THE VIDEO-COMMUNICATION SYSTEM

Yury Sergeev
Bachelor's thesis
Autumn 2010
Degree Programme in Information Technology
Oulu University of Applied Sciences

PREFACE

This Bachelor's thesis was commissioned by Pehr Brahe Center for Industrial and Services ICT (PBOL ICT), Raahe, Finland. PBOL is a cooperative laboratory specialized in the research and development of information technology and wireless communications. PBOL's background organizations are Oulu University of Applied Sciences, University of Oulu, VTT Technical Research Centre of Finland and the City of Raahe.

ACKNOWLEDGEMENTS

I would like to thank PBOL laboratory for an opportunity to make my Bachelor's thesis, and for an interesting topic of my thesis project. Also, I would like to say thank you to my tutoring teacher - Markku Korhonen (for excellent management), Vadym Kramar - a technical supervisor of my work (for his advice in designing and implementation), my colleagues - Jori Karpinen and Victor Pervunin (for their willingness to help me with any project related issues), and finally, I want to thank all the staff of Pehr Brahe ICT for their friendliness.

ABSTRACT

Oulu University of Applied Sciences
”Degree Programme in Information Technology”,”Mobile Technology”

Author(s): Yury Sergeev
Title of Bachelor’s thesis: The Video-Communication System
Supervisor(s): Markku Korhonen
Term and year of completion: 2010 Number of pages: 58 + appendices 6

The aim of this Bachelor’s thesis was to develop a software product for the visual over-IP communication. The communication occurs between the representatives of a house-holding company or another service company and building inhabitants. The communication system must support video and audio streaming.

The software design model was made by using the UML language. It helped to understand the structure of the system**.** The client-server model was selected to represent the core relationships between the consistent parts of the system. The system consists of one server located in a company’s premises, and multiple clients in inhabitant’s apartments. The system’s code was implemented by using a Java programming language - its core advantage being a platform independence. The information about customers, such as a home address and a name, as well as the technical information about devices (IP and MAC addresses) in client’s apartment is stored into a database.

The client node of the system may be represented by several devices. They can be an STB with a TV screen, a touch screen panel or in the future - even a mobile phone. Video capturing can also be done by different devices. If the device has a built-in camera**,** it can be used for capturing. Another option is to use an IP or a USB camera.

The implemented system can be utilized in various applications. It can be used for end-to-end communication purposes or it can act as an advertisement service, which broadcasts video content to end users. The system accepts video input from numerous sources, including a surveillance camera, a local video file or data from another video stream. The system supports several video formats. The server takes responsibility for encoding the video into a format understood by the client.

Keywords: set-top box, video, broadcast, IP camera, JVLC, VoIP

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
”Tietotekniikan koulutusohjelma”, ”Mobiiliteknologia”

Tekijä(t): Yury Sergeev
Opinnäytetyön nimi: Videoviestintäjärjestelmä
Työn ohjaaja(t): Markku Korhonen
Työn valmistumislukukausi ja -vuosi: 2010 Sivumäärä: 58 + liitteet 6

Opinnäytetyön tavoitteena oli kehittää ohjelmistotuote IP-pohjaiseen viestintään. Viestintä tapahtuu asuintalon omistusyhtiön tai muun palveluyrityksen edustajien ja rakennuksen asukkaiden välillä. Viestintäjärjestelmän on tuettava videon ja äänen suoratoistoa.

UML-kuvauskieltä hyödynnettiin järjestelmän ohjelmistosuunnitelman mallintamisessa. Tämä auttoi ymmärtämään järjestelmän rakennetta. Järjestelmän osien välistä riippuvuutta kuvaamaan valittiin asiakas-palvelin-malli. Kokonaisuudessaan järjestelmä koostuu yhdestä yrityksen tiloissa sijaitsevasta palvelimesta sekä useista asuinhuoneistojen asiakaslaitteista. Ohjelmakoodi toteutettiin Java-ohjelmointikielellä, jonka keskeisenä etuna on alustariippumattomuus. Asiakastiedot, kuten kotiosoite ja nimi, tallennettiin asuinhuoneiston teknisten laitetietojen (IP- ja Mac-osoitteet) ohella tietokantaan.

Asiakkaan puolella järjestelmää on mahdollista käyttää useilla eri laitteilla. Näitä voivat olla television yhteydessä tietoverkkoa käyttävä digisovitin, kosketusnäyttöpaneeli tai tulevaisuudessa jopa matkapuhelin. Videokaappaus voidaan toteuttaa myös eri laitteilla. Jos laitteessa on kamera, sitä voidaan käyttää videokaappaukseen. Vaihtoehtoisesti on mahdollista käyttää IP- tai USB-kameraa.

Työn tuloksena toteutettua järjestelmää voidaan käyttää lukuisissa eri sovellustarkoituksissa, kuten esimerkiksi käyttäjien välisessä viestinnässä tai videosisältöä loppukäyttäjälle lähettävässä mainospalvelussa. Järjestelmä hyväksyy videotuloiksi eri lähteitä: valvontakamera, paikallinen videotiedosto tai data toisesta suoratoistovideosta. Järjestelmä tukee lisäksi useita videoformaatteja. Palvelin muuntaa lähdevideon asiakaslaitteen ymmärtämään formaattiin.

Asiasanat: digisovitin, video, lähetys, IP-kamera, JVLC, VoIP

TABLE OF CONTENTS

SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
AVC	Advanced Video Coding
AVI	Audio Video Interleave
CD	Compact Disc
DBMS	Database Management System
DVB	Digital Video Broadcasting
DVD	Digital Video Disc
EPG	Electronic Program Guide
FTP	File Transfer Protocol
HD	High Definition
HTTP	Hypertext Transfer Protocol
IPTV	Internet Protocol Television
Java SE	Java Standard Edition
JMF	Java Media Framework
JNA	Java Native Access
JPEG	Joint Photographic Experts Group
JVM	Java Virtual Machine
MAC	Media Access Control
MJPEG	Motion JPEG
MRL	Multimedia Resource Locator
PC	Personal Computer
PSTN	Public Switched Telephone Network
RTCP	Real-Time Transport Control Protocol
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMS	Short Message Service
SQL	Structured Query Language
STB	Set-Top Box
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VCD	Video CD
VLC	VideoLAN
VLM	VideoLAN Manager
VOD	Video On Demand
VoIP	Voice over IP

1. INTRODUCTION

One of the human needs is the need for communication. The digital age has brought into our lives a lot of new things, including new ways of communicating by using social networks, SMS, instant message, mails, VoIP and videoconference. They all allow people to communicate from a distance. This reduces the time and always stays online. High bandwidth broadband connections have become available to everyone in developed countries. They allow the transfer of not only textual information but also a video in real-time.

Two of the well-known categories of over-IP communication are VoIP and videoconference.

Voice over Internet Protocol (VoIP) is a family of transmission technologies for delivering voice communications over IP networks such as the Internet or other packet-switched networks. Voice-over-IP technology has a great advantage over the public switched telephone network (PSTN), because it is free of charge. Depending on a network bandwidth, VoIP makes it possible to use more than one phone call. Regardless of location, customers can call from any place where there is an access to the Internet for connecting to a VoIP provider. The available integration with other services via the Internet, including video calls, a message transmitting and data during a call, an audio conference, an address book management, and information available on calls to other subscribers.

Currently, high-bandwidth networks can not only transmit messages, audio, but also video content. The high performance of modern devices can allow broadcasting a video of High Definition (HD) quality. With this feature, it is possible to raise the communication to a new level.

Videoconference is a technology providing a simultaneous two-way transmission, processing, transformation and presentation of interactive media at a distance in real-time. Videoconferencing is used for reducing travel costs in geographically distributed organizations and for distance learning. Information from the outside world comes to us through five senses. They are defined as sight, touch, taste, hearing and smell. The most important of which are hearing and sight. Research psychologists have concluded that in a telephone conversation on an average about 20% of the information are auditory perceived, whereas in the course of personal communication - 80%, and in the course of the videoconferencing session - 60%. If the interlocutors communicating across the sound (auditory) channel are added to a visual non-verbal language (e.g. gestures, facial expressions), the interlocutors will improve the efficiency of information perception. As we can see from their characteristics, videoconference is fairly close to a personal communication and far from a telephone conversation. Videoconference allows organizing conferences with multiple parties and exchanging more information (e.g. presentations, documents).

Research problems and methods

During the development stage the following problems required solutions:

- Organizing a delivery of video and audio streams via Java-coded application(s)
- Organizing a delivery of video stream from a variety of sources to a variety of target screens

In this project I used the Java programming language. Choosing Java as the language was crucial for its combination of features:

- Platform Independence
- Object Orientation
- Security

The research method used for this work is the constructive method. “Constructive research method implies building of an artifact (practical, theoretical or both) that solves a domain problem (including a model for existing phenomena) in order to create knowledge about how the problem can be solved (or understood, explained or modeled), and if previous solutions/models exist, how the solution/model is better than previous ones”. [26]

During the development stage various materials were used as sources of information: technical books, internet articles and available source codes. All the material was carefully studied and searched for the most suitable solutions of problems.

2. DEFINITIONS

The system is regarded as a software product for Voice-over-IP video communication. From now on the Video-Communication System will be referred to as a “system” or a “given system” in this document.

The system will enable a video communication between representatives of a house-holding company or other service companies (including social and health-care) and building inhabitants (as well as between inhabitants themselves).

This system is described from the inhabitants’ point of view. They will later be referred as a “user” in this document.

The other part of the communication (a house-holding company or another service company) will be referred as a “serving company” in this document.

A video multicast from a serving company or a video-call from another user reaching the user will be referred as a “down-link” in this document.

An employee of serving company who initialises the down-link will be referred as an “operator” in this document.

A peer-to-peer video-call from the user to the operator or to another user will be referred as an “up-link” in this document.

The work requires an evaluation of existing systems/platforms/solutions. The aim is to consider Open Source as an opportunity over proprietary solutions. By selecting a software platform or developing PBOL’s own one, the system prototype can be built and integrated into an existing infrastructure. Then, a required number of testing and development cycles have to be performed in order to achieve a required level of satisfaction among test users.

An opportunity to use IPTV for developing a video-communication system for multicast and peer-to-peer communication has to be discovered.

A system like this may be enhanced with a rich selection of other functionalities, such as a video door-intercom system, an environment observation system and other video systems. This system is an essential part of the Ubiquitous Home Environment which will complement the other systems in many possible Ubiquitous Computing scenarios.

Developing the system will be done in three stages:

1. Developing the down-link functionality

2. Developing the up-link functionality

3. Joining both, down-link and up-link into one system

4. Developing a conference-call-like functionality (The operator will be able to receive multiple video calls from many users.)

2.1 Hardware

The first type of device in the system is a device in a tenant's apartment for presenting video data. It can be a TV display with an Ethernet - enabled set-top box. The set-top box is a small computer providing two-way communications on an IP network and decoding the video streaming media. [37] Another case is to use a PC, for example a Touch-screen panel, connected to an IP network. This device has two main advantages. First, it enables one to interact directly with what is displayed, rather than indirectly with a cursor controlled by a mouse or touchpad. Secondly, one does not need to hold any intermediate device in hands to operate it.

The second type of device in the system is a video capture device. It can be an IP camera. The IP camera utilizes Internet Protocol for sending and receiving data via a computer network (WAN/LAN). The IP camera can be wireless. That allows cameras to be placed just about anywhere without using any cables. The camera should have a high quality imaging with support of high screen resolution in case if the client wants to present video data on a wide screen TV. The better image quality uses a higher bandwidth for transmitting: a typical IP camera with resolution of 640x480 pixels and 10 frames per second (10 fps) in MJPEG mode requires about 3 Mbit/s of network bandwidth. Although newer codecs offer the use much less bandwidth, H.264 on average requires 5-6x less bandwidth than MJPEG.

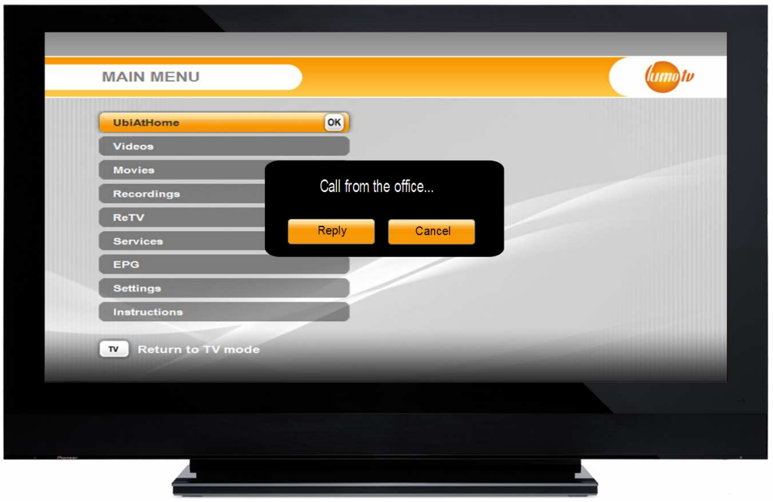
Another option for the second type of device is a webcam. A webcam is a video capture device connected to a computer or computer network, often using a USB port or, if connected to a network, Ethernet or Wi-Fi. As one of the options, a PC with a built-in camera can be used as a client in a tenant's apartment.

The third type of device in the system is a dedicated server in a company's office. It should have enough resources because it carries the main payload. If necessary, the server will encode a video into a suitable format for the customer device.

2.2 User's perception

From user's point of view the most important issue is security. One must be sure that, when unused, the camera is switched off and nobody can see what is happening in the apartment without the user's permission. TV is a device for presenting the content received from STB, which itself, is connected to an IP-TV network. The TV should be complemented with an IP camera. This is used for video-communication.

In case of a call, the user receives a notification on the TV screen. The message box contains a service name or the name of the caller, registered in the system (see Figure 1).



Call from the office...

Figure 1. Call notification.

By choosing “reply” the user is directed to the screen, where the video of the caller is displayed.

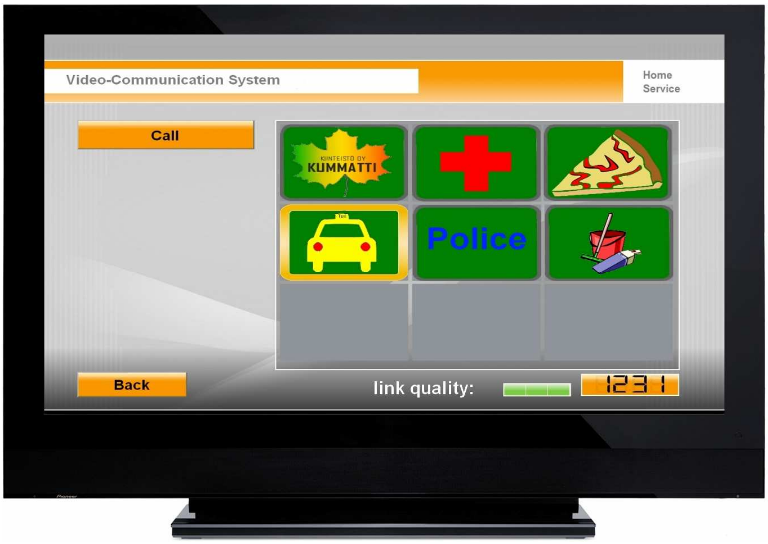


Figure 2. Services available to user

The Figure 2 shows the “Home Service” screen of the “Video-Communication System”. It allows the user to call for other services e.g. taxi or ordering a pizza. In order to proceed with the call, the user must select one of the available services from the menu.

If operator responds, the user is automatically redirected to the channel where the video content from an operator is broadcasting. Otherwise, the user is notified that the operator is busy and the call is rejected.

2.3 Technologies

2.3.1 Video over network

Video can be delivered to end users in several different ways. Video on Demand and Live Streaming are the most well-know. “Streaming video is content sent in compressed form over the Internet and displayed by the viewer in real time”. [40] Video on Demand (VOD) is a system, which allows users to select and watch/listen to video content on demand. “IPTV technology is often used to bring video on demand to televisions and personal computers”. [39] A live video stream in contrast to a video on demand is only available at the times video is transmitted. Video on demand on the contrary is always available on client’s request.

A unicast system (Figure 3), where each client has a direct connection to the server, has an advantage for its easiness to implement. A disadvantage is in the capacity required from the server to cope with all the video streams because an own video stream is transmitted to each client and this of course needs a greater network bandwidth. Unicast is the norm for most Internet connections, but does not scale well when many users want to view the same program concurrently.

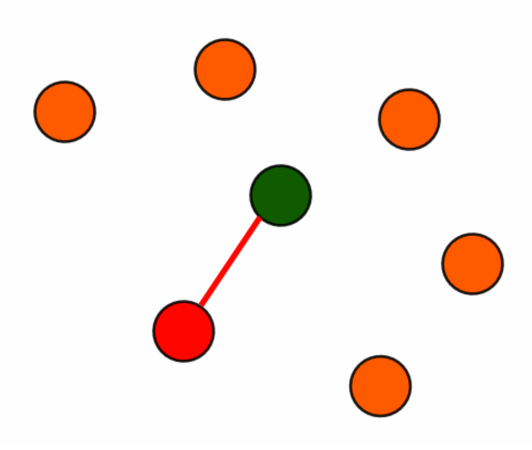


Figure 3. Unicast streaming

A multicast (Figure 4) in contrast to unicast has one video stream to several clients. This reduces the data replication and consequent server/network loads. One potential disadvantage of multicasting is the loss of video on demand functionality. Continuous streaming of broadcast usually precludes the recipient's ability to control playback.

