e File Transfer in/out of network coverage

Tabela 7. Requisitos de serviço de *file transfer*

### 3.3.5 In Call Share

A partilha de conteúdos durante uma chamada de voz tem que obedecer a um conjunto de comportamentos específicos. O utilizador tem que ter controlo sobre quando partilhar ou cancelar uma partilha sem que isto cause impacto na chamada corrente.

ID	Name
FR-RCSE-CS-01	In Call Share

Tabela 8. Requisitos do serviço de *content share*

### 3.3.6 Settings

A aplicação deve oferecer ao utilizador um conjunto de opções de configuração consistentes entre aplicações *joyn*.

ID	Name
FR-RCSE-ST-01	RCS-e Settings - management menu/options

Tabela 9. Requisitos de preferências

### 3.3.7 Roaming

O utilizador tem que ter controlo sobre a utilização dos serviços *joyn* enquanto em *roaming*, de modo a prevenir possíveis encargos sobre o plano de dados sem o seu consentimento. A aplicação tem que desativar os seus serviços e estes têm que ser reativados manualmente pelo utilizador.

ID	Name
FR-RCSE-RM-01	Automatic service deactivation/suspension when on roaming

Tabela 10. Requisitos de roaming

### 3.3.8 Multi-device/Multi-client Scenarios

A aplicação tem de ter a capacidade para gerir cenários em que o utilizador possui múltiplos dispositivos *joyn* associados e em que o dispositivo já tem um cliente *joyn* instalado.

ID	Name
FR-RCSE-MM-01	Association of a secondary device to a <i>joyn</i> identity
FR-RCSE-MM-02	Multiple clients on the same device management

Tabela 11. Requisitos de cenários *multi-device* e *multi-client*

## 3.4 *joyn* Blackbird

O *joyn Blackbird* é um subconjunto da versão 5 da especificação RCS que se encontra ainda em desenvolvimento. Devido a este fator, foram apenas selecionados os requisitos da aplicação que se encontram no plano do projeto. São apresentados por subsecções, sendo cada uma relativa a um diferente aspeto de funcionalidade ou serviço.

### 3.4.1 Service Activation

A aplicação deve oferecer a possibilidade de o utilizador se registar no serviço pela primeira vez através de uma ligação Wi-Fi.

ID	Name
FR-RCS-BB-01	Service Activation over non-cellular network

Tabela 12. Requisitos de ativação do serviço

### 3.4.2 Video Share orientation

A aplicação tem que suportar a correção automática da orientação do vídeo recebido através da partilha de conteúdos durante uma chamada. De modo a maximizar a utilização do ecrã do dispositivo, a aplicação deve também suportar alterações na orientação do dispositivo e ajustar o tamanho do vídeo de acordo.

ID	Name
FR-RCS-BB-02	Video Share orientation correction
FR-RCS-BB-03	Video Share device orientation

Tabela 13. Requisitos do serviço de *video share*

### 3.4.3 Location Share

A aplicação tem que oferecer o serviço de partilha de localizações. Esta funcionalidade tem de estar disponível no serviço de *Chat 1-2-1*, *Group Chat* e *Content Share*. O utilizador tem que ter a opção de decidir partilhar a sua localização atual ou uma introduzida por ele.

ID	Name
FR-RCS-BB-04	Share current location
FR-RCS-BB-05	Share place location

Tabela 14. Requisitos do serviço de *location share*

## 3.5 Requisitos Diferenciadores

A aplicação não está limitada aos requisitos das especificações, é importante proporcionar uma experiência rica ao utilizador de modo a estimular o uso frequente da aplicação. Os seguintes requisitos são reflexo dessas funcionalidades extra da aplicação.

### 3.5.1 SMS Inbox

De forma a centralizar as mensagens recebidas, a aplicação tem que integrar o sistema de mensagens SMS nativo. As SMS recebidas têm de ser agregadas ao histórico local de conversas e o utilizador tem de ter controlo para poder desativar esta funcionalidade. A opção de enviar uma mensagem pelo serviço de *Chat* ou por SMS deve estar à disposição do utilizador. Para contactos sem os serviços *joyn*, apenas a segunda opção é providenciada.

ID	Name
FR-DIF-SI-01	SMS composer
FR-DIF-SI-02	Conversation history mashup
FR-DIF-SI-03	Message type filter
FR-DIF-SI-04	Send as SMS

Tabela 15. Requisitos da integração de SMS

### 3.5.2 Content Backup

A aplicação tem que oferecer mecanismos de *backup*, de modo a que o utilizador consiga guardar uma cópia dos seus conteúdos e restaurar os mesmo.

ID	Name
FR-DIF-CB-01	Content Backup
FR-DIF-CB-02	Backup Restore

Tabela 16. Requisitos de backup de conteúdos

### 3.5.3 Messaging

A aplicação tem que oferecer uma experiência de *Chat* de forma não intrusiva e multifuncional. O envio de mensagens em condições de falta de ligação de dados ou à rede tem que ser possível, sendo estas guardadas localmente e enviadas quando as estiverem reunidas as condições para tal. A partilha de imagens também tem que ser enriquecida, oferecendo ao utilizador uma funcionalidade de desenho.

ID	Name
FR-DIF-MS-01	Message Queue
FR-DIF-MS-02	Chat message composer orientation
FR-DIF-MS-03	Drawing functionality

Tabela 17. Requisitos da funcionalidade de *messaging*



## 4. Arquitetura

Este pretende dar a conhecer a arquitetura da aplicação. Na primeira secção são descritas as tecnologias envolvidas na arquitetura, de seguida é descrita uma visão mais geral de todo o sistema, explicando a arquitetura do sistema de comunicação no qual a aplicação está integrada e termina com a arquitetura e descrição dos diferentes componentes da aplicação em si.

### 4.1 Tecnologias

Antes de apresentar a arquitetura da aplicação, é necessário fazer uma breve introdução à plataforma Android e aos protocolos usados.

#### 4.1.1 Android

O desenvolvimento para a plataforma Android envolve a utilização de um conjunto de componente base a partir dos quais uma aplicação é desenvolvida [39].

##### Activities [40]

Uma *activity* representa um ecrã da aplicação e respetiva interface com a qual o utilizador interage. Estas são responsáveis por controlar os conteúdos mostrados e pelas transições entre os diferentes ecrãs da aplicação. Normalmente uma aplicação consiste num conjunto de *activities* que operam de forma independente umas das outras.

##### Services [41]

Um *service* é um componente que corre em background da aplicação cuja principal função é processar operações de longa duração.

##### Content Providers [42]

Um *content provider* permite a armazenamento e gestão de dados da aplicação.

##### Broadcast Receivers [43]

Um *broadcast receiver* é um componente que processa eventos difundidos pelo sistema, outras aplicações ou mesmo pela própria aplicação.

##### Intents [44]

Um *intent* é uma mensagem assíncrona usada na comunicação entre *activities*, *services* e *broadcast receivers*. É identificado por uma determinada ação e pode conter dados adicionais.

### 4.1.2 Protocolos

Relativamente à comunicação entre os diferentes intervenientes num sistema de RCS, esta é realizada recorrendo principalmente aos seguintes protocolos:

#### SIP [45]

O *Session Initiation Protocol* é um protocolo de comunicações direcionado para a criação, gestão e finalização de sessões com um ou mais utilizadores. Apesar da sua aplicação mais conhecida ser num contexto de comunicações VoIP, é a o protocolo base para todas as comunicações RCS. Os principais pontos a destacar relativamente ao SIP são:

- definir uma localização de um utilizador, atribuindo-lhe um nome e fazendo a tradução para um endereço de rede;
- Permitir a negociação de serviços entre os participantes de uma sessão, de modo a identificar e a acordar os serviços disponíveis entre eles.

Devido ao seu design modular possibilita a integração com outros protocolos como SDP, RTP, entre outros.

#### SDP [46]

O *Session Description Protocol* é usado para definir os parâmetros de inicialização de uma sessão de comunicação multimédia. É responsável pela negociação do tipo de conteúdos, formato e todas as propriedades a ele associadas.

#### MSRP [47]

O *Message Session Relay Protocol* é o protocolo usado para a transmissão de dados entre utilizadores ligados à mesma sessão. É responsável pela comunicação em certos serviços do RCS, como o *Chat*, *File Transfer* e *Image Share*.

#### RTP [48]

O *Real-Time Transport Protocol* é um protocolo usado na transmissão de áudio e vídeo em redes IP. É usado em sistemas de comunicação e entretenimento que envolvem *streaming* de conteúdos como vídeo. É o protocolo usado no serviço de *Video Share* do RCS.

## 4.2 Visão geral

Para os clientes RCS comunicarem entre si, a operadora que fornece o serviço necessita de respeitar certos requisitos a nível da sua rede. O diagrama da figura 3 ilustra, de uma forma geral, o sistema que permite essa comunicação.

