2010 Fourth International Conference on Next Generation Mobile Applications, Services and Technologies

An IMS-Based Testbed for Fleet Management Services

Mario Cao, Manuel Alvarez-Campana, Raquel Perez Leal, Joaquín Navarro
ETS Ing. Telecomunicación
Universidad Politécnica de Madrid
Spain
{mcao, mac, rperez, navarro}@dit.upm.es

Eduardo Polo
INDRA
Spain
epoloc@indra.es

Abstract— **The IP Multimedia Subsystem (IMS) is considered the current Service Delivery Platform standard for providing multimedia services in Next Generation Networks (NGN). Although fleet management services are traditionally focused on voice communications, they are evolving towards broadband multimedia ones. For this reason, this paper presents an open IMS-based platform that provides advanced multimedia professional services in the aforementioned scenario. After a detailed analysis of the fleet communication requirements, the IMS-based architecture of the platform is presented. Then, the deployed services are described, from the basic ones to those more advanced, e.g. Push-To-Talk (PTT) and videoconferencing. It stands out the feature of the platform, which provides an environment that facilitates the development, deployment and integration of new services. We finally discuss the importance of the contributions made by the platform for the fleet management scenarios.**

Keywords- Fleet Management, IMS, Next Generation Networks, Push-To-Talk, SIP, Testbed, Videoconferencing.

I. INTRODUCTION

Nowadays, the demand for professional communication systems is constantly increasing. This type of communications can support a wide range of services, depending on the considered specific sector. However, it presents two common features, the evolution towards broadband multimedia services and the convergence of fixed and mobile users. Examples of professional services are communications systems for public safety and emergencies, transportation and traffic, land management agencies and infrastructure maintenance facilities, which provide several services such as electricity, water and gas. Typically, these professional communications integrate various user organizations with different operating features, even independently of each other. This requires the configuration of the system as a fleet communication system, which implies high-level requirements.

Operation of professional communications, in particular in the case of emergencies, has been addressed from many different viewpoints. For example, projects such as MESA [1], PSC Europe [2] and CHORIST [3] address aspects of services and architectures in some depth.

This article presents a demonstration platform for fleet communications services based on IMS architecture (IP Multimedia Subsystem) [4]-[7]. The IMS provides several benefits for this type of scenarios, such as the implicit support of multimedia services, the access technology independence and the flexibility for creating new applications. In addition, the platform fits many of the specific requirements of fleet management systems, such as the support of group communications, half-duplex operation, and the system operator role, which is endowed with privileges for supervising and monitoring users.

The incorporation of IMS advanced converged services is not usually available in fleet management platforms. This testbed provides presence services, XDMS server, and traceability functions of the scenario. Among other services, Push-To-Talk over Cellular, video conferencing and integration of external services into the platform are supported.

Other main features of the platform are the use of open source code and its modularity. Furthermore, easy replication and deployment of new services is available with the use of virtualization techniques.

The following section defines the application scenario. It analyzes the main requirements for fleet communication systems, the services provided by the platform, the capacity of IMS to meet these requirements and the benefits it contributes with. In section III, it is detailed the application of the IMS-based architecture to the scenario and its peculiarities. In sections IV and V, this article deals respectively with the deployment of basic and advanced services. Finally, section VI summarizes the main conclusions of the paper.

II. SCENARIO

A. *Fleet Management Communication Systems*

The main users of fleet professional systems use communication networks, mobile in general, in a different way than users of general purpose communications systems. This difference has caused that each technology generation has specific systems which satisfy the requirements of each type of user. Such systems should offer a wide range of services that can be classified according to several criteria, as follows:

- Depending on the content of the information to communicate, services should provide: voice, data, video and multimedia.
- According to participants in the communications: individual and group calls.

978-0-7695-4121-1/10 $26.00 © 2010 IEEE
DOI 10.1109/NGMAST.2010.24

- Depending on the associated transport options: unicast, multi-unicast or multicast and broadcast services.
- Given the possible simultaneity of the upstream and downstream traffic: unidirectional, duplex or half duplex services.