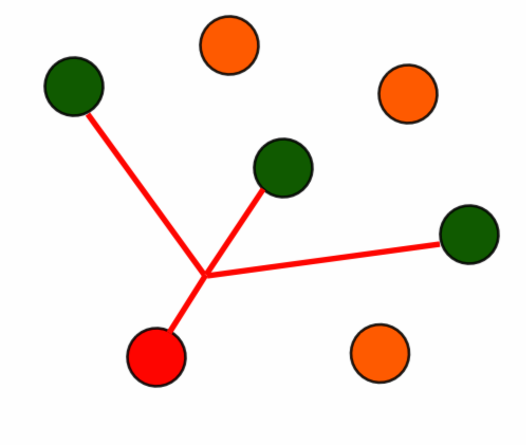


Figure 4. Multicast streaming.

“IP Multicast provides a mean to send a single media stream to a group of recipients on a computer network. A multicast protocol, usually IGMP, is used to manage the delivery of multicast streams to the groups of recipients on a LAN”. [44]

Peer-to-Peer networks now offer a possibility to distribute video content to an unlimited number of users and reducing the bandwidth on the source. The use of centralized server architecture is not needed since the video distribution throughout the network is handled by each client. Bandwidth savings at the server side can be achieved through P2P video content delivery systems. These P2P systems reduce the server’s work enormously and make the system more scalable because a new peer joining the network has no longer an impact on the server’s resources. In a P2P system there is always the risk for a source peer to suddenly fail or disconnect.

2.3.2 Protocols

- “UDP sends the media stream as a series of small packets”. [44] There is no mechanism within the protocol to guarantee delivery. UDP can’t detect loss or corruption and recover data.

- TCP protocol in contrast to UDP delivers datagrams, called segments, in the form of byte streams once the connection is established.TCP is used in those cases that require a guaranteed delivery of data. It uses checksums to check the integrity of packages.

- RTP (Real-time Transport Protocol) is provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services.

- “RTCP (Real-Time Transport Control Protocol) is a sister protocol of the Real-time Transport Protocol (RTP). RTCP provides out-of-band statistics and control information for an RTP stream”. [45]

- RTSP (Real Time Streaming Protocol) is a network control protocol. The RTSP protocol has similarities to HTTP, but RTSP has new requests. The protocol is used to establish and control media sessions between server and client.

2.3.3 Video handing in Java

Java supports handling of multimedia files with the Java Media Framework API (JMF) [32], JFFMPEG [17] and VLC/JVLC. This allows using video in Java.

1. Java Media Framework API (JMF)

“JMF is a Java library that enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, play, stream, and transcode multiple media formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications”. [35]

JMF supports a wide array of media types, including

- protocols: FILE, HTTP, FTP, RTP
- audio: AIFF, AU, AVI, GSM, MIDI, MP2, MP3*, QT, RMF, WAV
- video: AVI, MPEG-1, QT, H.261, H.263
- other: HotMedia

2. VLC and JVLC

VLC media player (initially VideoLAN Client) is a multimedia player for various audio and video formats (MPEG-2, MPEG-4, H.264, DivX, MPEG-1, mp3, ogg, aac ...) as well as DVDs, VCDs, and various streaming protocols. It can also be used as a media converter or server to stream in unicast or multicast. Additionally the player allows the use of the H.264 / AVC codec, with an implemented x264 library. JVLC is an interface to control the VLC video player. It will use the libvlc.dll to manage and start the videoplayer. Libvlc represents the underlying API of VLC. VLC itself is just a wrapper around libvlc. By utilizing libvlc developers can take advantage of all the complex functionalities of VLC. JNA with libvlc.dll provides easy access to get all functionalities of videoplayer.

Figure 5 shows the VideoLan streaming solution. VLM or VideoLAN Manager is used to control video streams. The VLM is the part of the VLC. VideoLAN Manager is a small media manager designed to control multiple streams with only one instance of VLC. It allows multiple streaming and video on demand (VoD).

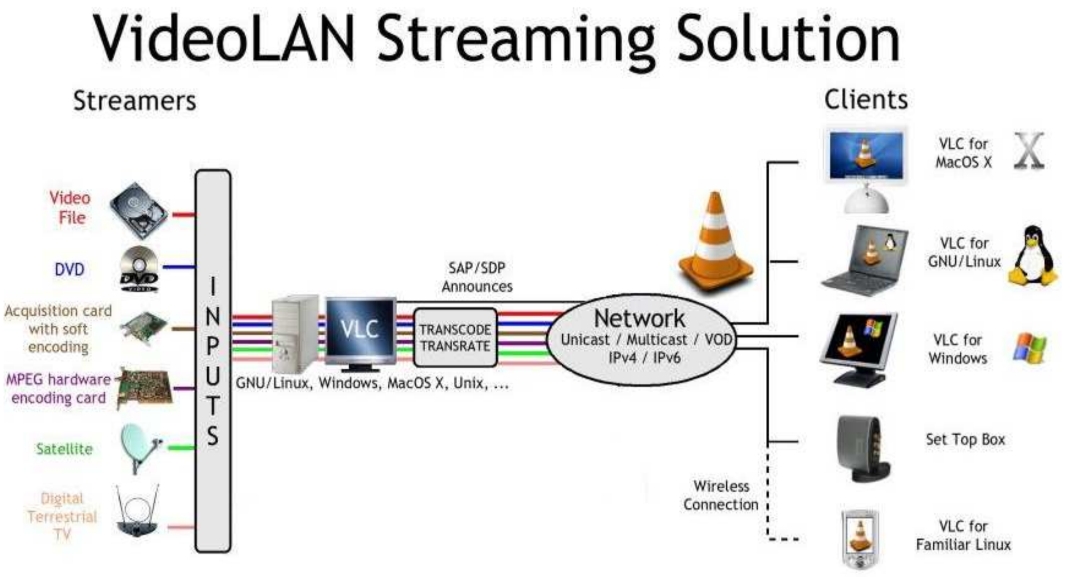


Figure 5. VideoLAN streaming solution

There are three types of media streaming:

- “Broadcast: A broadcast media is very close to a TV program or channel. It is launched, stopped or paused by the administrator and may be repeated several times. The client has no control over this media”. [34]
- “Schedules: A *Schedule* is a script with a date. When the schedule date is reached, the script is launched”. [34]
- “VOD: A vod media is commonly used for Video on Demand. It will be launched only if a vod client asks for it”. [34]

3. JFFMPEG

JFFMPEG is another open source project built on FFMPEG. “FFMPEG is a complete, cross-platform solution to record, convert and stream audio and video”. [36] “JFFMEPG is a plug-in for JMF that allows the playback of a number of common audio and video formats”. [46] A JNI wrapper allows calls directly into the full FFMPEG library. But again there is only support for decoding video streams, there’s no encoding for video streams.

5. Summary

Project	Decoding	Encoding	H264 playback	H264 encoding	RTSP protocol
JMF	yes (but limited to some video codecs)	yes (but limited to some video codecs)	no	no	client side
JFFMPEG	yes	no	yes	no	client side
JVLC	yes	yes	yes	yes	client/server side

Table 1: Video Supporting Projects for Java

From all of these open-source software projects reviewed in the development, the JVLC suits best for this work. Unlike all the other, JVLC supports a large number of codecs. It has a full RTSP protocol support. Table 1 gives an overview of the available projects for video handling in Java.

2.3.4 JNA (Java Native Access)

JNA provides Java programs an easy access to native shared libraries (DLLs on Windows). JNA allows you to call directly into native functions using natural Java method invocation. The Java call looks just like it does in native code. The developer uses a Java interface to describe functions and structures in the target native library.

2.3.5 JAVA

The Java technology is both an object-oriented programming language and a platform. It is based on the concept of a virtual Java-machine (Java virtual machine - JVM) - the translator between the language and the underlying operating system and hardware platform. All implementations of a programming language should contain a JVM, due to which programs written in Java, run on any operating system, where an appropriate version of JVM exits.

The Java programming language is unusual because on one side Java programs are compiled (into an intermediate language called Java byte-code), but on another side they are interpreted (byte-code parsed and executed within the JVM). The compilation is done once as an interpretation occurs each time a program is started. The compiled byte-code is a form of optimized machine code for the JVM; the interpreter and the realization of JVM.

The platform for Java, which exists in three different versions, consists of JVM and the application programming interface Java API (Java Application Programming Interface). A large set of its ready-made software components simplifies the design and deployment of applets (applets) and applications

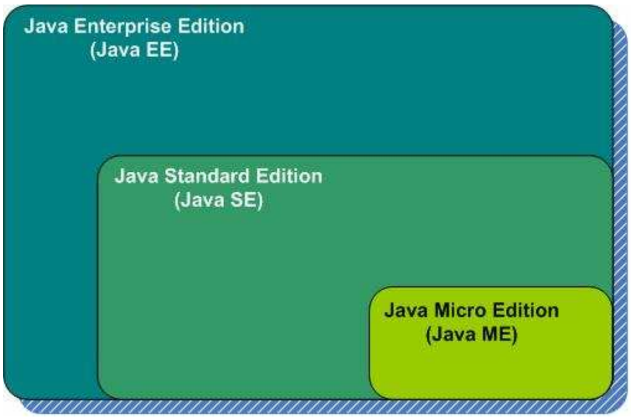


Figure 6. Editions of the Java platform presented.

Each of them is used:

- "Java Platform, Micro Edition (Java ME) — targeting environments with limited resources.
- Java Platform, Standard Edition (Java SE) — targeting workstation environments.
- Java Platform, Enterprise Edition (Java EE) — targeting large distributed enterprise or Internet environments." [47]

2.3.6 MySQL DBMS

MySQL is a database management system (RDBMS) that runs as a server providing a multi-user access to a number of databases. Relational database is a body of related information stored in two-dimensional tables. MySQL uses Structured Query Language (SQL). This is a language that allows you to create and work with relational databases, which are collections of related information stored in the tables.

It features:

- Free-software project

- Lowest Total Cost of Ownership

Save on database licensing costs

- Management Ease

Average time from software download to complete installation is less than fifteen minutes

- Flexibility

MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX

- MySQL Connectors

"MySQL provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language of choice. In addition, a native C library allows developers to embed MySQL directly into their applications". [48]

The figure 7 shows the interaction between the database and clients.

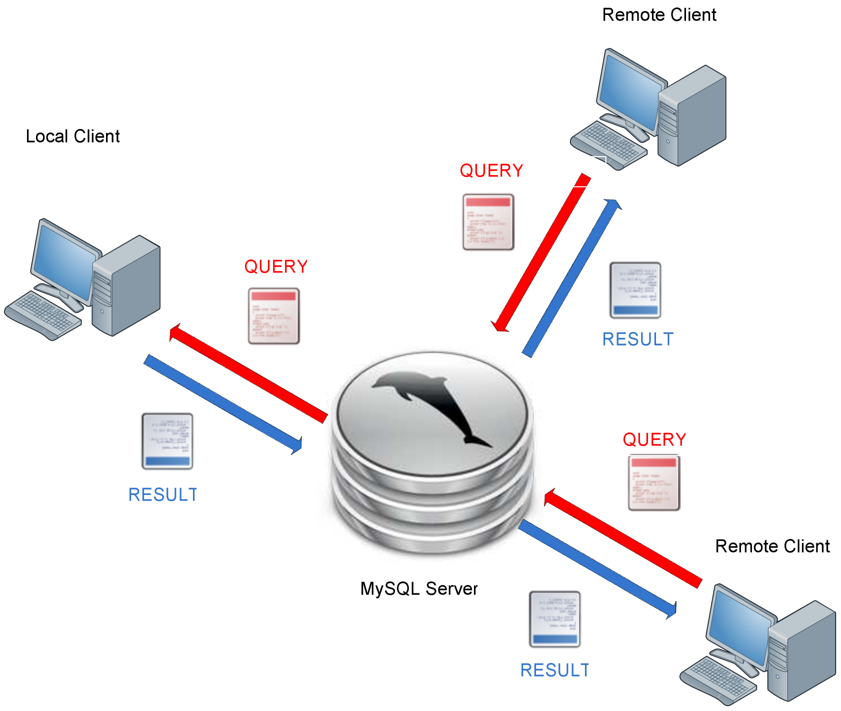


Figure 7. A multithreading MySQL database server.

2.3.7 MPEG TS

MPEG Transport Streams are combining (multiplexing) multiple program channels (typically digital video channels) into a signal communication channel (Figure 8). It's used by DVB, DVD and HDTV. The program may or may not have a common time base. The transport stream has a fixed length packet size:

- Intended for a non-error free environment
- Easier to detect start and end of frames
- Easy to recover from packet loss/corruption
- More difficult to produce and demultiplex than program stream

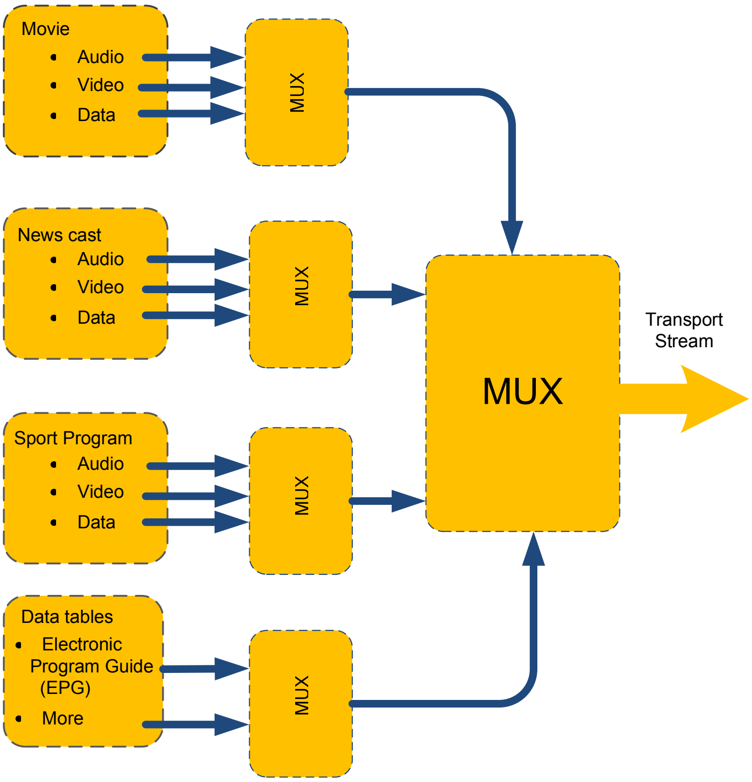


Figure 8. MPEG Transport Stream

2.3.8 IP-TV

Word IPTV has many different definitions. By the International Telecommunication Union focus group on IPTV (ITU-T FG IPTV) defines IPTV as follows:

"IPTV is defined as multimedia services such as television/video/audio/text/graphics/data delivered over IP based networks managed to provide the required level of quality of service and experience, security, interactivity and reliability". [21]

Another official and more detailed definition of IPTV is given by Alliance for Telecommunications Industry Solutions (ATIS) IPTV Exploratory Group in 2005:

"IPTV is defined as the secure and reliable delivery to subscribers of entertainment video and related services. These services may include, for example, Live TV, Video On Demand (VOD) and Interactive TV (iTV)*. These services are delivered across an access agnostic, packet switched network that employs the IP protocol to transport the audio, video and control signals. In contrast to video over the public Internet, with IPTV deployments, network security and performance are tightly managed to ensure a superior entertainment experience, resulting in a compelling business environment for content providers, advertisers and customers alike".* [22]

Both of these concepts combined into one give a transmission of multimedia data to customers over an IP based network. Instead of a signal being transmitted via satellite or cable, The IPTV signal is transmitted via a broadband line. This makes it possible to view IPTV by other compatible multimedia devices, such as a PC or a mobile phone. IPTV is typically bundled with other services like Video on Demand (VOD), Voice over IP (VoIP) and Web access.

The IP-based platform offers significant advantages, including the ability to integrate television with other IP based services like high speed Internet access and VoIP.