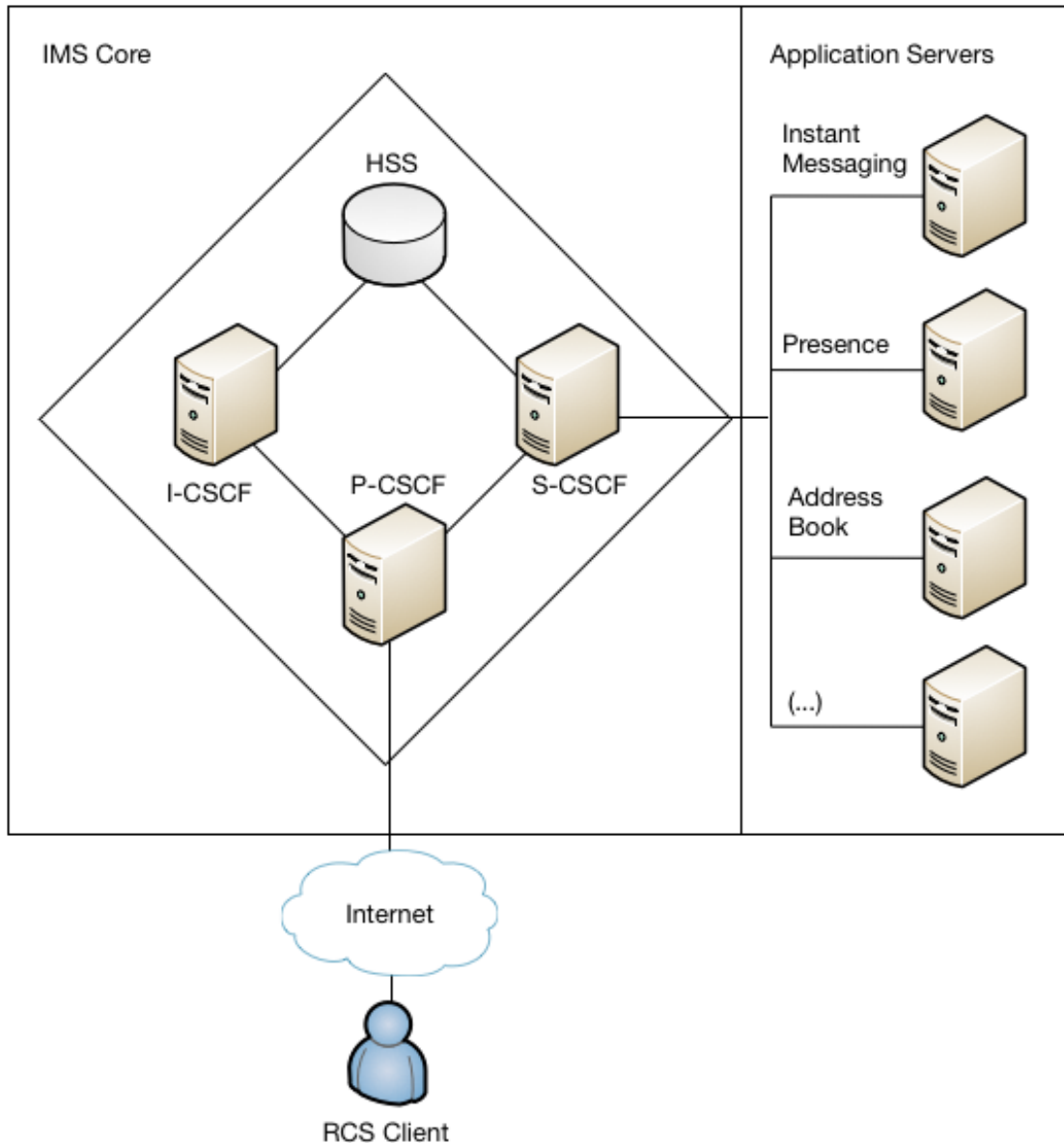


Figura 3 - Diagrama de comunicação de um sistema RCS

Neste diagrama é possível identificar os três componentes principais do sistema: o *IMS Core*, os *Application Servers* e o cliente RCS.

#### 4.2.1 IP Multimedia Core Network Subsystem

O *IMS Core* apresenta dois componentes principais, o HSS e o CSCF.

O *Home Subscriber Server* (HSS) é responsável por manter a informação relativa a configurações e gestão de identidades dos utilizadores. Em sistemas com serviços de presença ou localização, é também responsável por manter informação relativa ao estado do utilizador.

O *Call Session Control Function* (CSCF) por sua vez é um conjunto de servidores SIP ou proxies:

- P-CSCF – o qual é o primeiro ponto de contacto entre o cliente RCS com o IMS. É responsável por assegurar a segurança das mensagens trocas entre o sistema e o cliente e por alocar recursos para sessões de comunicação.
- I-CSCF – é o primeiro ponto de contacto entre redes IMS distintas. É responsável por consultar o HSS a fim de escolher o S-CSCF para um determinado utilizador.
- S-CSCF – é o principal componente do CSCF, sendo responsável pelo processamento da atividade dos utilizadores, mantendo registos relativos a localização e autenticação do utilizador e pelo processamento de pedidos, fazendo o redireccionamento para o *Application Server* correspondente.

### 4.2.2 Application Servers

Os *application servers* são os responsáveis por alojar e executar serviços. Num contexto de RCS, são responsáveis por serviços como os de *Instant Messaging*, Presença e Network Address Book.

### 4.2.3 Cliente RCS

E por último o cliente RCS, o qual é o ponto de partida para todas as comunicações e partilha de conteúdos.

## 4.3 Arquitetura da aplicação

A aplicação pode ser dividida em dois componentes principais: a camada lógica e a Camada Gráfica. Ambos são desenvolvidos para a plataforma Android e o nível da API mínimo suportado é a 8 – Android 2.2.x, nome de código FROYO [49].

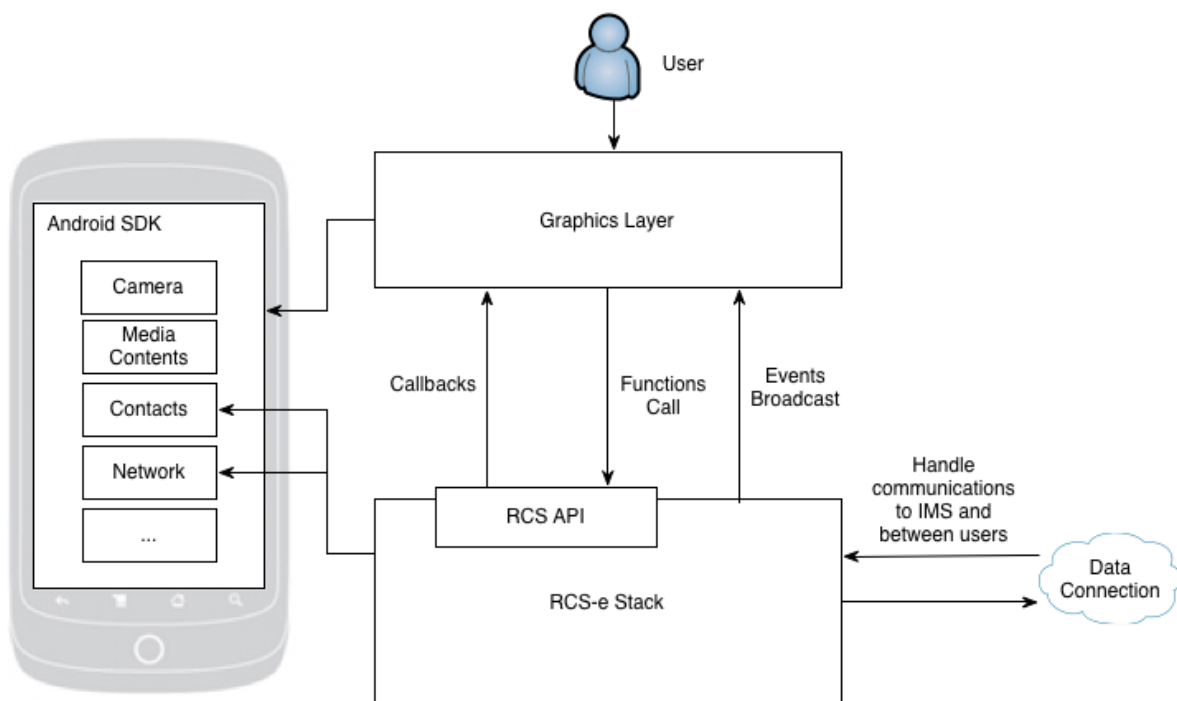


Figura 4 - Arquitetura de alto nível

Como é ilustrado na figura 4, ambos fazem uso do SDK do Android [50], sendo que, a nível do acesso a conteúdos e funcionalidades do dispositivo, a stack apenas acede à gestão de contactos e comunicações, enquanto que a camada gráfica é responsável por gerir todos os restantes acessos a dados necessários à interface da aplicação ou a serviços, como o de partilha de ficheiros ou vídeo.

A camada gráfica é o ponto de início para qualquer tipo de comunicação originada pelo utilizador e a *stack* o ponto de início para qualquer tipo de comunicação recebida, como é possível verificar nos diagramas anteriores. Uma vez que a camada gráfica e a *stack* correm em processos diferentes, apenas a *stack* é inicializada com o *boot* do dispositivo. Desta forma são poupados recursos uma vez que a camada gráfica apenas é inicializada quando o utilizador recebe ou pretende iniciar algum tipo de comunicação.

### 4.3.1 RCS-e stack

A RCS-e *stack* é um projeto *open-source*, desenvolvido pela Orange Labs [51], de uma implementação dos atuais standards de comunicação do RCS-e para a plataforma Android. Esta disponibiliza todos os serviços de RCS-e, incluindo os *Hot Fixes* e *Social Presence* e encontra-se em desenvolvimento do suporte para o *BlackBird*.

#### 4.3.1.1 Clientes RCS

A aplicação é desenvolvida tendo por base esta *stack*, sendo que o estagiário não teve um papel ativo na avaliação e escolha desta, uma vez que a aplicação já se encontrava em desenvolvimento quando este integrou a equipa. No entanto é importante identificar as alternativas existentes no mercado.

Empresa	Nome do Cliente	Tipo de Acreditação	Nível de Acreditação
WIT-Software	WIT Mobile Communicator for Android	Full	joyn Hot Fixes
Neusoft Mobile Solutions Oy	Silta RCS-e client for Android	Full	joyn Hot Fixes
Nable Communications, Inc	Nable RCS-e Client for Android	Full	joyn Hot Fixes
Orange France	joynApp	Provisional	joyn Hot Fixes

Tabela 18 - Comparação de clientes RCS

Como é possível verificar na tabela 18, existem vários clientes disponíveis no mercado e todos oferecem o nível de acreditação completo dos *joyn Hot Fixes* [52]. À exceção do joynApp, a qual tem por base a *stack* RCE-s em questão, disponibilizada sob uma licença

*open-source*, todos os clientes são produtos das empresas apresentadas. Este aspeto dificulta a obtenção de informações mais concretas como a arquitetura, daí apenas ter sido usada a acreditação como vector de avaliação.

#### 4.3.1.2 Desenvolvimento

Visto ser um projeto em desenvolvimento, nem sempre estão disponíveis todas as funcionalidades necessárias ou estas não estão implementadas totalmente. Cabe à equipa de desenvolvimento identificar estes casos, encontrar soluções e proceder à sua implementação. Isto implica um estudo detalhado da especificação RCS-e antes de proceder com quaisquer modificações.