All this services can be demanded with different QoS requirements and communication models (one-to-one, one-to-many, one-to-all).

Additionally, the role of the fleet manager is a key feature of this type of communications. They should be able to control and monitor communications of users under their responsibility. This requires monitoring and control services of several types, e.g. presence, group and privilege management, real-time location, calls-listening, emergency communications, etc.

Given this diversity of applications and scenarios, it is appropriate to raise an approach of incorporation of flexible services. It should facilitate the development and adaptation of applications according to the needs of the different sectors of professional users.

Following the above mentioned criteria, the testbed described in this article incorporates representative services for fleet management systems. The platform contributes with the combination of traditional benefits of such systems and the new broadband technologies.

B. *IMS Applicability and Contributions*

As a first step in defining the architecture of the platform, it is desirable to study the approaches to develop and deploy services that are being defined by relevant standardization organizations. Thus in the case of IETF, SIP (Session Initiation Protocol) [5] is used as the signaling protocol for the provision of services. More specifically, within the specifications of UMTS Release 5, 3GPP conceived the IMS (IP Multimedia Subsystem) [4] platform in order to support IP multimedia services based on SIP.

Later on, in parallel with the evolution of SIP, the IMS has been extended in Releases 6 and 7 with the support of new applications (instant messaging, presence, push-to-talk, etc.). These new specifications are aligned with the 3GPP2 (CDMA2000) and also with the architecture of NGN (Next Generation Networks) developed by ETSI / TISPAN, [6]. The IMS concept has transcended to other communications systems, mobile (CDMA2000) and fixed (DSL, cable), becoming one of the main paradigms for supporting advanced services in next generation networks.

Fleet communication systems have specific requirements, widely accepted, with a variety of services and scenarios depending on each professional user. Therefore, it is necessary to use a flexible environment that allows easy deployment of services and a proper evolution of the applications, as mentioned above. The use of an IMS-based architecture is the logical evolution in the scenarios where SIP (not IMS) is used as the signaling protocol. An IMS-based architecture introduces some advantages, such as, QoS (Quality of System) assurance, security, accounting, etc. Another important thing worth mentioning is the adaptability to the introduction of broadband technologies, novel in the fleet professional systems, which traditionally are only focused on voice communications. From the point of view of the interoperability requirements, the first two factors are especially interesting. For all the foregoing reasons, the platform, presented in this article, is based on the IMS architecture.

III. IMS-BASED ARCHITECTURE

Fig. 1 shows the architecture of the platform. At the top of the figure there are two RTP mixers and the application servers, which are specialized in the provision of services. The central part of the system is the IMS Core, providing essential user management and control functions. The access layer is placed at the bottom. Being the IMS access agnostic, a number of different technologies can be used in this layer. For fleet management scenarios, these will be usually wireless (e.g, WiFi, WiMAX, UMTS, HSPA, LTE…), although fixed technologies can be also used for an operation center.

A. *General Criteria*

In the proposed platform it is important that the components have open source code and that they could be freely distributed. This will reduce the costs associated to the platform deployment and, at the same time, it will provide flexibility in case some software changes are required. However, it should be taken into account the possible service and maintenance issues generated by open-source solutions, especially for the critical applications considered in the paper.

In addition, special attention is paid in the modularity of the design and the ease of deployment. Through virtualization applications, such as VMWare [8], it is possible to replicate components and deploy them without hardly any effort. These features can be essential in fleet management scenarios that require an almost immediate deployment, e.g. in situations where emergency groups are required. In the case of this testbed, all elements are distributed in different servers depending on their nature.