“A switched IP network also allows for the delivery of significantly more content and functionality. In a typical TV or satellite network, using broadcast video technology, all the content constantly flows downstream to each customer, and the customer switches the content at the set-top box. The customer can select from as many choices as the telecomms, cable or satellite company can stuff into the “pipe” flowing into the home. A switched IP network works differently. Content remains in the network, and only the content the customer selects is sent into the customer’s home. That frees up bandwidth, and the customer’s choice is less restricted by the size of the “pipe” into the home. This also implies that the customer's privacy could be compromised to a greater extent than is possible with traditional TV or satellite networks. It may also provide a means to hack into, or at least disrupt the private network”. [23]

2.3.9 Hibox platform

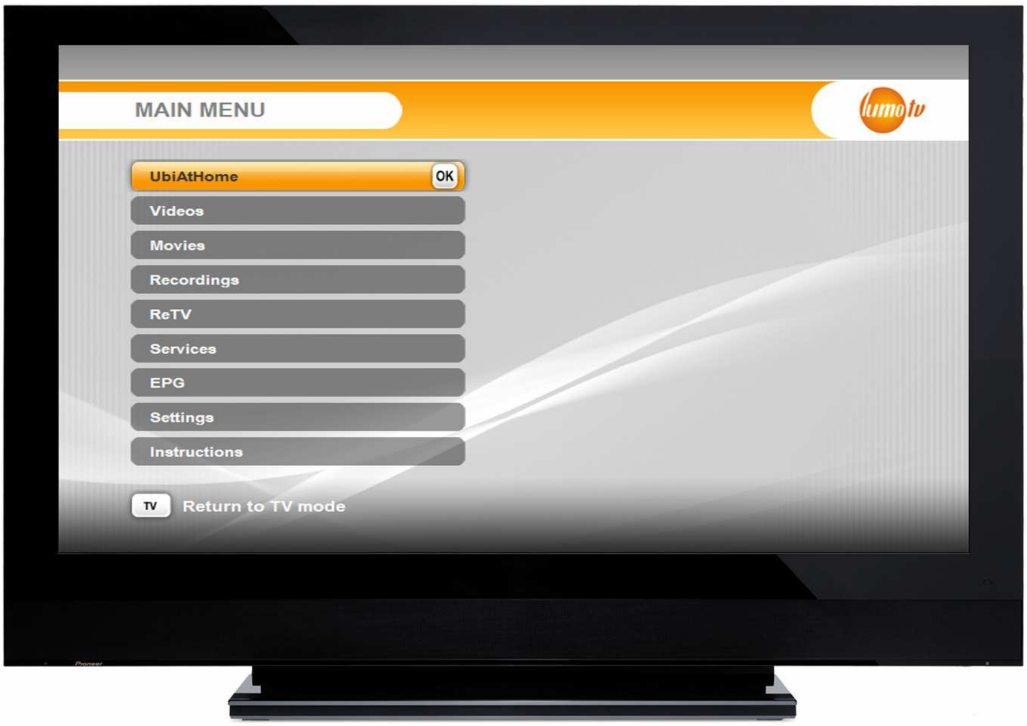


Figure 9. Example of UI developed on a base of Hibox platform

Hibox deluxe (Figure 9) is a proprietary “complete IPTV middleware for delivering TV, Video-On-Demand and interactive services to customers in any IP-based network. Hibox deluXe transforms the television into a truly expedient and user friendly entertainment center, providing the user with all the main uses of a television, computer and other equipment. Using the Hibox deluXe solution the user has easy access to TV programs, movies, recording and various interactive services". [33]

Key features:

- TV-channels-IPTV,DVB-T,DVB-C or DVB-S
- TV-On-Demand, Video-On-Demand
- Network-PVR, Timeshift-TV, Catch-up-TV
- Billing & provisioning
- STB – independent

2.3.10 KreaTV Application Platform

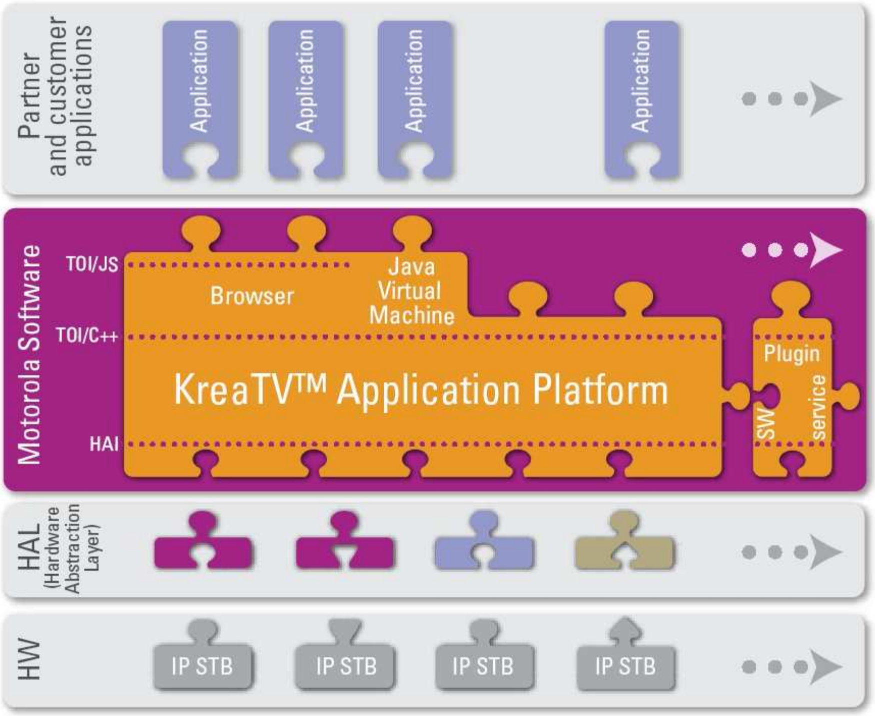


Figure 10. KreaTV platform

"The Motorola KreaTV Application Platform (Figure 10) is the ultimate open software platform for IP set-tops. It is reliable, flexible and cost-effective". [31]
Features:

- "General software platform for fast application development and integration
- Pre-integrated with leading middleware applications, Conditional Access systems and video servers
- Open standard interfaces for browser or native application level development
- Separates software and hardware for increased product lifetime

- A true multi-application IP STB software for increased performance
- Modular architecture designed for high reliability in operation
- Full scalability in number of hardware models, subscribers and services” [31]

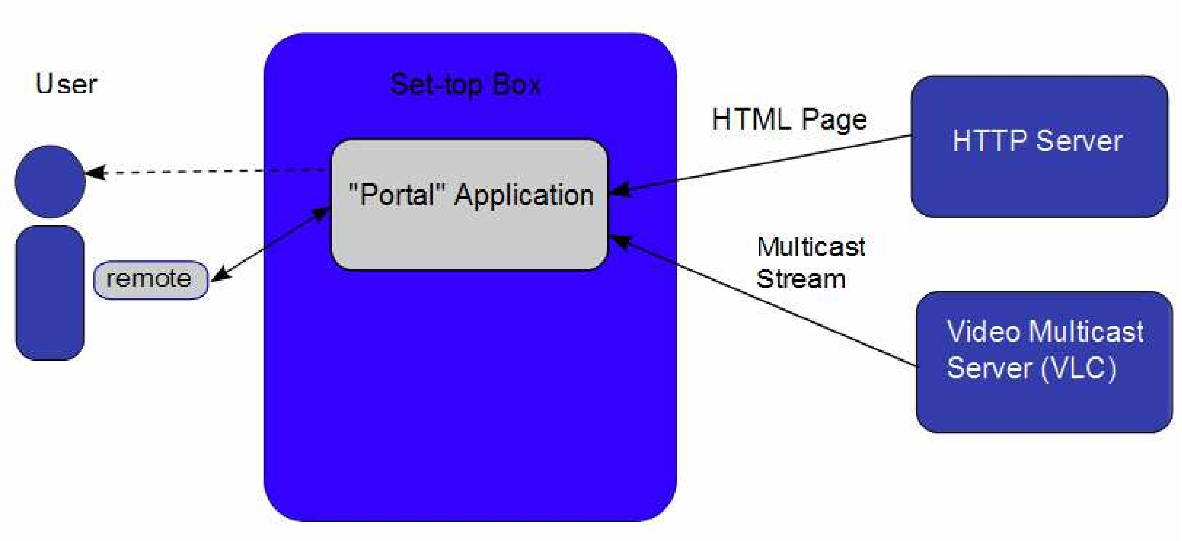


Figure 11. Motorola Set-top Box and interaction with users

The portal application is a modified Mozilla browser without a navigation bar. It loads and displays a single HTML page. This browser supports “TOI/JS” JavaScript API, which allows the interaction with a user. The idea of video multicasting is that each TV channel is a different multicast stream on a different IP address. For multicast streaming, the server can use a VLC application. (Figure 11)

“The IPTV middleware uses client/server architecture and the client often resides at the STB. The middleware controls the customer interaction with the service and, therefore, plays a very important role in controlling the customer’s overall experience”. [30]

For this project an IPTV middleware is the Hibox platform. The Hibox middleware tells the clients (STBs or PCs) where they can find the video feed (i.e. provides the metadata) and then the client’s media player is responsible for the actual playback (RTP). The most important thing is that the source video feed is compatible with the client’s media player. Using the Motorola STBs or the VLC-browser plug-in as a client, it is possible to try a VideoLAN capture and encoding to 720x576 25fps MPEG2 MPEG-TS on some multicast URL (224.1.2.3:1111 for example) from a camera or a capture card. Then you must add this URL as a new TV channel in the middleware (url = 'igmp://224.1.2.3:1111', any name, any rating epg id etc.) so that the STB can find it. STB / PC needs to reload the user interface (UI) in order to "see" the newly added channel. It is also possible to make a unicast stream and add that as a channel, movie or video.

2.3.11 SIP

Session Initiation Protocol (SIP) is used for VOIP and other text and multimedia sessions, like instant message, video and other services.

SIP is very much like the HTTP protocol and the Simple Mail Transfer Protocol (SMTP). Messages consist of headers and a message body. The protocol suffers from NAT or firewall restrictions. The voice and video stream communications in SIP applications are carried over another application protocol, the Real-time Transport Protocol (RTP). Parameters (port numbers, protocols and codecs) for these media streams are defined and negotiated using the Session Description Protocol (SDP) which is transported in the SIP packet body. The Figure 12 presents SIP communication between User A and User B.

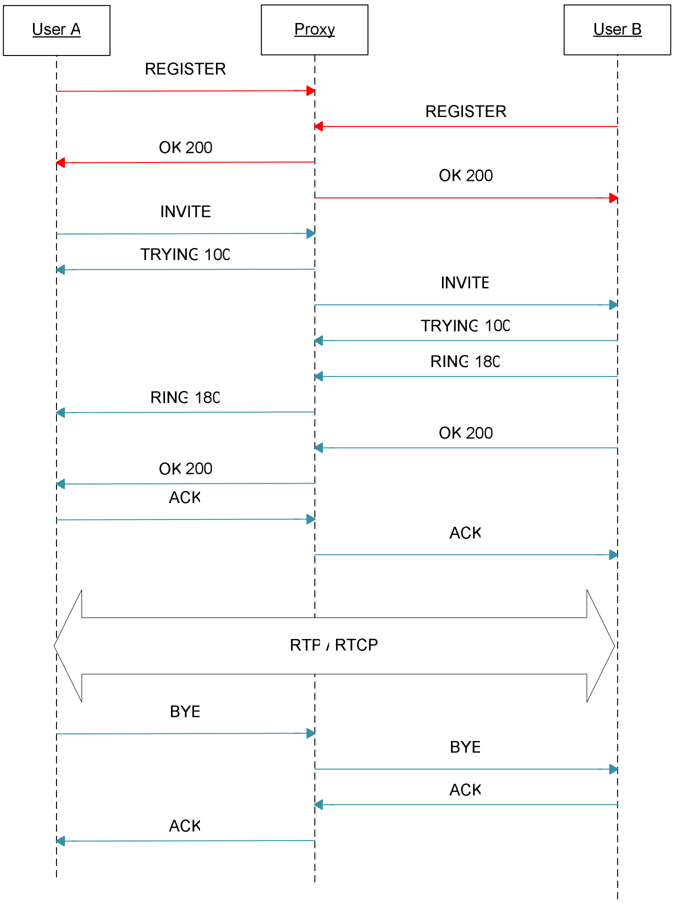


Figure 12. SIP Communication

- “**The first step is the user register.** The users must register themselves to be found by other users. In this case, the terminals send a REGISTER request, where the fields "from" and "to" correspond to the registered user. The Proxy server, who acts as Register, consults if the user can be authenticated and sends an OK message if there is no problem”. [9]

-“**The following transaction corresponds to a session establishment.** This session consists of an INVITE request of the user to the proxy. Immediately, the proxy sends a TRYING 100 to stop the broadcastings and reroute the request to the B user. The B user sends a Ringing 180 when the telephone begins to ring and it is also reroute by the proxy to the A user. Finally, the OK 200 message corresponds to the accept process (the user B response the call)”. [9]

-“**At this moment the call is established**, and the RTP transport protocol starts with the parameters (ports, addresses, codecs, etc.) of the SDP protocol”. [9]

-“**The last transaction corresponds to a session end**. This is carried out with an only BYE request to the Proxy, and later reroute to the B user. This user replies with an OK 200 message to confirm that the final message has been received correctly”. [9]

3. TECHNICAL INFRASTRUCTURE

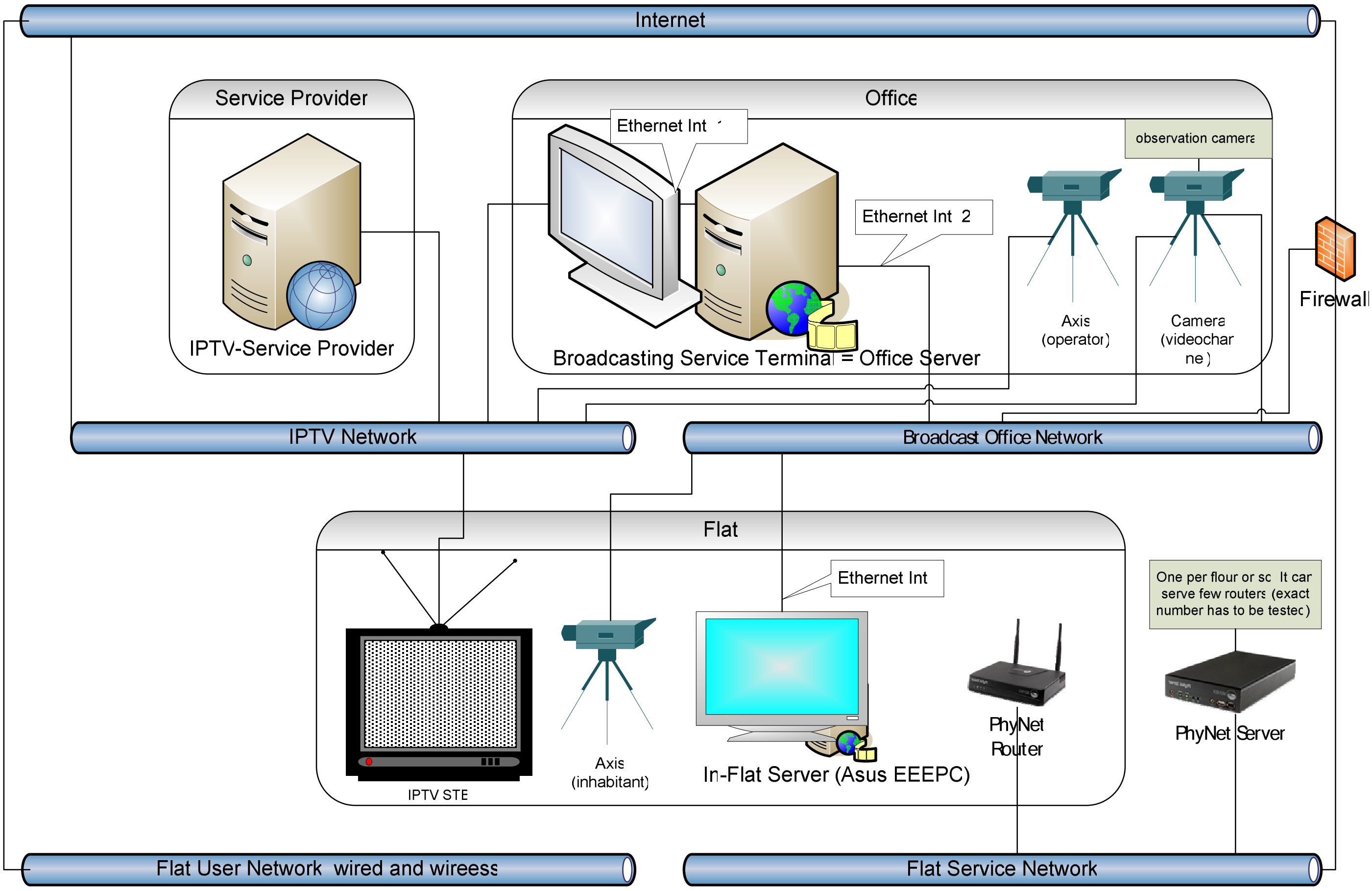


Figure 13. Technical infrastructure

The technical infrastructure (Figure 13) consists of a house with flats and the service office. Those premises are equipped with various electronic devices in different networks. The network infrastructure is divided into the following items: a flat network, an IPTV network and an office network. A video is transmitted over the IPTV network because it is a private network with no access to the Internet. Therefore it is more secure. An STB device connected to the IPTV network is providing video content. This is also done by the office server, which broadcasts the content to users. The IPTV network also contains an IP camera in the office. The flat network includes a sensor server from the UbiAtHome project. The office network interconnects the main server, flat camera, touch-screen panel and other office equipment.