Devido ao seu carácter *open-source*, muitas das alterações feitas pela a equipa são contribuídas para o projeto. A política de contribuição reflete-se nos seguintes aspetos:

- Alterações que sejam fulcrais para o correto funcionamento dos serviços RCS são comunicadas à equipa de desenvolvimento da Orange, sendo que a decisão da integração das mesmas numa futura versão, está sempre do lado da referida.
- Alterações que acrescentem valor funcional à aplicação não são contribuídas, sendo mantidas durante o processo de atualização da *stack* para versões mais recentes.

Uma vez que a *stack* é usada em mais aplicações além da desenvolvida neste projeto, estas contribuições são um factor de suporte ao carácter de esforço conjunto das organizações para a introdução do *joyn* no mercado o mais rapidamente possível.

#### 4.3.1.3 Arquitetura

Relativamente à sua arquitetura, esta está implementada baseada em serviços. São estes que proporcionam a funcionalidade da *stack* e uma vez que correm em background, não interferem com a performance da camada gráfica.

Para aceder a toda a funcionalidade da *stack*, esta fornece uma API de alto-nível. Esta está implementada com base em AIDL [53] e *Intents*, responsáveis por providenciar o acesso aos dados e funcionalidades da *stack* e fornecerem informação relativa a eventos, respectivamente. Esta API oferece acesso às seguintes funcionalidades:

- *Capability API* – acesso à funcionalidade de *Capability Discovery*
- *Contacts API* – gestão de contactos e integração com a lista nativa
- *Presence API* – gestão da informação de *Social Presence* e suas funcionalidades
- *Rich Call API* – acesso às funcionalidades de *Video Share* e *Image Share* durante uma chamada de voz
- *Messaging API* – acesso aos serviços de *Chat 1-to-1*, *Group Chat* e *File Transfer*
- *Media API* – acesso a funcionalidades de reprodução e renderização de vídeo
- *Events Log API* – gestão do histórico de conversas, transferência de ficheiros e partilha de conteúdos durante chamadas
- *RCS Settings* – acesso à base de dados das preferências da aplicação e do utilizador

A utilização dessa API segue o mesmo padrão de implementação para a comunicação com as diferentes funcionalidades, pelo que na figura 5 é representada, de uma forma genérica, a comunicação entre a Camada Gráfica e a *stack*.

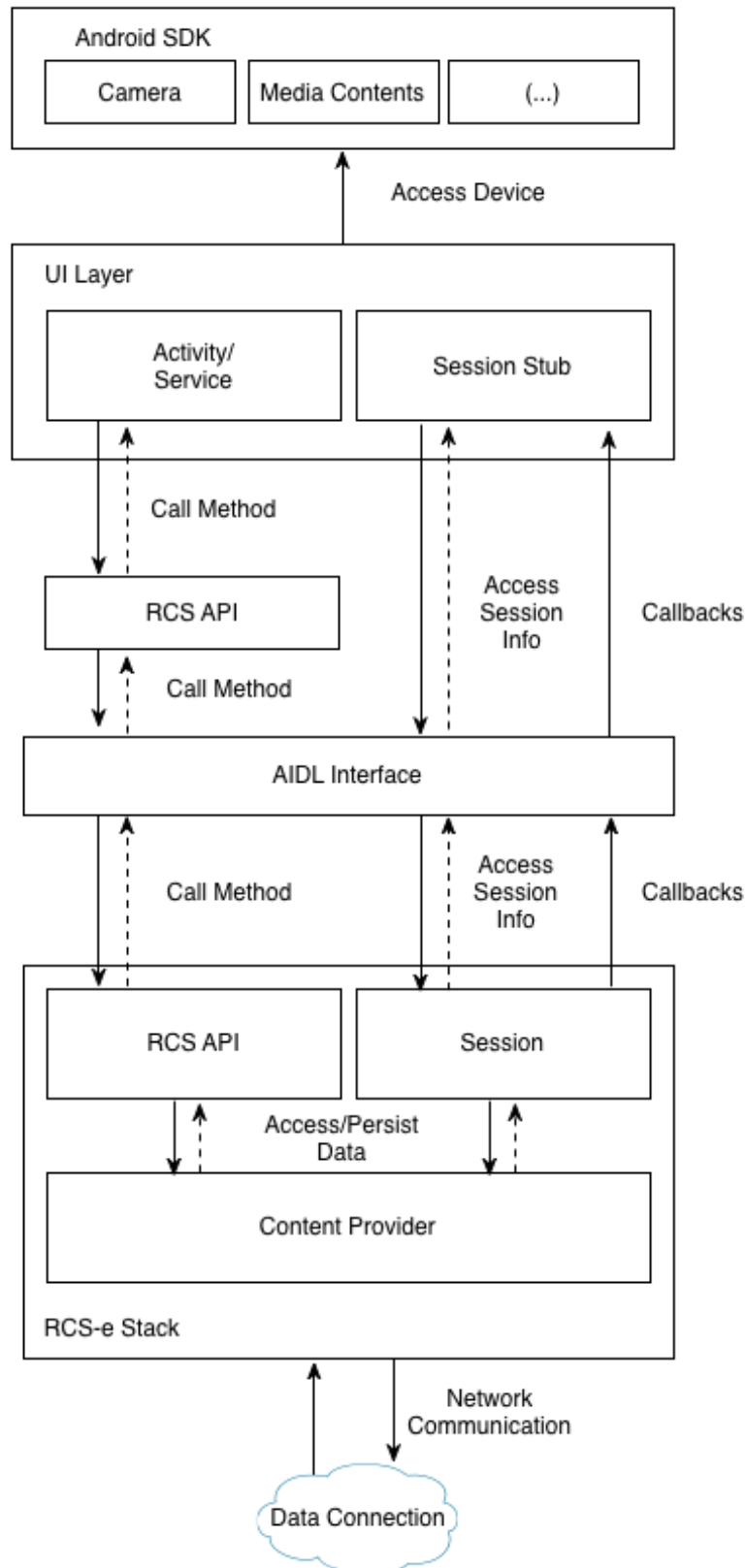


Figura 5 - Diagrama de comunicação entre a Camada Gráfica e a *stack*

A *stack* é também responsável pelo armazenamento de todos os dados da aplicação. Para isso dispõe de um conjunto de *Content Providers* que, em conjunto com os serviços, persistem toda a informação relativa à configuração das comunicações e aos próprios conteúdos do utilizador.

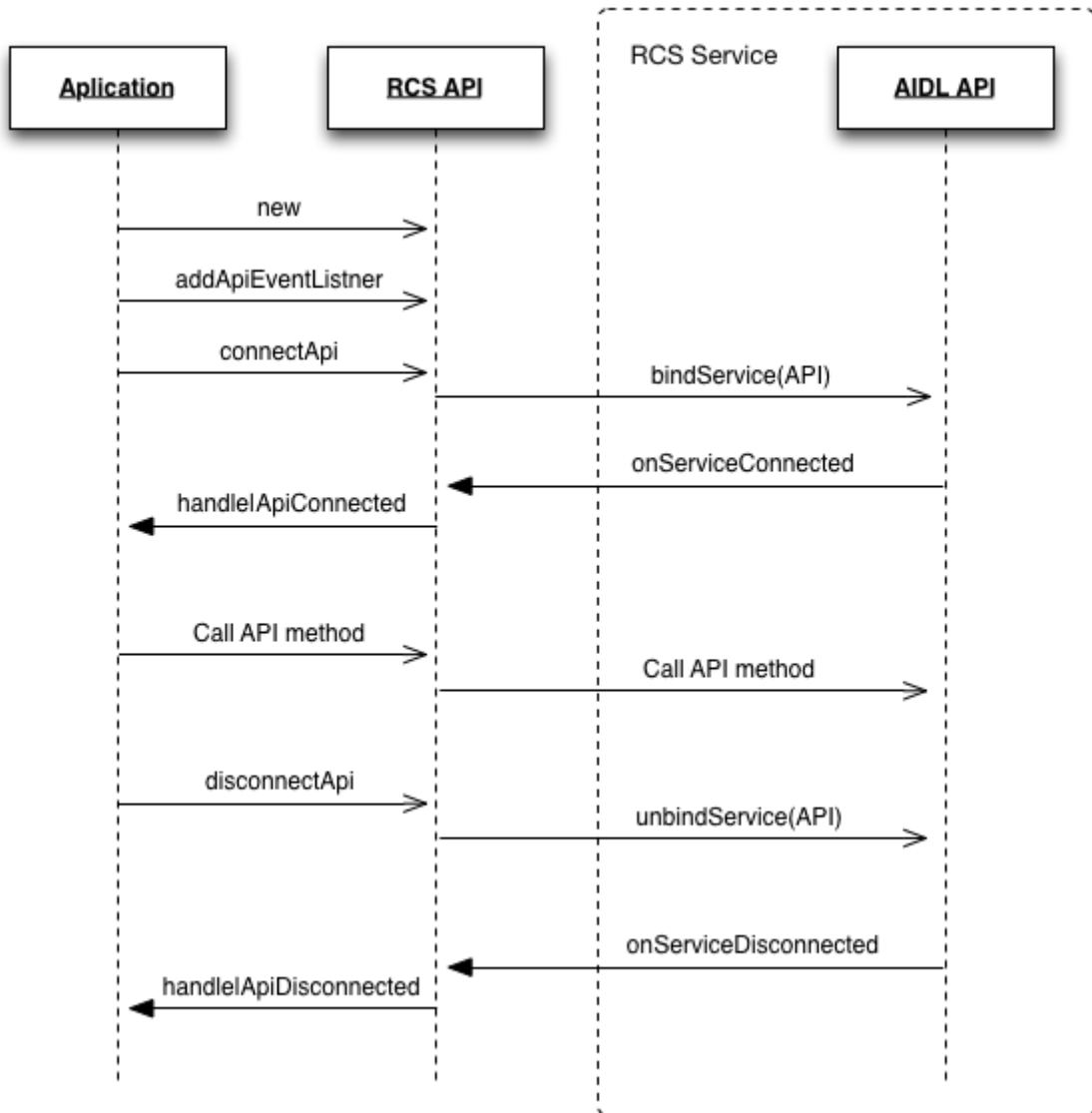


Figura 6. Diagrama de sequência da RCS API

O diagrama da figura 6 representa o processo genérico do uso das interfaces oferecidas pela API. A camada gráfica tem que criar uma nova instância do objecto da API em questão e esperar pelo *callback* do serviço que a notifica da disponibilidade para fazer os pedidos pretendidos.



