Masters in Informatics Engineering
Internship
Final Report

Audio/Video Conference Service Implementation Using the SIP Protocol

Filipe Estafero Henriques
estafero@student.dei.uc.pt

Supervisor DEI:
João Vilela
Date: 3rd July 2013

Supervisor WIT-Software:
Carlos Ferreira
Date: 3rd July 2013

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

Abstract

Nowadays, telecommunications are spread worldwide and play an important role on our life as well asthe Internet which is used with several objectives(work, play, chat and allows quick and easily get access to multimedia content.

To be able to compete with the Internet, the telecommunications industry had to come up with new ways of delivering services and multimedia content, using standards and providing compatibility across the operators. From this need, 3GPP created the IP Multimedia Subsystem (IMS) network architecture.

Another response emerged from the telecommunications world to reverse the revenues loss that has been happening on the last few years. The creation of the Rich Communications Suite (RCS) specification provides the user new ways of sharing content with others, in order to keep up with the type of services offered on applications categorized like Over The Top Content, that provide to the user features like calling or chatting without generating any revenues to the user's service provider.

On this project a new feature is added into RCS which allows a user to place a conference call amongst several other users with the click of a button, enabling also a set of value-added functionalities on these calls.

Keywords

"Audio Conference", "Conference", "Conference Call", "Group Call" "IMS", "IP Multimedia Subsystem", "RCS", "SIP","VoIP"

Index

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accouting
ACI	Activity Context Interface
ACID	Atomicity, Consistency, Isolation, Durability
ACK	Acknowledgement
API	Application Programming Interface
App	Application
AS	Application Server
ASR	Active Speech Recognition
CMP	Container Managed Persistent
CPU	Central Processing Unit
CSCF	Call Session Control Function
DTMF	Dual-Tone Multi-Frequency
GC	Group Call
GSM	Global System for Mobile Communications
GSMA	Global System for Mobile Communications Association
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I-CSCF	Interrogating-CSCF
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPsec	IP Security
IPDC	Inter Protocol Device Control
IPTV	Internet Protocol Television
ITU-T	International Telecommunication Union - Telecom
IVR	Interactive Voice Response
JAIN	Java APIs for Integrated Networks

JCP	Java Community Process
JEE	Java Platform, Enterprise Edition
JSON	JavaScript Object Notation
JSR	Java Specification Request
LTE	Long Term Evolution
MDCP	Media Device Control Protocol
Megaco	Media Gateway Control
MGCP	Media Gateway Control Protocol
MMS	Mobicents Media Server
MRF	Media Resource Function
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Processor
NAT	Network Address Translation
OAM&P	Operation, Administration, Maintenance & Provisioning
OMA	Open Mobile Alliance
OS	Operating System
OSA	Open Service Access
OTT	Over-The-Top Content
P-CSCF	Proxy-CSCF
PIN	Personal Identification Number
PRACK	Provisional Response Acknowledgement
QoS	Quality of Service
RA	Resource Adaptor
RADIUS	Remote Authentication for Dial in User Service
RCS	Rich Communications Suite
REST	Representational State Transfer
RFC	Request for Comments
RTP	Real-time Transport Protocol
S-CSCF	Serving-CSCF
SBB	Service Building Block
SCS	Service Capability Server

SDP	Session Description Protocol
SGCP	Simple Gateway Control Protocol
SIP	Session Initiation Protocol
SLEE	Service Logic Execution Environment
SV	Speaker Verification
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTS	Text-to-Speech
UA	User-Agent
UAC	User-Agent Client
UAS	User-Agent Server
UDP	User Datagram Protocol
UE	User Equipment
UI	User Interface
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
UX	User Experience
VoIP	Voice over IP
WCAS	WIT Communicator Application Server
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

Chapter 1
Introduction

This document describes the work done by the author during the academic year of 2012/2013. The work was developed within the scope of the Masters on Informatics Engineering on the Department of Informatics Engineering of the Faculty of Science and Technology of the University of Coimbra (DEI FCTUC), as part of the Thesis/Internship discipline.

The internship took place on a company named WIT-Software, S.A. and had as supervisors:

- Dr. João P. Vilela - DEI FCTUC
- Eng. Carlos Ferreira - WIT-Software, S.A.

1.1. WIT-Software, S.A.

The company was formed back in 2001, from a spinoff that started on the University of Coimbra (UC), with the support of the Pedro Nune's Institute (IPN).

It mainly operates on the telecommunications market developing services for the telecommunications operators/service providers. The company also has departments on the mobile applications area and IPTV.

1.2. Context

Telecomunications have been around for the last hundred years, evolving from fixed technologies to mobile ones. This change originated an incredible growth on the profits of the telecomunications industry over the years, as well as equipments sales, which represented enormous revenues on the last two decades. With the appearance of the Internet, people started changing the way they interact with each other and how multimedia content is consumed. Since the access, reliability and the bandwidth of the Internet has been improving over the years, providing new ways for people to comunicate, a shift was generated on the paradigm of the telecomunications industry, creating the need to develop new ways of delivering content and services.

A group named 3GPP was formed back in 1998, which is a result of an alliance between different telecommunication associations. Their main focus is to develop and standardize new, faster and reliable ways to deliver content and services over IP, maintaining compatibility across the different operators and previous technologies. It's from this need of convergence on the industry and improvement on the delivery of services and multimedia content that IMS was created. The objective is to allow to expand the type of contents offered, describing a network architecture that uses IETF standards widely adopted nowadays (SIP, RTP, etc.), allowing telecommunication operators to offer new services to their customers, which can be migrated between different IMS deployments with minor changes. This platform allows to develop new types of applications usuable across different types of devices (PC, mobile devices, television) and networks (wireless, cable, GSM, SS7), by creating an abstraction between the transport layer and the core of the network. The specification describes what kind of components exist in it, their functionality, the interfaces connecting each component and the flow of communication between each other to achieve certain operations.

Due to the fast growth of available applications for mobile devices on the last years, another threat emerged to the profits of the telecommunication operators. This threat is named OTT. These kind of services/applications, allow the user to perform operations like calling, texting or even sharing multimedia content with each other, without having to use the traditional ways (SMS, MMS, telephone call) available from the Services Providers. On the attempt of reversing the revenue losses caused by these type of applications, the GSMA group released the RCS specification. This group, formed by an association of mobile operators and other companies that operate on the same area, defined a new way for integrating natively into the mobile equipments or through an application, a set of features that provides functionalities like video sharing, file transfer, group chat and others.

The main aim of this project is to specify the procedures and describe the implementation of an audio and video conference service, which acts as an AS on the IMS network. It should allow any client on the network to enter on a previously scheduled conference by inputting an associated PIN with it. Besides that, it should also allow a RCS 5.1[RCS5_1] client to start a conference simply with a click of a button, from an ongoing group chat or a selected group of contacts. There is a set of procedures on the RCS specification which the service will have to be compliant with, yet the flow to initiate the conference from the previously mentioned cases will be specified from scratch, as currently the RCS 5.1 specification doesn't mention how to achieve this or anything related with this feature. The idea is to allow clients with the capability developed on this project, to be able to have a set of functionalities like presenting who's on the conference, mute someone by pressing a button, invite someone from the contact list to an ongoing conference and other functionlaties implemented natively on the equipment or through a distributed application and at the same time, take advantage of the features available on the RCS specification as for example the enhanced address book.

Because of limitations of the available free/open-source media servers/gateways (described ahead) the part of video conferencing was withrawn from the objectives of the project, since there was no available solution which could offer the necessary functionalities for a conference service and also provide video support.

Chapter 2
State of the Art

2.1. Background and Literature Review

In this part of the chapter, several different technologies and protocols are described, which are relevant for the development of this project, so that the reader is able to fully understand the concepts behind the objectives and the work developed on this internship. It also presents the available options for the development of the work proposed on this internship, such as different protocols for the same functionality.

Extra information can be found on Annex A which provides description on certain protocols, components and others that are not present on the main document like codecs, Diameter, etc.

2.1.1. SIP

SIP is a signaling protocol, which is named Session Initiation Protocol. It's used for initiating and manipulating VoIP calls as well as presence notifications. It can also be used to initiate an instant messaging session and a way to deliver messages for those sessions and many other functionalities. It is a text-based protocol, having been developed based on the HTTP protocol, functioning like a request-response protocol, indepent of the transport protocol. It was first defined on RFC 2543[RFC2543] by IETF [IETF] and having suffered several updates and extents on other RFC's like the adition of new methods. It's latest specification is described on RFC 3261[RFC3261]. There are several concepts like transactions and dialogs on the protocol, which are initiated with certain type of requests and terminated with other specific requests/responses, like the setup of a VoIP call until it's termination (dialog).

The messages are composed by an initial line, indicating the type and operation of the message and can have one or more headers, as well as an optional body containing any type of content (XML, SDP, etc.). There are 2 types of SIP messages, requests and responses. It supports an extent list of request methods, each with it's own functionality. Some of them are explained on Table 1.

Table 1 - RFC3261 defined SIP Request Methods

Method	**Description**
ACK	Informs the other peer that the last response was acknowledged
BYE	Terminates a session on progress between two peers
CANCEL	Allows to cancel a previously sent INVITE, that was not yet accepted by the other endpoint
INVITE	This method allows a user to invite another user to a session, which may envolve one or more streams, being the description of these, embeded on the message using SDP.
OPTIONS	Informs the other peer of the supported capabilities and/or queries it

	about the ones that it supports
REGISTER	Registers a user/UAC on a registrar server

There are also several different response codes, grouped into different categories, which are shown on Table 2, providing a wide range of description to the requester about the result of the processing on the receiver of the previously sent message.

Table 2 - SIP Response Codes

Response Code	**Description**
1xx - Provisional	Informs that the request previously sent was received and also that further action is on hold
2xx - Success	Previously received request was successfuly received and processed
3xx - Redirection	Further action needs to be taken by the peer that sent the request
4xx - Client Error	The request is not valid (client might not be allowed or it might contain bad syntax)
5xx - Server Error	A failure ocurred on one of the servers where the request has to go through, yet it seemed a valid one
6xx - Global Failure	Failure on the servers, stoping the request from being processed

Next, on Figure 1, is presented a simple scenario, with the exchange of messages between 2 users in order to start a voice call and terminate it. The INVITE starts a new dialog (if it's not already running), which will be terminated with a BYE request. This dialog has an identifier, and the properties of the dialog can be modified with other INVITE request, from any of the peers, without terminating it or starting a new one (mid-dialog).

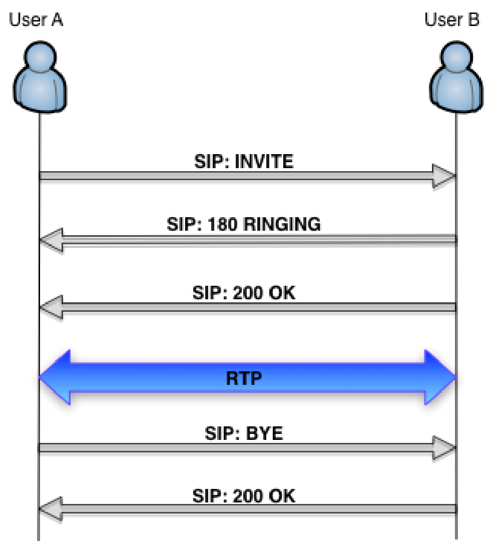
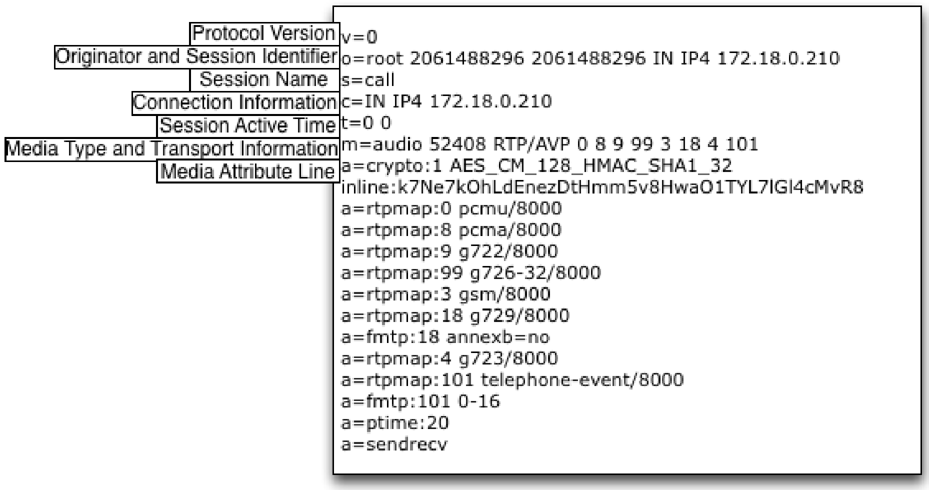


Figure 1 - SIP Signaling

While this scenario is presented, it doesn't mean that a setup of a session needs to follow exactly this flow of messages exchanging. There can be other messages envolved, like for example other provisional responses and the acknowledgement of these by the other user (ex: 100 Trying and the correspondant PRACK). On both the INVITE and 200 OK, each peer send SDP content on the messages, which is used to negotiate the codec used on the RTP stream for example for voice communication between each other. Both SDP and RTP protocols are explained next. A few more details on this protocol and some concepts within it, are given on Annex A, section A.1.

2.1.2. SDP

This protocol can be used to describe one or more media/data streams. These can be audio, video, messaging. It can be used to initiate the streams, modify the parameters of them (port, IP, codecs, etc.) or stopping one of them, being a protocol used to negotiate the mentioned properties between two endpoints. It can be embeded on SIP messages and it is a text-based protocol as well.



```
v=0
o=root 2061488296 2061488296 IN IP4 172.18.0.210
s=call
c=IN IP4 172.18.0.210
t=0 0
m=audio 52408 RTP/AVP 0 8 9 99 3 18 4 101
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:k7Ne7kOhLdEnezDtHmm5v8HwaO1TYL7lGl4cMvR8
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:9 g722/8000
a=rtpmap:99 g726-32/8000
a=rtpmap:3 gsm/8000
a=rtpmap:18 g729/8000
a=fmtp:18 annexb=no
a=rtpmap:4 g723/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
```

Figure 2 - SDP Message Example

As shown on Figure 2, on the content of an SDP offer is described for each media stream, the IP and port of the peer on which it is expecting to receive the data for that stream, the supported codecs, cryptographic parameters for the packets encription. There is also information for identifying the negotiated session, as well as the identification of each type of the supported media (audio, video, etc.).

2.1.3. RTP

The RTP protocol specifies a standardized format for transmiting media packets (audio and video). It was developed by IETF and it's current specification is described on RFC3550 [RFC3550]. It has many different applications, like audio/video calls, media streaming, IPTV, among others. On an RTP packet the payload representing the data itself is included, as well as other set of headers, which are usually useful for organizing the order of the content received as well to detect delays on the data transmission.

2.1.4. IMS

IMS stands for IP Multimedia Subsystem, which specifies several components on it's architecture as well as the interfaces used to communicate between each other and what protocols it should use. It also defines the flow of communications to achieve certain operations. It is specified by 3GPP, a group that is collaboration between different telecommunications associations. It is currently on the release 12[IMS12], having suffered several aditions like support for different network technologies (LTE, UMTS) and components changes across the different releases. The network, as specified of now, can be divided in 3 layers as seen on Figure 3.

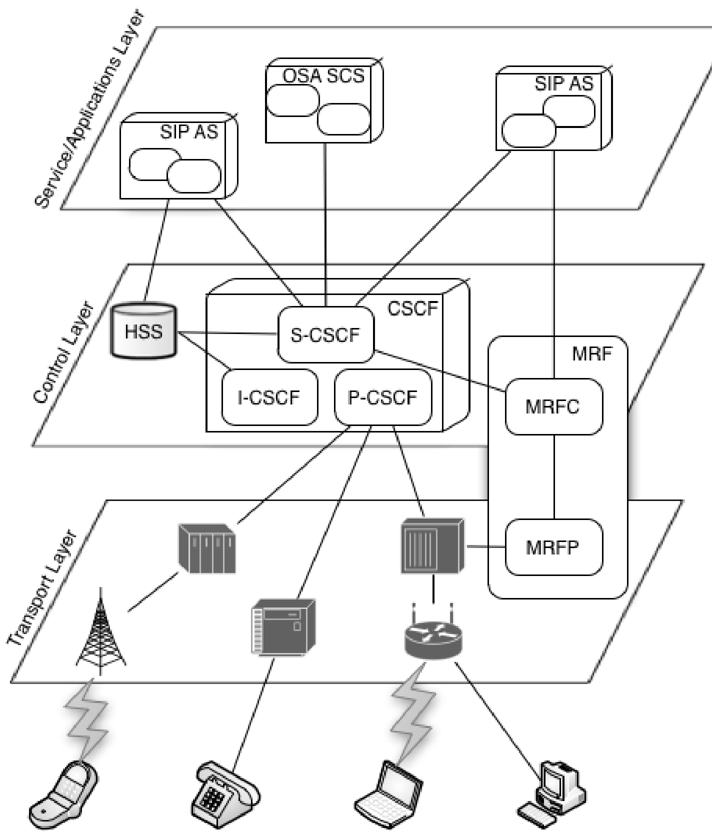


Figure 3 - IMS Layers

Each layer has it's own responsability as described next:

- **Service/Application**

This is where the operators blend/deploy their applications, although this is not totally true, as it can ocurr on other layers. Depending on the triggers defined within the Control Layer, the user is redirected by the S-CSCF to an application, which can perform functionalities like presence notification, conferencing, IPTV and many others.

- **Control**

This layer is responsible for routing the received messages to it's destination, or simply perform service control (register user, updating the registration timeout). It performs the connecting point between devices and services, also providing charging and provisioning facilities to the services/applications on the network.

- **Transport**

It's main responsibility, is to converge all the networks/protocols accessing to the IMS network, into standards used in it. This allows the network to be independent of what protocols/signals and/or networks are used to access to it, as there can be used converters, adaptors, interperters on this layer.

Since the main idea of the network is to allow convergence, it allows to deliver a wide variety of services like IPTV, Instant Messaging, Voice or Video exchange between a wide range of different type of devices and many other possible applications, since it allows to abstract the network type being used to acces the IMS network itself.

On section A.3 of the Annex A, more details can be found about the interfaces existant for communication between each one of the components that form the network architecture as well as some detail on the network's core components.

2.1.5. Media Gateway/Media Server

Media Servers or Media Gateways, are components, which can be distributed as software and hardware (box) or as a simple software. It's main responsibility is to handle multimedia streams (audio and/or video), serving as endpoints for these. It has functionalities like playing annoucements, streaming video, detect DTMF codes, voice recognition, text-to-speech, etc. It also can provide centralized endpoints where the UEs can connect for audio and/or video mixing. On the context of the IMS network architecture, it can be seen as the MRFP (more details provided on Annex A, section A.3)

2.1.6. Media Gateway Control Protocols

Several protocols have been proposed over the years as a standard to control/communicate with Media Gateways. There is currently a specification[RFC2805] which defines what are the requirements in terms of performance, connection, security and other aspects for these kind of protocols as well as the required operations to support and architecture. On this section of the document 2 of these type of protocols are presentend, which are relevant for this internship.