4. IMPLEMENTATION

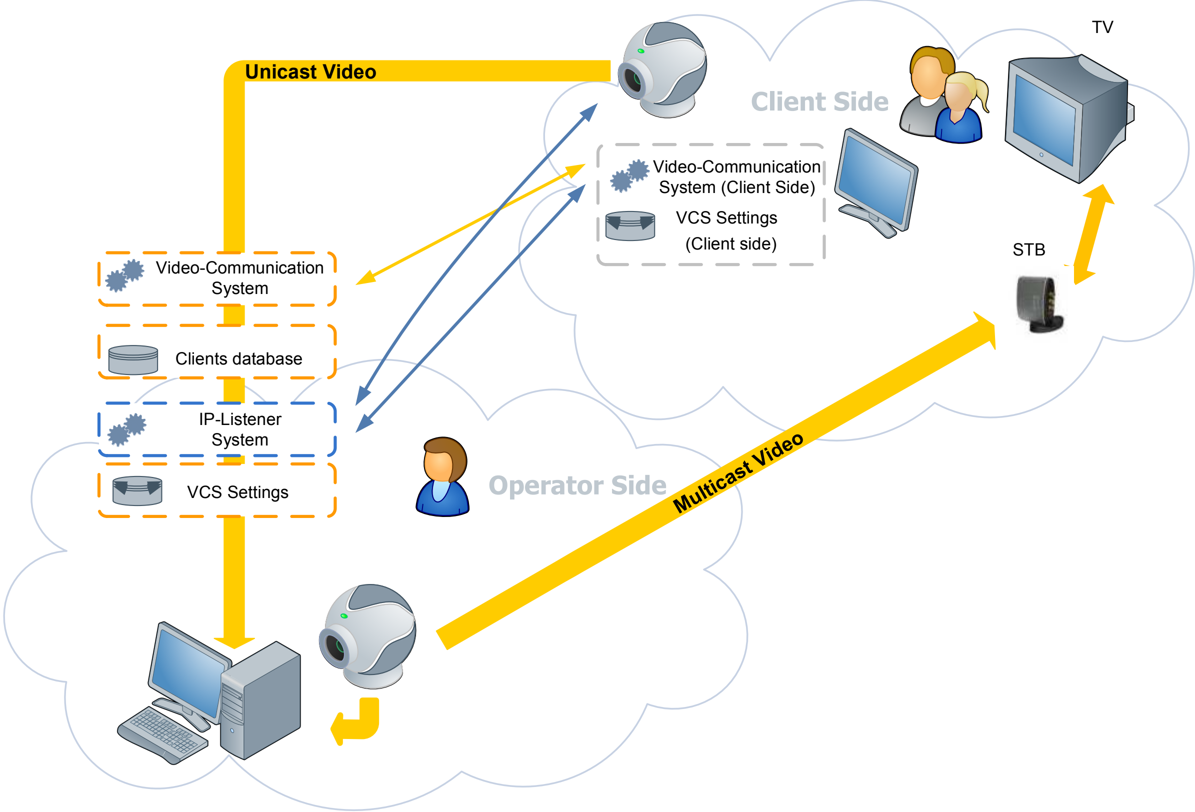


Figure 14. The structure of the system

The Figure 14 shows the entire structure of the system.

Client’s database contains a list of users, their MAC and IP addresses.

VCS Settings are the system configuration files.

IP-Listener System is a network service that monitors a specific port on the server. If some device sends a message of a certain type to the server and the MAC address of that device exists in the database, the server will then save the IP address of this particular device on the database entry with the corresponding MAC address.

The entire system consists of two major parts: the client and the operator. The operator side contains a server running the IP listener process, which listens to an IP address of the operator's program. This server is equipped with a camera, which is used for sending video data to the multicast IP address. Therefore the content is broadcasted to all clients. The camera can be wireless, but in order to avoid a bandwidth overhead, it was decided to use a wired one.

The client side can either consist of a STB with a TV and a camera or otherwise a touch screen panel with a built-in camera. The STB maintains a channel that is configured to fetch a video data broadcasted by the operator via the multicast address.

4.1 The tools used

During the design and development stage the following software applications and development tools were used:

- NetBeans - A free, open-source Integrated Development Environment (IDE) for software developers. NetBeans contains all the tools needed to create desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby.

- Wireshark - A network protocol analyzer for UNIX and Windows. It is used for network troubleshooting, analysis, software development, and education.

- VLC player - A media player, supporting many audio and video codecs and file formats. VLC media player can be used as a server and client to stream and receive network streams.

- MySQL Query Browser - A tool to help developers to write SQL queries and analyze data in a graphical environment.

- PS Pad - A freeware text editor intended for use by programmers

- J2SE SDK - Software Development Kit for Java language.

4.2 Multicast stream

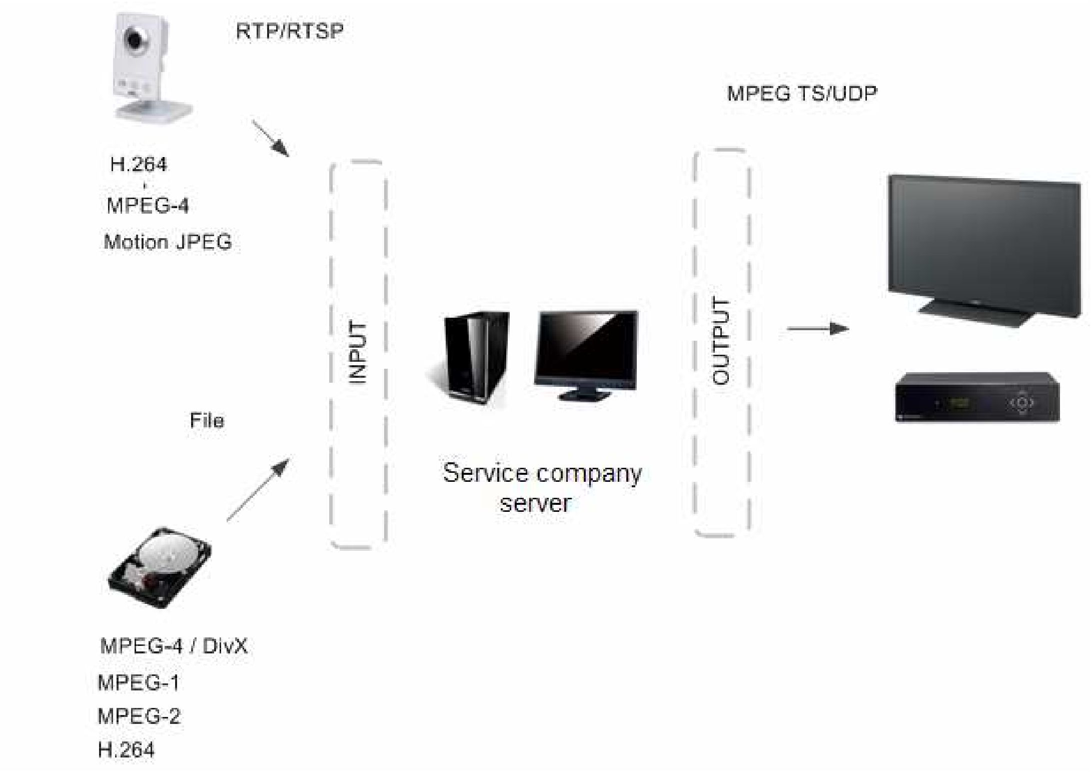


Figure 15. The process of multicasting a video to users

The figure 15 shows the process of broadcasting a video to users. The VLC media player receives an incoming video stream. This may be a local video file or a video stream from a camera. The input stream may have a different method for video compression. Once the video reaches the server, it is transcoded into a transport stream with an H.264 video compression and sent over a network to a multicast address, from which the client part of the system receives a video. The addBroadcast() method from the JVLC library was used to create a broadcast stream:

```
void addBroadcast(java.lang.String name,

            java.lang.String input,
            java.lang.String output,
            java.lang.String[] options,
            boolean enabled,
            boolean loop)
            throws VLCException
```

Adding a broadcast with one input:

Parameters:
name - name of a new broadcast
input - an input MRL (Multimedia Resource Locator)
output - an output MRL (a parameter for a "sout" variable)
options - additional options
enabled - boolean value for enabling a new broadcast
loop - a parameter for playing this broadcast in a loop
Throws:
VLCException

Appendix 1 contains the source code for broadcasting media to a multicast address.

```
String output = "#transcode{acodec=mp3,vb=128, samplerate=48000 ,canals=2, vcodec =”+
“h264,vb=2000,scale=1 }:std{access=udp,mux=ts,dst=239.235.1.1:1111}";
```

The variable “output” contains the parameters for the output stream. The description for parameters is:

transcode

This module is used for transcoding a stream, i.e. to change its codecs or the encoding bitrates. Some additional processing can be done during this process, such as re-scaling, deinterlacing, resampling, etc.

acodec= This options allows to specify a codec, to which the audio tracks of the input stream should be transcoded.

vb= A bitrate of the transcoded video stream, in kbit/s

samplerate= A samplerate of the transcoded audio stream, in Hz.

channels= This option allows to set a number of channels of the resulting audio stream.

vcodec= This options allows to specify a codec, to which the video tracks of the input stream should be transcoded.

scale= This option allows to give a ratio, by which the video should be rescaled while transcoded.

standard (alias std)

This module saves the stream to a file or sends it over a network, after having multiplexed it.

access= This option allows to set the medium used to save or send the stream.

mux= This option allows you to set the encapsulation method used for the resulting stream.

dst= This option allows to give information about the location where the stream should actually be saved or sent to.

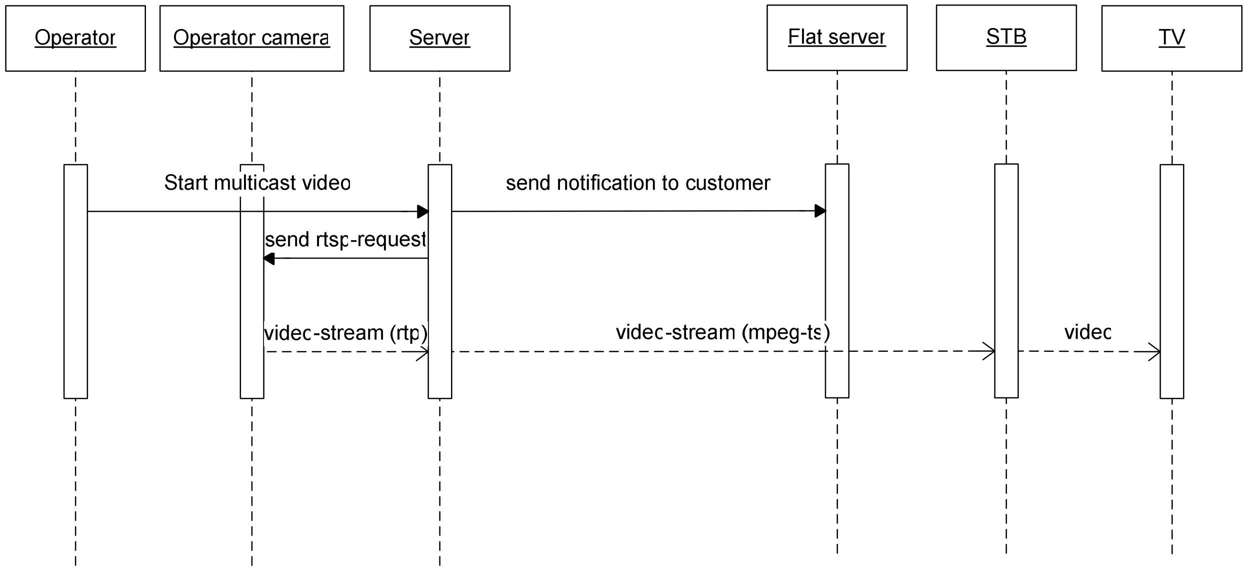


Figure 16. The sequence diagram of a video broadcasting process

The figure 16 shows the process of broadcasting a video stream to users. While the operator in the service office clicks a certain button of the server program, the program starts capturing a video from the operator's camera and broadcasts it over the IP-TV network. After this the server in the service office sends a message to the server in a flat, notifying that the video broadcast has been started. The camera and STB are using different protocols for video and audio streaming and data transfer. The camera supports the RTP format, while STB understands MPEG-TS. Therefore, the stream must be converted to the corresponding format. The process of converting the streams is done on the server using JVLC. The server receives audio and video streams in the RTP format, then converts and packs them to an MPEG-TS container and sends the result to STB. The STB has a special channel for a video broadcast from the operator. That channel is configured to fetch data from a multicast address. When the user receives a notification, he switches on a TV and switches to the channel, which will view a video from the operator.

4.3 User Interface

4.3.1 Operator side

4.3.1.1 Main panel

The figure 17 shows the main panel of the server side of the system. The button "Full Screen" switches the video display to the full screen mode and the "Esc" key leaves this mode back to original. The program supports two languages. The user interface's language change is done by pressing the "English" (and "Finnish") button. The volume can be adjusted with a slider. By pressing the "Mute" button the volume can be switched on/off.

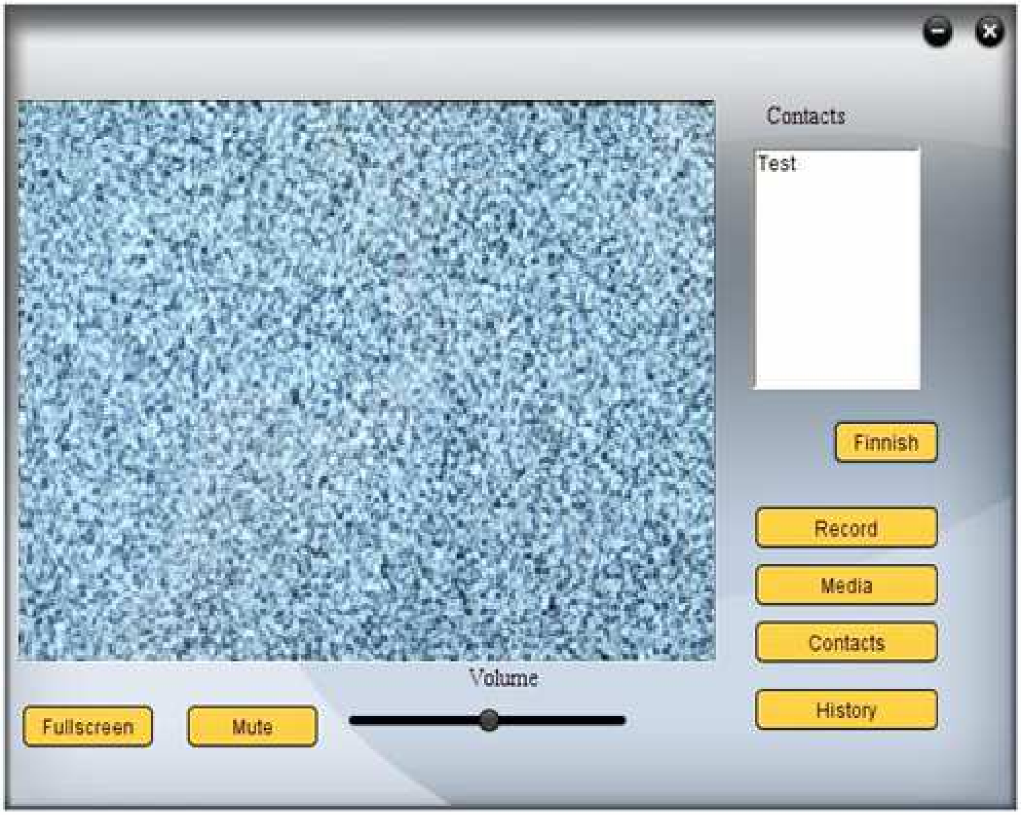


Figure 17. The panel view of the main service operator

The users list "Contacts" stores in a database all tenants who use the video communication system. The navigation between the main panels and few other panels of the program is done by pressing the "Contacts" - "Media" – "History" and "Record" buttons. When the user calls the operator, a notification message is received with an icon and a sound signal. After this the main panel with a displaying video opens up from the tenant's apartment. The caller's address is shown above the video screen.

Window control buttons:

⊗ - Exit program

⊖ - Minimize window to a system tray

4.3.1.2 Record panel

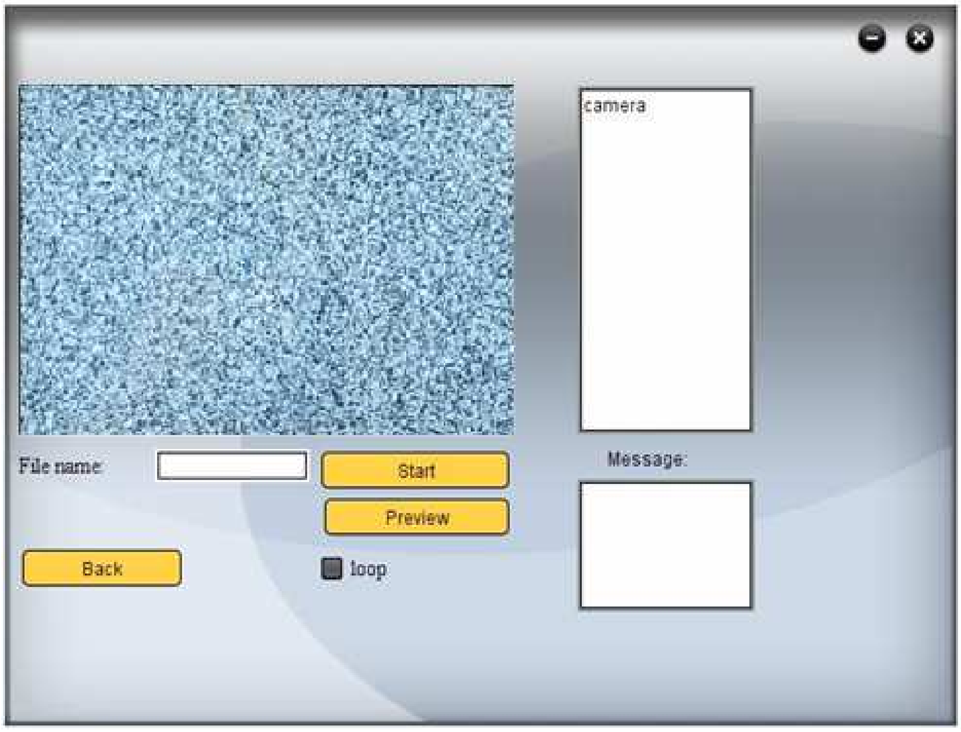


Figure 18. Record panel view

Record panel (Figure 18) is used to save the video from the camera to the hard drive. Once accomplished, the corresponding entry is added to the database.