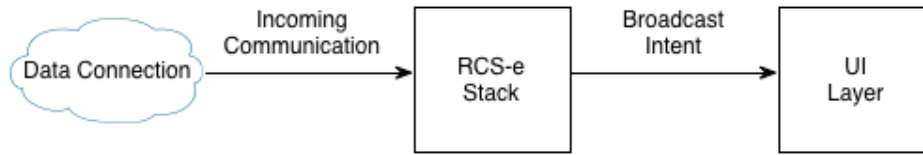


Figura 7 - Diagrama de difusão de Intents

A nível da recepção dos *Broadcast Intents* emitidos pela *stack*, é da responsabilidade da camada gráfica registar os *Intents* que pretende receber, os quais são fornecidos pela API. Quando certos eventos ocorrem, por exemplo a chegada de uma resposta de um pedido de *capabilities*, a *stack* transmite o respectivo *Intent* e a camada gráfica está encarregue de lidar com este, como é ilustrado na figura 7.

### 4.3.2 Camada Gráfica

É na camada gráfica que incidem a maioria dos desenvolvimentos da equipa de desenvolvimento. Esta é responsável por toda a interação do utilizador com a aplicação. Na figura 8 é ilustrada de uma forma genérica as interações e comunicações entre diferentes componentes.

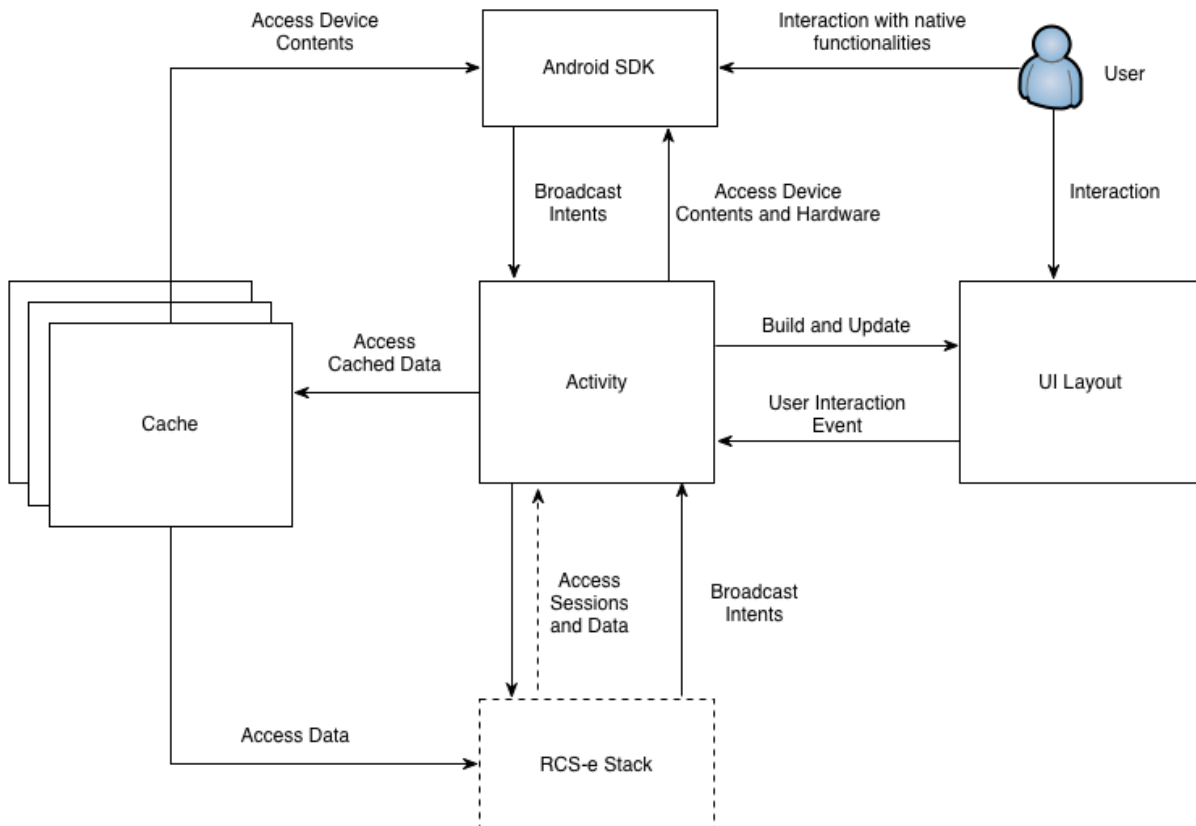


Figura 8 – Comunicação entre uma *Activity* e os restantes componentes

Tanto a lógica de obter os dados quer da *stack*, quer do dispositivo, bem como a alteração dinâmica dos conteúdos visualizados é realizada pelas *Activities*. Estas podem ser inicializadas ou por alguma interação do utilizador com a aplicação ou pela difusão de *Intents*, os quais têm origem na *stack* ou em funcionalidades nativas, como a partilha de imagens diretamente da galeria.

De modo a reduzir os tempos de carregamento de dados, estão implementadas algumas caches que mantêm informação relativa a contactos, histórico de mensagens, entre outras. Estas são inicializadas com o arranque da aplicação e mantêm-se sempre atualizadas pela recepção de *Intents* ou por ações do utilizador.

O UI *Layout*, para além de mostrar os conteúdos ao utilizador, têm um papel fulcral em todo o processo de interação da aplicação. Todas as interações do utilizador com o diversos elementos da interface dão origem a eventos, os quais são captados pelas *Views*. O respectivo resultado dessas interações é transmitido e processado pelas *Activities*, originando atualizações na UI corrente ou a criação e transição para uma nova *activity*.

A implementação com recurso às funcionalidades nativas do Android como o *Share* ou a inserção de informação e botões de atalho na lista de contactos, leva à criação de pontos de entrada para a aplicação e conseqüentemente a uma integração mais nativa, um dos pontos chave do *joyn*.

Um dos factores cruciais desta camada é a performance. Tempos de espera elevados para carregamento de dados ou operações pesadas que consumam demasiados recursos do dispositivo, levam a paragens no normal funcionamento da aplicação. Isto conduz a uma experiência deteriorada para o utilizador, o que, como consequência, leva a que este perca o interesse pela aplicação e a deixe de utilizar.

### 4.3.3 Bibliotecas externas

#### **Lib Phone Number [54]**

Sendo a área do processamento de contactos uma área com alguma complexidade, o que requeria um grande investimento de tempo por parte da equipa de desenvolvimento, é usada a *libphonenumber*. Esta biblioteca da Google fornece um vasto número de funcionalidades para o tratamento e validação de contactos telefónicos.

#### **ActionBarSherlock [55]**

Sendo que a *ActionBar* [56] do Android apenas disponível com a versão 4.0 do sistema operativo (Ice Cream Sandwich) [57], esta biblioteca oferece suporte para versões 2.x e superiores. Em versões 4.0 ou superiores, a biblioteca usa diretamente a *ActionBar* nativa. Para versões inferiores a biblioteca implementa uma réplica da *action bar* nativa, baseando-se na sua implementação e recursos.

## 5. Controlo de Qualidade

Neste capítulo é apresentado o plano de controlo de qualidade da aplicação. Relativamente aos testes realizados na aplicação, estes podem ser divididos em dois diferentes tipos: testes necessários para obter a acreditação da GSMA e testes executados durante o desenvolvimento da aplicação. Os seguintes pontos apresentam estes dois tipos.

### 5.1 Testes de Acreditação

Tratando-se de uma aplicação de RCS, os casos de teste mais relevantes ao desenvolvimento são os definidos pela GSMA. É a aprovação nestes testes que confere à aplicação determinado nível de acreditação.

A aplicação já está acreditada pela GSMA com o nível *joyn*, o que significa que reúne os requisitos descritos na especificação RCS-e 1.2.2. Para lhe ser atribuído o nível de *joyn Hot Fixes*, a aplicação tem que passar em todos os casos de teste definidos nos seguintes documentos:

- *Technical Test Case matrix* [58]
- *UX Test Cases Matrix* [59]
- *Quality check list* [60]

Estes testes de acreditação são realizados pelos *testers* da equipa, sendo eles os intervenientes na preparação e execução do processo de acreditação.

### 5.2 Testes de Desenvolvimento

Um dos desafios no desenvolvimento para Android é assegurar que a aplicação apresenta o mesmo comportamento e aspeto independentemente do dispositivo em que se encontra a correr. Os principais fatores que contribuem para isto são:

- A existência de uma grande variedade de dispositivos Android no mercado;
- A fragmentação da plataforma Android;
- Os OEMs desenvolverem e manterem uma versão customizada do sistema operativo para os seus dispositivos.

Com base nestes pontos, os *developers* não conseguem garantir que as suas implementações funcionem em todos os dispositivos existentes. Durante o desenvolvimento da aplicação, é da responsabilidade dos dois *testers* que integram a equipa assegurar que novas funcionalidades ou correções de *bugs* produzem o resultado esperado num vasto leque de dispositivos.

## 6. Trabalho Desenvolvido

Neste capítulo são expostos os desenvolvimentos efetuados na aplicação durante o primeiro semestre do período de estágio. Estes encontram-se divididos pelos diferentes serviços RCS ou aspetos sobre os quais incidiram.

### 6.1 Implementação

A tabela 19 representa a correspondência entre o trabalho desenvolvido e os requisitos funcionais definidos no capítulo 3.