MGCP

The MGCP protocol was specified based on the merge of the SGCP[SGCP] and IPDC[IPDC] protocols. This originated the definition[RFC2705] of the protocol, which allows a call agent to manipulate a media gateway, like manage endpoints, create, delete or modify connections to the endpoints, detect events on the and request notifications on the completion of certain events (annoucements, etc.). It uses SDP to negotiate and modify media streams with the gateway.

There are different types of endpoints (Announcement, Conference Bridge, IVR, Packet Relay, etc.) as well as different available packages for manipulating each one of the endpoints types, i.e. each endpoint type may or may not support a package (DTMF, Trunk, Announcement, Audio, etc.). When any request completes or fails, the Call Agent receives a response from the Gateway with an event code (Operation Completed or Operation Failed) and a response code, ex. the operation failed due to the reason explained on the response code.

2.1.7. Conference Standards using SIP

2.1.7.1. Conferencing Framework using SIP

Currently there is a standard [RFC4353], which specifies what kind of components should exist and it's responsibilities on a conference service, as presented on Figure 4.

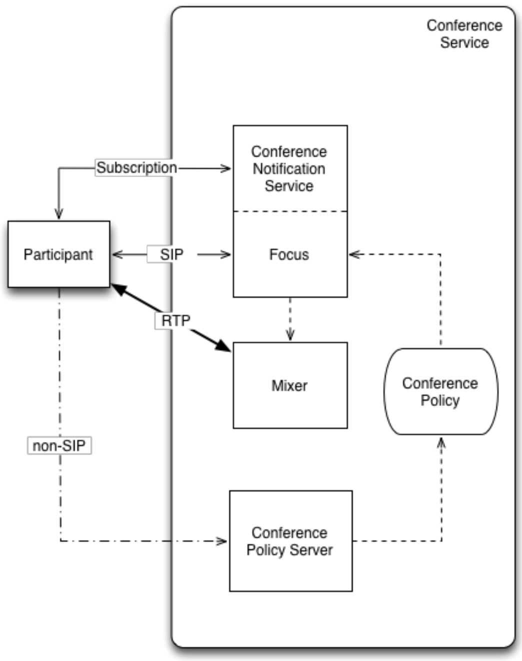


Figure 4 - Conferencing Framework with SIP

- **Focus**

  It is where all the conference's logic is present. The Focus is responsible for maintaining the SIP dialog with the participants, acting as the conference center, managing the addition or removal of participants to/from the conference, acording to the policies present on the Conference Policy Server.

- **Conference Policy Server**

  On this component, the service has the information on what kind of policies exists for the conferences, like the users that are authorized to join (list of users, joining time, approval by a moderator, etc.).

- **Mixer**

  It's main responsability is to join the different streams from all the users into a single endpoint or endpoints (depending on the number of different types of available media). It must take care of all the streams and mix the content received, outputing the result of the mixture to the users part of an endpoint.

- **Conference Notification Service**

  For keeping the conference participants informed about the presence of each other on the conference, this component is made available for notifying them of the departure or arrival of new users on the conference. It is considered as sub-component of the Focus.

The described components don't need to be phisically separated, as they can even be part of each other. It's description is intended to present the requirements for implementing a conference service, and what kind of operations should be made available, as well as the necessary procedures to achieve them.

There is also the notion of conference-aware participants and conference-unaware participants. On the first, the UA is aware that the conference's UA provides facilities for presence notifications, on which the user is able to receive information on the conference participants. On the other hand, a conference-unaware participant interprets the conference as a usual call to another peer.

2.1.7.2. Call Control - Conferencing for User Agents

It is described and considered as best practice[RFC4579] for managing conferences, a set of flows and parameters that shall be used in order to provide some conference control features to it's participants (in case these are conference-aware participants). The Focus shall act as a UA, performing the signaling part of the service and maintaining a relationship between the participants and the correspondent conference, i.e. each conference as a Focus.

- **Focus UA**

  As described above, it maintains the relationship between each conference and it's participants. It is unique per conference, since each conference will have it's own URI generated at the creation of the conference.

- **Conference Factory URI**

  This URI corresponds to the entry point of the service, from where a new URI will be retrieved to the participant, once he his authorized to enter or create a conference.

- **'isFocus' Parameter**

  This parameter allows participants to be aware that the UA is a conference, providing call control features for them. It is provided as a capability on the dialog between the participant and the conference as described on RFC3840[RFC3840].

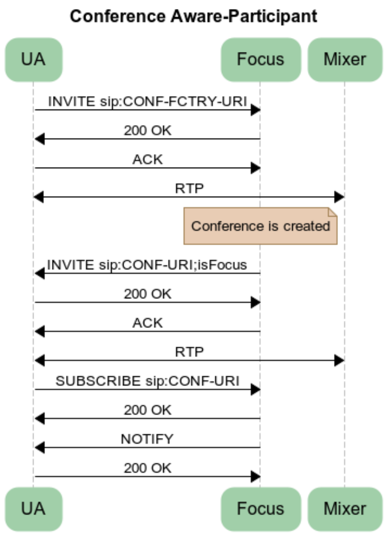


Figure 5 - Call Control and Conference-Aware Participant Flow Example

It also shows examples on the procedures necessary for a conference-aware participant to invite another user into a conference, by using the REFER method of the SIP protocol and some other other procedures that involve the use of headers present on the SIP protocol.

2.1.7.3. Presence Server

This service/server allows users to subscribe for notifications on a user's presence or to group's information (who's on the group). For performing this operations it uses SIP messages described below on Table 3, which can also carry embeded information on it's body usually using the XML protocol to describe it. It also allows the users to publish it's current mood (happy, dinning, busy, etc.).

In the IMS context, there is an organization named OMA that specifies different types of Enablers for the IMS network. These are services deployed on an IMS network usually available for other services/applications to use, providing well-defined interfaces that allow to create richer end-user services.

Table 3 - Presence Server SIP Messages

SIP Request	Objective
SUBSCRIBE	Requests the server to send notifications to the user that sent the request, when a change ocurrs on the subscribed package/user
NOTIFY	Notifies changes on the presence/status to the users that subscribed to a package/user
PUBLISH	Informs the server that the user that sent this request has changed it status, which is described on the body of the request

On the table above is presented the SIP requests used on this type of service/server and what are the functionality of each one of them.

SIP Event Package for Conference State

There is currently a well-established standard [RFC4575] that describes an event's package, which allows users to subscribe and receive notifications about users who join or leave the conference, as well as other types of information, as the media status (streams information) of each one as well as for the conference endpoint itself. The standard describes a XML schema with the previously mentioned information and more if needed (certain fields are merely optional), and the set of procedures needed to couple all the conference's parties into a single centralized notification endpoint.

2.1.8. Application Server Development Technologies

2.1.8.1. JAIN SLEE

JAIN SLEE is a Java programming language API specially designed for the telecommunications context, which allows the development of services/applications with high throughput, low latency, scalability and availability. These were the main goals when it was first specified by a collaboration of different telecommunication operators and software vendors as an event driven application server. Currently it is on version 1.1[JSR240]. It automaticly manages transactions and performs node replication (using configurations made by the developer), integrating the ACID properties. The main focus of the developer should be to keep the handling of the events, short and lightweight.

2.1.8.2. SIP Servlets

This technology is based on the HTTP Servlets. It is used to develop applications/services that are deployable across different containers (that have SIP Servlets support) with minor changes and allows integrating with other JEE components. The specification is currently on version 2.0[JSR359], having been released 2 other versions prior to the current one.

Since it has integration with JEE, the technology allows to develop converged applications, allowing to add SIP functionality to previously developed JEE applications.

2.1.9. RCS

With the growth on the use of OTT applications (skype, whatsApp, iMessage) by the clients, the operators started to notice some loss on their revenues. The OTT applications allow the users to perform the traditional type of features provided by the operators (voice calls, text messages, multimedia messages), except that using these type of applications frequently have no cost to their users, since the majority are usually free, preventing the operator from earning money with operations like texting, voice calls, video sharing and others (except the costs of the internet access from the mobile device).

RCS, or Rich Communications Suite, is the answer from the telecommunication's operators to this threat on their revenues. It describes a set of features and the needed procedures to perform certain operations, which work across different operators and vendor platforms, facilitating the adoption of services developed on IMS. The compliant clients, are branded/marketed as joyn, being this attributed by GSMA [GSMA] (who also releases the RCS specifications) after annalysing a required set of tests sent by the companies, which provides assurances like QoS, interoperability and functionality of the supported features.

The current version of the specification is RCS 5.1. It should not be interpreted as a defnition of standards to achieve certain types of features, or replacement for the OTT applications, but more like a choice made available to the end-user, which can come natively included with it's equipment (near future) or downloaded from the phone's respective mobile store/market.

Currently the specification describes a different set of features, some examples like presence sharing (mood, online/offline), location sharing, file transfer, video sharing, group chat (texting), VoIP, etc.

2.1.9.1 joyn by WIT Software

WIT Software became the official joyn distributor, selected by GSMA, after a long process which involved several ohter companies competing against each other in order to achieve it. Still, this doesn't mean that other companies can distribute their RCS client branded as joyn as explained previously.

2.1.10. Android

Android is an Operating System that targets mobile equipments (mobile/smart phones, tablets and similar), which was first developed by Android Inc. and then acquired by Google, making the source code available for everyone (open-source). The OS is based on Linux, and is widely distributed nowadays on a wide range of equipments.

A customized version of Java is used as the programming language in order to develop applications for this OS, while the layouts can be defined on a XML file or programaticly.

2.1.10.1. Service

In Android, a service is usually a part of an application which is continuosly running on the background of the same, even if the application is not currently on the foreground of the equipment.

2.1.10.2. Activity

An activity can be interpreted as a type of class on Android, that is responsible for creating and managing a window lifecycle. It is responsible to populate the view on this window, and also manage the interactions among the user and the items on the window.

It can also contain several Fragments (explained next), and manage the lifecycle of these Fragments.

2.1.10.3. Fragment

A Fragment is usually part of an Activity, whcih allows managing a specific part of the window within a Fragment and easily modify the behaviour and UI of that part of the window, by modifying the Fragment used and/or displayed. It's lifecycle is dependent on the lifecycle of the Activity that manages the Fragment, still the Activity can create or destroy Fragments as convenient.

Fragments are also very useful to display the same interface in different ways depending on the type of equipment (tablet or phone) and the size of the screen.

2.2. Available Solutions and Market Analysis

On this analysis, in terms of components that are required for the architecture of the solution, only free and/or open-source solutions were evaluated, since one of the main objectives of the works is to use these type of solutions.

2.2.1. Collaboration Services and Applications

There are currently several different services and applications that provide conferencing support. These can either be from a simple audio conference call to an extent of different functionalities like whiteboard, document presentation, support for video and others. These type of services/applications were studied in order to understand what are the most common ways to perform a conference call and gather information on which are the most common functionalities present on both a collaboration/conference service. Although the collaboration services are studied, these type of service is already being developed on another project, which will take advantage of the service developed in here in order to have conference call support.

Table 4 - Conference Services and Applications Comparison

	Adobe Connect	Arkadin	Banckle Meeting	Google Hangout	GoTo Meeting	Live On	Saba Meeting	Skype	Team Viewer	Über Conference	Web Conference	Web Ex
Screen Sharing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Application Sharing	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✓
Whiteboard	✓	✓	✓	✗	✗	✓	✓	✗	✓	✗	✓	✓
Video	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Documents Presentation	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓
Highlight Content	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗	✓	✓
File Sharing	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓
Poll	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗
Reactions	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
VoIP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Telephony	Dependant on 3rd parties	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓
Group Chat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Private Chat	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓
Q&A	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Web Browsing	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
Notes	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓
Cloud Files	✓	✗	✓	✓	✗	✗	✓	✗	✓	✗	✗	✓
Record	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓
Restrictions	✓	✗	✓	✗		✗	✓	✗	✓	✗	✓	✓
Portrait	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✓
Speaker Focus	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Remote Control	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓
Animation	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Block	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Mute	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓
Ad-hoc	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
E-mails	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗
Free	✗	✗	✓	✓	✗	✗	Limited	✓	Limited	✓	✗	Limited

2.2.2. Mobile Conference Applications

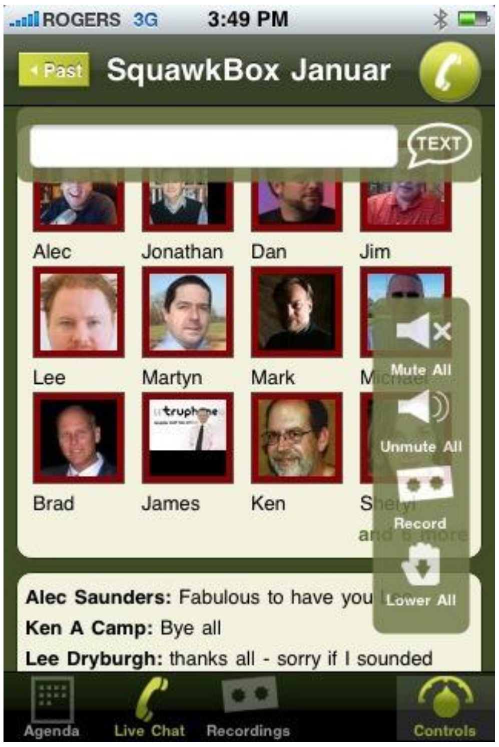
In order to evaluate which are the most common functionalities provided on a mobile conference application, a search was made in order to find which were the available ones on the market. As these type of applications are not very common as native mobile applications, very few were found.

Next are described each one of the applications studied.

2.2.2.1. Calliflower Mobile

http://www.calliflower.com/

Calliflower Mobile is an application for Apple's mobile devices (iPhone, iPad and iPod), which allows users to perform a serie of related features to conference call. For the free version, the users are limited to 70 per conference. In the paid version, a trial of 14 days is offered at the begining, and after that time the user is required to pay 50$/month, having access to all the features included on the app and the availability of 180 dial-in numbers over 34 countries, which allows participants who are calling to the conference not having to place a call for an international number which requires extra fees to be payed.



Alec Saunders: Fabulous to have you
Ken A Camp: Bye all
Lee Dryburgh: thanks all - sorry if I sounded

Figure 6 - Calliflower Mobile App [CLIFL1]

It also offers some interesting functionalities like the hand raising and calendar integration. Next are presented the main pros and cons of this mobile application.

Pros

Schedule Conferences	★
Conference Management	★
Group Chat	★

Recording	★

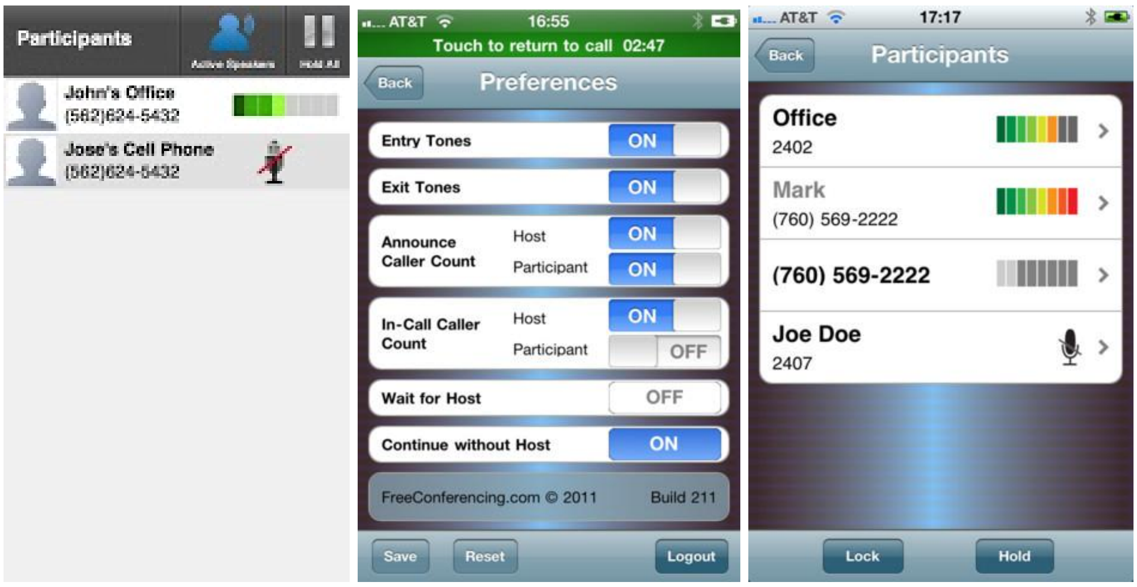
Cons

User Interface	✗
Problems to setup account	✗
Functionality	✗

2.2.2.2. Free Conferencing

http://www.freeconferencecall.com/

This application for Android and iOS is fragmented along several other applications (an HD one, international version, etc.), which makes it very confusing for the user to decide which one to download. Still after testing the regular version and comparing the functionalities of the other versions, it was soon perceuved that the only changes were the improvement of the UI aspects and the dial in numbers available for the conferences.



John's Office (562)624-5432
Jose's Cell Phone (562)624-5432

Entry Tones		ON
Exit Tones		ON
Announce Caller Count	Host	ON
	Participant	ON
In-Call Caller Count	Host	ON
	Participant	OFF
Wait for Host		OFF
Continue without Host		ON

FreeConferencing.com © 2011 Build 211

Office 2402
Mark (760) 569-2222
(760) 569-2222
Joe Doe 2407

Figure 7 - FreeConference Call Mobile Application [FCFC2] [FCFC1]

It is a pretty basic application, that is available for free and that provides very basic conferencing functionalities like muting, setting up a conference, etc. Other than that, it is possible to use a Q&A (Question and Answers) feature and also manage which type of announcements should be played.

Pros

Show speakers	★
Schedule Conferences	★
Manage conference properties	★
Recording	★

Cons

Very limited functionalities	×
Visual information is very poor	×
Wide variety of applications for the same purpose	×

2.2.2.3. FreeConference Mobile

http://www.freeconference.com/

This is an application made available at no cost for three distinct platforms, Android, iOS and Windows Phone 7. It has integration with several other services, like Facebook, Twiter, Outlook, etc. for sending conference information. Besides the usual conferencing functionalities, also allows to setup a personalized greeting sound.

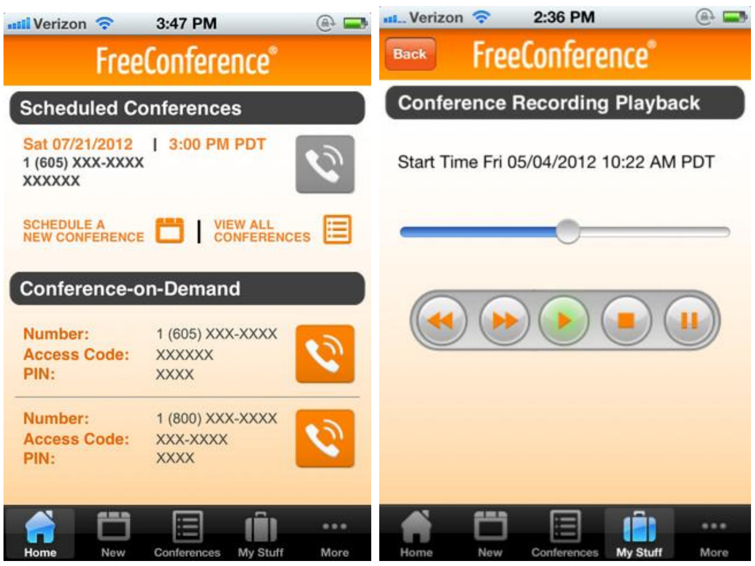


Figure 8 - FreeConference Mobile [FCFM1]

It is a quite simple application, i.e. doesn't really bring anything new compared to the other applications in terms of useful functionalities for conferencing.