The video preview can be seen by clicking the “Preview” button. The text written in the "Message" field will be sent to users along with the video as a notification message. The video files will be broadcasted repeatedly if the “loop” check-box is activated. Buttons "Start" and "Stop" are used to control the recording process.

4.3.1.3 Media panel

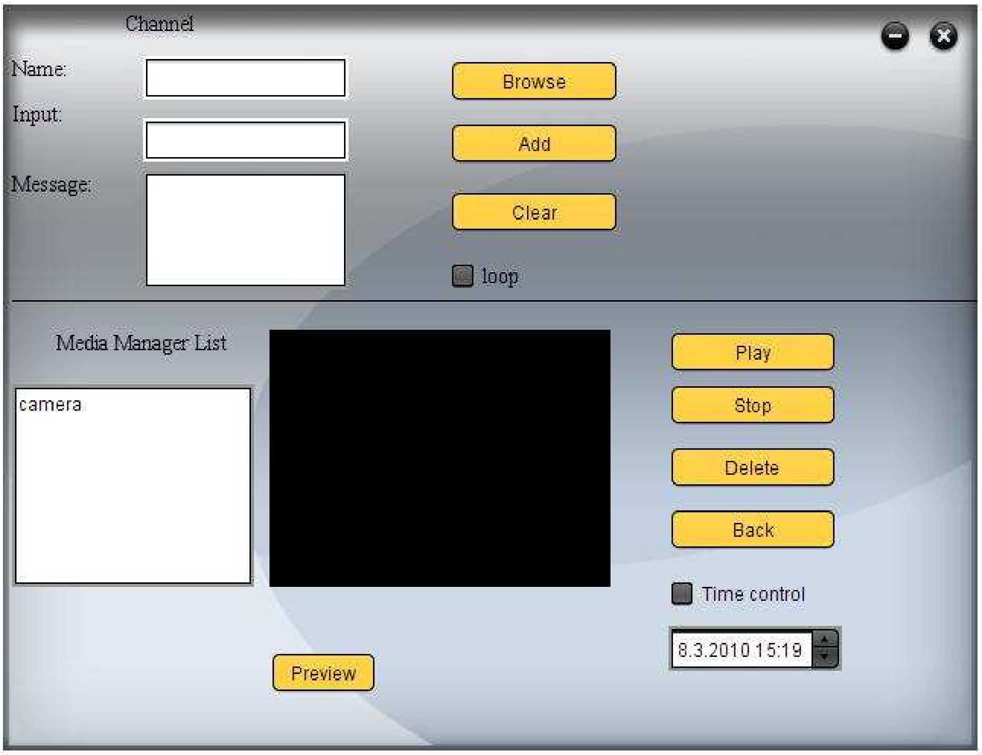


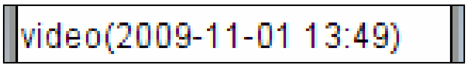
Figure 19. Media panel view

The media panel is used for searching and adding new files in order to broadcast and control selected media. The Media panel (see Figure 19) is split into two parts. The first part, Channel, is used for searching and adding new files for broadcasting. The second part, Media Manager List, is for controlling the selected media. Let's take a closer look at those parts.

The channel part consists of text fields and buttons. The "Name" field stores the media file's name. The "Input" field stores the absolute path to the actual media file. The system accepts local files as well as remote ones. In case the file is located in a remote location on the network, the user must specify a URL (e.g. rtsp: / / 10.20.44.4/axis-media/media.amp). If the user does not know an exact path for the local file, the "Browse" button can be considered. It initializes a file dialogue where the user can select a desired media file. A message sent along with the video can be input to a "Message" field. The "Clear" button erases all the text written inside editors. If the media is to be broadcasted continuously, a check box for the loop can be set. Finally, the "Add" button appends a new entry on Media Manager List.

The second part represents Multimedia Manager. For broadcasting, the list stores all available elements, which are executed by selecting them and clicking on the "Play" button. On the contrary, the "Stop" button will interrupt the broadcast. The "Delete" button erases the selected file from the list and the database. The video preview is launched by using the "Preview" button. If the user selects the "Time control" check-box, sets the desired date and time and clicks "Play", the broadcast will start as scheduled. In this case, the name of the selected element will be changed to be as the one in Figure 20. In order to stop the scheduled broadcast, the user simply needs to select the file from the list and press "Stop".

Finally, the “Back” button is used to navigate away from Media Panel.

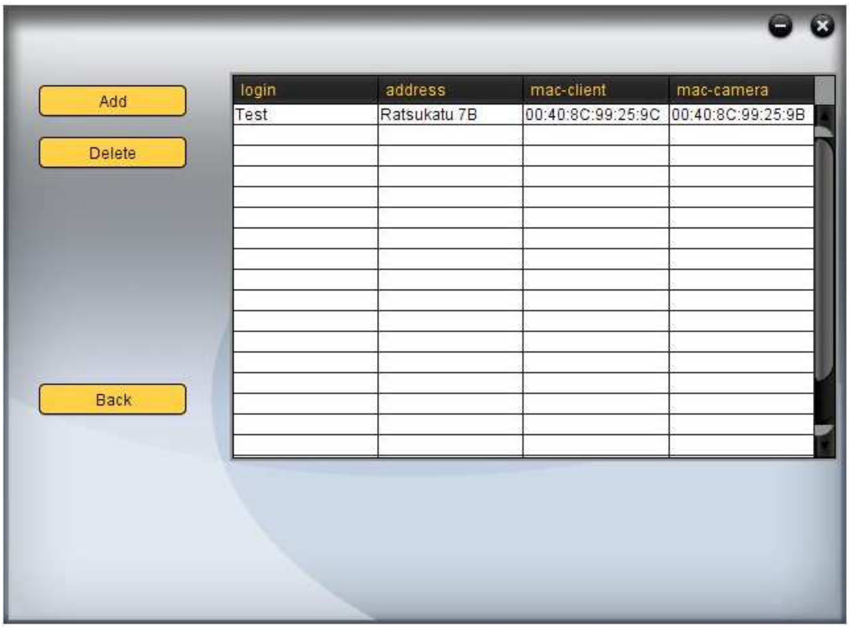


video(2009-11-01 13:49)

Figure 20. Scheduled video broadcast

4.3.1.4 Contacts panel

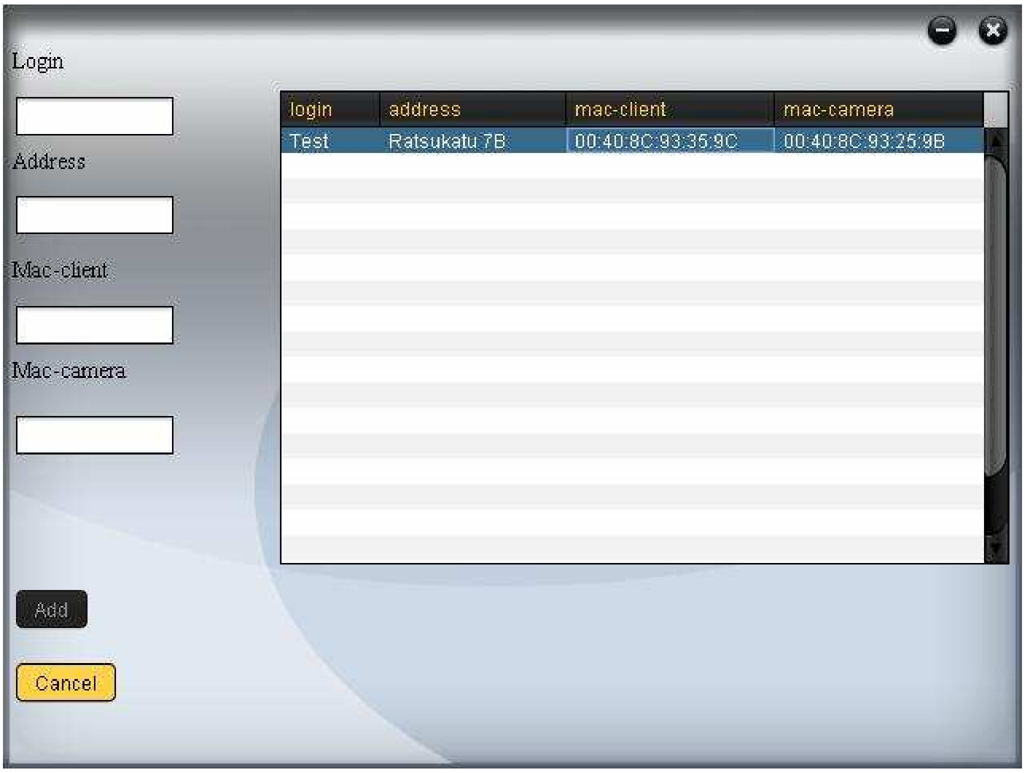
The contacts panel (Figure 21) is used for browsing the stored entries and modifying them from the “Contacts” database. It consists of a table and the following set of buttons: "Add", "Delete" and "Back". The table contains user info: ID, home address, MAC addresses of the flat server and the camera.



login	address	mac-client	mac-camera
Test	Ratsukatu 7B	00:40:8C:99:25:9C	00:40:8C:99:25:9B

Figure 21. Contacts panel view

The operator can add or delete any user from the database. Adding is accomplished by filling the new fields (“Login”, “Address”, “Mac-client” and “Mac-camera”) and clicking the “Add” button in the Contact panel (see Figure 22). The program verifies the input data and activates or deactivates the “Add” button. For example, if either a client MAC or a camera MAC field is in a wrong format, or if either a login or an address field is empty, the verification will fail. The MAC addresses must be in the correct format, e.g. 00:1A: 6B: 7B: 22:46. If the information is filled in correctly and the button “Add”, which is shown in the figure 22, is activated, the user can be added to the database. Otherwise, the user can be removed from the database using the “Delete” button. Finally, the “Back” button is used to navigate away from the “Contacts” panel.

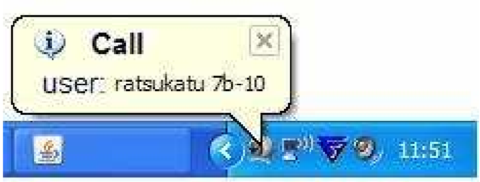


login	address	mac-client	mac-camera
Test	Ratsukatu 7B	00:40:8C:93:35:9C	00:40:8C:93:25:9B

Figure 22. Adding a new user to the system

4.3.1.5 Status bar icon

In case of a call, the operator receives a notification as demonstrated in Figure 23.



Call

user: ratsukatu 7b-10

Figure 23. Notice call

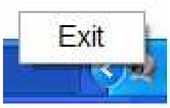
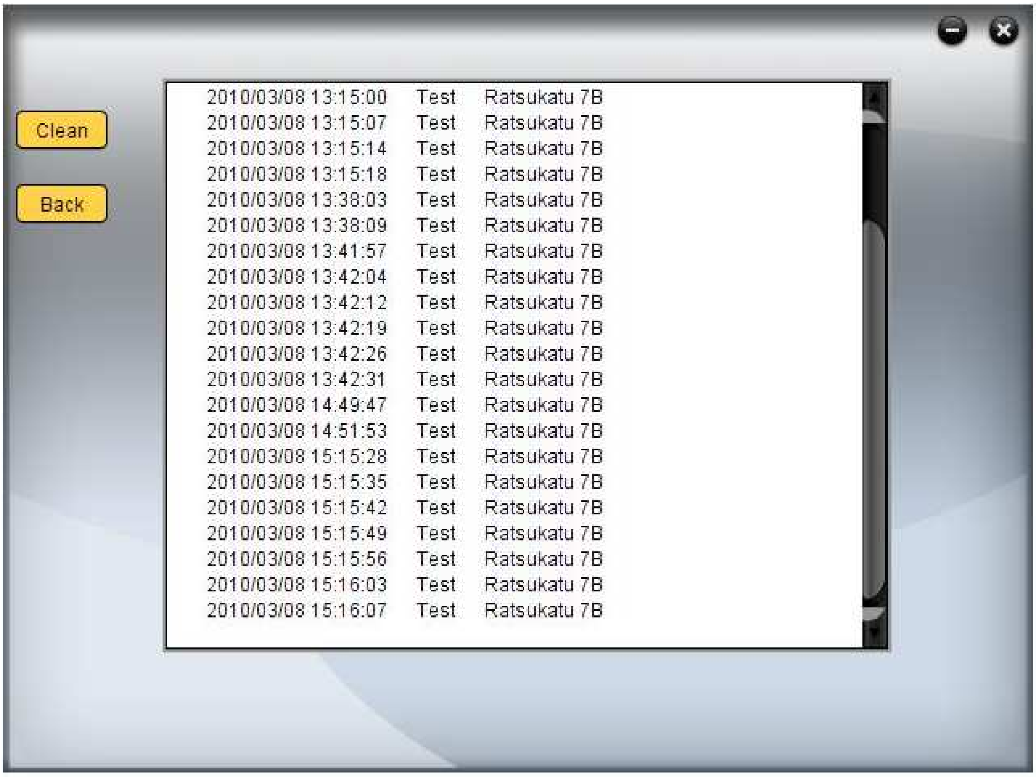


Figure 24. Exit menu option

Figure 24 shows the exit menu option. The program can be switched into an invisible mode by clicking the icon on the status bar.

4.3.1.6 History panel

The history panel (Figure 25) shows the information on received calls. It contains the following information: identity, home address and call time. The "Clean" button erases all the information. Again, the “Back” button is used to navigate away from the history panel.



```
2010/03/08 13:15:00   Test   Ratsukatu 7B
2010/03/08 13:15:07   Test   Ratsukatu 7B
2010/03/08 13:15:14   Test   Ratsukatu 7B
2010/03/08 13:15:18   Test   Ratsukatu 7B
2010/03/08 13:38:03   Test   Ratsukatu 7B
2010/03/08 13:38:09   Test   Ratsukatu 7B
2010/03/08 13:41:57   Test   Ratsukatu 7B
2010/03/08 13:42:04   Test   Ratsukatu 7B
2010/03/08 13:42:12   Test   Ratsukatu 7B
2010/03/08 13:42:19   Test   Ratsukatu 7B
2010/03/08 13:42:26   Test   Ratsukatu 7B
2010/03/08 13:42:31   Test   Ratsukatu 7B
2010/03/08 14:49:47   Test   Ratsukatu 7B
2010/03/08 14:51:53   Test   Ratsukatu 7B
2010/03/08 15:15:28   Test   Ratsukatu 7B
2010/03/08 15:15:35   Test   Ratsukatu 7B
2010/03/08 15:15:42   Test   Ratsukatu 7B
2010/03/08 15:15:49   Test   Ratsukatu 7B
2010/03/08 15:15:56   Test   Ratsukatu 7B
2010/03/08 15:16:03   Test   Ratsukatu 7B
2010/03/08 15:16:07   Test   Ratsukatu 7B
```

Figure 25. History panel view

4.3.1.7 Call

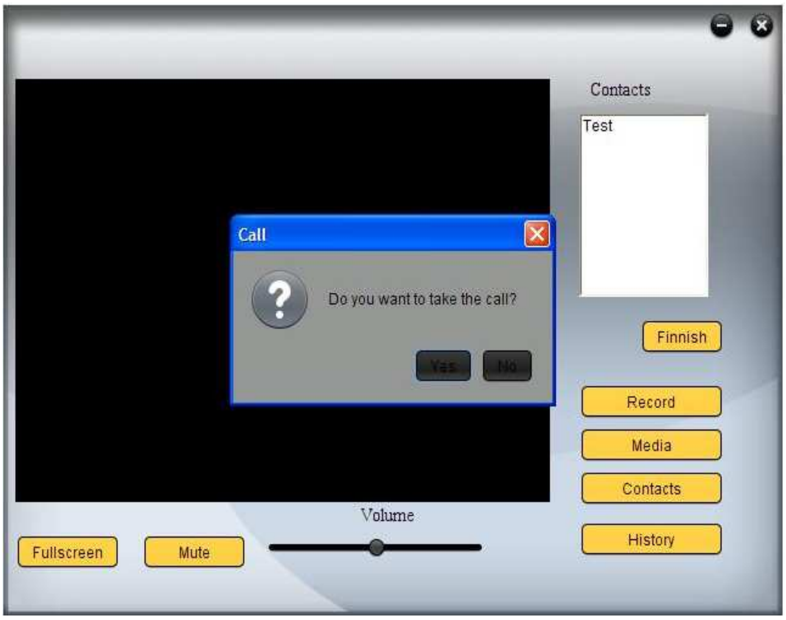

Do you want to take the call?