Finally, a key point of the platform is its ease of creation and deployment of new services. On one hand, the

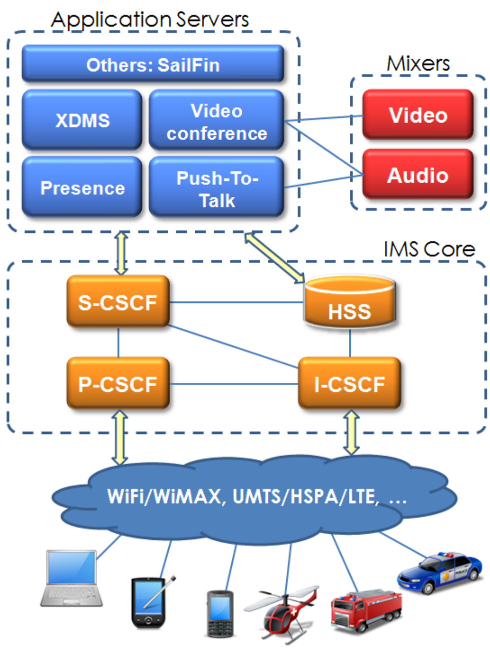

Figure 1. Overview of the IMS-Based platform for fleet management services.

design of the system proposed by 3GPP IMS provides great flexibility to associate new application servers that provide services to the scenario. On the other hand, other components, which are presented in following sections, were added to facilitate the creation and deployment of these application servers.

B. IMS Core

The IMS can be viewed as an infrastructure designed to ease the support of multimedia services over IP networks. The core consists of a series of elements that communicate with each other through various protocols, essentially from the IETF and OMA standards.

The main functional elements are fundamentally the CSCFs (Call Session Control Function). The CSCFs are SIP servers that provide registration, location and routing functions. There are three different types of CSCFs: Proxy-CSCF, Serving-CSCF and Interrogating-CSCF.

Another important element of the architecture is the HSS (Home Subscriber Server). It contains the main database with information of users and network services.

Finally, another element outside the IMS core that stands out is the MRF (Media Resource Function). It supports auxiliary functions in multimedia communications, such as the support of video conferences between multiple users.

For the deployment of IMS core, several non-commercial solutions were analyzed. From all of them, The Fokus OpenIMS Project [9] was the solution which better fitted in this testbed. For the platform, an open solution with all its elements clearly differentiated is required. Through this project under GPL v2 license, the source code of all components is available. This allows easy modification and better adaptation to the scenario of fleet management.

Other advanced components of the platform, designed to fill some OpenIMS lacks, are described in following sections of this document.

IV. BASIC SERVICES

This section briefly describes those basic services of the platform. Some are available and installable through the application manager of the operating system. Others are supported by specific open source applications that have been deployed. Finally, a traceability application developed for this platform is presented.

Besides the services described below, the platform also includes other basic services, such as time synchronization, domain name service, etc.

A. Presence and XDMS

Presence and group management services are essential in fleet management systems. The first service allows storing the presence information and the context of users. It also distributes the presence information to users who are subscribed. The group management service stores the lists and characteristics of the different user groups of the system.

The previously described solution for the core IMS does not provide presence services or XDMS (XML Document Management Server). For this reason, other open-source solutions were analyzed in order to deploy these services.

The adopted software solution is OpenSIPS [10]. Other alternatives, such as the Kamailio project [11] were also analyzed. Despite their great similarity, OpenSIPS was chosen because of its better integration with the used XDMS.

The XDMS stores the definitions of the various groups that exist in the system. This information is usually saved as XML documents. It also may store additional information about the members, such as attributes that characterize more precisely a certain group. The solution deployed is OpenXCAP [12], an open source XCAP server that meets the requirements of the scenario and fulfills the IETF and OMA specifications.

B. Scenario Traceability

The scenario traceability service is a tool for the supervision and system control. It allows the operator to monitor the scenario. The absence of standards in this regard hinders the creation of a common log event that supports traceability. For this reason, the logging events of the application are recorded into files.

Based on the above approach, a Java application, which fulfills this functionality, has been developed. It performs three major actions: the SSH connection to the virtual machines that contains the log files of the modules, the download and interpretation of log files, and the presentation of information in a user-friendly graphical interface.

V. ADVANCED SERVICES

This section presents the advanced services supported in the platform. It is important to emphasize the complexity and practically non-existence of free distribution solutions that support the below described services.