Pros

Schedule Conferences	★
Integration with other services	★
Recording	★

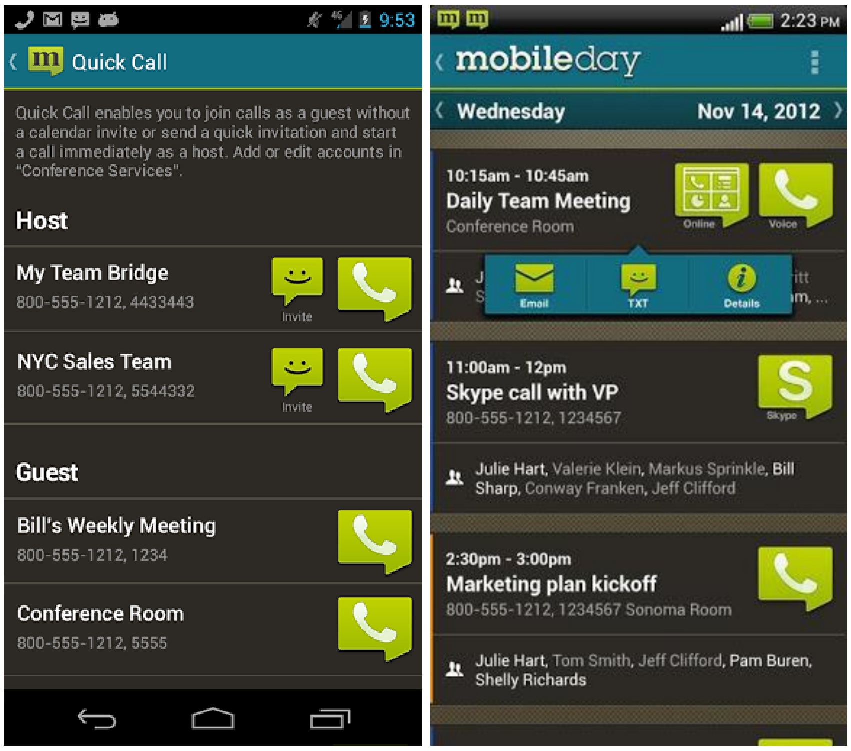
Cons

Very limited functionalities	×
UI a bit overloaded	×

2.2.2.4. MobileDay Conference Call

http://mobileday.com/

MobileDay provides an Android and iOS application, which allows users to save the conference data into the app from almost any conference provider and dial in when it's time, with just one click. It is a pretty basic application, still if the phone's signal drops and the user exits the conference call, the application allows the user to rejoin the conference with just one click.


Quick Call

Quick Call enables you to join calls as a guest without a calendar invite or send a quick invitation and start a call immediately as a host. Add or edit accounts in "Conference Services".

Host

My Team Bridge
800-555-1212, 4433443

NYC Sales Team
800-555-1212, 5544332

Guest

Bill's Weekly Meeting
800-555-1212, 1234

Conference Room
800-555-1212, 5555

mobileday

Wednesday Nov 14, 2012

10:15am - 10:45am
Daily Team Meeting
Conference Room

11:00am - 12pm
Skype call with VP
800-555-1212, 1234567

Julie Hart, Valerie Klein, Markus Sprinkle, Bill Sharp, Conway Franken, Jeff Clifford

2:30pm - 3:00pm
Marketing plan kickoff
800-555-1212, 1234567 Sonoma Room

Julie Hart, Tom Smith, Jeff Clifford, Pam Buren, Shelly Richards

Figure 9 - MobileDay Conference Call [MDCFC1]

The application is free of charge, and it also allows the users to text and email the conference details to it's contacts.

Pros

Save Scheduled Conferences	★
Integration with other services for logging in	★
User Interface	★

Cons

Very limited functionalities	✗
UI a bit overloaded with information	✗

2.2.2.5. UberConference

http://www.uberconference.com/

The UberConference mobile application was launched on December 2012 for both iOS and Android platforms. It presents a very intuitive UI, with a few conference management functionalities, as also statistics at the end of each conference call with details like for example who talked the most/least and the people that joined the conference.

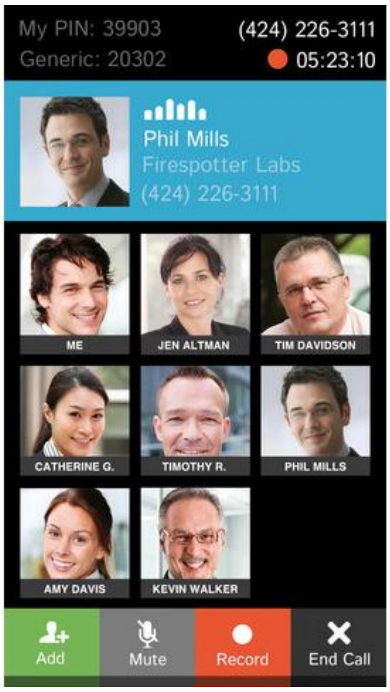


Figure 10 - UberConference Mobile App [UBR1]

Right now, it can be considered the app to battle against, since it offers almost all the main features needed on a conference application, with a very clean and intuitive UI, integrated with an excellent User Experience (UX).

Pros

Focus on talker	★
Conference management	★
User Interface	★
Recording	★

Cons

Group Chat	✗

2.2.3. IMS Components

The available open-source and/or free solutions for the components of a function IMS network architecture are very few.

2.2.3.1. Imszone

http://www.imszone.org/

It provides basicly all the components that form the core of an IMS network and has a very complete solution with several extra components already prepared to be added to the solution. Still the development and stopped a few years ago.

Core components	✓
Common interfaces	✓
Development	✗
Support	✗
Extra components (MRF, Presence, Charging System)	✓

2.2.3.2. Little IMS

http://confluence.cipango.org/display/LITTLEIMS/Home

It provides a little setup with a few missing interfaces and functionalities of the IMS network components.

The development of this IMS is made in SIP Servlets, providing most of the necessary features for the deployment of a simple IMS network architecture.

Core components	✓
Common interfaces	✓
Development	✓
Support	✓

2.2.3.3. OpenIMS

http://www.openimscore.org/

Provides the necessary components to setup a simple IMS network, and has an active community.

The updates are not as frequent as they used to be, still there are some updates from time to time, in order to keep the solution up to date with protocol changes and IMS specification changes.

Core components	✓
Common interfaces	✓
Development	✓
Support	✓
Extra components (Presence, Charging System, OAM&P)	✓

2.2.4. Media Gateway/Media Server

There were several solutions found for this component, but none provided all the necessary features needed for the work. The comparison of all the media servers/media gateways found is presented on Table 5.

Table 5 - Media Servers / Media Gateways Comparison

		ImsZone	Kurento	mediactrl	Mobicents Media Server	SIP Express Media Server	MCU Medooze Media Server
Audio Codecs	RAW	✗	✗	✓	✗	✓	✗
	G.711	✓	✗	✗	✓	✗	✓
	PCMU(u-law)	✗	✗	✓	✓	✗	✗
	PCMA(A-law)	✗	✓	✓	✓	✓	✗
	GSM	✓	✓	✓	✓	✗	✓
	speex	✗	✓	✗	✓	✓	✓
	G.729	✓	✗	✗	✓	✓	✗
	DTMF	✗	✗	✗	✓	✓	✗
	AMR	✗	✓	✗	✗	✓	✗
	MPA	✗	✓	✗	✗	✗	✗
	L16	✗	✗	✗	✗	✓	✗
	G.722	✗	✗	✗	✗	✓	✗
	SILK	✗	✗	✗	✗	✓	✗
	iSAC	✗	✗	✗	✗	✓	✗
	G.726	✗	✗	✗	✗	✓	✗
	FLV1	✗	✗	✗	✗	✗	✓
	nelly	✗	✗	✗	✗	✗	✓
Video Codecs	H.261	✗	✗	✓	✓	✗	✗
	H.263	✗	✓	✓	✗	✗	✓
	H.263+	✗	✗	✓	✗	✗	✓
	H.264	✗	✓	✓	✗	✗	✓
Endpoints and Functionalities	Streaming	✓	✗	✓	✓	✗	✓
	Conferencing	✓	✗	✓	✓	✓	✓
	Recording	✓	✗	✗	✓	✓	✓
	Play back	✗	✗	✗	✓	✓	✗
	IVR	✗	✗	✓	✓	✓	✗
	TTS	✗	✗	✗	✓	✗	✗
	Transcoding	✗	✓	✓	✓	✓	✓
	ASR	✗	✗	✗	✗	✓	✗

	SV	✗	✗	✗	✗	✗	✗
Control Protocol	MGCP	✗	✓	✗	✓	✗	✗
	Megaco/H.248	✗	✗	✗	✗	✗	✗
	MSCML over SIP	✓	✗	✗	✗	✗	✗
	VXML over SIP	✗	✗	✓	✗	✗	✗
	CCXML over SIP	✗	✗	✗	✗	✗	✗
	MSML over SIP	✗	✗	✗	✗	✗	✗
	JSR-309	✗	✗	✗	✓	✗	✗
	XMLRPC	✗	✗	✗	✗	✓	✓
Others	Last Update	2009	February 2012	2009	March 2013	July 2012	January 2013
	Last release	1.0	0.9.0	0.4	3.0.0Final	1.5.0 Release	revision705
	Language	C/C++	Java	C/C++	Java	C++/Python	C/C++

2.2.5. Containers

On Annex A, section A.6 can also be found the available SIP Servlets containers and a brief description of each one of them.

2.2.5.1. JAIN SLEE

mobicents http://www.mobicents.org/slee/intro.html

The only free or open-source available container / application server for the JAIN SLEE technology was Mobicents JAIN SLEE which supports the version 1.1 of the specification [JSR240]. It also provides several different resource adaptors (better explained on the next chapter).

Chapter 3
Objectives and Requirements

3.1. General Objectives

The aim of this project as previously explained is the implementantion of an audio and video conferencing service. It should allow a user to place a call to the service and get access to a previously scheduled conference call by inputing a provided PIN. This might be a moderator or a participant PIN, which are associated with a unique conference.
The service must also provide the ability for a RCS 5.1 compliant client to start a conference call from an ongoing group chat or a selected list of contacts with a simple click of a button. It should also have the ability to notify the user about the presence participants on the conference. This procedure is currently not part of the RCS specification, so it provides the oportunity to create a new functionality which is not part of any of the RCS specifications, by adding a new feature.
The software used for the implementation of the conference service should be based on open-source/free availabale solutions. The developed solution should also be deployed on an IMS network and compliant with the current specification, defined on the release 11[IMS11].
On the next sections the objectives will be explained on more detail, as well as the requirements which will be separated on 2 main sections, i.e. the Conference Service, which acts as a simples conference provider that allows the users to joyn a conference call with provided PINs (host and attendee), and the Group Call feature added to a RCS Client, which enables value-added functionalities on the conferences, simplifying the iteraction between the user and the conference and also allowing to setup conferences on the go.

3.2. Methodology

Since the technologies on the industry are suffering quick changes nowadays, and at any time updates can be made on the current specifications, the chosen methodology was Scrum. This requires a Product Owner and a Scrum Master, which lead a development team without to many constraints, as the decisions are made with the input of all the participants on the process, some more frequently than others, and without too many obstacles on the development of the product's features. On a top level, the priority of the features is based on the market's evolution, adjusting the bakclog's items priorities to it.

Table 6 - SCRUM Roles on the Internship

Role	Name
Product Owner	Paulo Sousa
Scrum Master	Carlos Ferreira
Development Team	Filipe Henriques

The sprints were defined to have a duration of 2 weeks, allowing a flexible way to change the priority of the features very quickly, adapting the product to the demands of the industry.

An illustration of the used process is presented on Figure 11, which also presents the involved parties at each step of the process.

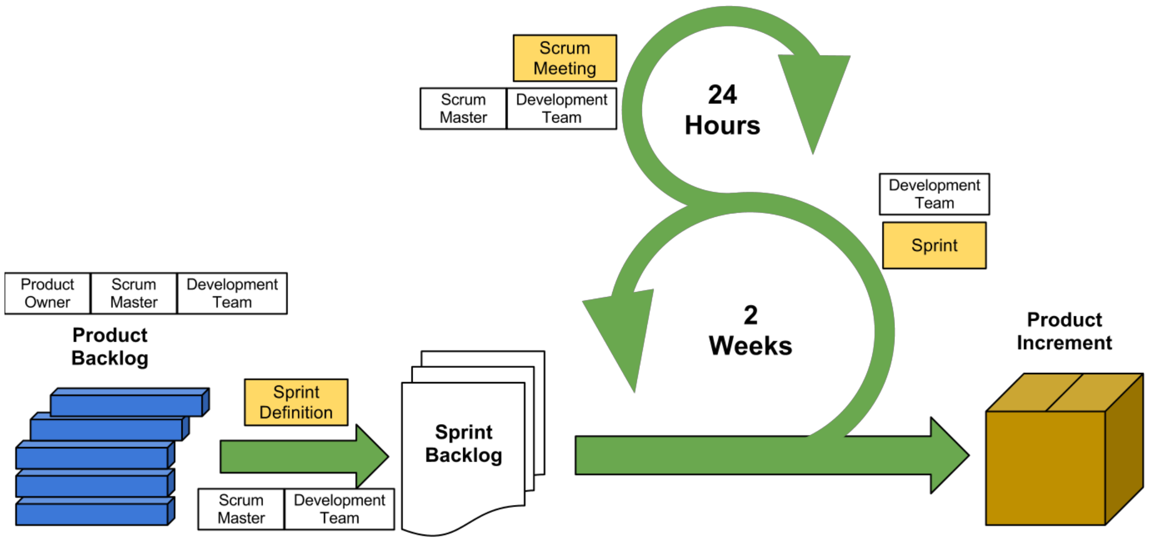


Figure 11 - SCRUM Framework

On the sprint definitions a set of user stories from the product backlog are moved to the sprint backlog, which were previously estimated on an effort/points classification, usually based on the one that seems simpler in terms of complexity. Based on these estimations, on each sprint definition the backlog will be fullfilled based on a max effort definition, fitting the user stories to it. The daily scrum meetings are meant for discussing the difficulties and progress made on the work developed on the previous day.

As the process is repeated, the max effort for each sprint is redefined (if necessary) on each sprint retrsopective, which provides a time for both the team and scrum master to overview what went well or wrong on the last sprint. This allows the team to create a better estimation of each individual story from the backlog and the sprints efforts that are tightly coupled to the team performance and well-defined estimations.

3.3. Conference Service

The service should provide basic conferencing support for the users, allowing to setup conferences, which have 2 PIN's associated with them, one for the moderator and another for the participants. The conference will only start when the moderator joins for the first time. Before that, if the participants enter the conference, they won't be able to talk to each other and shall only be allowed to hear a waiting music.

The moderator should have the possibility to enter a pre-defined code to mute or unmute all the participants.

It will be provided an administration website, so that the system administrator is able to verify the active conferences, as also the number of users present on each conference and the state of the same (active, waiting for moderator or inactive).

The service has also a component which is here called Notification Service, which is responsible for managing the users subscription to the conference event packages, which allows to notify the users about status change of the conference participants and also provide information about the conference (PIN and break-in number).

It should be also provided the possiblity for a regular phone call (through GSM network) to join a conference, using a break-in functionality which in this case will be provided by IPBrick that is already setup on the company, allowing very easily to integrate with the service developed on this project. It allows the server to have a related phone number to it, for which the users will be able to call into the coference, which provides NAT functionlalities from the comapnie's intranetwork to the user's equipment network.

Below on Table 7 is presented an overall view of the requirements of the Conference Service.

Table 7 - Conference AS High-Level Requirements

Requirement	Summary
[CNF1]	Request PIN authentication before redirecting user to a conference
[CNF2]	Provide audio mixing on the conference
[CNF3]	Terminate call with user when max PIN insertion attempts are reached
[CNF4]	Play annoucement when user correctly inserts a PIN matching a conference
[CNF5]	Play annoucement when user inserts an invalid PIN
[CNF6]	Play annoucement when user has reached max PIN insertion attempts
[CNF7]	Allow a user to join a conference as participant
[CNF8]	Allow a user to join a conference as moderator
[CNF9]	Play hold music while the participants are waiting for moderator
[CNF10]	Play an annoucement when someone joins the conference
[CNF11]	Play an annoucement when someone leaves the conference
[CNF12]	Allow a user to mute himself
[CNF13]	Allow a user to unmute himself
[CNF14]	Allow moderator to mute participants
[CNF15]	Allow moderator to unmute participants
[CNF16]	Allow a conference-aware participant to subscribe the conference event
[CNF17]	Receive a request from a user to invite other users
[CNF18]	Invite other users into the conference which were invited by other user
[CNF19]	Notify participants subscribed to the conference event about changes on the conference participants presence and conference information

[CNF20]	Provide break-in functionality for internetworking with GSM calls

An illustration of the Conference Service is shown on Figure 12, describing which type of components and functionality should be present on the developed solution. It's also presented the used protocols to communicate between each other and which components interact with each other.