Figure 26. Notice call

A user calls the operator who receives a notification (Figure 26). The operator has an option to choose 'Yes' and start the conversation, or choose "No" and the client will be prompted that the operator is busy.

4.3.2 Client side

4.3.2.1 For IP-TV

Once the video broadcasting starts, the user receives a notification message (Figure 27) with an audio signal on the desktop. Notification box will be automatically dismissed after 5 minutes if the user doesn't click the "Ok" button.

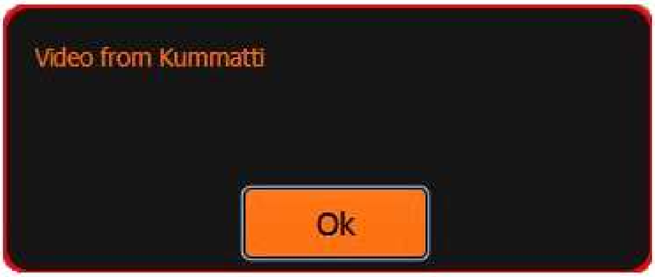

Video from Kummatti

Figure 27. Notification message

In order to call the operator, the user must switch on the camera (Figure 28). Once the camera's lens highlights green, the video and audio connection is ready for communication. Ending a call is done by switching the camera off.

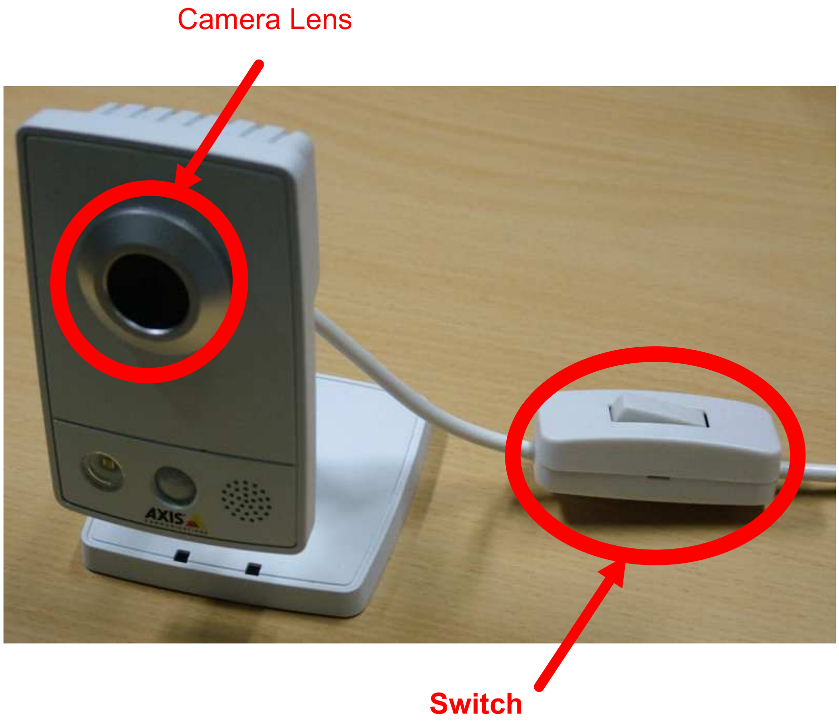
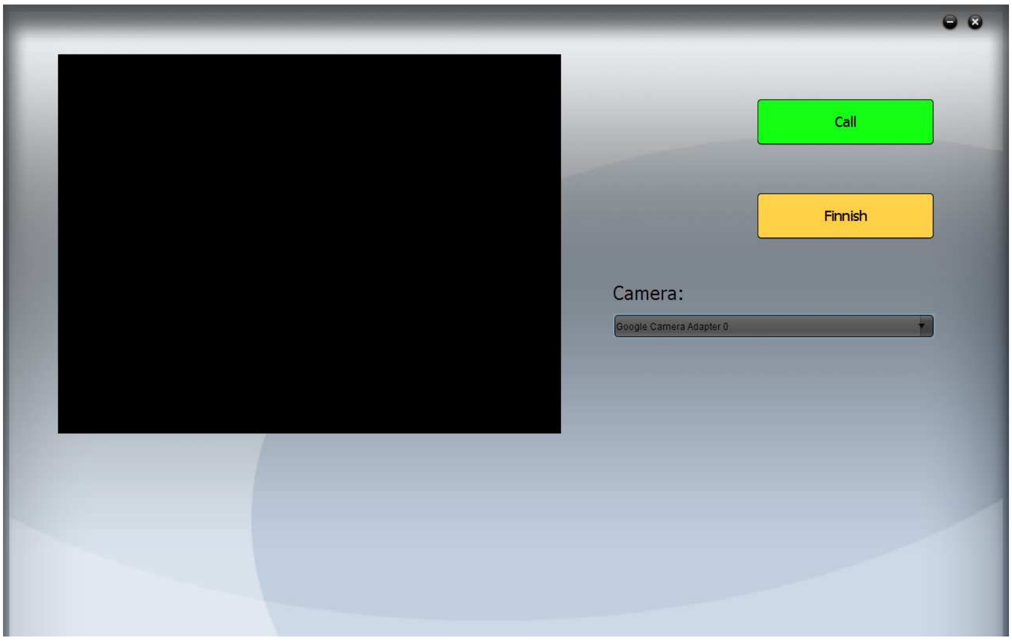


Figure 28. User’s camera with the switch

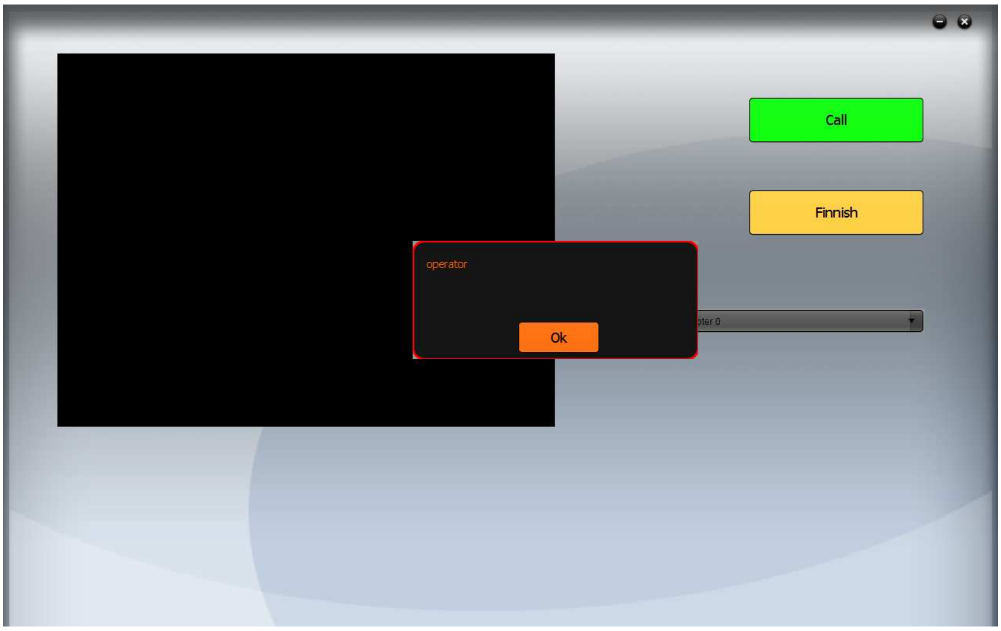
4.3.2.2 For desktop



Camera: Google Camera Adapter 0

Figure 29. The desktop version of the client program

The Figure 29 shows the desktop version of the client’s side of the system. The “Call” button is used to call the operator. The “Finnish” (“English”) button is for changing the current language of the user interface. When the operator calls a user, the client receives a notification message (Figure 30).



operator

Figure 30. Notification message

4.4 Database design

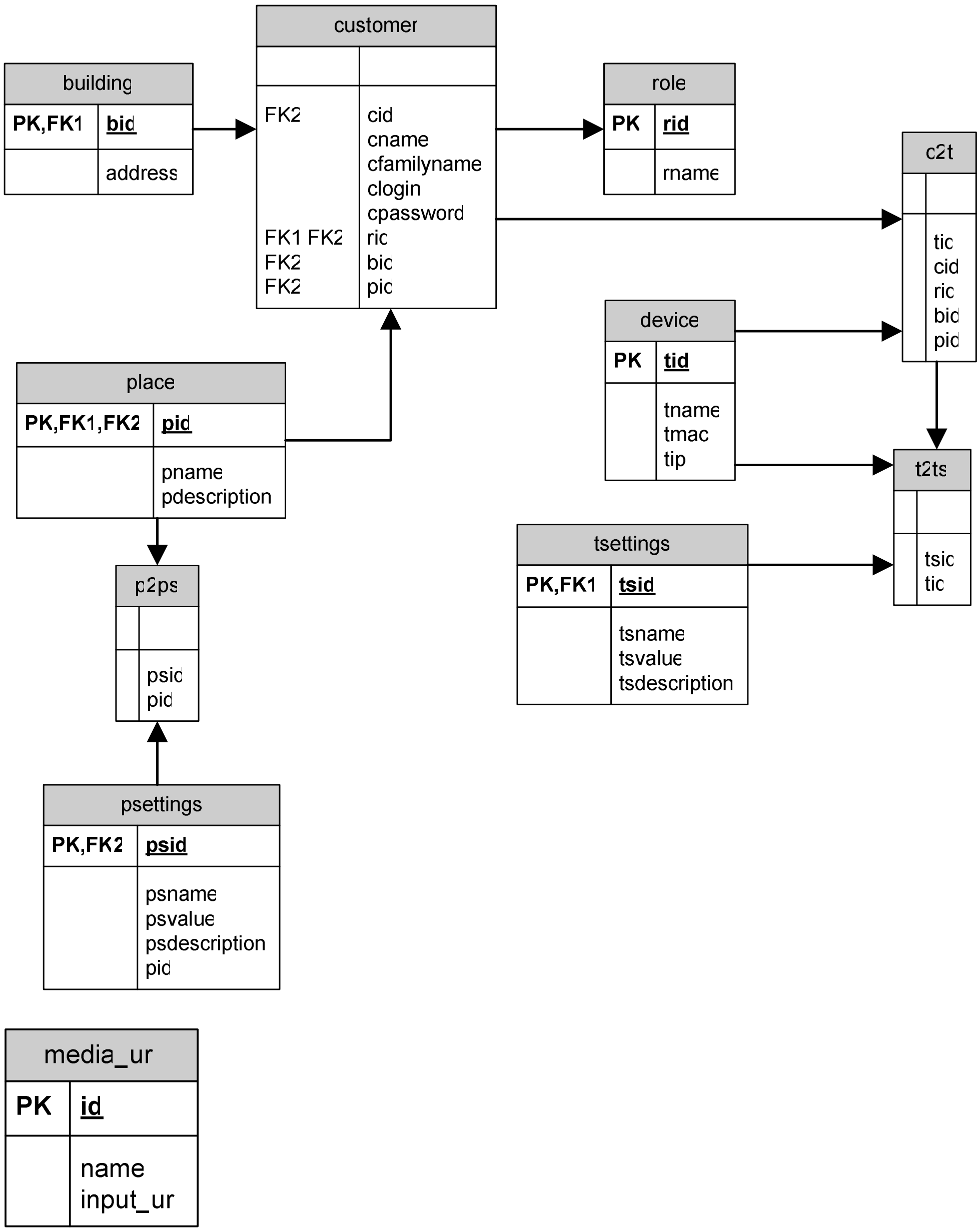
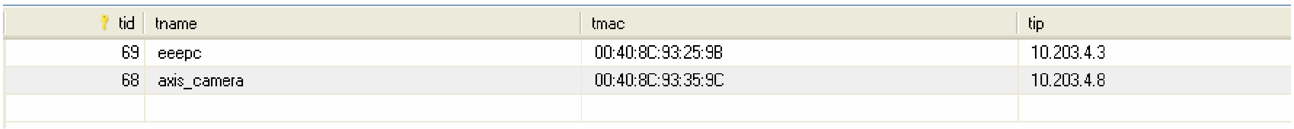


Figure 31. System “Auth” database

The figure 31 describes the database design of the system. At the moment not every table is used. The table “building” stores user addresses, whereas the table “customer” stores information about users (name, family name, login, and password).

Table “device”



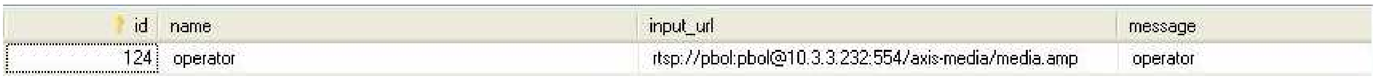
tid	tname	tmac	tip
69	eeepc	00:40:8C:93:25:9B	10.203.4.3
68	axis_camera	00:40:8C:93:35:9C	10.203.4.8

The table “device” stores technical information about users’ equipment (MAC and IP addresses)

EeePC - client’s computer. It stores MAC and IP addresses of the client’s server. It is used as user’s notice board, when the operator wants to speak with a client or a client calls the operator and the operator is busy.

Axis_camera - client’s camera. Used for video communication.

Table “media_urls”



id	name	input_url	message
124	operator	rtsp://pbol:pbol@10.3.3.232:554/axis-media/media.amp	operator

This table stores broadcast elements.

name - the name of multimedia source

input_url - address of a media file

message – user prompted message

4.5 Preferences platform

The system properties are stored in a file called “props.properties” on the services company’s server and the flat server.

4.5.1 Office server

```
props.properties
#Video system settings
// Port, which is used to receive MAC and IP addresses from customers.

port = 2434

//  URL is used to connect to the “auth” database

auth_db_url = jdbc:mysql://localhost/auth?user=user&password=pass

//URL  is used to connect to the database’s “dbconnector” table

queries_db_url = jdbc:mysql://localhost/dbconnector?user=user&password=pass

// Call log file

log_file = C:\\\\UbiAtHome\\\\log.txt
```

4.5.2 Flat server

props.properties

```
#Video system settings

// File for sound notification

incall_sound = E:\\bicycle_bell.wav

// notification delay

incall_show_delay = 300000


// Port for customer notification

incall_port = 535

#IPNotifier settings

// Office server IP address

server_address = 10.225.2.5

// Port, which use to send MAC and IP addresses

server_port = 3434
```

4.6 Communication protocol between the server and the users

The system utilizes two methods for communication between the server and users. The first one is used when the client uses an IP camera, whereas the second one suits best the equipment with a built-in or USB interfaced camera. Let’s see the first method in more details:

4.6.1 IP camera

The system is responsible for determining the IP addresses of the equipment without a static IP address. This is called a “Hare-Wolf” system. Another of its functionality is to determine the state of the camera: is it currently turned on or off. The camera is activated and deactivated manually by using a switch. The main advantage of this solution is security, because the camera can be disconnected at the hardware level, ensuring that nobody can access it while it is switched off. As a disadvantage, the camera takes too long time to boot.

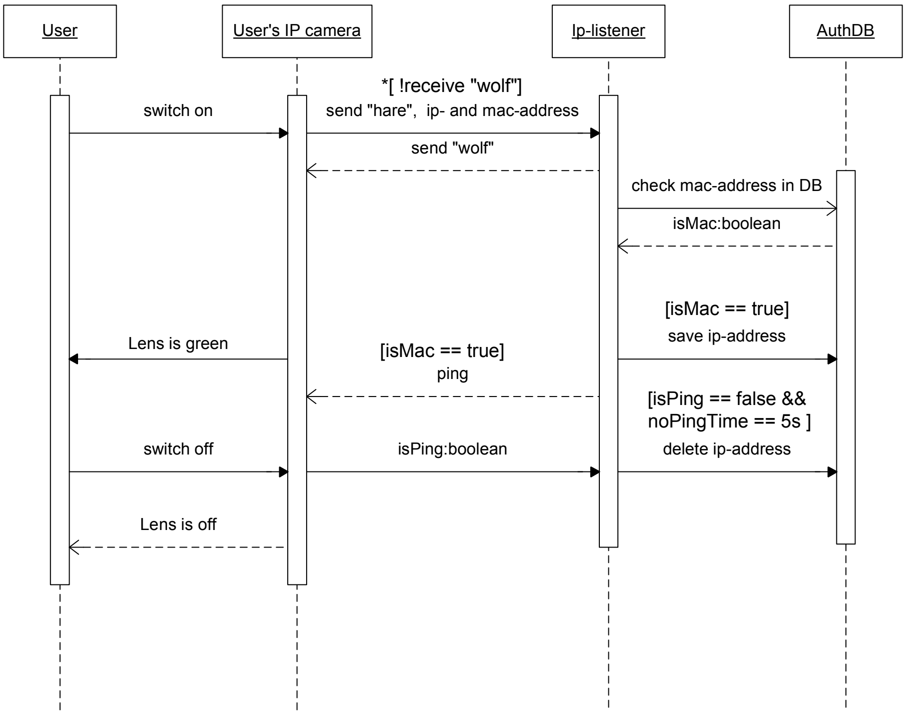


Figure 32. The sequence diagram of the “Hare-Wolf” system

The “Hare-Wolf” system contains two sides: a server and a client. The clients can have dynamic IP addresses, but the main server, which is running the IP listener process, must have a static IP. Using the software on operator’s side, she or he can edit the list of clients. The customer table in the database contains the login name, street address and MAC addresses for camera and flat server. The operation of the system can be described as the following (Figure 32):

Once a client switches on the flat server or the camera, it sends its MAC address and IP address to the main server. This represents the “hare” signal. On the other side, the main server checks whether the client with this particular MAC address is already in the database or not. If true, then the received IP address is stored in the database for the corresponding client. Basically, if the IP address belongs to an IP camera, the main server starts to ping this device and sends the “wolf” signal. When the device receives this signal, it initializes itself, switching the camera on. The green lamp indicates that the connection is now on and the conversation can be started.