Firstly, the components used for the multimedia support are presented. Secondly, the advanced services deployed on the platform are detailed.

A. Multimedia Support

In an IMS scenario where different clients have to interact, the solutions must meet the following requirements:

- Manage RTP (Real-Time Transport Protocol) streams.
- Mix real-time multimedia flows.
- Support for more used codecs in IMS scenarios.

The solution presented above, OpenIMS, is not provided with a MRF. With the intention of minimizing its deficiency in the architecture, two RTP mixers are deployed; one for audio and another one for video. These media mixers are invoked directly by the application servers through specific mechanisms. Although they do not fulfill MRF functions, they enable the use of multimedia services in fleet communications scenarios.

For audio mixing, jVoiceBridge has been deployed [13]. The software, developed in Java by Sun Microsystems Laboratories, is licensed under the GPL v2. This mixer meets the above requirements and since it is an open source module, it is possible to make small modifications in order to achieve a full operative integration with the IMS core.

Video mixing is even more complex than audio mixing, because it is necessary to handle larger amounts of information. Currently, the open source solutions available are very rare. The ffmpeg project [14] is clearly highlighted as the most used solution. It also stands out the VideoLAN project [15], which is world-wide known by its multimedia platform VLC. This application can be used as a server and it also provides the plug-in Mosaic, which has the ability to create mosaics, where different video inputs are shown. This functionality is used by the videoconferencing service later described.

Both mixers have well-defined interfaces which can be used to support different services on the platform. This enables the possibility of deploying new multimedia services in the future.

B. *Push-To-Talk*

The Push-To-Talk (PTT) service allows bidirectional voice communication in half-duplex between two or more participants. The manner of use of this service is very similar to radio technologies, whose common scenario is the diffusion of voice among a group of users.

The PTT server is inspired in the OMA PoC (Push-to-talk over Celullar) [16]. It implements the necessary logic for the PoC session control and audio mixing capabilities.

After a wide analysis, it can be concluded that all PoC solutions were commercial. There were no operational open-source solutions available. For this reason, a prototype was developed as a proof of concept. It is based on Java technology for portability reasons and on the audio mixer jVoiceBridge above presented.

For the design of this service the OMA PoC specifications v1.0.4 has been taken as the starting point. The functionality of the PoC server and client can be divided into two different planes: control and media.

On one hand, the server implements the following functionalities:

- **Control**: PoC session management, user management associated to PoC sessions and speech permission management.
- **Media**: users do not need to use the same audio codec, so the PoC server should be able to transcode the audio streams when necessary.

On the other hand, the client implements:

- **Control**: connection with PoC server, retrieval of PoC session status, and speech permission request.
- **Media**: decoding and playback of audio received as RTP stream, and delivery and coding of the captured audio.

For the control plane of the server and client, the TBCP (Talk Burst Control Protocol) described in the OMA specification has been emulated. It satisfies the client and server functionalities previously mentioned.

For the complex task of transcoding audio streams in real time, the server uses the audio mixer jVoiceBridge. Although it is not necessary to mix different audio sources, this mixer offers all the features needed: conferences management, RTP streams management and transcoding. In order to provide the PoC service using this solution, a specific interface has been developed. It silences the users according to the speech permissions they have.

A major advantage of using this mixer is that the PoC client can use conventional SIP clients to fulfill the functions of the media plane. It also reduces the development complexity of the PoC client. An important feature of this design is that the SIP client is not aware of the existence of the PoC client and its consequent session management.

In Fig. 2, a scheme of the PoC service components is shown. There, it can be observed the distinction between client and server, and the communication between them. It also presents a sample sequence of steps required to establish a PoC session. In the control plane, the TBCP is used for the establishment and management of the PoC session (step 1). Then, the server connects to the audio mixer (step 2) in order to call the user using SIP protocol (step 3). Finally the audio is exchanged through RTP streams (step 4).

Fig. 3 presents a screenshot of the PoC client user interface. In this example the user has already established the session. It shows the current status of the session, the speech permission status and the last message, sent or received. In this case, the state is connected, the speech permission is idle, which means that no one has asked for permission and the last message is related to the connection status.