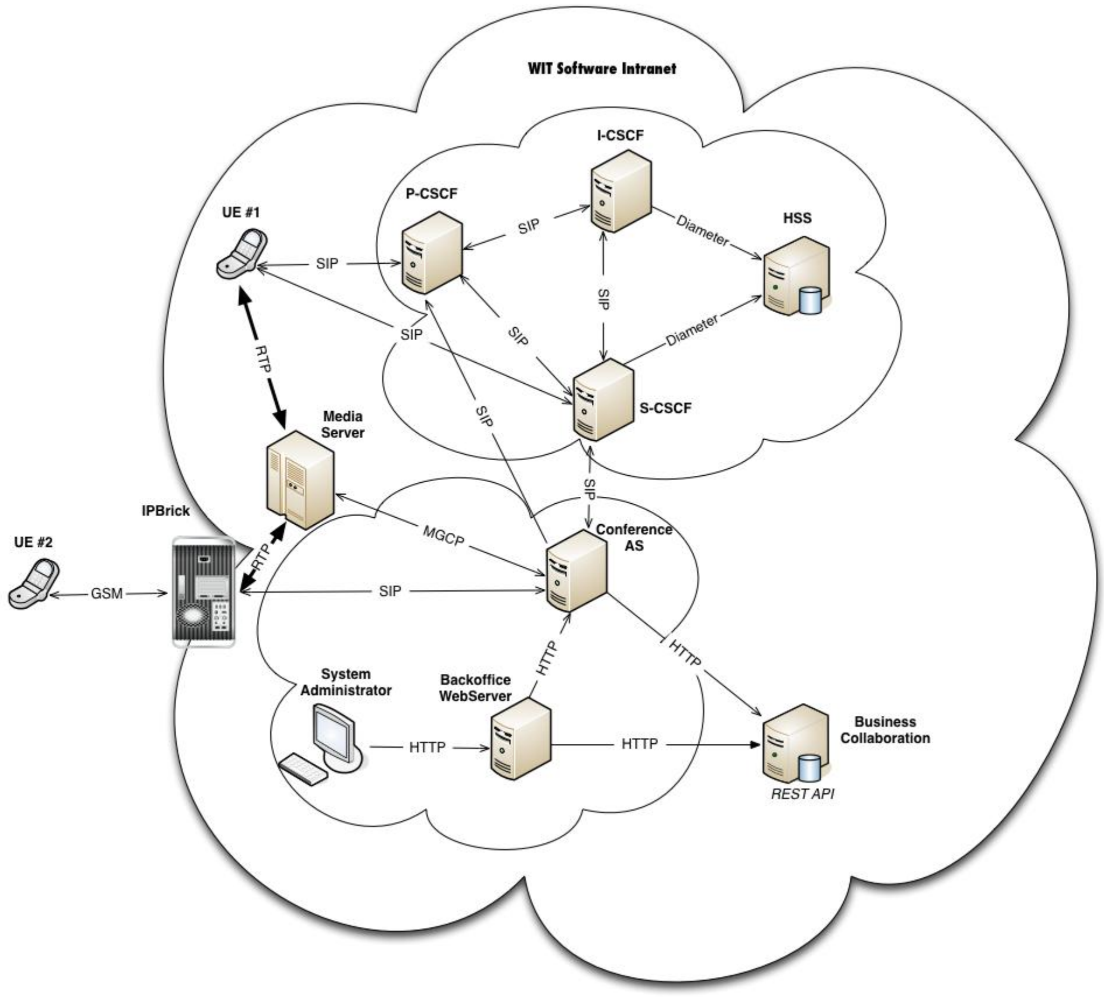


Figure 12 - Conference Service High-Level Architecture

There are also a set of non-functional requirements which the service should fulfill, being these described on Table 19. As

Table 8 - Conference Service Non-Functional Requirements

Requirement	Summary	Description
[CNN1]	Use standard protocols	Use standard and well established protocols on the industry
[CNN2]	Not dependent on 3rd party software	The providers shall not be dependent on other companies for fixing or updating certain features of the software, i.e. the software can easily be exchanged by others

		since the solution shall use standard protocols
[CNN3]	IMS Release 11 compliant	The solution shall be compliant with the IMS Release 11 for deployment on one network architecture of this type
[CNN4]	Use open-source and/or free software	Use software which is free of charges and/or provides the source code
[CNN5]	High-throughput	The service should handle a good number of connections/second as well as a wide number of user connected at the same time across several conferences (of course this is depentend on the hardware being used)
[CNN6]	Compliant with RFC4353[RFC4353]	Compliant with the defined standard on how to plan and develop a conferencing framework
[CNN7]	Compliant with RFC4575[RFC4575]	Standard that defines how to use the conference event package and what information shall be included on it
[CNN8]	Compliant with RFC4579[RFC4579]	Specifies how an AS shall act in order to offer conference control features to clients compliant with this RFC

For the requirement [CNN5], which specifies that the solution shall provide/support High-Throughput, it shall be verified with a tool named SIPp[SIPP], which allows to define several different scenarios and run against the developed solution. It offers information on the number of calls failed, the number of messages retransmission, total time of the test, etc.

3.3.1. Administration Website

On this website, the system administrator shall be able to verify information and status of the system where the media server is deployed (CPU, memory, disk and network). There should be also available information about the number of active conferences and the state of every conference scheduled.

Table 9 - Conference Service Administration Website High-Level Requirements

Requirement	Summary
[CFA1]	Create a conference by generating the moderator and participant PINs
[CFA2]	Present conferences' moderator and participant PINs
[CFA3]	Present conferences' status (active, waiting for moderator or inactive)

[CFA4]	Show conferences' current users number
[CFA5]	Number of Conference Endpoints being used
[CFA6]	Number of IVR Endpoints being used
[CFA7]	Media Server System Load

The website should only be available to the system administrator as specified on Table 10.

Table 10 - Conference Service Administration Website Non-Functional Requirements

Requirement	Summary
[CFN1]	Should only be accessible from the intranet

3.3.2. Users Website

This part of the objective was withrawed, since it is currently being developed on another project on the company. In this case the developed service shall act as an enabler for other services. The project is a collaboration service, which offers several capabilities like Group Chat, Conference Call, etc. For this reason this part related to this project was abandoned.

3.4. Group/Conference Call - RCS Feature

This feature will be a custom service developed based on the RCS specification for Group Chat, more precisely on the version 5.1 of it [RCS5_1]. The specifications allows the development of custom features on the top of the existant ones, which allows to take advantage of the RCS capability discovery.

Table 11 - RCS Client High-Level Requirements

Requirement	Summary
[RGC1]	Verify contact capabilities
[RGC2]	Start conference call from a group chat
[RGC3]	Start conference call from a selected list of contacts
[RGC4]	Receive inviation to a conference call
[RGC5]	Subscribe to the notification service of the conference call
[RGC6]	Receive notifications with information related to the conference participants
[RGC7]	Invite selected users from list of contacts to an ongoing conference
[RGC8]	Allow user to mute himself
[RGC9]	Allow user to unmute himself
[RGC10]	Mute selected participants as moderator

[RGC11]	Unmute selected participants as moderator
[RGC12]	Turn speaker on
[RGC13]	Turn speaker off
[RGC14]	Send conference information through email
[RGC15]	Send conference information through SMS
[RGC16]	Exit conference

On the above table, the High-Level requirements for this feature are enumerated, allowing to give an overall view of the objective of the feature. A few more were thought, still the limitations of the media server that was selected (explained below), stops from being able to acomplish them. Some example of these are video conferencing and focus on the current person talking.

Below on Figure 13, is described the architecture of the system and the existant interfaces to communicate between the different components, as well the procotols used.

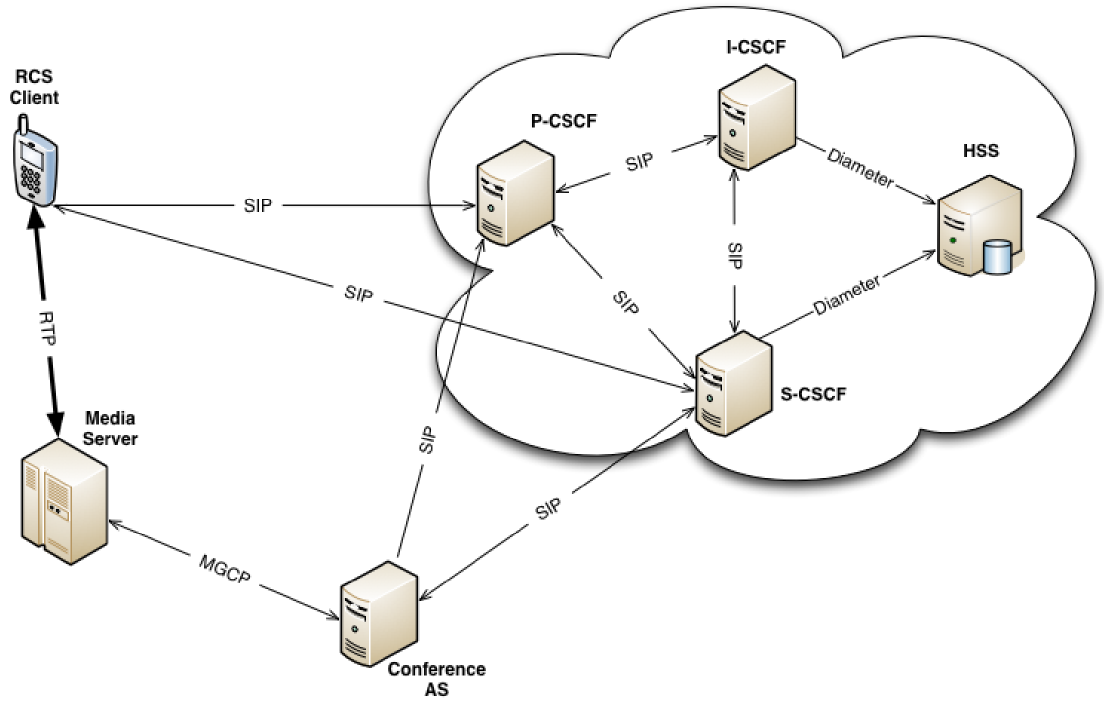


Figure 13 - RCS and Group Call Feature High-Level Architecture

3.4.1. Feature and Application

On this section are described the requirements both in terms of specification and application.

The only non-functional requirement is that it is compliant with version 5.1 of the RCS specification [RCS5_1], so taking also advantage of the Group Chat definitions present on it.

Table 12 - RCS Group Call Feature Non-Functional Requirements

Requirement	Summary	Achieved

[RGN1]	RCS 5.1 Compliant	✓

3.4.2. User Interface and Notifications

The UI requirements for the conference service that is going to be integrated with the existant application on the company (joyn from WIT Software for Android) are described here.

On the table below (Table 13) are presented the required notifications that should appear when determined actions are taken by the moderator, or certain events ocurr on the conference.

Table 13 - RCS Group Call App UI Requirements for Notifications

Requirement	Summary	Description	Roles
[RUI1]	Muted by the moderator	The moderator selects a participant and mutes him until further action	Participant
[RUI2]	Allowed to unmute by the moderator	After selecting the partcipant the moderator unmutes the user, allowing him to talk to the conference	Participant
[RUI5]	User joins the conference	A user joins the conference either by invitation or by dialing in	Participant, Moderator
[RUI6]	User leaves the conference	The user loses connection, leaves the conference or is removed from the conference by the moderator	Participant, Moderator
[RUI]	User is invited to the conference	The user is invited to the conference, appearing as pending	Participant, Moderator

3.5. Software

Since one of the main goals of the project is to use open-source solutions, all the software was chosen according to this constraint, after evaluating the most fited solutions available for free, according to the previously described requirements. Furthermore, some solutions for certain components were chosen due to being the software used/developed by the company (joyn, WIT Business Collaboration).

Table 14 - Used Software

Module	Solution	Version
IMS - HSS	OpenIMS - FHoSS	revision1182

IMS CSCF - I-CSCF, S-CSCF, P-CSCF	OpenIMS	revision1182
Media Server/Media Gateway	Mobicents Media Server	3.0.0 Final
AS Development Techonlogy	JAIN SLEE	1.1
Container/Application Server	Mobicents JAIN SLEE	2.7.0
RCS Client	WIT Software - joyn Android	
Configuration Server	Standalone AutoConfiguration	
Conference Information Storage	WIT Business Collaboration	

3.5.1. IMS - OpenIMS

After evaluating all the available solutions for both the CSCF components and HSS, OpenIMS was chosen for setup of an IMS network. It has an active community and is currently under development for improvement, bug fixes and compliance with the most recent changes to the protocols that are part of the IMS network (ex: diameter). Since the solution is also used on the company for prototyping purposes, the company collaborators could also provide some guidance if any difficulty appeared while setting it up. It should be noted that the solution although is viable as a development environment for testing and experimenting, it shall not be used on a production environment. Even the contributors to the project specify that it's purpose is not to be used on production scenarios. On production environments, the Service Provider's IMS network solution is used.

3.5.2. Media Server/Media Gateway - Mobicents Media Server

Since one of the objectives of this project is to use available open-source solutions on the development of the work, this seemed the better option among the other ones available, since it supports several needed functionalities and has an active community that is currently fixing problems, improving the current implementation and developing new functionalities. Although it currently doesn't support video conferencing, it's development is currently being started. The video functionality can be added to the developed solution once it's development is over.

It was decided not to use other media server, since this is the only one that provides functionalities for accessing conferences like DTMF detection, IVR endpoints and similar. For this reasons the MCU Medooze Media Server was discarded.

3.5.3. Application Server Development Technology - JAIN SLEE

Before deciding on which technology should be used to develop the service, some simple scenarios were implemented on both SIP Servlets and JAIN SLEE technologies. As previously mention JAIN SLEE has a very hard learning curve when compared to SIP Servlets, especially when the developer already has experience developing JEE applications. Although it was a bit difficult to start using the JAIN SLEE technology for development, it was easily perceived how it is very easy to take advantage of it's well defined transactional environment [SLEEVSVL].

3.5.4. Container / Application Server - Mobicents JAIN SLEE

The only available open-source or free solution that supported the deployment of JAIN SLEE services/applications was the JAIN SLEE container/application server developed by Mobicents.

3.5.5. RCS Client - joyn by WIT Software for Android

Due to the fact that this is a product being developed on the company, it was decided that it would be used for implementing the Group Call RCS feature proposed on this document. It will not be part of the oficial development, but instead as a proof of concept for demonstrating the feature working along the other ones implementend on the product that are compliant with the RCS-e version [RCS_e] of the specification.

3.5.6. Configuration Server - Local Configuration

Since the joyn application from WIT Software supported local configuration for development purposes, it was decided to use this option in order to configure the servers and account configurations of the application. For this, several xml files can be distributed with the application, containg the configuration details, allowing the tester or developer to choose which configuration he wants to use.

3.5.7. Conference Information Storage - WIT Business Collaboration

The initial objective of the internship was that it would be inserted on a common project along with other internships, which would be separated by Conferencing Service, Instant Messaging and Business Collaboration Service. The Business Collaboration Service would be responsible for keeping the information centralized on one point for both the services, providing also an interface for interacting with both services. Since the feature of maintaining the Conference Service data on the Business Collaboration database was developed before the project suffered several changes on the objectives, these was the chosen storage point for the conferences' data. This decision will still allow to develop a single application where the user can have both conferencing and instant messaging features as also other functionalities.

3.6. Protocols

Table 15 - Used Protocols

Protocol	Description	Transport Layer	Secure Transport Layer/Protocol
SIP	Client-IMS CSCF, IMS CSCF-Conference AS and IPBrick-Conference AS signaling protocol	UDP/IP or TCP/IP	SIP over TLS or IPSec
MGCP	Conference AS-Media Server control for endpoints and connections management	UDP/IP	MGCP over IPSec
RTP	Real Time Media (voice exchange)	UDP/IP	Secure RTP(SRTP)
HTTP	Conference AS-Business Collaboration database information access and modification.	TCP/IP	HTTPS

3.6.1. SIP

Since the main focus of the project is to develop a service using the SIP protocol, this is the protocol that is being used for signaling between the client and the developed Conference

AS. Besides that, the signaling protocol specified on the network architecture being used for signaling between the UA and the AS (with some other components between) is SIP.

3.6.2. MGCP

In order to communicate and control the media server, MGCP was the chosen protocol. Although Megaco/H.248 is the defined protocol on the IMS specification for controlling the MRFP (media gateway/media server), none of the available open-source solutions had support for it. After choosing the solution to use as the media server, 2 protocols were avaiable, MGCP and JSR-309. The choice ended up being MGCP, since the support for JSR-309 is much more limited on the current vesion of the MMS and the syntax/operations of the MGCP protocol are very similar to the ones defined on the Megaco/H.248 protocol. Also, the JSR-309 implementation on the Mobicents Media Server is made on top/based on it's MGCP protocol adaptor.

3.6.3. RTP

On the IMS architecture, voice and video media should be transmitted using the RTP. Also, since the network establishes that voice and video transmission should be done over IP (with signal converters between the network and the UE when necessary) and one of the most popular protocols for media transmission over IP is this one.

3.6.4. HTTP

For the database part of the Conference Service, the API provided in REST must use the HTTP protocol, since the RESTful webservices rely on the methods provided by the HTTP protocol.

The chosen protocol for providing the administration backoffice was REST. It allows to transport data independently of the used format/protocol (XML, text, JSON, etc.), which allows to easily change the format to represent the data on the future if needed, without having to change the transport protocol.

Chapter 4
System Description

On this chapter is explained the procedures and flows specified for achieving the objectives and requirements previously defined. An architectural overview of both Conference Service and Group Call was already provided on Figure 12 and Figure 13, so in this chapter is described both the features necessary to fullfil and messages exchange procedures.

4.1. Conference Service

The service allows users to schedule conferences and invite other users to join by providing the PIN to them. There are 2 types of PIN per conference that identifies 2 types of users:

- Participant
- Moderator

Both the PINs follow the same pattern, i.e. they are always composed of 5 digits ending with an #. This was defined on the Business Collaboration platform.

When the participants join a conference on which the moderator hasn't yet entered, they will be hearing hold music until the moderator enters the conference. Each participant will have the ability to mute himself or unmute and turn his video on or off. The moderator has also the ability to mute all the other participants or unmute them and to turn all the other participants video on or off. Below on Table 16 are presented the codes that provide these functionalities.

Table 16 - Conference Service DTMF Codes for Media Control

Role	Action	DTMF Codes Sequence
Participant or Moderator	Mute myself	*1
	Unmute myself	*2
Moderator	Mute participants	*5
	Unmute participants	*6

The moderator doesn't necessarily need to be only one user at a time per conference. Each user that has the conference's moderator PIN can be a moderator, yet no moderator will be able to control the media stream of another one. The conferences are considered terminated when there is no one left on them and the moderator has previously entered the conference.

4.1.1. Use-Cases

On this section, a brief description is given on each use-case of the Conference Service, although the name of each one is very much self-explanatory.

- **Place Call to Conference as Participant**

The user places a call to the conference number previously provided. Then the user is asked to input a PIN associated with the conference that he wants to enter (in this case the participant PIN). After successfully inserting the PIN, the user is redirected to the conference.

- **Place Call to Conference as Moderator**

  On this use-case, the user calls the conference number and when is asked to input the PIN, it enters the moderator PIN associated with the conference he wants to participate in. When the PIN is inserted with success, the user is redirected to the conference.

- **Moderator**
  - **Mute participants**

    After the moderator inserts the code correspondent to this functionality, all the conference participants that are present or enter before the user unmutes them, will not be able to speak to the conference.

  - **Unmute participants**

    When the moderator inserts the code for unmuting the participants, all those that were muted by the moderator will be able to speak for the conference again or for the first time in the case they entered after the moderator muted the participants.

- **Mute myself**

  The user introduces the correspondent code and after that anything that he says will not be transmitted to the conference if the moderator didn't already mute him.

- **Unmute myself**

  If the user isn't currently muted by the moderator, it will be able to talk again to the conference after inserting the correspondent code if he previously muted himself.

The "media control" use-cases follow a very similar sequence of messages, dependent if either is the moderator or the participant to perform it. If it is the participant, the conference Focus will send a RE-INVITE request back to the user in order to change the media streams. If it is the moderator, a RE-INVITE will be sent to all the conference participants.

4.1.2. Sequence Diagrams

On this section, some sequence diagrams are presented for describing the necessary exchange of messages among the different components present on the service architecture (Figure 12), which serve as implementation guidelines. The developed AS will act as an MRFC, since it will fullfil the responsability of managing the MRFP resources (Media Server). More sequence diagrams can be found on Annex C, section C.1.2.

4.1.2.1. Place Call to Conference

On this diagram is presented the sequence of messages necessary to setup a call between the user (UE) and the conference service. On this case, the user will be asked to input the PIN associated with the conference that it intends to enter.