Serviço/ Funcionalidade	Trabalho desenvolvido	Requisitos relacionados
<b>Contactos</b>	O mesmo número de telefone em diferentes contactos	FR-RCSE-TP-05 FR-RCSE-TP-05
	Contactos com múltiplos números de telefone	FR-RCSE-TP-05 FR-RCSE-TP-05
<b>Content Sharing</b>	Call on Hold	FR-RCSE-TP-10
	Video Share em Landscape	FR-RCSE-CS-01 FR-RCS-BB-03
<b>File Transfer</b>	File Transfer Auto-Accept	FR-RCSE-FT-01
	File Transfer thumbnails	FR-RCSE-BB-06
<b>Messaging</b>	Separação entre 1-to-1 e Group Chat	FR-RCSE-CH-03
	Sair explicitamente de um Group Chat	FR-RCSE-CH-03
	Definição de Alias	FR-RCSE-CH-03
	Serviços com contato bloqueado	FR-RCSE-FS-01
	Group Chat com contactos bloqueados	FR-RCSE-FS-02
	Redesign do ComService	FR-RCSE-FS-01 FR-RCSE-FS-02 FR-RCSE-TP-04 FR-RCSE-TP-09 FR-RCSE-CH-01 FR-RCSE-CH-02 FR-RCSE-CH-03

		FR-RCSE-CH-04 FR-RCSE-CH-05 FR-RCSE-FT-02 FR-RCSE-FT-03
	Funcionalidade de Desenho	FR-DIF-MS-03
<b>Preferences</b>	Preferences Design	FR-RCSE-ST-01 FR-RCSE-RM-01
<b>Interface Geral da Aplicação</b>	Integração da ActionBar	

Tabela 19 - Correspondência entre trabalho desenvolvido e requisitos

## 6.2 Contactos

As implementações a nível dos contactos incidiram sobre a situação de o mesmo número poder estar em mais do que um contacto, a situação de um contacto poder ter mais do que um número de telefone associado e a integração com a lista nativa.

### 6.2.1 O mesmo número de telefone em diferentes contactos

Relativamente à situação de um mesmo número poder estar em mais do que um contacto, o problema recaía sobre a decisão de qual dos nomes apresentar para esse ao número. Este problema contemplava a recepção de comunicações e a apresentação de histórico de conversas.

A solução desejada passava por apresentar um comportamento a que o utilizador já estivesse habituado, o nativo do seu dispositivo. Posto isto, começou por ser feita uma análise a uma variada seleção de dispositivos, de diferentes fabricantes e versões do sistema operativo. Foi testado o comportamento tanto para chamadas como para mensagens.

Desta análise conclui-se que o comportamento não era sempre constante. Uns dispositivos apresentavam o último contacto inserido com o número em questão, outros o que tinha sido inserido em primeiro e outros o que tinha a posição mais cimeira na ordem alfabética. Estes diferentes comportamentos devem-se ao facto de cada fabricante ter a sua própria implementação da gestão dos contactos.

Após discussão com o *Product Owner* e o designer, a solução mais consensual encontrada foi excluir a apresentação de qualquer nome de contacto e identificar as comunicações diretamente pelo número de telefone.

O nome do contacto era obtido diretamente da cache de contactos implementada na aplicação. Esta estava organizada em função dos identificadores da lista nativa, pelo que foi criada uma organização paralela, esta em função de números de telefone. Desta forma, a verificação dos contactos a que certo número está associado é feita rapidamente, e a obtenção de nomes de contactos otimizada.

Foi também necessário modificar as referências dos contactos nas várias atividades para passarem a ser em função do número de telefone e não em função do ID da lista nativa.

### 6.2.2 Contactos com múltiplos números de telefone

Quanto à situação de contactos com múltiplos números de telefone, o problema passava pela apresentação destes na lista de contactos da aplicação. O contacto aparecia repetido, mas cada entrada na lista apresentava um número de telefone diferente.

A solução passou por criar apenas uma entrada na lista sem qualquer número de telefone apresentado. Quando o utilizador pressiona no contacto, uma janela de diálogo é apresentada com uma lista dos vários números.

É de salientar que quando o utilizador pressiona num contacto da lista, é feito um pedido de *capabilities* para este. Este comportamento foi mantido para os contactos dentro da janela de diálogo.

## 6.3 Content Sharing

A nível da partilha de conteúdos durante uma chamada, foi desenvolvido trabalho em dois aspectos: a identificação de uma chamada em espera e a orientação do ecrã do dispositivo quando está a ser partilhado vídeo em tempo real.

### 6.3.1 Call on Hold

A identificação de uma chamada em espera apresentava-se como um desafio para o programador e não como uma simples chamada à API do Android.

A API do Android apenas permite identificar uma chamada por apenas três estados: a chamar, ativa e inativa. Não há acesso ao estado de ‘em espera’.

A investigação da solução passou por várias alternativas:

- Uso de *reflection* para aceder às classes internas do sistema – sem sucesso, visto que para obter a informação pretendida é necessário aceder ao processo sobre o qual o sistema nativo está a correr e por razões de segurança não é fornecido acesso.
- Intercepção das trocas de mensagens na *Radio Interface Layer* – sem sucesso, visto que é a comunicação entre a RIL e o sistema operativo é feita através de um *daemon*. Isto implicava a própria implementação desse *daemon*.
- Leitura dos rádio *logs* – este foi o único caso em que foi obtido algum sucesso. Os *logs* de comunicação são internos e independentes dos *logs* de sistema e os referentes ao estado das chamadas eram constantes entre a grande variedade de dispositivos testada. Com a versão *Jellybean* do Android a permissão de leitura desses *logs* foi revogada, invalidando a solução.

Não foi encontrada uma solução válida, pelo que a funcionalidade não foi implementada.

### 6.3.2 Video Share em Landscape

A aplicação apenas suportava *Video Share* com a orientação do dispositivo em *portrait*.

A solução passou por criar um *layout* específico para a orientação em *landscape*. A mudança entre os dois tipos fica ao encargo do sistema operativo.

Foi também implementado um novo design para os botões de opções e o cabeçalho identificativo do contacto com o qual se encontra em chamada.

A interação com o utilizador foi também modificada, sendo que o cabeçalho e os botões passaram a esconder-se automaticamente passados 8 segundos com uma animação de desvanecimento e voltam a aparecer com o pressionar de qualquer área do ecrã.

Para o caso de ambos os utilizadores estarem a partilhar vídeo, o vídeo recebido é mostrado em fundo e o transmitido numa pequena janela sobreposta com o primeiro no canto do ecrã. Foi necessário implementar uma animação para essa janela deslizar para o topo do ecrã quando o cabeçalho e menu são escondidos.

Visto que estas estão a renderizar vídeo em tempo real, tratam-se de umas *Views* não convencionais. Foi encontrado o problema de estas não poderem ser animadas. A solução encontrada passou por:

- Criar uma imagem com o último *frame* de vídeo mostrado antes de iniciar a animação
- Rodar essa imagem horizontalmente caso a câmara usada seja frontal
- Sobrepor o vídeo com essa imagem e esconder a *View* de transmissão
- Animar essa imagem a deslizar para o topo do ecrã
- Alterar a posição da *View* de transmissão de vídeo para o topo do ecrã
- Esconder a imagem do último *frame*

Para o aparecimento do cabeçalho e menu, é reproduzido o mesmo processo, mas na direção oposta.

## 6.4 File Transfer

Os desenvolvimentos na transferência de ficheiros estão relacionados com a funcionalidade aceitar automaticamente as transferências e da gestão de *thumbnails* de convites *File Transfer*.

### 6.4.1 File Transfer Auto-Accept

O *File Share Auto-Accept* implicou a alteração do ecrã de preferências e da alteração do comportamento na *stack* RCS-e.

No ecrã de preferências foi necessário inserir uma nova opção para ativar e desativar esta funcionalidade. Esta opção tinha que se refletir na *stack*, visto que é esta que controla a recepção das comunicações e consequentemente as transferências de ficheiros.

A *stack* não estava preparada para guardar esta preferência do utilizador, pelo que foi adicionada uma nova. Isto implicou uma alteração na base de dados das preferências da *stack* e consequente definição dos métodos para o acesso ao seu valor.

Esse valor passou assim a ser alterado na *stack* consoante o utilizador o altera no ecrã de preferências.

Outro problema na *stack* foi o facto de esta apenas estar preparada para esperar a confirmação da transferência por parte do utilizador. Foi necessário alterar esse comportamento de modo a que, quando chega um pedido de *File Share*, é verificado o valor da preferência do utilizador. Caso esta esteja ativa, a transferência é iniciada de imediato.

#### 6.4.2 File Transfer thumbnails

Quando um convite para a partilha de um ficheiro é recebido, se este for uma imagem, é acompanhado de um *thumbnail*. Este tem que ser mostrado na interface, possibilitando ao utilizador visualizar, ainda que com baixa qualidade, a imagem que está a receber.

De modo a manter a interface consistente, para transferências falhadas ou rejeitadas é mostrado o *thumbnail* em vez da imagem por defeito. Para isso foi necessário guardar os *thumbnails* recebidos no *SDcard* e fazer a sua gestão, apagando-o se a transferência for bem sucedida.

Isto levou a que essas imagens passassem a aparecer na Galeria do dispositivo, o que não era desejado. Possíveis soluções para o problema passavam por esconder a pasta ou adicionar um ficheiro com o nome '*nomedia*' de modo a que a indexação da Galeria ignorasse os conteúdos da diretoria. Nenhuma destas soluções foi considerada válida visto que em certos dispositivos os *thumbnails* continuavam a aparecer na Galeria. A solução encontrada passou por guardar os ficheiros com uma extensão diferente, tendo isto em conta tanto escrita como na leitura dos ficheiros.

### 6.5 Messaging

A nível do serviços de *Chat 1-to-1* e *Group Chat*, os desenvolvimentos incidiram sobre a implementação do suporte para *Group Chat*, o início de serviços RCS com contactos bloqueados, uma nova funcionalidade de desenho e um *redesign* do serviço de gestão de *chats* e *file transfers*.

#### 6.5.1 Separação entre 1-to-1 e Group Chat

Com a introdução do serviço de *Group Chat* na aplicação, foi necessário fazer a separação entre este e o *1-to-1 Chat*. Já existia um implementação parcial de conversas de grupo, mas esta além de não estar devidamente implementada, estava integrada na mesma *Activity* do *Chat 1-to-1*.

Uma vez que certas funcionalidade são comuns aos dois serviços, a solução passou por criar uma classe abstracta da qual ambos *Chat 1-to-1* e *Group Chat* são subclasses. A implementação de uma interface não se justificava, uma vez que certos métodos são implementados da mesma maneira para os dois casos.

Todo este processo levou também a uma *refactoring* do *Chat 1-to-1*, de modo a simplificar algumas implementações.