If the client with this MAC address is not in the database, the main server waits for the next message. While a connection with the device breaks, its IP address in the database is replaced with "-", which means the device is currently off. To summarize the above, if the database server recognizes the address of an IP camera, then the camera is turned on and vice versa.

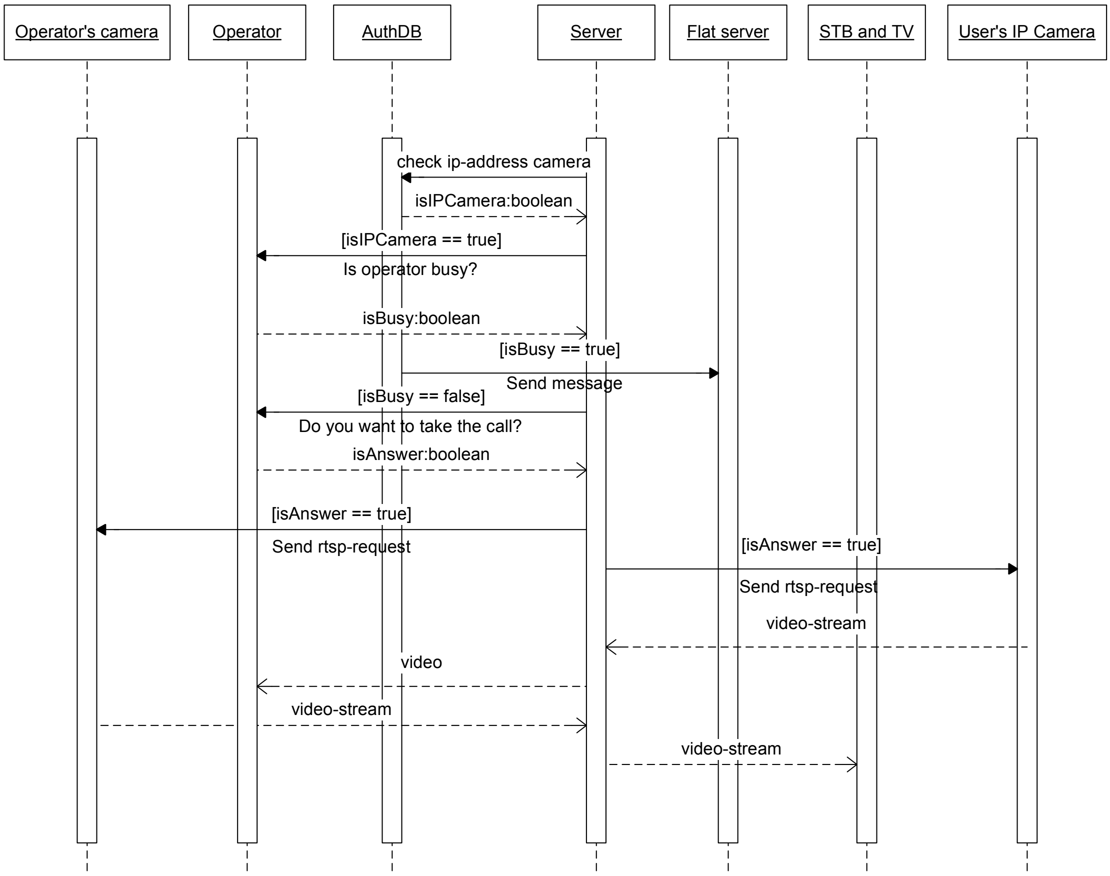


Figure 33. Sequence diagram shows how a user with an IP camera calls the operator

The server side of the “Hare-Wolf” system continuously monitors the camera’s IP address. If the database already contains the address, the operator receives a prompt with “Is operator busy?” message. If the operator chooses “Yes”, a message “Operator is busy” is sent to the customer. Otherwise, the video and video communication is started. This process is shown on the Figure 33.

The IP camera’s operating system is based on Linux OS and supports a scripting language. Scripting is a quick and very useful method to increase camera’s functionality. Therefore, the client-side of the system for the IP-camera is written in the scripting language.

Once the user switches on the camera, a script (see Appendix 2) sends MAC and IP addresses to a remote server.

A built-in application, parhandclient, can be used to get, set, add and remove parameters, for example:

parhandclient get root.Network.Interface.I1.Active.MACAddress – gets a MAC address from a network interface and sends it using the NC application to the server.

NC

NC or netcat is a common Linux command which is valuable for event handling. It reads and writes data across network connections, using the TCP or UDP protocol. NC can be used to send a string from a product to a specific port on a remote host.

Syntax

nc [options] [ip] [port]

Open a pipe to ip:port.

Options

-4 Use IPv4 (default).

-6 Use IPv6.

-u UDP mode.

-p port Local port number.

-i secs Delay interval for lines sent.

-w secs Timeout for connects and final net reads.

-D dscp Set IP dscp field.

-P prio Set VLAN user priority.

Script that controls the lens highlighting colour is described in Appendix 3.

The script uses the statusled command to change the lens highlighting colour.

statusled

Syntax

statusled [off|green|red|yellow|flash [INTERVAL]|init [INTERVAL]|usage [INTERVAL]]

Set the status-LED.

The indicator flashes and briefly displays orange during a start-up and self-test routines. After this, the indicator displays green to indicate a healthy unit status. Red will be displayed only if a problem has occurred.

Application nc listening port 4444 and when receiving:

“green” status-LED becomes green

“red” status-LED becomes red

“yellow” status-LED becomes yellow

Method that was used to obtain MAC addresses is described in Appendix 4.

4.6.2 Built-in or USB camera

The second method for communication between the server and the users utilizes a separate protocol. It was designed for the clients who are using built-in or USB cameras. This method is based entirely on the software implementation and communication between a client and the server by sent requests. The advantage of this method is the speed. The waiting time between the response from the user to the call and the time a conversation can actually be started, is minimized. This method is well-suited for the mobile phones or computers with a built-in camera.

The user clicks the “Call” button and the client side of the system sends the server a request to begin a conversation. If the operator agrees to accept the call, the server sends a request to the operator’s camera and to the client side for starting to send video. This process is shown on the Figure 34 above.

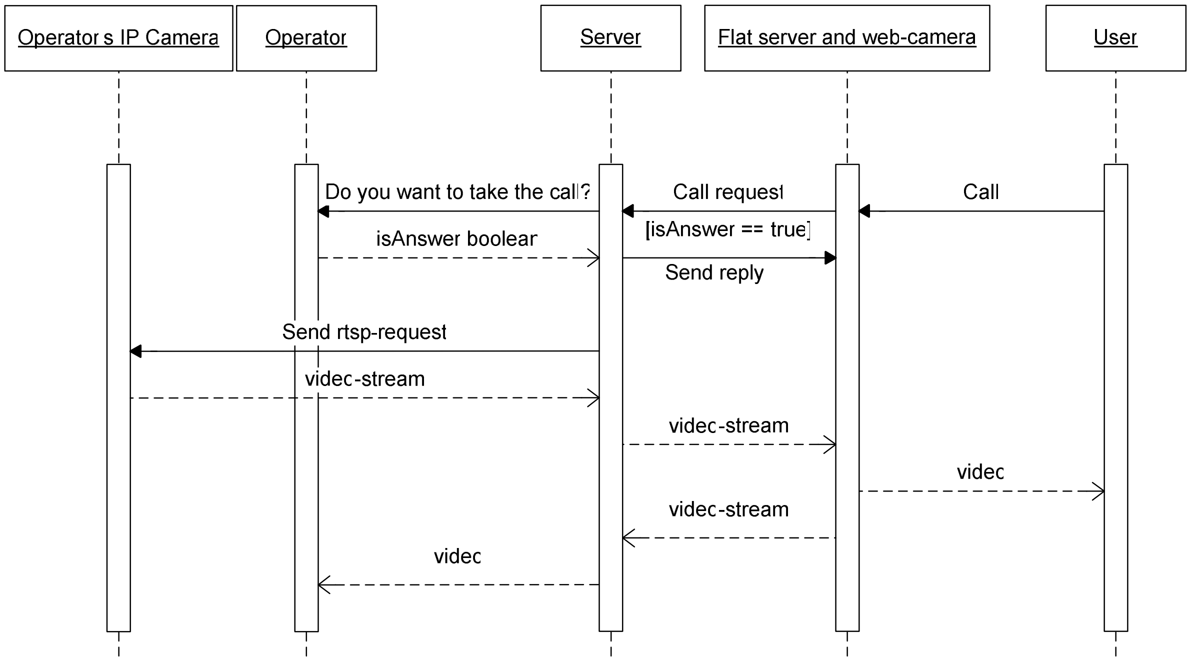


Figure 34. Sequence diagram shows how a user with a USB or built-in camera calls the operator

The dsj library was used to gain access to audio and video capturing devices. The dsj is a project to provide a Java wrapper around Microsoft's DirectShow API.

```
DSFilterInfo[] filters = null;

//   getFilters ( int category )
//   Returns a FilterInfo array for installed filters in the given category “VideoInputDevices”

filters = DSEnvironment.getFilters( 7 );

for (int y = 0; y < filters.length;y++){

          log.debug("filter "+y+": "+  filters[y].getName());

 }
```

Figure 35. The code for obtaining a list of capturing devices:

The Figure 35 shows the source code used to obtain names of all the capturing devices. It is possible to select a preferred capturing device from a list.

For example, for a PC with the Logitech QuickCam Express USB camera, the resulting output will be as follows:

```
19.44.33 (MainPanel): filter 0: Google Camera Adapter 0

19.44.33 (MainPanel): filter 1: Google Camera Adapter 1

19.44.33 (MainPanel): filter 2: Logitech QuickCam Express
```

Similar results, but displayed by using a graphical user interface (GUI) are shown on the Figure 36.

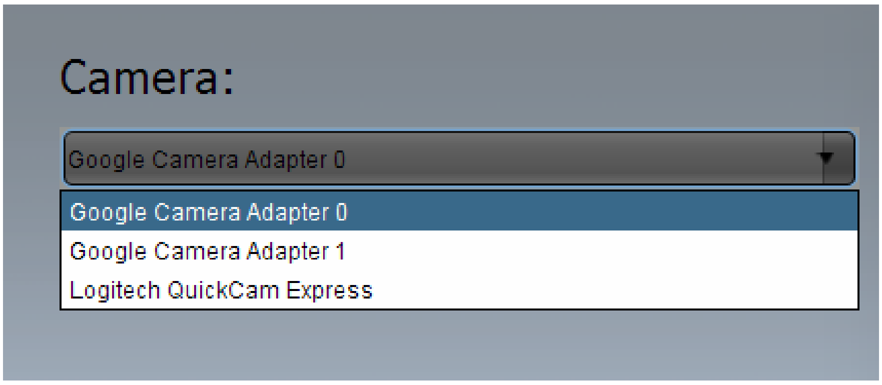


Figure 36. Capturing devices

4.7 Look and Feel

“Nimbus a cross platform Look and Feel was introduced in the Java SE 6 update 10 release. It uses a Java 2D vector graphics to draw a user interface (UI), rather than static bitmaps, so the UI can be rendered at any resolution. One of the biggest advantages of Nimbus is that it is highly customizable. The user can change the look in almost any way he or she can imagine”. [41] The user can:

- “Resize a component – Components are available in three additional sizes: mini, small, and large.
- Change the colour theme – You can change any of the colours used in the Nimbus look and feel.
- Skin a component – You have a full control over how a component is rendered”. [41]

The Nimbus Look and Feel has a set of default colours, but the user can use an own colour scheme.

All of the colours used by Nimbus are stored as a set of UIManager properties. The user can change any or all of these properties before setting the Look and Feel.

The Nimbus Look and Feel was allowed to change the Look and Feel of the graphical user interface. The Look and Feel is a term used in respect of a graphical user interface and comprises aspects of its design, including elements such as colors, shapes, layout, and typefaces (the "look"), as well as the behavior of dynamic elements such as buttons, boxes, and menus (the "feel")”. [42]

The source code for activating Nimbus Look and Feel and changing colour scheme:

```
// Changing the Color Theme

    UIManager.put("nimbusBase", new Color(0, 0, 0));

    UIManager.put("control", new Color(149, 151, 149));

    UIManager.put("nimbusBlueGrey", new Color(0, 0, 0));

// Set the  look and feel

    UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
```

4.8 Using JVLC

The JVLC library was used to play video data. The code for playing a video from a multicast address udp://@239.255.1.1:1111 is described in Appendix 5.

Variable “arg” contains VLC options.

```
String[] arg = new String[4];

    arg[0] = "--intf=dummy";

    arg[1] = "--ignore-config";
```

```
arg[2] = "--no-plugins-cache";

arg[3] = "--plugin-path=plugins";
```

VLC options:

--intf=<string> Interface module. This is the main interface used by VLC. The default behaviour is to automatically select the best module available.

--ignore-config No configuration option will be loaded not saved to a config file (defaulty enabled)

--no-plugins-cache Use a plugins cache (defaultly enabled), which will greatly improve the startup time of VLC (Defaultly enabled).

--plugin-path=<string> Modules search path. Additional path for VLC to look for its modules. You can add several paths by concatenating them by using"PATH_SEP" as a separator

The "Playlist" object was used to add media via the "Add" method, and to play a media file via the "Play" method.

5. TESTING

“Usually, quality is constrained to such topics as correctness, completeness, security, but can also include more technical requirements as described under the ISO standard ISO/IEC 9126, such as capability, reliability, efficiency, portability, maintainability, compatibility, and usability”. [11]

5.1 Testing methods

Software testing consisted of the following levels:

5.1.1 Unit testing

JUnit is a framework for implementing tests. It is written in Java by Erich Gamma and Kent Beck. “It provides a simple way to explicitly test specific areas of a Java program. Test increases programmer’s productivity and stability of program code whilst reducing programmer’s stress and the time spent debugging”. [43] Once the writing of tests using JUnit is finished, there is confidence that the changes on the source code actually work. This confidence allows you to get more aggressive on refactoring code and adding new features. JUnit promotes the idea of “first testing then coding”. In JUnit, it’s to setup test data for a unit, which defines the expected output and the code until the tests pass. In object-oriented systems, these units typically are classes and methods. JUnit tests allow one to write code faster while increasing quality.

5.1.2 Alpha testing

“Alpha testing: Simulated or actual operational testing by potential users/customers or an independent test team at the developers’ site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of testing for internal acceptance”. [13]

At the end of the development process the application has been tested for errors. One of the issues was related to the video transfer. Video was transmitted with delays. The replacement of MJPEG video compression with MPEG4/H.264 was decided upon. This solution increased the server’s capacity and the delay was decreased.

MJPEG sends a series of still, complete images. MPEG4/H.264 only sends the part of the image that is changing. Currently Motion JPEG offers a higher quality image. The MPEG4/H.264 image quality is lower. MPEG4/H.264 uses more compression and results in less bandwidth consumption, and if recording, requires less disk space.

5.1.3 Beta testing

“Beta testing: Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for off-the-shelf software in order to acquire feedback from the market”. [13]

In order to acquire a feedback the system was set up for beta testing in a real apartment, provided by the Kummatti Ltd. housing company. A tenant will explore the system’s functionality during the following year and report possible faults and opinions. One part of the system was set up for an operator in the company's office and another one - in a populated flat. The operator will broadcast some information to an inhabitant. If the inhabitant will require some service, he or she will make a call to the operator.

During the testing stage the system was refined and the following issue was observed: When a tenant calls the operator, the sound signal is triggered. The server side of the system automatically answers the call and the communication starts. But if the operator didn't heard the sound signal or was away at the moment of a call, the tenant had an access to the operator's camera and could see what is happening in the office. This issue was corrected by adding an extra functionality. So now, when the tenant calls the office, the operator is receiving a prompt question "Do you want to answer". If confirmed, the tenant is able to see and hear the operator.

5.1.4 Stability testing

"Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of non-functional software testing is often referred to as load (or endurance) testing". [13]

Stability tests are carried out by leaving the system working for several days. At the end of the test the system must still be operational.
The first stability test of the system has failed because of the lost connection to the database. The problem was solved and the following tests passed successfully.

5.1.5 Usability testing

Usability testing is needed to check if the user interface is easy to use and understand. [13]

The system was shown to several people in order to receive a feedback how system's user interface must be improved for better understanding. The results of the tests have shown that the selected user interface was clear and easy to learn. The customer received a manual describing how the user interface should be used.