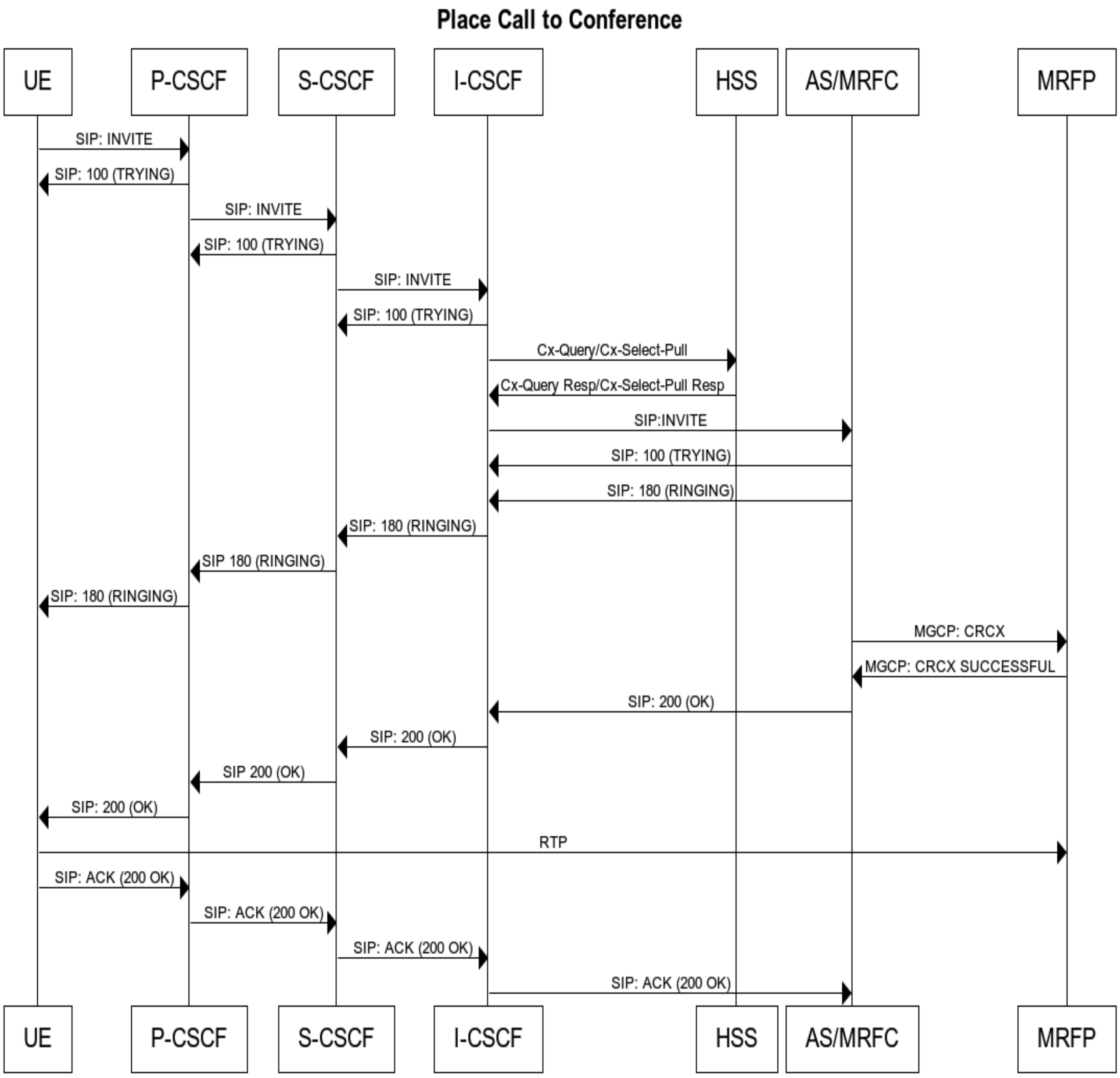


Figure 14 - Conference Service Call Processing Sequence Diagram

4.1.2.2. PIN Insertion

The diagram starts with a loop which represents the maximum attempts for the user to input a correct PIN. It uses a the PlayCollect functionality of the MGCP protocol, which allows to request the Media Server to collect a set of DTMF codes in this case that matched a defined pattern and play an initial annoucement and different annoucements if it either matched or not the pattern.

Table 17 - MGCP Command RQNT PlayCollect Request Parameters

Parameter	Value
Pattern	[0-9]{5}#
Initial Annoucement	"Please input the conference PIN"
Failure Annoucement	"Please try again"

As previously described, the Conference Service depends on an external service for maintaing the information of the conferences scheduled and the PINs associated. As presented on the diagram, the service provides a REST API for accessing this information.

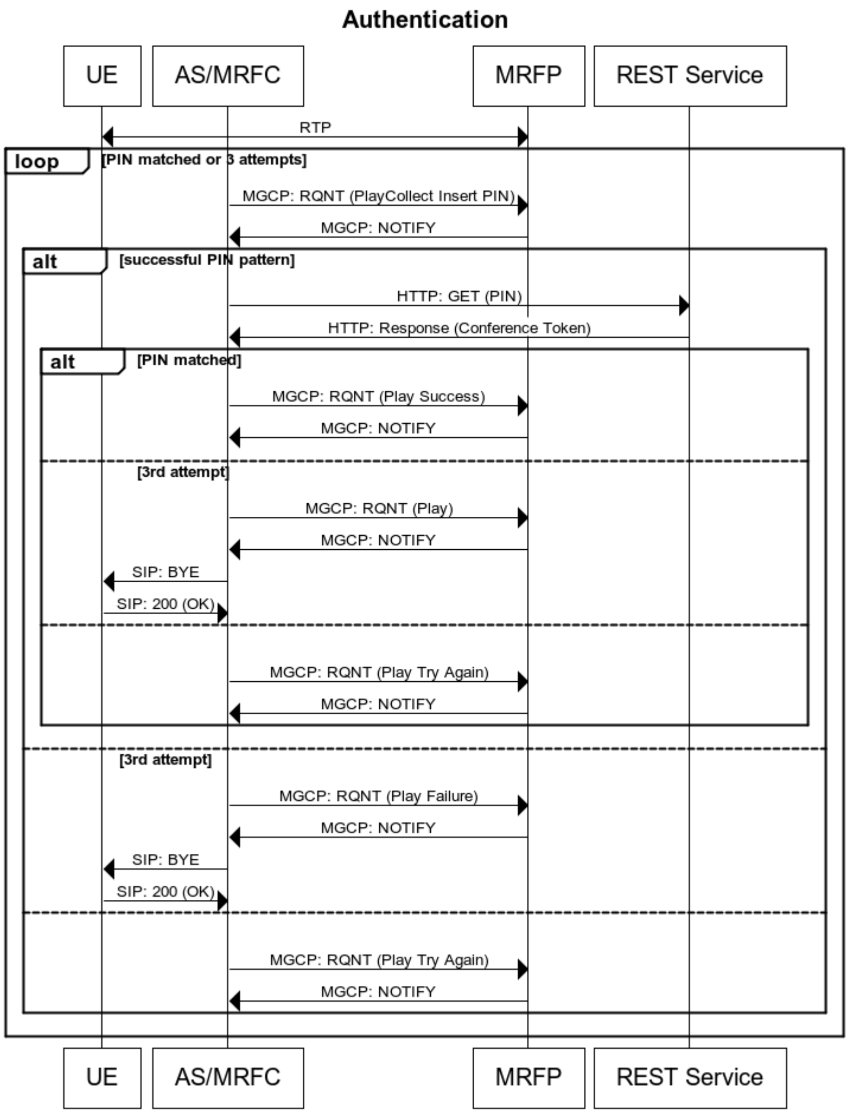


Figure 15 - Conference Service User PIN Insertion and Verification Sequence Diagram

4.1.2.3. Participant Joins a Conference

The diagram is a continuation of the previous one, when the PIN is correspondent to a participant one and is successfuly inserted. The user is first redirected to the conference, which contains 2 endpoints, an IVR one for annoucements and detection of DTMF codes input. The other endpoint is a Conference endpoint, which is responsible for mixing the audio and video transmitted from all the connected users and distribute to back the resultant mix to all of them. The described setup is illustrated below on Figure 16.

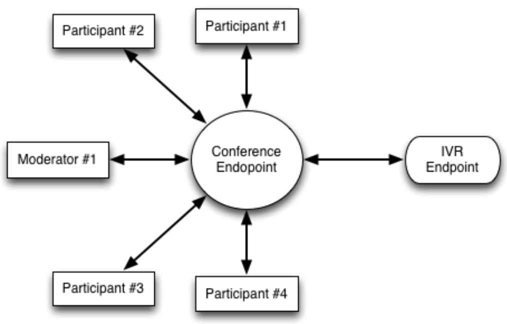


Figure 16 - Conference Service Endpoints Illustration

When a participant enters and the moderator hasn't already entered the conference, it will be put on a waiting status, with hold music. At this point, the participants won't be able to talk to each other. Once a moderator enters, the conference starts, and the participants' media sessions are modified, allowing them to send audio and/or video data to the conference. Everytime a participant or moderator enters or leaves once a conference is active, an annoucement will be played to all the users present on the conference.

It is also possible to verify on this diagram, the necessary procedures necessary to execute when a user leaves a conference.

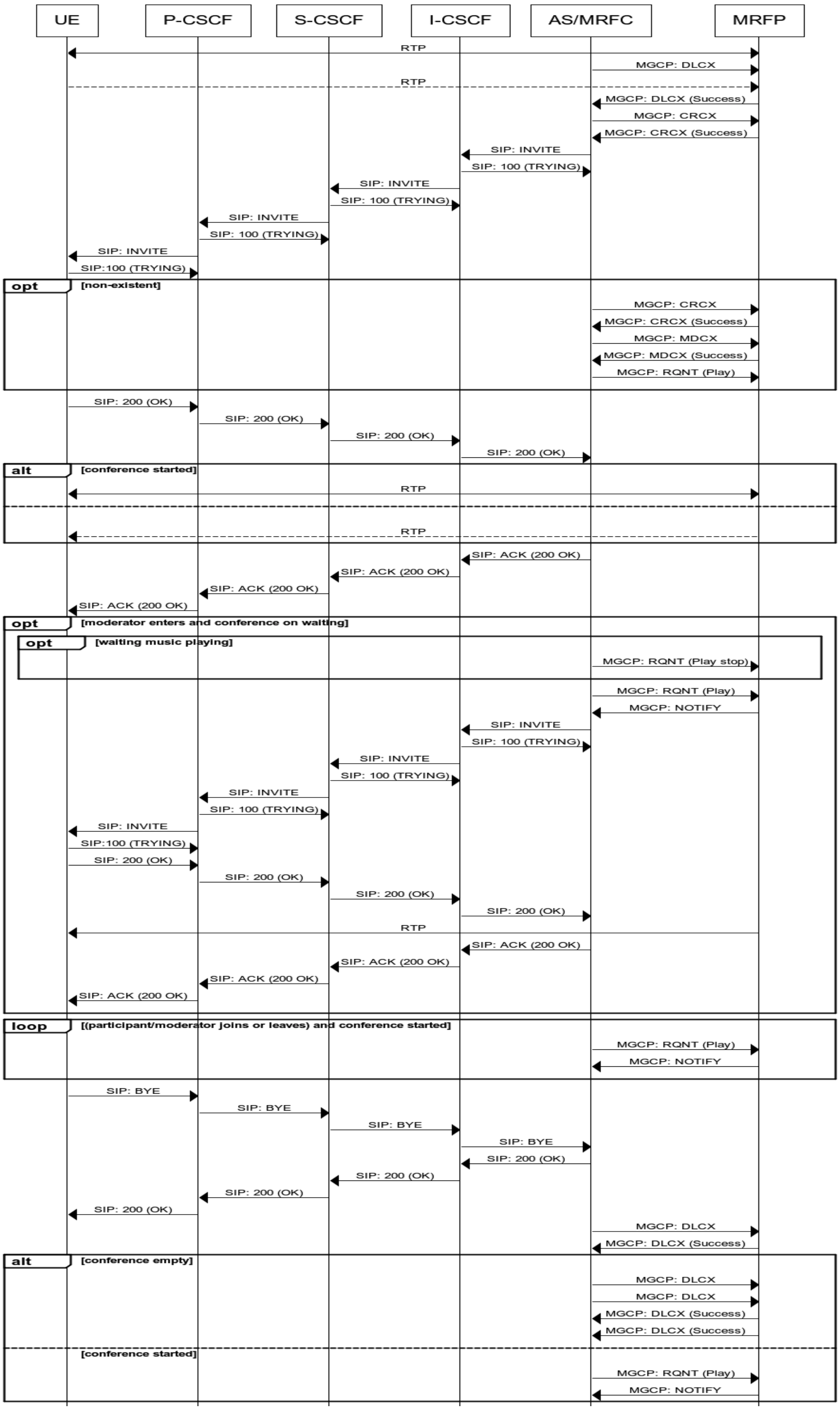
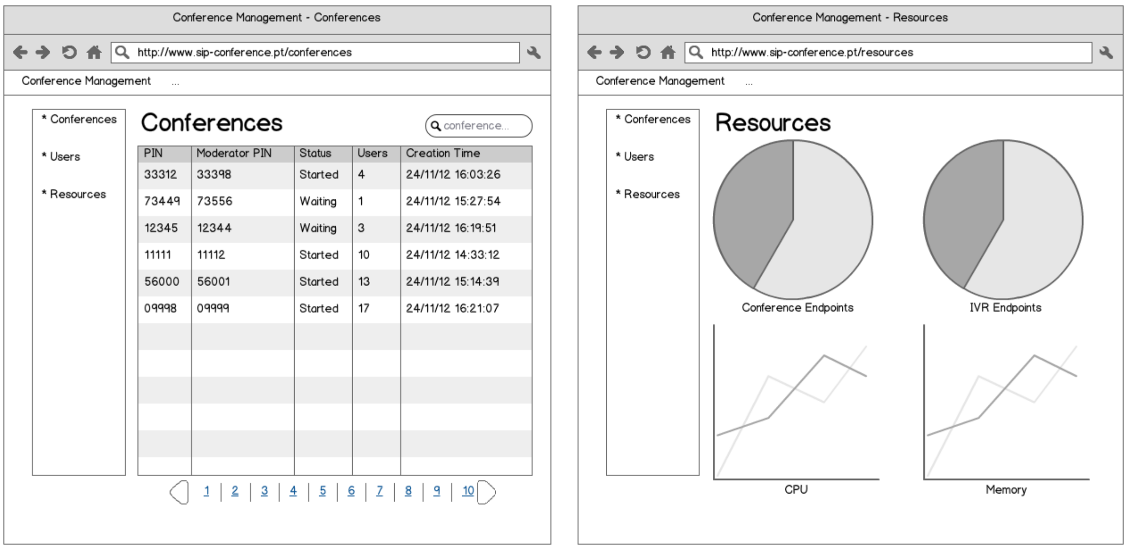


Figure 17 - Conferencing Service Participant Joins Conference Sequence Diagram

4.1.3. Administration Website

On this webpage, the administrator shall be able to verify which conferences are active, the number of users on each conference and create new conferences. It provides a listing of all the scheduled conference, as well as their state and number of users currently present on each one. It will also show the date and hour of when the conference started (if it's state is currently active).

The administrator will also have information on the state of the computational resources of the system where the Media Server is installed, and will also be able to verify the endpoints currently allocated on it.


Conference Management - Conferences
http://www.sip-conference.pt/conferences
Conference Management ...
* Conferences
* Users
* Resources
Conferences
conference...
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
Conference Management - Resources
http://www.sip-conference.pt/resources
Conference Management ...
* Conferences
* Users
* Resources
Resources
Conference Endpoints
IVR Endpoints
CPU
Memory

PIN	Moderator PIN	Status	Users	Creation Time
33312	33398	Started	4	24/11/12 16:03:26
73449	73556	Waiting	1	24/11/12 15:27:54
12345	12344	Waiting	3	24/11/12 16:19:51
11111	11112	Started	10	24/11/12 14:33:12
56000	56001	Started	13	24/11/12 15:14:39
09998	09999	Started	17	24/11/12 16:21:07

Figure 18 - Conference Service Administration Website Mockups

4.1.3.1. Use-Cases

- **Create conference**

  After pressing the button for creating a conference, the system should generate 2 unique PINs (participant and moderator) and present them to the user

- **Present conferences**

  Present information on the status of each conference, as well as it's PINs, number of users in it and also the time when the conference started after the user selects the link for viewing the conferences information

- **Present system load and resources**

  When the user selects to see the system information, information about the CPU, memory, disk and network usage should be presented to him from the machine where the media server is deployed

- **Present media server allocated resources**

  After selecting the option to check the system information, the user shall be able to see the media server allocated resources (ex: conference endpoints and IVR endpoints)

4.2. Group Call - RCS Feature

The description of this feature is based on the RCS 5.1 [RCS5_1] specification of the Group Chat feature, following the same structure so that readers, which are already familiar with it, may quickly and easily comprehend the procedures defined on this document for implementing the proposed feature. For maintaining compliance with the Group Chat functionality, the configuration parameter CONF-FCTY-URI for indicating the focus URI is named as CONF-CALL-FCTY-URI for the Group Call feature.

The full specification of the feature, as well as some sequence diagrams describing the flow of messages exchange in order to execute a particular functionality, is presented on Annex C, section C.2.

4.2.1. Feature description

The Group Call service enables the users to perform voice and video communications between more than one user intantly, by providing a centralized point where all users part of a Group Call session are connected, mixing both sound and video streams from each one and sending the result to each user. It also provides information about the users present on the current sessions as also the media status of each one (audio and video).

The following RCS features are described:

- Interworking of participants in a Group Call

  This feature allows participants without a device without the Group Call capability to participate on a Group Call.

- User Alias (Display Name)

  When starting a new session, a user defined display name can be defined to be shown to the other Group Call participants.

- Leaving a Group Call

  If a user leaves a regular Group Call, will be able to rejoin it later if it is still ongoing. If the Group Call is marked as Closed a user won't be able to rejoin after leaving it

A user will only be able to start a Group Call if the Service Provider it belongs to has a Conferencing Server deployed.

If a Conferencing Server is deployed, the Service Provider should have the configuration parameter CONF-CALL-FCTY-URI set, which determines the URI to where the terminal will send the SIP INVITE request for initiating a Group Call.

Since this feature is a custom one (not meaning the contrary even if it was part of the RCS specification), a Service Provider is not obligated to provide the Group Call functionality, i.e. if there is not present any value for the CONF-CALL-FCTY-URI configuration parameter, the user shouldn't be able to initiate a Group Call. Although, even if the user hasn't got the capability to start a Group Call, it doesn't mean that it shouldn't be able to join a Group Call session.

4.2.2. High Level Requirements

Next is presented the high level requirements of the Group Call feature:

- Client devices:
    - Ability to request a regular Group Call or a closed Group Call
- Conferencing Server:
    - Provide interworking on a Group Call between users without an RCS device
    - Provide a regular Group Call or a closed Group Call and must indicate to the the participants if the Group Call is closed or not

4.2.3. Technical Realization

4.2.3.1. Initiating a Group Call

When a user wants to initiate a Group Call, he starts by selecting some of his cotacts either from the address book or from the Group Call application. The number of maximum allowed participants is defined on the configuration parameter MAX_AD-HOC_GROUP_SIZE. All the contacts from the address book are selectable, unless the configuration parameter GC_NON_RCS isn't defined, which means that interworking with devices that don't support Group Call capability isn't available from the Service Provider. In this case the user will have to use the capability exchange service to verify if the user supports or not the Group Call capability.

When the user selects the contacts that he wants to invite, a SIP INVITE request to the CONF-CALL-FCTY-URI with the participant list is sent. At this point the focus will send SIP INVITE requests to all the participants indicated on the initiator's request. The invited users devices shall show a notification saying that is an invitation to a group call.

The users are able to either accept or reject the invitation. When the first user accepts, a 200 OK will be sent to the initiator's, setting up the Group Call.

A Group Call initiator can place a subject for the Group Call, which should appear on the invitation notification to the other users.