### 6.5.2 Sair explicitamente de um Group Chat

O serviço de *Group Chat* tem que disponibilizar ao utilizador a opção de sair explicitamente dele. Isto implica que a conversa fica fechada e o utilizador apenas pode consultar o histórico local existente. Já não são recebidas notificações de novas mensagens ou eventos.

A solução passou por introduzir uma nova opção no menu de contexto do ecrã de *Group Chat*. Quando selecionada, o *Group Chat* é fechado, sendo o utilizador remetido para o histórico de conversas. Aí é possível verificar que a conversa está encerrada pelo ícone que é apresentado no canto superior direito da entrada na lista.

Quando o utilizador seleciona essa conversa, é-lhe apresentado o ecrã de *Group Chat*, mas este possui todas as opções de comunicação marcadas como indisponíveis, ficando este limitado à consulta, cópia ou remoção de mensagens do histórico.

### 6.5.3 Definição de Alias

Quando um utilizador não está presente na lista de contactos de outro com o qual pretende estabelecer uma comunicação, é identificado pelo seu *Alias*. Quando definido, o receptor vê esse contacto identificado por um nome e não um de telemóvel.

A definição deste *Alias* foi implementada em dois sítios distintos:

- Como preferência do utilizador
- Diretamente no ecrã de *Group Chat*, através de uma opção no menu de fundo

Em ambas as situações o comportamento é exatamente o mesmo, sendo apresentada um janela de diálogo ao utilizador com uma área de introdução de texto limitada a 20 caracteres.

Definido o *Alias*, este passa a ser enviado nos pedidos SIP do utilizador.

### 6.5.4 Serviços com contato bloqueado

Relativamente ao início de serviços RCS-e com um contacto bloqueado, o utilizador tem que o desbloquear de modo a proceder com o serviço pretendido.

A solução passou por implementar uma janela de diálogo para informar o utilizador que o contacto em questão está bloqueado. Nesta é oferecida a opção de desbloqueio que, caso seja a selecionada, desbloqueia o contacto e inicia o serviço que disparou este diálogo.

Uma vez que o ecrã de *Chat 1-to-1* possui vários pontos de acesso a serviços RCS-e, para cada um deles é verificado o estado do contacto e mostrado a janela de diálogo se assim se justificar.

### 6.5.5 Group Chat com contactos bloqueados

Quanto ao início de um *Group Chat* com contactos bloqueados, tal como no início de um *Chat 1-to-1* para esse caso, descrito no ponto 5.4.1, o utilizador tem que desbloquear esse(s) contacto(s) para proceder com o serviço.

Uma vez que a seleção de contactos para *Group Chat* é feita na lista de contactos da aplicação, uma verificação aos contactos selecionados é realizada. Caso se encontrem contactos bloqueados nessa seleção, é apresentada uma janela de diálogo ao utilizador a informar desta situação e a oferecer a opção de desbloquear o(s) contacto(s) em questão.

Consoante a escolha do utilizador, dois resultados são possíveis:

- Aceita desbloquear – os contactos são desbloqueados e tem início um *Group Chat*.
- Cancela – o *Group Chat* não é iniciado e os contactos bloqueados são removidos da seleção, mantendo os restantes.

### **6.5.6 Redesign do ComService**

Um dos principais elementos da gestão de comunicações da aplicação é o serviço de gestão de sessões de *chat* e *file transfer* – o ComService. Com a adição do suporte para *Group Chat* e os constantes desenvolvimentos no suporte para o serviço de *Chat 1-to-1* e *File Transfer*, a sua arquitetura revelou-se desadequada e sem qualquer controlo na gestão dos recursos dos dispositivos, pelo que foi necessário projetar e implementar uma nova arquitetura. Na figura 9 é apresentada a arquitetura implementada.

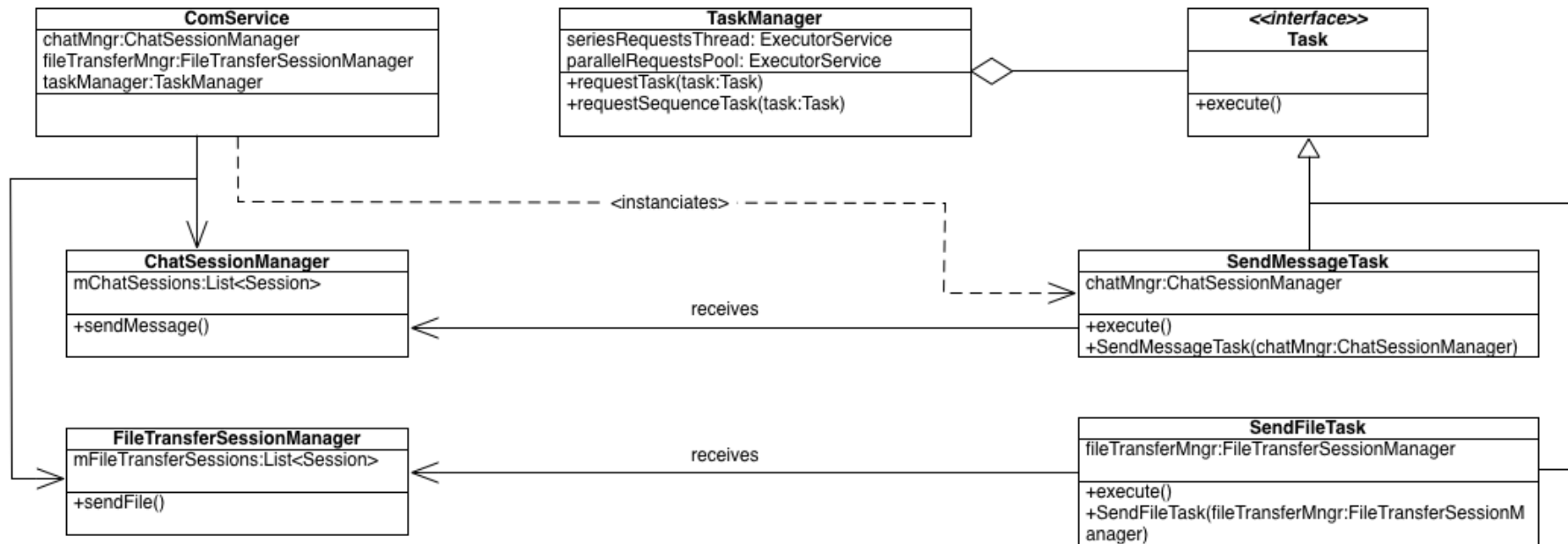


Figura 9 - Diagrama de classes do serviço de gestão de chat e file transfer

Uma vez que a principal função do ComService é não só processar os pedidos das diferentes *Activities* envolvidas no *Chat 1-to-1*, *Group Chat* e *File Transfer*, mas também notificar as mesmas de eventos provenientes da *stack*, a arquitetura definida baseou-se na Command Pattern. Quando um novo pedido chega ao ComService, este instancia o correspondente comando (Task) e submete-o para o *invoker* (TaskManager) a fim de ser processado. Uma vez que em alguns casos é necessário processar os comandos de forma sequencial, como por exemplo no envio de mensagens de *chat*, a classe TaskManager dispõe de uma *thread* só para esse tipo de processamento.

As classes ChatSessionManager e FileTransferManager são os *receivers*, sendo estas que executam as operações pedidas. Ambas são responsáveis por gerir as sessões de *chat* e *file transfer*, respectivamente. Estas classes já existiam, estando implementadas como *singletons*, pelo que foi feito um *refactoring* de modo a serem apenas instanciadas apenas pelo ComService. Isto levou também a um *refactoring* de todas as classes com acessos diretos a estes *singletons*, passando a serem tratados como pedidos ao ComService.

A forma de fazer pedidos ao ComService foi também redesenhada, passando as *Activities* a ter que se ligarem ao serviço antes de poderem fazer qualquer pedido. O processo desde a ligação até à chamada de algum método é descrito na figura 10.

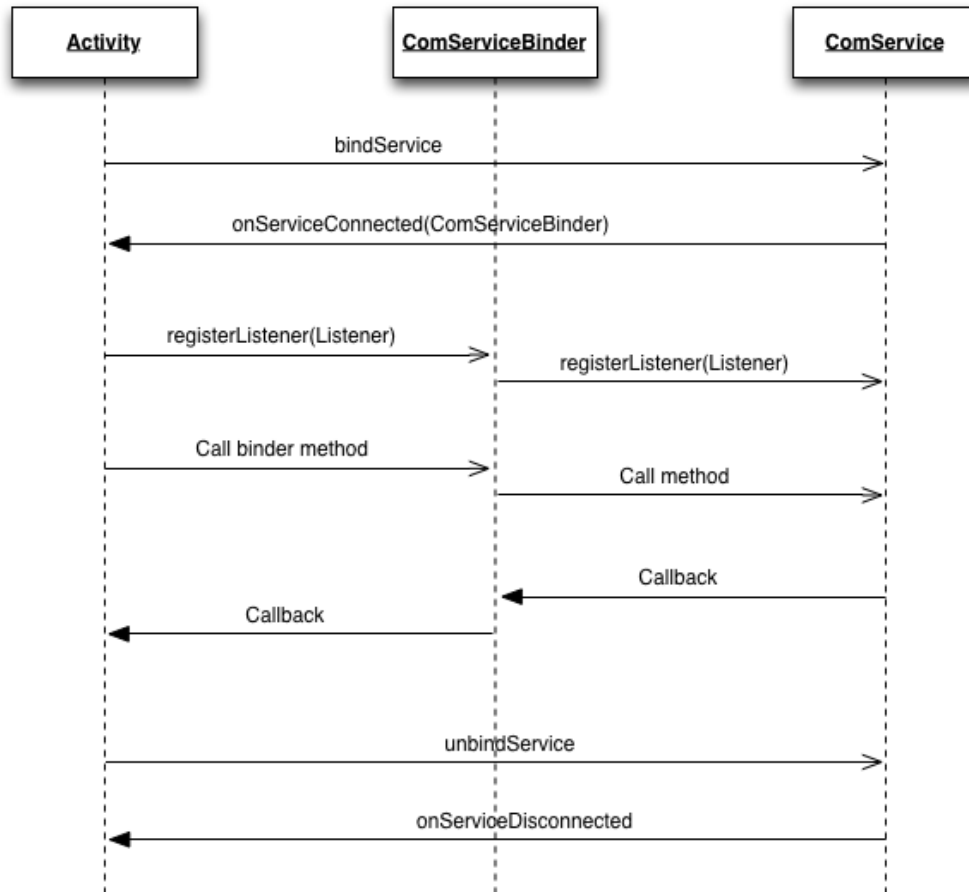


Figura 10 - Diagrama de sequência da comunicação com o ComService

Como é ilustrado, quando a ligação ao ComService é efectuada é passado um *Binder* [61] para a *activity*. Todos os pedidos ao ComService são feitos a partir dessa interface, inclusive o registo de *callbacks* para novos eventos de comunicação.