The development of the PoC client user interface using Java makes the application platform-independent. The design of this solution also increases the interoperability between different SIP clients.

C. *Videoconferencing*

The videoconferencing service is defined as a conversation between more than two users on which audio and video streams are exchanged. The component associated with this service is the MRF, which is specialized in coding, decoding and mixing multimedia streams. Due to the complex tasks it performs, the computational load of this element is usually extremely high.

We have studied various software solutions. The conclusion of this study is that there are multiple

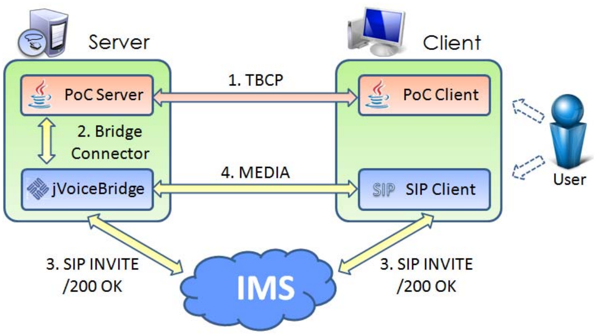

Figure 2. PoC Session establishment.

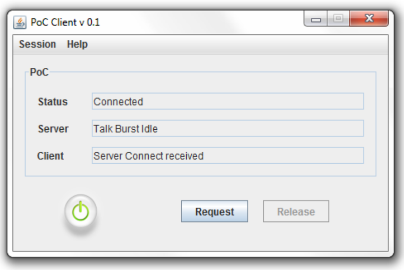

Figure 3. Screenshot of the PoC Client graphical user interface.

alternatives, but the vast majority of them are commercial. There are some free alternatives, but they do not meet the platform requirements. For example, VMukti [17] supports videoconferencing, but it is based on peer-to-peer technologies and web user interfaces.

Other projects, such as Confiance [18] or MEDIACTRL [19], provide prototypes that could meet the requirements needed to support this service. However, the first project has been several years inactive and the second one has only audio mixing capabilities, but no video.

For these reasons, a prototype was developed in Java in order to support the videoconference service in the testbed. The design is based on a client-server model and on the reusability of other free software components. This proposed solution does not follow the IMS videoconferencing standards (TS 24.147), but could be used as starting point for future developments that fulfill the mentioned standard.

For the design of the videoconference server and client, two levels are considered:

- **Control**: development of a simple conferencing protocol for the communication between clients and server.
- **Media**: use of mixers and multiplexers for audio and video streams received by users.

For the media mixing a modular approach was applied, where the audio and video streams are processed separately.

Firstly, the requirements to support the audio in the service are similar to those previously seen for the PoC service. The main difference in this case, is that audio should be mixed into a single stream in order to be able to be played on conventional SIP clients. For this reason, the aforementioned jVoiceBridge is used. This element is responsible for relaying RTP streams and transcoding audio streams as needed.

Secondly, the requirements to support video are:

- **Video Codec**: a video codec, preferably a known standard, should be able to support the service.
- **RTP**: used for receiving and sending multimedia streams. This is the standard protocol used in IMS scenarios.
- **Presentation**: users who are part of the videoconference should be capable of displaying correctly all the video streams.

In order to fulfill the first two requirements, the aforementioned ffmpeg project is used. These libraries support several codecs and are also able to handle RTP streams. The codec chosen for the service is the H.264, one of the most used codecs for video conferencing applications.

However, for the proper provision of the service, additional software is required. It should allow the mixing of the different video streams which are sent from all the users of the conference. For this task, the plug-in Mosaic is used. It assembles all the video inputs as a single RTP stream.

Finally a streaming mechanism is required in order to provide the mixed stream to the client. This function is covered by the application VLC itself, since it supports video streaming using RTSP (Real Time Streaming Protocol). This protocol has as its main functions the establishment and control of media sessions.