When the users join the Group Call, they should subscribe to the conference event package using the URI either provided on Contact header of the SIP 200 OK response for the Group Call initiator or the SIP INVITE request to the invitees.

4.2.3.2. Adding participants to a Group Call

The participants may or not be able to add other participants to a Group Call, depending on the policy of their Service Provider, which may define that only the Group Call initiator is able to add participants to it.

If the conference event package provides a value for the maximum-usercount and user-count elements [RFC4575] when a participant subscribes to the conference notification service of the Group Call focus, it may add participants until that maximum is reached. If the parameter is not defined, the MAX_AD-HOC_GROUP_SIZE shall be used.

If a Closed Group Call is ongoing, no one can add participants to it and a SIP Error response shall be sent to the clients that try to.

4.2.3.3. Closing a Group Call

A participant can close it's session to a Group Call from the Group Call application. When the user ends it's sessions, a SIP BYE request will be sent to the focus and the participant's device shall unsuscribe the conference event package. When this occurs the client shall set it's state do disconnect, and a notification should appear on the Group Call participants devices that the user has left the Group Call and shall be removed from the participants list.

A Group Call session shall be closed when there are no users present in it.

4.2.4. Implementation guidelines and examples

The presented interfaces are presented just for example and guidance, they don't specify any concrete UI requirements. First are defined the entry points to the service and the associated UX. After that a set of flows is presented as well the UI changes along those

4.2.4.1. Entry points on the group call service

There are 3 ways to start a Group Call as presented on the next UX:

1. Address book/Call-log

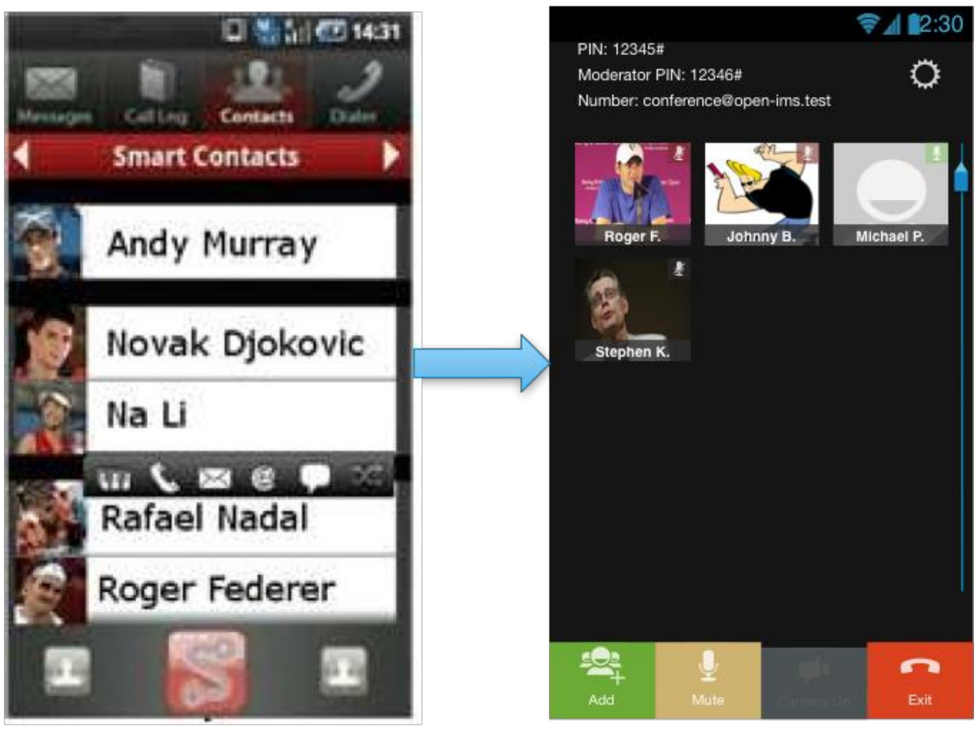


Figure 19 - Group Call - Address Book Entry Point UX (Adapted from [RCS5_1])

2. Group Call Application

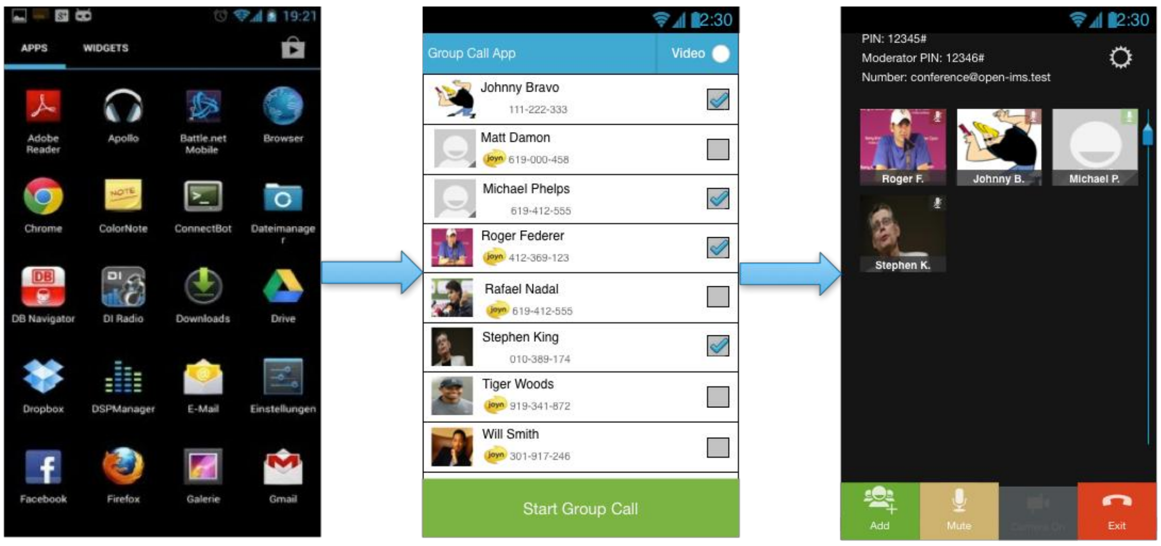


Figure 20 - Group Call - Application Entry Point UX (Adapted from [ANDRS])

3. Group Chat

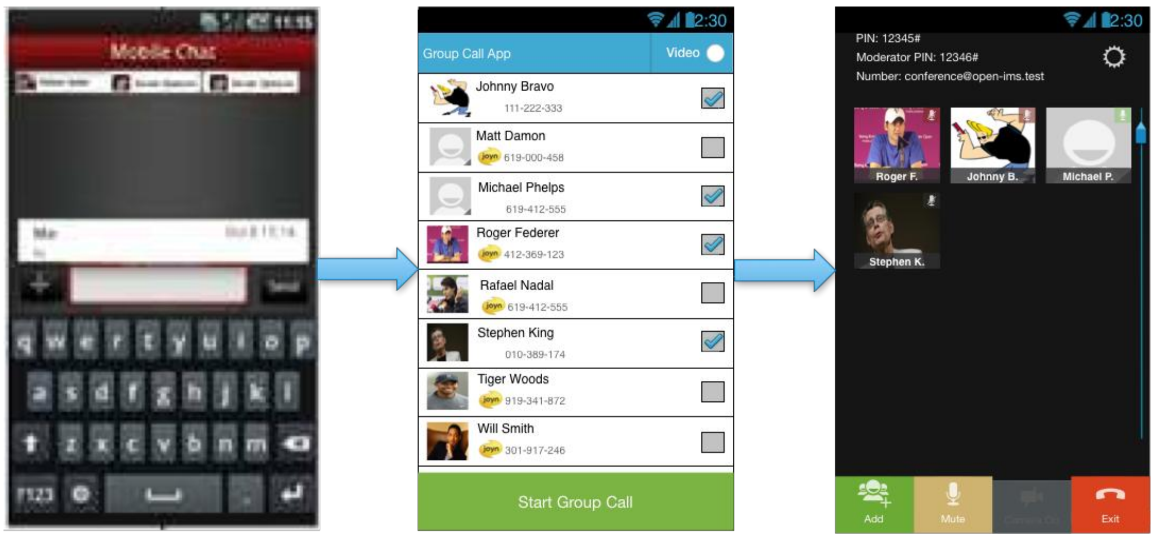


Figure 21 - Group Call - Group Chat Entry Point UX (Adapted from [RCS5_1])

4.2.5. RCS Client - joyn by WIT Software for Android

The architecture of the joyn by WIT Software for Android, relies on a library called COMLib. This one is responsible for executing all the logical operations necessary to perform any functionality related to the RCS specifiaction.

For this new feature (Group Call) to work, it will be necessary to develop a new service on the library in order to performs all the necessary logical operations related to the feature. Since the current architecture of the application allows to decouple the client logic from the presentation view (layouts) it is very easy to change or implement new layouts for the application. Further details on this library can be found on Annex E, which might not be available since it is confidential.

As for the UI, new activities will need to be created that will be responsible to handle both incoming Group Call invitations and present the ongoing Group Call.

Chapter 5
Development

5.1. Testbed Setup

The testbed is composed of a single virtual machine, being it's allocated resources and used operating system described below on Table 18. This testbed will serve as host for the different servers needed for the development of the solution.

Table 18 - Testbed Resources and Operating System

Resource Type	Allocated Resources
CPU	1 Core (Unknown CPU)
Memory	2 GB DDR3 1333 MHZ
Storage Size	16 GB
Operating System	Ubuntu Linux 10.04 LTS

For each software, component or server installed, is created a single user space for it, allowing to isolate the user space from each others which is considered a good practice in terms of security.

5.1.1. HSS and CSCF Components - OpenIMS

A few modifications were necessary for the component to work, basicly new DNS entries had to be created for each of the components (P-CSCF, I-CSCF, S-CSCF). A few other modifications also had to be done on the P-CSCF so that it would listen for request coming from outside the localhost, so it was modified on it's configuration file to listen on the public IP of the host where it was installed.

5.1.2. Media Server/Media Gateway - Mobicents Media Server

It was quite easily to setyp the Media Server, all that had to be done was to specify the IP where it would listen for the Media Gateway Control Protocol requests (in this case the localhost was used since the AS and the Media Server are on the same machine). It was also necessary to configure it's public IP according to the host, which would be used for establishing RTP sessions with other peers.

5.1.3. Container/Application Server - Mobicents JAIN SLEE

The container installation was quite straightforward. On the other hand, the setup of the Resource Adaptor for SIP had to be configured before being deployed on the container, so that the OUTBOUT_PROXY would be the P-CSCF of the IMS network where it is located on.

5.2. Conference Service

The Conference Service fullfilled almost all of the Non-Function Requirements specified on Table 8 A load test was not realized, since the Container where the AS is deployed, the Media Server which is very resource hungry (mainly due to transcoding features), the IMS network and a webserver which has deployed the Collaboration Service that contains the conferences' information, are all in the same VM. Under this conditions and with the machine characterisitcs, the values would not be satisfatory. For realizing this test, the setup presented on the Figure 12 would need to be setup, and some changes would also needed to be done on the Conference Service, so that it wouldn't be necessary PINs for the users to join conference, and by that distribute a variable amount of users per conference.

Below is shown that except the previously mentioned requirement, which had some constraints associated, all the other Non-Functional Requirements were fully achieved.

Table 19 - Conference Service Non-Functional Requirements Completed

Requirement	Summary	Achieved
[CNN1]	Use standard protocols	✓
[CNN2]	Not dependent on 3rd party software	✓
[CNN3]	IMS Release 11 compliant	✓
[CNN4]	Use open-source and/or free software	✓
~~**[CNN5]**~~	~~High-throughput~~	✖
[CNN6]	Compliant with RFC4353[RFC4353]	✓
[CNN7]	Compliant with RFC4575[RFC4575]	✓
[CNN8]	Compliant with RFC4579[RFC4579]	✓

As for the acomplished High-Level Requirements presented at Table 7 it can be seen on Annex D section D.1. In terms of the detailed requirement analysis presented on Annex C, section C.1.1, all of them were fullfilled.

The final structure of the developed Conference Service, is presented at the figure below, which presents the developed services as well as the direction of the communication between all and the points from where a call or subscription is placed between the user and the Conference Service or the other way around (for invitations to the conference).

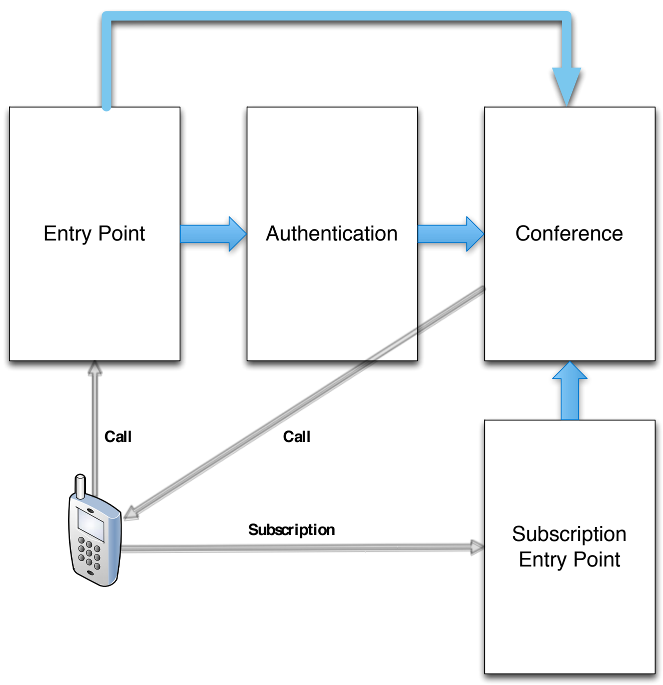


Figure 22 - Conference Service - Services overall view

Next is presented a description of which one of the services and the structure of the same, explaining which SBBs it contains and what are the responsibilities of each one of them. Notice that each service only has one root SBB (not to be confused with SBB entity), which can have several child SBBs.

5.2.1. Services

5.2.1.1. Entry Point

This service is responsible for registering the AS on the break-in component when the Conference Service is deployed/activated and maintaining that registration across all it's lifetime, before the current registration expires.

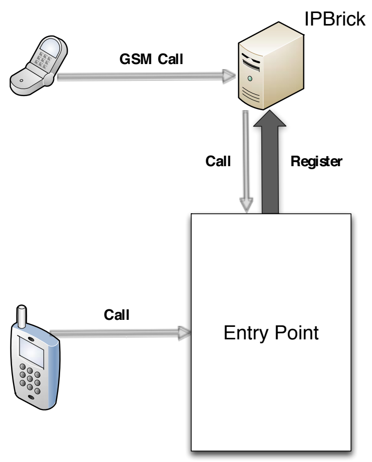


Figure 23 - Entry Point Service Illustration

It is also the first contact point between the user equipment and the Conference Service, where it is decided if the INVITE request which establishes a new Dialog is correspondent to a Group Call request or a regular call. For this the SBB entity verifies if the request only contains a Content-Type of application/sdp or else if it contains a multipart/mixed.

```
if (contentType.getContentType().compareToIgnoreCase("multipart") == 0
                            && contentType.getContentSubType().compareToIgnoreCase("mixed") == 0)
         {

                            StartGroupCallEvent gcEvent = new StartGroupCallEvent(request,
evt.getServerTransaction(), contId, null, null, null, subject);
                            this.setStartGCEvent(gcEvent);
                            this.setAci(aci);
                            try {
                                     ConfStatus confStatus = (ConfStatus)
this.getConfStatusSbb().create(ChildRelationExt.DEFAULT_CHILD_NAME);
                                     if(!confStatus.createConference())
                                     {
                                              respond(evt, Response.SERVER_INTERNAL_ERROR);
                                     }
                            } catch (Exception e) {
                                     e.printStackTrace();
                                     respond(evt, Response.SERVER_INTERNAL_ERROR);
                            }


         }
         else if (contentType.getContentType().compareToIgnoreCase("application") == 0
                            && contentType.getContentSubType().compareToIgnoreCase("sdp") == 0)
         {
                  SessionEvent sessionEvent=new SessionEvent(evt.getRequest(), evt.getServerTransaction());
                  this.fireAuthenticationInitiate(sessionEvent, aci, null);
         }
         else
         {
                  respond(evt, Response.NOT_ACCEPTABLE);

         }
```

If it is a regular call, an event is fired to the Authentication Service for handling this request and start the authentication process or else another type of event is fired targeting the Conference Service (JAIN SLEE Service), in order to fullfill the necessary procedures to start the Group Call, but only after a new Conference token and PINs are obtained from the Collaboration Service.

5.2.1.2. Authentication

The Authentication Service is responsible for managing the authentication process of the user to the conference he wants to join. It is composed of three SBBs, and has as root SBB the Authentication SBB which has 2 child SBBs.

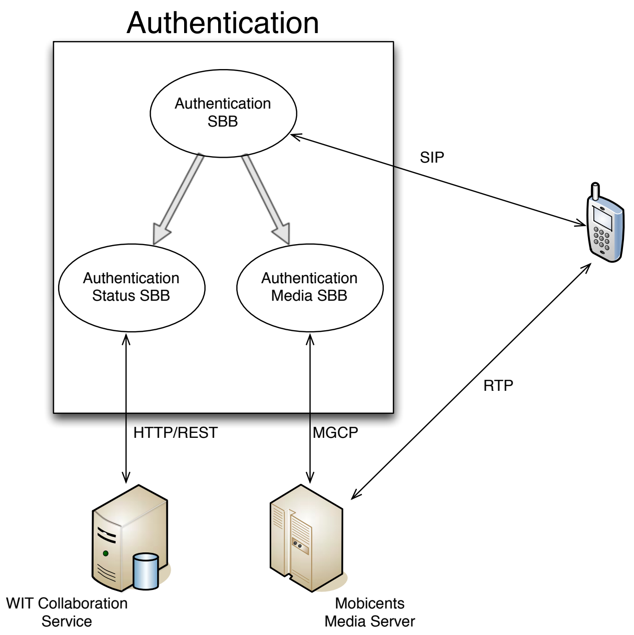


Figure 24 - Authentication Service Illustration

Authentication SBB

This SBB is responsible for handlong all the SIP requests and responses as also sending the necessary ones, as well as managing it's 2 child SBBs. At the creation of each SBB entity of this type, the other 2 child SBBs are automaticly created as seen below.

```
public void sbbPostCreate() throws javax.slee.CreateException {
                (...)
                this.getAuthStatusSbb().create(ChildRelationExt.DEFAULT_CHILD_NAME);
                this.getAuthMediaSbb().create(ChildRelationExt.DEFAULT_CHILD_NAME);
                (...)

        }
```

Depending on the successful match of a PIN entered by the user (with a maximum of 3 attempts) the SBB will fire an event to the Conference Service (JAIN SLEE Service), which can either create or not a new SBB entity (which is explained next). If the user reaches the maximum attempts, the SBB will send a BYE request and remove all the allocated resources for the user (media connections).

Authentication Media SBB

As it is perceived by the name, this SBB handles all the media related actions, from the creation of the IVR endpoint, removal of it, announcements played for the user and DTMF collect.

Authentication Status SBB

The main responsibility of this SBB is to handle the requests and responses from the Collaboration Service, i.e., when a user inputs a PIN, the Authentication Media SBB notifies it's parent (Authentication SBB), which accesses this SBB for sending the PIN to the mentioned service. Upon response, this service parses it, and deliveres the response to it+s parent (which is a conference token of which the PIN is associated if it exists on the Collaboration Service).