### 6.5.7 Funcionalidade de Desenho

Foi desenvolvida uma nova funcionalidade que permite ao utilizador desenhar e criar as suas próprias imagens. À exceção das ferramentas de desenho, toda a restante interface foi implementada pelo estagiário.

## 6.6 Preferences

Quanto às preferências do utilizador, os desenvolvimentos são referentes à implementação de um novo design.

### 6.6.1 Preferences Design

O ecrã de preferências sofreu uma remodelação total no seu aspecto.

A implementação deste ecrã baseava-se na implementação nativa de preferências. Apesar do comportamento nativo ser sempre o prioritário, este ecrã está integrado com a aplicação e não com as preferências do sistema. O seu aspecto gráfico não era coerente com o da aplicação.

A solução passou por implementar um *layout* de preferências customizado. Este passou a contemplar um cabeçalho e uma lista, sendo que esta última contempla todas as preferências já existentes.

## 6.7 Interface Geral da Aplicação

Toda a interface da aplicação sofreu uma grande alteração com a integração da ActionBar do Android, pelo que os desenvolvimentos são apresentados de seguida.

### 6.7.1 Integração da ActionBar

De modo a apresentar uma interface mais familiar para os utilizadores de Android, toda a interface da aplicação foi atualizada para suportar a ActionBar. Uma vez que este componente apenas está disponível para versões 4.0 ou superiores do Android, recorreu-se à única biblioteca que oferece suporte para versões anteriores, a ActionBarSherlock.

Uma vez que este componente oferece bastantes opções de customização, foi aplicado o design fornecido de modo a manter o *branding* da aplicação.

Um dos problemas encontrados foi a customização do menu de opções em dispositivos com o botão físico. Enquanto que em dispositivos sem esse botão, o menu de opções está acessível através diretamente na ActionBar (botão identificado por três pontos verticais) e o menu gerado é um componente da mesma, não surgiram problemas na customização. Porém, para dispositivos com o botão físico, o menu de opções aparece no fundo do ecrã e este é nativo, não permitindo qualquer tipo de customização. A solução encontrada passou por criar um novo componente que apresentasse o mesmo comportamento, mas seguindo o design definido.

## 6.8 Controlo de qualidade

Além das implementações referidas nos capítulos anteriores, foi desenvolvido trabalho a nível do controlo de qualidade da aplicação. Uma vez que os desenvolvimentos a este nível se tratam de correções do comportamentos anormais da aplicação e não de implementação de novas funcionalidades, a sua descrição não acrescenta valor ao relatório. No entanto, é importante referir que o controlo de qualidade é sempre uma constante no desenvolvimento da aplicação.

## 7. Gestão do Projeto

Neste capítulo é apresentado o processo de gestão do projeto. Uma vez que é utilizado o *Scrum* de uma forma rigorosa, obedecendo a todas as regras e práticas, começa-se por apresentar a equipa *Scrum* [62], passando-se para a descrição dos eventos que têm lugar durante o desenvolvimento e terminando com os artefactos e ferramentas de gestão do projeto.

### 7.1 A Equipa Scrum

Durante a sua permanência na empresa, o estagiário está integrado numa das equipas da divisão de *Telco*. Esta equipa não age de forma independente, uma vez que é parte integrante da equipa *Scrum*. Os elementos constituintes da equipa pertencem à empresa na qual decorre o estágio e ao cliente, como é ilustrado na figura 6.



Figura 11 - Estrutura da equipa *Scrum*

O estagiário integra a equipa de desenvolvimento, participando em todos eventos *Scrum* e tem todas as responsabilidades expectáveis desse cargo.

## 7.2 Os Eventos Scrum

Os eventos realizam-se regularmente e com uma duração fixa como é ilustrado na figura 7.

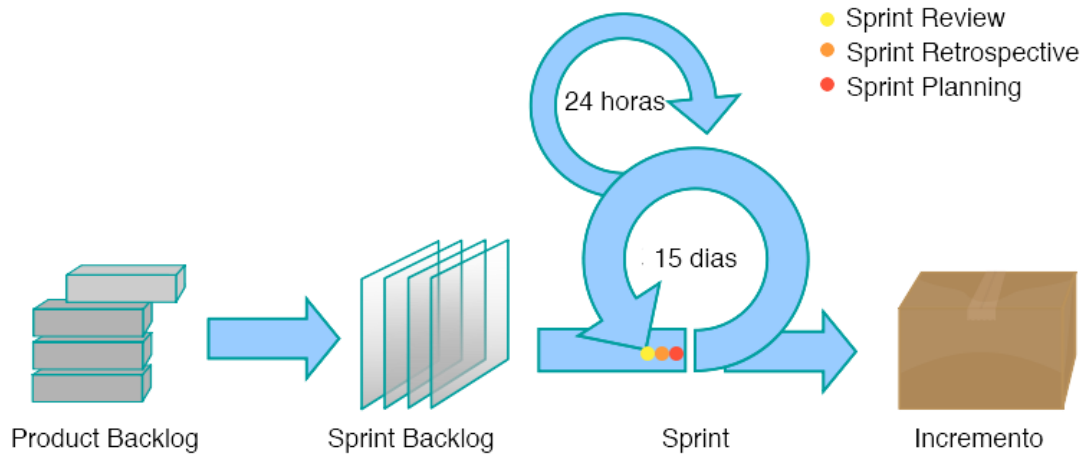


Figura 12. Diagrama do processo Scrum

O *sprint* começa sempre à Terça-feira e tem uma duração de duas semanas. Durante esse período, realiza-se a *Daily Scrum* todos os dias a uma hora fixa.

Os restantes eventos têm lugar à Terça-feira com toda a equipa *Scrum*. Esta sucessão de reuniões é iniciada pela *Sprint Review*, a qual tem uma duração de 2 horas. Sendo procedida da *Sprint Retrospective*, tendo esta uma duração mais curta, nunca excedendo 1 hora. Por fim realiza-se a *Sprint Planning Meeting*, sendo esta limitada a 2 horas para cada parte, totalizando 4 horas de duração.

Toda a comunicação é feita em inglês, de forma a reforçar a transparência de todo o processo.

## 7.3 Artefactos

O *Product Backlog* e o *Sprint Backlog* são ambos geridos com o auxílio do *Scrumwise*, e todos os itens são definidos como *User Stories* isto é, descrições de funções ou requisitos sob a forma de uma ou mais frases, as quais enunciam o ‘quem’, ‘o quê’ e ‘porquê’ de um modo simples e conciso.

Relativamente ao incremento resultante de cada *sprint*, este é distribuído internamente pelo cliente de modo a obter não só feedback relativo ao estado atual, mas também possíveis contribuições para novas funcionalidades.

## 7.4 Ferramentas de Gestão

Para a uma gestão do desenvolvimento são utilizadas duas ferramentas. Estas trazem grandes vantagens de um ponto de vista da organização do projeto e gestão e planeamento de sprints.

#### 7.4.1 Scrumwise [63]

O *Scrumwise* é uma ferramenta de *Scrum* que permite gerir um projeto de uma forma acessível a toda a equipa *Scrum*. As suas principais funcionalidades são:

- Gerir um *Product Backlog*, permitindo visualizar o que foi desenvolvido, quando foi desenvolvido e desenvolvimentos futuros.
- Planear um *Sprint*
- Gerir um *Sprint Backlog*, possibilitando visualizar o progresso dos desenvolvimentos correntes.

#### 7.4.1 Planning Poker [64]

O *Planning Poker* é uma ferramenta usada nas estimativas feitas pelos membros da equipa de desenvolvimento. Cada participante tem à sua disposição um baralho de cartas com vários valores (0, 1/2, 2, 3, 5, 8, 13, 20, 40, 100 e '?') e perante a *User Story* que esteja em discussão, o participante escolhe o valor que melhor reflete o esforço envolvido na sua realização. Esta escolha é secreta até que todos os participantes realizem a sua ou o moderador termine o período de votação.



## 8. Planeamento

Neste capítulo apresenta-se a evolução do planeamento ao longo do período de estágio. Na primeira secção é abordado o planeamento geral do projeto, sendo enumerados os pontos mais importantes do seu desenvolvimento. Na secção seguinte é apresentado o plano do primeiro semestre. O capítulo termina com uma secção relativa ao planeamento do segundo semestre, sendo comparado o plano definido inicialmente com os desenvolvimentos realizados ao longo desse período.

### 8.1 Plano do Projeto

No início do período de estágio, a aplicação encontrava-se em fase de acreditação para o RCS-e. Os desenvolvimentos seguintes focaram-se na implementação dos *joyn Hot Fixes*, os quais estavam planeados serem concluídos até Fevereiro de 2013.

Existe um lançamento da aplicação para o mercado, tendo como data o mês de Abril. No decorrer do período entre o início do ano de 2013 e esse lançamento, o foco dos desenvolvimentos vai ser a implementação de funcionalidades que enriqueçam a aplicação e alguns dos requisitos do *joyn Blackbird*.

Após o lançamento para o mercado e até ao final do período de estágio, o plano é continuar a implementar as funcionalidades do *joyn Blackbird*.

Durante toda a duração do estágio, é de salientar o investimento no controlo de qualidade da aplicação, com mais incidência nos períodos que antecedem a meta interna e o lançamento para o mercado.

### 8.2 Plano para o Primeiro Semestre

O trabalho desenvolvido no primeiro semestre é dividido em duas categorias, o desenvolvimento da aplicação e a elaboração do relatório de estágio. As tarefas desempenhadas durante este período são descritas na tabela 20.