5.1.6 Internationalization and localization

"Internationalization and localization is needed to test these aspects of software, for which a pseudo localization method can be used. It will verify that the application still works, even after it has been translated into a new language or adapted for a new culture (such as different currencies or time zones)". [13]

The program's interface is localized in two languages – English and Finnish. The localization has been revised by the native speaking testers. The found defects in localization were corrected.

5.2 Test environment

Home equipment:

6. TV Sony Bravia KDL-40W4500
7. STB Motorola VIP1910
8. Touch screen panel Asus Eee Top PC 1602
9. IP camera AXIS M1031-W

Office equipment:

- IP camera AXIS M1031-W
- Server PC

Device name	Device picture	Description
TV Sony Bravia KDL-40W4500		**Product Type** LCD TV **Diagonal Size** 40 - widescreen **Resolution** 1920 x 1080 **Display Format** 1080p (FullHD) **Enhanced Refresh Rate** 100 Hz **Digital TV Tuner** DVB-C, DVB-T **Analogue TV Tuner** PAL, SECAM **MPEG Decoder** MPEG-4 **Sound Output Mode** Stereo **Speaker System** 2 speakers **Built-in Decoders** Dolby Digital **USB Port** Yes , 1 port(s) **Remote Control** Remote control - infrared **Video Interface** Component, composite, HDMI, SCART

Asus Eee Top PC 1602		**LCD** 15.6" 16:9 Wide Panel **CPU + Chipset** Intel Atom N270 + 945 GSE **Memory** DDR II 1GB **HDD** 160G SATAII 5400rpm **Graphics** On board graphics **Build Camera** 1.3M pixel Web camera **Mic** Array Mic **LAN** 10/100/1000 Mbps **Wireless** 802.11 n **Audio chip** 4W Hifi speaker x 2 + SRS Premium Sound System **IO port** 3 audio ports for 5.1channel, Gigabit LAN x 1, USB 2.0 x 4
AXIS M1031-W		**Video compression** H.264 (MPEG-4 Part 10/AVC), Motion JPEG MPEG-4 Part 2 (ISO/IEC 14496-2) **Resolutions**: 640x480 to 160x120 **Audio compression** AAC-LC 8/16 kHz, G.711 PCM 8kHz, G.726 ADPCM 8 kHz **Wireless interface:** IEEE 802.11g/b Invisibly integrated antenna **Security** Password protection, IP address filtering, HTTPS** encryption, digest authentication, user access log WEP 64/128 bit, WPA/WPA2-PSK **Supported protocols** IPv4/v6, HTTP, HTTPS**, QoS Layer 3 DiffServ, FTP, SMTP,Bonjour, UPnP, SNMPv1/v2c/v3(MIB-II), DNS, DynDNS, NTP,RTSP, RTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS

STB Motorola VIP1910		**Hardware** Microprocessor RISC processor with integrated TV video processing and MPEG decoder Main memory 128 MB DRAM Video memory 32 MB DRAM Flash memory 4 MB Accessories Infrared remote control Infrared keyboard (optional) **Operating systems** Linux **Streaming video** MPEG-1 (SD) MPEG-2 (HD and SD H.264 (HD and SD **Video on demand** Interactive Video-on-Demand (IVOD) and Near Video-on-Demand (NVOD) Protocol RTSP Verified compliance with leading video server suppliers **Streaming audio** Audio types MPEG (Layer 1,2,3), PCM, AC3 , AAC

6. FUTURE DEVELOPMENT

An essential target for a future development is a SIP server and adaptive clients for the SIP protocol. That will allow the control of multimedia communication sessions and creation of multiparty (multicast) sessions consisting of one or several media streams. The SIP protocol is a standard and usually it is used for VoIP. This protocol allows the use of other applications supporting VoIP. That gives an opportunity for clients to interact with each other, which is not supported in the current version of the system.

The speed of mobile networks and the powerful processors in mobile devices allow nowadays the transfer of video and audio over such networks. Another idea for the future development is to implement a mobile version of the client (Figure 37). This will make the use of the system possible virtually anywhere.

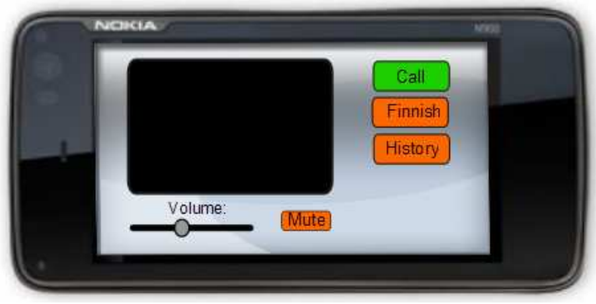


Figure 37. The mobile version of the client side of the system

Another idea for future development is to develop a web version of the client. “Web-based VoIP is based on the concept of click to talk, which is a form of Web-based communication in which a person clicks an object (i.e. button, image, or text) to request an immediate connection with another person in real-time either by phone call, Voice-over-Internet-Protocol (VoIP), or text”. [24]

Another important direction in development is to make the system more secure. The video and audio stream from the Axis camera is protected by a password. The camera requires user authentication (via a name and password) before a session can be streamed.

Once the stream is converted to another format, the traffic becomes unsecure.

New features could be added to the system, e.g. the Video on Demand (VOD) access to videos streamed from the operator. While the operator initializes a video broadcast, the data is stored on the server and shared to all customers. If a customer has missed the video content broadcasted by the operator, it will be possible to demand that video from the server at anytime.

One more idea for the future development is to split the functionality to support:

- Voice-only communication
- Multimedia communication

7. CONCLUSION

My thesis work was finished with better results than expected. I developed a system that enables a video communication between the representatives of a house-holding company or other service companies (including social and health-care) and building inhabitants. At the moment the functionality of the system allows a video-calling between a single client and the company's representative and a video-multicasting from the services company to all its clients and a return call from any of the user. The few cases demanding a further development are the video calls between tenants and videoconferencing - a functionality that enables the service company to receive simultaneous calls from multiple users in one session. A working prototype of the system was implemented and installed into a real environment. [38] Initial tests have passed, and now the system is ready to be tested in real life conditions. During the following year the system will be tested as a part of a living Lab by a person living in those apartments.

The idea of how to accomplish the task came into my mind during the research process**.** I inspected and tested many public Java libraries for handling the video data before I selected JVLC. A selected approach that is presented in this document solved the issue of playing a video in a Java-based application and allowed to broadcast and decode video and audio streams.

The process of developing a video-communication system was very exciting as well as very challenging. However, it was a very interesting experience. During the thesis project I significantly improved my professional skills in Java programming language and learned how IPTV systems work on the Hibox platform. Also, I have reviewed the protocols for the transfer and control of a video stream. I believe that in my further professional life I can make use of the skills acquired during this project.

8. LIST OF REFERENCES

1. Real Time Streaming Protocol
   http://www.ietf.org/rfc/rfc2326.txt
   Last access day: 26 October 2009
2. Specification sheet STB Motorola VIP1910
   http://www.motorola.com/staticfiles/Business/Products/TV%20Video%20Distribution/Set-tops/IP%20Set-tops/VIP1920/_Documents/Static%20Files/VIP1900-9_specsheet_090508.pdf?localeId=33
   Last access day: 23 August 2010
3. Product overview Axis M1031-W
   http://www.axis.com/products/cam_m1031w/
   Last access day: 23 August 2010
4. Datasheet Axis M1031-W
   http://www.axis.com/files/datasheet/ds_m10_37647_en_1005_lo.pdf
   Last access day: 23 August 2010
5. Product overview EeeTop PC ET1602
   http://usa.asus.com/product.aspx?P_ID=XPEvtodKRTfnQbCm
   Last access day: 23 August 2010
6. Forum about products from ORACLE
   http://forums.sun.com
   Last access day: 11 June 2010
7. HiBox platform overview
   http://hibox.com
   Last access day: 23 August 2010
8. SIP protocol
   http://www.voip-info.org/wiki/view/SIP
   Last access day: 17 April 2010
9. Session Initial Protocol communication example
   http://www.en.voipforo.com/SIP/SIP_example.php
   Last access day:
10. Session Initial Protocol
    http://www.ietf.org/rfc/rfc3261.txt
    Last access day: 7 September 2010
11. Software testing
    http://wapedia.mobi/en/Software_testing?t=9.
    Last access day: 30 August 2010
12. ISO 9126 Software Quality Characteristics
    http://www.sqa.net/iso9126.html
    Last access day: 30 August 2010
13. Standard glossary of terms used in Software Testing
    http://www.astqb.org/educational-resources/glossary.php#A
    Last access day: 30 August 2010
14. NetBeans Overview
    http://netbeans.org/
    Last access day: 29 Mart 2010
15. Wireshark Overview
    http://www.wireshark.org/
    Last access day: 15 August 2010
16. VLC Overview
    http://www.videolan.org/vlc/
    Last access day: 10 August 2010

17. Jffmpeg Overview
http://jffmpeg.sourceforge.net
Last access day: 12 August 2010
18. Dsj Overview
http://www.humatic.de/htools/dsj.htm
Last access day: 12 August 2010
19. Modifying the Look and Feel
http://v1.dione.zcu.cz/java/docs/tutorial/uiswing/lookandfeel/index.html
Last access day:
20. VLC command-line help
http://wiki.videolan.org/VLC_command-line_help
Last access day: 12 August 2010
21. IPTV Standardization on Track Say Industry Experts
http://www.itu.int/ITU-T/newslog/IPTV+Standardization+On+Track+Say+Industry+Experts.aspx
Last access day: 25 July 2010
22. ATIS IPTV Exploratory Group Report and Recommendation to the TOPS Council
http://www.atis.org/tops/IEG/ATIS_IPTV_EG_RPT_final.pdf
Last access day: 25 July 2010
23. IPTV
http://transanatolia.eu/Analyses/Fixed%20Access%20Networks/iptv.htm
Last access day: 7 September 2010
24. Web-based VoIP
http://en.wikipedia.org/wiki/Web-based_VoIP
Last access day: 21 Mart 2010
25. Mobile VoIP
http://www.cs.dartmouth.edu/~dfk/papers/mills-tettey-mvoip.pdf
Last access day: 2 September 2010
26. Constructivist Research and Info-Computational Knowledge Generation
http://www.idt.mdh.se/projects/PIFF/ConstructiveResearchParadigm.pdf
Last access day: 3 September 2010
27. Vadim Kramar, Jouni Kivirinta: Video-Communication System Requirements Specification.
28. Voice over Internet Protocol
http://www.fcc.gov/voip/
Last access day: 4 Mart 2010
29. Videoconferencing
http://www.worldlingo.com/ma/enwiki/en/Videoconferencing
Last access day: 11 December 2009
30. Middleware in IPTV
http://world-of-iptv.com/middleware.php
Last access day: 8 September 2010
31. KreaTV platform
http://www.motorola.com/business/v/index.jsp?vgnextoid=f0319e3eeed46110VgnVCM1000008406b00aRCRD&vgnextchannel=d814a608774b6110VgnVCM1000008406b00aRCRD
Last access day: 8 September 2010
32. Java Media Framework API
http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html
Last access day: 12 August 2010
33. Hibox Systems
http://www.hibox.tv/en/docs/Hibox-Product_brochure.pdf
Last access day: 5 September 2010

34. VideoLAN Manager
http://www.videolan.org/doc/streaming-howto/en/ch05.html
Last access day: 5 September 2010

35. Java Media Framework Overview
http://www.oracle.com/technetwork/java/javase/index-135334.html
Last access day: 12 August 2010

36. FFMPEG Overview
http://ffmpeg.org/
Last access day: 8 September 2010

37. Set-top box
http://www.iptvinformation.net/CategoryView,category,STB.aspx
Last access day: 13 September 2010

38. Article “Apu tulee eteisen seinästä” in the newspaper ”Raahen seutu”
http://www.raahenseutu.fi/cs/Satellite?c=AMArticle_C&childpagename=RSE_newssite/AMLayout&cid=1194637980450&p=1194615915250&pagename=RSEWrapper&rendermode=previewnoinsite
Last access day: 12 September 2010

39. IPTV
http://www.nevron.eu/support/knowledge-library/abc-of-iptv
Last access day: 11 September 2010

40. Streaming video
http://www.edb.utexas.edu/minliu/multimedia/Streaming%20Video.pdf
Last access day: 14 September 2010

41. Look of Nimbus
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/custom.html
Last access day: 17 September 2010

42. Look and Feel
http://nextgencon.com/about/smart-client-look-and-feel/
Last access day: 11 August 2010

43. JUnit testing utility tutorial
https://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/junit/junit.pdf
Last access day: 3 May 2010

44. Clear Streaming
http://www.cleartechnology.com/clear-streaming.html
Last access day: 21 September 2010

45. RTCP protocol
https://supportforums.cisco.com/docs/DOC-1116
Last access day: 2 August 2010

46. Jffmpeg
http://jffmpeg.sourceforge.net/
Last access day: 2 September 2010

47. Java
http://java-work.blogspot.com/2008/08/java-advantages-and-disadvantages.html
Last access day: 19 August 2010

48. MySql connector
http://www.mysql.com/products/connector/
Last access day: 27 August 2010

9. APPENDICES

APPENDIX 1

THE SOURCE CODE FOR A MULTICAST VIDEO

```
String output = "#transcode{acodec=mp3,vb=128, samplerate=48000 ,canals=2, vcodec =”+
“h264,vb=2000,scale=1 }:std{access=udp,mux=ts,dst=239.235.1.1:1111}";

// String input -  input stream

// String name – the name of the new broadcast

public void AddBroadcast(String name, String input) {

      vlm.addBroadcast(name, input, output, null, true, false);

      vlm.setMediaOutput(name, output);

     // Pause 1 s

       try {

            Thread.sleep(1000);

        } catch (InterruptedException ex) {

              Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);

         }

        vlm.playMedia(name);

  }

         } else {

            log.debug("Network Interface is not found!");

         }

      } catch (Exception ex) {

         log.debug(ex);

      }

      return result.toString();

   }
```

APPENDIX 2

THE SCRIPT TO SEND A MAC ADDRESS FROM AN IP CAMERA TO A SERVER

```
#!/bin/sh

str1="hare#"

mac="00:40:8C:99:25:9E"

str5=";"

str4="wolf"

// do while don’t receive “wolf” from server

while [ "$str3" != "wolf" ]

do

// Get MAC address

  str6=`parhandclient get root.Network.Interface.I1.Active.MACAddress`

result=$str1$mac$str5

  echo $result

// send MAC address to server

  str3=`echo $result | nc 10.115.35.113 4444`

  echo $str3

done

echo "end programm"
```

APPENDIX 3

THE SCRIPT TO CONTROL THE COLOUR OF LIGHT AROUND LENS OF AN IP CAMERA

```
#!/bin/sh
// Read data from port 4444
 str=`nc -l -p 4444`
// If camera receives:
//  “green” – green color
//  “red” – red color
//  “yellow” – yellow color
 case $str in
     green) echo "green"
           statusled green
           sh /etc/my_script
           ;;
     yellow) echo "yellow"
           statusled yellow
           sh /etc/my_script
           ;;
     off) echo "off"
           statusled off
           sh /etc/my_script
           ;;
     red) echo "red"
          statusled red
          sh /etc/my_script
          ;;
  Esac
```

APPENDIX 4

THE METHOD USED FOR OBTAINING A MAC ADDRESS

```
private String getMAC() {
        StringBuffer result = new StringBuffer();
        try {
            InetAddress address = InetAddress.getLocalHost();
            NetworkInterface ni = NetworkInterface.getByInetAddress(address);
            if (ni != null) {
                byte[] mac = ni.getHardwareAddress();
                if (mac != null) {
                    for (int i = 0; i < mac.length; i++) {
                        result.append(String.format("%02X%s", mac[i], (i < mac.length - 1) ? ":" : ""));
                    }
              } else {
                    log.debug("Address does not exist!");
              }
            } else {
                log.debug("Network Interface is not found!");
            }
        } catch (Exception ex) {
            log.debug(ex);
        }
        return result.toString();
    }
```

APPENDIX 5

THE SOURCE CODE FOR PLAYING A VIDEO FROM A MULTICAST ADDRESS

```
/*      VLC options
*     --intf=<string>                  Interface module. This is the main interface used by VLC. The default
*                                      behaviour is to automatically select the best module available.
*     --ignore-config                  No configuration option will be loaded nor saved to a config file (default enabled)
*     --no-plugins-cache               Use a plugins cache (enabled by default) Use a plugins cache,
*                                      which will greatly improve the startup time of VLC. (enabled by default)
*    --plugin-path=<string>            Modules search path.  Additional path for VLC to look for its modules. You can
*                                      add several paths by concatenating them using ”PATH_SEP” as separator
*/

       String[] arg = new String[4];

        arg[0] = "--intf=dummy";

        arg[1] = "--ignore-config";

        arg[2] = "--no-plugins-cache";

        arg[3] = "--plugin-path=plugins";

        JVLC jvlc = new JVLC(arg);

        Playlist playlist = new Playlist(jvlc);

//    Add item to playlist

        playlist.add("udp://@239.235.1.1:1111", " ");

// Plays the current item in the playlist.

        playlist.play();
```