5.2.1.3. Conference

This can be considered the main service of the Conference Service, as it is responsible for managing the conferences and the subscriptions also. It was divided into quite several childs per conference, since it allows to easily add more features if it is done this way. The root SBB is the Conference SBB (one SBB entity per unique conference that is related to the conference token), which contains a Conference Status SBB, Conference Media SBB, Notification Service SBB and a variable number of Conference User SBB (depending on the number of participants on the conference at the moment, since it is 1 Conference User SBB per participant).

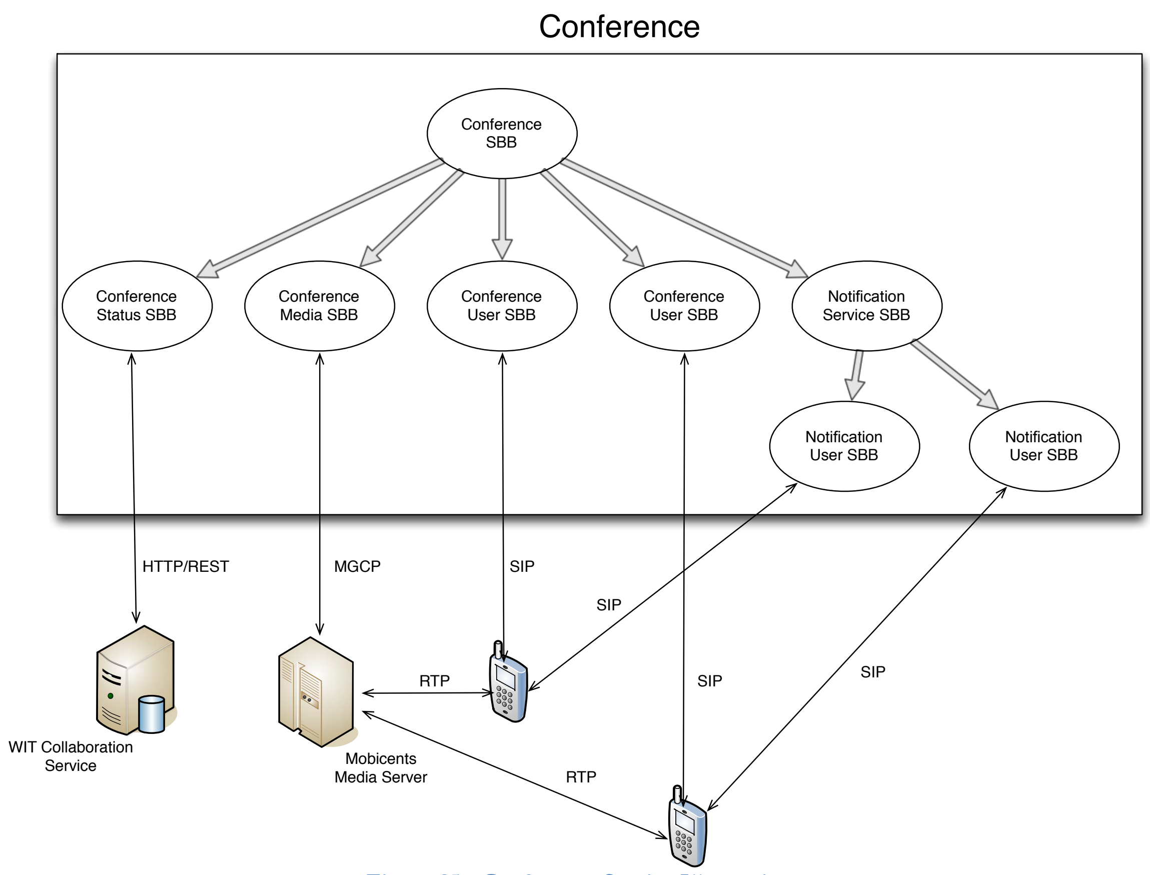


Figure 25 - Conference Service Illustration

The Notification Service SBB contains also child SBB's of it's own, which are related to the subscriptions for the conference event package of the conference it relates to.

Conference SBB

This is where all the conference logic is placed in, i.e., if something happens on any of it's childs, this SBB is responsible for doing the necessary procedures to handle it, by accessing the necessary child SBBs.

For this, it provides a few callback methods by implementing each SBB parent's interface. The child SBBs are all created when this one is created (with the exception of the Conference User SBB).

```
public interface ConferenceSbbLocalObject extends SbbLocalObjectExt, Conference,
ConfStatusParentSbbLocalObject, ConfMediaParentSbbLocalObject, ConferenceUserParentSbbLocalObject,
NotificationServiceParentSbbLocalObject {

                // NOTE: For better consistency, use Conference to define the SBB LocalObject
business methods


        }
```

There is only one Conference SBB per conference, since at the time when receiving a new event, it has an initial event selector, which defines the event's custom name as the conference token (which is contained on the event information). With this it will create a new SBB entity if there is no other created with the same event name, else the event will be fired into the existant SBB and processed as a user who is trying to join a currently active call.

Conference Media SBB

```
public InitialEventSelector onStartGroupCallEventInitialEventSelect(InitialEventSelector ies) {
                // TODO: Implement this method properly.
                // Please see Section 8.6.4 (Initial event selector method) of the
                // JAIN SLEE 1.1 Specification for more information.
                Object event=ies.getEvent();
                if(event instanceof StartGroupCallEvent)
                {
                        StartGroupCallEvent request=(StartGroupCallEvent) event;
                        ies.setCustomName(request.getConference());
                }
                else
                        ies.setInitialEvent(false);

                return ies;

        }
```

Like the Authentication Media SBB, this one is also responsible for handling all the media related actions. In this case, all this actions are from the conference where this is located at.

Conference Status SBB

This SBB is responsible for communicating with the Collaboration Service. In this case it provides information to the collaboration service related to the number of users present in the conference, by adding or removing a user each time anyone enters or leaves the conference.

Conference User SBB

Keeps all the information related to the user in terms of SIP and media data (SDP). It handles all the requests and responses to and from the user exchanged with the conference.

Notification Service SBB

This is the SBB which manages the participants status and the documents which contains this information. Everytime a user enters, leves or is invited, the Conference SBB notifies this one to change a particular users state. After that all the subscribed users (which are managed on this SBB), are updated with the new status of the changed user.

```
@Override
        public void addUser(String username, String fromURI, UserStatus status)
        {
                ArrayList<String> present = this.getPresent();
                if(!present.contains(fromURI))
                {
                        present.add(fromURI);
                        this.setPresent(present);
                }

                User user=new User(username, this.parseUserStatus(status));

                HashMap<String, User> fromVsStatus = this.getFromVsStatus();
                fromVsStatus.put(fromURI, user);
                this.setFromVsStatus(fromVsStatus);

                this.updateDocument();

                this.notifyUsers(status, fromURI);

        }
```

Notification User SBB

Here is ketp the subscription status and the related SIP dialog. It is also responsible for managing the expires timer of the subscription made by the user, and notify it's parent (Notification Service SBB) if the subscription terminates, while also sending a NOTIFY request with Susbcription-State header with terminated value.

When the user refreshes the subscription or starts a new subscription, the Expires header value is used to schedule when the timer shall fire the expired event. This event is cancelled everytime the user refreshes the subscription, and a new one is scheduled.

5.2.2. Administration Website

Due to unexpected problems, the implementation of some of the Conference Service Adminsitration Website functionalities were left to be done. As these were not prioritary compared with other funccionalities, they were left for last.

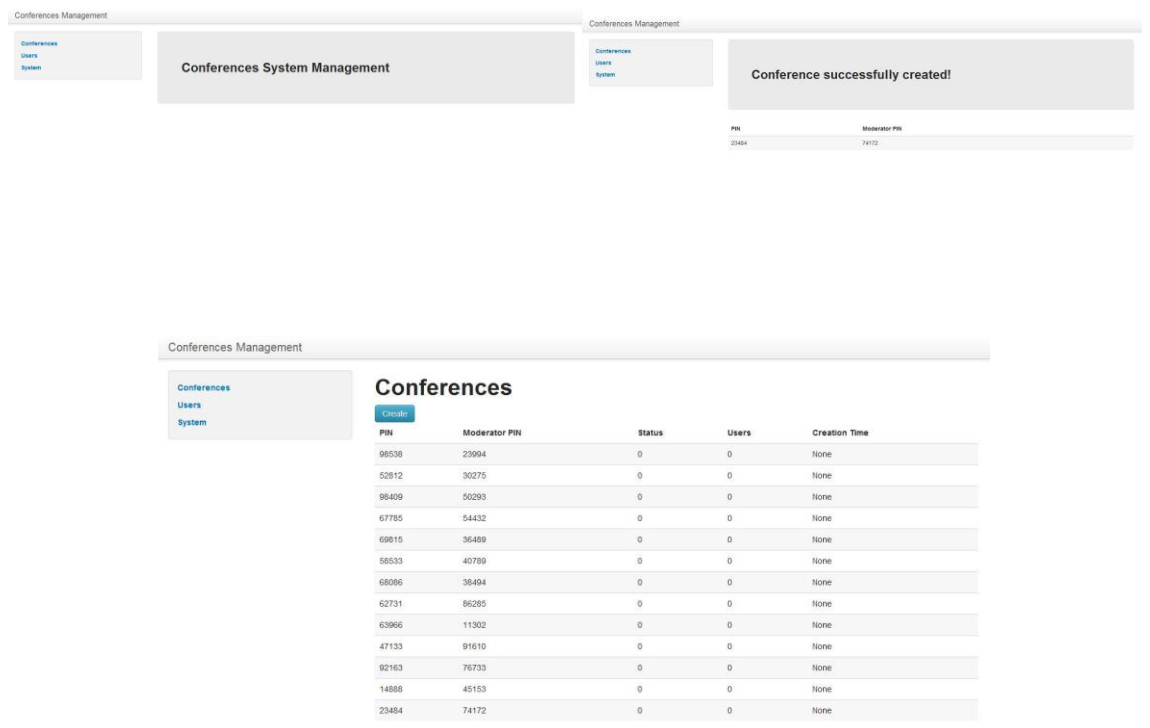
Below can be seen the requirements completed when compared with Table 9.

Table 20 - Conference Service Administration Website High-Level Requirements Completed

Requirement	Summary	Achieved
[CFA1]	Create a conference by generating the moderator and participant PINs	✔
[CFA2]	Present conferences' moderator and participant PINs	✔
[CFA3]	Present conferences' status (active, waiting for moderator or inactive)	✔
[CFA4]	Show conferences' current users number	✔
[CFA5]	Number of Conference Endpoints being used	✖

[CFA6]	Number of IVR Endpoints being used	✖
[CFA7]	Media Server System Load	✖

Below on Figure 26 are presented the layouts developed, which currently allows the administrator to create conferences and verify their status. Worth mentioning that the website is only accessible from the intranet according to the non-functional requirement presented on Table 10.



Conferences System Management

Conference successfully created!

PIN	Moderator PIN
23484	74172

Conferences

PIN	Moderator PIN	Status	Users	Creation Time
98538	23994	0	0	None
52812	30275	0	0	None
98409	50293	0	0	None
67785	54432	0	0	None
69815	36489	0	0	None
58533	40789	0	0	None
68086	38494	0	0	None
62731	86285	0	0	None
63966	11302	0	0	None
47133	91610	0	0	None
92163	76733	0	0	None
14888	45153	0	0	None
23484	74172	0	0	None

Figure 26 - Conference Service Administration Website Final Layouts

The webpage makes use of the Twitter bootstrap framework, which allows developing visually nicer web pages in a very easy way.

5.3. Group Call

The Group Call feature was implemented successfuly except the management features. Some of the features still need some discussion, since some questions were raised due to the privacy of the users while presenting an internal demo on the company.

5.3.1. Android

The Group Call Feature integrated into the joyn application by WIT Software for Android, is separated into 2 Activities, 1 Service and 1 Fragment. There is also one module called GroupCallsManager, which is the responsible for notifying the interested activities on events of a certain conference, or provide the interface for those acitivities to communicate with the COMLib.

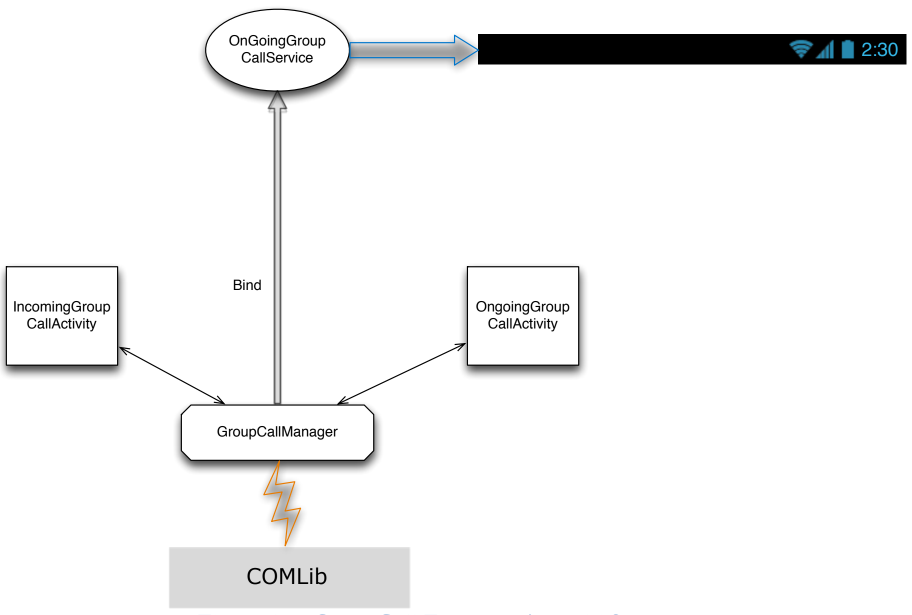


Figure 27 - GroupCall Feature - Android Structure

When an incoming Group Call invitation arrives, it is responsible for initiating a new IncomingGroupCallActivity for handling the new request. As for the GroupCallsOngoingGroupCallService, it binds to the service, in order to allow the activities to place the necessary information and icon on the Android Notification Bar.

When a user accepts or starts a Group Call, the OngoingGroupCallActivity will be started, which receives as parameter the conference URI, that allows to filter the events for only the present Group Call.

The activity is kept on the GroupCallsManager associated with the correspondent conference URI, and when any new event arrives there that is related to that conference URI, the GroupCallsManager notifies the activity.

All the actions made by the user, go through the GroupCallsManager, i.e., either if it is the IncomingGroupCallActivity or the OngoingrGroupCall activity, any action that they will need to do, will go through the GroupCallsManager.

The Group Call UI of an ongoing conference call is separated into the top bar, bottom bar and a space for a fragment to place it's own content. This allows to easily manage if the appearing information is related to the Group Call itself, or for example, presenting a Group Chat which can be associated with this Group Call (future use-case).

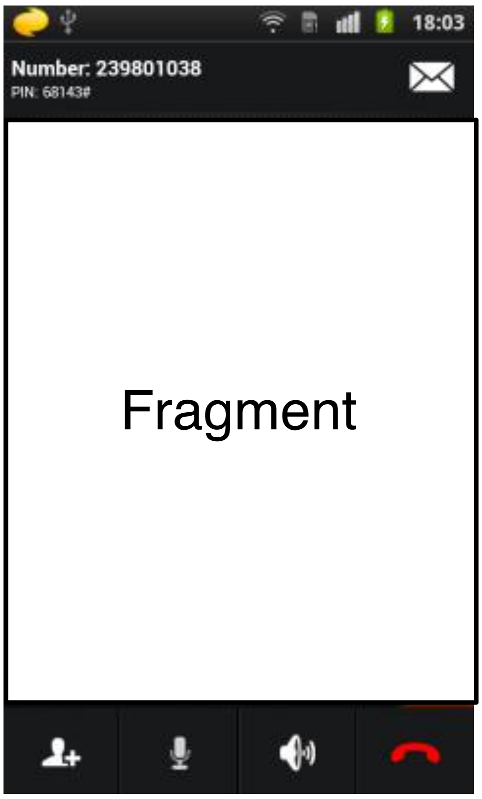


Figure 28 - Ongoing Conference UI Division

The participants screen is a simple adapter which has several GroupCallParticipantEntry, on which tries to load the contact photo if any exists, or else it will return the default contact photo from Android.

Both the SMS and Mail functionalities were already implemented on the joyn application through an API, and it was a simple call to certain methods present on the API in order to implement these functionalities. The contact list selector was also already developed, still the contact list selector for only contacts with email that allowed multi-selection had to be implemented on this internship.

5.3.2. COMLib

The description of this library and the work done in it is a confidential part of this project, which is only available on Annex E that is not available to the general public. This is where the client logic of the joyn application is implemented.

5.3.3. User Experience

In this section a few examples of the UX will be shown and described. The other missing UX related to each one of the use-cases of the application, can be seen on the Annex D section D.2.1.