Categoria	Descrição
Documentação	Estado da arte
Documentação	Análise de requisitos
Documentação	Elaboração do relatório intermédio de estágio
Hot Fixes	Gestão de contactos bloqueados
Hot Fixes	Identificação de chamadas em espera
Hot Fixes	Serviço de Group Chat
Hot Fixes	Preferências
Hot Fixes	Funcionalidade de File Share Auto-Accept
Blackbird	Orientação do Video Share
Controlo de Qualidade	Melhoramentos de performance e correção de erros

Tabela 20. Planeamento do primeiro semestre

### 8.3 Plano para o Segundo Semestre

O plano inicialmente definido para o segundo semestre é apresentado na tabela 21.

Categoria	Descrição
Documentação	Elaboração do relatório final de estágio
Hot Fixes	Ambientes Multi-device/Multi-client
Blackbird	Funcionalidade ‘Zoom-to-Fit’ para Video Share
Blackbird	Registo de utilizador via Wi-Fi
Blackbird	Serviço de Location Share
SMS Inbox	Integração dos serviços de chat com o serviço de SMS
Content Backup	Backup de conteúdos do utilizador
Message Queue	Armazenamento local de mensagens a enviar
Chat Orientation	Orientação dos ecrãs de Chat
Controlo de Qualidade	Melhoramentos de performance e correção de erros

Tabela 21. Planeamento do segundo semestre

Devido ao facto de o estagiário estar integrado numa equipa de desenvolvimento e a gestão do projeto ser feita com base no *Scrum*, o trabalho realizado acabou por divergir bastante do planeamento. Na tabela 22 é apresentado o trabalho desenvolvido ao longo do segundo semestre.

Categoria	Descrição
Documentação	Elaboração do relatório final de estágio
Blackbird	File Transfer thumbnails
Desenvolvimento	Integração da ActionBar
Desenvolvimento	Funcionalidade de Desenho
Controlo de Qualidade	Redesign do ComService
Controlo de Qualidade	Melhoramentos de performance e correção de erros

Tabela 22 - Trabalho desenvolvido no segundo semestre

## **9. Conclusões**

O resultado deste estágio é certamente vantajoso para o mercado das comunicações móveis. A contribuição no desenvolvimento da aplicação RCS e o seu lançamento para o mercado são provas disso.

Em termos pessoais, o facto de estar integrado numa equipa Scrum e lidar diariamente com o cliente foi sem dúvida um ponto bastante motivador em todo o processo de desenvolvimento.

Em termos de competências, o estágio contribuiu bastante para o enriquecimento das competências técnicas do estagiário, não só ao nível de desenvolvimento de software como também na gestão de um projeto.

## 10. Referências

1. “Ovum estimates that operators lost \$13.9bn in 2011 due to social messaging”, Ovum [Online], [http://ovum.com/press\\_releases/ovum-estimates-that-operators-lost-13-9bn-in-2011-due-to-social-messaging/](http://ovum.com/press_releases/ovum-estimates-that-operators-lost-13-9bn-in-2011-due-to-social-messaging/)
2. “Ovum figures indicate that operators will lose \$54bn by 2016 due to smartphone messaging apps”, Ovum [Online], [http://ovum.com/press\\_releases/ovum-figures-indicate-that-operators-will-lose-54bn-by-2016-due-to-smartphone-messaging-apps/](http://ovum.com/press_releases/ovum-figures-indicate-that-operators-will-lose-54bn-by-2016-due-to-smartphone-messaging-apps/)
3. “Rich Communications”, GSMA [Online], <http://www.gsma.com/rcs/>
4. “Spanish Mobile Operators Launch Nationwide Joyn Services”, GSMA [Online], <http://www.gsma.com/newsroom/spanish-mobile-operators-launch-nationwide-joyn-services>
5. “MetroPCS Takes Consumer Mobile Experience to New Levels as the First 4G LTE Mobile Service Provider in the World to Deploy Rich Communication Services”, Metro Investor Relations [Online], <http://investor.metropcs.com/>
6. “Korean operators launch joyn service”, Telecompaper [Online], <http://www.telecompaper.com/news/korean-operators-launch-joyn-service--915959>
7. Facebook [Online], <https://play.google.com/store/apps/details?id=com.facebook.orca>
8. Skype [Online], <https://play.google.com/store/apps/details?id=com.skype.raider>
9. WhatsApp [Online], <https://play.google.com/store/apps/details?id=com.whatsapp>
10. KakaoTalk [Online], <https://play.google.com/store/apps/details?id=com.kakao.talk>
11. LINE [Online], <https://play.google.com/store/apps/details?id=jp.naver.line.android>
12. Tango [Online], <https://play.google.com/store/apps/details?id=com.sgiggle.production>
13. Viber [Online], <https://play.google.com/store/apps/details?id=com.viber.voip>
14. ChatON [Online], <https://play.google.com/store/apps/details?id=com.sec.chaton>
15. Kik [Online], <https://play.google.com/store/apps/details?id=kik.android>
16. ooVoo [Online], <https://play.google.com/store/apps/details?id=com.oovoo>
17. WeChat [Online], <https://play.google.com/store/apps/details?id=com.tencent.mm>
18. Voxer [Online], <https://play.google.com/store/apps/details?id=com.rebelvox.voxer>

19. Fring [Online], <https://play.google.com/store/apps/details?id=com.fring>
20. textPlus [Online],  
<https://play.google.com/store/apps/details?id=com.gogii.textplus>
21. Cubie [Online], <https://play.google.com/store/apps/details?id=com.liquable.nemo>
22. FreePP [Online],  
<https://play.google.com/store/apps/details?id=com.browan.freeppmobile.android>
23. Hike [Online], <https://play.google.com/store/apps/details?id=com.bsb.hike>
24. Yuilop [Online], <https://play.google.com/store/apps/details?id=com.yuilop>
25. Text Me! [Online],  
<https://play.google.com/store/apps/details?id=com.textmeinc.textme>
26. Aire [Online], <https://play.google.com/store/apps/details?id=com.pingshow.amper>
27. Vippie [Online],  
<https://play.google.com/store/apps/details?id=com.voipswitch.vippie2>
28. “Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time” [Online],  
<http://www.gartner.com/newsroom/id/2573415>
29. Google Play [Online], <https://play.google.com/store>
30. RCS Release 1 [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/Release-1.zip>
31. RCS-e [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/V1.2.pdf>
32. RCS Release 2 [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/Release-2.zip>
33. Hot Fixes Implementation Guidelines 3.0 [Online], [http://www.gsma.com/rcs/wp-content/uploads/2012/11/RCS-e\\_Implementation\\_guidelines\\_v3\\_0\\_clean.pdf](http://www.gsma.com/rcs/wp-content/uploads/2012/11/RCS-e_Implementation_guidelines_v3_0_clean.pdf)
34. Hot Fixes UX Guidelines 2.2 [Online], <http://www.gsma.com/rcs/v1-2-joyn-hot-fixes-user-experience-guidelines>
35. RCS Release 3 [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/Release-3.zip>
36. RCS Release 4 [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/Release-4.zip>
37. RCS Release 5 [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/10/RCS-5.0-UNI-V1.0.zip>

38. android-rcs-ims-stack [Online], <http://code.google.com/p/libphonenumber/>
39. Android Application Fundamentals [Online],  
<http://developer.android.com/guide/components/fundamentals.html>
40. Android Activities [Online],  
<http://developer.android.com/guide/components/activities.html>
41. Android Services [Online],  
<http://developer.android.com/guide/components/services.html>
42. Android Content Providers [Online],  
<http://developer.android.com/guide/topics/providers/content-providers.html>
43. Android Broadcast Receivers [Online],  
<http://developer.android.com/reference/android/content/BroadcastReceiver.html>
44. Android Intents [Online],  
<http://developer.android.com/guide/components/intents-filters.html>
45. SIP (RFC 3261)[Online], <http://www.ietf.org/rfc/rfc3261.txt>
46. SDP (RFC 4566)[Online], <http://tools.ietf.org/html/rfc4566>
47. MSRP (RFC 4975)[Online], <http://tools.ietf.org/html/rfc4975>
48. RTP (RFC 3550)[Online], <http://www.ietf.org/rfc/rfc3550.txt>
49. Android Froyo [Online], <http://developer.android.com/about/versions/android-2.2-highlights.html>
50. Android SDK [Online], <http://developer.android.com/sdk/index.html>
51. android-rcs-ims-stack [Online], <http://code.google.com/p/libphonenumber/>
52. GSMA Accreditation [Online], <http://www.gsma.com/rcs/interoperability-testing/accreditation>
53. Android AIDL [Online],  
<http://developer.android.com/guide/components/aidl.html>
54. Lib Phone Number [Online], <http://code.google.com/p/libphonenumber/>
55. Action Bar Sherlock [Online], <http://actionbarsherlock.com/>
56. Android Action Bar [Online],  
<http://developer.android.com/guide/topics/ui/actionbar.html>
57. Android Ice Cream Sandwich [Online],  
<http://developer.android.com/about/versions/android-4.0-highlights.html>

58. Technical Test Case matrix [Online], [http://www.gsma.com/rcs/wp-content/uploads/2013/03/RCS\\_IOT\\_001\\_RCS\\_v1\\_2\\_Test\\_Cases\\_Matrix\\_v4.1\\_15\\_02\\_13\\_clean.xlsm](http://www.gsma.com/rcs/wp-content/uploads/2013/03/RCS_IOT_001_RCS_v1_2_Test_Cases_Matrix_v4.1_15_02_13_clean.xlsm)
59. UX Test Cases Matrix [Online], <http://www.gsma.com/rcs/wp-content/uploads/2013/02/RCS-1-2-UX-Test-Cases-Matrix-v-2.0.xlsx>
60. Quality check list [Online], <http://www.gsma.com/rcs/wp-content/uploads/2012/12/RCS-v1.2-Quality-check-list.xlsx>
61. Android Binder [Online], <http://developer.android.com/reference/android/os/Binder.html>
62. Scrum Alliance [Online], <http://scrumalliance.org/>
63. Scrumwise [Online], <http://www.scrumwise.com/>
64. Planning Poker [Online], <http://www.planningpoker.com/>