Fig. 4 shows a sample sequence of how a videoconference session is established. It can be observed the different components of the service architecture. The complexity of this service can be appreciated by the number of elements deployed as by the operational load required by the server. First, the conference protocol is used for the creation and management of the conference (step 1). Then, the server initiates simultaneously two different action sequences for each type of data, audio and video. The audio sequence is analogous to the sequence described in Fig. 2 (steps 3a, 4a). In order to support video, the client starts a VLC client, which capture video. In the same time, the server starts the VLC Server, which mix the incoming video streams (step 3b). Then, the video is exchanged as RTP streams (step 4b) and is displayed by a VLC client, which uses RTSP (step 5b, 6b).

The audio communication between clients follows the same diagram as in the previously described for the PoC service. The only peculiarity in the audio communication establishment between clients is that the capture of users audio is mixed and distributed to all other users. Thus, each user will receive a single RTP stream that contains the sound of the captures from the rest of conference participants (not the own capture).

In Fig. 5, a sample of the mosaic of a user who is using the service is shown. It can be seen that in the videoconference there are four participants.

D. Integration of external services into the platform

This section presents a procedure to integrate external services into the platform.

The platform design makes possible the integration of new services with different signaling systems to the signaling protocol SIP, used in this platform. The key features in which this integration is based are:

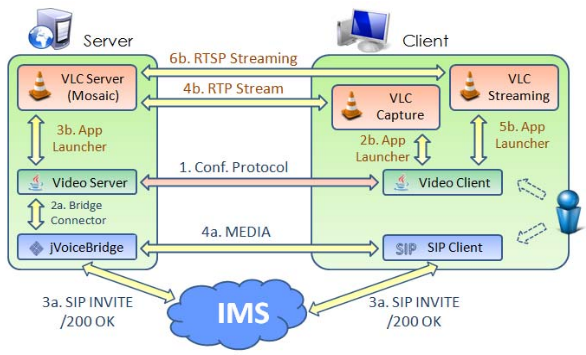

Figure 4. Videoconferencing session establishment.

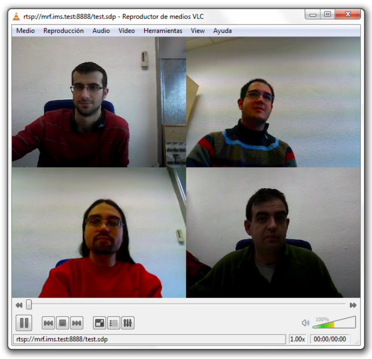

Figure 5. Screenshot of mosaic displayed on a client of a videoconference of four participants.

- Full abstraction of the different signaling systems.
- Central element that captures the events of a system and generates the appropriate response in the IMS signaling through some existing function in the platform.

In Fig. 6 the integration method of new external services can be seen. On one side is the platform with SIP as signaling protocol and, on the other side, an external system which can have different types of connected inputs.

The solution proposed for the central element is based on SailFin [20], an open source Sip Servlets Container which allows easy implementation and deployment of services.

This approach preserves system scalability and modular design of this testbed.

A good example to illustrate this procedure of integration is the capture of surveillance alarm events. On one hand, there is the platform described in this article; on the other hand, an external surveillance system with proprietary signaling. This system contains a network of devices: cameras, sensors, etc. Any event generated by these devices is forwarded to the central element, the application server. It analyzes the captured event and generates a specific response, e.g. a notification addressed to the system administrator via SIP MESSAGE.

Other examples that fit perfectly with the proposed scheme might be: calendars, which generate events based on scheduled events; global messages, which are sent to all users of the platform and contains critical information received by external systems; and subscriptions to RSS Feed.

VI. Conclusion

This paper presents an implementation of a platform for the provision of fleet management services based on IMS and free software. The flexibility for deploying new services and the access agnostic nature of the platform fulfill the requirements of fleet management professional systems.

In addition to the IMS core elements, the platform provides several services which are representative of fleet management communication systems, such as voice/video calls, instant messaging, push-to-talk, videoconferencing, etc.