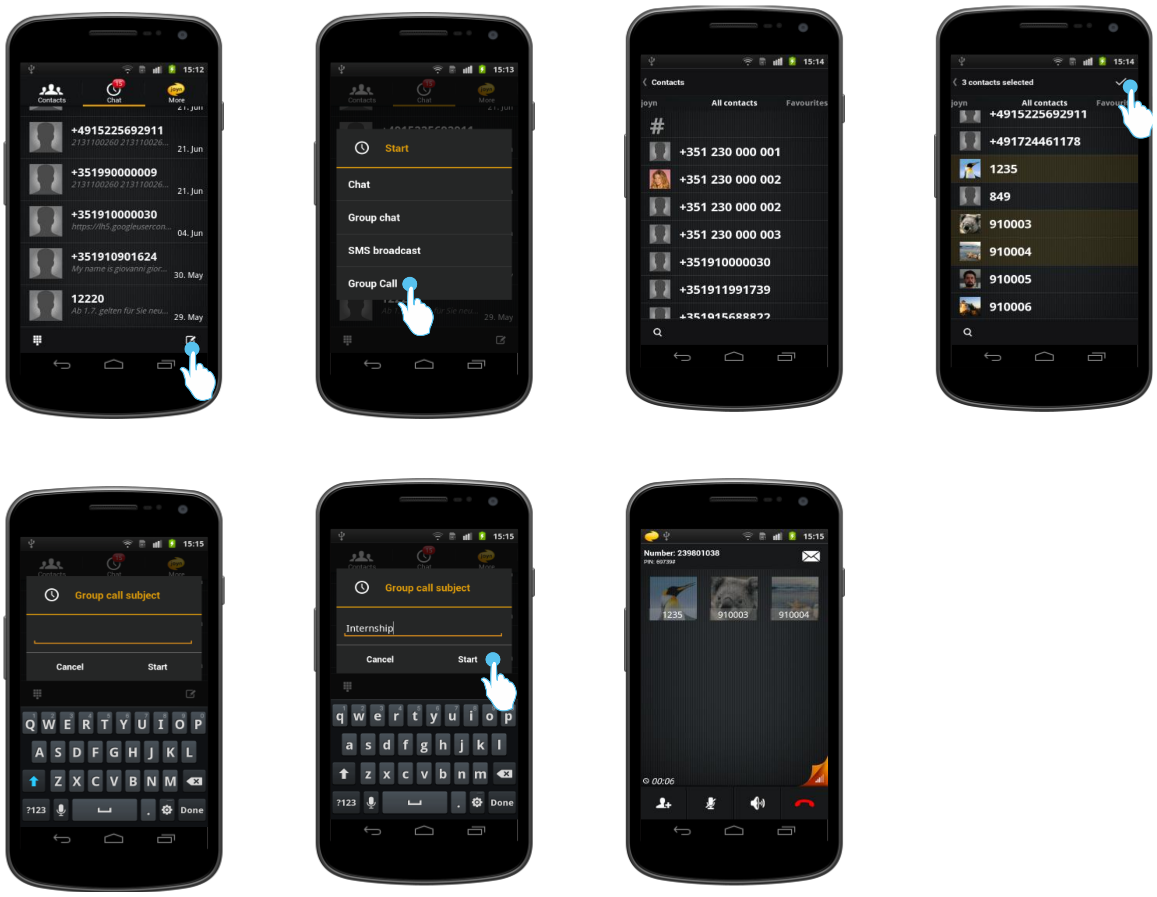
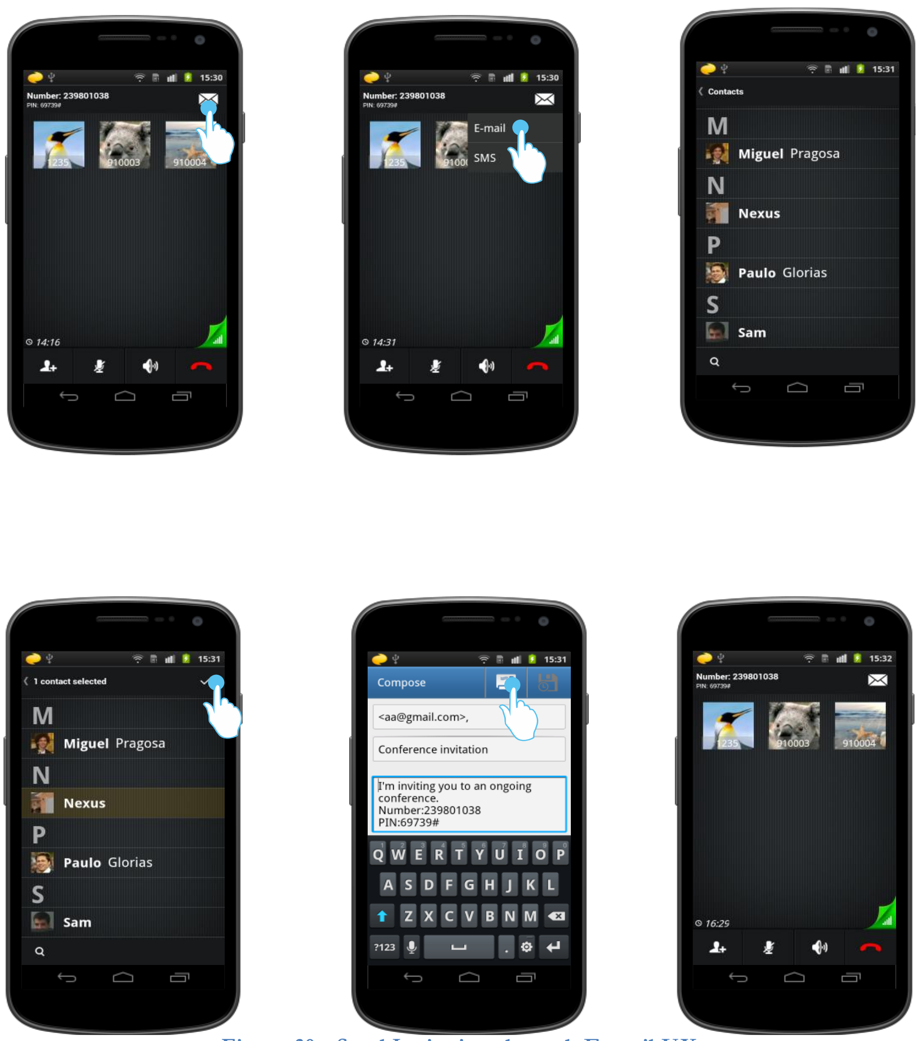


Figure 29 - Start a Group Call UX

On the previous figure is possible to see the user selecting the option to initiate a new Group Call from the history tab. After this the user selects the contacts with who he wishes to start the conference call. When the user ends the previous step, he will be taken into the subject dialog, where he will be able to place a subject for this Group Call. After that the user is in the conference call (if he currently has network).



<aa@gmail.com>,

Conference invitation

I'm inviting you to an ongoing conference.
Number:239801038
PIN:69739#

Figure 30 - Send Invitation through E-mail UX

On this example the user selects the message icon in order to send the conference information to some of his contacts. He then selects to use the email options (here is assumed that he has already setup the email account) and selects from the filtered list of contacts with e-mail available, to which ones he wants to send the invitation with the conference information. After that the e-mail app of his preference is opened, with the destinations, subject and e-mail text already set to go. When the user decides to send or cancel the e-mail, he will get back to the conference call screen.

5.4. Workplan VS Completed

A few changes on the initial workplan were made due to other parallel project which made some of the initial planned features to be abandoned. Also there were some problems while integrating the joyn application with the solution used as IMS network (OpenIMS), which say that support GRUU [RFC5627] (useful for multi-device handling), but instead lead to problems until 2 weeks of debugging of the OpenIMS to conclude this. This support had to be disabled from joyn (which violates the spec, in case of multi-device use from the same user)

Also there was another week which was somewhat troublesome, since the main trunk of the joyn was constantly being changed, and some of the updates brought compatibility problems with the work developed until then. The solution, was to stop mergin the code from the trunk, once a stable version was available.

Since the project will not end with the termination of this internship, a few advancements are already being made on the Conference Mangament functionalities as shown on the table below that compares the workplan for this semester and the work that was completed.

Table 21 - 2nd Semester's Workplan VS Completed Work

Component	Summary	Description	Completed
Administration Website	Implement missing layouts	Create the missing layouts for the Conference Service Administration Website	✖
Administration Website	Integrate the administration website with the Conference AS	Provide information on the media server resources allocated	✖
~~**User Website**~~	~~Implement the users website~~	~~Create the layouts~~	~~✖~~
~~**User Website**~~	~~Integrate the users website with the Conference REST service~~	~~Allow the users to create/schedule conferences and manage ongoing conferences~~	~~✖~~
~~**User Website**~~	~~Integrate the users website with the Conference AS~~	~~Allow the moderator to mute and unmute participants through the webpage~~	~~✖~~
Conference AS	Modify the Conference SBB implementation	Separate the Conference SBB into multiple SBBs	✓
Conference AS	Handle user's webpage actions	Handle the user's webpage requests and perform the actions required	✖
Conference AS	Handle moderator actions	Implement handling of DTMF codes insertion for actions like mute and unmute the moderator and other participants	✖
Conference AS	Handle participant actions	Implement handling of DTMF codes insertion for actions like mute and unmute the participant	✖
~~**Conference AS**~~	~~Verify capabilities~~	~~Implement capabilities handling on the Conference AS~~	~~✖~~
Conference AS	Handle user disconnection	End user session when the	✓

		call is ended without any BYE received (ex: network failure)	
Conference AS	Implement Conference Event Package	When the user hasn't got the Group Call capability, the Conference AS has to notify the users when conference-unaware participants leave the conference	✓
Conference AS	Create and get Conference Event Package id from the presence server	When a new conference call is create, the Conference AS needs to create a Conference Event Package for that conference	✓
Conference AS	Initiate conference call from group chat	Handle INVITE request for Group Call and perform the necessary procedures to initiate conference with the other invited users	✓
Conference AS	Inform participants of any change on one participant's media (muted, camera on, etc.)	Send a NOTIFY request to the RCS users compliant with the Group Call capability, in order to receive updates when someone is muted/unmuted or turns it's camera on/off	✓
Conference AS	Implement notifications to the conference participants on media changes	Implement functionalities described on the RFC 4575[RFC4575], which allow to notify users about media changes on the other participants and their current roles	✓
RCS Client - Logic	Implement Group Call capability	Allow RCS users compliant with the Group Call capability, to receive other users presence status when joining a conference	✓
RCS Client - Logic	Implement moderator's conference functionalities	Allow the conference's moderator to manipulate it, by adding/removing video support and manage other users media (audio, video)	✓
RCS Client - Layout	Implement the call layout	Implement the Group Call User Interface	✓

RCS Client - Layout	Implement the management layout	Implement the Group Call Management User Interface	✔

Chapter 7
Conclusions and Future Work

The area of the telecommunications industry is suffering a big threat, but has been trying to produce responses that are able to overcome these. Part of this response is both the IMS network architecture, which provides a great environment for the development of new applications and services that take advantage of the converged platform that it allows to. The RCS initiative is one of these, that took advantage of the possibilities made available by the IMS network, and provided a new set of features and functionalities that weren't usual to be offered by the Services Providers (Mobile Operators).

As for future work, it would be interesting to try to specify a way to turn a 1-1 voice cal into a Group Call, either if it is circuit switched or over IP.

References

[AND_A] Activity | Android Developers

http://developer.android.com/reference/android/app/Activity.html

28th June 2013

[AND_F] Fragment | Android Developers

http://developer.android.com/reference/android/app/Fragment.html

28th June 2013

[AND_S] Service | Android Developers

http://developer.android.com/reference/android/app/Service.html

28th June 2013

[ANDR] Android (operating system) - Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Android_(operating_system)

28th June 2013

[ANDRS] Android 4.2 Jelly Bean based CM 10.1 is now available for Galaxy S3 i9300

http://www.android.gs/android-4-2-jelly-bean-based-cm-10-1-is-now-available-for-galaxy-s3-i9300/

28th June 2013

[CLIFL1] Calliflower for iPhone, iPod touch and iPad on the iTunes App Store

https://itunes.apple.com/za/app/calliflower/id303569280

28th June 2013

[FCFC1] Free Conferencing for iPhone, iPod touch, and iPad on the iTunes App Store

https://itunes.apple.com/us/app/free-conferencing/id413615935

28th June 2013

[FCFC2] Freeconferencing - Android Applications on Google Play

https://play.google.com/store/apps/details?id=air.com.freeconferencecall.FreeConferencingPhone

28th June 2013

[FCFM1] FreeConference Mobile for iPhone, iPod touch, and iPad on the iTunes App Store
https://itunes.apple.com/us/app/freeconference-mobile/id476365007

[GSMA] GSMA
http://www.gsma.com/
28th December 2012

[IETF] Internet Engineering Task Force (IETF)
http://www.ietf.org/
28th December 2012

[IMS11] 3GPP IMS Specification Release 11
http://www.3gpp.org/Release-11
22nd September 2012

[IMS12] 3GPP IMS Specification Release 12
http://www.3gpp.org/Release-12
28th December 2012

[IPBR] IPBrick HomePage - Portugal
http://www.ipbrick.pt/
28th June 2013

[IPDC] draft-elliott-ipdc-media-00 - IPDC Media Control Protocol IETF Draft
http://tools.ietf.org/html/draft-elliott-ipdc-media-00
23rd November 2012

[JSR22] JAIN™ SLEE API Specification JCP JSR
http://www.jcp.org/en/jsr/detail?id=22
23rd November 2012

[JSR23] JAIN™ MGCP API Specification JCP JSR
http://www.jcp.org/en/jsr/detail?id=23
23rd November 2012

[JSR116] SIP Servlet API JCP JSR
http://www.jcp.org/en/jsr/detail?id=116

23rd November 2012

[JSR240] JAIN™ SLEE (JSLEE) v1.1 JCP JSR
http://www.jcp.org/en/jsr/detail?id=240
23rd November 2012

[JSR289] SIP Servlet v1.1 JCP JSR
http://www.jcp.org/en/jsr/detail?id=289
28th December 2012

[JSR309] Media Server Control JCP JSR
http://www.jcp.org/en/jsr/detail?id=309
5th January 2013

[JSR359] SIP Servlet 2.0 JCP JSR
http://www.jcp.org/en/jsr/detail?id=359
28th December 2012

[MDCFC1] MobileDay Conference Call - Android Applications on Google Play
https://play.google.com/store/apps/details?id=com.mobileday
28th June 2013

[MDCP] Media Device Control Protocol IETF Draft
http://tools.ietf.org/pdf/draft-sijben-megaco-mdcp-00.pdf
21st December 2012

[RCS_e] GSMA, RCS-e Specification
http://www.gsma.com/rcs/wp-content/uploads/2012/03/rcs-e_advanced_comms_specification_v1_2_2_approved.pdf

[RCS5_1] GSMA, RCS 5.1 Specification
http://www.gsma.com/rcs/wp-content/uploads/2012/10/RCS5.1-UNI-V1.0.zip
27th December 2012

[RFC1889] RTP: A Transport Protocol for Real-Time Applications IETF RFC
http://tools.ietf.org/html/rfc1889
12th December 2012

[RFC1890] RTP Profile for Audio and Video Conferences with Minimal Control IETF RFC

http://tools.ietf.org/html/rfc1890

12th December 2012

[RFC2327] SDP: Session Description Protocol IETF RFC

http://tools.ietf.org/html/rfc2327

12th December 2012

[RFC2543] SIP: Session Initiation Protocol IETF RFC

http://tools.ietf.org/html/rfc2543

12th December 2012

[RFC2705] Media Gateway Control Protocol (MGCP) Version 1.0 IETF RFC

http://tools.ietf.org/html/rfc2705

12th December 2012

[RFC2805] Media Gateway Control Protocol Architecture and Requirements IETF RFC

http://tools.ietf.org/html/rfc2805

12th December 2012

[RFC2885] Megaco Protocol Version 0.8 IETF RFC

http://tools.ietf.org/html/rfc2885

12th December 2012

[RFC2886] Megaco Errata IETF RFC

http://tools.ietf.org/html/rfc2886

12th December 2012

[RFC3015] Megaco Protocol 1.0 IETF RFC

http://tools.ietf.org/html/rfc3015

20th December 2012

[RFC3261] SIP: Session Initiation Protocol IETF RFC

http://tools.ietf.org/html/rfc3261

5th September 2012

[RFC3262] Reliability of Provisional Responses in the Session Initiation Protocol (SIP) IETF RFC

http://tools.ietf.org/html/rfc3262

12th December 2012

[RFC3263] Session Initiation Protocol (SIP): Locating SIP Servers IETF RFC

	http://tools.ietf.org/html/rfc3263
	12th December 2012
[RFC3264]	An Offer/Answer Model with the Session Description Protocol (SDP) IETF RFC
	http://tools.ietf.org/html/rfc3264
	19th September 2012
[RFC3265]	Session Initiation Protocol (SIP)-Specific Event Notification IETF RFC
	http://tools.ietf.org/html/rfc3265
	11th December 2012
[RFC3266]	Support for IPv6 in Session Description Protocol (SDP) IETF RFC
	http://tools.ietf.org/html/rfc3266
	12th December 2012
[RFC3435]	Media Gateway Control Protocol (MGCP) Version 1.0 IETF RFC
	http://tools.ietf.org/html/rfc3435
	25th September 2012
[RFC3525]	Gateway Control Protocol Version 1 IETF RFC
	http://tools.ietf.org/html/rfc3525
	12th December 2012
[RFC3550]	RTP: A Transport Protocol for Real-Time Applications IETF RFC
	http://tools.ietf.org/html/rfc3550
	12th December 2012
[RFC3551]	RTP Profile for Audio and Video Conferences with Minimal Control IETF RFC
	http://tools.ietf.org/html/rfc3551
	12th December 2012
[RFC3588]	Diameter Base Protocol IETF RFC
	http://tools.ietf.org/html/rfc3588
	12th December 2012
[RFC3660]	Basic Media Gateway Control Protocol (MGCP) Packages IETF RFC

http://tools.ietf.org/html/rfc3660

3rd Octoboer 2012

[RFC3661] Media Gateway Control Protocol (MGCP) Return Code Usage IETF RFC

http://tools.ietf.org/html/rfc3661

10th Octoboer 2012

[RFC3711] The Secure Real-time Transport Protocol (SRTP) IETF RFC

http://tools.ietf.org/html/rfc3711

12th December 2012

[RFC3840] Indicating User Agent Capabilities in the Session Initiation Protocol (SIP) IETF RFC

http://tools.ietf.org/html/rfc3840

12th December 2012

[RFC4353] A Framework for Conferencing with the Session Initiation Protocol (SIP) IETF RFC

http://tools.ietf.org/html/rfc4353

21st October 2012

[RFC4566] SDP: Session Description Protocol IETF RFC

http://tools.ietf.org/html/rfc4566

7th October 2012

[RFC4575] A Session Initiation Protocol (SIP) Event Package for Conference State IETF RFC

http://tools.ietf.org/html/rfc4575

2nd February 2013

[RFC4579] Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents IETF RFC

http://tools.ietf.org/html/rfc4579

5th November 2012

[RFC4585] Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF) IETF RFC

http://tools.ietf.org/html/rfc4585

12th December 2012

[RFC5125] Reclassification of RFC 3525 to Historic IETF RFC

http://tools.ietf.org/html/rfc5125

12th December 2012

[RFC5367] Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP) IETF RFC

http://tools.ietf.org/html/rfc5367

23rd November 2012

[RFC5506] Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences IETF RFC

http://tools.ietf.org/html/rfc5506

12th December 2012

[RFC5627] Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP) IETF RFC

http://tools.ietf.org/html/rfc5627

27th March 2013

[RFC5719] Updated IANA Considerations for Diameter Command Code Allocations IETF RFC

http://tools.ietf.org/html/rfc5719

10th January 2012

[RFC5727] Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area IETF RFC

http://tools.ietf.org/html/rfc5727

12th December 2012

[RFC5729] Clarifications on the Routing of Diameter Requests Based on the Username and the Realm IETF RFC

http://tools.ietf.org/html/rfc5729

12th December 2012

[RFC5761] Multiplexing RTP Data and Control Packets on a Single Port IETF RFC

http://tools.ietf.org/html/rfc5761

12th December 2012

[RFC6051] Rapid Synchronisation of RTP Flows IETF RFC

http://tools.ietf.org/html/rfc6051

12th December 2012

[RFC6141]	Re-INVITE and Target-Refresh Request Handling in the Session Initiation Protocol (SIP) IETF RFC http://tools.ietf.org/html/rfc6141 20th June 2013
[RFC6222]	Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs) IETF RFC http://tools.ietf.org/html/rfc6222 *12th December 2012*
[RFC6408]	Diameter Straightforward-Naming Authority Pointer (S-NAPTR) Usage IETF RFC http://tools.ietf.org/html/rfc6408 *12th December 2012*
[RFC6446]	Session Initiation Protocol (SIP) Event Notification Extension for Notification Rate Control IETF RFC http://tools.ietf.org/html/rfc6446 *12th December 2012*
[RFC6733]	Diameter Base Protocol IETF RFC http://tools.ietf.org/html/rfc6733 *12th December 2012*
[SGCP]	Simple Gateway Control Protocol (SGCP, version 1.0) http://www.argreenhouse.com/SGCP/sgcp-v1-0.html *18th December 2012*
[SIPP]	Welcome to SIPp http://sipp.sourceforge.net/ *27th June 2013*
[SLEEVSVL]	Other Technology - JAIN SLEE and SIP Servlets http://jainslee.org/othertechnologies/sleevssipservlet.html *2nd March 2013*
[THREAD]	Thread (computing) - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Thread_(computing) *4th January 2013*
[UBR1]	UberConference iOS, Android Apps Make Conference Calls Less Painful | CIO Blogs

http://blogs.cio.com/iphone/17724/uberconference-ios-android-apps-make-conference-calls-less-painful

28th June 2013