It should be emphasized the minimal economical costs associated with the deployment of the platform in a real scenario. Furthermore the effort associated with the deployment of the platform is minimal due to the use of virtualization techniques. In addition, it also highlights the development environment provided by the platform, which facilitates the integration of new services into it. Therefore, this testbed can be perfectly used as a basis for future deployments of real systems.

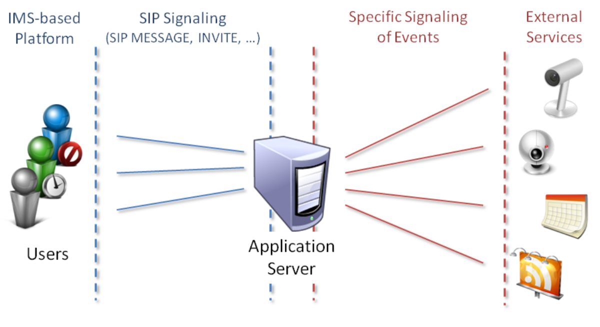

Figure 6. Diagram for the integration procedure of external services.

To conclude, this work opens up numerous lines of future work, like the integration of new services, its performance in real scenarios or the development of a complete MRF solution based on the multimedia support given in this platform.

Acknowledgment

This work has been developed within the scope of the TelMAX Project (www.proyectotelmax.es) (CENIT Nº 20071036) and is partially funded by CDTI -Centro para el Desarrollo Tecnológico e Industrial-, of the Spanish Ministry of Science and Innovation, under the INGENIO 2010 Program.

References

[1] MESA Project, “Service Specification Group - Services and Applications; Statement of Requirements”, MESA TS V 3.1.3(2005-04) Draft, April 2005.

[2] PSC Europe Project, “Forum for Public Safety Communication Europe, PSC Europe Website”. [Online]. Available: http://www.publicsafetycommunication.eu/. [Accessed: January 2010].

[3] CHORIST Project, “Report on Telecom services definition” (v-1.0), D SP4.D2, April 2007.

[4] 3GPP, “IP Multimedia Subsystem (IMS); Stage 2”, TS 23.228, April 2004.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: session initiation protocol”, IETF RFC 3261, June 2002.

[6] TISPAN specifications. [Online]. Available: see www.etsi.org/tispan. [Accessed: January 2010].

[7] G. Camarillo, M.-A. Garcia-Martin. “The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds”. John Wiley & Sons. August 2004.

[8] VMWare. [Online]. Available: http://www.vmware.com/. [Accessed: January 2010].

[9] T. Magedanz, D. Witaszek, and K. Knuettel, “The IMS Play Ground @ FOKUS - an Open Testbed for Next Generation Network Multimedia Services”, TridentCom, Trento, Italy, February 2005.

[10] OpenSIPS Project. [Online]. Available: http://www.opensips.org/. [Accessed: January 2010].

[11] Kamailio Project. [Online]. Available: http://www.kamailio.org/. [Accessed: January 2010].

[12] OpenXCAP Project. [Online]. Available: http://openxcap.org/. [Accessed: January 2010].

[13] jVoiceBridge Project. [Online]. Available: https://jvoicebridge.dev.java.net/. [Accessed: January 2010].

[14] FFmpeg Project. [Online]. Available: http://ffmpeg.org/. [Accessed: January 2010].

[15] VideoLAN Project. [Online]. Available: http://www.videolan.org/. [Accessed: January 2010].

[16] OMA Push to talk over Cellular V1.0.4, Release Date: 2009. [Online]. Available: http://www.openmobilealliance.org/. [Accessed: January 2010].

[17] Vmukti Project. [Online]. Available: http://sourceforge.net/projects/vmukti/. [Accessed: January 2010].

[18] Confiance Project. [Online]. Available: http://confiance.sourceforge.net/. [Accessed: January 2010].

[19] MediaCTRL Project. [Online]. Available: http://mediactrl.sourceforge.net/. [Accessed: January 2010].

[20] SailFin Project. [Online]. Available: https://sailfin.dev.java.net/. [Accessed: January 2010].